Local and Global Explanations of Agent Behavior: Integrating Strategy Summaries with Saliency Maps

Tobias Huber[a], Katharina Weitz[a], Elisabeth Andr[a], Ofra Amir[b]

[a] *Universitt Augsburg, Universittsstrae 6a, Augsburg, Germany*
[b] *Technion - Israel Institute of Technology*

Abstract

With advances in reinforcement learning (RL), agents are now being developed in high-stakes application domains such as healthcare and transportation. Explaining the behavior of these agents is challenging, as the environments in which they act have large state spaces, and their decision-making can be affected by delayed rewards, making it difficult to analyze their behavior. To address this problem, several approaches have been developed. Some approaches attempt to convey the *global* behavior of the agent, describing the actions it takes in different states. Other approaches devised *local* explanations which provide information regarding the agent's decision-making in a particular state. In this paper, we combine global and local explanation methods, and evaluate their joint and separate contributions, providing (to the best of our knowledge) the first user study of combined local and global explanations for RL agents. Specifically, we augment strategy summaries that extract important trajectories of states from simulations of the agent with saliency maps which show what information the agent attends to. Our results show that the choice of what states to include in the summary (global information) strongly affects people's understanding of agents: participants shown summaries that included important states significantly outperformed participants who were presented with agent behavior in a randomly set of chosen world-states. We find mixed results with respect to augmenting demonstrations with saliency maps (local information), as the addition of saliency maps did not significantly improve performance in most cases. However, we do find some evidence that saliency maps can help users better understand what information the agent relies on in its decision making, suggesting avenues for future work that can further improve explanations of RL agents.

Keywords: Explainable AI, Strategy Summarization, Saliency Maps, Reinforcement Learning, Deep Learning

1. Introduction

The maturing of artificial intelligence (AI) methods has led to the introduction of intelligent systems in areas such as healthcare and transportation [69]. Since these systems are used by people in such high-stakes domains, it is crucial

for users to be able to understand and anticipate their behavior. For instance, a driver of an autonomous vehicle will need to anticipate situations in which the car fails and hands over control to her, while a clinician will need to understand the treatment regime recommended by an agent to determine whether it aligns with the patient's preferences.

The recognition of the importance of human understanding of agents' behavior, together with the complexity of current AI systems, have led to a growing interest in developing "explainable AI" methods [22, 29, 2]. The idea of making AI systems explainable is itself not new, and was already discussed since the early days of expert systems [71, 19]. However, state-of-the-art AI algorithms use more complex representations and algorithms (e.g., deep neural networks), making them harder to interpret. For example, in contrast to classical agent planning approaches such as the belief-desire-intention (BDI) framework [58] in which the goals of the agent are explicitly defined, current agents often use policies trained using complex reward functions and feature representations that are difficult for people to understand.

In this paper, we focus on the problem of describing and explaining the behavior of agents operating in sequential decision-making settings, which are trained in a deep reinforcement learning framework. In particular, we explore the usefulness of *global* and *local* post-hoc explanations [53] of agent behavior. Global explanations describe the overall policy of the agent, that is, the actions it takes in different regions of the state space. An example of such global explanations are strategy summaries [7], which show demonstrations of the agent's behavior in a carefully selected set of world states. Local explanations, in contrast, aim to explain specific decisions made by the agent. For instance, saliency maps are used to show users what information the agent is attending to [28].

We explore the combination of global and local information describing agent policies. The motivation for integrating the two approaches is their complementary nature: while local explanations can help users understand what information the agent attends to in specific situations, they do not provide any information about its behavior in different contexts. This is reinforced by a previous study conducted by Alqaraawi et al. [4] who evaluated local explanations and came to the conclusion that sole instance-level explanations are not sufficient and should be augmented with global information. Similarly, while demonstrating what actions the agent takes in a wide range of scenarios can provide users with a sense of the overall strategy of the agent, it does not provide any explanations as to what information the agent was considering when choosing how to act in a certain situation.

To examine the benefits of these two complementary approaches and their relative usefulness, we integrate strategy summaries with saliency maps. Specifically, we adapt the HIGHLIGHTS-DIV algorithm for generating strategy summaries from our previous work [5] such that it can be applied to deep learning settings, and integrate it with saliency maps that are generated based on Layer-Wise Relevance Propagation (LRP) (using a method we previously published in [36]). We combine these two approaches by adding to the summary generated by HIGHLIGHTS-DIV saliency maps showing what the agent attends to.

We evaluate this combination of global and local explanations in a user study in which we explore both the benefits of HIGHLIGHTS-DIV summaries and the benefits of adding saliency maps to strategy summaries. Specifically, we compare random summaries and HIGHLIGHTS-DIV summaries, both with and without the addition of saliency maps. Study participants complete two types of tasks requiring the analysis of different agents trained to play the game of Pacman: an agent comparison task in which they compare the performance of two agents, and a retrospection task, in which they reflect on an agent's strategy. We chose those tasks to investigate whether the users trusted the right agent and to evaluate their mental models of the agents, respectively.

Our results show that participants who were shown HIGHLIGHTS-DIV summaries performed better on both tasks compared to participants who were shown random summaries, and were also more satisfied with HIGHLIGHTS-DIV summaries. We find mixed results with respect to the benefits of adding saliency maps to summaries, which improved participants' ability to identify some aspects of agents' strategies, but in most cases did not lead to improved performance.

The paper makes the following contributions:

- It demonstrates that the HIGHLIGHTS-DIV algorithm, which was so far only used on classic reinforcement learning, can be applied to deep reinforcement learning agents with slight adjustments.

- It proposes a joint local and global explanation approach for RL agents by integrating LRP saliency maps and HIGHLIGHTS-DIV summaries.

- It evaluates the combination of global and local summaries in a user study, demonstrating the benefits of HIGHLIGHTS-DIV summaries and the potential benefits and limitations of local explanations based on saliency maps.

The remainder of this article is structured as follows: Section 2 reviews prior work on explainable intelligent agents, Sections 3 and 4 describe our previous works on local and global explanations, respectively. Section 5 details our combined implementation of those two methods, including the adaptation of HIGHLIGHTS-DIV to deep reinforcement learning. We describe the empirical evaluation we conducted in Section 6, and its results are summarized in Section 7. Finally, we discuss the results of the study and future directions in Section 8, and conclude in Section 9.

2. Related Work

In this section, we review related works on explainable AI. We begin with a short review of global and local explanation methods of machine learning models, elaborating on the use of saliency maps, which we also make use of. We then discuss in more depth prior works on global and local explanations of policies of agents operating in sequential decision-making settings such as RL agents.

Global and local methods for interpretable machine learning. Broadly, our work relates to the problem of interpretable machine learning, that is, explanations for the decisions of prediction models [22]. Few interpretable machine learning approaches provide global explanations, e.g., by showing examples of a set of instances and specifying how they were classified [59, 40] or by generating prototypical images that maximize the activation of specific neurons [66].

The majority of methods focus on local explanations that explain single decisions of the model. To this end, various methods to measure the relevance of a part of the input for the model's decision have been proposed. For visual input, this information is often displayed as saliency maps that highlight how relevant each pixel is for a particular decision of the agent. Since the input for the Atari agents we use in this study is visual, we will use the word saliency map method even if the very same algorithm can be used on non-visual input data.

Gradient-based saliency map generation methods [66, 68, 70, 64] utilize the derivative with respect to the input to estimate how much a small change in this input's value would change the prediction.

Occlusion-based methods ([81, 59, 62]), occlude areas inside the input and measure how much this changes the model's prediction. The idea behind this is to introduce uncertainty to the occluded area and to see how much the model is influenced by the loss of information in that area. Occlusion-based methods often come with the advantage of being independent of the model's structure but with the drawback of not being as precise as some model-specific methods.

In contrast to the aforementioned methods for generating saliency maps, Bach et al. [12] proposed Layer-wise Relevance Propagation (LRP), which uses the intermediate activations of the neurons during the forward pass to estimate the contribution of each input pixels to prediction.

Common to all interpretable ML approaches is that they focus on one-shot decisions. Thus, they do not fully address the problem of explaining behavior in sequential decision making settings, where the agent takes actions, earns rewards and affects the state of the world.

Local explanations of agent behavior. Several approaches have been introduced for explaining specific decisions in the context of Markov Decision Processes (MDP). Some works attempt to provide justifications for a policy [39, 38, 21] by making statements about particular actions choices (e.g. an action was chosen because it will lead to a state that has higher value with higher probability). Others provide causal explanations by integrating a causal structure of the domain [63, 74]. Krarup et al. [41] propose methods for generating contrastive explanations to explain action choices.

In this paper, we focus on the use of saliency maps for local explanations. Several works have implemented saliency maps in the context of Deep Reinforcement Learning (DRL). Because many DRL algorithms utilize CNNs it is possible to directly use the methods we covered in the previous paragraph on those algorithms. Zahavy et al. [80] and Wang et al. [76] for example used gradient-based saliency maps on traditional and Dueling Deep Q-Network (DQN) algorithms.

4

Greydanus et al. [28] and Iyer et al. [37] propose novel occlusion-based algorithms, where Greydanus et al. use Gaussian blur instead of complete occlusion and Iyer et al. utilize template matching to identify objects in the input. This allows them to train a new agent on this additional information and then selectively occlude those objects.

Lapuschkin et al. [44] used LRP to visualize the classical DQN architecture. In this paper, we use a more selective LRP variant which we tested on RL agents in our previous work [36] (see section 3).

Global explanations of agent behavior. Several global explanation methods describing what actions an agent takes in different states have been proposed. Hayes et al. [31] developed a system that allows users to "debug" an agent's strategy by querying its decisions in situations specified by the user. In contrast to this approach, strategy summarization methods select a set of important states to share with the user, such that the user does not need to query the agent with respect to specific states. We note that the two approaches are complementary. Booth et al. [15] compile logical formulas that specify when certain behaviors occur, e.g., by stating for which regime in the state space an agent will perform a particular action. However, this approach requires a state representation that is understandable to the user, which may not be the case in many complex domains, especially when DRL is used.

Our work takes the approach of summarizing agent policies (which we refer to as "strategy summaries") by demonstrating the behavior of an agent in a subset of world states which are considered important by the agent [7, 6]. Several methods have been proposed for selecting the subset of demonstrations to present in a summary. Some methods choose states that best enable the reconstruction of the original policy, using computational models such as inverse reinforcement learning (inferring the agent's reward function) or imitation learning (constructing a mapping from states to agents' actions) [35, 43]. An alternative approach uses heuristics for identifying "interesting" situations. The HIGHLIGHTS-DIV algorithm we utilize falls into this category, as it selects states based on the distribution of Q-values of different actions. We chose to use this approach since it does not make any assumptions about people's reasoning, is simpler computationally and was shown to improve users' understanding of agent behavior. Similar approaches have been developed in parallel [34, 65], varying in the specific formulation of the interestingness criteria used to determine which states to include in the summary.

Another recent line of work explored the problem of generating plans that are more understandable to people [42, 18, 17]. The idea underlying this approach is that by having a model of human plans in a domain, it is possible to generate plans that achieve the desired goal while being as consistent as possible with people's mental models. However, in contrast to the strategy summarization approach, these approaches have only considered goal-based plans for short-term tasks. Furthermore, they require a model of how people plan in the domain, which might not always be feasible to obtain.

Evaluation of explanation methods for RL agents. Some recent user studies examined the use of saliency maps and strategy summaries to explain the behavior of RL agents to people.

Alqaraawi et al. [4] and Selvaraju et al. [64] found that participants, who saw saliency maps, were able to predict the decision of an image classification model better then participants who did not see them. However, the participants were still only correct in about 60% of the cases and Alqaraawi et al. proposed to look beyond instance-level explanations in the future. For actual RL agents, Iyer et al. [37] and Anderson et al. [9] also used an action prediction task to evaluate saliency maps but found no clear advantage of saliency maps. In addition to the prediciton task, Anderson et al. used a retrospection task to get an even better understanding of participants' mental models and, in addition to saliency maps, investigated reward decomposition [23] and a combination of both methods. Here, they found significant positive effects for reward decomposition and the combined approach and a marginally significant (p = 0.086) effect in favor of saliency maps.

Strategy summaries have been evaluated using several different tasks. Huang et al. [35] and Lage et al. [43] asked participants to predict what actions an agent would take based on summaries optimized for policy reconstructions. Their results show that summary methods that better match with people's computational models lead to improved action prediction, but that people may use different models in different contexts. Summaries generated by a variety of interestingness criteria were shown to improve people's ability to identify regions of the state space in which an agent spends more time and regions of the state space in which an agent requires additional training [65]. Importance-based summaries (e.g. HIGHLIGHTS-DIV) were shown to improve people's ability to identify the better performing agent in an agent comparison task [5] and their ability to decide whether to trust an agent in specific world states [34].

In sum, this work extends the existing state-of-the-art in explanations of RL agents, by proposing an integrated global and local explanation method, which enhances HIGHLIGHTS-DIV summaries (global) with LRP saliency maps (local), and conducting a user study to examine the joint and separate contributions of the local and global information to people's understanding of the behavior of RL agents.

3. Saliency Maps

In this section, we describe the local explanation method which we use in our combined local and global explanation approach. While the development of the local explanation method is not the focus of this paper, we include the details of the approach for completeness. We revisit the foundations of Layer-wise Relevance Propagation (LRP) and show how to use it on the original DQN. Then we describe our previously published *argmax*-rule, an adjustment to this algorithm, which generates more selective saliency maps and which we use in this work. In addition to some previously published illustrations of the selectivity of

the *argmax*-rule, we implemented new sanity checks for our saliency maps and report their results.

3.1. Foundations

LRP does not describe a specific algorithm but a concept which can be applied to any classifier f that fulfills the following two requirements. First, f has to be decomposable into several layers of computation where each layer can be modeled as a vector of real-valued functions. Secondly, the first layer has to be the input x of the classifier containing, for example, the input pixels of an image, and the last layer has to be the real-valued prediction of the classifier $f(x)$. Any DRL agent fulfills those requirements if we only consider the output value that corresponds to the action we want to analyze.

For a given input x, the goal of any method following the LRP concept is to assign relevance values R_j^l to each computational unit j of each layer of computation l, in such a way that R_j^l measures the local contribution of the unit j to the prediction $f(x)$. A method of calculating those relevance values R_j^l is said to follow the LRP concept if it sets the relevance value of the output unit to be the prediction $f(x)$ and calculates all other relevance values by defining

$$R_j^l := \sum_{k \in \{j \text{ is input for neuron } k\}} R_{j \leftarrow k}^{l,l+1}, \tag{1}$$

for **messages** $R_{j \leftarrow k}^{l,l+1}$, such that

$$R_k^{l+1} = \sum_{j \in \{j \text{ is input for neuron } k\}} R_{j \leftarrow k}^{l,l+1}. \tag{2}$$

In this way a LRP variant is determined by choosing messages $R_{j \leftarrow k}^{l,l+1}$. Through definition 1 it is then possible to calculate all relevance values R_j^l in a backward pass, starting from the prediction $f(x)$ and going towards the input layer. Furthermore, equation 2 gives rise to

$$\sum_k R_k^{l+1} = \sum_k \sum_{j \in \{j \text{ is input for neuron } k\}} R_{j \leftarrow k}^{l,l+1}$$
$$= \sum_j \sum_{k \in \{j \text{ is input for neuron } k\}} R_{j \leftarrow k}^{l,l+1} = \sum_j R_j^l.$$

This ensures that the relevance values of each layer l are a linear decomposition of the prediction

$$f(x) = \cdots = \sum_{j=1}^{dim(l)} R_j^l = \cdots = \sum_{j=1}^{dim(input)} R_j^{input}.$$

Such a linear decomposition is easier to interpret than the original classifier because we can think of positive values R_j^l to contribute evidence in favor of the

decision of the classifier and of negative relevance values to contribute evidence against the decision.

To use LRP on a DQN agent we first have to look at its network architecture. The DQN f, as introduced by Mnih et al. [51], consists of three convolutional layers $conv_1, ..., conv_3$ followed by two fully connected layers fc_1 and fc_2. For an input x we write $fc_i(x)$ and $conv_i(x)$ for the output of the layers fc_i and $conv_i$, respectively, during the forward pass that calculates $f(x)$. In this notation, the Q-Values (i. e. the output of the whole DQN) are $fc_2(x)$.

Following the LRP notation, we denote the relevance value of the j-th neuron in the layer l with R_j^l. As described above, we have to define messages $R_{j \leftarrow k}^{l,l+1}$ for any two consecutive Layers $l, l+1$ to determine a LRP variant. For now we assume that $l+1$ is one of the fully connected layers fc_i. The convolutional case works analogously and will be covered in more detail in the next section. $R_{j \leftarrow k}^{l,l+1}$ should measure the contribution of the j-th neuron of fc_{i-1} to the k-th neuron of fc_i, therefore we have to look at the calculation of $fc_i(x)_k$. The fully connected layer fc_i uses a weight matrix W_i, a bias vector b_i and an activation function σ_i as parameters for its output. Let W_i^k be the k-th row of W_i and b_i^k the k-th entry of b_i. Then the activation of the k-th neuron in $fc_i(x)$ is

$$\sigma_i(W_i^k \cdot fc_{i-1}(x) + b_i^k),$$

where $\cdot$ denotes the dot product and fc_0 is the flattened output of $conv_3$.

Usually the ReLU function $\sigma(x) = max(0, x)$ is used as activation function σ_i in the DQN architecture. Bach et al. [12] argue that any monotonous increasing function σ with $\sigma(0) = 0$, like the ReLU function, conserves the relevance of the dot product $W_i^k \cdot fc_{i-1}(x)$. Newer LRP variants, like the one used by Montavon et al. [54], also omit the bias when defining $R_{j \leftarrow k}^{l,l+1}$. With those two assumptions the relevance of each neuron of fc_{i-1} to $fc_i(x)_k$ is the same as their contribution to the dot product $W_i^k \cdot fc_{i-1}(x) = \sum_j w_{jk} fc_{i-1}(x)_j$. This is a linear decomposition, so we can use $w_{jk} fc_{i-1}(x)_j$ to measure the contribution of the j-th neuron of fc_{i-1}.

Since we want to find the parts of the input that contributed evidence in favor of the decision of the DQN agent, we restrict ourself to the positive parts of that sum. That is, we set

$$z_{jk}^+ := \begin{cases} w_{jk} fc_{i-1}(x)_j & \text{if } w_{jk} fc_{i-1}(x)_j > 0 \\ 0 & \text{if } w_{jk} fc_{i-1}(x)_j \leq 0 \end{cases}.$$

With this, we define the messages as $R_{j \leftarrow k}^{l,l+1} := \frac{z_{jk}^+}{\sum_j z_{jk}^+} R_k^{l+1}$. This method is called z^+-rule (without bias) and satisfies the LRP equation 2.

3.2. An argmax approach to LRP

In this subsection, we introduce our adjustment to the LRP variant called z^+-rule which we revisited in the previous subsection. Recent work [37, 24] indicates that DRL agents focus on certain objects within the visual input.
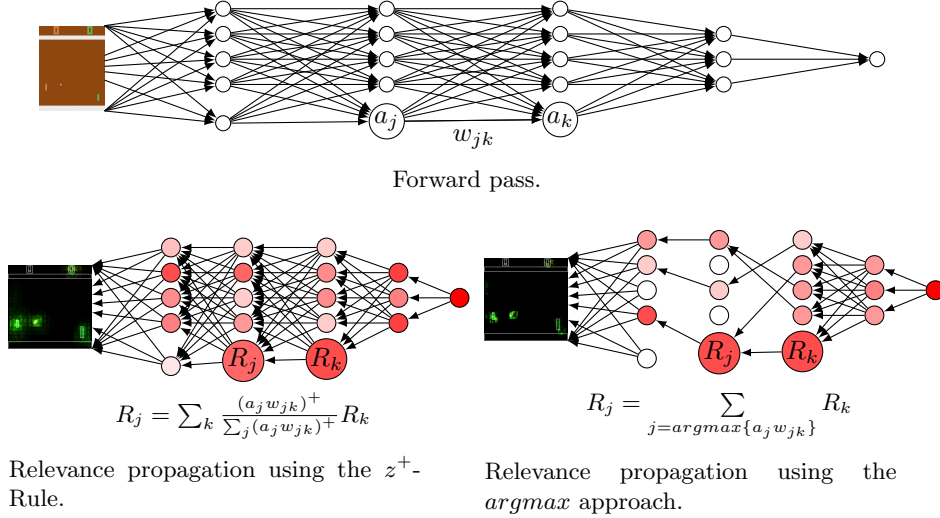
8

Forward pass.

$$R_j = \sum_k \frac{(a_j w_{jk})^+}{\sum_j (a_j w_{jk})^+} R_k$$

Relevance propagation using the z^+-Rule.

$$R_j = \sum_{j=argmax\{a_j w_{jk}\}} R_k$$

Relevance propagation using the $argmax$ approach.

Figure 1: A visualization of how our $argmax$ approach differs from the z^+ Rule.

With our approach, we aim to generate saliency maps that reflect this property by focusing on the most relevant parts of the input instead of giving too many details. For this purpose, we propose to use an $argmax$ function to find the most contributing neurons in each convolutional layer.

This idea is inspired by Mopuri et al. [55], who generated visualizations for neural networks solely based on the positions of neurons that provide evidence in favor of the prediction. During this process, they follow only the most contributing neurons in each convolutional layer. Our method adds relevance values to the positions of those neurons and therefore expands the approach of Mopuri et al. by an additional dimension of information. Since those relevance values follow the LRP concept, they also possess the advantageous properties of the LRP concept like conservation of the prediction value.

As we have seen in the foundations section 3.1, a LRP method is defined by its messages $R_{j \leftarrow k}^{l,l+1}$ which propagate the relevance from a layer $l+1$ to the preceding layer l. If $l+1$ is a fully connected layer fc_i of the DQN (see section 3.1 for our notation of the DQN architecture), we use the same messages that are used in the z^+-rule. In the case that l and $l+1$ are convolutional layers $conv_{i-1}$ and $conv_i$, we propose new messages based on the $argmax$ function. To define those messages we analyze how the activation of a neuron $conv_i(x)_k$ was calculated during the forward pass. Let W and A denote the weight kernel and part of $conv_{i-1}(x)$ respectively that were used to calculate $conv_i(x)_k$ during the forward pass. If we write W and A in appropriate vector form, we get

$$conv_i(x)_k = \sigma(\sum_j w_j a_j + b),$$

where σ denotes the activation function of $conv_i$ and b the bias corresponding to

W. Analogously to the z^+-rule we assume that the activation function and the bias can be neglected when determining the relevance values of the inputs a_i. We propose to use an $argmax$ function to find the most relevant input neurons by defining the messages in the following way

$$R_{j \leftarrow k}^{l,l+1} := \begin{cases} R_k^{l+1} & \text{if } j = argmax\{w_j a_j\} \\ 0 & \text{if not.} \end{cases}$$

This definition satisfies the LRP condition given by equation 2 because the only non vanishing summand of the sum

$$\sum_{j \in \{j \text{ is input for neuron } k\}} R_{j \leftarrow k}^{l,l+1}$$

is R_k^{l+1}.

If we use the same $argmax$ approach to propagate relevance values from $conv_1$ to the input $conv_0$, then we get very sparse saliency maps where only a few neurons are highlighted. If we highlight the entire areas of the input $conv_0$ that were used to calculate relevant neurons of $conv_1$, then we lose information about the relevance values inside those areas. Therefore, we draw inspiration from the guided Grad-CAM approach introduced in [64]. Guided Grad-CAM uses one thorough relevance analysis for the neurons of the last convolutional layer to get relevant areas for the specific prediction and another thorough relevance calculation for the input pixels to get fine granular relevance values inside those areas. We already did a thorough analysis of the neurons of the last convolutional layer by using the z^+-rule on the fully connected layers. By following the most relevant neurons through the convolutional layers we keep track of the input areas that contributed the most to those values. Mimicking the second thorough analysis of the Guided Grad-CAM approach we propose to use the z^+-rule to propagate relevance values from $conv_1$ to $conv_0$. This generates fine granular relevance values inside the areas identified by following the most contributing neurons and ascertains that those relevance values follow the LRP concept.

Figure 1 visualizes the differences between our $argmax$ approach and the z^+-rule. An implementation of our algorithm that builds up on the iNNvestigate framework [3] can be found here: `https://github.com/HuTobias/LRP_argmax`.

3.3. Illustration of the Selectivity of the argmax-rule

In order to verify that our $argmax$ approach, described in section 3, creates more selective saliency maps than the z^+-rule (see section 3.1), we tested our approach on three different Atari 2600 games. For all games, we trained an agent using the DQN implementation of the OpenAI baselines framework [20]. The results of all experiments are shown in our previous work [36]. We review the Pacman results here, since we use this game in the user study evaluating our combined explanation approach.
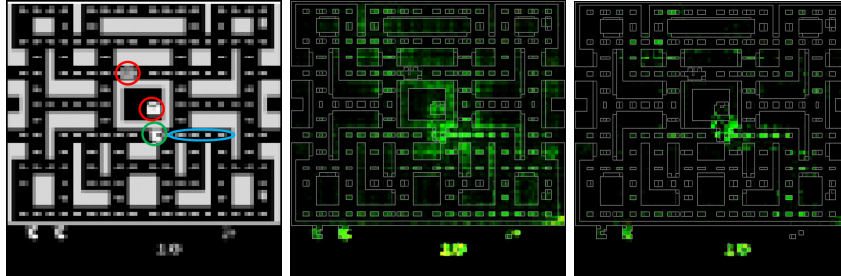
10

Figure 2: The left image shows a screen of Pacman. The player (green circle) has to collect pellets (blue area) while avoiding ghosts (red circles). The saliency map created for this game-state by the z^+-rule (middle) highlights a huge area as relevant while our *argmax* approach (right) focuses on the vicinity of the player.

In the game Pacman the player has to navigate through a maze and collect pellets while avoiding enemy ghosts. Because this game contains many important objects and gives the agent a huge variety of possible strategies, DQN agents struggle in this environment and perform worse than the average human player (see [51]). Explainable AI methods are especially desirable in environments like this, where the agent is struggling, because they help us to understand where the agent had difficulties. The saliency maps created with the z^+-rule (figure 2) reflect the complexity of Pacman by showing that the agent tries to look at nearly all of the objects in the game. This information might be helpful to optimize the DRL agent, but it also distracts from the areas which influenced the agents' decision the most. Figure 2 shows that the saliency map created by the *argmax* approach is more focused on the vicinity of the agent and makes it clearer what the agent is focusing on the most. Figure 2 also illustrates that a fine-granular saliency map in the vicinity of the agent is necessary to see that the agent will most likely decide on moving to the right as his next action.

3.4. Sanity Checks

It is not yet possible to verify whether a saliency map algorithm perfectly reflects what a model learned. However, a basic prerequisite for this is that the saliency maps depend on the weights learned by the model. To verify this, Adebayo et al. [1] proposed sanity checks that cascadingly randomize each layer of the network, starting with the output layer. If the saliency maps depend on the learned weights, then this will lead to increasingly different visualisations. Sixt et al. [67] applied the sanity checks to several LRP variants but they have never been used on our *argmax*-rule. Therefore, we implemented the sanity checks[1] for our *argmax*-rule and test it on the regular Pacman agents described in section 6. An example of these tests for a single state is shown in Fig. 3.

---

[1]The code we used for the sanity checks can be found here: `https://github.com/HuTobias/HIGHLIGHTS-LRP/tree/master/sanity_checks`

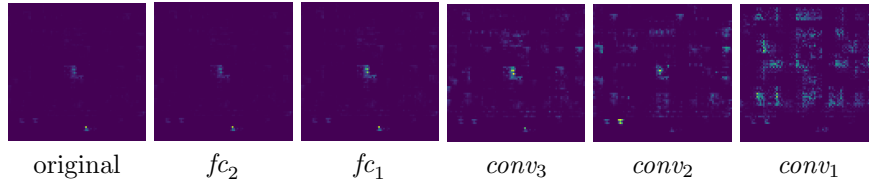| original | fc_2 | fc_1 | $conv_3$ | $conv_2$ | $conv_1$ |

Figure 3: Example for how the LRP-argmax saliency maps change when the network's layers are randomized cascadingly, beginning with output layer fc_2.

To measure how similar two saliency maps are we use three different metrics proposed by Adebayo et al. [1]: Spearman rank correlation, structural similarity (ssim) and Pearson correlation of the histogram of gradients. To account for a possible change of sign in the saliency maps, we adopt an approach by Sixt et al [67] and use the maximum similarity of the original and the inverted saliency map. That means that for two saliency maps $S, S' \in \mathbb{R}^{m \times n \times c}$ and a similarity measurement $sim : \mathbb{R}^{m \times n \times c} \times \mathbb{R}^{m \times n \times c} \to \mathbb{R}$ we calculate the actual similarity with

$$\max(sim(S, S'), sim(\mathbf{1} - S, S')) \tag{3}$$

where $\mathbf{1} \in \mathbb{R}^{m \times n \times c}$ is filled with 1s. Fig. 4 shows the average similarities per randomized layer for a gameplay stream of 1000 states.
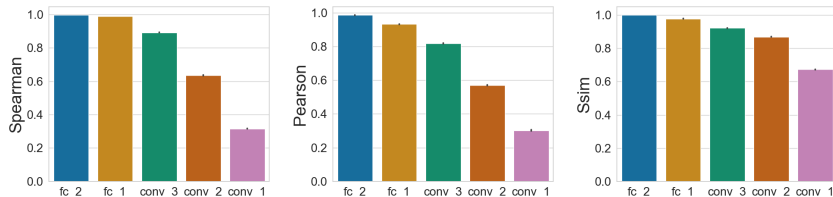

Figure 4: The average similarities between saliency maps for the fully trained agent and agents where the layers have been randomized cascadingly, starting with the last layer fc_2. The values are based on a stream of 1000 actions in the Atari 2600 Pacman game.

The relatively high values for the structural similarity (ssim) can be explained by the high amount of intersecting zeros in all saliency maps. Apart from that, we see the same trends already observed by Sixt et al. [67] and Adebayo et al.[1]: the sanity maps do analyze the learned weights but the fully connected layers are not sufficiently analyzed. As a consequence, the saliency maps are not class discriminatory. However, class discriminatory saliency maps often come with other drawbacks like being noise [67] or not analyzing all layers [64].

4. Strategy Summarization

This section describes the strategy summarization approach to global explanations, and the HIGHLIGHTS algorithm and its extension HIGHLIGHTS-DIV, which we developed and evaluated in prior work [5].

Our formalization of the summarization problem assumes that the agent uses a Markov Decision Process (MDP), where A is the set of actions available to the agent, S is the set of states, R: $S \times A \to \mathbb{R}$ is the reward function, which maps each state and action to a reward, and Tr is the transition probability function, i.e., $Tr(s', a, s)$ defines the probability of reaching state s', when taking action a in state s. The agent has a policy π which specifies which action to take in each of the states.

We formalize the problem of summarizing an agent's behavior as follows: from execution traces of an agent, choose a set $T = \langle t_1, ..., t_k \rangle$ of trajectories to include in the summary, where each trajectory is composed of a sequence of l consecutive states and the actions taken in those states $\langle (s_i, a_i), ..., (s_{i+l-1}, a_{i+l-1}) \rangle$. We consider trajectories rather than single states because seeing what action was taken by the agent in a specific state might not be meaningful without a broader context (e.g., watching a self-driving car for one second will not reveal much useful information). Because it is infeasible that people will be able to review the behavior of an agent in all possible states, we assume a limited budget k for the size of the summary, such that $|T| = k$. This budget limits the amount of time and cognitive effort that a person needs to invest in reviewing the agent's behavior.

There are several factors that could be considered when deciding which states to include in a summary, such as the effect of taking a different action in that state, the diversity of the states that are included in the summary and the frequency at which states are likely to be encountered by the agent. The approach we describe here focuses on the first factor, which we refer to as the "importance" of a state. Intuitively, a good summary should provide a person reviewing the summary with a sense of the agent's behavior in states that the person considers important (e.g., when making a mistake would be very costly). The importance of states included in the summary could substantially affect the ability of a person to assess an agent's capabilities. For example, imagine a summary of a self-driving car that only shows the car driving on a highway with no interruptions. This summary would provide people with very little understanding of how the car might act in other, more important, scenarios (e.g., when another car drives into its lane, when there is road construction). In contrast, a summary showing the self-driving car in a range on more interesting situations (e.g., overtaking another car, breaking when a person enters the road) would convey more useful information to people reviewing it.

4.1. The "Highlights" Algorithm

The HIGHLIGHTS algorithm generates a summary of an agent's behavior from simulations of the agent in an online manner. It uses the notion of state *importance* [73] to decide which states to include in the summary. Intuitively, a state is considered important if taking a wrong action in that state can lead to a significant decrease in future rewards, as determined by the agent's Q-values. Formally, the importance of a state, denoted $I(s)$, is defined as:

$$I(s) = \max_a Q^{\pi}_{(s,a)} - \min_a Q^{\pi}_{(s,a)} \tag{4}$$

13

This measure has been shown to be useful for choosing teaching opportunities in the context of student-teacher reinforcement learning [73, 8].

Before providing a detailed pseudo-code of the algorithm, we describe its operation at a high-level. HIGHLIGHTS generates a summary that includes trajectories that capture the most important states that an agent encountered in a given number of simulations. To do so, at each step it evaluates the importance of the state and adds it to the summary if its importance value is greater than the minimal value currently represented in the summary (replacing the minimal importance state). To provide more context to the user, for each such state HIGHLIGHTS also extracts a trajectory of states neighboring it and the actions taken in those states.

A pseudo-code of the HIGHLIGHTS algorithm is given in Algorithm 1. Table 1 summarizes the parameters of the algorithm. HIGHLIGHTS takes as input the policy of the agent π which is used to determine the agent's actions in the simulation and state importance values, the budget for the number of trajectories to include in the summary (k) and the length of each trajectory surrounding a state (l). Each such trajectory includes both states preceding the important state and states that were encountered immediately after it. The number of subsequent states to include is determined by the $statesAfter$ parameter (the number of preceding states can be derived from this parameter and l). We also specify the number of simulations that can be run ($numSimulations$), and the minimal "break" interval between trajectories ($intervalSize$) which is used to prevent overlaps between trajectories. HIGHLIGHTS outputs a summary of the agent's behavior, which is a set of trajectories (T).

Parameter	Description (value used in experiments)
k	Summary budget, i.e., number of trajectories (5)
l	Length of each trajectory (40)
$numSimulations$	The number of simulations run by HIGHLIGHTS (50)
$intervalSize$	Minimal number of states between two trajectories in the summary (50)
$statesAfter$	Number of states following s to include in the trajectory (10)

Table 1: Parameters of the HIGHLIGHTS algorithm and the values assigned to them in the experiments reported in [5] (in parentheses).

The algorithm maintains two data structures: T is a priority queue (line 2), which will eventually hold the trajectories chosen for the summary; t is a list of state-action pairs (line 3), which holds the current trajectory the agent encounters. The procedure runs simulations of the agent acting in the domain. At each step of the simulation, the agent takes an action based on its policy and advances to a new state (line 8). That state-action pair is added to the current trajectory (line 11). If the current trajectory reached its maximal length, the oldest state in the trajectory is removed (lines 9-10). HIGHLIGHTS computes the importance of s based on the Q-values of the agent itself, as defined in

Equation 4 (line 14).

If a sufficient number of states were encountered since the last trajectory was added to the summary, state s will be considered for the summary (the $c == 0$ condition in line 17). State s will be added to the summary if one of two conditions hold: either the size of the current summary is smaller than the summary size budget, or the importance of s is greater than the minimal importance value of a state currently represented in the summary (line 17). If one of these conditions holds, a trajectory corresponding to s will be added to the summary. The representation of a trajectory in the summary (a $summaryTrajectory$ object) consists of the set of state-action pairs in the trajectory (which will be presented in the summary), and the importance value I_s based on which the trajectory was added (such that it could be compared with the importance of states encountered later). This object (st) is initialized with the importance value (line 20) and is added to the summary (line 21), replacing the trajectory with minimal importance if the summary reached the budget limit (lines 18-19). Because the trajectory will also include states that follow s, the final set of state-action pairs in the trajectory is updated later (lines 15-16). Last, we set the state counter c to the interval size, such that the immediate states following s will not be considered for the summary. At the end of each simulation, the number of runs is incremented (line 24). The algorithm terminates when it reaches the specified number of simulations.

Originally, HIGHLIGHTS was implemented in an online algorithm because it is less costly, both in terms of runtime and in terms of memory usage. In addition, such an algorithm can be incorporated into the agent's own learning process without additional cost. In this paper, we adapt the algorithm to work offline, as described in Section 5.

4.2. Considering State Diversity: the HIGHLIGHTS-DIV algorithm

Because HIGHLIGHTS considers the importance of states in isolation when deciding whether to add them to the summary, the produced summary might include trajectories that are similar to each other. This could happen in domains in which the most important scenarios tend to be similar to each other. To mitigate this problem, we developed a simple extension to the HIGHLIGHTS algorithm, which we call HIGHLIGHTS-DIV. Similarly to HIGHLIGHTS, this algorithm also determines which states to include in the summary based on their importance. However, it also attempts to avoid including a very similar set of states in the summary, thus potentially utilizing the summary budget more effectively.

HIGHLIGHTS-DIV takes into consideration the diversity of states in the following way: when evaluating a state s, it first identifies the state most similar to s that is currently included in the summary[2], denoted s'. Then, instead of

---

[2]We assume that distance metric to compare states can be defined. This can be done in many domains, e.g., by computing Euclidean distance if states are represented by feature vectors.

Algorithm 1: The HIGHLIGHTS algorithm.

Input: $\pi, k, l, numSimulations, intervalSize, statesAfter$

Output: T

1   $runs = 0$

2   $T \leftarrow PriorityQueue(k, importanceComparator)$

3   $t \leftarrow$ empty list

4   $c = 0$

5   **while** $(runs < numSimulations)$ **do**

6      $sim = InitializeSimulation()$

7   **while** $(!sim.ended())$ **do**

8         $(s, a) \leftarrow sim.advanceState(\pi)$

9   **if** $(|t| == l)$ **then**

10           $t.remove()$

11        $t.add((s, a))$

12   **if** $(c > 0)$ **then**

13           $c = c - 1$

14        $I_s \leftarrow computeImportance(\pi, s)$

15   **if** $(IntervalSize - c == statesAfter)$ **then**

16   lastSummaryTrajectory.setTrajectory(t)

17   **if** $((|T| < k)$ *or* $(I_s > minImportance(T)))$ *and* $(c == 0))$ **then**

18   **if** $|T| == k$ **then**

19   T.pop()

20           $st \leftarrow$ new $summaryTrajectory(I_s)$

21           $T.add(st)$

22           $lastSummaryTrajectory \leftarrow st$

23           $c = intervalSize$

24   runs = runs+1

comparing the importance of a state to the minimal importance value that is currently included in the summary, HIGHLIGHTS-DIV compares I_s to $I_{s'}$. If I_s is greater than $I_{s'}$, the trajectory which includes s' in the summary will be replaced with the current trajectory (which includes s). This approach allows less important states to remain represented in the summary (because they will not be compared to some of the more important states that differ from them), potentially increasing the diversity of trajectories in the summary and thus conveying more information to users.

4.3. Empirical evaluation of HIGHLIGHTS and HIGHLIGHTS-DIV

We summarize the main results of the study conducted in our previous work, which demonstrated the usefulness of HIGHLIGHTS and HIGHLIGHTS-DIV summaries. For complete details of the study design and its results see Amir & Amir [5]. The performance of the basic HIGHLIGHTS algorithm was compared with that of two baselines: (1) random summaries generated by sampling k trajectories uniformly from the agent's execution trace, and (2) summaries generated from the first k trajectories the agent encounters. The task used in the study was identifying the agent that performs better in pairwise comparisons, based on the summaries. Three Ms. Pacman agents were trained varying in their quality: a high-quality agent, medium-quality agent and low-quality agent. This was achieved by varying the number of training episodes.

In the first experiment, 40 participants recruited from Amazon Mechanical Turk (23 female, mean age = 35.35, STD = 10.4), were asked to make the pairwise agent comparisons based on summaries generated by either the basic HIGHLIGHTS algorithm or one of the two baselines (Random or First). The study used a within-subject design, such that each participant completed nine comparison tasks showing all combinations of pairs of agents and the summary method (e.g., comparing the high-quality agent to the low quality agent based on the HIGHLIGHTS summary). In the second experiment 48 additional participants (25 female, mean age=36, STD=11.6), performed the same task, but this time summaries were generated either by HIGHLIGHTS-DIV, basic HIGH-LIGHTS or the random baseline (since the "first" baseline led to the worst performance in the first experiment). In both experiments, participants were incentivized to answer correctly as they received a bonus payment depending on their performance.

Results aggregated from both experiments are shown in Figure 5. Both HIGHLIGHTS and HIGHLIGHTS-DIV summaries led to significantly improved performance of participants compared to the baselines. HIGHLIGHTS-DIV further led to improved performance compared to HIGHLIGHTS, especially when comparing the medium quality agent with the high quality agent, which was the hardest comparison to make as their actual performance did not differ by much. Participants also expressed a subjective preference to HIGHLIGHTS summaries compared to baselines.
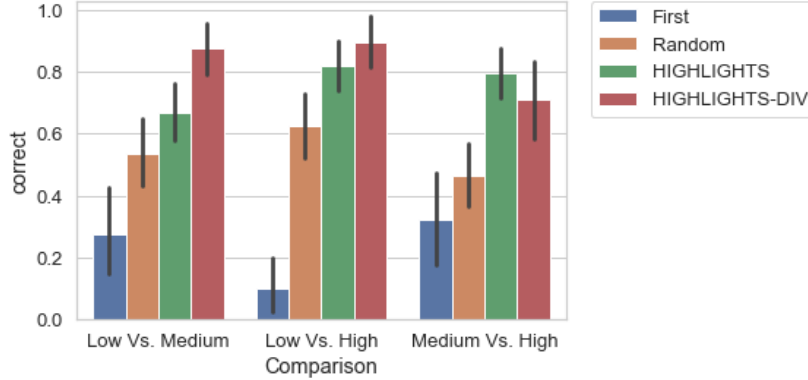
17

Figure 5: Correctness rates of participants (aggregated from both experiments) in choosing the better performing agent. The x-axis shows the three different pairwise agent comparison tasks (low quality agent vs. medium quality agent, etc.). In all cases, HIGHLIGHTS and HIGHLIGHTS-DIV outperformed the two baselines. HIGHLIGHTS-DIV led to significant improvement over HIGHLIGHTS only in the Low Vs. Medium agent comparison, which was the most difficult comparison as the two agents were most similar to each other in performance.

5. Integrating Local and Global Information

In this section, we describe our integration of the local LRP-argmax saliency maps described in section 3 into the global HIGHLIGHTS-DIV summaries described in section 4.2. To this end, we describe how the agents were trained, what adjustments we made to HIGHLIGHTS-DIV for the deep reinforcement learning algorithm we used, how we generated saliency maps and how the combined information is displayed.

Agent training. To evaluate our combined explanation approach, we trained several Pacman agents using the OpenAI baselines [20] implementation of the DQN-algorithm [52]. The network architecture used in this implementation is described in 3.1. The environment we use is the Atari 2600 game MsPacman included in the Arcade Learning Environment (ALE) [14], which we refer to in this work as Pacman for simplicity.

For each step in a game, the input state consists of the four last frames (the raw pixel values of a single screen of the game) f_1 to f_4. Each frame f_i is converted to grey scale and scaled down to 84×84 pixels. The frames are then stacked to enable the agent to see temporal differences (i.e. movement). The agent chooses an action only every four frames and this action is repeated for the next four frames. In Pacman, the agent has nine different actions to choose from, which correspond to the meaningful actions that can be achieved with an Atari 2600 controller (do nothing, up, down, left, right, up-left, up-right, down-left, down-right).

The reward is based on the ALE [14] reward function that uses the increase of the in-game score at the beginning of the four frames of a state compared to

18

the score after those frames. The final reward functions we used are detailed in section 6, when we describe the agents used in the empirical evaluation.

Generating gameplay streams and saliency maps. Since deep neural networks increase the time that the agent needs for each prediction and the LRP-analysis of this decision requires additional computation time, we recorded a stream of $10,000$ steps for each agent and used them to create our summaries. These streams also increase the reproducibility of our experiments[3].

We computed the average in-game score of each trained agent over the entire stream. This allows us to objectively say which agent achieved the most points during the simulations used for our summaries and therefore gives us a ground-truth for the agent comparison task (see section 6).

Since the Atari 2600 version of Pacman does not respond to input for the first 250 frames (empirically tested) after the game starts, we exclude those frames from the streams. Furthermore, we force the agent to repeat the 'do nothing' action for a random amount of steps between 0 and 30, until it is allowed to choose actions based on its policy. This method introduces randomness into the deterministic Pacman game and is also used during training by the DQN algorithm [52, 20]. Saliency maps are created using the LRP-argmax algorithm described in Section 3.

Adjustments to HIGHLIGHTS-DIV. For the summaries, we make several adjustments to the HIGHLIGHTS-DIV algorithm described in Section 4.2, to adapt it to the DQN settings. First, we change the way importance is calculated. Instead of using equation 4 which calculates the importance by comparing the highest with the lowest Q-value, we use the difference between the highest and second highest Q-values. Let secondhighest be the operation that finds the second highest value in a set, then this can be written as:

$$I(s) = \max_a Q^{\pi}_{(s,a)} - \underset{a}{\text{secondhighest}}\, Q^{\pi}_{(s,a)} \tag{5}$$

While examining the gap between the best and worst actions worked well in a simpler Pacman environment in which there were only four possible actions, it did not generalize well to the Atari environment where there is a larger number of actions. One possible explanation for this is that some of the 9 actions of the Pacman environment overlap. For example, "left" and "top left" can be used interchangeably in many states. Therefore the agent might ignore some of the actions completely. To verify this, we examined the frequency of choosing each action, and found that two of the three agents we trained were clearly biased against certain actions.[4] Therefore, some Q-values are largely uninformed by exploration and might have arbitrarily low values, making the worst Q-value

---

[3]Since the streams are fairly big we did not upload them. They are available upon request from the authors.

[4]The results can be seen in `https://github.com/HuTobias/HIGHLIGHTS-LRP/tree/master/action_checks`

non-informative. For the diversity computation in HIGHLIGHTS-DIV, we use Euclidean distance over the raw $84 \times 84 \times 4$ input states.

Since we pre-generated a stream of $10,000$ states, we implement an offline version of HIGHLIGHTS-DIV that selects the states for the summary retrospectively from the generated stream. The procedure begins by sorting the states based on their Q-values, and adding them to the summary according to this ordering. To reduce the number of overlapping trajectories, we compare each new state with all states in the current summary and corresponding context states (this is equivalent to the HIGHLIGHTS-DIV variant). To find a suitable threshold that determines when a state is too similar to the states that were already selected for the summary, we randomly pick a subset of $1,000$ random states from the recorded stream and calculate the similarity between each pair of states in this set. Then, we set the threshold to be a percentile of the distribution of those similarity values. We empirically found (by manually examining a sample of states) that using a threshold of 3% led to no obvious duplicate trajectories for any of the agents.

Video generation. The videos we generate from the states chosen by the summary show 30 frames per second. To emphasize that demonstrations show different trajectories, they are separated by a black screen that appears for 1 second (inspired by the fade-out effect used in [65]). To prevent the users from using the in-game score to gauge how good an agent is, we mask the bottom half of the screen with black pixels. In pilot studies, participants complained that the videos were flickering too much. One of the reasons for this is that the Atari 2600 implementation of Pacman does not show every object in every frame to save computing power. Since we showed all frame after each other these objects appeared to be blinking and distracted the viewers. To combat this problem, we do not display the current frame f_i. Instead we display $\max(f_i, f_{i-1})$, the maximum of each pixel over the current frame f_i and the preceding frame f_{i-1}. While this introduces some artifacts (e.g. red pellets showing through blue ghosts) it considerably reduces the flickering.

Another measure we take against this flickering is to interpolate between the different saliency maps instead of showing a completely different saliency map for each frame. Let f_1 to f_4 be the four frames of an input state and let s_1 to s_4 be the saliency maps for each of these frames that analyze the agent's decision in this state. For $i < 4$ the action that Pacman will take after frame f_i is not related to the saliency map s_i, since the agent only decides on a new action every four frames and is still repeating the action that he decided on based on the last state (composed of the 4 frames before f_1). Therefore we show the saliency map s_4 over the frame f_4 and for the other frames ($i < 4$) we interpolate between the last shown saliency map and s_4.

Before this interpolation we normalize the saliency maps to have a maximum of 1 and a minimum of 0. We do this over all 4 frames of the states $s_1, ..., s_4$ to avoid losing information that might be transported in the magnitude of relevance values between the frames.

Finally, we add the interpolated saliency maps to the green channel of the

original screen frame. Our complete implementation can be found here: `https://github.com/HuTobias/HIGHLIGHTS-LRP`

6. Empirical Evaluation

To evaluate our hypothesis that there is benefit to combining global and local explanations of RL agents, we conducted a user study. In this study, participants were asked to compare different agents and to reflect on the strategies of agents based on the information they were shown. We next describe in detail the study design, the specific hypotheses we tested, and the metrics we used to evaluate the results.

6.1. Study Design

Empirical domain. We used the Atari game Pacman for our experiments (see section 5 for the specific implementation). Atari games are a common benchmark for state of the art reinforcement learning algorithms [14, 20, 51, 76] and to test explanation methods for those algorithms [5, 28, 36, 44, 77]. We chose Pacman since it is not as reaction-based as some other Atari games (e.g. Breakout or Enduro) and allows the RL agents to develop different strategies. Furthermore, no additional domain knowledge is necessary to understand Pacman and the rules are not too complicated. This enables us to conduct a study with a wide range of participants by simply explaining the rules at the beginning of the study.

In the game, Pacman obtains points by eating food pellets while navigating through a maze and escaping ghosts. There are two types of pellets: regular pills for which Pacman receives 10 points, and power pills that are worth 50 points and also turn the ghosts blue, which makes them edible by Pacman. Pacman receives 200, 400, 800, 1600 points for each ghost it eats successively. At random intervals cherries spawn and move through the labyrinth. Eating a cherry gives 100 points.

To evaluate participants' ability to differentiate between alternative agents and analyze their strategies, we trained agents that behave qualitatively different. To this end, we modified the reward function used for training (similar approach to that used by Sequeira et al. [65]), resulting in three types of agents. As mentioned in section 5, we based all of those reward functions on the default ALE [14] reward function, which measures the increase in in-game score (as described above) between the first and last frame of a state.

- *Regular agent*: This agent was trained using the default reward function of the ALE [5].

---

[5] To remove unnecessary magnitude we divided the rewards by the factor 10, such that a regular pill gives a reward of 1.

- *Power pill agent*: This agent was trained using a reward function that only assigned positive rewards to eating power pills[6].

- *Fear-ghosts agent*: This agent used the default ALE reward function but was given an additional negative reward of -100 when being eaten by ghosts, causing it to more strongly fear ghosts (which is implicitly learned by other agent due to the lack of future rewards caused by being eaten).

Each agent was trained for 5 Million steps with with the algorithm described in section 5. At the end of this training period the best performing policy is restored.

Experimental conditions. To evaluate the potential benefits of integrating global and local explanations, and their relative importance, we assigned participants to four different conditions (summarized in Table 2). The first two conditions included only global information, while the remaining two conditions integrated local explanations as well:

- **Random Summaries (R)**: In this condition, participants were shown summaries that were generated by randomly selecting state-action pairs from the streams of the Pacman agents playing the game. We note that since each state had the same probability of being chosen, in practice states that are encountered more frequently will be more likely included. Hence, this is equivalent to selecting states based on the likelihood of encountering them. To ensure that the randomly generated summary was not, by chance, particularly good or particularly bad, we generated 10 different random summaries and randomly assigned them to participants in this condition.

- **HIGHLIGHTS-DIV summaries (H)**: In this condition, participants were shown summaries generated by the HIGHLIGHTS-DIV algorithm. The specific implementation of this algorithm and the parameters we used for diversity are described in section 5.

- **Random Summaries+Saliency ($R+S$)**: These summaries included the same states as those shown in the R summaries, but each image was overlaid with a saliency map generated by the LRP-argmax algorithm described in section 3.

- **HIGHLIGHTS-DIV summaries+Saliency ($H+S$)**: These summaries included the same states as those shown in the H summaries, where each image was overlaid with a saliency map generated by the LRP-argmax algorithm described in section 3.

---

[6]We achieved this by only giving the agent a reward if the increase in score was between 50 and 99. The range is necessary since Pacman is forced to eat at least one regular pill directly before it eats a power pill.

	'Random' summaries	HIGHLIGHTS-DIV
No saliency maps	R	H
LRP saliency maps	$R+S$	$H+S$

Table 2: The four study conditions.

We used a budget of $k = 5$ for the summaries. That is, each summary included 5 base states chosen either randomly or by HIGHLIGHTS-DIV, where for each state we included a surrounding context window of 10 states that occurred right before and after the chosen state and an interval size of 10 states to prevent directly successive states in the summary.

The video creation and saliency map overlay process is described in detail in section 5. All video summaries used in the study are available online.[7].

We note that we did not include a condition that shows only local explanations, since by definition a local explanation is given for a specific state, forcing us to make some choice about which states to show (which means making a global decision). However, the $R+S$ condition simulates a scenario where local explanations are shown for randomly selected states.

Participants. We recruited participants through Amazon Mechnical Turk ($N = 134$, the majority of participants were between the ages of 25 and 44, 47 females). Participation was limited to people from the US, UK, or Canada (to ensure sufficient English level) with task approval rate greater than 97%. Since saliency maps are not designed for color blind people, the participants were also asked if they were color blind and stopped from participating if they are.

Procedure. Participants were first asked to answer demographic questions (age, gender) and questions regarding their experience with Pacman and their views on AI. Then, they were shown a tutorial explaining the rules of the game Pacman and were asked to play the game to familiarize themselves with it. To verify that participants understood the rules, they were asked to complete a quiz, and were only allowed to proceed with the survey after answering all questions correctly. After completing the quiz, they were given information and another quiz regarding the Pacman agent video summaries. In conditions $R+S$ and $H+S$, this also included an explanation and a quiz about saliency maps. Then, they proceeded to the main experimental tasks. See Appendix D for the complete questionnaire. Participants were compensated as follows: they received \$4 base payment, and an additional bonus of 10 cents for each correct answer. The study protocol was approved by the Institutional Review Board at the Technion.

Main tasks. We aimed to investigate three aspects related to the participants in the study: (1) the mental model of the participant about the agent, (2)

---

[7]https://github.com/HuTobias/HIGHLIGHTS-LRP/tree/master/Survey_videos
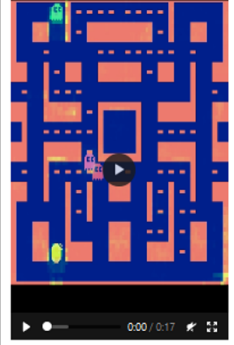
23

participants' ability to assess agents' performance (appropriate trust), and (3) participants' satisfaction with respect to the explanations presented.

Task 1: Eliciting Mental Models through Retrospection. By mental model, we understand the cognitive representation that the participant has about a complex model [30, 56], in our case, the agent. Humans automatically form mental models of agents based on their behavior [10]. These mental models help users understand and explain an agent's behavior. The examination of participants' mental models and their correctness helps to verify if explainable AI has been successfully applied [61, 11]. To evaluate which mental models participants have formed about the agent's behavior, we designed a **retrospection task**. Here we used a task reflection method inspired by prior studies [9, 65], which is recommended by Hoffman et al. [32]. This task asked the participants to analyze the behavior of the three different AI agents, *Regular agent*, *Power pill agent* and *Fear-ghosts agent*. The ordering of the agents was randomized. Specifically, participants were shown the video summary (according to the condition they were assigned to), and were asked to briefly describe the strategy of the AI agent (textual), and to select up to 3 objects that they think were most important to the strategy of the agent (the possible objects were Pacman, power pills, normal pills, ghosts, blue ghosts and cherries). They were also asked how confident they were in their responses, and to justify their reasoning. Figure 6 shows a sketch of a retrospection task.

Task 2: Measuring Appropriate Trust through Agent Comparison. We use the term appropriate trust, based on the work of Lee and See [45] who present a conceptual 'trust in automation' framework. They define appropriate trust as a well-calibrated trust that matches the true capabilities of a technical system. We measure the appropriate trust using an **agent comparison task**. Here, the participants were shown summaries of two of the three agents at a time, and were asked to indicate which agent performs better in the Pacman game (similar to tasks used in [5, 64]). They thus made three comparisons (*Regular agent* Vs. *Power pill agent*, *Regular agent* Vs. *Fear-ghosts agent* and *Power pill agent* Vs. *Fear-ghosts agent*). We do not ask the participant directly about their trust in the two agents shown. Instead, the participants have to choose one of the two agents that they would like to to play on their behalf (see Figure 7). This implicit question reveals which agent participants consider more reliable and qualified for the task. As in the retrospection task, they were asked to indicate their level of confidence and to provide a textual justification for their decision. The ordering of the three agent comparisons was randomized.

Explanation satisfaction questions. Miller [49, 50] argues that the end users' impressions about the agent should be queried and included into the evaluations of the explainable AI methods. This would ensure that the developed explanation methods are comprehensible not only to ML-experts but also to end-users. We address this concern in our study by measuring participants' subjective satisfaction. To this end, we used **explanation satisfaction questions** adapted from the questionnaire proposed by Hoffman et al. [32]. We did this separately for the retrospection task (immediately after completing the
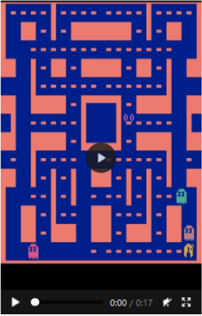
Figure 6: A sketch of the retrospection task: participants were asked to analyze the behavior of each agent by providing a textual description of its strategy and identifying the objects that are most important to its decision-making. The full task can be seen in Appendix D.

three retrospection tasks) and for the agent comparison task (after completing the three comparisons), as we hypothesized there may be differences in the usefulness of the summaries for these two different types of tasks. Specifically, participants were asked the following questions using a 5-point Likert scale:
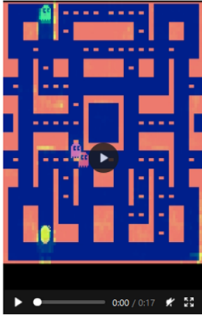
1. From watching the videos of the AI agents, I got an idea of the agents' strategies.
2. The videos showing the AI agents play contain sufficient detail about the agents' behavior.
3. The videos showing the AI agents play contain irrelevant details.
4. The videos showing the AI agents play were useful for *the task*. (only shown in groups R and H)
5. The gameplay scenarios shown in the videos were useful for *the task*. (only shown in groups $R+S$ and $H+S$)

25

Figure 7: A sketch of the agent comparison task: participants were asked to choose which agent they would like to play on their behalf (i.e, identify the better performing agent) according to the two summary videos. The full task can be seen in Appendix D.

6. The green highlighting in the videos was useful for for *the task*. (only shown in groups *R+S* and *H+S*)

We substituted *the task* with either *anlayzing the agents' behavior* or *choosing the agent that performs better*, depending on the task they had just completed.

6.2. Hypotheses

Overall, we hypothesized that HIGHILIGHTS-DIV summaries will be more useful than random summaries in both the retrospection and agent comparison tasks, and that adding saliency maps will further improve participants' performance. More specifically, we state the following hypotheses:

- H1: For both tasks, participants shown summaries generated by HIGHLIGHTS-DIV will perform better than participants shown randomly generated summaries. That is, performance in H will be better than performance in R and performance in $H+S$ will be better than performance in $R+S$. We expect HIGLIGHTS-DIV summaries to be more useful as they demonstrate the agent's behavior in more meaningful states, which should help both in

26

identifying which agent performs better (in line with prior findings [5, 34]), as well as in determining whether an agent is capable of performing well in certain scenarios [34]. We expect similar effects in terms of participants' explanation satisfaction in each task.

- H2: For both tasks, adding saliency maps will improve participant's performance and satisfaction. That is, we expect the performance in $R+S$ will be better than in R and similarly that performance in $H+S$ will be better than in H. Here, too, we expect similar effects in terms of participants' explanation satisfaction in each task. We expect this to be the case as the saliency maps allow people to see not only what actions the agent chooses, but also what information it attends to. Previous studies also found positive effects of saliency maps on participants' mental models [9, 4] and on their ability to choose the better performing prediction model [64]

- H3: The effect of the summary generation method on satisfaction and performance will be greater than that of the inclusion of saliency maps in the agent comparison task. That is, we expect that global information will be more crucial for identifying the better performing agent, as it explicitly demonstrates how the agents act.

- H4: The effect of adding saliency maps on satisfaction and performance will be stronger than that of the summary generation method in the retrospection task. Since saliency maps explicitly show what information the agent attends to, we hypothesize it will contribute more to identifying the agent's strategy. However, this is complicated by the fact that random summaries might not include interesting scenarios, making saliency maps less helpful in this case. Therefore, our more specific hypothesis are:

  - H4.1: Participants in the saliency conditions will be more likely to identify Pacman, the main source of information for our agents, as an important object.
  - H4.2: Participants in the HIGHLIGHTS conditions will be more likely to identify objects that relate to agent goals, such as power pills and blue ghosts. Therefore, they will also more accurately describe the agents' strategies.

6.3. Analysis

We analyze the main hypotheses using the the non-parametric Mann-Whitney test [48], as our dependent variables are not normally distributed. We report effect sizes using rank biserial correlation [72]. Additionally, we report the mean values and the 95% confidence interval (CI) computed using the bootstrap method. In all plots the error bars correspond to the 95% confidence intervals.

To make sure that the participants involved in our analysis did in fact watch the videos of the agents, we recorded whether they clicked play on each video in

addition to how often each video was paused. We did not force them to watch the videos to filter out participants that would have just pressed play to avoid the forcing mechanism. Since we saw from the raw data that some participants only stopped watching videos after the retrospection task, we checked each task separately. As a heuristic to measure how attentively a user watched the videos of a task, we took the sum of pauses of the videos in this task, where watching a video until the end was recorded as a pause and not clicking play was counted as -1 pause. Based on this heuristic we removed all participants from the retrospection task who did not have at least three pauses (5 participants) and all participants from agent comparison task who did not have at least six pauses (11 participants). The number of necessary pauses in each task is equal to the number of videos in this task.

For evaluating the retrospection task we use a scoring system, where two of the authors involved in the training of the agents assigned a score to each item for each agent before the study started (see Appendix C for details). For example for the *Power pill agent*, which was only rewarded when it ate a Power pill, selecting the Power pill or Pacman increased the score by 1 point and including any other item reduced the score by 1 point. Furthermore, selecting more than three items resulted in a score of zero, since the participants were told to select a maximum of three items.

Inspired by Anderson et al. [9] we use summative content analysis [33] to evaluate participants' textual responses. An independent coder (not one of the authors) classified responses to the questions "Please briefly describe the strategy of the AI agent shown in the video above" in the retrospection task, and the question "Please briefly explain how you came to your selection" in both the retrospection task and the agent comparison task. Each question was asked three times (once for each agent description or agent comparison) resulting in 402 answers per question. For the first question, the coder identified 67 different concepts in the answers. For example, the answer "The strategy of this Pacman agents seems to be to mainly avoid the ghosts as it eats the normal pills on the screen. Although it can be seen eating a power pill, the clip still does not show Pacman seeking out and eating the ghosts" was coded to "prioritizing normal pills", "avoiding ghosts" and "do not care about blue ghosts". We aggregated those concepts to 16 groups by combining similar concepts like "eating normal pills" and "prioritizing normal pills".

To evaluate the correctness of participants' answers we implemented a simple scoring system. For each agent and for each answer group, we decided whether it is correct, irrelevant or wrong, based on predefined 'ground-truth' answers that two of the authors, who were involved in the training of the agents, wrote for each agent before the study started. The exact groups and their assigned scores can be found in Appendix C and the open-sourced code.

The answers to the second question regarding participants' justifications of their responses were classified into six categories (the answer could be based on the game rules, the saliency maps, the gameplay, participants' interpretation and two categories for unjustified or unrelated justifications which we grouped into one "unjustified" category) and an additional seventh category for the agent

28

comparison task, that encoded that the user could not decide between the two agent and guessed.

We note that the classifications assigned by the coder are not mutually exclusive.

7. Results

In this section, we report the results of our study. We first describe the characteristics of the participant population with respect to their AI experience, attitude towards AI and Pacman experience. Then we assess the main hypotheses (H1–H4) (results summarized in Table 3) and further provide a descriptive analysis of additional variables such as participants' confidence and analysis of mistakes.

AI and Pacman experience. We verify that participants in different conditions did not differ much in their AI experience and views and in their experience with the game Pacman. To this end we asked them when they played Pacman for the last time and across all four conditions the majority of participants answered: 'I played Pacman more than 5 years ago'. After receiving a short description of what AI is (using a formulation based on Russel [60]), 104 participants stated that they had experience with AI. The exact kind of experience ranged from 'I know AI from the media' (78 participants) to 'I do research on AI related topics' (14 participants). On average the users had a positive attitude towards AI (mean of 3.95 on a 5-point Likert scale). There are no meaningful differences between the conditions (see Appendix A for more details).

(H1) Participants shown HIGHLIGHT-DIV summaries performed better than participants shown random summaries. Participants' correctness rates for the agent comparison task are shown in Figure 8(b). These results support H1, which states that HIGHLIGHTS-DIV summaries will lead to improved performance in both the agent comparison task and the retrospection task. The exact definition of performance per task is described in more detail in section 6.3. Specifically, in the agent comparison task we find that participants in condition H significantly outperformed participants in condition R (H: mean=2.1, 95% CI=[1.83, 2.33], R: mean= 1.63, 95% CI=[1.34, 1.91], Mann-Whitney test U=334.5, $p = 0.014$, r_{rb}=0.3)[8]. While participants in the $H+S$ condition achieved higher mean correctness rates than participants in the $R+S$ condition, this difference is not statistically significant ($H+S$: mean=0.71, 95% CI=[0.6, 0.82], $R+S$: mean=0.65, 95% CI=[0.54, 0.75], Mann-Whitney test U=391, $p = 0.180$, r_{rb}=0.13). Similarly, participants' average explanation satisfaction ratings, shown in Fig. 9(b), indicate that participants in condition H were more satisfied with the videos they received than the other participants. However, this difference is not significant (see Table 3).

---

[8]Here 95% CI is the 95% confidence interval and r_{rb} is Rank biserial correlation.

Task	Variable	Effect of strategy summarization:		Effect of saliency maps:	
		$H > R$	$H+S > R+S$	$R+S > R$	$H+S > H$
retrospection task	score	0.008^*	$3.3e-05^*$	0.965	0.514
	satisfaction	0.021^*	0.035^*	0.677	0.710
	text score				$0.088^{\dagger}$
agent comparison task	score	0.014^*	0.180	$0.062^{\dagger}$	0.307
	satisfaction	0.147	0.235	0.627	0.833

Table 3: Summary of all significance tests (calculated with Mann-Whitney tests). The * denotes statistically significant differences and † denotes a p-value < 0.1.

(a) Total score (summed over all three agents) for the object selection in the retrospection task. The scoring system is described in 6.3.

(b) Number of correct agent selections in the agent comparison task (Out of three selections).
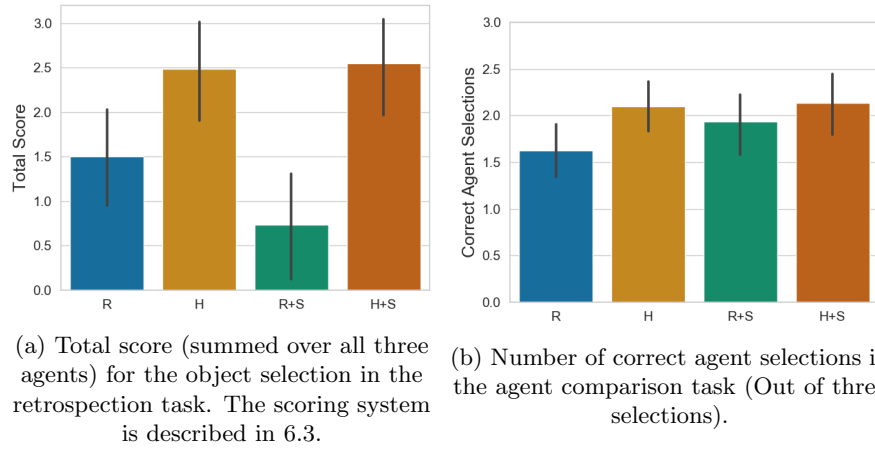
Figure 8: Comparison of participants' average performance in each task, by condition. Participants in the HIGHLIGHTS conditions H and $H+S$ outperformed the random conditions R and $R+S$. Saliency maps only had a slight positive effect when added to random summaries in the agent comparison task

(a) Participants' satisfaction in the retrospection task averaged over all explanations satisfaction questions.

(b) Participants' satisfaction in the agent comparison task averaged over all explanation satisfaction questions.
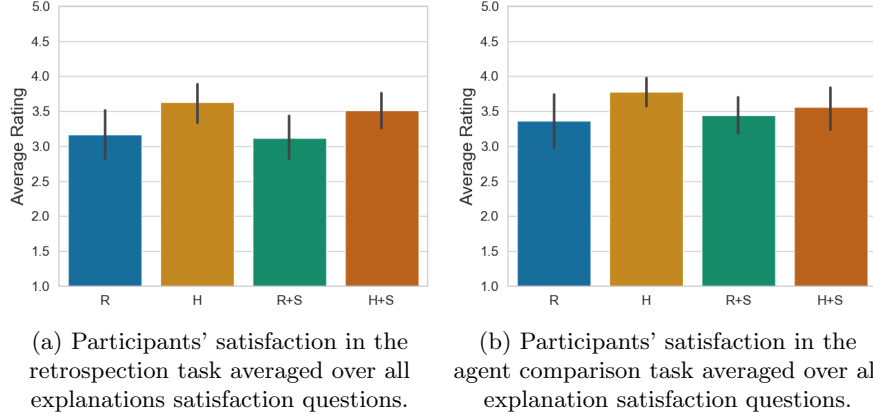
Figure 9: Comparison of participants' average explanation satisfaction in each task, by condition. Each participant rated their agreement with several statements adapted from the explanation satisfaction questions proposed by Hoffman et al. [32] on a 5-point Likert scale (see Section 6.1). Participant's final rating was averaged over all those ratings, reversing the rating of the negative statements. Overall, participants in the HIGHLIGHTS conditions H and $H+S$ rated the explanations highest.

With respect to participants' performance during the retrospection task, we find even stronger results (Fig. 8(a)) then in the agent comparison task, further supporting H1. Here too, participants in condition H obtained a higher score in the object selection sub-task than participants in condition R (H: mean=2.5, 95% CI=[1.89, 3.03], R: mean=1.5, 95% CI=[0.92, 2.06], Mann-Whitney test U=346.5, $p = 0.008$, r_{rb}=0.34) and participants in the $H+S$ condition received a higer score then participants in the $R+S$ condition ($H+S$: mean=2.55, 95% CI=[2.02, 3.06], $R+S$: mean=0.73, 95% CI=[0.13, 1.31], Mann-Whitney test U=206.5, $p = 0.00003$, r_{rb}=0.58). We found analogous significant differences in participants' explanation satisfaction during the retrospection task (Fig. 9(a)). Here, participants in condition H were more satisfied than participants in condition R (H: mean=3.63, 95% CI=[3.35, 3.88], R: mean=3.17, 95% CI=[2.82, 3.5], Mann-Whitney test U=373.0, $p = 0.021$, r_{rb}=0.29) and participants in the $H+S$ condition were more satisfied than participants in the $R+S$ condition ($H+S$: mean=3.52, 95% CI=[3.25, 3.78], $R+S$: mean=3.12, 95% CI=[2.81, 3.43], Mann-Whitney test U=364.5, $p = 0.035$, r_{rb} 0.27).

(H2) Adding saliency maps improved performance in some areas depending on the task. There were no significant differences supporting our second hypothesis H2 which predicted that adding saliency maps will improve participants' performance in both tasks. Nevertheless, we report two positive effects of saliency maps that are only marginally[9] significant and which might guide future research

---

[9]In accordance with convention (Vogt et al. [75]), we use *marginally significant* to describe $0.05 \leq p < 0.1$

in this area. For the agent comparison task, we find that the saliency maps only improved performance when added to random summaries (R: mean=0.54, 95% CI=[0.45, 0.64], $R+S$: mean=0.65, 95% CI=[0.54, 0.75], Mann-Whitney test U=390.5, $p = 0.062$, r_{rb}=0.21). Fig. 8(a) shows that the saliency maps did not help participants identify the most important objects in the retrospection task. However, the summative content analysis of participants' textual descriptions of the agents' strategies, shown in Fig 10, indicates that saliency maps helped participants to correctly describe how the agents use those objects. The descriptions of the agents' strategies written by participants in condition $H+S$ received a higher score than the ones by participants in condition H (H: mean=1.50, 95% CI=[0.97, 2.0], $H+S$: mean=2.13, 95% CI=[1.55, 2.71], Mann-Whitney test U=400, $p = 0.088$, r_{rb}=0.195).

(H3 + H4) The effect of the summary generation method was greater than that of adding saliency maps. We hypothesized that the summary generation method will affect the performance of participants more than the addition of saliency maps in the agent comparison task (H3), and that the saliency maps will have a greater effect than the summary method in the retrospection task (H4). The study results support H3: we found that participants shown HIGHLIGHTS-DIV summaries significantly outperformed participants shown random summaries in the agent comparison task, while adding saliency maps only improved performance for the random summaries, and to a lesser extent.

For selecting the most important objects for the agent's strategy in the retrospection task, the addition of saliency maps did not improve performance, while HIGHLIGHTS-DIV summaries did improve performance compared to the random summaries. Therefore we reject H4, even though the results shown in Fig. 10 indicate that saliency maps improved the textual descriptions of the agent's strategy written by participants in $H+S$ compared to H.

In line with Hypothesis H4.1, Fig. 11 indicates that the improvement of the descriptions of the agents' strategies mainly stems from participants in the saliency groups $R+S$ and $H+S$ identifying that the agent mostly payed attention to the vicinity of Pacman. This effect was not as strong in the object selection question, since it did not capture the participants' reasoning.

Sub-Hypothesis H4.2 stated that strategy summarization would help participants identify the goals of the agents. The results shown in Fig. 12 support this Hypothesis, since participants in the HIGHLIGHTS-DIV conditions H and $H+S$ identified the correct goals of the agent more often.

Participants' Justifications. Across all groups, most participants mainly based their justifications on the agents' gameplay (Fig. B.21). In the saliency conditions, most participants did not mention the saliency maps in their justifications. On average, less than one out of 3 justifications in $H+S$ and in $R+S$ referred to the green highlighting during the retrospection task and during the agent comparison task even fewer participants mentioned them (see Fig. B.22 for more details).
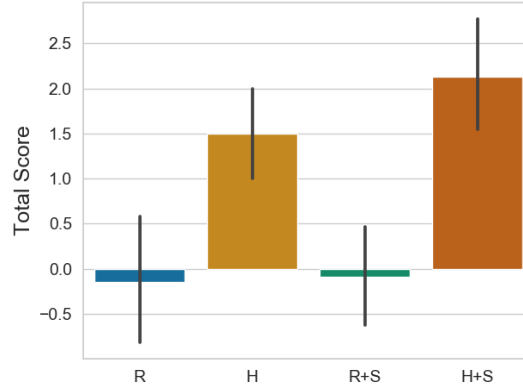
Figure 10: Participants' total score for their textual descriptions of the agents strategy during the agent comparison task (summed over all three agents). The scoring function is described in 6.3. The descriptions of participants in the HIGHLIGHTS-DIV conditions H and $H+S$ received a higher score than those of participants in the random conditions. The addition of saliency maps ($H+S$) slightly improved this effect further.
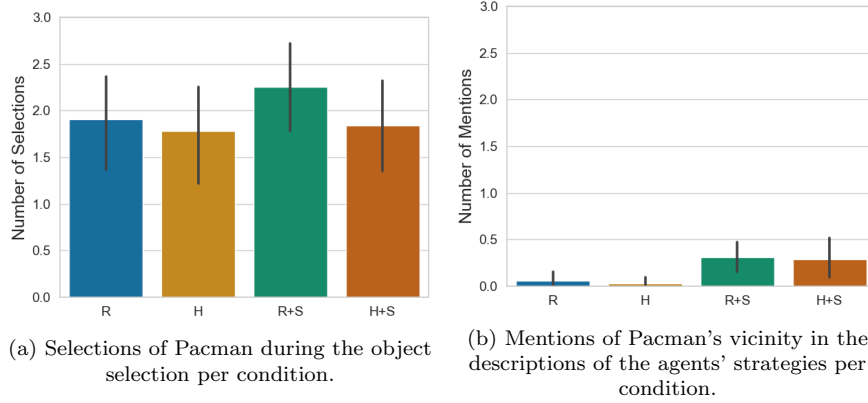

(a) Selections of Pacman during the object selection per condition.

(b) Mentions of Pacman's vicinity in the descriptions of the agents' strategies per condition.

Figure 11: The average number of times that participants correctly selected Pacman during the object selection (a), or referred to its vicinity in their textual descriptions (b) of the agents' strategies (sum over all three agents). The results indicate that saliency maps help the participants to identify what information the agents use.

33

(a) Selections of the agent's specific goals in the object selection, per condition.

(b) Mentions of the agent's specific goals in the strategy descriptions, per condition.
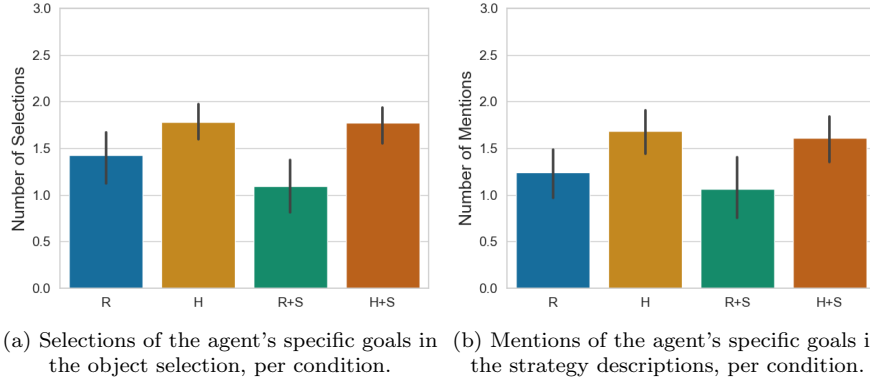
Figure 12: The number of times that participants identified the agent's specific goal in the object selection (a) and strategy description (b) components of the retrospection task. The results are in line with Hypothesis H4.2 that strategy summarization helps to identify the agents' goals.

Another interesting point we found in participants' justifications during the retrospection task is that participants in H gave more unjustified explanations than any other condition (H: mean=0.66, compared to the second highest condition $R+S$: mean=0.38). This is just an observation and did not repeat in the agent comparison task but it might be interesting to investigate further in the future. The values for all conditions can be seen in Fig. B.23.

Participants' confidence and viewing dynamics. In addition to the main metrics used in our study, we further measured participants' confidence (and in particular whether they were more confident when they answered correctly), and their viewing dynamics of the summaries (time and number of pauses). However, apart from a slight positive effect for the participants in condition H, there were no interesting differences in the three aforementioned variables (see Fig. B.18 to B.20 and Appendix B for additional details).

8. Discussion & Future Work

With the increasing use of RL agents in high-stakes domains, there is a growing need in developing and understanding methods for describing the behavior of these agents to their human users. In this paper, we explored the combination of global information describing agent behavior, in the form of strategy summaries, with local information in the form of saliency maps. To this end, we augmented HIGHLIGHTS-DIV [5] summaries, which select important and diverse states (adapted to DQN agents), with saliency maps generated using the LRP-argmax algorithm [36].

We implemented the combined approach in the Atari Pacman environment, and evaluated the separate and joint benefits of showing users global and local information about the agent. We used two types of tasks: a retrospection task about the agent's strategy and a agent comparison task.

Strategy summarization. The results of this study reinforce our prior findings [5] showing that summaries generated by HIGHLIGHTS-DIV lead to significantly improved performance of participants in the agent comparison task compared to random summaries, and show that this result generalizes to RL agents based on neural networks. Furthermore, they show that HIGHLIGHTS-DIV summaries were more useful for analyzing agent strategies and were preferred by participants. Overall, in our study, the choice of states that are shown to participants was more important than the inclusion of local explanations in the form of saliency maps.

Limitations of saliency maps. With respect to the addition of saliency maps, we found mixed results. In contrast to previous studies about saliency maps for image classification tasks, which found weak positive effects for saliency maps [4, 64], there were no significant differences between the saliency and non-saliency conditions in our study. When examining participants' answer justifications, we observed that most participants did not mention utilizing the saliency maps, which may provide a partial explanation to their lack of contribution to participants' performance. Especially in the agent comparison task, participants seldom mentioned the saliency maps even though there was a marginally significant difference between performance of participants in condition R and in condition $R+S$. Participants' comments also reflect their dissatisfaction with saliency maps, e.g., "I do not believe that the green highlighting was useful or relevant" and "The green highlights didn't seem to help much". This suggests that saliency maps in their current form may not be accessible enough to the average user.

Based on the comments from the participants and in depth feedback we received in pilot studies, we note some possible accessibility barriers. First, when saliency maps are shown as part of a video, it may be difficult for users to keep track of the agent's attention, compared to displays of static saliency maps, as done in previous user studies [64, 9, 4]. For instance, one participant reported that "[i]t wasn't so easy to see the green area, it needed to be bigger or more prominent to be of more use." We tried to take measures against this by using a selective saliency map generation algorithm (LRP-argmax) and interpolating between selected saliency maps to reduce the amount of information, as well as allowing participants to pause the video at any time. However, this does not seem to be enough.

Second, participants were not accustomed to interpreting saliency maps, which can be non-intuitive to non-experts. One participant even commented that "[he/she] feel[s] as though this came with somewhat of a learning curve". In our pilot studies we noticed that people who were familiar with reinforcement learning or deep learning could more easily interpret saliency maps than those who were not. For example, some participants said that they thought the agent was good when its attention was spread to different areas because they inferred it considered more information, while in fact the agent was attending to different regions because it did not yet learn what the important information is. Similarly, one study participant commented: "...I don't know if I would prefer an AI

that 'looked' around more at the board, or focused more in a small area to accomplish a task". It is possible that prior studies which used saliency maps for interpreting image classification [4, 64] did not encounter this problem due to the more intuitive nature of the task. Interpreting a visual highlighting for image classification only requires identifying objects that contributed to the classification, while in RL there is an added layer of complexity as interpretation also requires making inferences regarding how the highlighted regions affect the agent's long-term sequential decision-making policy.

Finally, while the sanity checks reported in Section 3.4 showed that our saliency maps do analyze what the network learned, they were also found to be indifferent to specific actions. Since prior studies have shown that users find class discriminatory explanations more useful for understanding agents' decisions [27, 46, 16], the lack of discrimination between certain actions can be detrimental to the usefulness of saliency maps.

Potential of saliency maps. Regarding the potential of saliency maps, we made encouraging observations. Even though saliency maps did not significantly increase participants' scores in the simple object selection part of the retrospection task, they did result in improved scores in the textual strategy description. The difference between our HIGHLIGHTS-DIV conditions $H+S$ and H is similar to the one observed by Anderson et al. [9] ($p=0.086$ compared to our $p=0.088$), who also evaluated participants' mental models for RL agents utilizing a strategy description task. The poor result of our random condition $R+S$ can be explained by the fact that Anderson et al. implicitly chose meaningful states, which we only did with our global explanation method in the HIGHLIGHTS-DIV conditions.

A possible reason for the difference between the object selection and the strategy description sub-tasks is the higher complexity of strategy description. It requires participants to not only identify the correct objects but also to describe how they are used. Under this assumption, the increased performance of participants in condition $H+S$ suggests that saliency maps were useful for putting the objects in the correct context. For example, participants' textual descriptions showed that, while the non-saliency groups know that Pacman is important (most likely based on the fact that it is important for them as players), they did not identify it as a central source of information for the agent.

Second, we observed in the agent comparison task that saliency maps alone improved participants' ability to place appropriate trust into different agents when comparing conditions R and $R+S$. There, performance was comparable to the performance of participants in the HIGHLIGHTS-DIV conditions, H and $H+S$. This indicates that there is valuable information for this kind of task within saliency maps. The lacking improvement of condition $H+S$ compared to H might be explained by the accessibility issues of saliency maps mentioned earlier. When presented with strategy summaries, participants may have had less reason to rely on the non-intuitive saliency maps.

Combination of local and global explanations. It is important to note that the positive effects of saliency maps in the retrospection task were only visible in the HIGHLIGHTS-DIV conditions H and $H+S$, reinforcing our claim that the choice of states is crucial for explaining RL agents. Therefore, even if the limitations of saliency maps mentioned above are addressed, the potential benefits might only be visible and likely reinforced by a combination with strategy summarization techniques. We note that studies that evaluate local explanations typically implicitly make a global decision about which states to present local explanations for [9, 47]. Our results suggest that this implicit choice may have a substantial impact on participants' understanding of agent behavior.

In the retrospection task, we observed that local explanations in the form of saliency maps were useful for identifying what objects the agent attends to (see Fig. 11), while strategy summaries were more useful for identifying the agent's goals (see Fig. 12). This was reflected by participants' utterances such as: "The agent seemed to be paying attention to the area directly in front of it and partly to the areas directly to each side." and "Pacman wanted those ghosts! His goal was to move as fast as he could towards them." and suggests that the two approaches are indeed complementary. The local saliency maps contribute to users' understanding of the agents *attention*, as they reflect the information the agent attends to, while strategy summaries contribute to users' understanding of the agent's *intentions*, as they reflect how the agent acts.

Taken together, our results suggest that there is potential for a combined explanation framework in the future, if the accessibility issues of saliency maps are addressed.

Study limitations. Our study has several limitations. First, we used a single domain in our user study. However, other recent work has used strategy summaries similar in spirit to HIGHLIGHTS-DIV in another domain [65] and several works have used saliency maps in other domains (e.g., several Atari games including Pong and Space invaders were used by Greydanus et al. [28]).

Second, while our combined explanation approach is easily adaptable to other global explanation methods which choose an informative subset of states, and local methods that highlight relevant information in those states, our study only explored one combination of a particular global explanation method and a particular local explanation method. We chose the HIGHLIGHTS-DIV summary method since strategy summary approaches that are based on policy reconstruction require making various assumptions about people's computational models, and that these models differ depending on context [43]. We chose saliency maps as a local method both because it is visual and thus can be integrated with a visual summary, and also because other methods typically require additional models or assumptions (e.g., causal explanations [47] require a causal graph of the domain). The specific choice of the LRP-argmax algorithm was motivated by its selectivity, which reduces the amount of information that participants have to process. The accessibility problems of saliency maps we identified were mainly related to the presentation of the information. This indicates that simply highlighting how relevant parts of the input are for the

prediction of an agent is insufficient even when based on other saliency map algorithms.

Future work. There are several directions we intend to explore in future work. First, as discussed earlier, there is a need to make saliency maps more understandable to users. To this end, we plan to augment saliency maps with textual explanations that help users interpret the information correctly, similar to how Rabold et al. [57] did with LIME explanations. Hereby, we aim to train a machine learning model on descriptions written by domain experts confronted with the combination of HIGHLIGHTS-DIV and saliency maps presented in this work. Furthermore, we plan to build up on our previous work [79] and explore the presentation of those textual explanations through virtual agents.

Second, we plan to explore interaction approaches that involve the user in the process, e.g., by only showing local information when the user asks for it as we did in the context of cooperative annotation [13]. This could reduce cognitive load while increasing the user's attention to the local information when it is needed.

Finally, to verify that our results generalize beyond simulated environments, we would like to conduct user studies in real-world domains such as healthcare. Explainability is crucial in AI systems deployed in the medical field (e.g., pain classification [78]) since possible errors could lead to dire consequences. RL methods face additional challenges and requirements in the healthcare domain where random exploration of the state space is not possible and evaluation is challenging [25, 26], making explanation methods even more important. In recent work, we have begun exploring the use of strategy summaries in healthcare using an HIV simulator [43], and intend to further explore this direction.

9. Conclusion

This work is a first step toward the development of combined explanation methods for reinforcement learning (RL) agents that provide users with both global information regarding the agent's strategy, as well as local information regarding its decision-making in specific world-states. To this end, we present a joint global and local explanation method, building on our prior work on strategy summaries (HIGHLIGHTS-DIV) and on generating saliency maps for deep RL agents (LRP-argmax). This method is easily adaptable to other global and local algorithms.

To evaluate this combined global and local explanation method, as well as the contribution of each explanation type, we conducted a user study. Hereby, we examined participants mental models through a retrospection task and used an agent comparison task to investigate whether their trust was appropriate given agents' capabilities.

Regarding the usefulness of *global strategy summaries*, our results show that HIGHLIGHTS-DIV summaries (1) help to establish appropriate trust in agents based on neural networks (extending prior results about classic RL agents [5]) and (2) improve participants' mental models of those agents.

The evaluation of *local explanations* in the form of LRP saliency maps reveals strengths as well as weaknesses. On the one hand, our analysis shows that reinforcement learning comes with additional usability challenges not present in previously evaluated image classification tasks. First, presenting saliency maps on videos instead of static images [9, 4] overwhelms users with a lot of information in a short amount of time and increases the risk of overlooking crucial information. Second, compared to more intuitive image classification tasks [4, 64], the average users lacks experience to correctly infer how the highlighted regions affect the agent's long-term sequential decision-making.

On the other hand, the results indicate that saliency maps have the potential to (1) extend users' mental models beyond strategy summaries by providing insight into what information the agent used and (2) improve users' ability to choose the better agent even with random summaries.

Taken together, the results support a combination of local and global explanations, since participants in the combined explanation condition received the highest scores during our survey. However, our evaluation suggests that simply highlighting pixels that are relevant for the agent's decision is insufficient for RL agents and that more work is needed to increase the accessibility of saliency maps.

References

[1] Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B., 2018. Sanity checks for saliency maps. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 31. Curran Associates, Inc., pp. 9505–9515.
URL   http://papers.nips.cc/paper/8160-sanity-checks-for-saliency-maps.pdf

[2] Aha, D., Darrell, T., Pazzani, M., Reid, D., Sammut, C., Stone, P., 2017. IJCAI-17 workshop on explainable AI (XAI). In: IJCAI-17 Workshop on Explainable AI (XAI).

[3] Alber, M., Lapuschkin, S., Seegerer, P., Hägele, M., Schütt, K. T., Montavon, G., Samek, W., Müller, K.-R., Dähne, S., Kindermans, P.-J., 2018. iNNvestigate neural networks! arXiv preprint arXiv:1808.04260.

[4] Alqaraawi, A., Schuessler, M., Weiß, P., Costanza, E., Berthouze, N., 2020. Evaluating saliency map explanations for convolutional neural networks: A user study. arXiv preprint arXiv:2002.00772.

[5] Amir, D., Amir, O., 2018. Highlights: Summarizing agent behavior to people. In: Proc. of the 17th International conference on Autonomous Agents and Multi-Agent Systems (AAMAS).

[6] Amir, O., Doshi-Velez, F., Sarne, D., 2018. Agent strategy summarization. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. International Foundation for Autonomous Agents and Multiagent Systems, pp. 1203–1207.

[7] Amir, O., Doshi-Velez, F., Sarne, D., 2019. Summarizing agent strategies. Autonomous Agents and Multi-Agent Systems 33 (5), 628–644.

[8] Amir, O., Kamar, E., Kolobov, A., Grosz, B. J., 2016. Interactive teaching strategies for agent training. International Joint Conferences on Artificial Intelligence.

[9] Anderson, A., Dodge, J., Sadarangani, A., Juozapaitis, Z., Newman, E., Irvine, J., Chattopadhyay, S., Fern, A., Burnett, M., 7 2019. Explaining reinforcement learning to mere mortals: An empirical study. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. International Joint Conferences on Artificial Intelligence Organization, pp. 1328–1334.
URL https://doi.org/10.24963/ijcai.2019/184

[10] Anjomshoae, S., Najjar, A., Calvaresi, D., Främling, K., 2019. Explainable agents and robots: Results from a systematic literature review. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. AAMAS 19. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, p. 10781088.

[11] Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F., 2020. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion 58, 82 – 115.
URL   http://www.sciencedirect.com/science/article/pii/S1566253519308103

[12] Bach, S., Binder, A., Montavon, G., Klauschen, F., Mller, K.-R., Samek, W., 07 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLOS ONE 10 (7), 1–46.
URL https://doi.org/10.1371/journal.pone.0130140

[13] Baur, T., Heimerl, A., Lingenfelser, F., Wagner, J., Valstar, M. F., Schuller, B., André, E., 2020. eXplainable cooperative machine learning with NOVA. KI-Künstliche Intelligenz, 1–22.

[14] Bellemare, M. G., Naddaf, Y., Veness, J., Bowling, M., jun 2013. The arcade learning environment: An evaluation platform for general agents. Journal of Artificial Intelligence Research 47, 253–279.

[15] Booth, S., Muise, C., Shah, J., 7 2019. Evaluating the interpretability of the knowledge compilation map: Communicating logical statements effectively. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. International Joint Conferences on Artificial Intelligence Organization, pp. 5801–5807.
URL https://doi.org/10.24963/ijcai.2019/804

[16] Byrne, R. M. J., 7 2019. Counterfactuals in explainable artificial intelligence (XAI): Evidence from human reasoning. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. International Joint Conferences on Artificial Intelligence Organization, pp. 6276–6282.
URL https://doi.org/10.24963/ijcai.2019/876

[17] Cashmore, M., Collins, A., Krarup, B., Krivic, S., Magazzeni, D., Smith, D., 2019. Towards explainable AI planning as a service. arXiv preprint arXiv:1908.05059.

[18] Chakraborti, T., Sreedharan, S., Grover, S., Kambhampati, S., 2019. Plan explanations as model reconciliation. In: 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI). IEEE, pp. 258–266.

[19] Chandrasekaran, B., Tanner, M. C., Josephson, J. R., 1989. Explaining control strategies in problem solving. IEEE Intelligent Systems (1), 9–15.

[20] Dhariwal, P., Hesse, C., Klimov, O., Nichol, A., Plappert, M., Radford, A., Schulman, J., Sidor, S., Wu, Y., Zhokhov, P., 2017. Openai baselines. https://github.com/openai/baselines.

[21] Dodson, T., Mattei, N., Goldsmith, J., 2011. A natural language argumentation interface for explanation generation in markov decision processes. Algorithmic Decision Theory, 42–55.

[22] Doshi-Velez, F., Kim, B., 2017. A roadmap for a rigorous science of interpretability. arXiv preprint arXiv:1702.08608.

[23] Erwig, M., Fern, A., Murali, M., Koul, A., 2018. Explaining deep adaptive programs via reward decomposition. In: IJCAI/ECAI Workshop on Explainable Artificial Intelligence.

[24] Goel, V., Weng, J., Poupart, P., 2018. Unsupervised video object segmentation for deep reinforcement learning. In: Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, Montréal, Canada. pp. 5688–5699.
URL http://papers.nips.cc/paper/7811-unsupervised-video-object-segmentation-for-deep-reinforcement-learning

[25] Gottesman, O., Johansson, F., Komorowski, M., Faisal, A., Sontag, D., Doshi-Velez, F., Celi, L. A., 2019. Guidelines for reinforcement learning in healthcare. Nat Med 25 (1), 16–18.

[26] Gottesman, O., Johansson, F., Meier, J., Dent, J., Lee, D., Srinivasan, S., Zhang, L., Ding, Y., Wihl, D., Peng, X., et al., 2018. Evaluating reinforcement learning algorithms in observational health settings. arXiv preprint arXiv:1805.12298.

[27] Goudet, O., Kalainathan, D., Caillou, P., Guyon, I., Lopez-Paz, D., Sebag, M., 2018. Learning Functional Causal Models with Generative Neural Networks. Springer International Publishing, Cham, pp. 39–80.
URL https://doi.org/10.1007/978-3-319-98131-4_3

[28] Greydanus, S., Koul, A., Dodge, J., Fern, A., 2018. Visualizing and understanding atari agents. In: Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden. pp. 1787–1796.

[29] Gunning, D., 2017. Explainable artificial intelligence (XAI). Defense Advanced Research Projects Agency (DARPA), nd Web.

[30] Halasz, F. G., Moran, T. P., 1983. Mental models and problem solving in using a calculator. In: Proceedings of the SIGCHI conference on Human Factors in Computing Systems. pp. 212–216.

[31] Hayes, B., Shah, J. A., 2017. Improving robot controller transparency through autonomous policy explanation. In: Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction. ACM, pp. 303–312.

[32] Hoffman, R. R., Mueller, S. T., Klein, G., Litman, J., 2018. Metrics for explainable AI: Challenges and prospects. arXiv preprint arXiv:1812.04608.

[33] Hsieh, H.-F., Shannon, S. E., 2005. Three approaches to qualitative content analysis. Qualitative health research 15 (9), 1277–1288.

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

[35] Huang, S. H., Held, D., Abbeel, P., Dragan, A. D., 2017. Enabling robots to communicate their objectives. In: Robotics: Science and Systems XIII, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, July 12-16, 2017. pp. 1–18.
URL http://www.roboticsproceedings.org/rss13/p59.html

[36] Huber, T., Schiller, D., André, E., 2019. Enhancing explainability of deep reinforcement learning through selective layer-wise relevance propagation. In: Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz). Springer, pp. 188–202.

[37] Iyer, R., Li, Y., Li, H., Lewis, M., Sundar, R., Sycara, K. P., 2018. Transparency and explanation in deep reinforcement learning neural networks. In: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2018, New Orleans, LA, USA. pp. 144–150.

[38] Khan, O., Poupart, P., Black, J., Sucar, L., Morales, E., Hoey, J., 2011. Automatically generated explanations for markov decision processes. Decision Theory Models for Applications in AI: Concepts and Solutions, 144–163.

[39] Khan, O. Z., Poupart, P., Black, J. P., 2009. Minimal sufficient explanations for factored markov decision processes. In: ICAPS.

[40] Kim, B., Khanna, R., Koyejo, O. O., 2016. Examples are not enough, learn to criticize! criticism for interpretability. In: Advances in Neural Information Processing Systems. pp. 2280–2288.

[41] Krarup, B., Cashmore, M., Magazzeni, D., Miller, T., 2019. Model-based contrastive explanations for explainable planning. In: Proceedings of the ICAPS 2019 Workshop on Explainable Planning (XAIP).

[42] Kulkarni, A., Zha, Y., Chakraborti, T., Vadlamudi, S. G., Zhang, Y., Kambhampati, S., 2019. Explicable planning as minimizing distance from expected behavior. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. International Foundation for Autonomous Agents and Multiagent Systems, pp. 2075–2077.

[43] Lage, I., Lifschitz, D., Doshi-Velez, F., Amir, O., 7 2019. Exploring computational user models for agent policy summarization. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. International Joint Conferences on Artificial Intelligence Organization, pp. 1401–1407.
URL https://doi.org/10.24963/ijcai.2019/194

[44] Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., Müller, K.-R., 2019. Unmasking clever hans predictors and assessing what machines really learn. Nature Communications 10 (1), 1096.

[45] Lee, J. D., See, K. A., 2004. Trust in automation: Designing for appropriate reliance. Human factors 46 (1), 50–80.

[46] Lopez-Paz, D., Nishihara, R., Chintala, S., Scholkopf, B., Bottou, L., July 2017. Discovering causal signals in images. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[47] Madumal, P., Miller, T., Sonenberg, L., Vetere, F., 2019. Explainable reinforcement learning through a causal lens. arXiv preprint arXiv:1905.10958.

[48] McKnight, P. E., Najab, J., 2010. Mann-whitney u test. The Corsini encyclopedia of psychology, 1–1.

[49] Miller, T., 2018. Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence 267, 1–38.

[50] Miller, T., Howe, P., Sonenberg, L., 2017. Explainable ai: Beware of inmates running the asylum. In: IJCAI-17 Workshop on Explainable AI (XAI). Vol. 36.

[51] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al., 2015. Human-level control through deep reinforcement learning. Nature 518 (7540), 529–533.

[52] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al., 2015. Human-level control through deep reinforcement learning. Nature 518 (7540), 529.

[53] Molnar, C., 2019. Interpretable machine learning: A guide for making black box models explainable.(2019). URL https://christophm. github. io/interpretable-ml-book.

[54] Montavon, G., Samek, W., Müller, K., 2018. Methods for interpreting and understanding deep neural networks. Digital Signal Processing 73, 1–15. URL https://doi.org/10.1016/j.dsp.2017.10.011

[55] Mopuri, K. R., Garg, U., Babu, R. V., 2019. CNN fixations: an unraveling approach to visualize the discriminative image regions. IEEE Transactions on Image Processing 28 (5), 2116–2125.

[56] Norman, D. A., 2014. Some observations on mental models. In: Mental models. Psychology Press, pp. 15–22.

[57] Rabold, J., Deininger, H., Siebers, M., Schmid, U., 2019. Enriching visual with verbal explanations for relational concepts - combining LIME with Aleph. In: PKDD/ECML Workshops.

[58] Rao, A. S., Georgeff, M. P., et al., 1995. BDI agents: from theory to practice. In: ICMAS. Vol. 95. pp. 312–319.

[59] Ribeiro, M. T., Singh, S., Guestrin, C., 2016. Why should i trust you?: Explaining the predictions of any classifier. In: Proc. of ACM International Conference on Knowledge Discovery and Data Mining. ACM, pp. 1135–1144.

[60] Russell, S., Norvig, P., 2016. Artificial intelligence: A modern approach global edition. Pearson.

[61] Rutjes, H., Willemsen, M., IJsselsteijn, W., 5 2019. Considerations on explainable ai and users mental models. In: Where is the Human? Bridging the Gap Between AI and HCI. Association for Computing Machinery, Inc, United States.

[62] Schulz, K., Sixt, L., Tombari, F., Landgraf, T., 2020. Restricting the flow: Information bottlenecks for attribution. CoRR abs/2001.00396.
URL http://arxiv.org/abs/2001.00396

[63] Seegebarth, B., Müller, F., Schattenberg, B., Biundo, S., 2012. Making hybrid plans more clear to human users-a formal approach for generating sound explanations. In: Twenty-Second International Conference on Automated Planning and Scheduling.

[64] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2020. Grad-CAM: Visual explanations from deep networks via gradient-based localization. Int. J. Comput. Vis. 128 (2), 336–359.
URL https://doi.org/10.1007/s11263-019-01228-7

[65] Sequeira, P., Yeh, E., Gervasio, M. T., 2019. Interestingness elements for explainable reinforcement learning through introspection. In: IUI Workshops. p. 7.

[66] Simonyan, K., Vedaldi, A., Zisserman, A., 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. CoRR abs/1312.6034.
URL http://arxiv.org/abs/1312.6034

[67] Sixt, L., Granz, M., Landgraf, T., 2019. When explanations lie: Why modified BP attribution fails. CoRR abs/1912.09818.

[68] Springenberg, J. T., Dosovitskiy, A., Brox, T., Riedmiller, M. A., 2014. Striving for simplicity: The all convolutional net. CoRR abs/1412.6806.

[69] Stone, P., Brooks, R., Brynjolfsson, E., Calo, R., Etzioni, O., Hager, G., Hirschberg, J., Kalyanakrishnan, S., Kamar, E., Kraus, S., Leyton-Brown, K., Parkes, D., William, P., AnnaLee, S., Julie, S., Milind, T., Astro, T., 2016. Artificial intelligence and life in 2030. One Hundred Year Study on Artificial Intelligence: Report of the 2015-2016 Study Panel.

[70] Sundararajan, M., Taly, A., Yan, Q., 2017. Axiomatic attribution for deep networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, pp. 3319–3328.

[71] Swartout, W. R., 1983. Xplain: A system for creating and explaining expert consulting programs. Artificial intelligence 21 (3), 285–325.

[72] Tomczak, M., Tomczak, E., 2014. The need to report effect size estimates revisited. an overview of some recommended measures of effect size. Trends in Sport Sciences 21 (1).

[73] Torrey, L., Taylor, M., 2013. Teaching on a budget: Agents advising agents in reinforcement learning. In: Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems. International Foundation for Autonomous Agents and Multiagent Systems, pp. 1053–1060.

[74] van der Waa, J., van Diggelen, J., van den Bosch, K., Neerincx, M. A., 2018. Contrastive explanations for reinforcement learning in terms of expected consequences. CoRR abs/1807.08706.
URL http://arxiv.org/abs/1807.08706

[75] Vogt, W., 2005. Dictionary of Statistics & Methodology: A Nontechnical Guide for the Social Sciences. Sage Publications.

[76] Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M., de Freitas, N., 2016. Dueling network architectures for deep reinforcement learning. In: Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA. pp. 1995–2003.

[77] Weitkamp, L., van der Pol, E., Akata, Z., Feb. 2019. Visual rationalizations in deep reinforcement learning for atari games. CoRR arXiv:1902.00566.

[78] Weitz, K., Hassan, T., Schmid, U., Garbas, J.-U., 2019. Deep-learned faces of pain and emotions: Elucidating the differences of facial expressions with the help of explainable AI methods. tm-Technisches Messen 86 (7-8), 404–412.

[79] Weitz, K., Schiller, D., Schlagowski, R., Huber, T., André, E., 2019. "Do you trust me?": Increasing user-trust by integrating virtual agents in explainable AI interaction design. In: Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents. IVA '19. ACM, New York, NY, USA, pp. 7–9.

[80] Zahavy, T., Ben-Zrihem, N., Mannor, S., 2016. Graying the black box: Understanding DQNs. In: Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA. pp. 1899–1908.

[81] Zeiler, M. D., Fergus, R., 2014. Visualizing and understanding convolutional networks. In: Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, Proceedings, Part I. pp. 818–833.

[82] Zhang, B., Dafoe, A., 2019. Artificial intelligence: American attitudes and trends. Available at SSRN 3312874.

Appendix A. Participants Demographics

In this section, we provide moe details regarding participants' demographics. As Fig. A.13 shows, most participants were between 18 and 34 years old. There were no major differences in gender distribution between the four conditions (Fig. A.14).
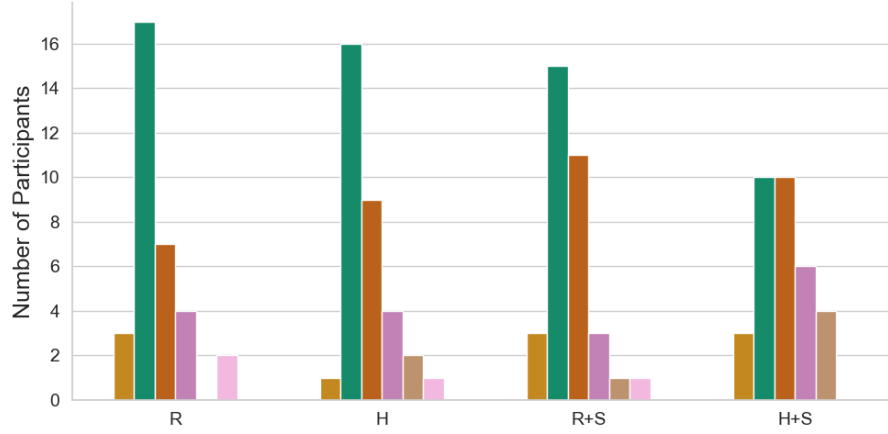
Figure A.13: The number of participants in each age group per condition. The bars show from left to right: "18-24", "25-34","35-44", " 45-54", "55-64" and "65 or older". The categories "17 or younger" and "do not want to specify" were never selected.
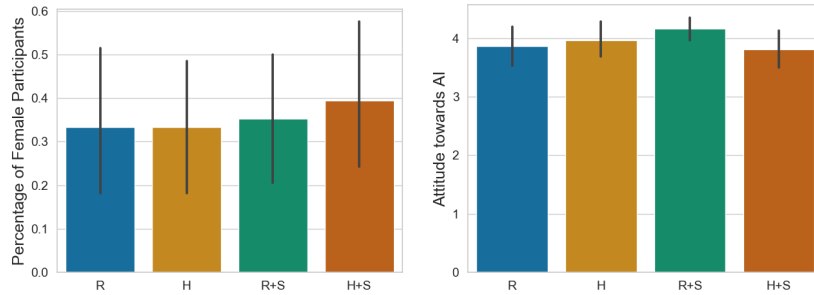
Figure A.14: Number of female participants per condition.

Figure A.15: The average attitude towards AI, rated on a 5 point Likert scale.

We verified that participants in different conditions did not differ much in their AI experience and views and in their Pacman experience. To this end, we asked them when they played Pacman for the last time (1="never", 2="more than 5 years ago", 3="less then 5 years ago", 4="less than 1 year ago"). Across all four conditions the median group was 2:"I played Pacman more than 5 years ago". A comparison is shown in figure A.16.

47
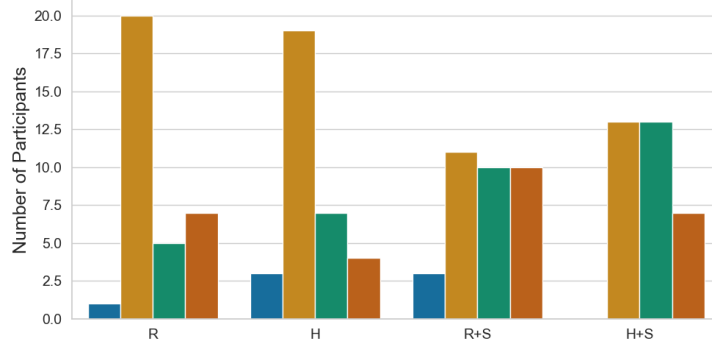
Figure A.16: The Pacman experience across all conditions where the bars depict when the participants played Pacman the last time. From left to right the bars represent: "never", "more than 5 years ago", "less then 5 years ago" and "less than 1 year ago".

For the AI experience we adapted a description of AI from Zhang et al. [82] and Russel [60] to "The following questions ask about Artificial Intelligence (AI). Colloquially, the term 'artificial intelligence' is often used to describe machines (or computers) that mimic 'cognitive' functions that humans associate with the human mind, such as 'learning' and 'problem solving'. AI agents are already able to perform some complex tasks better than the median human (today). Examples for such intelligent agents are search engines, chatbots, chessbots and voice assistants."

After that, every participant who stated to have AI experience (104 across all conditions) had to select one or more of the following items:

- 1: I know AI from the media.

- 2: I use AI technology in my private life.

- 3: I use AI technology in my work.

- 4: I took at least one AI related course.

- 5: I do research on AI related topics.

- Other:

The last free form option was used exactly once and read "work on MTurk". The distribution of the other items for each condition is shown in Fig. A.17.

To measure the participants' attitude towards AI we adapted a question from Zhang et al [82] and asked them to rate their answer to the question "Suppose that AI agents would achieve high-level performance in more areas one day. How positive or negative do you expect the overall impact of such AI agents to be on humanity in the long run?" on scale from 5 point Likert scale from "Extremely negative" to "Extremely positive". The results are shown in Fig. A.15.

48
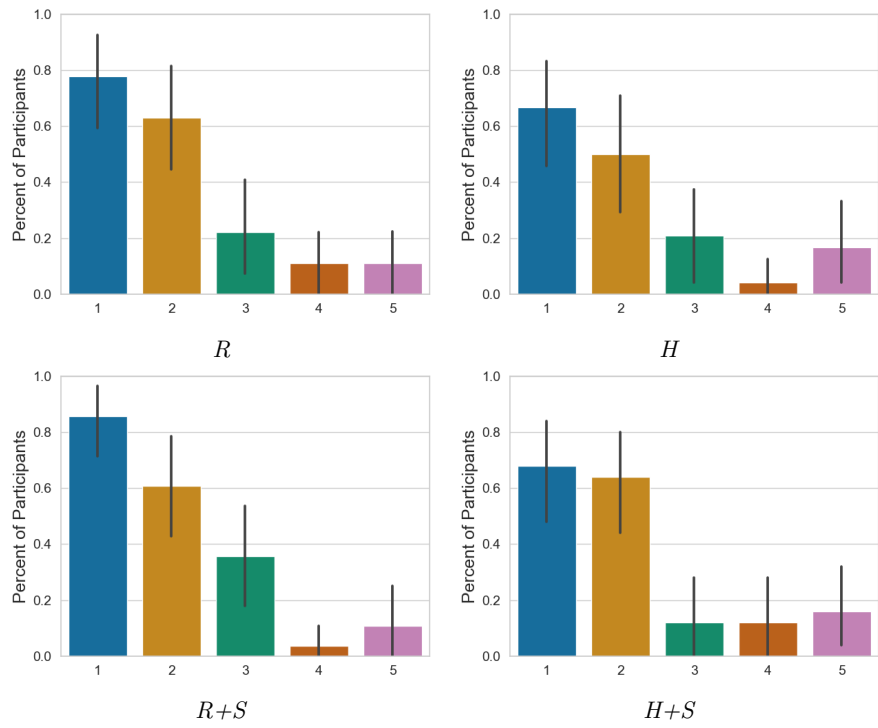
Figure A.17: Distribution of the chosen AI experience items for each condition. The x-axis depicts the items described above.

Appendix B. Supplementary Results

In this section, we present additional information about the results of the study that goes beyond the main hypotheses we explored and described in the paper.

Confidence, time and pauses. To investigate whether participants were confident in their decisions , they had to rate the confidence in each of their selections (item selection in the retrospection task and agent selection in the agent comparison task) on a 7 point Likert scale. The results across each task are shown in Fig. B.18.

(a) retrospection task   (b) agent comparison task
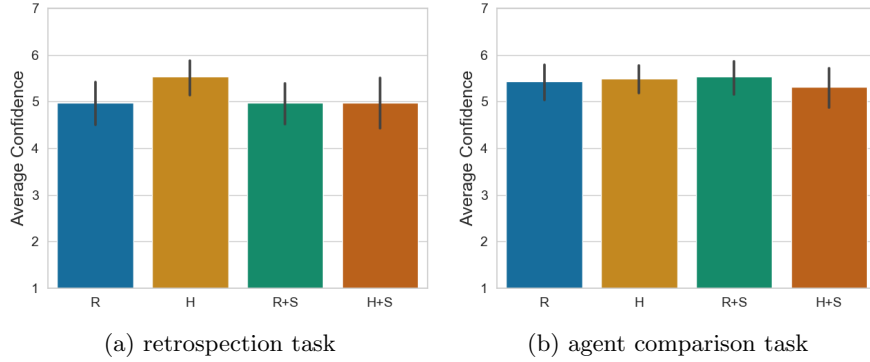
Figure B.18: The average confidence that participants in each condition had in their answers during each task.

To evaluate whether participants were especially diligent or effective during the tasks, we measured the time that each participant stayed on each of page of the survey and calculated the average time per task (each task consists of three pages). Furthermore, we kept track of each time a video was paused, as described in section 6.3. The average completion times of participants and the average number of pauses are shown in Fig. B.19 and B.20, respectively (shown in boxplots due to the presence of several outliers that strongly affect the mean values).

Fig. B.18 (a) shows that participants in condition *H* were slightly more confident on average in their analysis of the agents. This is also reflected by the lesser amount of time per analysis (Fig. B.19 (a)) and pauses (Fig. B.20 (a)). Apart from this, there are no obvious differences between the average confidence, time and pause values for each task (Fig. B.18 to B.20).

Participants' justifications. As described in section 6.3, an independent coder identified different concepts inside the participants' justifications. Figure B.22 shows the average number of mentions of *gameplay* and of *saliency maps* in the different tasks, across the different conditions. As discussed in section 7, most participants mainly based their justifications on the agents' gameplay (Fig. B.21) and, in the saliency conditions, participants seldom mention the

50

(a) retrospection task

(b) agent comparison task
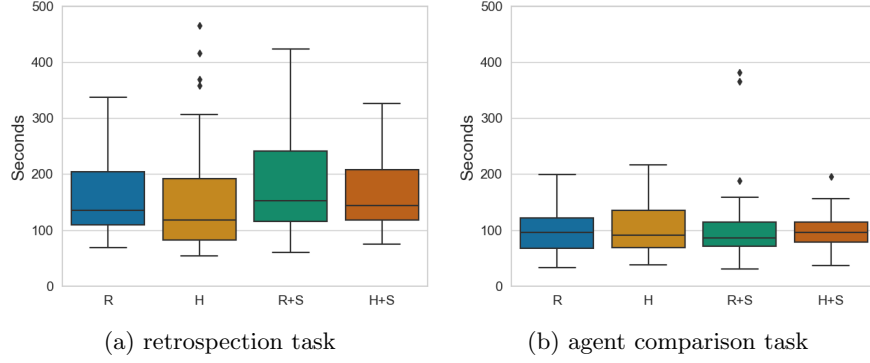
Figure B.19: The average time taken by participants in each condition per agent analysis (a) and comparison of agent pairs (b).
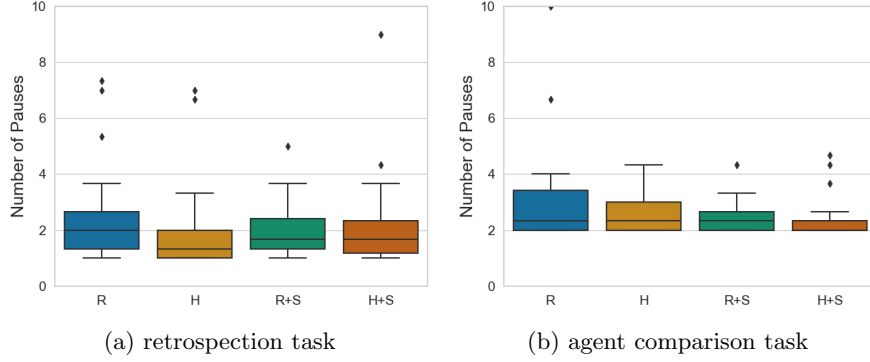

(a) retrospection task

(b) agent comparison task

Figure B.20: The average number of times that participants in each condition paused the videos during each agent analysis (a) and comparison of agent pairs (b).

saliency maps in their justifications (see Fig. B.22). Finally, Fig. B.23 shows that participants in condition H gave more unjustified explanations in the retrospection task. However, this observation did not repeat in the agent comparison task.

Appendix C. Evaluation of the Retrospection Task

As described in section 6.3, we evaluated participants' scores in the object selection part of the retrospection task with a simple scoring function based on predefined answers by two of the authors involved in the training of the agents. Hereby, we assign a score of 1 to each object that is connected to the agents' specific goal and their source of information (Pacman's position for all agents), -1 for each object that was not related to the agents' reward function and -0.5 to objects that were related to the reward but on which the agent did not focus. The specific scores are shown in table C.24.

retrospection task   agent comparison task
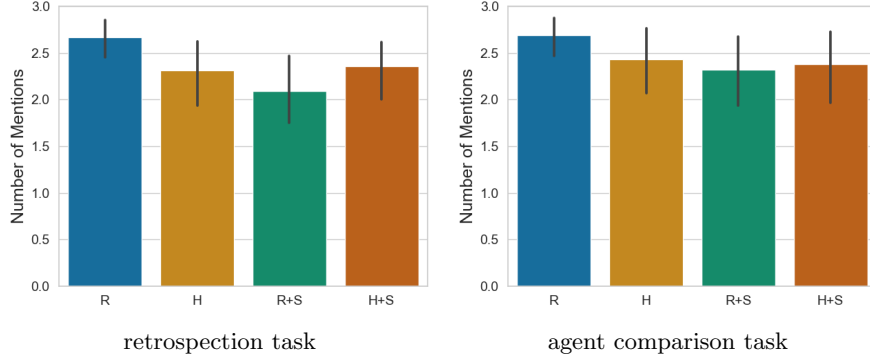
Figure B.21: Comparison of how often the participants referenced the agents' **gameplay** in their justifications for their answers.

(a) retrospection task   (b) agent comparison task
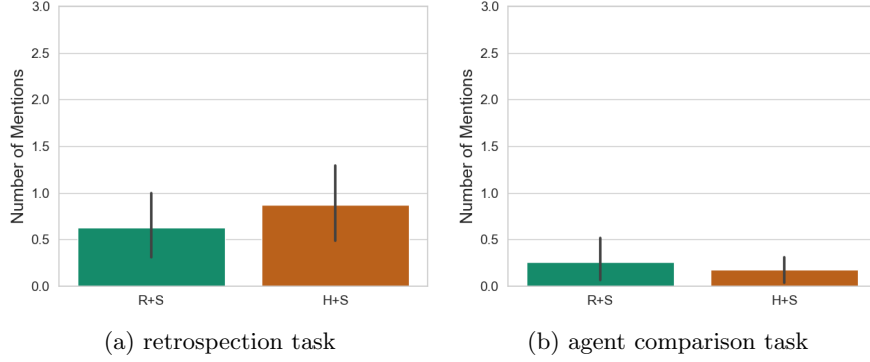
Figure B.22: Comparison of how often the participants referenced the green highlighting of the LRP-argmax **saliency maps** in their justifications for their answers.

For the free form answers to the question "Please describe the strategy of the AI agent" an independent coder identified various not mutually exclusive concepts contained in the participants answers. We aggregated these concepts into the following 16 groups, where the coder used 'G' for ghosts, 'PP' for power pills and 'NP' for normal pills:

1. *eating power pills*: "eating PP", "eating as many PP as possible", "eat PP when ghosts are near", "eat PP when ghosts are near", "prioritizing PP", "prioritizing PP to eat ghosts", "prioritizing PP , but not eat ghosts", "eat PP to get points"
2. *ignore power pills*: "do not care about PP"
3. *eat normal pills*: "eat NP to get points", "eating NP", "eating as many NP as possible", "prioritizing NP", "clearing the stage"
4. *ignore normal pills*: "do not care about NP", "focus on areas wihtout [sic] NP"

52

(a) retrospection task   (b) agent comparison task
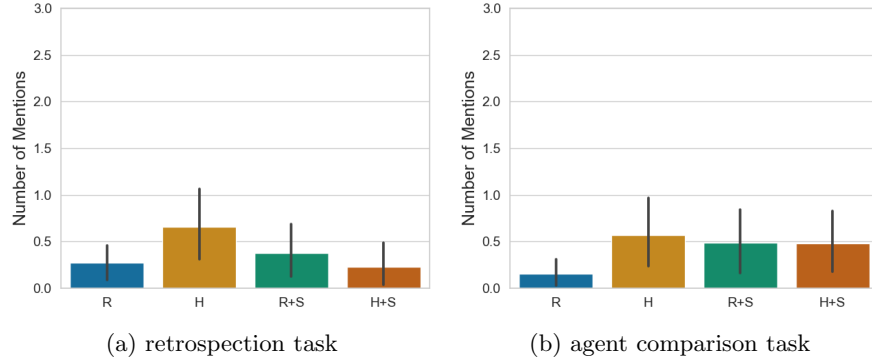
Figure B.23: Comparison of how often the participants justifications contained **unjustified** arguments.

selected object	Power pill agent	Regular agent	Fear-ghosts agent
"Pacman"	1	1	1
"normal pill"	-1	-0.5	-0.5
"power pill"	1	-0.5	-0.5
"normal ghost"	-1	-0.5	1
"blue ghost"	-1	1	1
"cherry"	-1	-0.5	-0.5

Figure C.24: Caption

5. *avoid ghosts*: "avoiding G", "avoiding G strongly", "wait for G to go away", "outmanoveuring G", "hiding from G", "mislead ghosts", "avoids being eaten / caught", "avoiding to lose / staying alive", "stays away from danger"

6. *move towards ghosts*: "being close to G", "trying to eat G NON blue", "(easily) caught by G", "easily caught by G"

7. *ignore ghosts*: "do not care about G"

8. *making ghosts blue*: "making G blue"

9. *eat blue ghosts*: "being close to blue G", "eating as many G as possible", "eat blue G to get points", "chasing/going for G", "eating the blue G", "eating to jail many G"(jailing since the ghosts move back to jail after being eaten),"prioritizing PP to eat ghosts"

10. *avoid blue ghosts*: "avoiding blue G"

11. *ignore blue ghosts*: "do not care about blue G", "prioritizing PP , but not eat ghosts"

12. *eat cherry*: "prioritizing cherry", "eat cherry to get points", "going for cherry", "eating cherry"

13. *ignore cherry*: "do not care about cherry"

14. *random movement*: "moving randomly", "move all over map", "switching directions /back&forth", "not moving / being stuck", "sticking to walls /

outside", "confused", "without strategy /random", "not planning ahead", "switching directions"

15. *focus on Pacman*: "focus on PM", "focus on whats in front of/around PM", "stuck to itself"

16. *staying in corners*: "staying in corners"

These groups are used to define a simple scoring function. Depending on the agent, each group could either be positive, neutral or negative. Positive groups contain concepts that are in line with the predefined descriptions of the agents' strategies by two of the authors involved in the training. Neutral groups consist of correct observations, which are byproducts of the agent's strategy, and negative concepts go against the agent's strategy. Each positive group contained in an answer increased the participant's score by 1 and each negative group decreased the score by −1. Here, we define a group to be "contained in an answer" if at least one concept of this group was included in the answer. Neutral groups did not affect the score.

Power pill agent:

- *positive:* "eat power pill","ignore normal pill","ignore ghosts","ignore blue ghost","ignore cherry","focus on Pacman", "staying in corners"

- *neutral:* "eat normal pill","making ghosts blue"

Regular agent:

- *positive*: "ignore cherry","focus on Pacman","making ghosts blue","eat blue ghost"

- *neutral*: "eat normal pill", "eat power pill", "ignore ghosts"

Fear-ghosts agent:

- *positive*: "avoid ghost","focus on Pacman","making ghosts blue","eat blue ghost","ignore cherry"

- *neutral*:"eat normal pill", "eat power pill"

Appendix D. Questionnaire

In this section, we provide the complete questionnaire used in the study. On the first page the participants were asked to provide personal information:

Personal information

What is your age?
Choose one of the following answers

○ 17 or younger
○ 18 - 24
○ 25 - 34
○ 35 - 44
○ 45 - 54
○ 55 - 64
○ 65 or older
○ I prefer not to specify

To which gender identity do you most identify?
Choose one of the following answers

○ Male
○ Female
○ I prefer not to answer.
○ Other: [   ]

Do you have a colour vision impairment?

✔ Yes	⊘ No

Did you play Pacman before?
Choose one of the following answers

○ No
○ The last time I played Pacman was less than 1 year ago.
○ The last time I played Pacman was less than 5 years ago.
○ The last time I played Pacman was more than 5 years ago.

The following questions ask about Artificial Intelligence (AI). Colloquially, the term "artificial intelligence" is often used to describe machines (or computers) that mimic "cognitive" functions that humans associate with the human mind, such as "learning" and "problem solving". AI agents are already able to perform some complex tasks better than the median human (today). Examples for such intelligent agents are search engines, chatbots, chessbots and voice assistants.

Do you have experience with AI (Artificial Intelligence)?

✔ Yes	⊘ No

Do you have experience with AI (Artificial Intelligence)?

✔ Yes	⊘ No

✳What kind of AI experience do you have?

ⓘ Check all that apply

☐ I know AI from the media.

☐ I use AI technology in my private life.

☐ I use AI technology in my work.

☐ I took at least one AI related course.

☐ I do research on AI related topics.

☐ Other: [_____]

✳Suppose that AI agents would achieve high-level performance in more areas one day.

	1: Extremely negative	2	3	4	5: Extremely positive	I don't know
How positive or negative do you expect the overall impact of such AI agents to be on humanity in the long run?	○	○	○	○	○	○
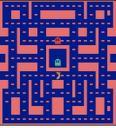
Information about Pacman:

Instructions

In this survey, you will be shown summaries of Pacman games. These videos will show you parts of games played by various AI agents trained to play Pacman. Later you will be asked to complete 6 tasks based on those videos. You will earn a 10 cent bonus for each task you solved correctly.

Please carefully read the following description of Pacman. There will be a short quiz to make sure you understood the information correctly.

A Pacman game takes place in a Labyrinth:

Pacman's ( ) goal is to eat as many "pills" (pink rectangles) as possible to earn points, while escaping the ghosts ( ).
Pacman receives 10 points for each pill, and dies when it's eaten by a ghost.

In addition to the regular pellets, there are also four special "power pills" (large pink rectangles,). Pacman receives 50 points for eating a power pill.

Eating a power pill also gives Pacman limited time during which the ghosts become blue ( ) and Pacman can eat the ghosts. Pacman receives 200, 400, 800, 1600 points for each ghost it eats successively.

After a ghost is eaten, its eyes ( ) move back to the ghost box in the middle of the screen ( ) and the ghost restarts from there in his normal form.

At random intervalls cherrys ( ) spawn and move through the labyrinth.
If PacMan eats the cherry he gets 100 points.

Pacman starts the game with three lives and looses one life each time it gets eaten by a ghost.

Please use this link to play the game and get familiar with it (While the objects look different in that version the rules are the same as described above.):
Pacman

This quiz tests whether the participants understood the information about Pacman. Participants were sent back to the previous page if they got an answer wrong.

Quiz Pacman

Please fill out this quiz to show that you understood the information.

* Eating ▬ gives Pacman:

ⓘ Choose one of the following answers

○ No points

○ 1 point

○ 10 points

* What is shown in this image:

ⓘ Choose one of the following answers

○ A ghost.

○ A blue ghost.

○ Pacman.

* What is shown in this picture?

ⓘ Choose one of the following answers

○ a piece of wall.

○ a power pill.

○ a regular pill.

* When do ghosts become blue?

ⓘ Choose one of the following answers

○ Every 2 minutes

○ When Pacman eats a power pill

○ When Pacman gets close to a ghost

* What happens when ghosts are blue?

ⓘ Choose one of the following answers

○ Pacman can eat them

○ Ghosts move faster

○ Ghosts move slower

Additional information about the provided explainable AI methods. The information about saliency maps was only displayed if the participant was in one of the saliency conditions.

Additional Information

In the following questions you will be shown videos that **summarize the behavior** of an AI agent that was trained to play Pacman.
These videos contain **several scenes** from a longer session of gameplay and aim to show you, in a limited amount of time (about 20 seconds), **how the agent plays the game.**

To aid you in your tasks, the screens will be augmented with green heatmaps that show **how relevant a pixel was for the decision of the AI agent** that controlls Pacman ("what the agent is looking at"). On some screens the green color may appear yellowish.

The **brighter the green** highlighting of a pixel is, the **more relevance** it had for the decision of the AI agent.
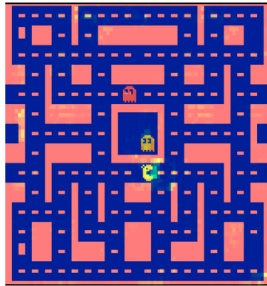
In this example, Pacman is paying alot of attention on the area infront of itself

This quiz tests whether the participants understood the information about the provided explainable AI methods. Participants were sent back to the previous page if they got an answer wrong.

Second Quiz

Please fill out this quiz to show that you understood the information.

✳A green area is:

❶ Choose one of the following answers

○ relevant for the Pacman agent.

○ relevant for your decision.

○ relevant for the ghosts.

✳If the green highlighting of an area is very bright then:

❶ Choose one of the following answers

○ the area is worth more points.

○ the area is more relevant.

○ the area is less relevant.

✳The videos in the following questions show:

❶ Choose one of the following answers

○ scenes where the Pacman agent performed very good.

○ a summary of how the Pacman agent behaves.

○ scenes where the Pacman agent performed very bad.

This is the retrospection task that was repeated for each of the three agents in a randomized order:

Analyze the agent

In these three tasks you will be shown **summaries of different AI agents** trained to play Pacman. For each AI agent, you will be asked to **analyze the strategy of that particular AI agent**. To this end, you will be asked to describe the AI agents' strategy and to select the objects that were the most important for the strategy of that particular AI agent. You will receive a bonus of 10 cents for each agent that you analyze correctly.

The video shows a summary of typical behavior of a Pacman agent. Please watch the whole video.
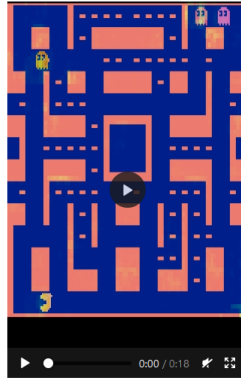
*Please briefly describe the strategy of the AI agent shown in the video above:

*Based on this video, select the objects that you think were most important for the strategy of this particular AI agent.

Select a **maximum of 3** (it does not necessarily have to be 3) Objects.

ⓘ Check all that apply

☐ Pacman

☐ Normal pills

☐ Power pills

☐ Ghosts

☐ Blue Ghosts

☐ Cherrys

61

∗How confident are you in your selection?

	Not at all confident						very confident
How confident are you that you chose the right objects?	○	○	○	○	○	○	○

∗Please briefly explain how you came to your selection:

After all three agents, the participants were asked for their satisfaction:

Explanation Satisfaction

*

	1 : I disagree strongly.	2	3	4	5 : I agree strongly.
From watching the videos of the AI agents, I got an idea of the agents' strategies.	○	○	○	○	○
The videos showing the AI agents play contain sufficient detail about the agents' behavior.	○	○	○	○	○
The videos showing the AI agents play contain irrelevant details.	○	○	○	○	○
The gameplay scenarios shown in the videos were useful for for anlayzing the agents' behavior	○	○	○	○	○
The green highlighting in the videos was useful for for anlayzing the agents' behavior	○	○	○	○	○

Do you have additional comments?

This is the agent comparison task that was repeated for each combination of the three agents in a randomized order:

Compare the agents

Below are videos showing parts of games played by two different Pacman agents:
Agent A (left) and Agent B (right).

Based on these videos, choose which Pacman agent you'd like to play on your behalf. You will receive a bonus of 10 cents for each time you select the Pacman agent that achieves higher points on average.
Note: Press play to start a video. You can pause and rewatch the videos by pressing the button again. Please watch the whole videos.

Agent A   Agent B

0:00 / 0:18   0:00 / 0:18
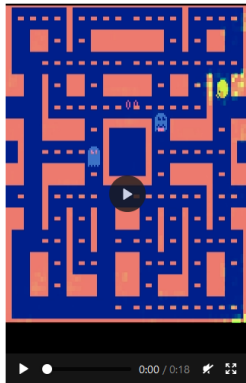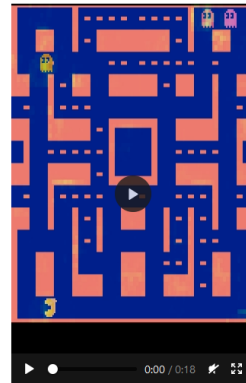
*Which of the Pacman agents do you choose to play on your behalf?

ⓘ Choose one of the following answers

○ Player A

○ Player B

*How confident are you that you chose the better agent?

	Not at all confident					very confi-dent
How confident are you that you chose the better player?	○	○	○	○	○	○

*Please briefly explain your selection:

After all three comparisons, the participants were asked for their satisfaction again:

Explanation Satisfaction

	1 : I disagree strongly.	2	3	4	5 : I agree strongly.
From watching the videos of the AI agents, I got an idea of the agents' strategies.	○	○	○	○	○
The videos showing the AI agents play contain sufficient detail about the agents' behavior.	○	○	○	○	○
The videos showing the AI agents play contain irrelevant details.	○	○	○	○	○
The gameplay scenarios shown in the videos were useful for choosing the agent that performs better.	○	○	○	○	○
The green highlighting in the videos was useful for choosing the agent that performs better.	○	○	○	○	○

Do you have additional comments?