Machine Learning (2024) 113:355–441
https://doi.org/10.1007/s10994-023-06479-7

Explainable reinforcement learning (XRL): a systematic literature review and taxonomy

Yanzhe Bekkemoen[1]

Received: 5 December 2022 / Revised: 14 September 2023 / Accepted: 20 October 2023 /
Published online: 29 November 2023

Abstract
In recent years, reinforcement learning (RL) systems have shown impressive performance and remarkable achievements. Many achievements can be attributed to combining RL with deep learning. However, those systems lack explainability, which refers to our understanding of the system's decision-making process. In response to this challenge, the new explainable RL (XRL) field has emerged and grown rapidly to help us understand RL systems. This systematic literature review aims to give a unified view of the field by reviewing ten existing XRL literature reviews and 189 XRL studies from the past five years. Furthermore, we seek to organize these studies into a new taxonomy, discuss each area in detail, and draw connections between methods and stakeholder questions (e.g., "how can I get the agent to do _?"). Finally, we look at the research trends in XRL, recommend XRL methods, and present some exciting research directions for future research. We hope stakeholders, such as RL researchers and practitioners, will utilize this literature review as a comprehensive resource to overview existing state-of-the-art XRL methods. Additionally, we strive to help find research gaps and quickly identify methods that answer stakeholder questions.

Keywords Reinforcement learning · Explainable artificial intelligence · Interpretability · Explainability · Explanation

1 Introduction

We have recently seen astounding achievements by reinforcement learning (RL) agents. In games like Go, Chess, Shogi, and Atari, RL agents have outperformed human players (Silver et al., 2016, 2017; Schrittwieser et al., 2020). While in real-time strategy games like StarCraft II, the RL agent AlphaStar ranks in the top 0.2% of human players as of August

Editor: Javier Garcia.

✉ Yanzhe Bekkemoen
yanzhe.bekkemoen@ntnu.no

[1] Department of Computer Science, Norwegian University of Science and Technology, 7491 Trondheim, Norway

2019 (Vinyals et al., 2019a, 2019b). In poker, RL agents have beaten human professionals (Brown & Sandholm, 2017). Many of those advancements were achieved by leveraging neural networks (NNs) by the RL research community (Mnih et al., 2013, 2015).

Although the research community has achieved many incredible feats, there are still unsolved challenges. One of these challenges is the incomprehensibility of the RL agents. In high-stake domains like healthcare, autonomous driving, criminal justice, and finance, using uninterpretable artificial intelligence (AI) systems is unacceptable. For example, Lapuschkin et al. (2019) demonstrate that a classifier trained on the PASCAL visual object classes dataset (Everingham et al., 2010) could use a watermark on an image to decide the image's label. In another example, the correctional offender management profiling for alternative sanctions system used in the United States to assess potential recidivism risk has been accused by ProPublica of being racially biased (Angwin et al., 2016; Larson et al., 2016). When it comes to laws, the launch of the European Union's General Data Protection Regulation introduces the right to explanations of all automated decisions for individuals (Sovrano et al., 2020). All these examples demonstrate problems with the use of AI systems, and as a result, using RL and machine learning (ML) in general is getting more complicated. Many of these problems become even more problematic when using NNs. For example, NNs' predictions can change based on modifications in images imperceptible by human eyes (Szegedy et al., 2014). Furthermore, Nguyen et al. (2015) demonstrate that NNs can classify humanly unrecognizable observations wrongly with high confidence.

These aforementioned examples of difficulties have caused a renewed interest in explainable artificial intelligence (XAI) (Guidotti et al., 2019; Arrieta et al., 2020; Burkart & Huber, 2021; Ras et al., 2022; Minh et al., 2022). Likewise, this has resulted in a new emerging subfield, explainable reinforcement learning (XRL). XRL is a research field focusing specifically on explaining RL agents, whereas XAI focuses on many forms of learning like unsupervised and supervised learning. In supervised learning, we assume observations are independent and identically distributed. Further, the goal is empirical risk minimization with immediate response. In contrast, the agent in RL learns to maximize the return with rewards as the responses, which are not necessarily provided immediately. Hence, the agent needs to consider the short-term and long-term consequences in addition to the immediate response when learning to make decisions. Accordingly, we must develop new methods to explain these RL specific characteristics that explanation methods of supervised learning cannot explain.

Researchers have published numerous literature reviews on XRL responding to new challenges explaining RL agents. However, because of the fast development, many recent studies on XRL are not covered in these reviews. Moreover, a unified view of the field that structures and organizes these XRL reviews is missing. This systematic literature review provides a unified view of the XRL field. Furthermore, we aim to help stakeholders (e.g., RL researchers and practitioners) become acquainted with the state-of-the-art XRL methods and find research gaps. Lastly, we seek to help stakeholders find a suitable method to answer their questions. For example, which method should the stakeholder apply if the stakeholder wants to know, "how can I get the agent to do _?" We achieve these goals by first finding XRL studies through a systematic search and selection process. Afterward, we summarize existing literature reviews, structure the XRL studies into a new taxonomy, and outline what kind of stakeholder questions they can answer. Next, we provide a detailed view of the state-of-the-art XRL methods by closely examining the taxonomy and its methods. Finally, we look at the XRL research trends, recommend XRL methods for different stakeholder questions, and propose future directions for XRL based on the reviewed studies.

This systematic literature review is structured as follows. First, we describe the research method used to conduct this systematic literature review in Sect. 2. Next, Sect. 3 describes

the background on RL and XAI needed to understand this systematic literature review. In addition, in the same section, we outline some related research fields. Then, Sect. 4 summarizes existing XRL literature reviews and shows how our systematic literature review differs. Section 5 overviews XRL by providing a taxonomy that categorizes the different XRL methods. In the same section, we show different explanation types and RL explainability characteristics, which describe stakeholder questions. Afterward, Sects. 6, 7 and 8 review XRL methods by following the taxonomy. Based on the reviewed methods, we look at XRL trends in Sect. 9.1, recommend XRL methods in Sect. 9.2, and discuss future directions for the XRL research field in Sect. 9.3. We conclude this systematic literature review in Sect. 10. Finally, Section Appendix A gives a concise summary of reviewed methods with various details. To summarize, our contributions are:

- A summary of existing XRL reviews and their contributions.
- A new taxonomy reflecting the large body of XRL studies, divided into (1) interpretable agent, (2) intrinsic explainability, and (3) post hoc explainability. Furthermore, the taxonomy organizes studies based on how explanations are conveyed: (1) via generation, (2) via representation, or (3) via inspection.
- An overview of which explanations types and RL explainability characteristics the different taxonomy categories provide.
- A comprehensive look at 189 XRL studies found using a systematic approach with a concise overview in Section Appendix A. For each study, the appendix details the scope, the focus, experimentation environment(s) or task(s) (or both), if it performs a user study, and if the code has been open sourced.
- An overview of the trends in XRL, recommendation for XRL methods, and future directions to address current challenges based on the reviewed studies.

2 Research method

This section outlines our systematic approach to identifying, evaluating, and reporting studies on XRL methods. To avoid bias in literature selection and make it reproducible and complete, we chose to do the review systematically. This systematic literature review was carried out by partially following the guidelines by Kitchenham et al. (2020). We describe the research questions in Sect. 2.1. Section 2.2 describes the study selection process with the overall process depicted in Fig. 1.

2.1 Research questions

This systematic literature review aims to answer the following research questions:

- What are the existing XRL literature reviews and their contributions?
- What are the state-of-the-art XRL methods, and how can we organize them?
- What kind of stakeholder questions can these methods answer (e.g., "how does the agent work?" and "why did the agent do _?")?
- How are the methods evaluated (i.e., were user studies performed? and in what domain or task (or both) do they evaluate their method?)?
- What are the research trends in XRL?

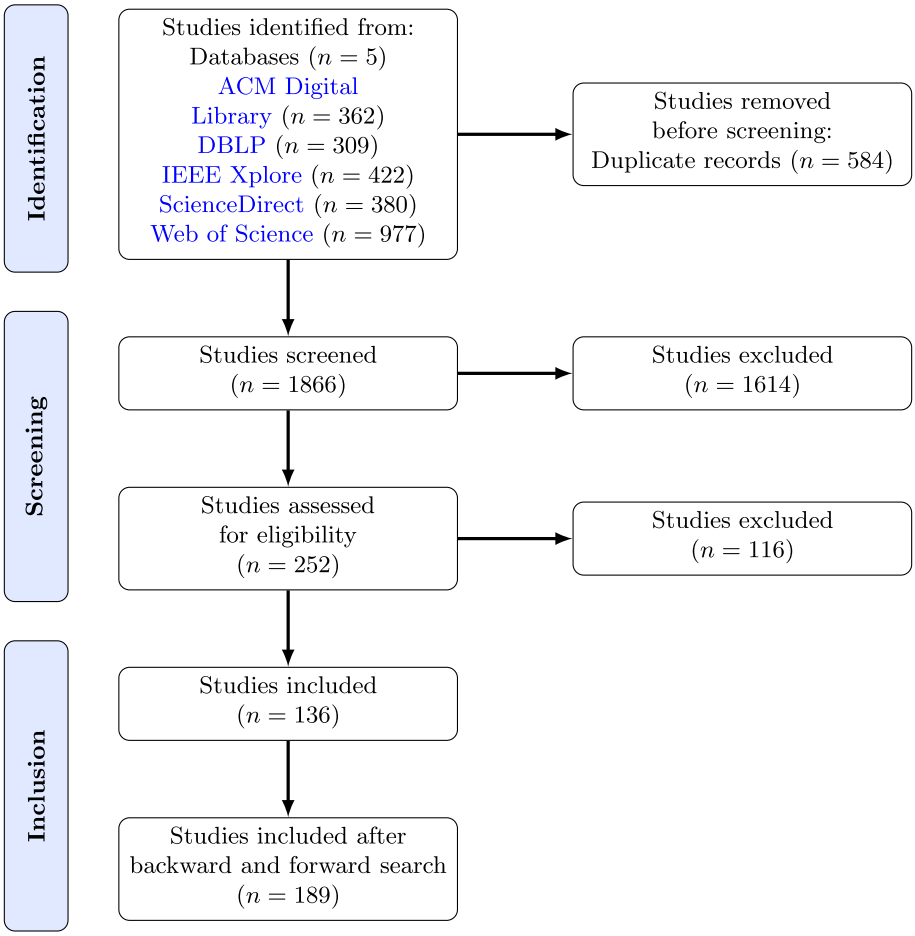

Fig. 1 The study selection process was performed via three steps, identification, screening, and inclusion. In the identification step, we searched five different databases and removed duplicates automatically. Next, we screened the studies found using a two-stage process and removed studies using pre-defined selection criteria. Finally, we added relevant studies and performed forward and backward searches on them to add additional studies. The figure structure is adapted from the Preferred Reporting Items for Systematic Reviews and Meta-Analyses statement (Page et al., 2021)

2.2 Study selection process

We started by finding related work (see Sect. 4) and used those as a basis for constructing our search string. Next, we created the following search string: ("reinforcement learning" AND ("explanation" OR "explainability" OR "explainable" OR "XAI" OR "explainable AI" OR "interpretable" OR "transparency" OR "transparent" OR "understandable" OR "interpret" OR "black box")) OR "XRL".

Using the search string, we searched the title and abstract (when available) in the following electronic databases: (1) ACM Digital Library, (2) DBLP, (3) IEEE Xplore, (4) ScienceDirect, and (5) Web of Science. We removed the duplicates automatically using

Paperpile (LLC, Cambridge, MA) since the databases overlap. Our search is limited to studies published after 2017 and before July 2022, with few exceptions. We chose 2017 since not many XRL studies existed before this year, and other reviews already cover them. Moreover, 2017 was the year Defense Advanced Research Projects Agency (DARPA) launched its XAI program (Gunning & Aha, 2019).

The author conducted the selection process by following the method *Selection process for lone researchers* (Kitchenham et al., 2020, Page 318). Specifically, we applied the test-retest approach, where studies are reassessed later to check if they still fit the research questions and selection criteria. When uncertain, studies were discussed with a third party. To select studies, we used the following selection criteria:

(1) The study focuses on explainability in RL. Specifically, we omit studies where explainability is the by-product and not driven by it.
(2) The study does not focus on the multi-agent RL.
(3) The study is peer-reviewed.
(4) The study is in English.

Studies were selected in two stages using these selection criteria. In the first stage, we screened the title and abstract for relevance. After the first stage, we screened the full text to decide on inclusion based on the same selection criteria. We included 136 studies after two passes of screening. By forward and backward searching those 136 included studies, we found 53 additional relevant studies. In total, there are 189 relevant studies on the XRL topic included. Figure 2 depicts the studies included distributed by year. The number of studies included suggests increasing interest in XRL. The first study selection process on October 13, 2021, found 121 relevant studies. However, to keep this review updated on state-of-the-art XRL methods and reviews, the entire study selection process was reperformed on July 24, 2022, resulting in 183 studies. As the term "XRL" was not included in the original searches, a new search on the term "XRL" was performed on July 6, 2023, resulting in 189 total studies.

3 Background

This section provides the necessary background to understand the literature review's content. We give a general overview of RL in Sect. 3.1 and XAI in Sect. 3.2. Finally, we overview some research fields related to XRL in Sect. 3.3.

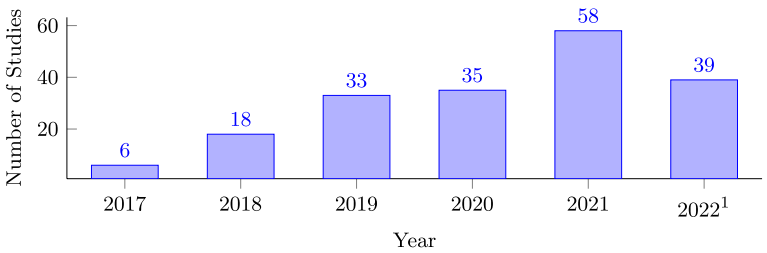

Fig. 2 The number of studies reviewed, distributed by the year published. [1]The number of studies for 2022 does not include the entire year

3.1 Reinforcement learning

RL (Sutton & Barto, 2018) is a subfield of ML and is also known by its less popular names: approximate dynamic programming (DP) and neuro-DP (Bertsekas & Tsitsiklis, 1996). DP in the name signifies the importance of DP (Bellman, 1952, 1966) as the foundation of RL. RL is a framework for constructing intelligent agents that learn to make decisions through interactions with the environment rather than via instructions. In RL, the feedback on decisions is provided through rewards, which in psychology is known as reinforcement. The feedback differs from supervised learning because the feedback is not necessarily being given on every decision made by an agent. As a result, decisions in RL have short-term and long-term consequences in addition to immediate consequences. Moreover, in supervised learning, the observations are independent and identically distributed, which is not true for RL. The RL framework is built on the reward hypothesis that states we can formulate the learning goal as maximizing the expected cumulative reward, thus, focusing on a sum instead of a single quantity. The expected cumulative reward is known as the expected return.

This section provides the RL background needed to understand the rest of this review. First, we formally define the Markov decision process (MDP) in Sect. 3.1.1. Then, in Sect. 3.1.2, the RL problem is defined, which is the goal of RL.

3.1.1 Markov decision process

An MDP formalizes the sequential decision-making problem mathematically. In an MDP, the actions affect each other and the feedback is given via rewards that are potentially not supplied for every action taken. As a result, the agent in the decision-making problem must consider both immediate and future rewards. Formally, an MDP is defined as a tuple $\langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$ where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions and $\gamma \in [0, 1]$ is the discount factor. The transition function $p(s'|s, a)$ is a conditional probability distribution that defines the dynamics of the MDP, where $s', s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$. In the state s, the actions available are indicated by the set $\mathcal{A}(s)$. In the MDP, we assume states have complete information. Furthermore, we assume that the probability of transitioning from s to s' depends solely on s and not the entire history, thus, satisfying the Markov property. The reward function $r(s, a)$ provides the reward of taking an action $a \in \mathcal{A}(s)$ in the state $s \in \mathcal{S}$, and can optionally rely on the next state s'. The reward $R \in \mathcal{R}$ is bounded by $\pm R_{\max}$. All possible rewards are denoted by the set $\mathcal{R}$, which is a finite subset of $\mathbb{R}$. In sum, all of these stated elements together form an MDP.

A policy π is a mapping from a state $s \in \mathcal{S}$ to an action $a \in \mathcal{A}(s)$. In the stochastic case, the policy yields a probability distribution over all actions. Like the transition function in the MDP, the policy is modeled such that it is only conditioned on the current state and not the whole history of states and actions. A trajectory is a sequence of states and actions defined by $\tau = (s_0, a_0, s_1, s_2, \dots)$ where $s_0 \sim p_0$ and where p_0 denotes the start state distribution. We choose or sample the action from a policy π and sample the next state from the transition function. Thus, we can create trajectories with access to these two functions.

3.1.2 Problem

The RL problem is about discovering a policy π that maximizes the expected return (Achiam, 2018). Assume that we have an MDP and a policy π as defined earlier. Then, the probability of a T-step trajectory τ conditioned on the policy π is

$$p(\tau|\pi) = p_0(s_0)\prod_{t=0}^{T-1} p(s_{t+1}|s_t, a_t)\pi(a_t|s_t) \quad \text{where} \quad s_0 \sim p_0. \tag{1}$$

Under the policy π, the probability of taking action a_t given the state s_t is denoted $\pi(a_t|s_t)$.

We define the infinite-horizon discounted return over a trajectory τ by

$$r(\tau) = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \leq \sum_{t=0}^{\infty} \gamma^t R_{\max} = \frac{R_{\max}}{1-\gamma}, \tag{2}$$

where $\gamma < 1$. The infinite-horizon discounted return is used for several reasons (Russell & Norvig, 2020): (1) based on empirical results, humans and animals prefer rewards as soon as possible, (2) in a financial setting, it is better to invest now than later, (3) the uncertainty of rewards increases as time passes, and (4) it is mathematically convenient as shown in Eq. (2).

Based on the trajectory probability and the return, we express the expected return given an MDP and a policy π by

$$J(\pi) = \int_{\tau} p(\tau|\pi) r(\tau) d\tau = \mathbb{E}_{\tau\sim\pi}[r(\tau)]. \tag{3}$$

Finally, we define the RL problem by

$$\pi_* = \arg\max_{\pi} J(\pi). \tag{4}$$

That is, finding the optimal policy π_* that maximizes the expected return.

3.2 Explainable artificial intelligence

XAI defines an AI that can be understood by a human, including how it works, its strengths and weaknesses, and the behavior it will exhibit in unseen situations. A black box is the opposite, where the system's internal mechanisms are either incomprehensible or not accessible to a human. The term XAI was popularized by DARPA when they launched the XAI program in May 2017. The word explainable was chosen to signify that an XAI system actively explains to increase a human's understanding of it. Furthermore, they use the word explainable to emphasize the interest in the human psychology of explanation. XAI has been a research interest since the early 1970 s, with expert systems like MYCIN (Buchanan & Shortliffe, 1984) and GUIDON (Clancey, 1987). However, increasingly widespread use and interest in AI have renewed the attention on XAI, especially since the success of deep NN in the ImageNet 2012 challenge (Krizhevsky et al., 2012). The recent interest in XAI has quickly created a tremendous amount of new research.

We organized the section as follows. First, Sect. 3.2.1 loosely discusses XAI terminologies. Then, Sect. 3.2.2 explains why we need explainability. Afterward, Sect. 3.2.3

describes the different stakeholders that consume explanations. Next, Sect. 3.2.4 introduces some explanation properties. Lastly, Sect. 3.2.5 overviews explanation evaluations.

3.2.1 Terminologies

Interpretability and explainability are often used interchangeably in the literature (Gilpin et al., 2018; Arrieta et al., 2020). This section loosely discusses them since there is no consensus on a single definition. According to the dictionary (Merriam-Webster, 2022), the word interpret means "to explain or tell the meaning of" or "present in understandable terms". In the context of XAI, Doshi-Velez and Kim (2017) define interpretability as "the ability to explain or to present in understandable terms to a human". The human is what we define as the stakeholder, which we elaborate on in Sect. 3.2.3. Murdoch et al. (2019) define interpretable ML as "the extraction of relevant knowledge from an ML model concerning relationships either contained in data or learned by the model". Whereas Lipton (2018) suggests that interpretability refers to several ideas and is not limited to one concept. According to Gilpin et al. (2018), explainability differs from interpretability. An interpretable system is not necessarily explainable, while the opposite is true. They define an explainable system as a system that: (1) can justify its decisions, (2) is interactable, and (3) is auditable. In this review, we follow Gilpin et al. (2018) and distinguish that interpretability is passive while explainability is active. When we want to refer to both terms, we write explainability. Thus, we think of both interpretable RL and explainable RL when we talk about XRL.

The technical definition of an explanation remains elusive. According to Gilpin et al. (2022), explanations are objects created due to their functional roles, stakeholders (referred to as the audience in the study), and capabilities. The functional role refers to why stakeholders want or need explanations. The stakeholder is the receiver of the explanation, also known as the explainee. Capabilities are about the AI system's logical thinking process and its degree of access to the process.

3.2.2 Explainability needs

Explainability needs aim to answer on a high level why we need XAI in the first place, unlike stakeholder questions (i.e., the specific questions a stakeholder wants to get answered, for example, "how can I get the agent to do _?"). We need explainability because the deployment cost is not included in the AI system's learning objective (Doshi-Velez & Kim, 2017; Lipton, 2018). When the AI system is learning, it tries to optimize the test predictive performance in supervised learning or the return in RL. However, the test predictive performance and the return might not capture the real-world deployment costs because it is difficult or impossible to formally write it down mathematically. For example, when the RL agent moves from the training environment to deployment, we want robustness to the distributional shift. Still, like in the supervised setting, it cannot be easily encoded mathematically. The problem at hand might also require a flexible approximator that is not interpretable. Furthermore, ensuring the objectives are sound by auditing all possible situations is infeasible. The literature has defined several reasons for explainability needs (Doshi-Velez & Kim, 2017; Lipton, 2018; Arrieta et al., 2020; Burkart & Huber, 2021). We list some examples here:

Trust The concept of trust is difficult to define and has been defined differently by different researchers across disciplines (Simpson, 2012; Robbins, 2016). One way to understand

trust is whether a stakeholder is willing to delegate the decision-making to the AI system. Thus, if a stakeholder is inclined to let the AI system decide on its behalf, then it trusts the system. Also, trust can be a stakeholder's confidence that the system will behave as intended.

New insight This need is about the ability to extract knowledge from the AI system to gain a new understanding of the problem at hand. We create the system not necessarily to make decisions but to gain novel insight into the domain.

Making adjustments The idea of changing an AI system encompasses correcting and improving it. Different quantities, such as accuracy and return indicate the system's performance but lack in their ability to find, fix, and improve the system. Hence, knowing how the system works and its strengths and weaknesses is required to find bugs, fix them, determine when the system might fail, and improve it.

Fairness and being ethical These two needs are related to ensuring that the AI system does not make decisions that, for example, might discriminate based on skin color or gender and complies with ethical standards (Goodman & Flaxman, 2017).

Apart from the aforementioned reasons, there are other reasons like effective human and AI collaboration (Hayes & Shah, 2017), privacy (Arrieta et al., 2020), and accountability (Doshi-Velez et al., 2017) that motivate the need for explainability.

3.2.3 Stakeholders

When we discuss explainability, we should reason about it in relation to an audience and their need for explanation (Kirsch, 2017). This signifies that XAI is not an isolated field concerning only ML researchers, but an interdisciplinary field that involves, for instance, human-computer interaction. Suppose that some explanations might be helpful and understandable for AI researchers. However, they might not be helpful or even understandable for the AI system's end-users. The reason is that these two groups have different goals for explainability and expertise. In short, to talk meaningfully about explainability, it should be in the context of a specific stakeholder, such as developers, domain experts, or end-users.

There have been several works in the literature discussing and proposing stakeholder frameworks for ML explainability (Weller, 2017; Preece et al., 2018; Tomsett et al., 2018; Ribera & Lapedriza, 2019; Hohman et al., 2019; Mohseni et al., 2021; Langer et al., 2021). The stakeholder frameworks differ between studies, but there are generally two ways to group stakeholders based on their role or expertise (Suresh et al., 2021). In the role-based frameworks, stakeholders are grouped by their roles and the explainability needs align with their role. In the second group, the stakeholders are grouped by their expertise, and their explainability needs result from their expertise.

3.2.4 Explanation properties

Researchers have proposed different explanation properties with the growing research on XAI (Lipton, 2018; Murdoch et al., 2019, Murphy et al., 2023, Chapter 33.3). Depending on the situation, we need different explanation properties. In this section, we overview some of these properties:

Fidelity and faithfulness Fidelity describes the extent an explanation can accurately explain the model (Robnik-Sikonja & Bohanec, 2018; Guidotti et al., 2019; Jacovi & Goldberg, 2020). For example, how a distilled model explains the original model can

be measured through accuracy in agent distillation methods. Accuracy is defined as the number of correct predictions divided by the total number of predictions. Faithfulness also expresses the accuracy of an explanation. Jacovi and Goldberg (2020) state that the term faithfulness often differs between studies and is used inconsistently. Murphy et al. (2023) (Page 1076) define fidelity and faithfulness together. They discuss a measure of faithfulness in terms of how often a distilled model provides the same outputs as the original model. This is the same as how the other aforementioned studies use the term fidelity. Similarly, Robnik-Sikonja and Bohanec (2018) use these two terms in the same context. This shows that there is no clear distinction between these two terms in the literature.

Completeness This indicates whether an explanation conveys all factors relevant to the decision-making process.

Sparsity It refers to the notion of an explanation being small and compact, which is important since it is easier to understand explanations with fewer components to inspect.

Actionability It expresses changing the content of an explanation such that it only contains components that a stakeholder can adjust.

A more exhaustive overview of different properties can be found in the aforementioned book and studies.

3.2.5 Explanation evaluation

Evaluating explanations is difficult since there is not a single mathematical definition of explanations. Moreover, we must evaluate explanations by considering the explainability needs, the task, stakeholders, and constraints, such as time and attention. For instance, given two explanations and two tasks, the stakeholder might find the first explanation more helpful for the first task but not for the second task, which is explained better by the second explanation. The dependence on the overall setup makes the explanation evaluation difficult, emphasizing the importance of evaluating explanations using the intended setup. Researchers have proposed explanation evaluation taxonomies in the literature (Doshi-Velez & Kim, 2017; Mohseni et al., 2021). Doshi-Velez and Kim (2017) proposed to divide explanation evaluation into three levels:

Functionally grounded evaluation This evaluation type involves evaluating explanations computationally, such as measuring explanations' fidelity or sparsity. This type of evaluation involves no stakeholders and is cheap but does not evaluate explanations on the intended setup.

Human grounded evaluation It involves evaluating explanations using stakeholders but with simplified tasks. For example, participants recruited via Amazon Mechanical Turk with tasks in games. On the one hand, the evaluation does not include the intended stakeholders and tasks, giving only a partial picture. On the other hand, this evaluation form allows for a larger user pool and more feedback with fewer resources.

Application grounded evaluation It is the most accurate evaluation but also the most expensive since the evaluation involves the intended stakeholders and tasks. For instance, we can use medical doctors in medical diagnosis tasks to test explanations.

3.3 Related research fields

This systematic literature review investigates RL studies focusing explicitly on explainability. There exist other interesting research fields within RL that strive for similar goals

as explainability but achieve it in a different way, including human-in-the-loop RL and safe RL. We do not discuss works from these research areas to retain a focused scope on explainability in accordance with our selection criteria. Instead, we briefly describe them and point to more in-depth resources on these topics for further reading.

Human-in-the-loop RL includes studies where a human oracle provides the agent with feedback in real-time. With human-in-the-loop RL, it is possible to align humans' mental models of RL agents' behavior. In turn, this increases the predictability and trust in RL agents, which are similar to the goals of XRL. Studies within human-in-the-loop RL ranges from reward function specification (III & Sadigh, 2022) to exploration (Arakawa et al., 2018). One of the challenges in this field is how feedback from the human oracle should be modeled. In our review, we only included studies where human-in-the-loop is explicitly used for explainability (Fukuchi et al., 2017a, 2017b, 2022; Bewley & Lécué 2022; Cruz & Igarashi, 2021; Tabrez et al., 2019). For further reading, there are many human-in-the-loop RL surveys (Wirth et al., 2017; Li et al., 2019a; Cruz & Igarashi, 2020).

Safe RL aims to learn policies that perform well but at the same time ensure specified safety constraints are respected in training and deployment despite uncertainty. Safe RL is about making sure a policy avoids visiting states that are considered unsafe (Hans et al., 2008). Also, it is about making sure the policy can reach any state from the states it visits so that a negative outcome can be amended (Moldovan & Abbeel, 2012). Surveys that comprehensively cover this topic for further reading include García and Fernández (2015) and Gu et al. (2022).

4 Related work

The success of RL and the recent increasing interest in XAI have resulted in many XRL literature reviews. In this section, we give an overview of previous XRL literature reviews. Furthermore, Table 1 provides a detailed overview of these literature reviews' contributions and the number of studies they cover. Numerous relevant literature reviews exist for XAI (Arrieta et al., 2020; Burkart & Huber, 2021; Ras et al., 2022; Minh et al., 2022), but we only cover literature reviews focusing on RL.

As far as we know, Puiutta and Veith (2020) published the first XRL literature review. They provide an overview and categorization of XRL methods based on an existing XAI taxonomy (Arrieta et al., 2020). Their discussion points out that the connection between stakeholders and explanations is often not considered. They suggest that more studies should focus on interdisciplinary work to alleviate this issue. Similarly, Heuillet et al. (2021) adapt existing XAI taxonomy to categorize XRL techniques. Wells and Bednarz (2021) take a systematic approach to the literature review and follow the methodology by Kitchenham et al. (2009). Their systematic literature review focuses on answering two questions regarding XRL methods. First, what XRL methods exist in the literature? And second, what are their limitations? In contrast to the previous studies, Alharin et al. (2020) propose a novel taxonomy for categorizing XRL methods. Additionally, they described the taxonomy regarding different method properties.

Glanois et al. (2022) focus on interpretable RL. Several reasons motivate their review: (1) the need for interpretability, (2) the increasing number of studies on interpretable RL, and (3) the limited number of studies reviewed by previously mentioned literature reviews. They propose a new interpretable RL taxonomy and more thorough coverage of interpretable RL methods than Puiutta and Veith (2020), Alharin et al.

(2020), and Heuillet et al. (2021). They focus mainly on studies published in the past ten years.

The reviews by Puiutta and Veith (2020), Heuillet et al. (2021), and Wells and Bednarz (2021) provide a deep dive into XRL methods, but the scope is limited. As a result, Milani et al. (2022) propose a more extensive and newer literature review on XRL techniques. Additionally, they propose a new taxonomy for XRL methods. Building on the knowledge of the previous literature reviews, Krajna et al. (2022) introduce a new taxonomy for XRL techniques and explore XRL for the multi-agent setting. Vouros (2022) comprehensively reviews XRL methods, concentrating on the deep reinforcement learning (DRL) counterpart. He describes each reviewed XRL method thoroughly, detailing the motivation, assumptions, technical details, evaluation, and more. Finally, Hickling et al. (2022) introduce another XRL review and describe XRL methods and two existing XRL literature reviews (Wells & Bednarz, 2021; Vouros, 2022).

Differently from the other literature reviews, Dazeley et al. (2021a) go beyond reviewing existing XRL methods and concentrate on Broad-XAI. They define Broad-XAI as combining and integrating explanation strategies into an individual explanation that satisfies a stakeholder's need. Contrasting all previous studies, Zelvelder et al. (2021) review RL application domains and to what degree XRL is investigated in those application domains. Sakai and Nagai (2022) introduce a literature review on explainable autonomous robots, a related field to XRL. Lastly, Sado et al. (2023) describe methods that focus on explaining autonomous robots and agents, which overlap with XRL.

Our work differs in several ways compared to previous surveys and reviews:

- We propose a novel taxonomy from the perspective of the reviewed studies. Our taxonomy accommodates the large spectrum of XRL methods and has the finesse needed to compare and discuss categories of methods and methods within a category. We believe the categorization of methods in previous works makes doing these comparisons and discussions more challenging. First, in previous works, methods within a category can produce explanations using different mechanisms. For instance, both agent distillation and policy summarization methods produce global explanations but use different strategies. Second, they can express different types of information. For example, feature importance is mostly limited to where the agent looks, while textual justifications allow for explanations with richer semantics. Third, they can produce explanations that answer different stakeholder questions. For example, agent distillation methods can answer specific why questions, but policy summarization methods cannot. Fourth, they can convey explanations in different ways. Our taxonomy takes these points into account. Considering these differences from ours, we believe our taxonomy with finer divisions makes discussions and comparisons easier. We illustrate these issues below.

  Puiutta and Veith (2020), Heuillet et al. (2021) and Hickling et al. (2022) use taxonomy from XAI and do not propose a XRL specific taxonomy. Wells and Bednarz (2021) propose a new taxonomy, but some of their categories like *Visualization* and *Policy Summarization* are expansive. For example, the category *Policy Summarization* includes methods commonly known as policy summarization (Amir & Amir, 2018; Lage et al., 2019b) in the literature. Yet, it also includes agent distillation methods (Verma et al., 2018) and methods aimed at human-robot collaboration (Hayes & Shah, 2017). Alharin et al. (2020) introduce a new taxonomy but makes it difficult for the reader to compare categories. Categories on the same level range from *Computer Vision* and *Natural Language* to *Decision Trees* and *Summarization*. The former

Table 1 Summary of XRL literature reviews

Reference	#S	Contributions
Puiutta and Veith (2020)	15	Categorize XRL methods based on: (1) the extent of the explanation (global versus local) and (2) how an explanation is obtained (intrinsic versus post hoc). Present and discuss selected XRL methods in greater detail. Discuss the balance in the taxonomy categories, the connection between stakeholders and explanations, and explanation evaluation.
Alharin et al. (2020)	53	Categorize XRL methods using a new taxonomy consisting of (1) decision tree, (2) summarization, (3) computer vision, (4) natural language, 5) custom models, and 6) model reconciliation. Those categories are further divided into subcategories. Discuss the taxonomy regarding (1) post hoc methods or intrinsic method (or both), (2) the extent of the explanation, (3) the form of the explanation, (4) when the method is used, (5) model agnostic versus model specific, and (6) performance and fidelity. Discuss the difficulty of categorizing and evaluating XRL methods.
Heuillet et al. (2021)	19	Categorize XRL methods in a taxonomy divided into (1) transparent algorithms and (2) post hoc explainability. Transparent algorithms are grouped by: (1) representation learning, (2) simultaneous learning, and (3) hierarchical learning. Post hoc explainability consists of (1) interaction data and (2) saliency maps. Classify XRL methods based on: (1) the task and environment, (2) MDP versus POMDP, (3) the algorithms used, such as PPO and deep Q-network (DQN) used in studies, (4) the format of the explanation (images, text or diagrams, and local or global), and (5) the stakeholder. Present and discuss XRL and XAI methods that can be used for RL agents. Discuss the lack of an all-inclusive XRL method and the relationship between stakeholders and explanations.
Wells and Bednarz (2021)	33	Review the literature systematically to answer: (1) what XRL methods have been published and (2) their limitations. Categorize XRL methods based on their domain (games, robotics, grid word, networking, autonomous vehicles, and military) and their aim (visualization, human collaboration, verification, queriable explanation, and policy summarization). Discuss four disadvantages with existing XRL methods: (1) lack of scalability, (2) limited utilization of user studies and code availability, (3) explanation presentation, and (4) the limited number of inherently interpretable methods.

Table 1 (continued)

Reference	#S	Contributions
Dazeley et al. (2021a)	N/A	Propose an abstract framework for XRL termed causal XRL framework (CXF) based on the work by Böhm and Pfister (2015) and Dazeley et al. (2021b). Furthermore, a discussion on how CXF incorporates human explanation models. Describe existing XRL approaches with respect to their framework, named Simplified-CXF. They note that many XRL methods acquire ideas from the supervised learning setting. Find possibilities for future research paths, such as employing "hierarchical, multi-goal, multi-objective, and intrinsically motivated RL techniques".
Glanois et al. (2022)	119	Discuss interpretable RL techniques and categorize them into (1) interpretable inputs, (2) interpretable transition/reward models, and (3) interpretable decision-making (policies or value functions). Give a brief overview of XRL methods. Describe open research problems for future work: (1) full interpretability in RL, (2) interpretability versus performance, (3) interpretability versus scalability, and (4) evaluation of interpretability and explainability.
Milani et al. (2022)	49	Propose a new taxonomy for XRL that divides methods into groups by considering the RL agent component being explained by methods. The taxonomy consists of (1) feature importance, using the context at the present time to explain, (2) learning process and MDP, using interaction data to explain, and (3) policy-level, explaining long-term outcomes. They divide these categories into finer details. Organize RL techniques according to the novel taxonomy and give a comprehensive discussion. Point out future directions for XRL: (1) study XRL specific properties, (2) make evaluations that are common for XRL, and (3) create XRL specialized methods.
Vouros (2022)	N/A	Categorize and describe in detail XRL techniques regarding the issue they try to work out: (1) the model inspection problem, (2) the policy explanation problem, (3) the objectives explanation problem, and (4) the outcome explanation problem. Provide a framework that details the components of an explainable DRL system. Moreover, identifies the patterns used to create explainable DRL methods. State various problems for the future, such as "develop explainable deep RL in a principled way", "bridge to theory, while answering pragmatic concerns", and "build a toolbox for explaining DRL".

Table 1 (continued)

Reference	#S	Contributions
Krajna et al. (2022)	28	Categorize and discuss methods based on whether the explanation focuses on the immediate (reactive) or long-term (proactive) consequences. The reactive techniques are grouped by: (1) policy simplification, (2) reward decomposition, or (3) feature contribution and visual methods. The proactive methods are divided into (1) structural causal model, (2) explanation in terms of consequences, (3) hierarchical policy, and (4) relational RL. Extend Puiutta and Veith (2020)'s classification of XRL methods with (1) whether the environment is deterministic or stochastic, (2) if the policy is deterministic or stochastic, (3) whether it is for single or multi-agent, and (4) the explanation format. Discuss desirable properties of XRL explanations.
Hickling et al. (2022)	56	Describe two XRL literature reviews (Wells & Bednarz, 2021; Vouros, 2022). Categorize and describe XRL methods based on their application: (1) video game simulations, (2) vehicle guidance, (3) system control, (4) robotic manipulation, (5) network solutions, and (6) other applications. Propose two limitations with current methods: (1) scalability and (2) evaluation.

#S denotes the number of XRL studies reviewed. The study count is extracted from figures, tables, or lists. N/A refers to the study count not being readily available at the time of writing

denotes large research fields, while the latter is a machine learning model and a XRL technique. Similarly, the *Feature contribution and visual methods* category in Krajna et al. (2022) would be easier to discuss using finer divisions instead of as a single category. Glanois et al. (2022) include many useful studies that can promote interpretability, but they do not provide a taxonomy for XRL. In Milani et al. (2022), the subcategory *Directly Generate Explanations* contains both textual justification methods (Ehsan et al., 2018; Wang et al., 2019b; Hayes & Shah, 2017), feature importance methods that require specific architecture (Goel et al., 2018; Mott et al., 2019), and agnostic feature importance methods that do not involve training an agent (Greydanus et al., 2018; Shi et al., 2022). Vouros (2022) introduce an explainable deep RL specific taxonomy that consists of the large categories: *Solving the (1) Model Inspection, (2) Policy Explanation, (3) Objectives Explanation, and (4) Outcome Explanation Problem*. These categories can be extended to enable more nuanced comparisons and discussions. But in its current form, categories in the taxonomy are very broad. For instance, the category *Solving the Policy Explanation* contains both policy summarization (Amir & Amir, 2018; Huang et al., 2018) and agent distillation methods (Verma et al., 2018; Hüyük et al., 2021).

- Our literature review is the only systematic one besides Wells and Bednarz (2021) that aims for exhaustive and comprehensive searching for literature explicitly related to XRL. However, they cover less than a fifth of the number of studies compared to ours.
- Beyond reviewing XRL studies, we extensively summarize existing XRL surveys and a systematic review. This includes outlining their contributions and what challenges they consider currently unsolved in XRL. We believe this enables us to provide a broader view of the XRL field.
- We divide the category that is often known as post hoc explainability (as seen in Puiutta & Veith 2020; Heuillet et al., 2021; Arrieta et al., 2020) into two categories, post hoc explainability and intrinsic explainability. We believe such a division is important when stakeholders decide on a method to use, as the methods have different use cases and requirements. While the categories overlap, they also have some significant differences, such as performance impact, agent design and training, access to the agent's internal logic and the environment, and applicability. For instance, only post hoc explainability methods can be used if the stakeholder does not want to train or fine-tune an agent. We discuss these differences later in detail when we present the taxonomy.
- We describe the explanation types and RL explainability characteristics that different XRL method categories satisfy. This is an important aspect since stakeholders are the consumers of explanations. Furthermore, our review outlines how explanations are communicated. Whether explanations are conveyed via generation, representation, or inspection. These two things make it easier for stakeholders to find methods more suited for their use case that previous reviews lack.
- We outline trends in XRL and recommend methods based on stakeholder questions (e.g., "how does the agent work?" and "why did the agent do _?"). This is in addition to future directions that other surveys and reviews focus on.
- All reviewed methods are concisely summarized in the appendix. This includes, what is the motivation of the method, what it explains, and its evaluation.

5 Explainable reinforcement learning

Learning via interaction, potentially delayed feedback via reward, and short-term and long-term consequences set RL apart from supervised learning. To gain an in-depth understanding of the agent's decision-making process, RL explainability needs to solve new challenges in addition to those from supervised learning. We illustrate this explainability difference in Fig. 3. These new challenges include understanding the short-term and long-term consequences of the agent's behavior and not just the immediate reasons. Moreover, understanding the agent's learning objective based on how the environment assigns rewards. Finally, in the event of a distributional shift, understand how changes in the starting state distribution and transition function affect the agent. Section 5.1 describes how we organize and classify XRL studies. Section 5.2 describes explanation types and RL explainability characteristics, indicating the types of stakeholder questions groups of methods can answer.

5.1 Taxonomy

Our taxonomy was constructed through several iterations and changed numerous times based on (1) existing XAI taxonomies for supervised learning (Arrieta et al., 2020; Guidotti et al., 2019; Burkart & Huber, 2021; Minh et al., 2022; Ras et al., 2022), (2) previously proposed taxonomies for XRL, and (3) studies from the searches. As a result, our taxonomy is a product of studies from our searches and previous taxonomies. We divide XRL methods into three categories: (1) interpretable agent (IA), (2) intrinsic explainability (IE), and (3) post hoc explainability (PHE), as seen in Fig. 4. IA refers to the agent being readily comprehensible and providing an understanding of the underlying learned relationships. These methods achieve inherent interpretability by representing the agent with a simple function approximator. IE describes methods modifying the RL system before training to make it explainable. PHE is similar to IE but endows the RL system with explainability without modifying it. The methods within PHE aim to extract information about the agent and its behavior after training. Although the categories IE and PHE overlap, we divide the methods into two categories for several reasons:

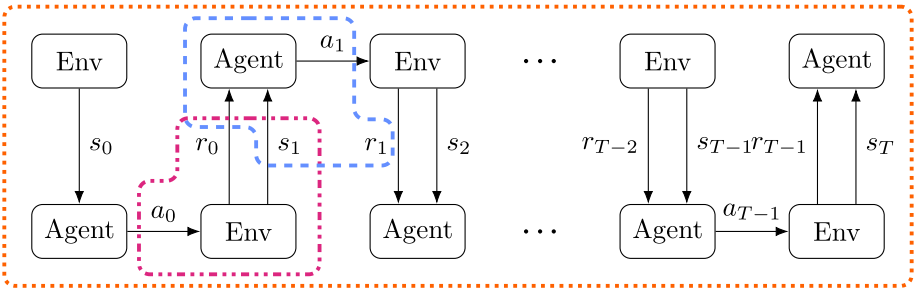

Fig. 3 The MDP of the agent interacting with the environment (abbreviated as env) unrolled. - - - illustrates comprehending immediate reasons for the agent's action, in other words, what explanation methods from supervised learning can explain. ····· and -·-· depict what we want to understand in addition to what - - - already provides. That is, we are interested in understanding the sequential nature of RL, including, for instance, short-term and long-term consequences of actions. Moreover, we are interested in understanding the environment as a way to comprehend the agent (Color figure online)

Performance impact The performance of RL agents is often positively affected or unchanged for methods in the IE category. Mott et al. (2019) show improved performance compared to models without attention bottlenecks. Likewise, Cultrera et al. (2020) show that adding attention leads to superior performance in addition to increased explainability. Tang et al. (2020) display better performance and generalization. Pan et al. (2019) indicate better data efficiency with their method. Other methods like Kim and Canny (2017) demonstrate competitive performance, but not significantly better. Similarly, Lin et al. (2021) do not show performance degradation and, in some cases, even perform better. Methods from other subcategories also show better performance. For example, Wang et al. (2021a) demonstrate that their method both converges faster and obtains higher episodic reward compared to other policies without modifications. Similar results are exhibited in Chen et al. (2022). Likewise, Kim et al. (2018) display better performance in comparison to other state-of-the-art methods. In Fukuchi et al. (2017a, 2017b), the explanation mechanism not only explains but also improves learning. In summary, IE methods increase the performance while methods in PHE do not affect the performance.

Agent architecture and training algorithm For methods in IE, the agent might require a specific neural network architecture, for example, Mott et al. (2019) with their model that has soft top-down attention mechanism, Lin et al. (2021) with their two-part agent, or Yang et al. (2019) with their variational autoencoder modified agent. Many methods in PHE have no such requirement. Nevertheless, some methods in PHE require specific prerequisites such as differentiability, but the requirement is less strict than in IE methods.

In IE, the agent's performance shown in studies is linked to the specific RL algorithm tested. Thus, the performance of methods in IE is uncertain on untested RL algorithms. PHE methods do not have such a concern since the training algorithm is detached from the explanation algorithm.

Training the agent For methods in IE, agents are trained from scratch or fine-tuned. Thus, a pre-trained agent cannot be explained unless it is modified and trained from the beginning or fine-tuned. This is disadvantageous if the performance of the already trained agent is satisfactory, and the stakeholder only wants to debug it for final verification. For PHE methods, the agent is not changed if training is involved. For example, training a distilled agent does not involve changing the original agent.

Applicability Methods from PHE can be applied to IE agents if certain prerequisites like functions being differentiable are satisfied. For example, a model with decomposed Q-values can be distilled into a decision tree or explained using feature importance methods.

Flexible agent access Many methods from PHE do not require access to the agent's internal logic. For example, the methods from the agent distillation category only require

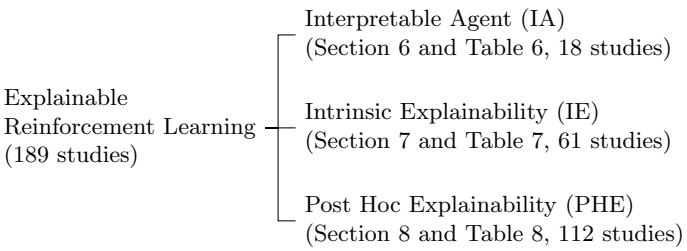

Fig. 4 Taxonomy of XRL methods. The categories do not sum to 189 studies because some span multiple categories

the state and the agent's corresponding action or Q-values. Another example is the important state and transition category in PHE, where many of the methods only require access to the agent's output in the form of Q-values, but not the internal logic. However, since the category is large, this does not apply to all methods in PHE. For instance, many feature importance methods require access to gradients while other perturbation-based feature importance methods only need to be able to probe the agent and receive its output.

Environment access Fewer PHE methods need access to the agent's internal logic compared to IE methods. But, in turn, greater access to the environment is required. Many of the PHE methods require access to input–output tuples that are assumed to be obtainable by simulating the agent in the environment or via a preexisting dataset. Although IE agents also necessitate this access, this happens during training for IE methods versus after training for PHE methods.

Our top-level categorization is similar to previous studies in XAI (Lipton, 2018; Murdoch et al., 2019; Du et al., 2020; Arrieta et al., 2020). However, the subcategories in this taxonomy are new and designed to accommodate the reviewed XRL studies. We go into details on these in the coming sections as follows. First, Sect. 6 details the methods that fall into the IA category. Next, Sect. 7 describes the methods in the IE category. Finally, Sect. 8 outlines the PHE method category.

Besides the taxonomy mentioned above, we classify the methods by their scope and focus in Section Appendix A. First, we define the method as global if it reveals the overall behavior of the agent, making it possible for the stakeholder to understand the agent's behavior in multiple states. In contrast, a local method only provides the logic behind the decision-making process that generalizes to a few states. We distinguish between two types of local scope: (1) methods explaining the short-term and long-term consequences, and (2) methods explaining using only the immediate context. Second, we classify methods by whether they try to: (1) solve the XRL problem, (2) solve RL specific problems (e.g., sample efficiency and generalization), and (3) solve application problems (e.g., applying XRL in healthcare, autonomous driving or some other domain).

5.2 Stakeholder questions: explanation types and RL explainability characteristics

Stakeholders have different questions they want to ask to satisfy their needs, and different XRL techniques provide different explanations. Some techniques might produce explanations that answer several questions, while others only answer one. This section first outlines six common explanation types used to explain stakeholders' questions (Lim et al., 2009; Mohseni et al., 2021). These explanation types are:

How does the agent work? A how explanation aims to give an all-inclusive answer to how the agent works and impart an understanding of its global behavior.

What did/will the agent do? A what explanation describes what the agent has done or will do. This is a descriptive explanation of the agent's behavior based on the history or predicted future.

Why did the agent do_? A why explanation justifies why the agent took a specific action.

Why did the agent not do _? The why not explanation describes why the agent did not choose a specific action, for instance, the stakeholder's anticipated action. This explanation type is also known as a contrastive explanation.

What would the agent do if _ happens? A what if explanation explains hypothetical questions of how the agent would behave in a specific situation. This type of explanation is known as a counterfactual explanation.

Howcan I get the agent to do _, given the current state? A how to explanation answers changes needed to get the agent to do a specific action. This explanation type is also known as a counterfactual explanation.

In addition to explanation types, we identify RL explainability characteristics. That is if the explanation produced includes information about short-term and long-term consequences or uses model information to explain (or both). We add this extra information since the explanation types do not provide the nuance needed to differentiate between different, for instance, why explanations. The short-term and long-term consequences describe if the explanation informs by referring to the future outcomes (e.g., what happens a few time-steps into the future or the result at the end of an episode). Model information refers to whether methods leverage the model (i.e., the transition and reward function) to explain the agent's behavior. We outline the explanation types and RL explainability characteristics for IA and the categories of IE and PHE in Tables 2, 3 and 4. The goal is to indicate what kind of stakeholder questions each category of methods can answer. Hence, making it more straightforward to find suitable methods to answer a particular question.

6 Interpretable agent

The interpretable agent (IA) category consists of agents innately understandable to humans. These methods do not require modifications to be interpretable. Instead, IA methods achieve interpretability by carefully choosing simple function approximators to represent the agent. The interpretable agent category aims to capture methods that are mainly motivated by interpretability. While there are many methods that support interpretability (Nikou et al., 2021; Illanes et al., 2020), interpretability is not their main motivation and is rather a by-product (Burkart & Huber, 2021). We do this in line with the selection criteria to keep this literature review focused on interpretability in RL.

The resulting explanation from these methods is the agent's representation, as illustrated in Fig. 5. Suppose that an agent is represented using a decision tree; then the decision tree itself is also the explanation. Their functional form allows stakeholders to inspect and understand them out of the box. Thus, the explanation is faithful to the policy being explained since it is the policy itself. Also, with their functional form comes inductive biases that bring advantages to generalization (Trivedi et al., 2021; Jiang & Luo, 2019). For instance, Jiang and Luo (2019) point out that relational inductive bias can help the policy generalize better than DRL policies that are not understandable. Furthermore, based on experiments on environments with symbolic representation, these methods offer competitive performance compared to their neural network counterpart (Silva et al., 2020; Qiu & Zhu, 2022; Trivedi et al., 2021).

Although these methods have many advantages, more complex environments may require functions represented using neural networks that are more flexible. Even if these methods can obtain high-performing policies in complex environments, decision trees might get too deep and rule lists too long, making them difficult to understand. Moreover, all of these methods are tested in environments where the state is low-dimensional with interpretable features. In environments where this is not the case, applying these inherently interpretable methods is not straightforward, for example, in environments using visual inputs like Atari (Bellemare et al., 2013). In these more complex environments, manual feature engineering is one possibility. However, one of the reasons why deep learning is performing so well is its ability to automatically extract features. Another approach is to use deep learning to extract features, but the feature extraction part is still a black box.

This section provides an overview of these methods organized by their functional form, as depicted in Fig. 6. Table 2 indicates the explanation types and the RL explainability characteristics that methods in IA can provide. The IA category does not explain the short-term and long-term consequences of actions since the MDP formalism only requires the agent to be reactive. The agent only needs to consider the current state and output an action. Consequently, we do not understand how the past or future (or both) affect the action choices at decision time by only inspecting the agent. Additional mechanisms are needed to gain an understanding of RL explainability characteristics for agents in the IA category.

6.1 Rule-based

The rule-based category presents methods where rules are used to express agents. The rules can be simple if-then conditionals or more complicated rules, for instance, incorporating fuzzy logic. "IF cart=right slope AND speed=high right then accelerate=positive" is an example of a rule learned by Hein et al. (2017b) in the mountain car environment to control the cart.

Hein et al. (2017b) describe the fuzzy particle swarm RL (FPSRL) method that focuses on industrial applications and interpretability. They represent policies using fuzzy rules and use a model-based approach to learn these rules. The model is learned and is leveraged to evade situations that can be dangerous when exploring while learning.

Real-world data usually includes numerous features, where many might not be helpful or redundant. Consequently, the resulting agent from FPSRL can be difficult to interpret since it uses all the features in all of its rules. In response to this difficulty, Hein et al. (2018a) describe the fuzzy genetic programming RL (FGPRL) method. To overcome this problem, FGPRL includes mechanisms to automatically select features, choose compact rules, and optimize policy parameters at the same time. Furthermore, besides introducing the new method, they also improved FPSRL by extending it with a new feature selection method. Hence, making it possible to apply FPSRL to industrial applications where states are high-dimensional.

Huang et al. (2020) present the interpretable fuzzy RL (IFRL) framework that uses the actor-critic architecture to learn policies represented as if-then rules. The rules produced by the framework are interpretable and allow stakeholders to add prior knowledge. Their method is motivated by previous methods' limitations. These limitations include specifying the policy structure beforehand and the inability to optimize policies before episodes end (Hein et al., 2017b, 2018b; Verma et al., 2018). Likmeta et al. (2020) introduce a rule-based policy for autonomous driving using RL. They sample the parameters from distributions that they optimize using gradient descent. Their work is based on the policy gradient with parameter-based exploration method (Sehnke et al., 2008). The method shifts exploration to the parameters to accommodate deterministic policies. In addition, it relaxes the differentiability requirement.

6.2 Mathematical expression

Physics has expressions that define complex phenomena in simple and compact mathematical expressions. Several studies present approaches representing RL policies using mathematical expressions to acquire interpretable agents by leveraging the same idea. The

equation $a = \frac{0.62}{\log(s_2)}$ exemplifies a simple policy produced by the method proposed by Landajuela et al. (2021) to control the cart in mountain car.

Like Hein et al. (2017b, 2018a, 2018b) present a model-based batch RL method that represents policies as mathematical expressions trained using genetic programming. Their work is motivated by interpretability, real-world applications, and difficulties with design choices regarding fuzzy rules. To find mathematical expressions representing value functions, Kubalík et al. (2021) use symbolic regression and genetic programming. Specifically, they describe three algorithms: symbolic value iteration, symbolic policy iteration, and a solution of the Bellman equation that can be obtained directly.

Kubalík et al. (2021)'s approach needs a model, and Hein et al. (2018b)'s approach results in lower performance than NN policies. Accordingly, Landajuela et al. (2021) present the deep symbolic policy method to discover policies represented using mathematical expressions where neural-guided search is leveraged. Their deep symbolic policy method consists of the policy generator and the policy evaluator. The generator is a recurrent NN that produces policies, and the evaluator assesses them and provides feedback to train the generator.

6.3 Logic-based

This category introduces methods that use logic expressions to represent the RL agent. Focusing on generalization and explainability in RL, Jiang and Luo (2019) present the framework neural logic RL (NLRL). The framework works with the policy gradient where states, actions, and policies are expressed in first-order logic. Their framework takes advantage of differentiable inductive logic programming (DILP) (Evans & Grefenstette, 2018) to learn interpretable and generalizable policies. Zhang et al. (2021b) present the off-policy differentiable logic RL (OPDLRL) framework. Their method tackles issues of execution efficiency, stability, and scalability of integrating DILP with DRL. OPDLRL solves the execution efficiency problem by using approximate inference and off-policy training. They employ maximum entropy RL to make the learning process stable. Lastly, they integrate hierarchical RL into the framework to make DILP scalable. The resulting framework resolves problems of combining DILP and DRL and yields interpretable policies. Another approach towards logic-based agents proposed by Gorji et al. (2021) apply a supervised learning method, the Tsetlin machine (TM) (Granmo, 2018), to RL by using a customized value iteration algorithm. Kimura et al. (2021) introduce a new method to learn interpretable rules as the policy using logical neural networks (Riegel et al., 2020). By using a semantic parser, the method first parses textual observations into first-order logical facts. Afterward, a logical neural network is inputted with these facts to learn rules.

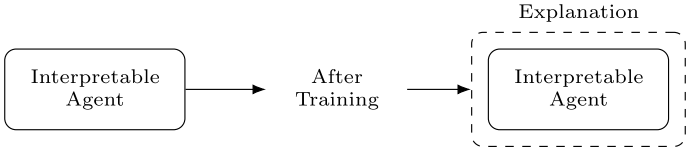

Fig. 5 The interpretable agent approach. The explanation is the agent itself communicated via its representation

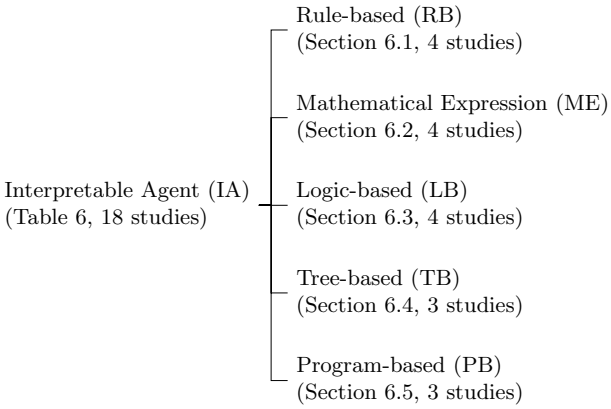

Fig. 6 Taxonomy of interpretable agent

Table 2 High-level overview of explanation types and RL explainability characteristics provided by IA methods

Explanation types						RL explainability characteristics	
How	What	Why	Why not	What if	How to	Short-term and long-term consequences	Model information
✓		✓	✓	✓	✓		

6.4 Tree-based

The tree-based category outlines the approaches that represent agents using a tree-based representation. Tree-based models such as decision trees (DTs) are considered interpretable, presuming they are small in terms of depth and simple based on how splits are executed. However, they cannot be trained online using continuous optimizing like NNs, thus, they are trained offline. Responding to these considerations, Silva et al. (2020) introduce the differentiable DTs (DDTs) approach that learns DTs using online optimization. They extend Suárez and Lutsko (1999)'s work by highlighting and fixing two problems that hinder interpretability: (1) how to do splits and (2) how many features to use in each split. Besides dealing with these two disadvantages concerning interpretability, they also provide a theoretical analysis of DDTs.

In DDTs (Silva et al., 2020), function approximators like NNs cannot be taken advantage of and the internal representations in the nodes cannot be substituted. Other works on DTs for RL, such as VIPER (Bastani et al., 2018) and MoËT (Vasic et al., 2022), use imitation learning. Consequently, Topin et al. (2021) propose the iterative bounding MDP (IBMDP). The IBMDP extends the MDP formalism by wrapping around it and adding bounds for state features and additional actions. The key is that a policy learned using IBMDP will equal a decision tree policy for the MDP. Thus, if we learn a neural network policy for the IBMDP, then a corresponding decision tree policy can be extracted for the MDP. The same work shows how existing RL algorithms can be modified to solve the IBMDP.

6.5 Program-based

The program-based category introduces methods that represent policies structured in domain-specific languages. Trivedi et al. (2021) apply a variational autoencoder (Kingma & Welling, 2014) to learn a latent program space. They train the variational autoencoder to reconstruct randomly produced programs where policies with similar behavior are close to each other in the latent space. After learning the latent program space, they find the agent's policy by maximizing the return using the cross-entropy method. Premade program templates are used by previous work on program-based policies (Verma et al., 2018, 2019). Since they produce the programs without templates, Trivedi et al. (2021) argue that their method produces more flexible policies. Qiu and Zhu (2022) propose a method to train program-based policies using policy gradient via differentiability requirement relaxation. Their method learns the architecture and parameters of the policy simultaneously by taking advantage of the progress in the neural architecture search literature. Similar to Trivedi et al. (2021), they avoid the issue of fidelity and faithfulness since an imitation learning based approach is not used. Unlike Trivedi et al. (2021), they do not need to learn a latent program space utilizing a premade dataset of programs. Cao et al. (2022) propose a domain-specific language synthesis method that adds the benefits from both imperative and declarative programming. With their method, they can synthesize hierarchical cause-effect logic programs that have good generalization and interpretability. They compare their method with various baselines in the MiniGrid environment, showing that their method has better learning ability, generalization, and interpretability.

7 Intrinsic explainability

Intrinsic explainability (IE) describes methods that modify the agent or model (or both) to make the RL system explainable. When we say model in this context, we refer to the transition and reward function. For instance, a method that reduces the state space before training, making the agent operate in the reduced state space. Accordingly, it becomes easier to comprehend the agent since the stakeholder needs to inspect fewer situations to gain a global understanding of the behavior. Alternatively, if the agent is represented as a NN, a method that modifies the NN architecture, such as adding an attention module, so the agent can produce saliency map explanations during the forward pass.

The methods change the agent to endow it with the ability to generate explanations. Figure 7 illustrates these examples of approaches where methods transform the agent or model (or both) into their explainable counterpart. IE methods apply the modifications before they train the agent. Consequently, the modifications enabling explainability are tied to the agent and its training and affect the agent's performance. We divide methods into categories based on how they represent and communicate the explanations. A complete overview of all IE categories is illustrated in Fig. 8.

Table 3 indicates explanation types and RL explainability characteristics provided by the different IE categories. As we can see, the methods in the subcategories produce diverse explanation types and can explain sequential information that methods from the IA category cannot. By offering this table, we hope it becomes easier for stakeholders to find a suitable method for their task.

7.1 Explanation via generation

This section describes IE methods that modify the agent to generate an object explicitly representing the explanation. The object can be a saliency map, a textual response, or some other explanatory object given to the stakeholder as the explanation. For instance, the explanation "the car stops because the light is red" (Ben-Younes et al., 2022) in autonomous driving illustrates the representation of explanations communicated by methods in this category.

7.1.1 Feature importance

In this section, we overview methods that modify the agent so it can explain by highlighting important features using saliency maps. Saliency maps are defined for most methods in this section as highlighting task-relevant information in the input. Nevertheless, there are some exceptions to how saliency maps are defined, which we outline at the end of this section. The modifications to the agent involve changing the agent's NN architecture. These methods mainly aim to answer the why question by pointing out features affecting the agent's behavior.

Kim and Canny (2017) propose an explainable self-driving agent represented using a modified convolutional NN architecture. The agent produces explanations in the form of saliency maps. Clustering and filtering are used to make the explanations concise after generating the saliency maps. These saliency maps emphasize important input parts that impact the agent's behavior causally. In a similar line of work, Cultrera et al. (2020) introduce an end-to-end model for autonomous driving that can explain its decision using saliency maps. Their approach does not involve post-processing in contrast to Kim and Canny (2017)'s approach, which does. Also working on driving agents, Bao et al. (2021) introduce the deep reinforced accident anticipation with visual explanation (DRIVE) model. In traffic accident anticipation systems that already exist, methods to create visual explanations are lacking. In response, DRIVE was created to make visual explanations in the context of accident anticipation. DRIVE merges two kinds of attention by leveraging the dynamic attention fusion method proposed by the authors. The result of combining these attentions is improved accident anticipation and better saliency maps.

Goel et al. (2018) propose the motion-oriented RL (MOREL) method. Their method is motivated by the need for more sample-efficient and explainable systems. In addition, they point out the disadvantage of requiring hand-crafted templates by a previous approach (Iyer et al., 2018). MOREL works by first learning a representation that can be used to find and segment objects in inputs. The representation is later utilized to train the policy. As a result, learning a high-performing policy requires fewer environmental interactions. Moreover, the learned representation makes creating saliency and optical flow maps possible. The saliency map emphasizes the agent's confidence that objects exist at given locations. At the same time, the motion of the objects is captured by the optical flow map. Mott et al. (2019) propose a new method that uses soft attention to create saliency map explanations. The explanations generated by their system aim to focus on features impacting the agent's behavior both in the present and future. According to the authors, compared to existing saliency methods for RL (Zahavy et al., 2017; Greydanus et al., 2018), explanations created by their method are easier to understand. Using the asynchronous advantage actor-critic (A3C) algorithm, Itaya et al. (2021) train convolutional NN architectures with two attention modules. The built-in attention

modules enable interpretation from two perspectives by explaining the control and state value separately. According to the results, the policy performs better with the attention modules and, at the same time, facilitates explainability. Aiming to capture the input content that causally affects the output, Dai et al. (2022c) introduce a module named conceptual embedding that they integrate into DRL agents. The conceptual embedding extracts concepts by compressing the high-dimensional state into a compact representation. After extracting concepts, importance values are assigned to them via perturbation to explain the agent. In this work, they assume that there is a causal relationship from observation to action. Thus, they can explain the cause and effect between concepts and actions.

Integrating attention modules into an agent's architecture can hamper its performance (Nikulin et al., 2019). Nikulin et al. (2019) describe a new module that can be inserted into a convolutional NN agent instead of proposing a new modified architecture. They demonstrate that the agent does not show degraded performance with the new module via experimentation and at the same time provide explainability.

Visual inputs contain many features, RL agents must distill inputs to obtain the relevant features. However, trying to extract them using brute force can affect training and explainability negatively. To resolve the issue, Zhang et al. (2021c) propose to divide the decision-making process into two parts, first finding the task-relevant features and then using those features to make decisions. To explain decisions, they describe the temporal-adaptive feature attention algorithm to explain the importance of the features. Similarly, Wei et al. (2022) introduce a feature selection approach based on attention. The introduced method identifies important features and then assesses the features' importance. Liu et al. (2022) present the adaptive region scoring (ARS) module, which is motivated by how humans process visual data. Their method is incorporated into an agent by modifying the feature extractor, which provides explainability.

In addition to explainability, several other reasons motivate many studies. Focused on generalization, Tang et al. (2020) use neuroevolution to train agents with self-attention. The self-attention module can be used to explain the agent's decision-making. Also motivated by generalization, a NN architecture of the agent with relational reasoning is proposed by Zambaldi et al. (2019), which is also used for explainability purposes. Josef and Degani (2020) introduce a DRL agent with built-in attention to provide explainability in the context of safe unmanned ground vehicle navigation that happens in rough terrains. Kim et al. (2022) describe integrating attention and risk-sensitive agents to yield explainability. In addition, they argue for being the first to work on saliency maps and risk-sensitive agents by reviewing several XRL studies. Finally, Wang et al. (2022) work on incorporating saliency map explanations into the exploration strategies of agents to explore more effectively.

Below, we concisely list how saliency is defined for each method:

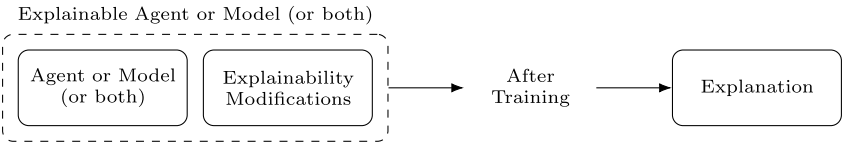

Fig. 7 In the intrinsic explainability approach, methods modify the agent or model (or both) to enable the RL system to become understandable and able to produce explanations

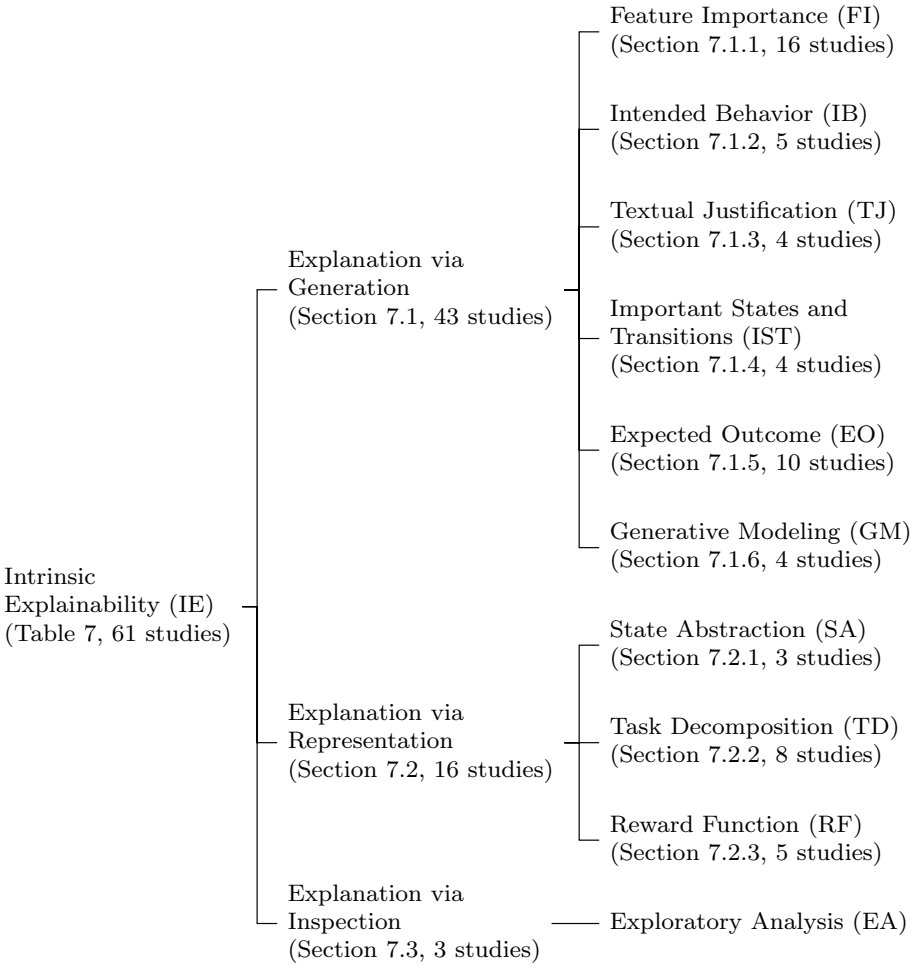

Fig. 8 Taxonomy of the intrinsic explainability. We separate the category based on how the explanation is conveyed: (1) via generation, (2) via representation, and (3) via inspection. The categories do not sum to 59 studies because some span multiple categories

- Goel et al. (2018) provide two different saliency maps, one highlighting objects and another highlighting flow information for each moving object.
- Mott et al. (2019) highlight important task-relevant information in visual inputs. From their method, we can get two different saliency maps, one on where the agent looks and the other on what the agent looks at.
- Zambaldi et al. (2019) produce saliency maps that show what different entities in the input space attend to, which shows the relationship between the entities.
- Bao et al. (2021) produce two different saliency maps, one highlighting the most salient objects, while the other focuses on risky regions in traffic accident anticipation. These two are merged via weighted sum to create a single saliency map that improves traffic accident anticipation.
- Dai et al. (2022c) highlight relevant concepts using perturbation. Concepts are found via a layer termed concept embedding that compresses the observation.

Table 3 High-level overview of categories of methods in the IE based on their explanation types and RL explainability characteristics

Category	Explanation types						RL explainability characteristics	
	How	What	Why	Why not	What if	How to	Short-term and long-term consequences	Model information
Feature importance			✓					
Intended behavior		✓					✓	
Textual justification			✓	✓				
Important states and transitions	✓	✓					✓	
Expected outcome			✓	✓			✓	
Generative Modeling				✓	✓	✓		
State abstraction	✓		✓					
Task decomposition		✓	✓				✓	
Reward function	✓							✓
Exploratory analysis	✓							

- Kim and Canny (2017), Nikulin et al. (2019), Cultrera et al. (2020), Josef and Degani (2020), Tang et al. (2020), Itaya et al. (2021), Zhang et al. (2021c), Kim et al. (2022), Liu et al. (2022), Wang et al. (2022), Wei et al. (2022) highlight task-relevant input features.

Methods in this category provide explanations that are easy to convey, as long as the stakeholder understands the features. These explanations can be used to confirm whether an agent is looking at "reasonable" features rather than spurious ones. In addition, explanations are generated during the forward pass, and thus, they do not require much more computational power. On the downside, for visual inputs, which most methods in this category focus on, the methods are mostly limited to where the agent looks. Ideally, a stakeholder would not only like to know where the agent is looking at, but also what it is looking at. For example, is it the car, the car's color, or the edges of the car triggering the agent's response? These ambiguities make it difficult to understand the explanation. Furthermore, it has been shown that it is hard for humans to detect spurious signals, even if the saliency explanations can show them (Adebayo et al., 2022). Compared to post hoc saliency methods, explanations of these methods are faithful since they are used in decision-making. Although this has been contended, for example, attention is not the same as explanation (Jain & Wallace, 2019; Wiegreffe & Pinter, 2019).

7.1.2 Intended behavior

The intended behavior category describes methods enabling the agent to inform the stakeholder about planned actions for several steps into the future. For example, the explanation, "I will go left" (Fukuchi et al., 2017b) by a robot in a human-robot collaboration task. Knowing the planned actions makes it possible for the stakeholder to anticipate the agent's

behavior. Thus, answering the what question with sequential information embedded into the explanation.

Focusing on human-robot collaboration, Hayes and Shah (2017) introduce a method to answer questions like "When do you do _?", "What do you do when _?" and "Why didn't you do _?". Their approach consists of parsing the stakeholder's queries by matching them with pre-made templates, finding states matching the stakeholder's queries, and generating explanations in natural language explaining the matching states. According to Fukuchi et al. (2017b), Hayes and Shah (2017)'s approach has three limitations. (1) Needs manual engineering, (2) assumes that the policy will not change, and (3) only the immediate context is used to explain actions. To resolve these issues, Fukuchi et al. (2017b) introduce the instruction-based behavior explanation (IBE) method that uses interactive RL. In this framework, an agent gets instructions from an expert. They assume that the agent followed the instructions if the agent received high rewards in the episode. The instructions can speed up the agent's learning and are saved for later use by the agent to explain its actions over a short term. They use clustering to explain situations with saved instructions. However, if the policy parameters get updated, this method will not work and needs to be revised. In response to this limitation, Fukuchi et al. (2017a) propose using a supervised learning approach instead of clustering to translate from state to explanation. Extending these two studies, Fukuchi et al. (2022) focus on the connection between the agent and the stakeholder. More specifically, they focus on the communication divergence that may arise between them due to different goals.

Leveraging probabilistic graphical models, Chen et al. (2022) and Wang et al. (2021a) propose an end-to-end driving system. To explain the driving system, they output a semantic mask that provides a bird's-eye-view of road conditions, objects in the car's surroundings, and routing information. The semantic mask shows the car's perception, comprehension of the driving situation, and planned driving route. The planned route gives the stakeholder an understanding of the vehicle's short-term behavior.

For human-robot collaboration, the type of explanation offered by methods in this category is useful since it reveals the agent's intent. The main use for these types of explanations is during real-time collaboration and when the main interest is in the agent's future behavior close in time. On the downside, depending on the task and how far into the future explanations explain, they may have limited usefulness. For these methods to be useful in real-time situations, the explanations must be sparse and fast to produce.

7.1.3 Textual justification

Textual justification methods enable the agent to provide textual explanations in natural language to the stakeholder. For example, the explanation "The car slows down because it is preparing to turn to the road." from Kim et al. (2018) explains the behavior of a driving policy. Although textual response explanations can answer a variety of questions, the existing methods in this category mainly respond to the why and why not questions.

To create textual explanations for driving agents, Kim et al. (2018) introduce a new method by extending Kim and Canny (2017)'s work. They create faithful explanations for the driving policy rather than making rationalizations that aim to explain how a human spectator would explain an action. To achieve this, they utilize visual explanations to produce textual justifications. The visual explanation is produced by an attention model

represented as a feed-forward neural network that outputs importance values. The neural network is given state features and the previous hidden state from the LSTM model that represents the policy. The textual explanation is similarly generated by a separate neural network, in this case, an LSTM model. In the same work, they create a new dataset, Berkeley deep drive-X, that partially enriches the Berkeley deep drive dataset (Xu et al., 2017) with textual justifications. Focusing on the same applications, Ben-Younes et al. (2022) describe a new method to create textual justifications. This method differs from the previously mentioned approach in two ways. First, they use a different approach to create faithful explanations. Second, they focus on generating explanations in the online setting. Besides these works focusing on autonomous driving, Wang et al. (2019b) describe a new method to create faithful textual justifications by leveraging attention.

Cruz and Igarashi (2021) propose interactive explanations using templates in natural language. Utilizing these interactive explanations, stakeholders can find and fix bugs. In addition, stakeholders can make the agent's behavior align with their preferences. In short, they propose actionable and interactive explanations that are more than just explanatory.

Textual explanations can be easier to understand for a larger group of stakeholders than other types of explanations. Depending on the design of the textual explanations, stakeholders do not need to understand the inner workings of the agent. However, the cost of textual explanation is higher, since some form of human intervention is often needed. If there is a dataset that can be used for explanations, it is often limited to certain domains. For example, the Berkeley deep drive-X can only be exclusively used for driving environments.

7.1.4 Important states and transitions

The methods within this category explain the agent by pinpointing important states and transitions encountered during learning or after. These states and transitions can be situations where an alternative action can significantly affect the agent's learning or future outcome (or both). The goal of these methods is to align the agent's and stakeholder's mental models through examples of situations. As these situations communicate diverse agent behavior and provide a global overview, the aim is to answer the how and what questions.

According to Dao et al. (2018), Zahavy et al. (2017)'s approach requires manual feature engineering, and Greydanus et al. (2018) provide local explanations that do not give insights into the training process (information about these methods is given in Sects. 8.1.1 and 8.2.1). Motived by these shortcomings, Dao et al. (2018) describe DRL-Monitor. DRL-Monitor saves important transitions the agent encounters during learning that can later be analyzed to gain insights. The approach extends the sparse Bayesian RL (SBRL) (Lee, 2017) method, which requires feature engineering that DRL-Monitor does not. On the downside, DRL-Monitor saves too many transitions, pointed out by Dao et al. (2021). This, in turn, makes it costly to use DRL-Monitor. To overcome this limitation, Dao et al. (2021) present a new approach to balance the information retained and the number of transitions saved. Accordingly, fewer transitions are saved, making it less laborious to analyze them. In a similar line of work, Mishra et al. (2018) introduce Visual-SBRL that aims to save important transitions. However, unlike DRL-Monitor, Visual-SBRL does feature engineering via an autoencoder.

The standard MDP formalism is extended into the lazy-MDP by Jacq et al. (2022). In the lazy-MDP, we have a policy trained to solve the standard MDP called the default policy. In addition, we have the lazy policy trained to solve the lazy-MDP. For every action the agent has to make, the agent can either delegate action selection to the default policy or use

the lazy policy and get a penalty. Thus, the lazy policy will only be used to act if the action selection is critical, where the penalty matters less than the outcome, showing a new way to identify critical states.

This form of explanation is useful if justifications for specific situations are not needed. It is valuable if the goal is to understand the agent's behavior in general. The difficulty with these methods is to find states helpful to the stakeholder, which can differ based on their needs. Thus, the importance measure used to assess which states to save must be adapted to the situation. Another difficulty is whether looking at a state is enough for the stakeholder to understand why a state was picked, or if more information is needed.

7.1.5 Expected outcome

This section presents studies that aim to answer the short-term and long-term consequences of the agent's decisions. The consequence can range from what is encountered for choosing a specific action to how much time it will take to reach the goal state because of that action. Additionally, the methods in this category can contrast the outcomes of different actions, thus, answering why not questions.

A set of methods decomposes the reward into interpretable components, since finer details provide a better understanding (Erwig et al., 2018; Juozapaitis et al., 2019; Anderson et al., 2019). For example, Juozapaitis et al. (2019) show that in a gridworld environment, the reward can be decomposed into the cliff, gold, monster, and treasure. By using decomposed reward rather than a single numerical value, the methods can, in turn, learn decomposed Q-values that are more meaningful than plain Q-values. These decomposed Q-values can be used to explain by pointing to the outcome and contrasting the consequences of various actions in a state. Although more meaningful, Anderson et al. (2019) demonstrated that different situations require different explanations. Moreover, they show that reward decomposition and saliency maps complement each other. Focusing on safety in human-robot collaboration, Iucci et al. (2021) introduce a new method that integrates the reward decomposition method with Hayes and Shah (2017)'s method. When both methods are used together, the stakeholder's trust increases because of better explainability. Likewise, Rietz et al. (2022) propose a new XRL method by extending the reward decomposition method. According to the authors, the reward decomposition method lacks a high-level overview and context. To resolve this issue, they integrate hierarchical RL with the reward decomposition method. Feit et al. (2022) focus on explaining deep RL for self-adaptive systems by combining two existing methods: reward decomposition (Juozapaitis et al., 2019) and interestingness elements (Sequeira et al., 2019). They argue reward decomposition suffers from not providing states that are interesting for a stakeholder to understand. Furthermore, the interestingness elements method provides states that may interest a stakeholder but does not provide details beyond that. Hence, by combining these methods, both weaknesses are addressed. Terra et al. (2022) introduce a new method, both ends explanations for RL (BEERL). BEERL aims to explain both input features and output rewards. They reason that existing methods like saliency methods only explain input features, while reward decomposition (Juozapaitis et al., 2019) only considers rewards when explaining. To explain both components, they propose BEERL which utilizes both by correlating feature importance with output rewards, giving stakeholders more comprehensive explanations.

Building upon the idea of contrasting the outcome in reward decomposition, Lin et al. (2021) propose a technique where they first construct interpretable features and then use them to predict Q-values. They construct a two-part agent, which they coined the embedded self-prediction model. The first part predicts the expected discounted cumulative features. At the same time, the second part utilizes these aggregated features to predict Q-values. By contrasting the expected cumulative features of the actions, their method can generate contrastive and minimal sufficient explanations (Erwig et al., 2018; Juozapaitis et al., 2019). Instead of explaining the result of an action, Yau et al. (2020) explain the time it takes for the agent to get to an episode's end. Yau et al. (2020) present an approach to estimate the expected discounted number of state visits from a state to explain the policy. Their approach is motivated by the fact that goal-oriented explanations are associated with 70% of daily life explanations. Additional information is saved during policy learning since this information cannot be extracted from the Q-function to create these goal-oriented explanations. Focusing on autonomous driving, Pan et al. (2019) introduce the semantic predictive control framework. Their method forecasts the evolution of features to explain to stakeholders the future outcome of actions.

Similar to the intended behavior category, methods in this category offer explanations explaining the future of a specific situation. In contrast, methods in this category offer more detailed explanations that justify actions in terms of future outcomes and not only short-term outcomes of actions. This is more helpful in situations where time is not a pressing matter and more detail is needed. To understand explanations presented by methods in this category, more domain knowledge is needed, since there is often reference to rewards and engineered input features.

7.1.6 Generative modeling

This category consists of approaches to understanding the agent using generative models. These generative models can, for example, be variational autoencoders and generative adversarial networks. By utilizing the latent encoding of these generative models, methods in this category can create why not, what if, and how to explanations.

Yang et al. (2019) propose a new architecture, the action conditioned (AC)-β variational autoencoder. First, the method disentangles the latent space into interpretable dimensions. Then, the policy uses the interpretable dimensions to make decisions and reconstructs them by conditioning on the actions. The goal is to understand how the interpretable dimensions affect the actions by moving in the latent space and reconstructing them using the decoder. Rupprecht et al. (2020) propose a generative model similar to the variational autoencoder. The new model comes with a new loss function and aims to generate counterfactual states to comprehend the RL agent. First, they modify the evidence lower bound so that the agent interprets both the inputs and reconstructions similarly. In addition, they extend the reconstruction loss to concentrate more on the crucial input areas. Finally, they introduce a new method to generate counterfactual states that can be interesting and useful.

Olson et al. (2019, 2021) present a method using deep generative models to make counterfactual states. The counterfactual states are made by moving in the latent space and used by the policy to make decisions. Doing so makes it possible to ask what if questions, in turn, understand the policy's behavior in new states. The proposed architecture utilizes the adversarial autoencoder (Makhzani et al., 2015) and the Wasserstein autoencoder (Tolstikhin et al., 2018).

One of the greatest challenges with methods in this category is to generate realistic counterfactual states. When generating counterfactual states, it is important for them to be in-distribution, something that is actually plausible. The danger is that generated states are out-of-distribution, showing an agent's unrealistic behavior. With unrealistic counterfactual states, a stakeholder might trust the results less. On the bright side, these methods offer a quicker way to understand an agent's behavior in interesting states without running numerous simulations to find these interesting states.

7.2 Explanation via representation

This section outlines methods providing interpretability through communicating the agents' representation like in Sect. 6, for instance, a decision tree agent where the decision tree itself is the explanation. Providing interpretability via representation also includes, for example, how we express the state space, since reducing it can also alleviate interpretability problems. Compared to Sect. 7.1, the methods in this section do not produce explanations by explicitly generating an object.

7.2.1 State abstraction

The state abstraction category details methods making the agents more interpretable by reducing the state space. The reduction typically happens by clustering states into abstract states. Reducing the state space reduces the number of situations we must consider when trying to understand the agent's behavior. Thus, not directly addressing explainability, but still helps to make it easier to interpret.

For environments where simulation costs are high, Bougie and Ichise (2020) argue that Verma et al. (2018)'s method might not be appropriate (information about the method in Sect. 8.2.2). Hence, they propose an approach where they create rules and then use them to cluster states. They train the agent by leveraging these abstract states deduced from the rules. As a result, this makes it possible to modify several Q-values simultaneously, increasing sample efficiency and interpretability. Akrour et al. (2021) introduce an approach using a mixture of experts represented using fuzzy logic with interpretable experts. Their agent chooses actions by considering the current state with a list of abstract states that has to be small to enable interpretability. The states representing each abstract state are chosen from interaction data to guarantee interpretability.

Focusing on the time discretization problem in batch RL and the healthcare setting, Zhang et al. (2021a) propose to create abstract states by locating states they term decision points. Decision points are states deemed important and where patients are given a different treatment, although similar. They use batch data to determine these decision points, cluster them, and train the agent using the resulting abstract states.

With a reduced state space, it is easier for a stakeholder to understand an agent's behavior. It might become easier to determine when certain actions are executed and what conditions trigger them. Also, since these reductions are used during the agent's learning, they can speed up learning. However, even the reduced state space can be overwhelming and large in complex environments. Thus, methods from this category might work better if they are combined with methods from the important states and transitions category to select or highlight abstracted states for inspection.

7.2.2 Task decomposition

The task decomposition category contains methods that make the agent interpretable by breaking a task into smaller and more compact problems. Essentially, the methods in this category use a divide-and-conquer procedure to solve the XRL problem. A common RL approach to dividing a task into subtasks is through hierarchical RL. Hierarchical RL decomposes a task into a hierarchy of subtasks, where we can solve the parent task by solving the child tasks, using them as primitive actions (Hengst, 2010).

Motivated by lifelong learning, Shu et al. (2018) describe a new hierarchical RL framework. In this framework, the agent learns to act by recursively utilizing previously learned policies to train new policies to solve a problem. In addition, a stochastic temporal grammar model is used to keep a tab on the connections between the tasks. Finally, each task is labeled using human language to keep the framework interpretable. Likewise, focusing on lifelong learning, Wu et al. (2020) introduce the model primitive hierarchical RL (MPHRL) framework. In MPHRL, they assume that substandard models of the world exist. These substandard world models perform well in a specific area but suboptimal outside. Utilizing these world models, they do task decomposition and learn several sub-policies. After training, these sub-policies are used by a gating controller as a mixture of experts to act. Beyret et al. (2019) propose a new hierarchical RL method, named dot-to-dot, that focuses on solving robotic manipulation tasks. In this method, a high-level policy learns sub-goals and manages and assigns subtasks to sub-policies based on learned sub-goals. The sub-policies try to maximize the return for the sub-goals, while the high-level policy tries to maximize the overall return. These subtasks are smaller and potentially more manageable. Thus, when a stakeholder wants to understand the agent, it can inspect the high-level policy without getting bogged down by details in the sub-policies. Likewise, Ye and Yang (2021) propose a hierarchical RL approach named hierarchical policy learning with intrinsic-extrinsic modeling (HIEM) for object finding tasks. HIEM similarly employs high-level and sub-policies to solve tasks. Gangopadhyay et al. (2022) introduce the hierarchical program-triggered RL (HPRL) framework, which focuses on autonomous driving. Similarly to the previous approaches, HPRL utilizes a high-level policy and sub-policies. Specific to HPRL, the high-level policy is represented as a structured program that can be inspected and overrule sub-policies for safety.

Lyu et al. (2019) present the new framework, symbolic deep RL (SDRL). SDRL consists of a symbolic planner, meta controller, and controller. The symbolic planner does long-term planning, the controller learns policies to act, and the meta controller evaluates and bridges these components. Hasanbeig et al. (2021) describe DeepSynth that aims to solve problems with sparse reward and partial observability. DeepSynth learns a deterministic finite automaton (DFA) that keeps track of the tasks' sequential dependencies. For each DFA state, there is a policy specializing in the task. The DFA can be inspected to gain insights into the decision-making process.

Breaking down an action into smaller actions will provide a stakeholder with a better understanding of how an agent behaves. However, they still miss shedding light on the smaller decomposed actions and only answer why questions for the high-level actions. This is especially difficult in cases where the smaller actions taken do not match human intuition. Moreover, there is the difficulty of how to decompose an action and how these methods will scale to more complex environments.

7.2.3 Reward function

The reward function category includes various methods that leverage the reward function to understand the agent. For example, explicitly representing the reward function in an interpretable format. Doing so helps stakeholders understand the agent's goal by understanding the reward assignment. In addition, using the reward function, a stakeholder can model and align agent behavior with its preferred behavior.

Tabrez et al. (2019) study human-robot collaboration where an accurate mental model of the task is crucial since it leads to safer and smoother teamwork. They describe the reward augmentation and repair through explanation (RARE) framework that aims to assist a stakeholder. More specifically, it assumes that the stakeholder has an internal reward function that the agent can estimate. If the reward function is wrong, the agent explains it to the stakeholder to help correct the reward function. Thus, their mental models will align, which improves teamwork.

To specify an agent's behavior, Li et al. (2019b) employ formal methods to define an interpretable reward function. Similarly, Bautista-Montesano et al. (2020) describe a new approach that uses fuzzy logic to define the reward function in the context of autonomous driving. To learn an interpretable reward function represented using a tree-based model, Bewley and Lécué (2022) describe an approach using preference-based RL. They create and refine tree-based reward functions using human preferences over behaviors. Compared to the previous methods, this approach advantageously offers several ways to express the reward function. Bica et al. (2021) aims to learn what if explanations of expert behavior from batch data in terms of an interpretable reward function. To achieve this, they leverage counterfactual reasoning and use batch RL since interactive learning is impossible in healthcare. The reward function is expressed as a linear function of the expected outcomes conditioned by history. A linear function is inherently interpretable and can be inspected to understand how experts from various organizations reason and value different outcomes when making decisions.

Reward functions can be difficult to specify (Abbeel & Ng, 2004), but having an interpretable reward function can increase the understanding of agent behavior. However, knowing the reward function and understanding how it works does not stop the agent from reward hacking and learning unwanted behavior. Hence, although having an interpretable reward function helps, it does not fully shed light on the behavior the agent will learn. There are exceptions to this where the reward function itself is closely tied to the learning process, for instance, Bewley and Lécué (2022). The main use case for methods in this category is when stakeholders want to modify the reward function and, in turn, change the agent's behavior.

7.3 Explanation via inspection: exploratory analysis

The explanation via inspection category presents studies that propose new agent representations, making it easier to analyze agents. However, a stakeholder needs to inspect, analyze, and assess the explanations manually to extract insights from the agents. Compared to the methods we have already seen, the explanations produced by the methods in this category are more open-ended.

Using a modified NN architecture, Annasamy and Sycara (2019) describe a new approach using Q-learning and an autoencoder with key-value memory. The key-value memory can be analyzed using t-distributed stochastic neighbor embedding

(t-SNE) (van der Maaten & Hinton, 2008) to produce global explanations. However, t-SNE can be challenging to use, although often used in the context of explainability (Wattenberg et al., 2016). In addition to these global explanations, the method can produce local explanations using saliency maps.

Focusing on robot collision avoidance, Kuramoto et al. (2020) present a new NN architecture where the network's hidden layers are easily visualizable to understand the agent. Tylkin et al. (2022) use neural circuit policies (NCPs) to represent agents in the flight domain. NCPs are used since they have few neurons making analyzing the agent easier, such as visualizing neuron activations and characterizing them using decision trees.

The open-endedness of these methods requires more human labor and is more suited in situations where there is sufficient time to explore the explanations. Nevertheless, they are very useful in cases where a more detailed analysis of an agent's inner workings is needed. In conclusion, these methods are suitable for situations where the neural network architecture needs to be more interpretable.

8 Post hoc explainability

Post hoc explainability (PHE) consists of methods applied to uninterpretable agents or models (or both). The aim is to extract insight and produce explanations without changing the agent, as shown in Fig. 9. A method being post hoc is not the same as being model-agnostic. For example, some techniques in this category can only be applied to NNs. There are various reasons for using an uninterpretable agent. For example, a company has invested in an existing agent and is satisfied with its performance. However, they need to extract explanations to demonstrate that it is safe. Instead of starting anew, the company can use post hoc techniques. Moreover, situations requiring flexible function approximators might make it impossible to use interpretable agents. This section overviews the different categories of the post hoc explainability approach depicted in Fig. 10.

Table 4 describes which explanation types and RL explainability characteristics the different PHE categories can provide. Like the methods in Sect. 7, the methods in the PHE category create a diverse set of explanations. Using this table as a guide, stakeholders can select categories to satisfy their explainability needs and narrow them down to specific methods.

8.1 Explanation via generation

Similar to Sect. 7.1, this category describes methods that generate an object as the explanation. The explanation can be visual, textual, or some other format. For instance, the explanation can be "I inspect a part when the stock feed is on and I detect a part" (Hayes & Shah, 2017) in a robotic inspection task. Unlike the methods in Sect. 7.1, we can apply methods in this category to pre-trained agents without modifying them.

8.1.1 Feature importance

The following section introduces post hoc feature importance methods. On the one hand, unlike the methods in Sect. 7.1.1, the post hoc ones are not built into the agent architecture, making these techniques more flexible. On the other hand, methods cannot positively

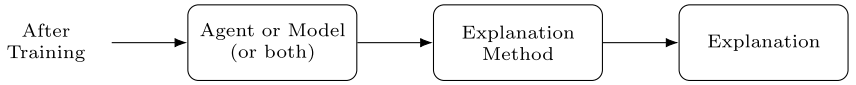

Fig. 9 The post hoc explainability approach. To comprehend the agent's decision-making process, we apply the explanation method to the agent or the model (or both) after training. In this context, the model refers to the transition and reward function

affect the training of agents. Moreover, we need to make sure that these methods produce faithful explanations as the explanation generation process is separate from the agent's decision-making process. This category of methods provides the same explanation as the methods presented in Sect. 7.1.1, namely the why explanation. Additionally, some techniques presented here can also give global explanations, like Shapley additive explanations (SHAP) (Lundberg & Lee, 2017). Thus, some methods provide the how explanation.

Zahavy et al. (2017) contribute with several techniques to better understand an agent. Similar to Simonyan and Zisserman (2015), they produce saliency maps using a

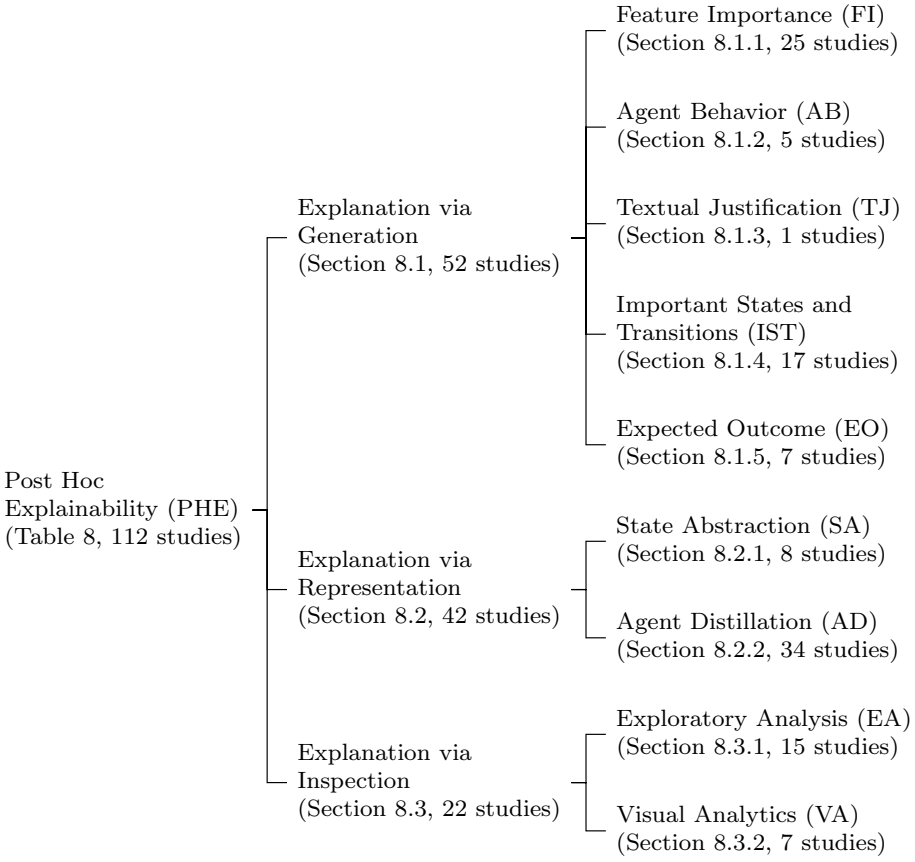

Fig. 10 Post hoc explainability taxonomy. We separate the category based on how the explanation is conveyed: (1) via generation, (2) via representation, and (3) via inspection. The categories do not sum up because some studies span multiple categories

Table 4 High-level overview of the categories in the PHE based on their explanation types and RL explainability characteristics

Category	Explanation types						RL Explainability Characteristics	
	How	What	Why	Why not	What if	How to	Short-term and long-term consequences	Model information
Feature importance			✓					
Agent behavior	✓		✓	✓			✓	
Textual justification			✓					
Important states and transitions	✓	✓					✓	
Expected outcome			✓	✓			✓	
State abstraction	✓	✓					✓	✓
Agent distillation	✓		✓	✓	✓	✓		
Exploratory analysis	✓		✓					✓
Visual analytics	✓	✓	✓				✓	✓

gradient-based backpropagation approach. However, gradient-based approaches can produce low-quality saliency maps. Therefore, Greydanus et al. (2018) introduce a perturbation-based approach to produce saliency maps instead. They perturb the input with Gaussian blur and measure the impact on the output to determine the importance of the different input parts. Likewise, Iyer et al. (2018) propose a perturbation-based approach but focus on object-level importance instead. They do it by perturbing the objects with the background color found using the template matching method. Puri et al. (2020) notice that Greydanus et al. (2018) and Iyer et al. (2018) aggregate over all actions causing loss of details. They introduce the specific and relevant feature attribution (SAFRA) that focuses on satisfying the two properties specificity and relevance. Consequently, more concise and task-relevant saliency maps are produced that have been shown to help chess players.

The conservation property states that the importance scores at the input sum up to the output value (Bach et al., 2015). According to Huber et al. (2019), existing feature importance methods for RL lack satisfying this property. In response, they extend the layer-wise relevance propagation (LRP) (Bach et al., 2015) to DQN and propose a new argmax rule to produce more concise explanations. Also, they extend the work to dueling Q-network (Wang et al., 2016). In an extended work, Huber et al. (2021) combines this method and a global explanation method (see Sect. 8.1.4) to study the effect of combining explanations. Atrey et al. (2020) review saliency methods for RL and demonstrate that they do not necessarily explain causal relations. In the same work, they conclude that saliency maps should be treated as exploratory information instead of explanatory.

Shi et al. (2022) highlight that applying Mott et al. (2019)'s method to pre-trained agents is impossible since the architecture cannot be modified. Instead, Shi et al. (2022) introduce a self-supervised interpretable network to generate explanations for agents whose architecture can no longer be changed. The method focuses on satisfying two properties, maximum behavior resemblance and minimum region retaining, to generate saliency maps with improved quality. In a later study, Shi et al. (2021b) propose the temporal-spatial causal interpretation model that focuses on understanding long-term behavior and temporal

relationships. The model relies on Granger causality, expressing that causes in the past affect future outcomes.

Besides saliency methods, several studies use SHAP to explain RL agents. SHAP uses a game theoretical approach to explain agents. Rizzo et al. (2019) use SHAP to explain a traffic signal control RL agent. Similarly, Jiang et al. (2022) apply SHAP to understand DRL driving agents. Wang et al. (2020) utilize SHAP in an automatic crane control task since perturbation-based saliency techniques are unsuitable for tabular data. Zhang et al. (2022) apply DeepSHAP to a DRL agent in a power system emergency control task. Liessner et al. (2021) introduce a new SHAP value representation for RL called the RL-SHAP diagram. They experimentally demonstrate the method on a longitudinal control task. Working on a lever manipulation task using a robotic manipulator, Remman and Lekkas (2021) use SHAP to explain RL agents. He et al. (2021) merge the class activation map (Zhou et al., 2016) and SHAP to create a new explanation method applied to a policy controlling an aerial vehicle. Besides the visual explanation, they complement it with textual information. Apart from these applications of Shapley values to RL, Beechey et al. (2023) present the first theoretical analysis of applying Shapley values to explain RL and show that previous uses are incorrect or incomplete.

Borrowing ideas from the supervised learning XAI literature, Weitkamp et al. (2018) and Joo and Kim (2019) apply the gradient-weighted class activation mapping (Selvaraju et al., 2017) to agents assigned to play Atari games. Similarly, Nie et al. (2019) use the gradient-weighted class activation mapping and deconvolutional network (Zeiler & Fergus, 2014) to interpret agents in a swarm robotic system individually. Also drawing from the supervised learning literature, Lim et al. (2021) apply the deep learning important features (Shrikumar et al., 2017) to comprehend an agent trained to control blood glucose. Focusing on the financial application, Guan and Liu (2021) employ integrated gradients (Sundararajan et al., 2017) to explain an agent for portfolio management. Also working on portfolio management, Shi et al. (2021a) use the class activation map to understand portfolio allocation. Kim and Choi (2021) employ several saliency methods, deep Taylor decomposition (Montavon et al., 2017), relative attribution propagation (Nam et al., 2020), and guided backpropagation (Springenberg et al., 2015) to understand a deep visuomotor policy for robotic manipulation. To accommodate negative inputs and outputs, they changed the relevance propagation approach.

Pan et al. (2020) present the explainable generative adversarial imitation learning (xGAIL) framework that produces both local and global explanations. xGAIL aims to explain agents trained using GAIL (Ho & Ermon, 2016). They produce local explanations utilizing a perturbation-based method. At the same time, global explanations are produced by finding observations that maximize the probability of interesting actions.

Methods in this category are closely tied to their counterparts in the IE category. One crucial difference between these two categories of methods is the fidelity. In the IE category, we need to worry less about fidelity since methods are built into the system and used during decision-making. For methods in this category, it is important to test their fidelity since the methods are independent from the decision-making. Moreover, a previous study has shown that feature importance methods producing saliency maps are not always faithful to the model they explain (Adebayo et al., 2018). The methods in this category are fitting for situations where stakeholders want answers to why questions using saliency maps but do not want to retrain the agent.

8.1.2 Agent behavior

This category contains methods to comprehend the agent by characterizing its behavior. For instance, understand what the agent will do in various situations.

Focusing on human-robot collaboration, Hayes and Shah (2017) introduce several methods to answer questions from stakeholders like "When do you do _?", "What do you do when _?" and "Why didn't you do _?" from the stakeholder, as previously mentioned.

By distilling interaction data, Acharya et al. (2020) describe a method to create a conceptual model of agent behavior. This conceptual model conveys the agent's strategy, conditions for their execution, and consequences. Stork et al. (2020) apply various distance measures to compare agents' behaviors. Furthermore, the distance measures are used to find important states and characterize the relationship between reward and behavior.

Many RL explanation methods do not exploit the full MDP formalism and focus on the underlying function approximator. In response, Finkelstein et al. (2021) utilize the full MDP formalism to explain the gap between the agent's behavior and the behavior that the stakeholder anticipated. To explain the gap, they apply abstraction and transformation methods that have previously been used to speed up policy learning.

The methods in this category use the agent's behavior to explain it. The mechanisms to generate these explanations vary greatly between the methods. From Hayes and Shah (2017) answering many types of questions about the agent's behavior to more specific answers, like the gap in stakeholders' mental models (Finkelstein et al., 2021). One downside with some methods in this category is their scalability; for example, Hayes and Shah (2017) require query templates that need manual intervention. Methods in this category are suited for cases where stakeholders want to understand the agent's behavior both locally and globally.

8.1.3 Textual justification

The method in this category aims to extract textual explanations expressed in natural language, similar to methods in Sect. 7.1.3. For example, the explanation "Object ghost and dot have drawn attention of Pacman. The Pacman moves right to eat the dot in the lower right even she is approaching the ghost in the lower right" (Wang et al., 2019b) in the Ms. Pac-Man game. Ehsan et al. (2018) present a method to generate more human-like explanations that translate state-action pairs to natural language expressions. Their method first collects a dataset of state-action pairs and natural language explanations. Then, it uses supervised learning to learn to translate state-action pairs into explanations. According to the authors, this approach offers several advantages, such as being fast to generate explanations and easier to interpret. However, on the downside, the method focuses on explaining how a human would explain the situation and may not reflect the agent's internal reasoning process.

Like its counterpart in the IE category, the method here offers explanations that are more human-like. Furthermore, it offers the flexibility for explanations to be semantically rich. On the downside, it can be more laborious to create these explanations that are only rationalizations.

8.1.4 Important states and transitions

The important states and transitions category contains techniques that explain the agent by showcasing important states and transitions. How the term important is defined depends on the particular method. The motivation for displaying important states and transitions is due to the simplicity of the resulting explanations. Inspecting the agent's behavior in the whole state space is impossible; thus, a trade-off is to review how the agent behaves in a few critical situations. These methods aim for the stakeholders to develop an accurate mental model of the agent's behavior by seeing examples of it in a few situations. Consequently, stakeholders will be able to anticipate how the agent will behave in seen and unseen situations, thus, gaining a global understanding. Besides imparting a global understanding, a few methods in this category try to explain critical situations in an episode after the fact.

Amir and Amir (2018) present an approach to generate a summary of the agent's behavior by showing how it acts in important states. They select important states based on an importance measure that uses the agent's Q-function. More specifically, the more significant the gap between the best and worst actions' Q-values, the higher the importance. In addition, they describe a method to avoid selecting redundant states. In a later work, they provide the entire conceptual framework of explaining agents by using summaries (Amir et al., 2019). They detail the different components of the agent summarization approach, how to evaluate these methods, and position them within related work. The components consist of selecting states, state representation, and the interface to communicate with the stakeholders. In a similar line of work, Huang et al. (2018) and Watkins et al. (2021) present methods to select important states. They aim to help stakeholders to build an accurate mental model of the agent. Thus, being able to determine when it is appropriate to trust the agent. Similar to the other work, Karino et al. (2020) use the Q-function and its variance to select important states. However, in contrast to the others, they additionally explore how these important states can help speed up learning.

Instead of exploiting the agent's output, Huang et al. (2019) use an algorithmic teaching approach to generate summaries. They assume humans do inverse RL and select the states based on their usefulness to learn the reward function. Rather than inverse RL, Lage et al. (2019a, 2019b) propose choosing states that are most helpful to imitating the agent through imitation learning. They experimentally explore the imitation learning and inverse RL approaches. Their results demonstrate the importance of using an appropriate method based on the situation, as no method fits all. Like the aforementioned approaches, the aim is for the stakeholder to develop an accurate mental of the agent. Sequeira and Gervasio (2020) describe a method that gathers interaction data and distills potentially interesting information from it, which they call interestingness elements (Sequeira et al., 2019). They use these interestingness elements to choose states and create agent summaries. Their results show that, on the one hand, more than one summarization approach is needed for a task to convey a complete understanding of agent behavior. On the other hand, too complex explanations can affect stakeholders negatively.

Huber et al. (2021) propose an explanation method that produces both local and global explanations. The method achieves that by integrating a saliency method with the agent summarization technique (Huber et al., 2019; Amir & Amir, 2018). Their results demonstrate that, although saliency maps provide useful information, in most situations,

adding saliency maps as an addition to the agent summary did not significantly improve the understanding.

Previously mentioned agent summarization methods are less suited when comparing two agents. According to Amitai and Amir (2022), agents with different performances can act similarly in important states. To compare two agents, they describe a new agent summarization technique named DISAGREEMENTS. They find states where two agents disagree via simulation. In a later work, Gajcin et al. (2021) argue that the DISAGREEMENTS method only conveys the difference between the agents' performances, but the agents may also differ in their preferred strategies. Therefore, they propose a new method that showcases the agents' gaps in performance and strategy preferences.

Frost et al. (2022) and Watkins et al. (2021) argue that seeing an agent's behavior from training time might be less helpful in the case of a distributional shift. They present a method to find states that instead convey test time behavior. The method achieves this by first defining a prior distribution of test time states. Then it uses an exploration policy to find states matching the prior distribution. Afterward, it runs the original policy from these states to construct the agent summary. Thus, it avoids initializing at out-of-reach states.

Unlike the other methods, Gottesman et al. (2020) propose a framework for off-policy evaluation using an influence function. An influence function is a technique from robust statistics and, in this context, used to determine the importance of transitions with respect to the policy parameters. The function answers what happens to the policy parameters if a transition is upweighted by an infinitesimal amount (Koh & Liang, 2017). The aim is to find important transitions in a batch of data using the influence function. Afterward, show these important transitions to an expert to validate the evaluation. Although not motivated by explainability, the influence function can be integrated into the agent summarization framework.

Besides providing a global understanding of the agent, some methods try to explain agent behavior in an episode. Sakai et al. (2021) determine the sub-goals in episodes and use them to construct the agent summaries that explain the agent's behavior in episodes. Instead of using sub-goals, Guo et al. (2021b) find the important transitions that affect the agent's return in episodes.

Like its counterpart in IE, this category offers global explanations. However, the way states and transitions are found is detached from the agent's learning. Accordingly, methods here do not affect the agent's performance. As noted by Lage et al. (2019a, 2019b), no single method will fit all situations. Thus, the methods here complement rather than outcompete each other. As we have seen throughout this section, the methods themselves have different use cases, from understanding a single agent to comparing two agents. The downside of these methods is that a simulator or access to a buffer of data points is needed to find these states and transitions.

8.1.5 Expected outcome

In this section, we look at methods that explain the outcome of the agent's behavior. For example, the long-term consequences of what kind of states and rewards the agent will observe and receive when taking a specific action.

van der Waa et al. (2018) introduce a method that constructs a policy based on the stakeholder's question and the agent's policy, called the foil policy. The method explains the outcome of the agent's actions alone but also in contrast with the foil policy. The explanation consists of outcomes in terms of actions that will be taken, states that will be encountered,

and rewards that will be received. These are translated into human-understandable concepts, similar to Hayes and Shah (2017). To create explanations that can answer questions from the stakeholder through a mutually understandable vocabulary, Sreedharan et al. (2022) describe a method that leverages a locally approximated model. More specifically, the method explains by referring to the outcome of a specific action and contrasting it to the stakeholder's suggestion or explaining that the suggested action cannot be executed. Differently from the other approaches, Davoodi and Komeili (2021) present a method that highlights features impacting the risk, which tells us something about the outcome. They define risk in terms of states where an episode ends before the expected time or leads to failures.

Cruz et al. (2019) introduce the memory-based explainable RL (MXRL) method. This method explains an action by referring to the probability and time needed to reach the goal. They use interaction data gathered to compute these values. To improve the efficiency of this method, Cruz et al. (2021) propose the learning-based and introspection-based methods that extend MXRL. These two approaches were later extended by Portugal et al. (2022) to accommodate continuous state spaces.

In contrast to similar methods in the IE category, the methods here use transition data to explain what they expect an agent to do. This requires that the agent can be simulated in the environment or that there is already data which can be used to analyze the agent. The methods here are more flexible in comparison to their counterpart in the IE category, as no modification to the agent is required. If a stakeholder wants to understand the long-term behavior from a state for a pretrained agent, methods from this category can be chosen.

8.2 Explanation via representation

Like Sects. 6 and 7.2, the methods in this category explain by referring to the representation rather than generating objects as explanations. The representation ranges from a Markov chain expressing the agent's behavior in the state space to a simplified alternative representation of the agent. Unlike the other categories, the representation is extracted from the agent after training. Thus, the explanation is not necessarily the agent itself.

8.2.1 State abstraction

State abstraction methods cluster states by employing various similarity measures to reduce the state space complexity, which entails trading off between explanation fidelity and complexity. The reduction makes it possible to explain the agent's behavior globally, providing an understanding of the overall agent-environment interaction dynamic. However, making a concise abstraction for large state spaces may be challenging. Nevertheless, getting a local explanation that explains short-term behavior is still more insightful than explaining a single state.

One of the first state abstraction methods for XRL was introduced by Zahavy et al. (2017). They present the semi aggregated MDP (SAMDP) that abstracts across states and actions. The SAMDP is an extension of the semi MDP and aggregated MDP and inherits both of their benefits. To overcome the need for human intervention in the SAMDP approach, Topin and Veloso (2019) present the abstract policy graph (APG) that builds a state space abstraction from interaction data. The APG represents the abstracted state space as a graph where the nodes are abstracted states and edges are actions denoting transitions between them. The authors present the APG Gen method

in the same work to build APGs. McCalmon et al. (2022) point out that the graphs produced by previous state abstraction methods are not interpretable beyond their structure. Moreover, for example, with APG, the graph size can be unbearably large in some situations, such as with stochastic policies. To overcome these hurdles, they describe the comprehensible abstracted policy summaries method. They make abstracted states interpretable by labeling them in human-understandable language. Focusing on visualizing the state space and value function, Nakamura and Shibuya (2020) introduce RL mapper method that extends the mapper (Singh et al., 2007) method. RL mapper visualizes the state space and value function by utilizing topological data analysis.

To express a recurrent NN policy in terms of a Moore machine, Koul et al. (2019) introduce the quantized bottleneck network (QBN) insertion. A QBN is an autoencoder with discretized latent space and can be used to construct discretized input and memory of the policy represented as Moore machines. To reduce the size of Moore machines produced, they employ standard Moore machine minimization techniques to translate them into minimal equivalent Moore machines. However, Danesh et al. (2021) notice that these standard Moore machine minimization techniques cause the resulting Moore machines hard to interpret. This is due to the techniques not considering state semantics. To resolve this issue and effectively reduce these Moore machines, Danesh et al. (2021) describe reductions that do not negatively affect interpretability.

While the other methods expect a trained agent, Bewley et al. (2022) introduce a method applicable to agents under training. They construct the state abstraction using interaction data and an information-theoretic divergence measure and express the abstract state space as a Markov chain. As the agent learns, the Markov chain will change; thus, several Markov chains are constructed, each assigned to a time window. The method can also be used to compare several policies.

The state abstraction methods offer comprehensive global explanations by showing groups of states and transitions between them. The difficulty for these methods is to interpret what the abstract states represent. Showing examples of states in an abstract state might not convey enough nuanced information. It might not be apparent why it is natural for the policy to consider them as similar. Also, McCalmon et al. (2022)'s approach to creating textual summarizations can be labor-intensive. Another issue is choosing the right number of abstract states and what kind of heuristic can be used for that. Too few abstract states can hide information from stakeholders, while too many can overwhelm them.

8.2.2 Agent distillation

The agent distillation category is a collection of methods trying to explain the agent by simplifying its decision-making logic. Specifically, methods in this category do it by treating the agent as an expert and using imitation learning to learn a distilled agent that is easier to interpret. The goal of the distilled agent is to imitate the original agent as well as possible, that is, having a high fidelity. However, beyond having high fidelity, it is also essential to consider when the distilled agent imitates the expert well, since rarely visited states might be less critical.

We often consider decision trees to be interpretable since it is possible to follow the entire reasoning process for a decision. Furthermore, if the decision tree is small, it is even globally interpretable instead of being locally only. Many methods use decision trees to

imitate, frequently with modifications to solve previous limitations and adapt to new use cases. We refer to them collectively as tree-based agent distillation methods. Liu et al. (2018) introduce the linear model u-tree (LMUT) that extends the u-tree method with linear models in the leaf nodes for increased flexibility. LMUT aims to approximate the Q-function of the agent. Coppens et al. (2019) describe the soft decision tree method (Frosst & Hinton, 2017) applied on Mario AI benchmark (Karakovskiy & Togelius, 2012). The expert we try to imitate often supplies both the action and Q-values. Bastani et al. (2018) utilize this fact and improve the dataset aggregation (DAgger) algorithm and propose Q-DAgger, which results in less complex distilled agents. The Q-DAgger method improves DAgger by prioritizing and sampling state-action pairs based on the Q-values. In addition, they propose the verifiability via iterative policy extraction (VIPER) method to extract tree-based agents leveraging Q-DAgger and show how these tree-based agents extracted can be verified. VIPER has been very influential for methods in this category and several other methods extend and improve upon it (Schmidt et al., 2021; Jayawardana et al., 2021; Zhu et al., 2021; Jhunjhunwala et al., 2020). Roth et al. (2021) propose an agent distillation method that extends VIPER (Bastani et al., 2018). They argue that previous tree-based methods are uninterpretable (Frosst & Hinton, 2017; Gupta et al., 2015) and that no domain-specific tree modifications have previously been proposed. Specifically, after extracting a decision tree policy from a DRL policy, the decision tree policy is improved by modifying it, such as adding or changing nodes. These modifications focus on finding and fixing unwanted behavior from the policy in navigation tasks, such as oscillating action selection. Besides VIPER and methods that extend it, numerous other studies utilize tree-based imitation agents to enable interpretability (Gjærum et al., 2021, 2021; Ghosh et al., 2021; Dhebar et al., 2022; Vasic et al., 2022; Dai et al., 2022b; Bewley & Lawry, 2021), with some focusing on first transforming the input to their interpretable counterpart (Bewley et al., 2020; Sieusahai & Guzdial, 2021; Liu et al., 2021).

Motivated by the fact that many agents are hard to understand and verify, Verma et al. (2018) propose the programmatically interpretable RL (PIRL) framework. With PIRL, learning agents represented as programs become possible, which uses an expert agent to guide the learning process. In a later work, Verma et al. (2019) describe the imitation-projected programmatic RL (PROPEL), a new method to learn program-based agents. PIRL and PROPEL are later extended by Larsen and Schmidt (2021) to accommodate a different program space. Finally, many of these methods are summarized in Bastani et al. (2020).

Rules, such as if-then statements, are another function representation used to express distilled agents. Nageshrao et al. (2019), for example, seek to obtain a distilled rule-based agent leveraging fuzzy logic trained using the evolving Takagi-Sugeno method (Angelov & Filev, 2004). Another approach proposed by Soares et al. (2021) first clusters states before distilling the agent, thus reducing the complexity of the resulting distilled agent. Skirzynski et al. (2021) seek to improve human decision-making by first distilling an agent into simple rules. These simple rules are converted into flowcharts that can assist humans in making better decisions. Honda and Hagiwara (2022) express states and actions using first-order logic and extract a distilled rule-based agent.

Driven by the human aspect of explainability, Madumal et al. (2020) describe the action influence model that builds on the structural causal model (Halpern & Pearl, 2005) by extending it with actions. They learn the actions' causal effect during learning by constructing the graph's structure beforehand. Using the graph to explain and create hypothetical scenarios, they generate why and contrastive explanations. Also focusing

on the human aspect of explainability, Mitsopoulos et al. (2021) describe utilizing cognitive models to understand agent behavior.

Focusing on traffic signal control and explainability, Ault et al. (2020) describe the regulatable precedence function as a representation for the distilled agent. A regulatable precedence function is a function that is monotonic in the state variables. They introduce several modifications of the DQN approach to express and learn regulatable precedence function agents. Working on the same problem, Wollenstein-Betech et al. (2020) utilize knowledge compilation techniques to comprehend DRL agents. Zhang et al. (2020a) present an agent distillation technique built upon the evolutionary feature synthesis regression algorithm (Arnaldo et al., 2015).

Hüyük et al. (2021) focus on understanding expert decision-making behavior rather than the behavior of an agent. To that end, they describe the model-based Bayesian method for interpretable policy learning (INTERPOLE). INTERPOLE approximates decision dynamics and boundaries and aims to satisfy three characteristics: (1) inherently interpretable, (2) partial observability accommodation, and (3) completely offline operation, which are needed in the healthcare setting. Focusing on the same problem, Pace et al. (2022) introduce the policy extraction through decision trees (POETREE) framework that builds probabilistic tree policies with recurrent structure. POETREE is designed to handle partial observability and offline training for the same reason as INTERPOLE.

Xie et al. (2022) use adversarial inverse RL to distill the reward function where the discriminator is represented using the logistic regression model. The resulting reward function provides a global explanation due to its simple functional form. After training, the function can be analyzed to understand how the agent values different situations.

The agent distillation category offers comprehensive global explanations that explain the whole decision-making process. However, since they only distill the input–output relation of the original agent, they might not explain the true underlying decision-making process. Instead, they give a plausible explanation that disregards the original agent's internal logic. Another issue is the complexity of the distilled agent. If the distilled agent is too complex, such as decision trees with large depth, they may not be as useful to a stakeholder. Thus, fidelity and accuracy often need to be traded in these models.

8.3 Explanation via inspection

The explanation via inspection category introduces methods applied to RL agents after training to extract understanding. For example, a new user interface dashboard that lets a stakeholder freely explore different scenarios to understand the agent or analyze the agent using various dimension reduction techniques. Like Sect. 7.3, the explanations extracted are open-ended and require human analysis to extract insight. However, in contrast to Sect. 7.3, the methods do not modify the agent or propose a custom architecture to extract explanations more easily.

8.3.1 Exploratory analysis

The exploratory analysis category contains various methods to extract knowledge about the agent's behavior. The methods range from dimension reduction techniques to applying several existing XAI methods to extract insight.

Sequeira et al. (2019) propose a new approach to understanding the task and agent behavior by analyzing the interaction data and deriving various insights. For example, how often does the agent encounter different states, or how often does the agent execute the same action in a state. The interaction data is gathered from the agent's past interaction with the environment. The method's analysis is independent of the environment and the underlying RL algorithm. Similarly, Ullauri et al. (2022) use interaction data to understand the agent. Their method is also model agnostic, like the previously mentioned one.

Druce et al. (2019) presents two metrics that can be used to understand the agent's generalization ability. Also, they present a method to understand how agents will behave in modified states via state intervention conditioned on the current state. The metrics and state intervention information are communicated in a new user interface described in the same study. Hilton et al. (2020) utilize XAI methods to understand an agent that they trained explicitly in the CoinRun (Cobbe et al., 2019) environment. These XAI methods include applying several feature importance methods and a dimension reduction technique. Similarly, utilizing dimension reduction, Agrawal and McComb (2022) seek to understand the exploration process of agents tasked with designing cyber-physical systems. They say that the information about the exploration process can be leveraged to choose algorithms for designing these systems.

Løver et al. (2021) apply several XAI methods to understand how a docking agent trained using DRL works. Specifically, they use (1) SHAP, (2) local interpretable model-agnostic explanations (LIME) (Ribeiro et al., 2016), and (3) LMT. Focusing on heating, ventilation, and air conditioning energy controller, Kotevska et al. (2020) introduce a comprehensive framework to understand these agents trained using DRL. The framework uses existing XAI methods to extract local (i.e., LIME) and global explanations (i.e., PDP (Friedman, 2001) and ICE (Goldstein et al., 2015)). Moreover, it gathers interaction data of the controller. These two sources of information are analyzed and visualized to understand the agent. Dai et al. (2022a) aim to comprehend how, in simulated robotics tasks, domain randomization affects DRL agents. To gain insight, they also apply XAI methods, test the agent in different environments, and use out-of-distribution generalization tests. Pankiewicz and Kowalczyk (2022) present to understand RL agents using a combination of techniques. These techniques include Integrated Gradients (Sundararajan et al., 2017), analysis of variance, hypothesis tests, and examining correlation on state-action data generated by the policy.

Russell and Santos (2019) aim to understand better the reward function an agent tries to optimize. They use LIME to create local explanations and decision trees to imitate the reward function to get global explanations. Likewise, using local explanations, specifically saliency maps, Michaud et al. (2020) use it to understand the reward function and how well it aligns with the stakeholder's preferred behavior. Similarly, utilizing saliency maps, Guo et al. (2021a) seeks to understand the relationship between human and machine attention. To accomplish that, they ask two questions. First, "how similar are the visual representations learned by RL agents and humans when performing the same task?". And second, "how do similarities and differences in these learned representations explain RL agents' performance on these tasks?".

This category is a collection of methods that can be used to explain RL agents. They are a mix of methods from the supervised XAI literature and offer a view of how methods together can support understanding RL agents. To aggregate insights from several methods, human intervention is needed. Furthermore, they do not explain short-term and long-term consequences, as they use methods that are mainly designed for supervised learning.

8.3.2 Visual analytics

Visual analytics systems provide interactive visualizations and analysis tools to better understand RL agents. They aim to help stakeholders better understand agents through insights into the agent's behavior and its internal representation, including how they change during training. Various sources of information are gathered and processed to create these visualizations. For instance, how the return per episode evolves or which actions are executed in a state throughout the training. Several visual analytics systems have been created for unsupervised learning (e.g., GANViz (Wang et al., 2018)) and supervised learning (e.g., CNNVis (Liu et al., 2017), RNNVis (Ming et al., 2017), and LSTMVis (Strobelt et al., 2018)). This section overviews the visual analytics method designed explicitly for RL.

Wang et al. (2019a) describe DQNViz, the first visual analytics system designed for RL specifically. DQNViz aims to help developers understand, debug, and improve DQN models. Their system is designed and evaluated together with deep learning experts. It provides different views into a DQN model, such as how the training evolves (e.g., Q-value change throughout training) or how the DQN performs in a single episode (e.g., action distribution per episode). Likewise, Seng et al. (2021) introduce a visual analytics system made to understand DQN models but does differ by providing insights into other aspects not covered by DQNViz. Jaunet et al. (2020) point out that previously proposed visual analytics systems cannot interpret agents with memory that are designed for environments with partial observability. They, therefore, propose DRLViz, a new visual analytics system focusing on understanding agents with memory and analyzing the memory in detail, such as understanding its role. The system was created with the help of experts and evaluated in the ViZDoom environment (Kempka et al., 2016). Another visual analytics system focusing on a different aspect of the agent is DynamicsExplorer (He et al., 2020). DynamicsExplorer aims to understand how trained agents are affected by the distribution shift of the environment. They test DynamicsExplorer in the marble maze game, a robotics control task (van Baar et al., 2019).

In contrast to the previously mentioned systems, Wang et al. (2021b) introduce DRLIVE, which focuses on being applicable to all RNN-based models. Furthermore, it seeks to be applicable to multiple game settings rather than a few selected. Besides these aforementioned systems, there are other visual analytics systems that aim to help experts to better understand RL agents (Mishra et al., 2022; Cheng et al., 2022).

Visual analytics systems provide comprehensive tools for a stakeholder to analyze an agent. However, they are often tailored to specific agents such as the DQN or specific environments like Atari games. They are also more suited for users with in-depth domain knowledge and RL knowledge as explanations are more open-ended and technical. Thus, they need human analysis to draw insights. Nevertheless, visual analytics provide comprehensive explanations that clarify all parts of an agent. This is especially useful for debugging and verification before deployment.

9 Discussion

In this section, we give a high-level analysis of the trends within XRL and recommend some methods for practitioners to use to explain RL agents that have stood the test of time. We call these methods foundational, as they have inspired many works that come after via

extensions and as baseline methods in many experiments. Finally, we look at some future directions that forthcoming XRL work should focus on.

9.1 Trends

Lately, XRL has focused on the PHE category, as seen in Table 5. Less work is done within the IE category, likely due to being more challenging. For example, adding a built-in explainability mechanism is undesirable if it negatively affects the agent's performance. This issue can be avoided by using PHE methods. From the perspective of this literature review, the IA category is receiving the least attention. However, this is because only studies mainly driven by interpretability are included and not by, for instance, generalization and sample efficiency.

We observe that feature importance methods from both IE and PHE are most researched among XRL methods. Other trending XRL methods are agent distillation methods that aim to explain the agent via distilled models such as decision trees. Aside from those three categories of methods, the categories expected outcome in IE and important states and transitions in PHE are popular in XRL research. State abstraction methods and visual analytics systems in PHE are less popular than previously mentioned categories but still important. Although exploratory analysis in PHE looks popular, it is much more diverse and is not one of the trending categories within XRL. Overall, the works within XRL are diverse; the

Table 5 Overview of the number of studies published each year for each category

	Category	Year					
		2017	2018	2019	2020	2021	2022[1]
	Interpretable agent	1	2	1	3	8	3
Intrinsic explainability	Feature importance	1	1	3	3	3	5
	Intended behavior	2				1	2
	Textual justification		1	1		1	1
	Important states and transitions		2			1	1
	Expected outcome		1	3	1	2	3
	Generative modeling			2	1	1	
	State abstraction				1	2	
	Task decomposition		1	2	1	2	2
	Reward function			2	1	1	1
	Exploratory analysis			1	1		1
Post hoc explainability	Feature importance	1	3	5	4	9	3
	Agent behavior	1			2	2	
	Textual justification		1				
	Important states and transitions		2	4	4	5	2
	Expected outcome		1	1		3	2
	State abstraction	1		2	2	1	2
	Agent distillation		3	3	7	15	6
	Exploratory analysis	1		3	5	2	4
	Visual analytics			1	2	2	2

[1] The number of studies for 2022 does not include the entire year

focus is spread across areas like feature importance, important states and transitions, agent distillation, expected outcome, state abstraction, and visual analytics.

9.2 Recommendations

This section recommends methods that are suited for different stakeholder questions. There is no single method suitable for all stakeholders' needs. Each method has its use cases, strengths, and weaknesses. For instance, feature importance methods are limited to explaining where an agent looks in the input space, but are easy to convey to stakeholders. They supply us with the ability to answer one specific stakeholder question, namely, "why did the agent do _?" regarding where the agent is looking. Methods that can answer stakeholder questions can be found via Tables 2, 3 and 4. Our focus here when recommending methods is based on whether a method has stood the test of time. The methods that have been extended a few times are more robust and proven to work in accordance with their experiments consistently. Also, one could argue they are considered more useful methods by researchers since more resources are used to study them. For example, this is apparent in Sect. 8.1.5 where most studies extend or take inspiration (or both) from Juozapaitis et al. (2019). Newly published studies are more likely to have stronger experimental results but have not been independently verified by other researchers.

Methods from important states and transitions, state abstraction, and agent distillation categories can be used to answer how questions. For example, HIGHLIGHTS (Amir & Amir, 2018) answers how questions and is a popular method many studies have extended. Stakeholders can use Amitai and Amir (2022)'s method if they want to compare two different agents. If stakeholders need more detailed how explanations, VIPER (Bastani et al., 2018) from the agent distillation category can be used and is used as a baseline for many studies. Likewise, there is the interpretable agent category that can provide detailed how explanations, such as Silva et al. (2020), Trivedi et al. (2021). Visual analytics systems like Wang et al. (2019a) offer comprehensive insights into an agent and can be utilized to answer how questions and many other questions. However, visual analytics systems should be reserved for situations where comprehensive explanations are needed, as they are more open-ended. Furthermore, when using visual analytics systems, stakeholders need expertise in RL as many technical terms are used in these explanations.

When it comes to human-robot collaboration that requires an agent to describe what it will do in real-time, methods from intended behavior are suitable. For example Fukuchi et al. (2017a, 2017b) provide explanations that are easy to digest but lack detail. For more comprehensive explanations, state abstraction methods like Topin and Veloso (2019), McCalmon et al. (2022) offer explanations that describe what the agent will do via a Markov chain. Another popular method answering what questions is Hayes and Shah (2017)'s method which others take inspiration from. Their method is more suitable when the stakeholder wants answers to questions such as, "when do you do _?" and "what do you do when _?" in natural language.

The why question is answered by many methods with varying details. The methods in the feature importance categories from both IE and PHE answer why questions. Methods range from those developed with RL in mind to others adapted from the supervised learning XAI literature. Puri et al. (2020) is specifically designed for RL and is a result of addressing weaknesses of many previous feature importance methods. If a stakeholder wants to understand what and where the agent is looking, Mott et al. (2019)'s method can be used. Juozapaitis et al. (2019)'s method can be used if stakeholders want explanations focusing on the

reward instead of the input. Like many of the other questions, methods from the interpretable agent category can provide detailed why explanations. The same applies to the agent distillation category, for example, Verma et al. (2018), Bastani et al. (2018). Methods from the interpretable agent, agent distillation, and feature importance categories are unable to answer why concerning long-term consequences. Thus, if stakeholders want explanations containing sequential information, the reward decomposition method by Juozapaitis et al. (2019) or Yau et al. (2020)'s method that also focuses on the outcome should be utilized.

The why not questions can be answered by the expected outcome category by contrasting the outcome of actions. As an example, Juozapaitis et al. (2019)'s method answers why not questions. Other methods like Yau et al. (2020)'s method also answer why not questions but uses time steps instead of rewards. Depending on the functional form of the distilled agent, the agent distillation category can answer these questions. For example, using VIPER (Bastani et al., 2018), a stakeholder can traverse the decision tree to answer the why not question. Additionally, methods from interpretable agents can be used.

The what if and how to questions are closely connected. Generative modeling approaches like Olson et al. (2021), Rupprecht et al. (2020) are two ways to answer these counterfactual questions. Madumal et al. (2020)'s method is another approach to answering counterfactual questions via causality. There are also tree-based methods like Liu et al. (2018), Bastani et al. (2018) answering these questions. These questions can be answered by inspecting paths in the decision trees. Similarly, like most questions, methods from the interpretable agent category can be utilized.

9.3 Future directions

We have seen numerous XRL studies throughout this review. However, there are still problems that remain open. Here, we highlight essential and fruitful avenues for future studies. More specially, we highlight five directions: (1) state the intent, (2) more research on interpretable agents in the context of XRL, (3) focus on RL specific aspects, (4) satisfying explanation properties, and (5) better evaluation.

Intent

Numerous review studies briefly explain why we need explainability. However, future studies should also state what kind of explainability needs their method specifically aims to satisfy (e.g., debugging or extracting novel insight from the domain). Moreover, they should describe the intended stakeholders and what kind of stakeholder questions the method seeks to answer (e.g., "how can I get the agent to do _?"). Finally, many studies evaluate the methods in toy environments or games, but the methods might be suitable or intended for other tasks. We urge researchers to state the intent to make it easier for stakeholders to find suitable methods. Accordingly, we believe XRL studies will have a broader adoption by other stakeholders besides researchers.

Interpretable agent

As pointed out in the supervised learning literature (Burkart & Huber, 2021; Rudin, 2019), are black box agents required, or can we design inherently interpretable agents? In situations for already deployed agents, post hoc explainability is desirable. Nevertheless, interpretable agents are still crucial to XRL since they truly reflect an agent's behavior rather than creating plausible explanations. Moreover, they have several advantages, such as being sample efficient and better at generalizing and are less researched in the context of XRL. Therefore, we recommend future research on XRL to focus more on interpretable agents. While much research from other related research areas can fit into

this category, we believe more XRL specific research with evaluations targeting interpretability explicitly is needed.

RL specific aspects

Many reviewed RL explanation methods borrow ideas from supervised learning (e.g., saliency methods) and explain the actions using only the immediate context. However, RL differs from supervised learning, and fewer studies try explaining characteristics unique to sequential decision-making. Also, RL can be model-free and model-based. If we use model-based RL, how can we explain the model? For example, why does the model predict s' as the future state and not some other state $\hat{s}$? How can we create an explanation that coherently explains the agent and model simultaneously? Furthermore, in the case of partial observability where a policy depends on the history and not just the current state. How can we explain to the stakeholders when the agent is no longer just reactive but also has an internal state? Based on these open problems, we recommend future studies to focus on developing methods leveraging and explaining these unique characteristics so that we can fully grasp the reasoning process of these agents.

Explanation properties

The authors outline several explanation properties in Chapter 33 of Murphy et al. (2023). When proposing new XRL methods, more focus should be placed on covering various explanation properties because different situations require different properties. For instance, few reviewed studies have explicitly focused on explaining time-critical decisions. In time-critical decision-making, we need explanations that generate quickly, are easily understandable, and do not necessarily cover all the reasons for a decision. In contrast, we might want complete explanations in situations without time constraints. In short, we should examine which explanation properties existing methods fulfill and work towards covering others to accommodate different situations and use cases that various stakeholders have.

XRL evaluation

XAI and XRL are large ecosystems with many components. Apart from the explanation method and the agent, there are other elements like the need for explainability, stakeholders, and explanation properties. A single evaluation without specifying the method's setup is dissatisfactory. Additionally, developing a holistic evaluation to cover all aspects is impossible. As pointed out by previous studies (Puiutta & Veith, 2020; Heuillet et al., 2021) and Section Appendix A, most studies evaluate methods using functionally-grounded evaluations. However, functionally-grounded evaluations do not consider the setup. We must carry out evaluations with respect to the task and the stakeholder rather than evaluating without specifications. Although costly, we recommend doing more human-grounded evaluations in future studies and, if possible, application-grounded evaluations.

Besides the aforementioned issues on evaluation, an equally important problem is the lack of standardized user studies. User studies without some standardization make it challenging to compare different studies. Thus, when researchers develop new methods, it is hard to know how the different methods compare and which to use. Consequently, we propose future studies to work toward more comparable standardized user studies.

10 Conclusions

We have systematically searched five electronic databases and reviewed 189 state-of-the-art XRL studies published within the last five years. Moreover, we have systematically

obtained ten existing XRL literature reviews and compared them to this review by showing how it is systematic, more comprehensive, and updated. This review proposed a new taxonomy that reflects the XRL studies reviewed and divided methods into three main categories: (1) interpretable agent, (2) intrinsic explainability, and (3) post hoc explainability. Also, the taxonomy organizes the studies based on how the explanations are communicated to stakeholders: (1) via generation, (2) via representation, or (3) via inspection. Each included study in this literature review was outlined, extracted for details, and organized into the taxonomy. Additionally, we overviewed which stakeholder questions can be satisfied by the different taxonomy categories. For example, if a category of methods can answer questions like, "how does the agent work?" and "why did the agent do _?". Afterward, we outline trends in XRL and make recommendations for XRL methods based on stakeholder questions. Finally, this review highlighted five future directions in this fast-growing field that tackle challenges hindering broader RL adoption. We intend to unify the XRL field with this review. Moreover, we hope this review can be a resource that helps stakeholders become acquainted with the state-of-the-art XRL and find suitable methods to answer their questions. Lastly, we seek to help researchers find research gaps with this review.

Appendix A Overview of XRL studies

In this appendix, we provide a concise overview of the reviewed studies. Table 6 presents the interpretable agent (IA) studies, Table 7 introduces the intrinsic explainability (IE) studies, and Table 8 overviews the post hoc explainability (PHE) studies. We

Table 6 Overview of interpretable agent studies. Figure 6 lists the acronyms for the category column. US refers to if a user study has been performed, and C refers to code being open-sourced

References	Category	Scope	Focus	Env(s) / Task(s)	US	C
Hein et al. (2017b)	RB	■	⇈ ⇊	• Mountain Car • Cart-Pole • Cart-Pole Swing-Up	✗	✗
Hein et al. (2018a)	RB	■	⇈ ⇊	• Cart-Pole Swing-Up • Industrial Benchmark (Hein et al., 2017a)	✗	✗
Huang et al. (2020)	RB	■	⇈	• Mountain Car • Continuous Gridworld, • Pendulum Position, • Tank Level Contro	✗	✗
Likmeta et al. (2020)	RB	■	⇈ ⇊	• Autonomous driving with scenarios: highway (lane change) and urban (crossroads and roundabout) using the simulation of urban mobility (López et al., 2018) simulator	✗	✗
Hein et al. (2018b)	ME	■	⇈ ⇊	• Mountain Car • Cart-Pole • Industrial Benchmark	✗	✗
Kubalík et al. (2021)	ME	■	⇈ ⇉	• Friction Compensation • 1-DOF and 2-DOF Pendulum Swing-Up • Magnetic Manipulation	✗	✗

Table 6 (continued)

References	Category	Scope	Focus	Env(s) / Task(s)	US	C
Landajuela et al. (2021)	ME	■	⇈	• Cart-Pole • Mountain Car • Pendulum • Inverted Double Pendulum • Inverted Pendulum Swing-Up • Lunar Lander • Hopper • Bipedal Walker	✗	✓
Videau et al. (2022)	ME	■	⇈	• Cart-Pole • Acrobot • Mountain Car • Pendulum • Inverted Double Pendulum • Inverted Pendulum Swing-Up • Hopper • Lunar Lander, • BipedalWalker • BipedalWalkerHardcore	✗	✓
Jiang and Luo (2019)	LB	■	⇈ ⇉	• Block Manipulation • Cliff Walking	✗	✓
Gorji et al. (2021)	LB	■	⇉	• Four Gridworld configurations	✗	✓
Zhang et al. (2021b)	LB	■	⇈ ⇉	• Block Manipulation • Car Avoiding	✗	✗
Kimura et al. (2021)	LB	■	⇈ ⇉	• TextWorld	✗	✗
Silva et al. (2020)	TB	■	⇈	• Cart-Pole • Lunar Lander • Simulated Wildfire Tracking • StarCraft II Learning Environment	✓	✗
Topin et al. (2021)	TB	■	⇈	• Cart-Pole • PrereqWorld • PothleWorld	✗	✗
Custode and Iacca (2021)	TB	■	⇈ ⇉	• Mountain Car • Lunar Lander	✗	✓
Trivedi et al. (2021)	PB	■	⇈ ⇉	• Karel domain (Gridworld)	✓	✓
Qiu and Zhu (2022)	PB	■	⇈ ⇉	• Several MuJoCo environments	✗	✓
Cao et al. (2022)	PB	■	⇈ ⇉	• MiniGrid	✗	✓

characterize each study by its subcategory, scope, focus, environment or task (or both), if it has performed a user study, and if it has open source code. We illustrate the meaning of scope in Fig. 11 and focus in Fig. 12.

Table 7 Overview on intrinsic explainability studies

References	Category	Scope	Focus	Env(s) / Task(s)	US	C
Kim and Canny (2017)	FI	□	⇈ ⇊	Driving dataset • Comma.ai (Santana & Hotz, 2016) • Udacity • HCE at Berkeley	✗	✗
Goel et al. (2018)	FI	□	⇈ ⇉	• Arcade Learning Environment (ALE)	✗	✓
Mott et al. (2019)	FI	□	⇈	• ALE	✗	✗
Nikulin et al. (2019)	FI	□	⇈	• ALE • Atari-HEAD dataset	✗	✓
Zambaldi et al. (2019)	FI	□	⇈ ⇉	• StarCraft II Learning Environment • Navigation and planning with "Box-World"	✗	✗
Cultrera et al. (2020)	FI	□	⇈ ⇊	• Autonomous driving datasets: Codevilla et al. (2018) • CARLA Simulator (Dosovitskiy et al., 2017)	✗	✗
Josef and Degani (2020)	FI	□	⇈ ⇊	• Generated terrain • Gazebo (Koenig & Howard, 2004)	✗	✗
Tang et al. (2020)	FI	□	⇈ ⇉	• Car Racing • Doom TakeCover	✗	✓
Bao et al. (2021)	FI	□	⇈ ⇊	Traffic accident datasets • DADA-2000 (Fang et al., 2019) • DAD (Chan et al., 2016)	✗	✓
Zhang et al. (2021c)	FI	□	⇈ ⇉	• Pendulum with added noise • MuJoCo environments • TORCS (Wymann et al., 2014)	✗	✓
Itaya et al. (2021)	FI	□	⇈	• ALE	✗	✗
Feit et al. (2022)	EO	◪ ■	⇈	• Simulator of Web Infrastructure and Management	✗	✓
Liu et al. (2022)	FI	□	⇈ ⇊	• ALE	✗	✗
Wang et al. (2022)	FI	□	⇈ ⇉	• ALE	✗	✗
Wei et al. (2022)	FI	□	⇈ ⇉	• Mountain Car • Pendulum • Cart-Pole • Acrobot	✗	✗

Table 7 (continued)

References	Category	Scope	Focus	Env(s) / Task(s)	US	C
Dai et al. (2022c)	FI	□	⇈	• Toy environment • ALE	✗	✗
Kim et al. (2022)	FI	□	⇈ ⇉	• 2D navigation with moving obstacles	✗	✗
Fukuchi et al. (2017a)	IB	◪	⇈ ⇊	• Lunar Lander	✗	✗
Fukuchi et al. (2017b)	IB	◪	⇈ ⇊	• Lunar Lander	✓	✗
Wang et al. (2021a)	IB	□	⇊ ⇈	• Highway on-ramp driving scenarios with CARLA Simulator	✗	✗
Chen et al. (2022)	IB	□	⇊ ⇈	• Urban driving scenarios such as intersections and roundabouts with CARLA Simulator	✗	✓
Fukuchi et al. (2022)	IB	□	⇈	• Lunar Lander	✓	✓
Kim et al. (2018)	TJ	□	⇈ ⇊	• Berkeley DeepDrive eXplanation Dataset. An extension of the Berkeley Deep Drive (Xu et al., 2017) dataset with textual justifications	✓	✓
Wang et al. (2019b)	TJ	□	⇈	• Ms. Pac-Man	✗	✗
Cruz and Igarashi (2021)	TJ	◪	⇈ ⇊	• Mario AI	✓	✗
Ben-Younes et al. (2022)	TJ	□	⇈ ⇊	• Honda Deep Drive (Ramanishka et al., 2018) dataset: video frames with cause labels • Berkeley deep drive explanation (Kim et al., 2018) dataset: explanations with natural language justifications	✗	✓
Dao et al. (2018)	IST	■	⇈	• Visual Maze (Gridworld) • ALE	✗	✗
Mishra et al. (2018)	IST	■	⇈	• Gridworld	✗	✗
Dao et al. (2021)	IST	■	⇈	• Breakout • arcade learning environment	✗	✗
Jacq et al. (2022)	IST	■	⇈ ⇉	• Gridworlds • ALE	✗	✗
Erwig et al. (2018)	EO	◪	⇈ ⇊	• Collecting fruits in Gridworld	✗	✗

Table 7 (continued)

References	Category	Scope	Focus	Env(s) / Task(s)	US	C
Juozapaitis et al. (2019)	EO	◪	⇈	• Cliffworld • Lunar Lander	✗	✗
Anderson et al. (2019)	EO	◪	⇈	• Self-made tank game, inspired by real-time strategy game, for experiments	✓	✓
Pan et al. (2019)	EO	◪	⇈ ⇉ ⇊	• Flappy Bird • Autonomous driving environments with varying task difficulty: TORCS (Wymann et al., 2014), CARLA Simulator and Grand Theft Auto V	✗	✓
Yau et al. (2020)	EO	◪	⇈	• Cart-Pole • Blackjack • Taxi (Gridworld)	✗	✓
Iucci et al. (2021)	EO PHE-AB	■ ◪ □	⇈	• Automated warehouse scenarios modeled in V-REP (Rohmer et al., 2013) simulator	✗	✓
Lin et al. (2021)	EO	◪	⇈	• Cart-pole • Lunar Lander • Tug of War	✗	✓
Rietz et al. (2022)	EO TD	◪	⇈	• 2D navigation with a static obstacle	✗	✓
Olson et al. (2019)	GM	□	⇈	• ALE	✓	✗
Yang et al. (2019)	GM	□	⇈	• dSprites (Matthey et al., 2017) • ALE • MuJoCo: Walker2d, Hopper, Half-Cheetah and Swimmer	✗	✗
Rupprecht et al. (2020)	GM	■	⇈	• ALE • Driving simulation	✗	✗
Olson et al. (2021)	GM	□	⇈	• ALE	✓	✓
Bougie and Ichise (2020)	SA	■	⇈ ⇉	• Trading task from real stock market data • Visual navigation	✗	✗
Zhang et al. (2021a)	SA	■	⇊	• Hypotension management data from MIMIC-III (Johnson et al., 2016a)	✗	✗

Table 7 (continued)

References	Category	Scope	Focus	Env(s) / Task(s)	US	C
Akrour et al. (2021)	SA	■	⇈	• PyBullet (Coumans & Bai, 2016–2021) • MuJoCo (Todorov et al., 2012) environments	✗	✓
Shu et al. (2018)	TD	◪	⇉	• Object manipulation tasks in Minecraft (Johnson et al., 2016b)	✗	✗
Beyret et al. (2019)	TD	◪	⇈	• FetchPush, FetchPickAndPlace and HandManipulateBlock in MuJoCo	✗	✗
Lyu et al. (2019)	TD	◪	⇉⇈	• Taxi domain (Gridworld)• Montezuma's Revenge	✗	✓
Wu et al. (2020)	TD	■	⇉	• Ant navigating mazes, Stacker arm picking up and placing different boxes in MuJoCo	✗	✗
Hasanbeig et al. (2021)	TD	◪	⇉	• Minecraft • Two mars-rover benchmarks • Robot-surve • slp-easy • slp-hard • frozen-lake • Montezuma's Revenge	✗	✓
Ye and Yang (2021)	TD	◪	⇉	• Robot object search in House3D simulation environment	✗	✓
Gangopadhyay et al. (2022)	TD	■	⇊⇈	• Different driving tasks • Pre-crash scenarios in CARLA Simulator	✗	✓
Tabrez et al. (2019)	RF	◪	⇈	• Color-based variate of Sudoku with a Rethink Robotics Sawyer manufacturing robot	✓	✗
Li et al. (2019b)	RF	■	⇉⇈	• Two robotic manipulators to perform a hot dog cooking and serving task	✗	✗
Bautista-Montesano et al. (2020)	RF	■	⇊⇈	• Racing tracks by Amazon Web Services: Baadal Track, SOLA Speedway, the 2020 DeepRacer Championship Track, the AWS Summit Raceway	✗	✓
Bica et al. (2021)	RF	■◪	⇊⇈	• Healthcare: simulated environment, ICU dataset from MIMIC-III	✗	✗

Table 7 (continued)

References	Category	Scope	Focus	Env(s) / Task(s)	US	C
Bewley and Lécué (2022)	RF	■	⇉ ↑↑	• Pendulum • RoboCar • Lunar Lander • Food Lava	✓	✗
Annasamy and Sycara (2019)	EI	■	↑↑ ⇉	• ALE	✗	✓
Kuramoto et al. (2020)	EI	■	↑↑	• Collision avoidance learning of a real robot in various physical environments in real time	✗	✗
Tylkin et al. (2022)	EI	■	↑↑	• Canyon Run simulation: aircraft control simulated • Drone Dodgeball: quadcopter control, first simulation than later transferred to a real system	✗	✗
Terra et al. (2022)	EO	□◪	↑↑	• Remote electrical antenna tilt control	✗	✗

Figure 8 lists the acronyms for the category column. US refers to if a user study has been performed, and C refers to code being open-sourced.

Table 8 Overview on post hoc explainability studies

References	Category	Scope	Focus	Env(s) / Task(s)	US	C
Zahavy et al. (2017) (based on Zahavy et al. (2016), Ben-Zrihem et al. (2016), Baram et al. (2017))	FI EA SA	■ ◪ □	↑↑	ALE	✗	✓
Weitkamp et al. (2018)	FI	□	↑↑	ALE	✗	✗
Greydanus et al. (2018)	FI	□	↑↑	ALE	✓	✓
Iyer et al. (2018)	FI	□	↑↑	ALE	✓	✗
Nie et al. (2019)	FI	□	⇉ ↑↑	Navigation	✗	✗
Huber et al. (2019)	FI	□	↑↑	ALE	✗	✗
Joo and Kim (2019)	FI	□	↑↑	ALE	✗	✗
Rizzo et al. (2019)	FI	□	↑↑ ↓↓	Traffic signal control	✗	✗
Pan et al. (2020)	FI EA	■ □	↑↑	Real-world taxi driving	✗	✓
Puri et al. (2020)	FI	□	↑↑	Chess Go ALE	✓	✓
Wang et al. (2020)	FI	■ □	↑↑	Crane control	✗	✗
Huber et al. (2021)	FI IST	■ □	↑↑	ALE	✓	✓
Liessner et al. (2021)	FI	■ □	↑↑ ↓↓	Longitudinal vehicle control	✗	✗
Lim et al. (2021)	FI	□	↑↑ ↓↓	Blood glucose control	✗	✗
He et al. (2021)	FI	■ □	↑↑ ↓↓	Unmanned aerial vehicle navigation	✗	✗
Remman and Lekkas (2021)	FI	■ □	↑↑ ⇉	Robotic manipulator control	✗	✗
Kim and Choi (2021)	FI	□	↑↑	Robotic manipulator control	✗	✗
Shi et al. (2021b)	FI	□	↑↑	ALE Duckietown (Paull et al., 2017)	✗	✗
Guan and Liu (2021)	FI	□	↑↑ ↓↓	Portfolio management	✗	✗
Shi et al. (2021a)	FI	□	↑↑ ↓↓	Portfolio management	✗	✗
Zhang et al. (2022)	FI	■ □	↑↑ ↓↓	Power system control	✗	✗
Jiang et al. (2022)	FI	■ □	↑↑ ↓↓	Autonomous driving	✗	✗

Table 8 (continued)

References	Category	Scope	Focus	Env(s) / Task(s)	US	C
Shi et al. (2022)	FI	□	↑↑	Duckietown ALE	✗	✓
Hayes and Shah (2017)	AB	■□	↑↑	Gridworld Cart-Pole Robot inspecting parts on a conveyor belt	✗	✗
Stork et al. (2020)	AB IST	■	↑↑	Gridworld Inverted Pendulum	✗	✗
Finkelstein et al. (2021)	AB	■	↑↑	Taxi Apple Picking Frozen Lake	✗	✓
Ehsan et al. (2018)	TJ	□	↑↑	Frogger	✓	✗
Amir and Amir (2018)	IST	■	↑↑	Ms. Pac-Man	✓	✗
Huang et al. (2018)	IST	■	↑↑	Highway driving	✓	✗
Huang et al. (2019)	IST	■	↑↑	Highway driving	✓	✗
Amir et al. (2019)	IST	■	↑↑	Conceptual framework	✗	✗
Lage et al. (2019a, 2019b)	IST	■	↑↑	Gridworld Ms. Pac-Man HIV Simulator	✓	✗
Gottesman et al. (2020)	IST	■	⇉	Medical cancer simulator MIMIC-III	✗	✓
Sequeira and Gervasio (2020)	IST	■	↑↑	Frogger	✓	✓
Karino et al. (2020)	IST	■	↑↑ ⇉	CliffWorld Breakout Walker2D	✗	✗
Sakai et al. (2021)	IST	■◪	↑↑	Minigrid with key and door	✓	✗

Table 8 (continued)

References	Category	Scope	Focus	Env(s) / Task(s)	US	C
Guo et al. (2021b)	IST	◪	↑↑	Pong You-Shall-Not-Pass Kick-And-Defend • Cart-Pole • Pendulum	✓	✓
Watkins et al. (2021)	IST	■	↑↑	• Highway driving • Minigrid	✓	✗
Gajcin et al. (2021)	IST EO	■	↑↑	• Highway driving	✗	✗
Frost et al. (2022)	IST	■	↑↑	• Minigrid	✓	✓
Amitai and Amir (2022)	IST	■	↑↑	• Frogger	✓	✓
van der Waa et al. (2018)	EO	◪	↑↑	• Gridworld	✓	✗
Cruz et al. (2019)	EO	◪	↑↑	• Gridworld	✗	✗
Davoodi and Komeili (2021)	EO	◪	↑↑	• BipedalWalker • Minigrid • Lunar Lander • MIMIC-III	✗	✗
Cruz et al. (2021)	EO	◪	↑↑	• Navigation task • Visual object sorting task	✗	✗
Portugal et al. (2022)	EO	◪	↑↑	• Car Racing	✗	✗
Sreedharan et al. (2022)	EO	◪	↑↑	• Montezuma's Revenge • Sokoban	✓	✓
Topin and Veloso (2019)	SA	■	↑↑	• PrereqWorld	✗	✗
Koul et al. (2019)	SA	■	↑↑	• Mode Counter environments • Tomita Grammars benchmark • ALE	✗	✓
Sreedharan et al. (2020)	SA	■	↑↑	• Domains from International Planning Competition 2011	✓	✗

Table 8 (continued)

References	Category	Scope	Focus	Env(s) / Task(s)	US	C
Nakamura and Shibuya (2020)	SA	■	↑↑	• 2D path search • Taxi (Gridworld)	✗	✗
Danesh et al. (2021)	SA	■	↑↑	• ALE • Acrobot • Cart-Pole • Lunar Lander	✗	✓
McCalmon et al. (2022)	SA	■	↑↑	• Blackjack • CliffWorld • Cart-Pole • Lunar Lander • Mountain Car	✓	✓
Bewley et al. (2022)	SA	■	↑↑	• ALE • Maze • Lunar Lander	✗	✗
Liu et al. (2018)	AD (tree-based)	■ □	↑↑	• Mountain Car • Cart-Pole • Flappy Bird	✗	✓
Bastani et al. (2018)	AD (tree-based)	■ □	⇉	• Pong • Cart-Pole • Half Cheetah	✗	✗
Verma et al. (2018)	AD (program-based)	■ □	↑↑ ⇉	• TORCS • Cart-Pole • Mountain Car • Acrobot	✗	✗
Coppens et al. (2019)	AD (tree-based	□	↑↑	• Mario AI	✗	✗
Nageshrao et al. (2019)	AD (rule-based)	■ □	↑↑ ⇊	• Driving with car following	✗	✗
Verma et al. (2019)	AD (program-based)	■ □	↑↑ ⇉	• TORCS • Mountain Car • Pendulum	✗	✓

Table 8 (continued)

References	Category	Scope	Focus	Env(s) / Task(s)	US	C
Zhang et al. (2020a)	AD (mathematical expression)	■ □	↑↑	• Cart-Pole • Acrobot • Mountain Car • Industrial Benchmark	✗	✗
Jhunjhunwala et al. (2020)	AD (tree-based)	■ □	↑↑	• Cart-Pole • Mountain Car • Lunar Lander	✗	✗
Bewley et al. (2020)	AD (tree-based)	■ □	↑↑	• Driving	✗	✗
Bastani et al. (2020)	AD (programmatic policy)	■ □	↑↑ ⇉	• Case study with results from their previous work	✗	✗
Madumal et al. (2020)	AD (causal graphical model)	■ ◪ □	↑↑	• Mountain Car • Cart-Pole • Taxi • Lunar Lander • BipedalWalker • Starcraft II	✓	✗
Ault et al. (2020)	AD (regulatable precedence function)	■ □	↑↑ ↓↓	• Traffic light control	✗	✓
Wollenstein-Betech et al. (2020)	AD (logic-based)	■ □	↑↑	• Traffic light control	✗	✗
Bewley and Lawry (2021)	AD (tree-based)	■ ◪ □	↑↑	• 2-dimensional road driving • Lunar Lander	✗	✓
Skirzynski et al. (2021)	AD (rule-based)	■ □	↑↑	• Mouselab-MDP	✓	✓
Soares et al. (2021)	AD (rule-based)	■ □	↑↑ ↓↓	• Driving	✗	✗
Hüyük et al. (2021)	AD (decision boundaries)	■ □	↑↑ ↓↓	• Healthcare dataset	✓	✓
Mitsopoulos et al. (2021)	AD (instance-based learning)	□	↑↑	• Two-beacon task in Starcraft II • Gridworld adversary task	✗	✗
Gjærum et al. (2021)	AD (tree-based)	■ □	↑↑ ↓↓	• Autonomous surface vessel docking	✗	✗

Table 8 (continued)

References	Category	Scope	Focus	Env(s) / Task(s)	US	C
Liu et al. (2021)	AD (tree-based)	■ □	↑↑	• Flappy Birds • ALE	✓	✓
Sieusahai and Guzdial (2021)	AD (tree-based)	■ □	↑↑	• ALE	✗	✗
Gjærum et al. (2021)	AD (tree-based)	■ □	↑↑ ⇊	• Autonomous surface vessel docking	✗	✗
Larsen and Schmidt (2021)	AD (program-based)	■ □	↑↑ ⇉	• Pendulum Swing-Up	✗	✗
Ghosh et al. (2021)	AD (tree-based)	■ □	↑↑ ⇊	• Mountain Car • Lane changing in highway driving	✗	✗
Jayawardana et al. (2021)	AD (tree-based)	■ □	↑↑ ⇉ ⇊	• Traffic light control	✗	✗
Schmidt et al. (2021)	AD (tree-based)	■ □	↑↑ ⇉	• Highway lane changing	✗	✗
Pace et al. (2022)	AD (tree-based)	■ ◪ □	↑↑ ⇊	• Healthcare dataset	✓	✗
Vasic et al. (2022)	AD (tree-based)	■ □	↑↑ ⇉	• Cart-Pole • Acrobot • Mountain Car • Lunar Lander • Pong • Pendulum	✗	✓
Dhebar et al. (2022)	AD (tree-based)	■ □	↑↑	• Cart-Pole • Mountain Car • Lunar Lander • Driving, car following • Acrobot	✗	✓
Honda and Hagiwara (2022)	AD (rule-based)	■ □	↑↑	• Mountain Car • Cart-Pole	✗	✗
Dai et al. (2022b)	AD (tree-based)	■ □	↑↑ ⇊	• Power system emergency control	✗	✗
Zhu et al. (2021)	AD (tree-based)	■ □	↑↑ ⇊	• Traffic light control	✗	✗
Xie et al. (2022)	AD	■	↑↑	• Summarization task	✗	✗
Sequeira et al. (2019)	EA	■ ◪	↑↑	• Proposal of a new XRL framework	✗	✗
Acharya et al. (2020)	AB EA	■ ◪	↑↑	• Gridworld	✗	✗

Table 8 (continued)

References	Category	Scope	Focus	Env(s) / Task(s)	US	C
Kotevska et al. (2020)	EA	■ □	↑↑	• Heating ventilation and air-conditioning device control	✗	✗
Russell and Santos (2019)	EA	■ □	↑↑	• Object World	✗	✗
Michaud et al. (2020)	EA	□	↑↑ ⇉	• Gridworld • ALE	✗	✓
Hilton et al. (2020)	EA	■ □	↑↑	• CoinRun	✗	✓
Guo et al. (2021a)	EA	□	↑↑	• Atari-HEAD (Zhang et al., 2020b) dataset	✗	✓
Løver et al. (2021)	EA	■ □	↑↑ ⇊	• Autonomous surface vessel docking	✗	✗
Dai et al. (2022a)	EA	■ □	↑↑	• Perform target-reaching with visuomotor control	✗	✓
Agrawal and McComb (2022)	EA	■	↑↑ ⇊	• Aerial vehicle • Race car design	✗	✗
Ullauri et al. (2022)	EA	■ ◪ □	↑↑	• Autonomous airborne base station control	✗	✗
Druce et al. (2019)	EA	◪ □	↑↑	• ALE	✓	✗
Wang et al. (2019a)	VA	■ ◪ □	↑↑	• ALE	✓	✗
Jaunet et al. (2020)	VA	■ ◪ □	↑↑	• ViZDoom	✓	✓
He et al. (2020)	VA	■ ◪ □	↑↑	• Ball-in-maze game	✓	✗
Wang et al. (2021b)	VA	■ ◪ □	↑↑	• ALE	✓	✗
Cheng et al. (2022)	VA	■ ◪ □	↑↑	• Lunar Lander	✓	✗
Seng et al. (2021)	VA	■ ◪ □	↑↑	• ALE	✗	✗
Mishra et al. (2022)	VA	■ ◪ □	↑↑	• Taxi navigation • Robot stacking boxes in an industrial environment • HIV drug recommendation	✓	✗

Table 8 (continued)

References	Category	Scope	Focus	Env(s) / Task(s)	US	C
Roth et al. (2021)	AD (tree-based)	■ □	↑↑ ↓↓	• Mobile Robot Navigation • Game Character Locomotion and Animation	✗	✗
Pankiewicz and Kowalczyk (2022)	EA	□ ■	↑↑	• Highway driving	✗	✗

Figure 10 lists the acronyms for the category column. US refers to if a user study has been performed, and C refers to code being open-sourced

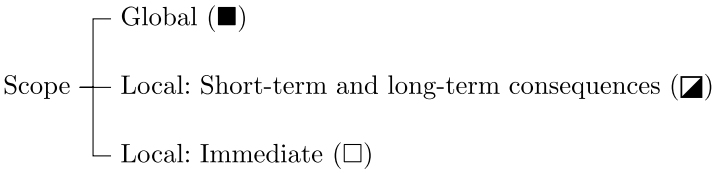

Fig. 11 The scope denotes the validity of the explanation. A global explanation justifies the agent in any state. In comparison, a local explanation explains only the behavior in a limited set of states. In sequential decision-making, we differentiate between two types of local explanations, one justifying actions using the short-term and long-term consequences and the other using the immediate context

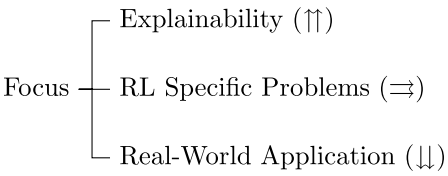

Fig. 12 The focus describes the motivation of the study in question, which can be more than one. We differentiate between three different types of motivation a study can have. If a study tries to: (1) solve the XRL problem, (2) improve an RL specific problem such as sample efficiency or generalization, and (3) solve real-world application problems

Acknowledgements The author would like to thank the anonymous reviewers for giving insightful and valuable feedback that has strengthened the literature review. Also, the author is grateful to Helge Langseth and Inga Strümke for insightful discussions and helpful feedback. Furthermore, the author would like to thank Melissa Yan for proofreading the manuscript.

Funding Open access funding provided by NTNU Norwegian University of Science and Technology (incl St. Olavs Hospital - Trondheim University Hospital). The funding was provided by the Norwegian University of Science and Technology's Department of Computer Science.

Declarations

Conflict of interest The author has no conflict of interest to declare that are relevant to the content of this article.

References

Abbeel, P., & Ng, AY. (2004). Apprenticeship learning via inverse reinforcement learning. In: C. E. Brodley (Ed.), *Machine learning, Proceedings of the twenty-first international conference* (ICML 2004), ACM International Conference Proceeding Series, vol 69. ACM https://doi.org/10.1145/1015330.1015430,

Acharya, A., Russell, R.L., & Ahmed, N.R. (2020). Explaining conditions for reinforcement learning behaviors from real and imagined data. NeurIPS Workshop on Challenges of Real-World RL https://doi.org/10.48550/ARXIV.2011.09004

Achiam, J. (2018). Spinning up in deep reinforcement learning. https://spinningup.openai.com/en/latest/index.html

Adebayo, J., Gilmer, J., Muelly, M., et al. (2018). Sanity checks for saliency maps. In S. Bengio , H. M. Wallach, H. Larochelle et al. (Eds.), *Advances in neural information processing systems 31: Annual conference on neural information processing systems* NeurIPS 2018, Montréal, pp 9525–9536, https://proceedings.neurips.cc/paper/2018/hash/294a8ed24b1ad22ec2e7efea049b8737-Abstract.html

Adebayo, J., Muelly, M., Abelson, H., et al. (2022). Post hoc explanations may be ineffective for detecting unknown spurious correlation. In *The tenth international conference on learning representations*, ICLR 2022, Virtual Event. OpenReview.net, https://openreview.net/forum?id=xNOVfCCvDpM

Agrawal, A., & McComb, C. (2022). Comparing strategies for visualizing the high-dimensional exploration behavior of CPS design agents. In *Proceedings of DESTION* pp. 64–69, https://doi.org/10.1109/DESTION56136.2022.00017

Akrour, R., Tateo, D., & Peters, J. (2021). Continuous action reinforcement learning from a mixture of interpretable experts. In *Proceedings of TPAMI*, pp. 1. https://doi.org/10.1109/TPAMI.2021.3103132

Alharin, A., Doan, T., & Sartipi, M. (2020). Reinforcement learning interpretation methods: A survey. *IEEE Access, 8*, 171058–171077. https://doi.org/10.1109/ACCESS.2020.3023394

Amir, D., & Amir, O. (2018). HIGHLIGHTS: Summarizing agent behavior to people. In E. André, S. Koenig, M. Dastani et al. (Eds.), *Proceedings of AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA/ACM, pp. 1168–1176, http://dl.acm.org/citation.cfm?id=3237869

Amir, O., Doshi-Velez, F., & Sarne, D. (2019). Summarizing agent strategies. *Autonomous Agents and Multi-Agent Systems, 33*(5), 628–644. https://doi.org/10.1007/s10458-019-09418-w

Amitai, Y., & Amir, O. (2022). "I Don't Think So": Summarizing policy disagreements for agent comparison. In *Proceedings of AAAI*, vol. *36*(5), pp. 5269–5276. https://doi.org/10.1609/aaai.v36i5.20463

Anderson, A., Dodge, J., Sadarangani, A., et al. (2019). Explaining reinforcement learning to mere mortals: An empirical study. In S. Kraus (Ed), *Proceedings of IJCAI*. ijcai.org, pp. 1328–1334, https://doi.org/10.24963/ijcai.2019/184

Angelov, P. P., & Filev, D. P. (2004). An approach to online identification of Takagi-Sugeno fuzzy models. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, 34*(1), 484–498. https://doi.org/10.1109/TSMCB.2003.817053

Angwin, J., Larson, J., Mattu, S., et al. (2016). *Machine bias*. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Annasamy, R.M., & Sycara, K.P. (2019). Towards better interpretability in deep Q-networks. In *Proceedings of AAAI*. AAAI Press, pp. 4561–4569, https://doi.org/10.1609/aaai.v33i01.33014561

Arakawa, R., Kobayashi, S., Unno, Y., et al. (2018). DQN-TAMER: Human-in-the-loop reinforcement learning with intractable feedback. CoRR abs/1810.11748. arXiv:1810.11748

Arnaldo, I., O'Reilly, U., & Veeramachaneni, K. (2015). Building predictive models via feature synthesis. In: S. Silva, A. I. Esparcia-Alcázar (Eds.), *Proceedings of GECCO*. ACM, pp. 983–990, https://doi.org/10.1145/2739480.2754693

Arrieta, A. B., Rodríguez, N. D., Ser, J. D., et al. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion, 58*, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

Atrey, A., Clary, K., & Jensen, D. D. (2020). Exploratory not explanatory: Counterfactual analysis of saliency maps for deep reinforcement learning. In *Proceedings of ICLR*. OpenReview.net, https://openreview.net/forum?id=rkl3m1BFDB

Ault, J., Hanna, J.P., Sharon, G. (2020). Learning an interpretable traffic signal control policy. In: A. E. F. Seghrouchni ,G. Sukthankar , B. An, et al (Eds.), *Proceedings of AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems, pp 88–96, https://doi.org/10.5555/3398761.3398777

Bach, S., Binder, A., Montavon, G., et al. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS One, 10*(7), 1–46. https://doi.org/10.1371/journal.pone.0130140

Bao, W., Yu, Q., & Kong, Y. (2021). DRIVE: Deep reinforced accident anticipation with visual explanation. In *Proceedings of ICCV*. IEEE, pp. 7599–7608 https://doi.org/10.1109/ICCV48922.2021.00752

Baram, N., Zahavy, T., & Mannor, S. (2017). Spatio-temporal abstractions in reinforcement learning through neural encoding. https://openreview.net/forum?id=r1yjkAtxe

Bastani, O., Inala, J.P., & Solar-Lezama, A. (2020). Interpretable, verifiable, and robust reinforcement learning via program synthesis. In A. Holzinger, R. Goebel, R. Fong, et al (Eds.), xxAI—beyond explainable AI—International workshop, Held in Conjunction with ICML 2020, Vienna, Lecture Notes in Computer Science, vol. 13200. Springer, pp. 207–228, https://doi.org/10.1007/978-3-031-04083-2_11

Bastani, O., Pu, Y., & Solar-Lezama, A. (2018). Verifiable reinforcement learning via policy extraction. In S. Bengio, H. M. Wallach, H. Larochelle, et al (Eds.) *Proceedings of NeurIPS*, pp. 2499–2509, https://proceedings.neurips.cc/paper/2018/hash/e6d8545daa42d5ced125a4bf747b3688-Abstract.html

Bautista-Montesano, R., Bustamante-Bello, R., & Ramirez-Mendoza, R. A. (2020). Explainable navigation system using fuzzy reinforcement learning. *International Journal on Interactive Design and Manufacturing (IJIDeM), 14*(4), 1411–1428. https://doi.org/10.1007/s12008-020-00717-1

Beechey, D., Smith, T.M.S., & Simsek, Ö. (2023). Explaining reinforcement learning with shapley values. In A. Krause, E. Brunskill, K. Cho, et al (Eds.), *International Conference on Machine Learning*, ICML 2023, Honolulu, Hawaii, Proceedings of Machine Learning Research, vol 202. PMLR, pp. 2003–2014, https://proceedings.mlr.press/v202/beechey23a.html

Bellemare, M. G., Naddaf, Y., Veness, J., et al. (2013). The Arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research, 47*, 253–279. https://doi.org/10.1613/jair.3912

Bellman, R. (1952). On the theory of dynamic programming. *Proceedings of the National Academy of Sciences, 38*(8), 716–719. https://doi.org/10.1073/pnas.38.8.716

Bellman, R. (1966). Dynamic programming. *Science, 153*(3731), 34–37. https://doi.org/10.1126/science.153.3731.34

Ben-Younes, H., Zablocki, É., Pérez, P., et al. (2022). Driving behavior explanation with multi-level fusion. *Pattern Recognition, 123*(108), 421. https://doi.org/10.1016/j.patcog.2021.108421

Ben-Zrihem, N., Zahavy, T., & Mannor, S. (2016). *Visualizing dynamics: From t-SNE to SEMI-MDPs*. ICML Workshop on Human Interpretability in Machine Learning https://doi.org/10.48550/ARXIV.1606.07112

Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-dynamic programming*, Optimization and neural computation series, vol 3. Athena Scientific, https://www.worldcat.org/oclc/35983505

Bewley, T., & Lawry, J. (2021). TripleTree: A versatile interpretable representation of black box agents and their environments. In *Proceedings AAAI*. AAAI Press, pp. 11,415–11,422, https://ojs.aaai.org/index.php/AAAI/article/view/17360

Bewley, T., & Lécué, F. (2022). Interpretable preference-based reinforcement learning with tree-structured reward functions. In P. Faliszewski, V. Mascardi, C. Pelachaud, et al (Eds.) *Proceedings of AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), pp 118–126, https://doi.org/10.5555/3535850.3535865

Bewley, T., Lawry, J., & Richards, A. (2020). Modelling agent policies with interpretable imitation learning. In F. Heintz, M. Milano & B. O'Sullivan (Eds.) *Proceedings of TAILOR*, Lecture Notes in Computer Science, vol 12641. (pp. 180–186). Springer https://doi.org/10.1007/978-3-030-73959-1_16

Bewley, T., Lawry, J., & Richards, A. (2022). Summarising and comparing agent dynamics with contrastive spatiotemporal abstraction. IJCAI Workshop on XAI abs/2201.07749. https://doi.org/10.48550/ARXIV.2201.07749

Beyret, B., Shafti, A., & Faisal, A.A. (2019). Dot-to-dot: Explainable hierarchical reinforcement learning for robotic manipulation. In *Proceedings of IROS* (pp. 5014–5019). IEEE https://doi.org/10.1109/IROS40897.2019.8968488

Bica, I., Jarrett, D., Hüyük, A., et al. (2021). Learning "What-if" explanations for sequential decision-making. In *Proceedings of ICLR*. OpenReview.net, https://openreview.net/forum?id=h0de3QWtGG

Böhm, G., & Pfister, H. R. (2015). How people explain their own and others' behavior: A theory of lay causal explanations. *Frontiers in Psychology, 6*, 55. https://doi.org/10.3389/fpsyg.2015.00139

Bougie, N., & Ichise, R. (2020). Towards interpretable reinforcement learning with state abstraction driven by external knowledge. *IEICE Transactions on Information and Systems, 103*(10), 2143–2153. https://doi.org/10.1587/transinf.2019EDP7170

Brown, N., & Sandholm, T. (2017). Libratus: The superhuman AI for no-limit poker. In C. Sierra (Ed) *Proceedings of IJCAI*. ijcai.org, (pp. 5226–5228) https://doi.org/10.24963/ijcai.2017/772

Buchanan, B. G., & Shortliffe, E. H. (1984). *Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project (The Addison-Wesley Series in Artificial Intelligence)*. Addison-Wesley Longman Publishing Co. Inc.

Burkart, N., & Huber, M. F. (2021). A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research, 70*, 245–317. https://doi.org/10.1613/jair.1.12228

Cao, Y., Li, Z., Yang, T., et al. (2022). GALOIS: Boosting deep reinforcement learning via generalizable logic synthesis. In: *NeurIPS*http://papers.nips.cc/paper_files/paper/2022/hash/7dd309df03d37643b96f5048b44da798-Abstract-Conference.html

Chan, F., Chen, Y., Xiang, Y., et al. (2016). Anticipating accidents in dashcam videos. In S. Lai, V. Lepetit, K. Nishino, et al (Eds.), *Proceedings of ACCV*, LNCS, vol 10114. (pp. 136–153). Springer https://doi.org/10.1007/978-3-319-54190-7_9

Cheng, S., Li, X., Shan, G., et al. (2022). ACMViz: A visual analytics approach to understand DRL-based autonomous control model. *Journal of Visualization, 25*(2), 427–442. https://doi.org/10.1007/s12650-021-00793-9

Chen, J., Li, S. E., & Tomizuka, M. (2022). Interpretable end-to-end urban autonomous driving with latent deep reinforcement learning. *IEEE Transactions on Intelligent Transportation System, 23*(6), 5068–5078. https://doi.org/10.1109/TITS.2020.3046646

Clancey, W. J. (1987). *Knowledge-based tutoring: The GUIDON program*. Cambridge: MIT Press.

Cobbe, K., Klimov, O., Hesse, C., et al. (2019). Quantifying generalization in reinforcement learning. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of ICML*, *Proceedings of machine learning research*, vol 97 (pp. 1282–1289). PMLR, http://proceedings.mlr.press/v97/cobbe19a.html

Codevilla, F., Müller, M., López, A.M., et al. (2018). End-to-end driving via conditional imitation learning. In *Proceedings of ICRA* (pp. 1–9). IEEE, https://doi.org/10.1109/ICRA.2018.8460487

Coppens, Y., Efthymiadis, K., Lenaerts, T., et al. (2019). Distilling deep reinforcement learning policies in soft decision trees. In *Proceedings of IJCAI/ECAI workshop on XAI*, https://researchportal.vub.be/en/publications/distilling-deep-reinforcement-learning-policies-in-soft-decision-

Coumans, E., & Bai, Y. (2016–2021). PyBullet, a Python module for physics simulation for games, robotics and machine learning. https://pybullet.org/

Cruz, C.A., & Igarashi, T. (2020). A survey on interactive reinforcement learning: Design principles and open challenges. In R. Wakkary, K. Andersen, W. Odom, et al (Eds.), *DIS '20: Designing interactive systems conference 2020*, Eindhoven, The Netherlands (pp. 1195–1209). ACM, https://doi.org/10.1145/3357236.3395525,

Cruz, C.A., & Igarashi, T. (2021). Interactive explanations: Diagnosis and repair of reinforcement learning based agent behaviors. In *Proceedings of CoG* (pp 1–8). IEEE, https://doi.org/10.1109/CoG52621.2021.9618999

Cruz, F., Dazeley, R., & Vamplew, P. (2019). Memory-based explainable reinforcement learning. In J. Liu & J. Bailey (Eds.), *AI 2019: Advances in artificial intelligence—32nd Australasian joint conference*, Adelaide, Proceedings, Lecture notes in computer science, vol. 11919 (pp 66–77). Springer, https://doi.org/10.1007/978-3-030-35288-2_6

Cruz, F., Dazeley, R., Vamplew, P., et al. (2021). Explainable robotic systems: Understanding goal-driven actions in a reinforcement learning scenario. Neural Computing and Applications S.I.: LatinX in AI Research. https://doi.org/10.1007/s00521-021-06425-5

Cultrera, L., Seidenari, L., Becattini, F., et al. (2020). Explaining autonomous driving by learning end-to-end visual attention. In *2020 IEEE/CVF conference on computer vision and pattern recognition*, CVPR Workshops 2020. Computer Vision Foundation/IEEE (pp. 1389–1398), https://doi.org/10.1109/CVPRW50498.2020.00178

Custode, L.L., & Iacca, G. (2021). A co-evolutionary approach to interpretable reinforcement learning in environments with continuous action spaces. In *Proceedings of SSCI* (pp 1–8). IEEE, https://doi.org/10.1109/SSCI50451.2021.9660048

Dai, T., Arulkumaran, K., Gerbert, T., et al. (2022). Analysing deep reinforcement learning agents trained with domain randomisation. *Neurocomputing, 493*, 143–165. https://doi.org/10.1016/j.neucom.2022.04.005

Dai, Y., Chen, Q., Zhang, J., et al. (2022). Enhanced oblique decision tree enabled policy extraction for deep reinforcement learning in power system emergency control. *Electric Power Systems Research, 209*(107), 932. https://doi.org/10.1016/j.epsr.2022.107932

Dai, Y., Ouyang, H., Zheng, H., et al. (2022). Interpreting a deep reinforcement learning model with conceptual embedding and performance analysis. *Applied Intelligence*. https://doi.org/10.1007/s10489-022-03788-7

Danesh, M. H., Koul, A., Fern, A., et al. (2021). Re-understanding finite-state representations of recurrent policy networks. In M. Meila & T. Zhang (Eds.), *Proceedings of ICML*, *Proceedings of machine learning research*, vol 139 (pp. 2388–2397). PMLR, http://proceedings.mlr.press/v139/danesh21a.html

Dao, G., Huff, W.H., & Lee, M. (2021). Learning sparse evidence-driven interpretation to understand deep reinforcement learning agents. In *IEEE symposium series on computational intelligence*, SSCI 2021, Orlando (pp. 1–7). IEEE, https://doi.org/10.1109/SSCI50451.2021.9660192

Dao, G., Mishra, I., & Lee, M. (2018). Deep reinforcement learning monitor for snapshot recording. In M. A. Wani, M. M. Kantardzic, M. S. Mouchaweh, et al (Eds.), *17th IEEE international conference on machine learning and applications*, ICMLA 2018, Orlando (pp 591–598). IEEE, https://doi.org/10.1109/ICMLA.2018.00095

Davoodi, O., & Komeili, M. (2021). Feature-based interpretable reinforcement learning based on state-transition models. In *Proceedings of SMC* (pp. 301–308). IEEE, https://doi.org/10.1109/SMC52423.2021.9658917

Dazeley, R., Vamplew, P., & Cruz, F. (2021a). Explainable reinforcement learning for broad-XAI: A conceptual framework and survey. arXiv:2108.09003

Dazeley, R., Vamplew, P., Foale, C., et al. (2021). Levels of explainable artificial intelligence for human-aligned conversational explanations. *Artificial Intelligence, 299*(103), 525. https://doi.org/10.1016/j.artint.2021.103525

Dhebar, Y., Deb, K., Nageshrao, S., et al. (2022). Toward interpretable-AI policies using evolutionary nonlinear decision trees for discrete-action systems. *IEEE Transactions on Cybernetics Early Access*. https://doi.org/10.1109/TCYB.2022.3180664

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. CoRR abs/1702.08608. https://doi.org/10.48550/ARXIV.1702.08608

Doshi-Velez, F., Kortz, M., Budish, R., et al. (2017). Accountability of AI under the law: The role of explanation. CoRR abs/1711.01134. https://doi.org/10.48550/ARXIV.1711.01134

Dosovitskiy, A., Ros, G., Codevilla, F., et al. (2017). CARLA: An open urban driving simulator. In *Proceedings of CoRL*, *Proceedings of MLR*, vol 78 (pp. 1–16). PMLR, http://proceedings.mlr.press/v78/dosovitskiy17a.html

Druce, J., Harradon, M., & Tittle, J. (2019). Explainable artificial intelligence (XAI) for increasing user trust in deep reinforcement learning driven autonomous systems. NeurIPS Workshop on Deep RL abs/2106.03775. https://doi.org/10.48550/ARXIV.2106.03775

Du, M., Liu, N., & Hu, X. (2020). Techniques for interpretable machine learning. *Communications of the ACM, 63*(1), 68–77. https://doi.org/10.1145/3359786

Ehsan, U., Harrison, B., Chan, L., et al. (2018). Rationalization: A neural machine translation approach to generating natural language explanations. In J. Furman, G. E. Marchant, H. Price, et al (Eds.) *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES 2018 (pp. 81–87). ACM, https://doi.org/10.1145/3278721.3278736

Erwig, M., Fern, A., Murali, M., et al. (2018). Explaining deep adaptive programs via reward decomposition. In *IJCAI/ECAI workshop on explainable AI*, https://par.nsf.gov/biblio/10096985

Evans, R., & Grefenstette, E. (2018). Learning explanatory rules from noisy data. *Journal of Artificial Intelligence Research, 61*, 1–64. https://doi.org/10.1613/jair.5714

Everingham, M., Gool, L. V., Williams, C. K. I., et al. (2010). The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision, 88*(2), 303–338. https://doi.org/10.1007/s11263-009-0275-4

Fang, J., Yan, D., Qiao, J., et al. (2019). DADA-2000: Can driving accident be predicted by driver attention*f* analyzed by a benchmark. In *Proceedings of ITSC* (pp. 4303–4309). IEEE, https://doi.org/10.1109/ITSC.2019.8917218

Feit, F., Metzger, A., & Pohl, K. (2022). Explaining online reinforcement learning decisions of self-adaptive systems. In R. Casadei, E. D. Nitto, I. Gerostathopoulos, et al (Eds.), *IEEE international conference on autonomic computing and self-organizing systems*, ACSOS 2022, Virtual (pp. 51–60). IEEE, https://doi.org/10.1109/ACSOS55765.2022.00023,

Finkelstein, M., Schlot, N.L., Liu, L., et al. (2021). Deep reinforcement learning explanation via model transforms. In *NeurIPS on Workshop Deep RL 2021*, https://openreview.net/forum?id=yRMehOHpRCy

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics, 29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Frosst, N., & Hinton, G. E. (2017). Distilling a neural network into a soft decision tree. In T. R. Besold & O. Kutz (Eds.), *Proceedings of the first international workshop on comprehensibility and explanation in AI and ML 2017 co-located with 16th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2017)*. CEUR Workshop Proceedings, vol 2071. CEUR-WS.org, http://ceur-ws.org/Vol-2071/CExAIIA_2017_paper_3.pdf

Frost, J., Watkins, O., Weiner, E., et al. (2022). *Explaining reinforcement learning policies through counterfactual trajectories*. ICML 2021 Workshop on HILL abs/2201.12462. https://doi.org/10.48550/ARXIV.2201.12462

Fukuchi, Y., Osawa, M., Yamakawa, H., et al. (2017a). Application of instruction-based behavior explanation to a reinforcement learning agent with changing policy. In D. Liu, S. Xie, Y. Li, et al (Eds.), *Neural information processing - 24th international conference, ICONIP 2017*, Proceedings, Part I, Lecture Notes in Computer Science, vol 10634 (pp 100–108). Springer, https://doi.org/10.1007/978-3-319-70087-8_11

Fukuchi, Y., Osawa, M., Yamakawa, H., et al. (2017b). Autonomous self-explanation of behavior for interactive reinforcement learning agents. In B. Wrede, Y. Nagai, T. Komatsu, et al (Eds.) *Proceedings of the 5th international conference on human agent interaction, HAI 2017* (pp. 97–101). ACM, https://doi.org/10.1145/3125739.3125746

Fukuchi, Y., Osawa, M., Yamakawa, H., et al. (2022). Explaining intelligent agent's future motion on basis of vocabulary learning with human goal inference. *IEEE Access, 10*, 54336–54347. https://doi.org/10.1109/ACCESS.2022.3176104

Gajcin, J., Nair, R., Pedapati, T., et al. (2021). Contrastive explanations for comparing preferences of reinforcement learning agents. AAAI Workshop on Interactive Machine Learning abs/2112.09462. https://doi.org/10.48550/ARXIV.2112.09462

Gangopadhyay, B., Soora, H., & Dasgupta, P. (2022). Hierarchical program-triggered reinforcement learning agents for automated driving. *IEEE Transactions on Intelligent Transportation Systems, 23*(8), 10902–10911. https://doi.org/10.1109/TITS.2021.3096998

García, J., & Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research, 16*, 1437–1480. https://doi.org/10.5555/2789272.2886795

Ghosh, A., Dhebar, Y.D., Guha, R., et al. (2021). Interpretable AI agent through nonlinear decision trees for lane change problem. In *IEEE symposium series on computational intelligence, SSCI 2021* (pp. 1–8). IEEE, https://doi.org/10.1109/SSCI50451.2021.9659552

Gilpin, L.H., Bau, D., Yuan, B.Z., et al. (2018). Explaining explanations: An overview of interpretability of machine learning. In F. Bonchi, F. J. Provost, T. Eliassi-Rad, et al (Eds.), *Proceedings of DSAA* (pp 80–89). IEEE, https://doi.org/10.1109/DSAA.2018.00018

Gilpin, L. H., Paley, A. R., Alam, M. A., et al. (2022). "Explanation" is not a technical term: The problem of ambiguity in XAI. CoRR. https://doi.org/10.48550/arXiv.2207.00007, arXiv:2207.00007

Gjærum, V. B., Rørvik, E. H., & Lekkas, A. M. (2021). Approximating a deep reinforcement learning docking agent using linear model trees. In *2021 European control conference, ECC 2021*, Virtual Event / Delft (pp 1465–1471). IEEE, https://doi.org/10.23919/ECC54610.2021.9655007

Gjærum, V. B., Strümke, I., Alsos, O. A., et al. (2021). Explaining a deep reinforcement learning docking agent using linear model trees with user adapted visualization. *Journal of Marine Science and Engineering*. https://doi.org/10.3390/jmse9111178

Glanois, C., Weng, P., Zimmer, M., et al. (2022). *A survey on interpretable reinforcement learning*. CoRR abs/2112.13112. https://doi.org/10.48550/arXiv.2112.13112

Goel, V., Weng, J., & Poupart, P. (2018). Unsupervised video object segmentation for deep reinforcement learning. In S. Bengio, H. M. Wallach, H. Larochelle, et al (Eds.) *Advances in neural information processing systems 31: annual conference on neural information processing systems 2018*, NeurIPS 2018 (pp 5688–5699), https://proceedings.neurips.cc/paper/2018/hash/96f2b50b5d3613adf9c27049b2a888c7-Abstract.html

Goldstein, A., Kapelner, A., Bleich, J., et al. (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics, 24*(1), 44–65. https://doi.org/10.1080/10618600.2014.907095

Goodman, B., & Flaxman, S. R. (2017). European union regulations on algorithmic decision-making and a "right to explanation''. *AI Magazine, 38*(3), 50–57. https://doi.org/10.1609/aimag.v38i3.2741

Gorji, S. R., Granmo, O., & Wiering, M. A. (2021). Explainable reinforcement learning with the tsetlin machine. In H. Fujita, A. Selamat, J. C. Lin, et al (Eds.), *Advances and trends in artificial intelligence. Artificial intelligence practices - 34th international conference on industrial, engineering and other applications of applied intelligent systems, IEA/AIE 2021*, Proceedings, Part I, Lecture Notes in Computer Science, vol 12798 (pp. 173–187). Springer, https://doi.org/10.1007/978-3-030-79457-6_15

Gottesman, O., Futoma, J., Liu, Y., et al. (2020). Interpretable off-policy evaluation in reinforcement learning by highlighting influential transitions. In *Proceedings of the 37th international conference on machine learning, ICML 2020, Virtual Event, Proceedings of machine learning research*, vol 119 (pp. 3658–3667). PMLR, http://proceedings.mlr.press/v119/gottesman20a.html

Granmo, O. (2018). The Tsetlin machine—A game theoretic bandit driven approach to optimal pattern recognition with propositional logic. CoRR abs/1804.01508. https://doi.org/10.48550/ARXIV.1804.01508

Greydanus, S., Koul, A., Dodge, J., et al. (2018). Visualizing and understanding atari agents. In J. G. Dy & A. Krause (Eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018. Proceedings of machine learning research*, vol 80 (pp. 1787–1796). PMLR, http://proceedings.mlr.press/v80/greydanus18a.html

Gu, S., Yang, L., Du, Y., et al. (2022). A review of safe reinforcement learning: Methods, theory and applications. CoRR. https://doi.org/10.48550/arXiv.2205.10330, arXiv:2205.10330

Guan, M., & Liu, X. (2021). Explainable deep reinforcement learning for portfolio management: An empirical approach. In A. Calinescu & L. Szpruch (Eds.) *ICAIF'21: 2nd ACM international conference on AI in Finance* (pp. 50:1–50:9). ACM, https://doi.org/10.1145/3490354.3494415

Guidotti, R., Monreale, A., Ruggieri, S., et al. (2019). A survey of methods for explaining black box models. *ACM Computing Surveys, 51*(5), 93:1-93:42. https://doi.org/10.1145/3236009

Gunning, D., & Aha, D. W. (2019). Darpa's explainable artificial intelligence (XAI) program. *AI Magazine, 40*(2), 44–58. https://doi.org/10.1609/aimag.v40i2.2850

Guo, W., Wu, X., Khan, U., et al. (2021b). EDGE: Explaining deep reinforcement learning policies. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, et al. (Eds.), *Advances in neural information processing systems 34: Annual conference on neural information processing systems 2021, NeurIPS 2021* (pp. 12222–12236), https://proceedings.neurips.cc/paper/2021/hash/65c89f5a9501a04c073b354f03791b1f-Abstract.html

Guo, S., Zhang, R., Liu, B., et al. (2021a). Machine versus human attention in deep reinforcement learning tasks. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, et al. (Eds.), *Advances in neural information processing systems 34: Annual conference on neural information processing systems 2021* (pp. 25370–25385), https://proceedings.neurips.cc/paper/2021/hash/d58e2f077670f4de9cd7963c857f2534-Abstract.html

Gupta, U.D., Talvitie, E., & Bowling, M. (2015). Policy tree: Adaptive representation for policy gradient. In B. Bonet & S. Koenig (Eds.), Proceedings of the twenty-ninth AAAI conference on artificial intelligence (pp. 2547–2553). AAAI Press, http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9781

Halpern, J. Y., & Pearl, J. (2005). Causes and explanations: A structural-model approach. Part ii: Explanations. *The British Journal for the Philosophy of Science, 56*(4), 889–911.

Hans, A., Schneegaß, D., Schäfer, A. M., et al. (2008). Safe exploration for reinforcement learning. In *16th European symposium on artificial neural networks, ESANN 2008* (pp. 143–148), https://www.esann.org/sites/default/files/proceedings/legacy/es2008-36.pdf

Hasanbeig, M., Jeppu, N.Y., Abate, A., et al. (2021). DeepSynth: Automata synthesis for automatic task segmentation in deep reinforcement learning. In *Thirty-Fifth AAAI conference on artificial intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021* (pp. 7647–7656). AAAI Press, https://ojs.aaai.org/index.php/AAAI/article/view/16935

Hayes, B., & Shah, J.A. (2017). Improving robot controller transparency through autonomous policy explanation. In B. Mutlu, M. Tscheligi, A. Weiss, et al. (Eds.) *Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction, HRI 2017* (pp. 303–312). ACM, https://doi.org/10.1145/2909824.3020233

He, W., Lee, T.Y., van Baar, J., et al. (2020). DynamicsExplorer: Visual analytics for robot control tasks involving dynamics and LSTM-based control policies. In *PacificVis* (pp. 36–45), https://doi.org/10.1109/PacificVis48177.2020.7127

He, L., Aouf, N., & Song, B. (2021). Explainable deep reinforcement learning for UAV autonomous path planning. *Aerospace Science and Technology, 118*(107), 052. https://doi.org/10.1016/j.ast.2021.107052

Hein, D., Depeweg, S., Tokic, M., et al. (2017a). A benchmark environment motivated by industrial control problems. In *SSCI* (pp. 1–8). IEEE, https://doi.org/10.1109/SSCI.2017.8280935

Hein, D., Udluft, S., & Runkler, T.A. (2018a). Generating interpretable fuzzy controllers using particle swarm optimization and genetic programming. In H. E. Aguirre & K. Takadama (Eds.), *Proceedings of the genetic and evolutionary computation conference companion, GECCO 2018* (pp. 1268–1275). ACM, https://doi.org/10.1145/3205651.3208277

Hein, D., Hentschel, A., Runkler, T. A., et al. (2017). Particle swarm optimization for generating interpretable fuzzy reinforcement learning policies. *Engineering Applications of Artificial Intelligence, 65*, 87–98. https://doi.org/10.1016/j.engappai.2017.07.005

Hein, D., Udluft, S., & Runkler, T. A. (2018). Interpretable policies for reinforcement learning by genetic programming. *Engineering Applications of Artificial Intelligence, 76*, 158–169. https://doi.org/10.1016/j.engappai.2018.09.007

Hengst, B. (2010). *Hierarchical reinforcement learning* (pp. 495–502). Boston: Springer.

Heuillet, A., Couthouis, F., & Rodríguez, N. D. (2021). Explainability in deep reinforcement learning. *Knowledge-Based Systems, 214*(106), 685. https://doi.org/10.1016/j.knosys.2020.106685

Hickling, T., Zenati, A., Aouf, N., et al. (2022). Explainability in deep reinforcement learning, a review into current methods and applications. CoRR abs/2207.01911. https://doi.org/10.48550/arXiv.2207.01911

Hilton, J., Cammarata, N., Carter, S., et al. (2020). Understanding RL vision. *Distill*. https://doi.org/10.23915/distill.00029

Ho, J., & Ermon, S. (2016). Generative adversarial imitation learning. In D. D. Lee, M. Sugiyama, U. von Luxburg, et al. (Eds.), *Advances in neural information processing systems 29: Annual conference on neural information processing systems 2016* (pp. 4565–4573), https://proceedings.neurips.cc/paper/2016/hash/cc7e2b878868cbae992d1fb743995d8f-Abstract.html

Hohman, F., Kahng, M., Pienta, R., et al. (2019). Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE Transactions on Visualization and Computer Graphics, 25*(8), 2674–2693. https://doi.org/10.1109/TVCG.2018.2843369

Honda, H., & Hagiwara, M. (2022). Deep-learning-based fuzzy symbolic processing with agents capable of knowledge communication. In A. P. Rocha, L. Steels, H. J. van den Herik (Eds.), *Proceedings of the 14th international conference on agents and artificial intelligence, ICAART 2022*, Vol. 3 (pp. 172–179). SCITEPRESS, https://doi.org/10.5220/0010796300003116

Huang, S.H., Bhatia, K., Abbeel, P., et al. (2018). Establishing appropriate trust via critical states. In *2018 IEEE/RSJ international conference on intelligent robots and systems, IROS 2018* (pp. 3929–3936). IEEE, https://doi.org/10.1109/IROS.2018.8593649

Huang, J., Angelov, P. P., & Yin, C. (2020). Interpretable policies for reinforcement learning by empirical fuzzy sets. *Engineering Applications of Artificial Intelligence, 91*(103), 559. https://doi.org/10.1016/j.engappai.2020.103559

Huang, S. H., Held, D., Abbeel, P., et al. (2019). Enabling robots to communicate their objectives. *Autonomous Robots, 43*(2), 309–326. https://doi.org/10.1007/s10514-018-9771-0

Huber, T., Schiller, D., & André, E. (2019). Enhancing explainability of deep reinforcement learning through selective layer-wise relevance propagation. In C. Benzmüller & H. Stuckenschmidt (Eds.), *KI 2019: Advances in Artificial Intelligence - 42nd German Conference on AI, Kassel*, Lecture Notes in Computer Science, vol. 11793 (pp. 188–202). Springer, https://doi.org/10.1007/978-3-030-30179-8_16

Huber, T., Weitz, K., André, E., et al. (2021). Local and global explanations of agent behavior: Integrating strategy summaries with saliency maps. *Artificial Intelligence, 301*(103), 571. https://doi.org/10.1016/j.artint.2021.103571

Hüyük, A., Jarrett, D., Tekin, C., et al. (2021). Explaining by imitating: Understanding decisions by interpretable policy learning. In *9th international conference on learning representations, ICLR 2021*. OpenReview.net, https://openreview.net/forum?id=unI5ucw_Jk

III, D. J. H., & Sadigh, D. (2022). Few-shot preference learning for human-in-the-loop RL. In K. Liu, D. Kulic, J. Ichnowski (Eds.), *Conference on robot learning, CoRL 2022, Proceedings of machine learning research*, vol 205 (pp. 2014–2025). PMLR, https://proceedings.mlr.press/v205/iii23a.html

Illanes, L., Yan, X., Icarte, R.T., et al. (2020). Symbolic plans as high-level instructions for reinforcement learning. In J. C. Beck, O. Buffet, J. Hoffmann, et al. (Eds.), *Proceedings of the thirtieth international conference on automated planning and scheduling* (pp. 540–550). AAAI Press, https://ojs.aaai.org/index.php/ICAPS/article/view/6750

Itaya, H., Hirakawa, T., Yamashita, T., et al. (2021). Visual explanation using attention mechanism in actor-critic-based deep reinforcement learning. In *International joint conference on neural networks, IJCNN 2021* (pp. 1–10). IEEE, https://doi.org/10.1109/IJCNN52387.2021.9534363

Iucci, A., Hata, A., Terra, A., et al. (2021). Explainable reinforcement learning for human-robot collaboration. In *20th international conference on advanced robotics, ICAR 2021* (pp. 927–934). IEEE, https://doi.org/10.1109/ICAR53236.2021.9659472

Iyer, R., Li, Y., Li, H., et al. (2018). Transparency and explanation in deep reinforcement learning neural networks. In J. Furman, G. E. Marchant, H. Price, et al. (Eds.), *Proceedings of the 2018 AAAI/ACM conference on AI, ethics, and society, AIES 2018* (pp. 144–150). ACM, https://doi.org/10.1145/3278721.3278776

Jacovi, A., & Goldberg, Y. (2020). Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In D. Jurafsky, J. Chai, N. Schluter, et al. (Eds.), *Proceedings of the 58th*

annual meeting of the association for computational linguistics, ACL 2020. Association for Computational Linguistics (pp. 4198–4205), https://doi.org/10.18653/v1/2020.acl-main.386

Jacq, A., Ferret, J., Pietquin, O., et al. (2022). Lazy-MDPs: Towards Interpretable RL by Learning When to Act. In: Faliszewski P, Mascardi V, Pelachaud C, et al (eds) 21st International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2022, Auckland, New Zealand, May 9-13, 2022. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), pp 669–677, https://doi.org/10.5555/3535850.3535926

Jain, S., & Wallace, B.C. (2019). Attention is not explanation. In J. Burstein, C. Doran, T. Solorio (Eds.), *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies, NAACL-HLT 2019*, Vol. 1 (Long and Short Papers). Association for Computational Linguistics (pp. 3543–3556), https://doi.org/10.18653/v1/n19-1357,

Jaunet, T., Vuillemot, R., & Wolf, C. (2020). DRLViz: Understanding decisions and memory in deep reinforcement learning. *Computer Graphics Forum, 39*(3), 49–61. https://doi.org/10.1111/cgf.13962

Jayawardana, V., Landler, A., & Wu, C. (2021). Mixed autonomous supervision in traffic signal control. In *24th IEEE international intelligent transportation systems conference, ITSC 2021* (pp. 1767–1773). IEEE, https://doi.org/10.1109/ITSC48978.2021.9565053

Jhunjhunwala, A., Lee, J., Sedwards, S., et al. (2020). Improved policy extraction via online Q-value distillation. In *2020 international joint conference on neural networks, IJCNN 2020* (pp. 1–8). IEEE, https://doi.org/10.1109/IJCNN48605.2020.9207648

Jiang, Z., & Luo, S. (2019). Neural logic reinforcement learning. In K. Chaudhuri, R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning, ICML 2019, Proceedings of machine learning research*, vol 97 (pp. 3110–3119). PMLR, http://proceedings.mlr.press/v97/jiang19a.html

Jiang, X., Zhang, J., & Wang, B. (2022). Energy-efficient driving for adaptive traffic signal control environment via explainable reinforcement learning. *Applied Sciences*. https://doi.org/10.3390/app12115380

Johnson, M., Hofmann, K., Hutton, T., et al. (2016b). The Malmo platform for artificial intelligence experimentation. In S. Kambhampati (Ed.) *Proceedings of IJCAI* (pp. 4246–4247). IJCAI/AAAI Press, http://www.ijcai.org/Abstract/16/643

Johnson, A. E. W., Pollard, T. J., Shen, L., et al. (2016). MIMIC-III, a freely accessible critical care database. *Scientific Data, 3*(1), 160035. https://doi.org/10.1038/sdata.2016.35

Joo, H., & Kim, K. (2019). Visualization of deep reinforcement learning using Grad-CAM: How AI plays atari games? In *IEEE conference on games, CoG 2019* (pp. 1–2). IEEE, https://doi.org/10.1109/CIG.2019.8847950

Josef, S., & Degani, A. (2020). Deep reinforcement learning for safe local planning of a ground vehicle in unknown rough terrain. *IEEE Robotics and Automation Letters, 5*(4), 6748–6755. https://doi.org/10.1109/LRA.2020.3011912

Juozapaitis, Z., Koul, A., Fern, A., et al. (2019). Explainable reinforcement learning via reward decomposition. In *IJCAI/ECAI workshop on explainable AI*, https://finale.seas.harvard.edu/publications/explainable-reinforcement-learning-reward-decomposition

Karakovskiy, S., & Togelius, J. (2012). The mario AI benchmark and competitions. *IEEE Transactions on Computational Intelligence and AI in Games, 4*(1), 55–67. https://doi.org/10.1109/TCIAIG.2012.2188528

Karino, I., Ohmura, Y., & Kuniyoshi, Y. (2020). Identifying critical states by the action-based variance of expected return. In I. Farkas, P. Masulli, S. Wermter (Eds.), *Artificial neural networks and machine learning - ICANN 2020 - 29th international conference on artificial neural networks*, Part I, Lecture notes in computer science, vol. 12396 (pp. 366–378), Springer. https://doi.org/10.1007/978-3-030-61609-0_29

Kempka, M., Wydmuch, M., Runc, G., et al. (2016). Vizdoom: A doom-based AI research platform for visual reinforcement learning. In *IEEE conference on computational intelligence and games, CIG 2016* (pp. 1–8). IEEE, https://doi.org/10.1109/CIG.2016.7860433

Kim, J., & Canny, J.F. (2017). Interpretable learning for self-driving cars by visualizing causal attention. In *IEEE international conference on computer vision, ICCV 2017*. IEEE Computer Society (pp. 2961–2969), https://doi.org/10.1109/ICCV.2017.320

Kim, S., & Choi, J. (2021). Explaining the decisions of deep policy networks for robotic manipulations. In *IEEE/RSJ international conference on intelligent robots and systems, IROS 2021* (pp. 2663–2669). IEEE, https://doi.org/10.1109/IROS51168.2021.9636594

Kim, W.K., Lee, Y., & Woo, H. (2022). Mean-variance based risk-sensitive reinforcement learning with interpretable attention. In *ICMVA 2022: The 5th international conference on machine vision and applications* (pp. 104–109). ACM, https://doi.org/10.1145/3523111.3523127

Kim, J., Rohrbach, A., Darrell, T., et al. (2018). Textual explanations for self-driving vehicles. In V. Ferrari, M. Hebert, C. Sminchisescu, et al. (Eds.) *Computer vision - ECCV 2018 - 15th European conference*, Proceedings, Part II, Lecture notes in computer science, vol 11206 (pp. 577–593). Springer, https://doi.org/10.1007/978-3-030-01216-8_35

Kimura, D., Ono, M., Chaudhury, S., et al. (2021). Neuro-symbolic reinforcement learning with first-order logic. In M. Moens, X. Huang, L. Specia, et al. (Eds.), *Proceedings of the 2021 conference on empirical methods in natural language processing, EMNLP 2021* . Association for computational linguistics (pp. 3505–3511), https://doi.org/10.18653/v1/2021.emnlp-main.283

Kingma, D.P., & Welling, M. (2014). Auto-encoding variational Bayes. In Y. Bengio & Y. LeCun (Eds.), *2nd international conference on learning representations, ICLR 2014, Conference Track Proceedings*, arxiv:1312.6114

Kirsch, A. (2017). Explain to whom? Putting the user in the center of explainable AI. In T. R. Besold & O. Kutz (Eds.) *Proceedings of the first international workshop on comprehensibility and explanation in AI and ML 2017 co-located with 16th international conference of the italian association for artificial intelligence (AI*IA 2017), CEUR Workshop Proceedings*, vol 2071. CEUR-WS.org, http://ceur-ws.org/Vol-2071/CExAIIA_2017_keynote_1.pdf

Kitchenham, B. A., Brereton, P., Budgen, D., et al. (2009). Systematic literature reviews in software engineering—A systematic literature review. *Information and Software Technology, 51*(1), 7–15. https://doi.org/10.1016/j.infsof.2008.09.009

Kitchenham, B. A., Budgen, D., & Brereton, P. (2020). *Evidence-based software engineering and systematic reviews*. Chapman and Hall/CRC.

Koenig, N.P., & Howard, A. (2004). Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *Proceedings of IROS* (pp. 2149–2154). IEEE, https://doi.org/10.1109/IROS.2004.1389727

Koh, P.W., & Liang, P. (2017). Understanding black-box predictions via influence functions. In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th international conference on machine learning, ICML 2017, Proceedings of machine learning research*, vol. 70 (pp. 1885–1894). PMLR, http://proceedings.mlr.press/v70/koh17a.html

Kotevska, O., Munk, J., Kurte, K.R., et al. (2020). Methodology for interpretable reinforcement learning model for HVAC energy control. In X. Wu, C. Jermaine, L. Xiong, et al. (Eds.), *2020 IEEE international conference on big data (IEEE BigData 2020)* (pp. 1555–1564). IEEE, https://doi.org/10.1109/BigData50022.2020.9377735

Koul, A., Fern, A., & Greydanus, S. (2019). Learning finite state representations of recurrent policy networks. In *7th international conference on learning representations, ICLR 2019*, 2019. OpenReview.net, https://openreview.net/forum?id=S1gOpsCctm

Krajna, A., Brcic, M., Lipic, T., et al. (2022). Explainability in reinforcement learning: perspective and position. CoRR abs/2203.11547. https://doi.org/10.48550/arXiv.2203.11547

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, et al. (Eds.), *Advances in neural information processing systems 25: 26th annual conference on neural information processing systems 2012* (pp. 1106–1114), https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

Kubalík, J., Derner, E., Zegklitz, J., et al. (2021). Symbolic regression methods for reinforcement learning. *IEEE Access, 9*, 139697–139711. https://doi.org/10.1109/ACCESS.2021.3119000

Kuramoto, S., Sawada, H., & Hartono, P. (2020). Visualization of topographical internal representation of learning robots. In *2020 international joint conference on neural networks, IJCNN 2020* (pp. 1–7). IEEE, https://doi.org/10.1109/IJCNN48605.2020.9206675

Lage, I., Lifschitz, D., Doshi-Velez, F., et al. (2019a). Exploring computational user models for agent policy summarization. In S. Kraus (Ed.), *Proceedings of the twenty-eighth international joint conference on artificial intelligence, IJCAI 2019* ijcai.org (pp. 1401–1407), https://doi.org/10.24963/ijcai.2019/194

Lage, I., Lifschitz, D., Doshi-Velez, F., et al. (2019b). Toward robust policy summarization. In E. Elkind, M. Veloso, N. Agmon, et al. (Eds.), *Proceedings of the 18th international conference on autonomous agents and multiagent systems, AAMAS '19*. International Foundation for Autonomous Agents and Multiagent Systems (pp. 2081–2083), http://dl.acm.org/citation.cfm?id=3332017

Landajuela, M., Petersen, B. K., Kim, S., et al. (2021). Discovering symbolic policies with deep reinforcement learning. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th international conference on machine learning, ICML 2021, Proceedings of machine learning research*, vol 139. (pp. 5979–5989). PMLR, http://proceedings.mlr.press/v139/landajuela21a.html

Langer, M., Oster, D., Speith, T., et al. (2021). What do we want from explainable artificial intelligence (XAI)?—A stakeholder perspective on XAI and a conceptual model guiding interdisciplinary XAI research. *Artificial Intelligence, 296*(103), 473. https://doi.org/10.1016/j.artint.2021.103473

Lapuschkin, S., Wäldchen, S., Binder, A., et al. (2019). Unmasking Clever Hans predictors and assessing what machines really learn. *Nature Communications, 10*(1), 1096. https://doi.org/10.1038/s41467-019-08987-4

Larsen, R., & Schmidt, M. N. (2021). Programmatic policy extraction by iterative local search. In N. Katzouris & A. Artikis (Eds.) *Inductive logic programming - 30th international conference, ILP 2021*, Lecture notes in computer science, vol 13191 (pp. 156–166). Springer, https://doi.org/10.1007/978-3-030-97454-1_11

Larson, J., Mattu, S., Kirchner, L., et al. (2016). How we analyzed the COMPAS recidivism algorithm. https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm

Lee, M. (2017). Sparse Bayesian reinforcement learning. PhD thesis, Colorado State University, https://mountainscholar.org/bitstream/handle/10217/183935/Lee_colostate_0053A_14302.pdf

Liessner, R., Dohmen, J., & Wiering, M. A. (2021). Explainable reinforcement learning for longitudinal control. In A. P. Rocha, L. Steels, H. J. van den Herik (Eds.), *Proceedings of the 13th international conference on agents and artificial intelligence, ICAART 2021*, Vol. 2. (pp. 874–881). SCITEPRESS, https://doi.org/10.5220/0010256208740881

Li, G., Gomez, R., Nakamura, K., et al. (2019). Human-centered reinforcement learning: A survey. *IEEE Transactions on Human-Machine Systems, 49*(4), 337–349. https://doi.org/10.1109/THMS.2019.2912447

Likmeta, A., Metelli, A. M., Tirinzoni, A., et al. (2020). Combining reinforcement learning with rule-based controllers for transparent and general decision-making in autonomous driving. *Robotics and Autonomous Systems, 131*(103), 568. https://doi.org/10.1016/j.robot.2020.103568

Lim, B.Y., Dey, A.K., & Avrahami, D. (2009). *Why and why not* explanations improve the intelligibility of context-aware intelligent systems. In D. R. O. Jr, R. B. Arthur, K. Hinckley, et al. (Eds.) *Proceedings of the 27th international conference on human factors in computing systems, CHI 2009* (pp. 2119–2128). ACM, https://doi.org/10.1145/1518701.1519023

Lim, M. H., Lee, W. H., Jeon, B., et al. (2021). A blood glucose control framework based on reinforcement learning with safety and interpretability: In silico validation. *IEEE Access, 9*, 105756–105775. https://doi.org/10.1109/ACCESS.2021.3100007

Lin, Z., Lam, K., & Fern, A. (2021). Contrastive explanations for reinforcement learning via embedded self predictions. In *9th international conference on learning representations, ICLR 2021*. OpenReview.net, https://openreview.net/forum?id=Ud3DSz72nYR

Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue, 16*(3), 31–57. https://doi.org/10.1145/3236386.3241340

Li, X., Serlin, Z., Yang, G., et al. (2019). A formal methods approach to interpretable reinforcement learning for robotic planning. *Science Robotics*. https://doi.org/10.1126/scirobotics.aay6276

Liu, G., Schulte, O., Zhu, W., et al. (2018). Toward interpretable deep reinforcement learning with linear model U-trees. In M. Berlingerio, F. Bonchi, T. Gärtner, et al. (Eds). *Machine learning and knowledge discovery in databases - European conference, ECML PKDD 2018*, Proceedings, Part II, Lecture notes in computer science, vol 11052 (pp. 414–429). Springer, https://doi.org/10.1007/978-3-030-10928-8_25

Liu, G., Sun, X., Schulte, O., et al. (2021). Learning tree interpretation from object representation for deep reinforcement learning. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, et al. (Eds.) *Advances in neural information processing systems 34: Annual conference on neural information processing systems 2021 NeurIPS* (pp. 19622–19636), https://proceedings.neurips.cc/paper/2021/hash/a35fe7f7fe8217b4369a0af4244d1fca-Abstract.html

Liu, Y., Wang, X., Chang, Y., et al. (2022). Towards explainable reinforcement learning using scoring mechanism augmented agents. In G. Memmi, B. Yang, L. Kong, et al. (Eds.), *Knowledge science, engineering and management - 15th international conference, KSEM 2022* Proceedings, Part II, Lecture notes in computer science, vol 13369 (pp. 547–558). Springer, https://doi.org/10.1007/978-3-031-10986-7_44

Liu, M., Shi, J., Li, Z., et al. (2017). Towards better analysis of deep convolutional neural networks. *IEEE Transactions on Visualization and Computer Graphics, 23*(1), 91–100. https://doi.org/10.1109/TVCG.2016.2598831

López, PÁ., Behrisch, M., Bieker-Walz, L., et al. (2018). Microscopic traffic simulation using SUMO. In W. Zhang, A. M. Bayen, J. J. S. Medina, et al. (Eds.), *Proceedings of ITSC* (pp. 2575–2582). IEEE, https://doi.org/10.1109/ITSC.2018.8569938

Løver, J., Gjærum, V. B., & Lekkas, A. M. (2021). Explainable AI methods on a deep reinforcement learning agent for automatic docking. *IFAC-PapersOnLine, 54*(16), 146–152. https://doi.org/10.1016/j.ifacol.2021.10.086

Lundberg, S.M., & Lee, S. (2017). A unified approach to interpreting model predictions. In I. Guyon, U. von Luxburg, S. Bengio, et al. (Eds.), *Advances in neural information processing systems 30: Annual conference on neural information processing systems 2017* (pp. 4765–4774), https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

Lyu, D., Yang, F., Liu, B., et al. (2019). SDRL: Interpretable and data-efficient deep reinforcement learning leveraging symbolic planning. In *The Thirty-Third AAAI conference on artificial intelligence, AAAI 2019, The thirty-first innovative applications of artificial intelligence conference, IAAI 2019*, The Ninth AAAI symposium on educational advances in artificial intelligence, EAAI 2019 (pp. 2970–2977). AAAI Press, https://doi.org/10.1609/aaai.v33i01.33012970

Madumal, P., Miller, T., Sonenberg, L., et al. (2020). Explainable reinforcement learning through a causal lens. In *The Thirty-Fourth AAAI conference on artificial intelligence, AAAI 2020, The thirty-second innovative applications of artificial intelligence conference, IAAI 2020, The tenth AAAI symposium on educational advances in artificial intelligence, EAAI 2020* (pp. 2493–2500). AAAI Press, https://ojs.aaai.org/index.php/AAAI/article/view/5631

Makhzani, A., Shlens, J., Jaitly, N., et al. (2015). Adversarial autoencoders. In *Proceedings of ICLR* abs/1511.05644. https://doi.org/10.48550/ARXIV.1511.05644

Matthey, L., Higgins, I., Hassabis, D., et al. (2017). dSprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/

McCalmon, J., Le, T., Alqahtani, S., et al. (2022). CAPS: Comprehensible abstract policy summaries for explaining reinforcement learning agents. In P. Faliszewski, V. Mascardi, C. Pelachaud, et al. (Eds.), *21st international conference on autonomous agents and multiagent systems, AAMAS 2022*. International foundation for autonomous agents and multiagent systems (IFAAMAS) (pp. 889–897), https://doi.org/10.5555/3535850.3535950

Merriam-Webster. (2022). Interpret definition and meaning. https://www.merriam-webster.com/dictionary/interpret

Michaud, E. J., Gleave, A., & Russell, S. (2020). Understanding learned reward functions. NeurIPS Workshop on Deep RL abs/2012.05862. https://doi.org/10.48550/ARXIV.2012.05862

Milani, S., Topin, N., Veloso, M., et al. (2022). A survey of explainable reinforcement learning. CoRR abs/2202.08434. https://doi.org/10.48550/arXiv.2202.08434

Ming, Y., Cao, S., Zhang, R., et al. (2017). Understanding hidden memories of recurrent neural networks. In B. D. Fisher, S. Liu, T. Schreck (Eds.), *Proceedings of VAST*. IEEE Computer Society (pp. 13–24), https://doi.org/10.1109/VAST.2017.8585721

Minh, D., Wang, H. X., Li, Y. F., et al. (2022). Explainable artificial intelligence: A comprehensive review. *Artificial Intelligence Review, 55*(5), 3503–3568. https://doi.org/10.1007/s10462-021-10088-y

Mishra, I., Dao, G., & Lee, M. (2018). Visual sparse Bayesian reinforcement learning: A framework for interpreting what an agent has learned. In *IEEE symposium series on computational intelligence, SSCI 2018* (pp. 1427–1434). IEEE, https://doi.org/10.1109/SSCI.2018.8628887

Mishra, A., Soni, U., Huang, J., et al. (2022). Why? Why not? When? Visual explanations of agent behaviour in reinforcement learning. In *2022 IEEE 15th Pacific Visualization Symposium (PacificVis)*. IEEE Computer Society, pp. 111–120, https://doi.org/10.1109/PacificVis53943.2022.00020

Mitsopoulos, K., Somers, S., Schooler, J., et al. (2021). Toward a psychology of deep reinforcement learning agents using a cognitive architecture. *Topics in Cognitive Science*. https://doi.org/10.1111/tops.12573

Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2013). Playing atari with deep reinforcement learning. CoRR abs/1312.5602. arXiv:1312.5602

Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. *Nature, 518*(7540), 529–533. https://doi.org/10.1038/nature14236

Mohseni, S., Zarei, N., & Ragan, E. D. (2021). A multidisciplinary survey and framework for design and evaluation of explainable AI systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)*. https://doi.org/10.1145/3387166

Moldovan, T.M., & Abbeel, P. (2012). Safe exploration in markov decision processes. In *Proceedings of the 29th international conference on machine learning, ICML 2012*. icml.cc / Omnipress, http://icml.cc/2012/papers/838.pdf

Montavon, G., Lapuschkin, S., Binder, A., et al. (2017). Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition, 65*, 211–222. https://doi.org/10.1016/j.patcog.2016.11.008

Mott, A., Zoran, D., Chrzanowski, M., et al. (2019). Towards interpretable reinforcement learning using attention augmented agents. In H. M. Wallach, H. Larochelle, A. Beygelzimer, et al. (Eds.), *Advances in neural information processing systems 32: Annual conference on neural information processing systems 2019, NeurIPS 2019*. (pp. 12329–12338), https://proceedings.neurips.cc/paper/2019/hash/e9510081ac30ffa83f10b68cde1cac07-Abstract.html

Murdoch, W. J., Singh, C., Kumbier, K., et al. (2019). Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences, 116*(44), 22071–22080. https://doi.org/10.1073/pnas.1900654116

Murphy, K. P., Kim, B., & Doshi-Velez, F. (2023). *Probabilistic machine learning: Advanced topics*. MIT Press.

Nageshrao, S., Costa, B., & Filev, D. P. (2019). Interpretable approximation of a deep reinforcement learning agent as a set of if-then rules. In M. A. Wani, T. M. Khoshgoftaar, D. Wang, et al. (Eds.), *18th IEEE international conference on machine learning and applications ICMLA 2019* (pp. 216–221). IEEE, https://doi.org/10.1109/ICMLA.2019.00041

Nakamura, Y., & Shibuya, T. (2020). Topological visualization method for understanding the landscape of value functions and structure of the state space in reinforcement learning. In A. P. Rocha, L. Steels, H. J. van den Herik (Eds.), *Proceedings of the 12th international conference on agents and artificial intelligence, ICAART 2020*, Vol. 2. (pp. 370–377). SCITEPRESS, https://doi.org/10.5220/0008913303700377

Nam, W., Gur, S., Choi, J., et al. (2020). Relative attributing propagation: Interpreting the comparative contributions of individual units in deep neural networks. In *Proceedings of AAAI* (pp. 2501–2508). AAAI Press, https://ojs.aaai.org/index.php/AAAI/article/view/5632

Nguyen, A.M., Yosinski, J., & Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of CVPR. IEEE Computer Society*, pp. 427–436, https://doi.org/10.1109/CVPR.2015.7298640

Nie, X., Hiraga, M., & Ohkura, K. (2019). Visualizing deep Q-learning to understanding behavior of swarm robotic system. In H. Sato, S. Iwanaga & A. Ishii (Eds.) *Proceedings of the 23rd Asia Pacific symposium on intelligent and evolutionary systems*, pp. 118–129. Springer, https://doi.org/10.1007/978-3-030-37442-6_11

Nikou, A., Mujumdar, A., Orlic, M., et al. (2021). Symbolic reinforcement learning for safe RAN control. In F. Dignum, A. Lomuscio, U. Endriss, et al. (Eds.), *AAMAS '21: 20th international conference on autonomous agents and multiagent systems* (pp. 1782–1784). ACM, https://doi.org/10.5555/3463952.3464236, https://www.ifaamas.org/Proceedings/aamas2021/pdfs/p1782.pdf

Nikulin, D., Ianina, A., Aliev, V., et al. (2019). Free-lunch saliency via attention in atari agents. In *2019 IEEE/CVF international conference on computer vision workshops, ICCV Workshops 2019* (pp. 4240–4249). IEEE, https://doi.org/10.1109/ICCVW.2019.00522

Olson, M.L., Neal, L., Li, F., et al. (2019). Counterfactual states for atari agents via generative deep learning. IJCAI 2019 workshop on explainable AI. arxiv:1909.12969

Olson, M. L., Khanna, R., Neal, L., et al. (2021). Counterfactual state explanations for reinforcement learning agents via generative deep learning. *Artificial Intelligence, 295*(103), 455. https://doi.org/10.1016/j.artint.2021.103455

Pace, A., Chan, A., & van der Schaar, M. (2022). POETREE: Interpretable policy learning with adaptive decision trees. In *Proceedings of international conference on learning representations*, https://openreview.net/forum?id=AJsI-ymaKn_

Page, M. J., McKenzie, J. E., Bossuyt, P. M., et al. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *Systematic Reviews, 10*(1), 89. https://doi.org/10.1186/s13643-021-01626-4

Pan, X., Chen, X., Cai, Q., et al. (2019). Semantic predictive control for explainable and efficient policy learning. In *International conference on robotics and automation, ICRA 2019* (pp. 3203–3209). IEEE, https://doi.org/10.1109/ICRA.2019.8794437

Pan, M., Huang, W., Li, Y., et al. (2020). xGAIL: Explainable generative adversarial imitation learning for explainable human decision analysis. In R. Gupta, Y. Liu, J. Tang, et al. (Eds.), *KDD '20: The 26th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 1334–1343). ACM, https://doi.org/10.1145/3394486.3403186

Pankiewicz, N., & Kowalczyk, P. (2022). Attributation analysis of reinforcement learning-based highway driver. *Electronics*. https://doi.org/10.3390/electronics11213599

Paull, L., Tani, J., Ahn, H., et al. (2017). Duckietown: An open, inexpensive and flexible platform for autonomy education and research. In *Proceedings of ICRA* (pp. 1497–1504). IEEE, https://doi.org/10.1109/ICRA.2017.7989179

Portugal, E., Cruz, F., Ayala, A., et al. (2022). Analysis of explainable goal-driven reinforcement learning in a continuous simulated environment. *Algorithms, 15*(3), 91. https://doi.org/10.3390/a15030091

Preece, A.D., Harborne, D., Braines, D., et al. (2018). Stakeholders in explainable AI. AAAI FSS-18: Artificial intelligence in government and public sector. https://doi.org/10.48550/ARXIV.1810.00184

Puiutta, E., & Veith, E.M.S.P. (2020). Explainable reinforcement learning: A survey. In A. Holzinger, P. Kieseberg, A. M. Tjoa, et al. (Eds.), *Machine learning and knowledge extraction - 4th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 international cross-domain conference, CD-MAKE 2020*, Proceedings, Lecture notes in computer science, vol 12279 (pp. 77–95). Springer, https://doi.org/10.1007/978-3-030-57321-8_5

Puri, N., Verma, S., Gupta, P., et al. (2020). Explain your move: Understanding agent actions using specific and relevant feature attribution. In *8th international conference on learning representations, ICLR 2020*. OpenReview.net https://openreview.net/forum?id=SJgzLkBKPB

Qiu, W., & Zhu, H. (2022). Programmatic reinforcement learning without oracles. In *The tenth international conference on learning representations, ICLR 2022*. OpenReview.net, https://openreview.net/forum?id=6Tk2noBdvxt

Ramanishka, V., Chen, Y., Misu, T., et al. (2018). Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning. In *Proceedings of CVPR*. Computer Vision Foundation/IEEE Computer Society (pp. 7699–7707), https://doi.org/10.1109/CVPR.2018.00803

Ras, G., Xie, N., van Gerven, M., et al. (2022). Explainable deep learning: A field guide for the uninitiated. *Journal of Artificial Intelligence Research, 73*, 329–396. https://doi.org/10.1613/jair.1.13200

Remman, S.B., & Lekkas, A.M. (2021). Robotic lever manipulation using hindsight experience replay and shapley additive explanations. In *2021 European control conference, ECC 2021* (pp. 586–593). IEEE, https://doi.org/10.23919/ECC54610.2021.9654850

Ribeiro, M.T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In B. Krishnapuram, M. Shah, A. J. Smola, et al. (Eds.), *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135–1144). ACM, https://doi.org/10.1145/2939672.2939778,

Ribera, M., & Lapedriza, À. (2019). Can we do better explanations? A proposal of user-centered explainable AI. In C. Trattner, D. Parra, N. Riche (Eds.), *Proceedings of ACM IUI workshops, CEUR Workshop Proceedings*, vol 2327. CEUR-WS.org, http://ceur-ws.org/Vol-2327/IUI19WS-ExSS2019-12.pdf

Riegel, R., Gray, A. G., Luus, F. P. S., et al. (2020). Logical neural networks. CoRR. arXiv:2006.13155

Rietz, F., Magg, S., Heintz, F., et al. (2022). Hierarchical goals contextualize local reward decomposition explanations. *Neural Computing and Applications Early Access*. https://doi.org/10.1007/s00521-022-07280-8

Rizzo, S.G., Vantini, G., & Chawla, S. (2019). Reinforcement learning with explainability for traffic signal control. In *2019 IEEE intelligent transportation systems conference, ITSC 2019* (pp. 3567–3572). IEEE, https://doi.org/10.1109/ITSC.2019.8917519

Robbins, B. G. (2016). What is trust? A multidisciplinary review, critique, and synthesis. *Sociology Compass, 10*(10), 972–986. https://doi.org/10.1111/soc4.12391

Robnik-Sikonja, M., & Bohanec, M. (2018). Perturbation-based explanations of prediction models. In J. Zhou & F. Chen (Eds.) *Human and machine learning—visible, explainable, trustworthy and transparent*. Human-Computer Interaction Series (pp. 159–175). Springer, https://doi.org/10.1007/978-3-319-90403-0_9

Rohmer, E., Singh, S.P.N., & Freese, M. (2013). V-REP: A versatile and scalable robot simulation framework. In: *2013 IEEE/RSJ international conference on intelligent robots and systems* (pp. 1321–1326). IEEE, https://doi.org/10.1109/IROS.2013.6696520

Roth, A.M., Liang, J., & Manocha, D. (2021). XAI-N: Sensor-based robot navigation using expert policies and decision trees. In *IEEE/RSJ international conference on intelligent robots and systems, IROS 2021* (pp. 2053–2060). IEEE, https://doi.org/10.1109/IROS51168.2021.9636759

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence, 1*(5), 206–215. https://doi.org/10.1038/s42256-019-0048-x

Rupprecht, C., Ibrahim, C., & Pal, C.J. (2020). Finding and visualizing weaknesses of deep reinforcement learning agents. In *8th international conference on learning representations, ICLR 2020*. OpenReview.net, https://openreview.net/forum?id=rylvYaNYDH

Russell, J., & Santos, E. (2019). Explaining reward functions in markov decision processes. In R. Barták & K. W. Brawner (Eds.), *Proceedings of the thirty-second international florida artificial intelligence research society conference* (pp. 56–61). AAAI Press, https://aaai.org/ocs/index.php/FLAIRS/FLAIRS19/paper/view/18275

Russell, S., & Norvig, P. (2020). *Artificial intelligence: A modern approach* (4th ed.). Pearson.

Sado, F., Loo, C. K., Liew, W. S., et al. (2023). Explainable goal-driven agents and robots–A comprehensive review. *ACM Computing Surveys*. https://doi.org/10.1145/3564240

Sakai, T., Miyazawa, K., Horii, T., et al. (2021). A framework of explanation generation toward reliable autonomous robots. *Advanced Robotics, 35*(17), 1054–1067. https://doi.org/10.1080/01691864.2021.1946423

Sakai, T., & Nagai, T. (2022). Explainable autonomous robots: A survey and perspective. *Advanced Robotics, 36*(5–6), 219–238. https://doi.org/10.1080/01691864.2022.2029720

Santana, E., & Hotz, G. (2016). Learning a driving simulator. CoRR abs/1608.01230. https://doi.org/10.48550/ARXIV.1608.01230

Schmidt, L.M., Kontes, G.D., Plinge, A., et al. (2021). Can you trust your autonomous car? Interpretable and verifiably safe reinforcement learning. In *IEEE intelligent vehicles symposium, IV 2021* (pp. 171–178). IEEE, https://doi.org/10.1109/IV48863.2021.9575328

Schrittwieser, J., Antonoglou, I., Hubert, T., et al. (2020). Mastering atari, go, chess and shogi by planning with a learned model. *Nature, 588*(7839), 604–609. https://doi.org/10.1038/s41586-020-03051-4

Sehnke, F., Osendorfer, C., Rückstieß, T., et al. (2008). Policy gradients with parameter-based exploration for control. In V. Kurková, R. Neruda, J. Koutník (Eds.) *Proceedings of ICANN, LNCS*, vol. 5163 (pp. 387–396). Springer, https://doi.org/10.1007/978-3-540-87536-9_40

Selvaraju, R.R., Cogswell, M., Das, A., et al. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *IEEE International conference on computer vision, ICCV 2017* (pp. 618–626). IEEE Computer Society, https://doi.org/10.1109/ICCV.2017.74,

Seng, D., Zhang, J., & Shi, X. (2021). Visual analysis of deep Q-network. *KSII Transactions on Internet and Information Systems*. https://doi.org/10.3837/tiis.2021.03.003

Sequeira, P., Yeh, E., & Gervasio, M.T. (2019). Interestingness elements for explainable reinforcement learning through introspection. In C. Trattner, D. Parra, N. Riche (Eds.), *Joint proceedings of the ACM IUI 2019 workshops co-located with the 24th ACM conference on intelligent user interfaces (ACM IUI 2019), CEUR workshop proceedings*, vol 2327. CEUR-WS.org, http://ceur-ws.org/Vol-2327/IUI19WS-ExSS2019-1.pdf

Sequeira, P., & Gervasio, M. T. (2020). Interestingness elements for explainable reinforcement learning: Understanding agents' capabilities and limitations. *Artificial Intelligence, 288*(103), 367. https://doi.org/10.1016/j.artint.2020.103367

Shi, S., Li, J., Li, G., et al. (2021a). XPM: An explainable deep reinforcement learning framework for portfolio management. In G. Demartini, G. Zuccon, J. S. Culpepper, et al. (Eds.), *CIKM '21: The 30th ACM international conference on information and knowledge management* (pp. 1661–1670). ACM, https://doi.org/10.1145/3459637.3482494

Shi, W., Huang, G., Song, S., et al. (2021). Temporal-spatial causal interpretations for vision-based reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence Early Access*. https://doi.org/10.1109/TPAMI.2021.3133717

Shi, W., Huang, G., Song, S., et al. (2022). Self-supervised discovering of interpretable features for reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*(5), 2712–2724. https://doi.org/10.1109/TPAMI.2020.3037898

Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning important features through propagating activation differences. In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th international conference on machine learning, ICML 2017, Proceedings of machine learning research*, vol 70 (pp. 3145–3153). PMLR, http://proceedings.mlr.press/v70/shrikumar17a.html

Shu, T., Xiong, C., & Socher, R. (2018). Hierarchical and interpretable skill acquisition in multi-task reinforcement learning. In *6th international conference on learning representations, ICLR 2018, Conference track proceedings*. OpenReview.net, https://openreview.net/forum?id=SJJQVZW0b

Sieusahai, A., & Guzdial, M. (2021). Explaining deep reinforcement learning agents in the atari domain through a surrogate model. In D. Thue & S. G. Ware (Eds.), *Proceedings of the seventeenth AAAI conference on artificial intelligence and interactive digital entertainment, AIIDE 2021* (pp. 82–90). AAAI Press, https://ojs.aaai.org/index.php/AIIDE/article/view/18894

Silva, A., Gombolay, M. C., Killian, T. W., et al. (2020). Optimization methods for interpretable differentiable decision trees applied to reinforcement learning. In S. Chiappa & R. Calandra (Eds.), *The 23rd international conference on artificial intelligence and statistics, AISTATS 2020, Proceedings of machine learning research*, vol 108 (pp. 1855–1865). PMLR, http://proceedings.mlr.press/v108/silva20a.html

Silver, D., Huang, A., Maddison, C. J., et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature, 529*(7587), 484–489. https://doi.org/10.1038/nature16961

Silver, D., Schrittwieser, J., Simonyan, K., et al. (2017). Mastering the game of Go without human knowledge. *Nature, 550*(7676), 354–359. https://doi.org/10.1038/nature24270

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In Y. Bengio & Y. LeCun (Eds.), *3rd international conference on learning representations, ICLR 2015, Conference track proceedings*, https://doi.org/10.48550/ARXIV.1409.1556

Simpson, T. W. (2012). What is trust? *Pacific Philosophical Quarterly, 93*(4), 550–569. https://doi.org/10.1111/j.1468-0114.2012.01438.x

Singh, G., Memoli, F., & Carlsson G. (2007). Topological methods for the analysis of high dimensional data sets and 3D object recognition. In M. Botsch, R. Pajarola, B. Chen, et al. (Eds.), Eurographics symposium on point-based graphics. The Eurographics Association, https://doi.org/10.2312/SPBG/SPBG07/091-100

Skirzynski, J., Becker, F., & Lieder, F. (2021). Automatic discovery of interpretable planning strategies. *Machine Learning, 110*(9), 2641–2683. https://doi.org/10.1007/s10994-021-05963-2

Soares, E. A., Angelov, P. P., Costa, B., et al. (2021). Explaining deep learning models through rule-based approximation and visualization. *IEEE Transactions on Fuzzy Systems, 29*(8), 2399–2407. https://doi.org/10.1109/TFUZZ.2020.2999776

Sovrano, F., Vitali, F., & Palmirani, M. (2020). Making things explainable vs explaining: Requirements and challenges under the GDPR. In V. Rodríguez-Doncel, M. Palmirani, M. Araszkiewicz, et al (Eds.), *Proceedings of AICOL, AICOL, XAILA, LNCS*, vol. 13048 (pp. 169–182). Springer, https://doi.org/10.1007/978-3-030-89811-3_12

Springenberg, J.T., Dosovitskiy, A., Brox, T., et al. (2015). Striving for simplicity: The all convolutional net. In Y. Bengio, Y. LeCun (Eds.), *3rd International conference on learning representations, ICLR 2015, Workshop track proceedings*, arxiv:1412.6806

Sreedharan, S., Soni, U., Verma, M., et al. (2022). Bridging the gap: Providing post-hoc symbolic explanations for sequential decision-making problems with inscrutable representations. In *The tenth international conference on learning representations, ICLR 2022*. OpenReview.net, https://openreview.net/forum?id=o-1v9hdSult

Sreedharan, S., Srivastava, S., & Kambhampati, S. (2020). TLdR: Policy summarization for factored SSP problems using temporal abstractions. In J. C. Beck, O. Buffet, J. Hoffmann, et al. (Eds.) *Proceedings of the thirtieth international conference on automated planning and scheduling* (pp. 272–280). AAAI Press, https://ojs.aaai.org/index.php/ICAPS/article/view/6671

Stork, J., Zaefferer, M., Bartz-Beielstein, T., et al. (2020). Understanding the behavior of reinforcement learning agents. In B. Filipic, E. A. Minisci, M. Vasile (Eds.), *Bioinspired optimization methods and their applications—9th international conference, BIOMA 2020, Proceedings, lecture notes in computer science*, vol 12438 (pp. 148–160). Springer, https://doi.org/10.1007/978-3-030-63710-1_12

Strobelt, H., Gehrmann, S., Pfister, H., et al. (2018). Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE Transactions on Visualization and Computer Graphics, 24*(1), 667–676. https://doi.org/10.1109/TVCG.2017.2744158

Suárez, A., & Lutsko, J. F. (1999). Globally optimal fuzzy decision trees for classification and regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 21*(12), 1297–1311. https://doi.org/10.1109/34.817409

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. In D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th international conference on machine learning, ICML 2017, Proceedings of machine learning research*, vol 70 (pp. 3319–3328). PMLR, http://proceedings.mlr.press/v70/sundararajan17a.html

Suresh, H., Gomez S. R., Nam, K. K., et al. (2021). Beyond expertise and roles: A framework to characterize the stakeholders of interpretable machine learning and their needs. In: Y. Kitamura, A. Quigley, K. Isbister, et al. (Eds.), *Proceedings of CHI* (pp. 74:1–74:16). ACM, https://doi.org/10.1145/3411764.3445088

Sutton, R.S., & Barto, A.G. (2018). *Reinforcement learning an introduction*, Second Edition. Adaptive Computation and nMachine Learning, MIT Press, https://mitpress.mit.edu/books/reinforcement-learning-second-edition

Szegedy, C., Zaremba, W., Sutskever, I., et al. (2014). Intriguing properties of neural networks. In Y. Bengio, Y. LeCun (Eds.), *Proceedings of ICLR*, https://doi.org/10.48550/ARXIV.1312.6199

Tabrez, A., Agrawal, S., & Hayes, B. (2019). Explanation-based reward coaching to improve human performance via reinforcement learning. In *14th ACM/IEEE international conference on human-robot interaction, HRI 2019* (pp. 249–257). IEEE, https://doi.org/10.1109/HRI.2019.8673104

Tang, Y., Nguyen, D., & Ha, D. (2020). Neuroevolution of self-interpretable agents. In C. A. C. Coello (Ed) *GECCO '20: Genetic and evolutionary computation conference*, (pp. 414–424). ACM, https://doi.org/10.1145/3377930.3389847

Terra, A., Inam, R., & Fersman, E. (2022). BEERL: Both ends explanations for reinforcement learning. *Applied Sciences*. https://doi.org/10.3390/app122110947

Todorov, E., Erez, T., & Tassa, Y. (2012). MuJoCo: A physics engine for model-based control. In *Proceedings of IROS* (pp. 5026–5033). IEEE, https://doi.org/10.1109/IROS.2012.6386109

Tolstikhin, I. O., Bousquet, O., Gelly, S., et al. (2018). Wasserstein auto-encoders. In *Proceedings of ICLR*. OpenReview.net, https://openreview.net/forum?id=HkL7n1-0b

Tomsett, R., Braines, D., Harborne, D., et al. (2018). Interpretable to whom? A role-based model for analyzing interpretable machine learning systems. *ICML 2018 workshop on human interpretability in machine learning*. arXiv: 1806.07552

Topin, N., & Veloso, M. (2019). Generation of policy-level explanations for reinforcement learning. In *The thirty-third AAAI conference on artificial intelligence, AAAI 2019, the thirty-first innovative applications of artificial intelligence conference, IAAI 2019, The ninth AAAI symposium on educational advances in artificial intelligence, EAAI 2019* (pp. 2514–2521). AAAI Press, https://doi.org/10.1609/aaai.v33i01.33012514

Topin, N., Milani, S., Fang, F., et al. (2021). Iterative bounding MDPs: Learning interpretable policies via non-interpretable methods. In *Thirty-fifth AAAI conference on artificial intelligence, AAAI 2021, thirty-third conference on innovative applications of artificial intelligence, IAAI 2021, The eleventh symposium on educational advances in artificial intelligence, EAAI 2021* (pp. 9923–9931). AAAI Press, https://ojs.aaai.org/index.php/AAAI/article/view/17192

Trivedi, D., Zhang, J., Sun, S., et al. (2021). Learning to synthesize programs as interpretable and generalizable policies. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, et al. (Eds.), *Advances in neural information processing systems 34: Annual conference on neural information processing systems 2021, NeurIPS 2021* (pp. 25,146–25,163), https://proceedings.neurips.cc/paper/2021/hash/d37124c4c79f357cb02c655671a432fa-Abstract.html

Tylkin, P., Wang, T., Palko, K., et al. (2022). Interpretable autonomous flight via compact visualizable neural circuit policies. *IEEE Robotics and Automation Letters, 7*(2), 3265–3272. https://doi.org/10.1109/LRA.2022.3146555

Ullauri, J. M. P., García-Domínguez, A., Bencomo, N., et al. (2022). Event-driven temporal models for explanations—ETeMoX: Explaining reinforcement learning. *Software and Systems Modeling, 21*(3), 1091–1113. https://doi.org/10.1007/s10270-021-00952-4

van Baar, J., Sullivan, A., Cordorel, R., et al. (2019). Sim-to-real transfer learning using robustified controllers in robotic tasks involving complex dynamics. In *Proceedings of of ICRA*. IEEE, pp 6001–6007, https://doi.org/10.1109/ICRA.2019.8793561

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *JMLR, 9*(86), 2579–2605.

van der Waa, J., van Diggelen, J., van den Bosch, K., et al. (2018). *Contrastive explanations for reinforcement learning in terms of expected consequences*. IJCAI Workshop on XAI abs/1807.08706. https://doi.org/10.48550/ARXIV.1807.08706

Vasic, M., Petrovic, A., Wang, K., et al. (2022). MoËT: Mixture of Expert Trees and its application to verifiable reinforcement learning. *Neural Networks, 151*, 34–47. https://doi.org/10.1016/j.neunet.2022.03.022

Verma, A., Le, H. M., Yue, Y., et al. (2019). Imitation-projected programmatic reinforcement learning. In H. M. Wallach, H. Larochelle, A. Beygelzimer, et al. (Eds.), *Advances in neural information processing systems 32: Annual conference on neural information processing systems 2019, NeurIPS 2019* (pp. 15,726–15,737), https://proceedings.neurips.cc/paper/2019/hash/5a44a53b7d26bb1e54c05222f186dcfb-Abstract.html

Verma, A., Murali, V., Singh, R., et al. (2018). Programmatically interpretable reinforcement learning. In J. G. Dy & A. Krause (Eds.), *Proceedings of the 35th international conference on machine learning, ICML 2018, Proceedings of machine learning research*, vol 80. (pp. 5052–5061). PMLR, http://proceedings.mlr.press/v80/verma18a.html

Videau, M., Leite, A., Teytaud, O., et al. (2022). Multi-objective genetic programming for explainable reinforcement learning. In E. Medvet, G. L. Pappa, B. Xue (Eds.) *Genetic programming—25th European conference, EuroGP 2022* Proceedings, Lecture notes in computer science, vol. 13223 (pp. 278–293). Springer, https://doi.org/10.1007/978-3-031-02056-8_18

Vinyals, O., Babuschkin, I., Chung, J., et al. (2019a). *AlphaStar: Mastering the real-time strategy game StarCraft II*. https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/

Vinyals, O., Babuschkin, I., Czarnecki, W. M., et al. (2019). Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature, 575*(7782), 350–354. https://doi.org/10.1038/s41586-019-1724-z

Vouros, G. A. (2022). Explainable deep reinforcement learning: State of the art and challenges. *ACM Computing Surveys*. https://doi.org/10.1145/3527448

Wang, X., Liu, Y., Chang, Y., et al. (2022). Incorporating explanations to balance the exploration and exploitation of deep reinforcement learning. In G. Memmi, B. Yang, L. Kong, et al. (Eds.), *Knowledge science, engineering and management—15th international conference, KSEM 2022*, Proceedings, Part II, Lecture notes in computer science, vol. 13369 (pp. 200–211). Springer, https://doi.org/10.1007/978-3-031-10986-7_16

Wang, Y., Mase, M., Egi, M. (2020). Attribution-based salience method towards interpretable reinforcement learning. In A. Martin, K. Hinkelmann, H. Fill, et al. (Eds.), *Proceedings of the AAAI 2020 spring symposium on combining machine learning and knowledge engineering in practice, AAAI-MAKE 2020*, Volume I, CEUR Workshop Proceedings, vol. 2600. CEUR-WS.org, http://ceur-ws.org/Vol-2600/short4.pdf

Wang, Z., Schaul, T., Hessel, M., et al. (2016). Dueling network architectures for deep reinforcement learning. In M. Balcan, K. Q. Weinberger (Eds.), *Proceedings of ICML, JMLR Workshop and Conference Proceedings*, vol. 48 (pp. 1995–2003). JMLR.org, http://proceedings.mlr.press/v48/wangf16.html

Wang, X., Yuan, S., Zhang, H., et al. (2019b). Verbal explanations for deep reinforcement learning neural networks with attention on extracted features. In *28th IEEE international conference on robot and human interactive communication, RO-MAN 2019* (pp. 1–7). IEEE, https://doi.org/10.1109/RO-MAN46459.2019.8956301

Wang, H., Gao, H., Yuan, S., et al. (2021). Interpretable decision-making for autonomous vehicles at highway on-ramps with latent space reinforcement learning. *IEEE Transactions on Vehicular Technology, 70*(9), 8707–8719. https://doi.org/10.1109/TVT.2021.3098321

Wang, J., Gou, L., Shen, H., et al. (2019). DQNViz: A visual analytics approach to understand deep Q-networks. *IEEE Transactions on Visualization and Computer Graphics, 25*(1), 288–298. https://doi.org/10.1109/TVCG.2018.2864504

Wang, J., Gou, L., Yang, H., et al. (2018). GANViz: A visual analytics approach to understand the adversarial game. *IEEE Transactions on Visualization and Computer Graphics, 24*(6), 1905–1917. https://doi.org/10.1109/TVCG.2018.2816223

Wang, J., Zhang, W., Yang, H., et al. (2021). Visual analytics for RNN-based deep reinforcement learning. *IEEE Transactions on Visualization and Computer Graphics Early Access*. https://doi.org/10.1109/TVCG.2021.3076749

Watkins, O., Huang, S., Frost, J., et al. (2021). Explaining robot policies. *Applied AI Letters, 2*(4), e52. https://doi.org/10.1002/ail2.52

Wattenberg, M., Viégas, F., & Johnson, I. (2016). How to Use t-SNE effectively. *Distill*. https://doi.org/10.23915/distill.00002

Wei, J., Qiu, Z., Wang, F., et al. (2022). Understanding via exploration: Discovery of interpretable features with deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*. https://doi.org/10.1109/TNNLS.2022.3184956

Weitkamp, L., van der Pol, E., & Akata, Z. (2018). Visual rationalizations in deep reinforcement learning for atari games. In M. Atzmueller & W. Duivesteijn (Eds.), *Artificial intelligence—30th Benelux conference, BNAIC 2018, Communications in computer and information science*, vol. 1021 (pp. 151–165). Springer, https://doi.org/10.1007/978-3-030-31978-6_12

Weller, A. (2017). Challenges for transparency. ICML Workshop on WHI . https://doi.org/10.48550/ARXIV.1708.01870, arXiv:1708.01870

Wells, L., & Bednarz, T. (2021). Explainable AI and reinforcement learning—A systematic review of current approaches and trends. *Frontiers in Artificial Intelligence, 4*(550), 030. https://doi.org/10.3389/frai.2021.550030

Wiegreffe, S., & Pinter, Y. (2019). Attention is not not explanation. In K. Inui, J. Jiang, V. Ng, et al. (Eds.), *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing, EMNLP-IJCNLP 2019*. Association for computational linguistics (pp. 11–20), https://doi.org/10.18653/v1/D19-1002

Wirth, C., Akrour, R., Neumann, G., et al. (2017). A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research, 18*(136), 1–46.

Wollenstein-Betech, S., Muise, C., Cassandras, C. G., et al. (2020). Explainability of intelligent transportation systems using knowledge compilation: a traffic light controller case. In *23rd IEEE international conference on intelligent transportation systems, ITSC 2020* (pp. 1–6). IEEE, https://doi.org/10.1109/ITSC45102.2020.9294213

Wu, B., Gupta, J. K., & Kochenderfer, M. J. (2020). Model primitives for hierarchical lifelong reinforcement learning. *Autonomous Agents and Multi-Agent Systems, 34*(1), 28. https://doi.org/10.1007/s10458-020-09451-0

Wymann, B., Espié, E., Guionneau, C., et al. (2014). TORCS, The open racing car simulator. http://www.torcs.org

Xie, Y., Vosoughi, S., & Hassanpour, S. (2022). Towards interpretable deep reinforcement learning models via inverse reinforcement learning. In *Proceedings of ICPR*arXiv:2203.16464

Xu, H., Gao, Y., Yu, F., et al. (2017). End-to-End Learning of Driving Models from Large-Scale Video Datasets. In: Proc. of CVPR. IEEE Computer Society, pp 3530–3538, https://doi.org/10.1109/CVPR.2017.376

Yang, J., Lee, G., Chang, S., et al. (2019). Towards governing agent's efficacy: Action-conditional β -VAE for deep transparent reinforcement learning. In W. S. Lee & T. Suzuki (Eds.), *Proceedings of the 11th Asian conference on machine learning, ACML 2019, Proceedings of machine learning research*, vol. 101 (pp. 32–47). PMLR, http://proceedings.mlr.press/v101/yang19a.html

Yau, H., Russell, C., & Hadfield, S. (2020). What did you think would happen? Explaining agent behaviour through intended outcomes. In H. Larochelle, M. Ranzato, R. Hadsell, et al. (Eds.), *Advances in neural information processing systems 33: Annual conference on neural information processing systems 2020, NeurIPS 2020*, https://proceedings.neurips.cc/paper/2020/hash/d5ab8dc7ef67ca92e41d730982c5c602-Abstract.html

Ye, X., & Yang, Y. (2021). Efficient robotic object search via HIEM: Hierarchical policy learning with intrinsic-extrinsic modeling. *IEEE Robotics and Automation Letters, 6*(3), 4425–4432. https://doi.org/10.1109/LRA.2021.3068906

Zahavy, T., Ben-Zrihem, N., & Mannor, S. (2016). Graying the black box: Understanding DQNs. In M. Balcan & K. Q. Weinberger (Eds.), *Proceedings of the 33nd international conference on machine learning, ICML 2016*, JMLR workshop and conference proceedings, vol. 48 (pp. 1899–1908). JMLR.org, http://proceedings.mlr.press/v48/zahavy16.html

Zahavy, T., Ben-Zrihem, N., & Mannor, S. (2017). Graying the black box: Understanding DQNs. CoRR, arXiv:1602.02658.

Zambaldi, V.F., Raposo, D., Santoro, A., et al. (2019). Deep reinforcement learning with relational inductive biases. In *7th international conference on learning representations, ICLR 2019*. OpenReview.net, https://openreview.net/forum?id=HkxaFoC9KQ

Zeiler, M.D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In D. J. Fleet, T. Pajdla, B. Schiele, et al. (Eds.) *Proceedings of ECCV*, Lecture notes in computer science, vol. 8689 (pp. 818–833). Springer, https://doi.org/10.1007/978-3-319-10590-1_53

Zelvelder, A. E., Westberg, M., & Främling, K. (2021). Assessing explainability in reinforcement learning. In D. Calvaresi, A. Najjar, M. Winikoff, et al. (Eds.), *Explainable and transparent AI and multi-agent systems—third international workshop, EXTRAAMAS 2021*, Lecture notes in computer science, vol. 12688 (pp. 223–240). Springer, https://doi.org/10.1007/978-3-030-82017-6_14

Zhang, L., Li, X., Wang, M., et al. (2021b). Off-policy differentiable logic reinforcement learning. In N. Oliver, F. Pérez-Cruz, S. Kramer, et al. (Eds.), *Machine learning and knowledge discovery in databases. Research Track - European Conference, ECML PKDD 2021*, Proceedings, Part II, Lecture notes in computer science, vol. 12976 (pp. 617–632). Springer, https://doi.org/10.1007/978-3-030-86520-7_38

Zhang, R., Walshe, C., Liu, Z., et al. (2020b). Atari-HEAD: Atari human eye-tracking and demonstration dataset. In *The thirty-fourth AAAI conference on artificial intelligence, AAAI 2020, the thirty-second innovative applications of artificial intelligence conference, IAAI 2020, The tenth AAAI symposium on educational advances in artificial intelligence, EAAI 2020* (pp. 6811–6820). AAAI Press, https://ojs.aaai.org/index.php/AAAI/article/view/6161

Zhang, K., Wang, Y., Du, J., et al. (2021a). *Identifying decision points for safe and interpretable reinforcement learning in hypotension treatment*. NeurIPS Workshop on Machine Learning for Health. arXiv:2101.03309

Zhang, Q., Ma, X., Yang, Y., et al. (2021). Learning to discover task-relevant features for interpretable reinforcement learning. *IEEE Robotics and Automation Letters, 6*(4), 6601–6607. https://doi.org/10.1109/LRA.2021.3091885

Zhang, K., Zhang, J. J., Xu, P., et al. (2022). Explainable AI in deep reinforcement learning models for power system emergency control. *IEEE Transactions on Computational Social Systems, 9*(2), 419–427. https://doi.org/10.1109/TCSS.2021.3096824

Zhang, H., Zhou, A., & Lin, X. (2020). Interpretable policy derivation for reinforcement learning based on evolutionary feature synthesis. *Complex & Intelligent Systems, 6*(3), 741–753. https://doi.org/10.1007/s40747-020-00175-y

Zhou, B., Khosla, A., Lapedriza, À., et al. (2016). Learning deep features for discriminative localization. In *2016 IEEE conference on computer vision and pattern recognition, CVPR 2016*. IEEE Computer Society (pp. 2921–2929), https://doi.org/10.1109/CVPR.2016.319,

Zhu, Y., Yin, X., Li, R., et al. (2021). Extracting decision tree from trained deep reinforcement learning in traffic signal control. In *2021 international conference on cyber-physical social intelligence (ICCSI)* (pp. 1–7), https://doi.org/10.1109/ICCSI53130.2021.9736263