Explainability in Deep Reinforcement Learning

Alexandre Heuillet[*], Fabien Couthouis[†], Natalia Díaz-Rodríguez[‡]

August 2020

Abstract

A large set of the explainable Artificial Intelligence (XAI) literature is emerging on feature relevance techniques to explain a deep neural network (DNN) output or explaining models that ingest image source data. However, assessing how XAI techniques can help understand models beyond classification tasks, e.g. for reinforcement learning (RL), has not been extensively studied. We review recent works in the direction to attain Explainable Reinforcement Learning (XRL), a relatively new subfield of Explainable Artificial Intelligence, intended to be used in general public applications, with diverse audiences, requiring ethical, responsible and trustable algorithms. In critical situations where it is essential to justify and explain the agent's behaviour, better explainability and interpretability of RL models could help gain scientific insight on the inner workings of what is still considered a black box. We evaluate mainly studies directly linking explainability to RL, and split these into two categories according to the way the explanations are generated: transparent algorithms and post-hoc explainaility. We also review the most prominent XAI works from the lenses of how they could potentially enlighten the further deployment of the latest advances in RL, in the demanding present and future of everyday problems.

Keywords: Reinforcement Learning, Explainable Artificial Intelligence, Machine Learning, Deep Learning, Responsible Artificial Intelligence, Representation Learning, Hierarchical RL, Saliency maps

[*]ENSEIRB-MATMECA, Bordeaux INP, France, aheuillet@enseirb-matmeca.fr, Equal contribution

[†]ENSC, Bordeaux INP, France, fcouthouis@ensc.fr, Equal contribution

[‡]ENSTA Paris, Institut Polytechnique Paris, Inria Flowers Team, France, natalia.diaz@ensta-paris.fr

1 Introduction

During the past decade, Artificial Intelligence (AI), and by extension Machine Learning (ML), have seen an unprecedented rise in both industry and research. The progressive improvement of computer hardware associated with the need to process larger and larger amounts of data made these underestimated techniques shine under a new light. Reinforcement Learning (RL) focuses on learning how to map situations to actions, in order to maximize a numerical reward signal [102]. The learner is not told which actions to take, but instead must discover which actions are the most rewarding by trying them. Reinforcement learning addresses the problem of how agents should learn a policy that take actions to maximize the cumulative reward through interaction with the environment [31].

Recent progress in Deep Learning (DL) for learning feature representations has significantly impacted RL, and the combination of both methods (known as deep RL) has led to remarkable results in a lot of areas. Typically, RL is used to solve optimisation problems when the system has a very large number of states and has a complex stochastic structure. Notable examples include training agents to play Atari games based on raw pixels [75, 76], board games [96, 97], complex real-world robotics problems such as manipulation [8] or grasping [54] and other real-world applications such as resource management in computer clusters [72], network traffic signal control [9], chemical reactions optimization [117] or recommendation systems [116].

The success of Deep RL could augur an imminent arrival in the industrial world. However, like many Machine Learning algorithms, RL algorithms suffer from a lack of explainability. This defect can be highly crippling as many promising RL applications (defense, finance, medicine, etc.) need a model that can explain its decisions and actions to human users [42] as a condition to their full acceptation by society. Furthermore, deep RL models are complex to debug for developers, as they rely on many factors: environment (in particular the design of the reward function), observations encoding, large DL models and the algorithm used to train the policy. Thus, an explainable model could aid fixing problems quicker and drastically speed up new development in RL methods. Those last two points are the main arguments in favor of the necessity of explainable reinforcement learning (XRL).

While explainability starts being well developed for standard ML models and neural networks [86, 69, 92], the particular domain of RL has yet many intricacies to be better understood: both in terms of its functioning, and in terms of conveying the decisions of an RL model to different audiences. The difficulty lies in the very recent human-level performance of deep RL algorithms and by their complexity, normally parameterized with thousands if not millions of parameters [16]. The present work intends to provide a non-exhaustive state-of-the-art review on explainable reinforcement learning, highlighting the main methods that we envision most promising. In the following, we will briefly recall some important concepts in XAI.

1.1 Explainable AI: Audience

Explaining a Machine Learning model may involve different goals: trustworthiness, causality, transferability, informativeness, fairness, confidence, accessibility, interactivity and privacy awareness. These goals have to be taken into account while explaining a model because the expected type of explanation may differ, depending on the pursued objective. For example, a saliency map explaining what is recognised as a dog on an input image does not tell us much about privacy awareness. In addition, each goal may be a dimension of interest, but only for a certain audience (the public to whom the explanations will be addressed). Indeed, the transferability of a model can be significative for a developer, since he/she can save time by training only one model for different tasks, while the user will not be impacted, if not aware, by this aspect.

The understandability of an ML model therefore depends on its transparency (its capacity to be understandable by itself) but also on human understanding. According to these considerations, it is essential to take into account the concept of *audience*, as the intelligibility and comprehensibility of a model is dependant on the goals and the cognitive skills of its users. Barredo Arrieta et al. [10] discuss these aspects with additional details.

Target audience	**Description**	**Explainability purposes**	**Pursued goals**
Experts	Domain experts, model users (e.g. medical doctors, insurance agents)	Trust the model itself, gain scientific knowledge	Trustworthiness, causality, transferability, informativeness, confidence, interactivity
Users	Users affected by model decisions	Understand their situation, verify fair decisions.	Trustworthiness, informativeness, fairness, accessibility, interactivity, privacy awareness
Developers	Developers, researchers, data scientists, product owners...	Ensure and improve product efficiency, research, new functionalities...	Transferability, informativeness, confidence
Executives	Managers, executive board members...	Assess regulatory compliance, understand corporate AI applications...	Causality, informativeness, confidence
Regulation	Regulatory entities/agencies	Certify model compliance with the legislation in force, audits, ...	Causality, informativeness, confidence, fairness, privacy awareness

Table 2: Target audience in XAI. This table shows the different objectives of explainability in Machine Learning models for different audience profiles. Inspired from the diagram presented in Barredo Arrieta et al. [10].

1.2 Evaluating explanations

The broad concept of evaluation is based on metrics aiming to compare how well a technique performs compared to another. In the case of model explainability, metrics should evaluate how well a model fits the definition of explainable and how well performs in a certain aspect of explainability.

Explanation evaluation in XAI has proven to be quite a challenging task. First because the concept of explainability in Machine Learning is not well or uniformly accepted by the community and there is not a clear definition and thus, not a clear consensus on which metrics to use. Secondly because an explanation is relative to a specific audience, which is sometimes difficult to deal with (in particular when this specific audience is composed of domain experts who can be hard to involve in a testing phase). Thirdly, because the quality of an explanation is always qualitative and subjective, since it depends on the audience, the pursued goal and even the human variability as two people can have a different level of understandability for the same explanation. That is why user studies are so popular to evaluate explanations as it makes possible to convert qualitative evaluations into quantitative ones, by asking questions on the accuracy and clarity of the explanation such as "Does this explanation allow you to understand why the model predicted that this image is a dog? Did the context helped the model?"... etc. Generally in XAI, there is only a single model to explain at a time; however, it is more complicated in XRL, as we generally want to explain a policy, or "Why the agent took action x in state s?".

Doshi-Velez et al. [30] propose an attempt to formulate some approaches to evaluate XAI methods. The authors introduce three main levels to evaluate the quality of the explanations provided by an XAI method. The table below summarizes them.

Level of evaluation	**Type of task**	**Required humans**	**Modus Operandi**
Application level	Real task	Domain expert	Put the explanation into the product and have it tested by the end user.
Human level	Simplified task	Layperson	Carry out the application level experiments with laypersons as it makes the experiment cheaper and it is easier to find more testers.
Function level	Proxy task	No human required	Uses a proxy to evaluate the explanation quality. Works best when the model class used has already been evaluated by someone else in human level. For instance, a proxy for decision trees can be the depth of the tree.

Table 4: Three main levels for evaluating the explanations provided by an XAI method (inspired from explanations provided in Doshi-Velez et al. [30]).

A common example of evaluation of an application level or human level task is to evaluate the quality of the mental model built by the user after seeing the explanation(s). Mental models can be described as internal representations, built upon experiences, and which allow to mentally simulate how something works in the real world. Hoffman et al. [48] propose to evaluate mental models by 1. Asking post-task questions on the behaviour of the agent (such as "How does it work?" or "What does it achieve?") and 2. Asking the participants to make predictions on the agent's next action. These evaluations are often done using Likert scales.

2 XAI in RL: State of the art and reviewed literature

We reviewed the state of the art on XRL and summarized it in Table 5. This table presents, for each paper, the task(s) for which an explanation is provided, the employed RL algorithms (whose Algorithms glossary can be found in the 6), and the provided type of explanations, i.e.: based on images, diagrams (graphical components such as barchart, plot, graph), or text. We also present the level of the provided explanation (*local* if it explains only predictions, *global* if it explains the whole model), and the audience concerned by the explanation, as discussed in Section 1.1.

Reference	**Task/ Environment**	**Decision process**	**Algorithm(s)**	**Explanation type (Level)**	**Target**
Relational Deep RL [111]	Planning + strategy games (Box-World/ Starcraft II)	POMDP	IMPALA	Images (Local)	Experts
Symbolic RL with Common Sense [24]	Game (object retrieval)	POMDP	SRL+CS, DQL	Images (Global)	Experts
Decoupling feature extraction from policy learning [84]	Robotics (grasping), and navigation	MDP	PPO	Diagram (state plot & image slider (Local)	Experts
Explainable RL via Reward Decomposition [53]	Game (grid and landing)	MDP	HRA, SARSA, Q-learning	Diagrams (Local)	Experts, Users, Executives
Explainable RL Through a Causal Lens [71]	Games (OpenAI benchmark and Starcraft II)	Both	PG, DQN, DDPG, A2C, SARSA	Diagrams, Text (Local)	Experts, Users, Executives

continues on next page

Shapley Q-value: A Local Reward Approach to Solve Global Reward Games [107]	Multiagents (Cooperative Navigation, Prey-and-Predator and Traffic Junction)	POMDP	DDPG	Diagrams (Local)	Experts
Dot-to-Dot: Explainable HRL For Robotic Manipulation [15]	Robotics (grasping)	MDP	DDPG, HER, HRL	Diagrams (Global)	Experts, Developers
Self-Educated Language Agent With HER For Instruction Following [23]	Instruction Following (MiniGrid)	MDP	Textual HER	Text (Local)	Experts, Users, Developers
Commonsense and Semantic-guided Navigation [110]	Room navigation	POMDP	-	Text (Global)	Experts
Boolean Task Algebra [103]	Game (grid)	MDP	DQN	Diagrams	Experts
Visualizing and Understanding Atari [41]	Games (Pong, Breakout, Space Invaders)	MDP	A3C	Images (Global)	Experts, Users, Developers
Interestingness Elements for XRL through Introspection [94, 93]	Arcade game (Frogger)	POMDP	Q-Learning	Images (Local)	Users
Composable DRL for Robotic Manipulation [43]	Robotics (pushing and reaching)	MDP	Soft Q-learning	Diagrams (Local)	Experts
Symbolic-Based Recognition of Contact States for Learning Assembly Skills [109]	Robotic grasping	POMDP	HMM, PAA, K-means	Diagrams (Local)	Experts

continues on next page

Safe Reinforcement Learning with Model Uncertainty Estimates [70]	Collision avoidance	POMDP	Monte Carlo Dropout, bootstrapping	Diagrams (Local)	Experts

Table 5: Summary of reviewed literature on explainable RL (XRL) and deep RL (DRL).

In Table 5 we summarized the literature focusing on explainable fundamental RL algorithms. However, we also reviewed articles about state of the art XAI techniques that can be used in the context of current RL which we did not include in Table 5. Next, we will describe the main ideas provided by these papers which can help bring explainability in RL. It is possible to classify all recent studies in two main categories: transparent methods and Post-Hoc explainability according to the XAI taxonomies in Barredo Arrieta et al. [10]. On the one hand, inherently transparent algorithms include by definition every algorithm which is understandable by itself, such as a decision-trees. On the other hand, Post-Hoc explainability includes all methods that provide explanations of an RL algorithm after its training, such as SHAP (SHapley Additive exPlanations) [69] or LIME [86] for standard ML models. Reviewed papers are referenced by type of explanation in Figure 2.

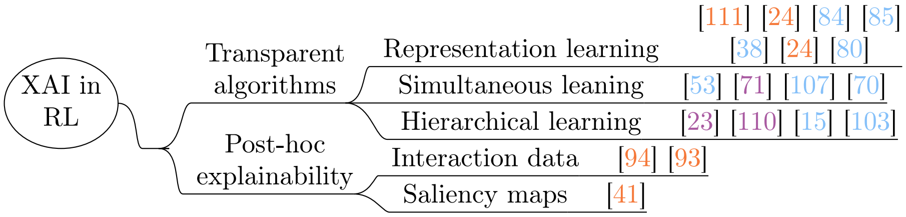


Figure 1: Taxonomy of the reviewed literature identified for bringing explainability to RL models. References in orange, purple, and light blue correspond to XAI techniques using images, text or diagrams, respectively.

2.1 Transparent algorithms

Transparent algorithms are well known and used in standard Machine Learning (e.g., linear regression, decision trees or rule-based systems). Their strength lie in the fact that they are designed to have a transparent architecture that makes them explainable by themselves, without the need of any external processing. However, it is quite different for RL, as standard DRL algorithms (e.g., DQN, PPO, DDPG, A2C...) are not transparent by nature. In addition, the large majority of studies related to transparency in XRL chose to build algorithms targeting only a specific task. Nonetheless, most of the time and contrary to standard Machine Learning models, transparent RL algorithms can achieve state of the art performance in these specific tasks [107, 24, 111].

2.1.1 Explanation through representation learning

Representation learning algorithms focuses on learning abstract features that characterize data, in order to make it easier to extract useful information when building predictors [63, 13]. These learned features have the advantage of having low dimensionality, which generally improves training speed and generalization of Deep Learning models [62, 63, 84].

In the context of RL, learning representations of states, actions or policy can be useful to explain a RL algorithm, as these representations can bring some clues on the functioning of the algorithm. Indeed, State Representation Learning (SRL) [63] is a particular type of representation learning that aims at building a low-dimensional and meaningful representation of a state space, by processing high-dimensional raw observation data (e.g., learn a position (x, y) from raw image pixels). This enables to capture the variations in the environment influenced by the agent's actions and thus, extrapolate explanations. SRL can be especially useful in RL for robotics and control [85, 84, 106, 28, 29], and can help to understand how the agent interprets the observations and what is relevant to learn to act, i.e., actionable or controllable features [62]. Indeed, the dimensionality reduction induced by SRL, coupled with the link to the control and possible disentanglement of variation factors, could be highly beneficial to improve our understanding capacity of the decisions made by RL algorithms using a state representation method [63]. For example, SRL can be used to split the state representation [84] according to the different training objectives to be optimized before learning a policy. This allows to allocate room for encoding each necessary objective within the embedding state to be learned (in that case, reward prediction, a reconstruction objective and an inverse model). In this context, tools such as S-RL Toolbox [85] allow sampling from the embedding state space (learned through SRL) to allow a visual interpretation of the model's internal state, and pairing it with its associated input observation. Comprehensibility is thus enhanced, more easily observing if smoothness is preserved in the state space, as well as whether other invariants related to learning specific control task are guaranteed.

There are several approaches employed for SRL: reconstructing the observations using autoencoders [6, 32], training a forward model to predict next state [49, 80], teach to an inverse model how to predict actions from previous state(s) [95, 80] or using prior knowledge to constrain the state space [52, 62].

Along the same lines, learning disentangled representations [47, 2, 4, 17] is another interesting idea used for unsupervised learning, which decomposes (or disentangles) each feature into narrowly defined variables and encodes them as separate low-dimensional features (generally using a Variational Autoencoders [59]). It is also possible to make use of this concept, as well as lifelong learning to learn more interpretable representations on unsupervised classification tasks. In addition, one could argue that learning through life would allow compacting and updating old knowledge with new one while preventing catastrophic forgetting [64]. Thus, this is a key concept that could lead to more versatile RL agents, being able to learn new tasks without forgetting the previous ones. Information Maximizing Generative Adversarial Networks (InfoGAN) [20] is another model based on the principles of learning disentangled representations. The noise vector used in traditional GANs is decomposed into two parts: z: incompressible noise; and c: the latent code used to target the salient semantic features of the data distribution. The main idea is to feed z and c to the generator G, to maximize the mutual information between c and $G(z, c)$, in order to assure that the information contained in c is preserved during the generation process. As a result, the InfoGAN model is able to create an interpretable representation via the latent code c (i.e., the values changing according to shape and features of the input data).

Some work has been done to learn representations by combining symbolic AI with deep RL in order to facilitate the use of background knowledge, the exploitation of learnt knowledge, and to improve generalization [38, 35, 88, 37] Consequently, it also improves the explainability of the algorithms, while preserving state-of-the-art performance.

Zambaldi et al. [111] propose making use of Inductive Logic Programming and self-attention to represent states, actions and policies using first order logic, using a mechanism similar to graph neural networks and more generally, message passing computations [25, 60, 12, 89]. In these kind of models entity-entity relations are explicitly computed when considering the messages passed between connected nodes of the graph as shown in Fig. 2. Self-attention is used here as a method to compute interactions between these different entities (i.e. relevant pixels in a RGB image for the example from [111]), and thus perform non-local pairwise relational computations. This technique allows an expert to visualize the agent's attention weights associated to its available actions and interpret how to improve the understanding of its strategy.

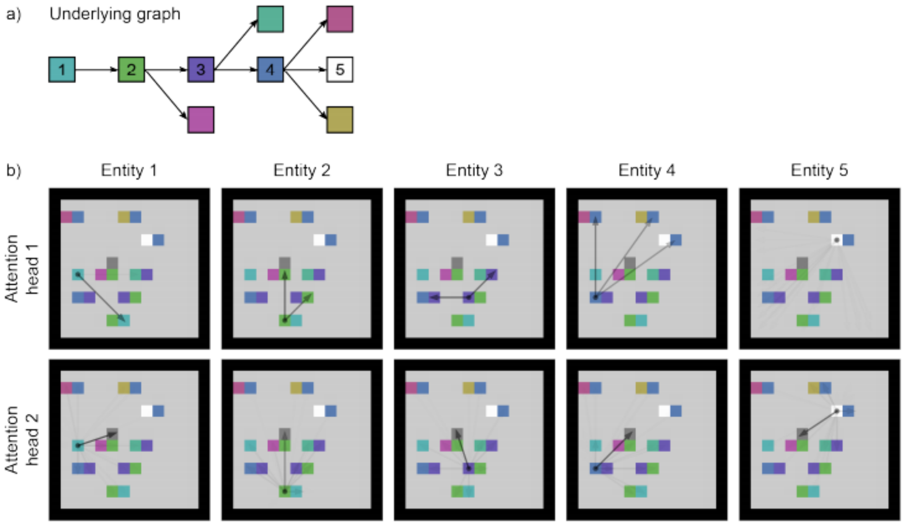


Figure 2: Visualization of attention weights for the *Box-World* task (environment created by authors of [111]), in which the agent has to open boxes to obtain either keys or rewards. **(a)** The underlying graph of one example level. **(b)** The result of the analysis for that level, using each entity (represented as colored pixels) along the solution path (1–5) as the source of attention. Boxes are represented by two adjacent colored pixels. On each box, the pixel on the right represents the box's lock and its color indicates which key can be used to open it. The pixel on the left indicates the content of the box which is inaccessible while the box is locked. Arrows point to the entities that the source is attending to. The arrow's transparency is determined by the corresponding attention weight (reproduced with permission of Vinicius Zambaldi [111]).

Another work that aims to incorporate common sense to the agent, in terms of symbolic abstraction to represent the problem, is in [24]. This method subdivides the world state representation into many sub-states, with a degree of associated importance based on how far the object is from the agent. This helps understand the relevance of the actions taken by the agent by determining which sub-states were chosen.

2.1.2 Simultaneous learning of the explanation and the policy

While standard DRL algorithms struggle to provide explanations, those can be tweaked to learn simultaneously both policy and explanation. Thus, explanations become a learned component of the model. These methods are recommended on specific problems where it is possible to introduce knowledge, such classifying rewards by types, adding relationships between states, etc... Thus, tweaking the algorithm to introduce some task knowledge and to learn explanations generally also improves performance. A general notion is that the knowledge gained from the auxiliary task objective must be useful for downstream tasks. In this direction, Juozapaitis et al. [53] introduced reward decomposition, whose main principle is to decompose the reward function into a sum of meaningful reward types. Authors used reward decomposition to improve performance on Cliffworld and Starcraft II, where each action can be classified according to its type. This method consists of using a custom *decomposed reward DQN* by defining a vector-valued reward function, where each component is the reward for a certain type so that actions can be compared in terms of trade-offs among the types. In the same way, the Q-function is also vector valued and each component gives action values that account for only a reward type. The sum of each of those vector-valued functions gives the overall Q or reward function. Learning multiple Q-functions, one for each type of reward, allows the model to learn the best policy while also learning the explanations (i.e. the type of reward that the agent wanted to maximize by his action, illustrated on Fig. 4). They introduce the concept of *Reward Difference Explanation* (RDX, in Fig. 3) which enables to understand the reasons why an action has an advantage (or disadvantage) over another. They also define *Minimal Sufficient Explanations* (MSX, See Fig. 4), in order to help humans identify a small set of the most important *reasons* why the agent choose specific actions over another. MSX+ and MSX- are sets of critical positive and negative reasons (respectively) for the actions preferred by the agent.

While reward decompositions help to understand the agent choice preferences between several actions, minimal sufficient explanations are used to help selecting the most important reward decompositions. Other works that faciliate the explainability of RL models by using reward-based losses for more interpretable RL are in [112, 95, 80].

In the same vein, Madumal et al. [71] use the way humans understand and represent knowledge through causal relationships and introduce an *action influence model*: a causal model which can explain the behaviour agents using causal explanations. Structural causal models [45] represent the world using random variables, some of which might have causal relationships, which can be described thanks to a set of structural equations. In this work, structural causal models are extended to include actions as part of the causal relationships. An action influence model is a tuple represented by the state-actions ensemble and the corresponding set of structural equations. The whole process is divided into 3 phases:

- Defining the qualitative causal relationships of variables as an action influence model.
- Learning the structural equations (as multivariate regression models during the training phase of the agent).

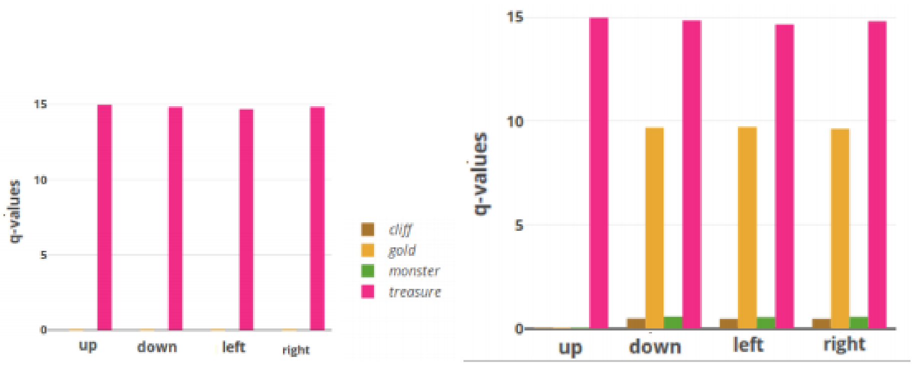


Figure 3: **Left** Reward Decompositions for DQN. **Right** Hybrid Reward Architecture (HRA) at cell (3,4) in Cliffworld. HRA predicts an extra “gold” reward for actions which do not lead to a terminal state. Reproduced with permission of Zoe Juozapaitis [53].

- Generating explanations, called *explanans*, by traversing the action influence graph (see Figure 5) from the root to the leaf reward node.

This kind of models allow encoding cause-effect relations between events (actions and states) as shown by the graph featured in Figure 5. Thus, they can be used to generate explanations of the agent behaviour ("why" and "why not" questions), based on knowledge about how actions influence the environment. Their method was evaluated through a user study showing that, compared to video game playing without any explanations and relevant variable explanations, this model performs significantly better on 1) task prediction and 2) explanation *goodness*. However, trust was not shown to be significantly improved.

Authors of [70] also learn explanations along with the model policy on pedestrians collision avoidance tasks. In this paper, an ensemble of LSTM networks was trained using Monte Carlo Dropout [34] and bootstrapping [79] to estimate collision probabilities and thus predict uncertainty estimates to detect novel observations. The magnitude of those uncertainty estimates was shown to reveal novel obstacles in a variety of scenarios, indicating that the model knows what it does not know. The result is a collision avoidance policy that can measure the novelty of an observation (via model uncertainty) and cautiously avoids pedestrians that exhibit unseen behavior. Measures of model uncertainty can also be used to identify unseen data during training or testing. Policies during simulation demonstrated to be more robust to novel observations and take safer actions than an uncertainty-unaware baseline. This work also responds to the problem of safe reinforcement learning [36], whose goal is to ensure reasonable system performance and/or respect safety constraints also at the deployment phase.

Some work has also been made to explain multiagent RL. Wang et al. [107] developed an approach named Shapley Q-values Deep Deterministic Policy Gradient (SQDDPG) to solve global reward games in a multiagent context based on Shapley values and DDPG. The proposed approach relies on distributing the global reward more efficiently across all agents. They show that integrating Shapley values into DDPG enables to share the global reward between all agents according to their contributions: the more the agent contributes, the more reward it will get. This contrasts to the

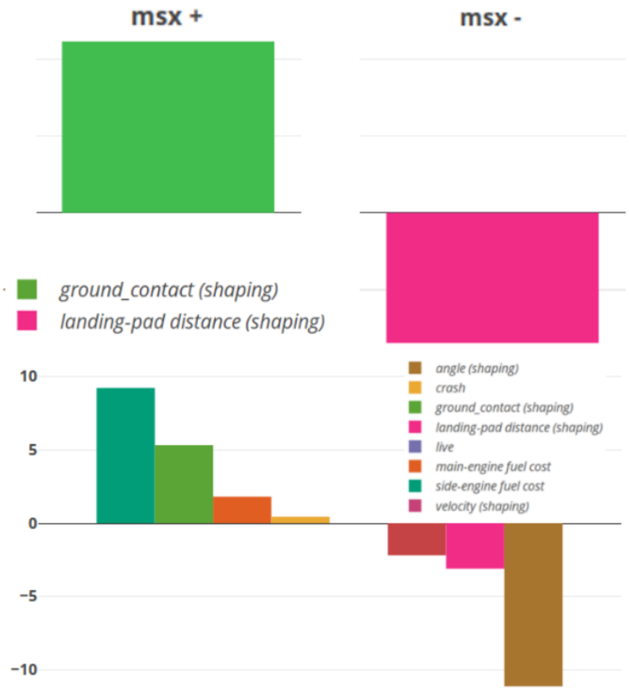


Figure 4: **Top** Minimal Sufficient Explanations (MSX) (fire down engine action vs. *do nothing* action) for decomposed reward DQN in *Lunar Lander* environment near landing site. The shaping rewards dominate decisions. **Bottom** RDX (noop vs. fire-main-engine) for HRA in Lunar Lander before a crash. The RDX shows that noop is preferred to avoid penalties such as fuel cost. Reproduced with permission of Zoe Juozapaitis [53].

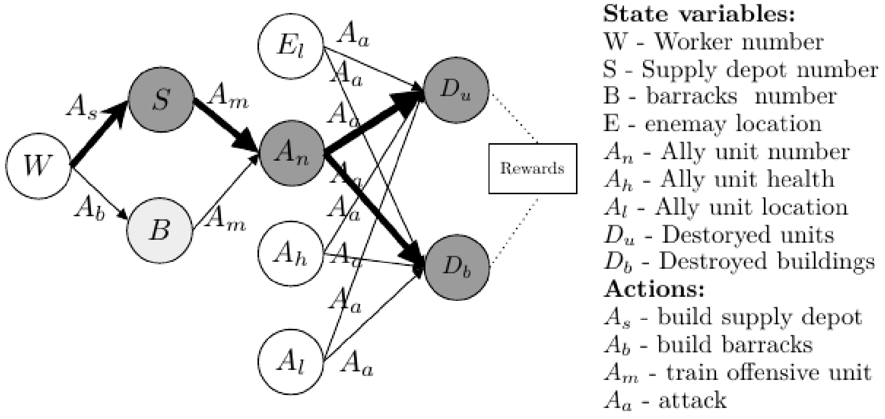
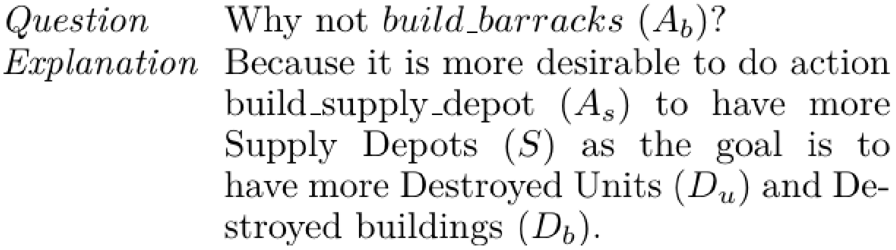


Figure 5: Action influence graph of a Starcraft II agent. The causal chain for action A_s is depicted in bold arrows and the extracted *explanan* (subset of causes given the explanation) is shown as darkened nodes. The counterfactual action (*why not* A_b) *explanan* is shown as grayed node (B). Here, A_s is the *explanandum*, the action for which the user needs explanation. Reproduced with permission of [71].



Question	Why not *build_barracks* (A_b)?
Explanation	Because it is more desirable to do action build_supply_depot (A_s) to have more Supply Depots (S) as the goal is to have more Destroyed Units (D_u) and Destroyed buildings (D_b).

Figure 6: Example of explanation when asking why a Starcraft II agent built supply depots rather than choosing to build barracks. Reproduced with permission of [71].

classical shared reward approach, which could cause inefficient learning by assigning rewards to an agent who contributed poorly. The experiments showed that SQDDPG presents faster convergence rate and fairer credit assignment in comparison with other algorithms (i.e. IA2C, IDDPG, COMA and MADDPG). This method allows to plot credit assignment to each agent, which can explain how the global reward is divided during training and what agent contributed the most to obtain the global reward.

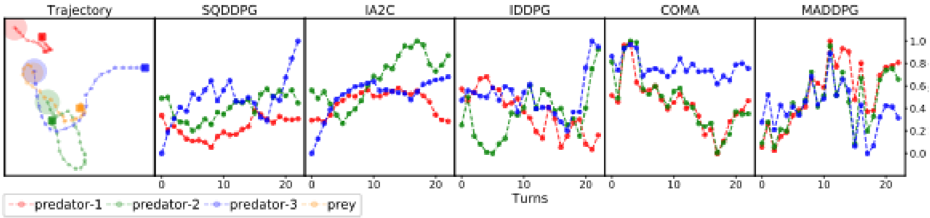


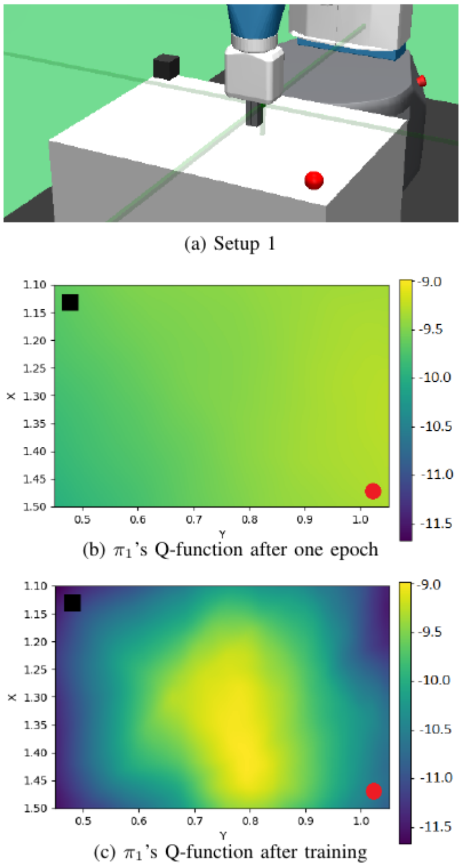
Figure 7: Credit assignment to each predator for a fixed trajectory in prey and predator task (Multiagent Particles environment [68]). **Left figure**: Trajectory sampled by an expert policy. The square represents the initial position whereas the circle indicates the final position of each agent. The dots on the trajectory indicate each agent's temporary positions. **Right figures**: normalized credit assignments generated by different multiagent RL algorithms according to this trajectory. SQDDPG presents fairer credit assignments in comparison with other methods. Reproduced with permission of Jianhong Wang [107].

2.1.3 Explanation through hierarchical goals

Methods based on Hierarchical RL [91] and sub-task decomposition [55] consist of a high level agent dividing the main goal into sub-goals for a low-level agent, which follows them one by one to perform the high-level task. By learning what sub-goals are optimal for the low-level agent, the high-level agent forms a representation of the environment that is interpretable by humans. Often, Hindsight Experience Replay (HER) [7] is used in order to ignore whether or not goals and sub-goals have been reached during an episode and to extract as much information as possible from past experience.

Beyret et al. [15] used this kind of methods along with HER for robotic manipulation (grasping and moving an item). The high level agent learns which sub-goals can make the low level agent reach the main goal while the low level agent learns to maximise the rewards for these sub-goals. The high-level agent provides a representation of the learned environment and the Q-values associated, which can be represented as heat maps as shown in Fig. 8.

Based on the same ideas, Cideron et al. [23] proposed Textual Hierarchical Experience Replay (THER) which extends the HER explanation to a natural language setting, allowing to learn from past experiences and to map goals to trajectories without the need of an external expert. The mapping function labels unsuccessful trajectories by automatically predicting a substitute goal. THER is composed of two models: the instruction generator which outputs a language encoding of the final state, and an agent model which picks an action given the last observations and the language-encoded goal.



(a) Setup 1

(b) π_1's Q-function after one epoch

(c) π_1's Q-function after training

Figure 8: Setup with initial state and goal diagonally opposed on the table. The heat maps show the value of the different areas (highest in yellow) for the high-level agent to predict a sub-goal. Black squares represent the position of the cube, the red circle is the end goal. Thus, the low-level agent will have a succession of sub-goals (e.g. multiple actions that the robotic arm must perform such as moving or opening its pinch) that will ultimately lead to the achievement of the high-level goal (i.e. grasping the red ball). Reproduced with permission of [15].

The model learns to encode goals and states via natural language, and thus can be interpreted by a human operator (Fig. 9).

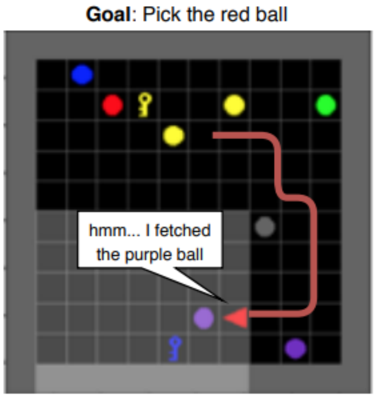


Figure 9: MiniGrid environment [21], where the agent is instructed through a textual string to pick up an object and place it next to another one. The model learns to represent the achieved goal (e.g. "Pick the purple ball") via language. As this achieved goal differs from the initial goal ("Pick the red ball"), the goal mapper relabels the episode, and both trajectories are appended to the replay buffer. Reproduced with permission of M. Seurin [23].

Another interesting work finds inspiration in human behaviour to improve generalization on a room navigation task, just like common sense and semantic understanding are used by humans to navigate unseen environments [110]. The entire model is composed of three parts: 1) a semantically grounded navigator used to predict the next action. 2) a common sense planning module, used for route planning. It predicts the next room, based on the observed scene, helps finding intermediate targets, and learns what rooms are near the current one. 3) the semantic grounding module used to recognize rooms; it allows the detection of the current room and incorporates semantic understanding by generating questions about what the agent saw (”Did you see a bathroom?”). Self-supervision is then used for fine tuning on unseen environment. The explainability can be brought from the outputs of all parts of the entire model. We can get information about what room is detected by the agent, what are the next rooms targeted (sub-goals), what are the rooms predicted around the current room and what are the rooms already seen by the agent.

An original idea proposed by Tasse et al. [103] consists of making an agent learn basic tasks and then allow it to perform new ones by composing the tasks previously learned in a boolean formula (i.e., with conjunctions, disjunctions and negations). The main strength of this method is that the agent is able to perform new tasks without the necessity of a learning phase. From an XRL point of view, the explainability comes from the fact that the agent is able to express its actions as boolean formulas, which are easily readable by humans.

2.2 Post-Hoc explainability

Post-Hoc explainability refers to explainability methods that rely on an analysis done after the RL algorithm finishes its training and execution. In other terms, it is a way of "enhancing" the considered RL algorithm from a black box to something that is somewhat explainable. Most Post-Hoc methods encountered were used in a perception context, i.e., when the data manipulated by the RL algorithm consisted of visual input such as images.

2.2.1 Explanation through saliency maps

When an RL algorithm is learning from images, it can be useful to know which elements of those images hold the most relevant information (i.e., the salient elements). These elements can be detected using saliency methods that produce saliency maps [92, 78]. In most cases, a saliency or heat map consists of a filter applied to an image that will highlight the areas salient for the agent.

A major advantage of saliency maps is that it can produce elements that are easily interpretable by humans, even non-experts. Of course, the interpreting difficulty of a saliency map greatly depends on the saliency method used to compute that map and other parameters such as the color scheme or the highlighting technique. A disadvantage is that they are very sensitive to different input variations, and schemes to debug such visual explanation may not be straightforward [51].

A very interesting example [41], introduces a new perturbation-based saliency computation method that produces crisp and easily interpretable saliency maps for RL agents playing OpenAI Gym environment *Atari 2600* games with Asynchronous Actor Critic [74]. The main idea is to apply a perturbation on the considered image that will remove information from a specific pixel without adding new information (by generating an interpolation from a Gaussian blur of the same image). Indeed, this perturbation can be interpreted as adding spatial uncertainty to the region around its point of application. This spatial uncertainty can help understand how removing information in a specific area of the input image affects the agent's policy, and is quantified with a saliency metric S. The saliency map is then produced by computing $S(i, j)$ for every pixel (i, j) of the input image, leading to images such as those in Fig. 10.

However, saliency methods are not a perfect solution in every situation, as pointed out in [58, 5]. They need to respect a certain number of rules, such as implementation invariance or input invariance in order to be reliable, especially when it comes to their relation with either the model or the input data.

2.2.2 Explanation through interaction data

In a more generic way, the behaviour of an agent can be explained by gathering data from its interaction with the environment while running, and analysing it in order to extract key information. For instance, Caselles-Dupré et al demonstrate that symmetry-based disentangled representation learning requires interaction and not only static perception [18].

This idea is exploited by Sequeira et al. [93] where interaction is the core basis upon which their Interestingness Framework is built. This framework relies on introspection, conducted by the autonomous RL agents: the agent extracts *interestingness elements* that denote meaningful interactions from their history of interaction with the environment. This is done using interaction data collected by the agent that is

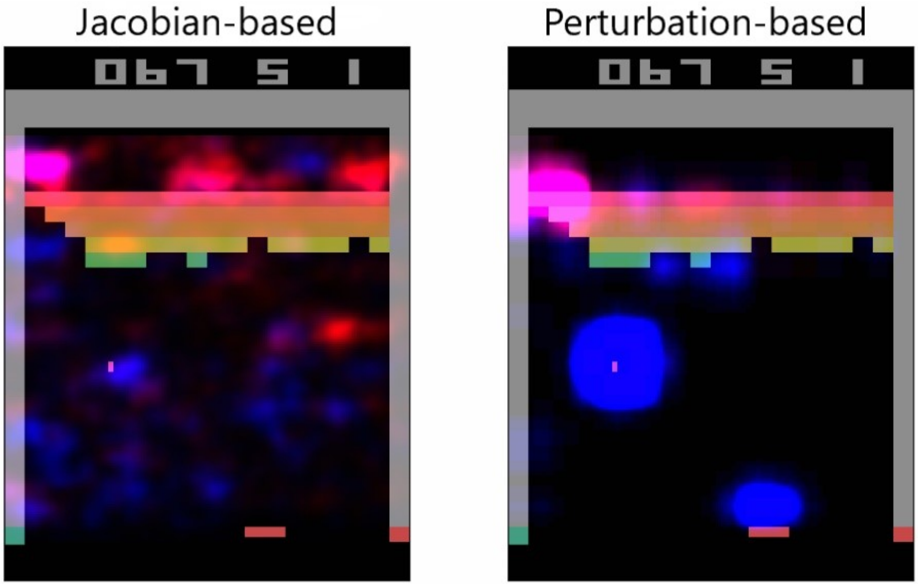


Figure 10: Comparison of Jacobian saliency (left) first introduced by Simonyan et al. [98] to the authors' perturbation-based approach (right) in an actor-critic model. Red indicates saliency for the critic; blue is saliency for the actor. Reproduced with permission of Sam Greydanus [41].

analysed using statistical methods organized in a three-level introspection analysis: level 0: Environment analysis, level 1: Interaction analysis; level 3: Meta-analysis. From these *interestingness elements*, it is then possible to generate visual explanations (in the form of videos compiling specific highlight situations of interest in the agent's behaviour), where the different introspection levels and their interconnections provide contextualized explanations.

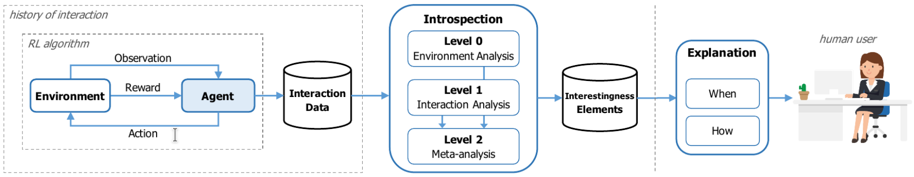


Figure 11: The interestingness framework. The introspection framework analyses interaction data collected by the agent and identifies interestingness elements of the interaction history. These elements are used by an explanation framework to expose the agent's behavior to a human user. Reproduced with permission from Pedro Sequeira [93].

The authors applied their framework to the game *Frogger* and used it to generate video highlights of agents that were included in a user study. The latter showed that no summarizing technique among those used to generate highlight videos is adapted to all types of agents and scenarios. A related result is that agents having a monotonous, predictable performance will lack the variety of interactions needed by the *interestingness* framework to generate pertinent explanations. Finally, counter-intuitively, highlighting all different aspects of an agent's interactions is not the best course of action ,as it may confuse users by consecutively showing the best and poorest performances of an agent.

2.3 Other concepts aiding XRL

Some studies encountered do not fit in the above categories for the main reason that they are not linked to RL or do not directly provide explanations but nonetheless, they are interesting concepts that could contribute to the creation of new XRL methods in the future.

2.3.1 Explainability of CNNs

Although deep neural networks have exhibited superior performance in various tasks, their interpretability is always their Achilles' heel. Since CNNs are still considered black boxes, many recent research papers focus on providing different levels and notions of explanations to make them more explainable.

As many RL models harness visual input DL models (for instance, when processing pixel observations), they could profit from better explainability of these algorithms. That way, the complete block of a CNN associated to learn a policy, would be explainable as whole. In addition, some techniques used in the visual domain, such as representation disentanglement could be relevant to apply in RL. Among the approaches detailed by Zhang et al. [115], one of the most promising aims at creating disentangled (interpretable) representations of the conv-layers of these networks [114, 113], as well as end-to-end learning of interpretable networks, working directly with comprehensible patterns, which are also a trending angle [108].

Explaining when, how, and under which conditions catastrophic forgetting [27] or memorizing of datasets occurs is another relevant aspect of life-long or continual learning [64] in DNNs yet not fully understood. An interesting method towards this vision is Learning Without Memorizing (LwM) [26], an extension of Learning Without Forgetting Multi-Class (LwF-MC) [65] applied to image classification. This model is able to incrementally learn new classes without forgetting classes previously learned and without storing data related them. The main idea is that at each step, a new model, the student, is trained to incrementally learn new classes, while the previous one, the teacher, only has knowledge of the base classes. By improving LwF-MC with the application of a new loss called Attention Distillation loss, LwM tries to preserve base classes knowledge across all models iterations. This new loss produces attention maps that can be studied by a human expert in order to interpret the model's logic by inspecting the areas that focus its attention.

Another approach for scene analysis aimed to build a graph where each node represents an object detected in the scene and is capable of building a context-aware representation of itself by sending messages to the other nodes [50]. This makes it possible for the network to support relational reasoning, allowing it to be effectively transparent. Thus, users are able to make textual inquiries about relationships between objects (e.g., "Is the plate next to a white bowl?").

2.3.2 Compositionality as a proxy tool to improve understandability

Compositionality is a universal concept stating that a complex (composed) problem can be decomposed into a set of simpler ones [11]. Thus, in the RL world, this idea can be translated into making an agent solve a complex task by hierarchically completing lesser ones (e.g. by first solving atomic ones as lesser tasks could also be complex) [81]. This provides reusability, enables quick initialization of policies and makes the learning process much faster by training an optimal policy for each reward and later combining them. Haarnoja et al. [43] showed that maximum entropy RL methods can produce much more composable policies. Empirical demonstrations were performed on a Sawyer robot trained to avoid a fix obstacle and to stack Lego blocks with both policies combined. They introduced the Soft Q-learning algorithm, based on maximum entropy RL [118] and energy-based models [44], as well as an extension of this algorithm that enables composition of learned skills. This kind of methods optimizing for compositionality does not provide a direct explanation tool; however compositionality can be qualitatively observed as self organized modules [46] and used to train multiple policies that benefit from being combined. Compositionality may also help better explain each policy along the training evolution in time, or each learned skill separately. However, it is also observed that compositionality may not emerge in the same manner as humans conceptually would understand it or expect it, e.g. based on symbolic abstract functionality modules. Some examples in language emergence in multi-agent RL settings show that generalization and acquisition speed [56] or language do not co-occur with compositionality, or that compositionality may not go hand in hand with language efficiency as in humans communication [19].

Distillation has also been used to learn task that are closely related and whose learning should improve speed up the learning of near tasks, in DisCoRL model [105], which helps transfer from simulation to real settings in navigation and goal based robotic tasks. We may then be able to further explain each policy along the training evolution timeline, or each learned skill separately.

2.3.3 Improving trust via imitation learning

Imitation learning is a way of enabling algorithms to learn from human demonstrations, such as teaching robots to learn assembly skills [109, 1]. While improving training time (compared to more traditional approaches [29]), this method also allows for better understanding of the agent's behaviour as it learns according to human expert actions [22]. It can also be a way to improve trust in the model, as it behaves seemingly as a human expert operator and can explain the basis of its decisions textually or verbally. Moreover, when encompassing human advice during training, it can be derived into advisable learning which further improves user trust as the model can understand human natural language and yields clear and precise explanations [57].

2.3.4 Transparency-oriented explanation building

Transparency has been given multiple meanings over time, especially in robotics and AI Ethics. Theodorou et al. [104] freshly define it as a mechanism to expose decision making that could allow AI models to be debugged like traditional programs, as they will communicate information about their operation in real time. However, the relevance of this information should adapt to the user's technological background, from simple progress bars to complex debug logs. An interesting concept is that an AI system could be created using a visual editor that can help communicate which decision will be taken in which situation (very much like decision trees). These concepts have already been successfully implemented in an RL setup using Temporal Difference (TD) error to create an emotional model of an agent [73].

3 Discussion

Despite explainable deep RL being still an emerging research field, we observed that numerous approaches were developed so far, as detailed in Section 2. However, there is no clear-cut method that serves all purposes. Most of the reviewed XRL methods are specifically designed to fit a particular task, often related to games or robotics and with no straight forward extension to other real-world RL applications. Furthermore, those methods cannot be generalized to other tasks or algorithms as they often make specific assumptions (e.g. on the MDP or environment properties). In fact in XRL there can be more than one model (as in Actor-Critic architectures) and different kinds of algorithms (DQN, DDPG, SARSA...) each with its own particularities. Moreover, there exists a wide variety of environments where each brings its own constraints. The necessity to adapt to the considered algorithm and environment means that it is hard to provide a holistic or *generic* explainability method. Thus, in our opinion, Shapley value-based methods [69, 107] can be considered as an interesting lead to contribute to this goal. Shapley values could be used to explain the roles taken by agents when learning a policy to achieve a collaborative task but also to detect defects in training agents or in the data fed to the network. In addition, as a post-hoc explainability method, it may be possible to generalize Shapley value computation to numerous RL environments and models in the same way it was done with SHAP [69] for other black boxes Deep Learning classifiers or regressors.

Meanwhile, the research community would benefit if more global-oriented approaches, which do not focus on a particular task or algorithm, were developed in the future, as it has already been done in general XAI, with for instance LIME [86] or SHAP [69].

Moreover, some promising approaches to bring explainability to RL include representation learning related concepts such as Hindsight Experience Replay, Hierarchical RL and self-attention. However, despite the ability of those concepts to improve performance and interpretability in a mathematical sense (in particular representation learning), they somehow lack concrete explanations targeted to end users, as they mostly target technical domain experts and researchers. This is a key element to further develop and allow the deployment of RL in the real world and to make algorithms more trustable and understandable by the general public.

The state of the art shows there is still room for progress to be made to better explain deep RL models in terms of different invariants preservation and other common assumptions of disentangled representation learning [67, 3].

4 Conclusion and Future Work

We reviewed and analyzed different state of the art approaches on RL and how XAI techniques can elucidate and inform their training, debugging and communication to different stakeholder audiences.

We focused on agent based RL in this work, however, explainability in RL involving humans (e.g. in collaborative problem solving [14]) should involve explainability methods to better assess when robots are able to perform the requested task, and when uncertainty is an indicator of better relying a task to a human. Equally important is to evaluate and explain other aspects in reinforcement learning, e.g. formally explaining the role of curriculum learning [82], quality diversity or other human-learning inspired aspects of open-ended learning [28, 77, 83]. Thus, more theoretic bases to serve explainable by design DRL are required. The future development of post-hoc XAI techniques should adapt to the requirements to build, train, and convey DRL models. Furthermore, it is worth noting that all presented methods decompose final prediction into additive components attributed to particular features [99], and thus interaction between features should be accounted for, and included in the explanation elaboration. Since most presented strategies to explain RL have mainly considered discrete model interpretations for explaining a model, as advocated in [100], continuous formulations of the proposed approaches (such as Integrated Gradients [101] based on the continuous extension of Shapley value, Aumann-Shapley value cost-sharing technique) should be devised in the future in RL contexts.

We believe the reviewed approaches and future extensions tackling the identified issues will likely be critical in the demanding future applications of RL. We advocate for the needs of targeting in the future more diverse audiences (developer, tester, end-user, general public) not yet approached in the development of XAI tools. Only this way we will produce actionable explanations and more comprehensive frameworks for explainable, trustable and responsible RL that can be deployed in practice.

5 Acknowledgements

We thank Sam Greydanus, Zoe Juozapaitis, Benjamin Beyret, Prashan Madumal, Pedro Sequiera, Jianhong Wang, Mathieu Seurin and Vinicius Zambaldi for allowing us to use their original images for illustration purposes. We also would like to thank Frédéric Herbreteau and Adrien Bennetot for their help and support.

References

[1] Pieter Abbeel and Andrew Ng. "Apprenticeship Learning via Inverse Reinforcement Learning". In: *Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004* (Sept. 2004). DOI: 10.1007/978-0-387-30164-8_417.

[2] Alessandro Achille, Tom Eccles, Loic Matthey, Chris Burgess, Nicholas Watters, Alexander Lerchner, and Irina Higgins. "Life-Long Disentangled Representation Learning with Cross-Domain Latent Homologies". In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Curran Associates, Inc., 2018, pp. 9873–9883. URL: http://papers.nips.cc/paper/8193-life-long-disentangled-representation-learning-with-cross-domain-latent-homologies.pdf.

[3] Alessandro Achille and Stefano Soatto. "A Separation Principle for Control in the Age of Deep Learning". In: *arXiv preprint arXiv:1711.03321* (2017).

[4] Alessandro Achille and Stefano Soatto. *Emergence of Invariance and Disentanglement in Deep Representations*. 2017. arXiv: 1706.01350 [cs.LG]. URL: https://arxiv.org/abs/1706.01350.

[5] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. "Sanity Checks for Saliency Maps". In: (2018). arXiv: 1810.03292 [cs.CV]. URL: https://arxiv.org/pdf/1810.03292.

[6] Samuel Alvernaz and Julian Togelius. *Autoencoder-augmented Neuroevolution for Visual Doom Playing*. 2017. arXiv: 1707.03902 [cs.AI]. URL: https://arxiv.org/pdf/1707.03902.

[7] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. "Hindsight Experience Replay". In: (2017). arXiv: 1707.01495 [cs.LG]. URL: https://arxiv.org/pdf/1707.01495.pdf.

[8] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Józefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, and et al. "Learning dexterous in-hand manipulation". In: *The International Journal of Robotics Research* (Nov. 2019), p. 027836491988744. ISSN: 1741-3176. DOI: 10.1177/0278364919887447. URL: http://dx.doi.org/10.1177/0278364919887447.

[9] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls. “Reinforcement learning-based multi-agent system for network traffic signal control”. In: *IET Intelligent Transport Systems* 4.2 (June 2010), pp. 128–135. ISSN: 1751-9578. DOI: 10.1049/iet-its.2009.0070. URL: http://web.eecs.utk.edu/~ielhanan/Papers/IET_ITS_2010.pdf.

[10] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. “Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI”. In: (2019). arXiv: 1910.10045 [cs.AI]. URL: https://arxiv.org/pdf/1910.10045.pdf.

[11] Marco Baroni. “Linguistic generalization and compositionality in modern artificial neural networks”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 375.1791 (Dec. 2019), p. 20190307. ISSN: 1471-2970. DOI: 10.1098/rstb.2019.0307. URL: https://arxiv.org/pdf/1904.00157.

[12] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. *Relational inductive biases, deep learning, and graph networks*. 2018. arXiv: 1806.01261 [cs.LG].

[13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. *Representation Learning: A Review and New Perspectives*. 2012. arXiv: 1206.5538 [cs.LG]. URL: https://arxiv.org/abs/1206.5538.

[14] Adrien Bennetot, Vicky Charisi, and Natalia Díaz-Rodríguez. “Should artificial agents ask for help in human-robot collaborative problem-solving?” In: *arXiv preprint arXiv:2006.00882* (2020). URL: https://arxiv.org/pdf/2006.00882.pdf.

[15] Benjamin Beyret, Ali Shafti, and Aldo Faisal. “Dot-to-Dot: Explainable Hierarchical Reinforcement Learning for Robotic Manipulation”. In: (Apr. 2019). URL: https://arxiv.org/pdf/1904.06703.pdf.

[16] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL].

[17] Hugo Caselles-Dupré, Michael Garcia Ortiz, and David Filliat. "Symmetry-Based Disentangled Representation Learning requires Interaction with Environments". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 4606–4615. URL: http://papers.nips.cc/paper/8709-symmetry-based-disentangled-representation-learning-requires-interaction-with-environments.pdf.

[18] Hugo Caselles-Dupré, Michael Garcia-Ortiz, and David Filliat. *Symmetry-Based Disentangled Representation Learning requires Interaction with Environments*. 2019. arXiv: 1904.00243 [cs.LG].

[19] Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. "Anti-efficient encoding in emergent communication". In: *Advances in Neural Information Processing Systems*. 2019, pp. 6290–6300. URL: https://papers.nips.cc/paper/8859-anti-efficient-encoding-in-emergent-communication.pdf.

[20] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets". In: *CoRR* abs/1606.03657 (2016). arXiv: 1606.03657. URL: http://arxiv.org/abs/1606.03657.

[21] Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. *Minimalistic Gridworld Environment for OpenAI Gym*. https://github.com/maximecb/gym-minigrid. 2018.

[22] Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. *Deep reinforcement learning from human preferences*. 2017. arXiv: 1706.03741 [stat.ML]. URL: https://arxiv.org/pdf/1706.03741.

[23] Geoffrey Cideron, Mathieu Seurin, Florian Strub, and Olivier Pietquin. "Self-Educated Language Agent With Hindsight Experience Replay For Instruction Following". In: (2019). arXiv: 1910.09451 [cs.LG]. URL: https://arxiv.org/pdf/1910.09451.pdf.

[24] Artur d'Avila Garcez, Aimore Resende Riquetti Dutra, and Eduardo Alonso. "Towards Symbolic Reinforcement Learning with Common Sense". In: *arXiv e-prints*, arXiv:1804.08597 (Apr. 2018), arXiv:1804.08597. arXiv: 1804.08597 [cs.LG]. URL: https://ui.adsabs.harvard.edu/abs/2018arXiv180408597D.

[25] Misha Denil, Sergio Gómez Colmenarejo, Serkan Cabi, David Saxton, and Nando de Freitas. *Programmable Agents*. 2017. arXiv: 1706.06383 [cs.AI].

[26] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. "Learning without Memorizing". In: (2018). arXiv: 1811.08051 [cs.CV]. URL: https://arxiv.org/pdf/1811.08051v2.pdf.

[27] Natalia Díaz-Rodríguez, Vincenzo Lomonaco, David Filliat, and Davide Maltoni. *Don't forget, there is more than forgetting: new metrics for Continual Learning*. 2018. arXiv: 1810.13166 [cs.AI]. URL: https://arxiv.org/pdf/1810.13166.

[28] Stephane Doncieux, Nicolas Bredeche, Léni Le Goff, Benoît Girard, Alexandre Coninx, Olivier Sigaud, Mehdi Khamassi, Natalia Díaz-Rodríguez, David Filliat, Timothy Hospedales, A. Eiben, and Richard Duro. "DREAM Architecture: a Developmental Approach to Open-Ended Learning in Robotics". In: (2020). arXiv: 2005.06223 [cs.AI]. URL: https://arxiv.org/abs/2005.06223.

[29] Stephane Doncieux, David Filliat, Natalia Díaz-Rodríguez, Timothy Hospedales, Richard Duro, Alexandre Coninx, Diederik M. Roijers, Benoît Girard, Nicolas Perrin, and Olivier Sigaud. "Open-Ended Learning: A Conceptual Framework Based on Representational Redescription". In: *Frontiers in Neurorobotics* 12 (2018), p. 59. ISSN: 1662-5218. DOI: 10.3389/fnbot.2018.00059. URL: https://www.frontiersin.org/article/10.3389/fnbot.2018.00059.

[30] Finale Doshi-Velez and Been Kim. "Towards A Rigorous Science of Interpretable Machine Learning". In: (2017). arXiv: 1702.08608 [stat.ML]. URL: https://arxiv.org/pdf/1702.08608.pdf.

[31] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. "Benchmarking Deep Reinforcement Learning for Continuous Control". In: (2016). arXiv: 1604.06778 [cs.LG]. URL: https://arxiv.org/pdf/1604.06778.pdf.

[32] Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. *Deep Spatial Autoencoders for Visuomotor Learning*. 2015. arXiv: 1509.06113 [cs.LG]. URL: https://arxiv.org/pdf/1509.06113.

[33] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. "Counterfactual Multi-Agent Policy Gradients". In: (2017). arXiv: 1705.08926 [cs.AI]. URL: https://arxiv.org/pdf/1705.08926.pdf.

[34] Yarin Gal and Zoubin Ghahramani. *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*. 2015. arXiv: 1506.02142 [stat.ML]. URL: https://arxiv.org/abs/1506.02142.

[35] Artur Garcez, Tarek Besold, Luc De Raedt, Peter Földiák, Pascal Hitzler, Thomas Icard, Kai-Uwe Kühnberger, Luís Lamb, Risto Miikkulainen, and Daniel Silver. "Neural-Symbolic Learning and Reasoning: Contributions and Challenges". In: Mar. 2015. DOI: 10.13140/2.1.1779.4243. URL: https://www.researchgate.net/publication/268034411_Neural-Symbolic_Learning_and_Reasoning_Contributions_and_Challenges.

[36] Javier García, Fern, and o Fernández. "A Comprehensive Survey on Safe Reinforcement Learning". In: *Journal of Machine Learning Research* 16.42 (2015), pp. 1437–1480. URL: http://jmlr.org/papers/v16/garcia15a.html.

[37] Marta Garnelo, Kai Arulkumaran, and Murray Shanahan. *Towards Deep Symbolic Reinforcement Learning*. 2016. arXiv: 1609.05518 [cs.AI]. URL: https://arxiv.org/pdf/1609.05518.

[38] Marta Garnelo and Murray Shanahan. "Reconciling deep learning with symbolic artificial intelligence: representing objects and relations". In: *Current Opinion in Behavioral Sciences* 29 (Oct. 2019), pp. 17–23. DOI: 10.1016/j.cobeha.2018.12.010. URL: https://www.researchgate.net/publication/336180670_Reconciling_deep_learning_with_symbolic_artificial_intelligence_representing_objects_and_relations.

[39] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2013. arXiv: 1311.2524 [cs.CV]. URL: https://arxiv.org/pdf/1311.2524.pdf.

[40] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Networks". In: (2014). arXiv: 1406.2661 [stat.ML]. URL: https://arxiv.org/pdf/1406.2661.pdf.

[41] Sam Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. "Visualizing and Understanding Atari Agents". In: (2017). arXiv: 1711.00138 [cs.AI]. URL: https://arxiv.org/pdf/1711.00138.

[42] David Gunning and David W Aha. "DARPA's Explainable Artificial Intelligence Program". In: *AI Magazine* 40.2 (2019), pp. 44–58. URL: https://search.proquest.com/openview/df03d4be3ad9847da3e414fb57bc1f10.

[43] Tuomas Haarnoja, Vitchyr Pong, Aurick Zhou, Murtaza Dalal, Pieter Abbeel, and Sergey Levine. "Composable Deep Reinforcement Learning for Robotic Manipulation". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)* (May 2018). DOI: 10.1109/icra.2018.8460756. URL: https://arxiv.org/pdf/1803.06773.pdf.

[44] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. *Reinforcement Learning with Deep Energy-Based Policies*. 2017. arXiv: 1702.08165 [cs.LG].

[45] Joseph Y. Halpern and Judea Pearl. "Causes and Explanations: A Structural-Model Approach. Part I: Causes". In: *The British Journal for the Philosophy of Science* 56.4 (Dec. 2005), pp. 843–887. ISSN: 0007-0882. DOI: 10.1093/bjps/axi147. eprint: https://academic.oup.com/bjps/article-pdf/56/4/843/4256158/axi147.pdf. URL: https://doi.org/10.1093/bjps/axi147.

[46] Dongqi Han, Kenji Doya, and Jun Tani. "Emergence of Hierarchy via Reinforcement Learning Using a Multiple Timescale Stochastic RNN". In: *CoRR* abs/1901.10113 (2019). arXiv: 1901.10113. URL: http://arxiv.org/abs/1901.10113.

[47] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. *Towards a Definition of Disentangled Representations*. 2018. arXiv: 1812.02230 [cs.LG]. URL: https://arxiv.org/abs/1812.02230.

[48] Robert R. Hoffman, Shane T. Mueller, Gary Klein, and Jordan Litman. *Metrics for Explainable AI: Challenges and Prospects*. 2018. arXiv: 1812.04608 [cs.AI]. URL: https://arxiv.org/pdf/1812.04608.

[49] Herke van Hoof, Nutan Chen, Maximilian Karl, Patrick van der Smagt, and Jan Peters. "Stable reinforcement learning with autoencoders for tactile and visual data". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2016), pp. 3928–3934. URL: https://www.ias.informatik.tu-darmstadt.de/uploads/Site/EditPublication/hoof2016IROS.pdf.

[50] Ronghang Hu, Anna Rohrbach, Trevor Darrell, and Kate Saenko. *Language-Conditioned Graph Networks for Relational Reasoning*. 2019. arXiv: 1905.04405 [cs.CV]. URL: https://arxiv.org/abs/1905.04405.

[51] Sarthak Jain and Byron C. Wallace. *Attention is not Explanation*. 2019. arXiv: 1902.10186 [cs.CL]. URL: https://arxiv.org/pdf/1902.10186.

[52] Rico Jonschkowski and Oliver Brock. "Learning state representations with robotic priors". In: *Autonomous Robots* (2015), pp. 407–428. URL: https://doi.org/10.1007/s10514-015-9459-7.

[53] Zoe Juozapaitis, Anurag Koul, Alan Fern, Martin Erwig, and Finale Doshi-Velez. "Explainable Reinforcement Learning via Reward Decomposition". In: (). URL: http://web.engr.oregonstate.edu/~afern/papers/reward_decomposition__workshop_final.pdf.

[54] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. "QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation". In: (2018). arXiv: 1806.10293 [cs.LG]. URL: https://arxiv.org/pdf/1806.10293.pdf.

[55] Hiroshi Kawano. "Hierarchical sub-task decomposition for reinforcement learning of multi-robot delivery mission". In: (2013), pp. 828–835. DOI: 10.1109/ICRA.2013.6630669. URL: https://doi.org/10.1109/ICRA.2013.6630669.

[56] Eugene Kharitonov and Marco Baroni. *Emergent Language Generalization and Acquisition Speed are not tied to Compositionality*. 2020. arXiv: 2004.03420 [cs.CL].

[57] Jinkyu Kim, Suhong Moon, Anna Rohrbach, Trevor Darrell, and John Canny. "Advisable Learning for Self-Driving Vehicles by Internalizing Observation-to-Action Rules". In: *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020. URL: https://openaccess.thecvf.com/content_CVPR_2020/html/Kim_Advisable_Learning_for_Self-Driving_Vehicles_by_Internalizing_Observation-to-Action_Rules_CVPR_2020_paper.html.

[58] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T. Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. "The (Un)reliability of saliency methods". In: (2017). arXiv: 1711.00867 [stat.ML]. URL: https://arxiv.org/pdf/1711.00867.

[59] Diederik P Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: (2013). arXiv: 1312.6114 [stat.ML]. URL: https://arxiv.org/pdf/1312.6114.pdf.

[60] Thomas N. Kipf and Max Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. 2016. arXiv: 1609.02907 [cs.LG].

[61] Yann LeCun, Yoshua Bengio, et al. "Convolutional networks for images, speech, and time series". In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995. URL: https://www.researchgate.net/publication/2453996_Convolutional_Networks_for_Images_Speech_and_Time-Series.

[62] T. Lesort, M. Seurin, X. Li, N. Díaz-Rodríguez, and D. Filliat. "Deep unsupervised state representation learning with robotic priors: a robustness analysis". In: *2019 International Joint Conference on Neural Networks (IJCNN)*. 2019, pp. 1–8. URL: https://hal.archives-ouvertes.fr/hal-02381375/document.

[63] Timothee Lesort, Natalia Díaz-Rodríguez, Jean-François Goudou, and David Filliat. "State representation learning for control: An overview". In: *Neural Networks* 108 (Dec. 2018), pp. 379–392. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2018.07.006. URL: https://arxiv.org/pdf/1802.04181.pdf.

[64] Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz Rodríguez. "Continual Learning for Robotics". In: *ArXiv* abs/1907.00182 (2019). URL: https://arxiv.org/pdf/1907.00182.pdf.

[65] Zhizhong Li and Derek Hoiem. "Learning without Forgetting". In: (2016). arXiv: 1606.09282 [cs.CV]. URL: https://arxiv.org/pdf/1606.09282.

[66] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. "Continuous control with deep reinforcement learning". In: (2015). arXiv: 1509.02971 [cs.LG]. URL: https://arxiv.org/abs/1509.02971.

[67] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. "Challenging common assumptions in the unsupervised learning of disentangled representations". In: *arXiv preprint arXiv:1811.12359* (2018). URL: https://arxiv.org/pdf/1811.12359.pdf.

[68] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments". In: *Neural Information Processing Systems (NIPS)* (2017). URL: https://arxiv.org/pdf/1706.02275.pdf.

[69] Scott Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: (2017). arXiv: 1705.07874 [cs.AI]. URL: https://arxiv.org/pdf/1705.07874.

[70] Björn Lütjens, Michael Everett, and Jonathan P. How. *Safe Reinforcement Learning with Model Uncertainty Estimates*. 2018. arXiv: 1810.08700 [cs.RO]. URL: https://arxiv.org/abs/1810.08700.

[71] Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. "Explainable Reinforcement Learning Through a Causal Lens". In: (2019). arXiv: 1905.10958 [cs.LG]. URL: https://arxiv.org/pdf/1905.10958.pdf.

[72] Hongzi Mao, Mohammad Alizadeh, Ishai Menache, and Srikanth Kandula. "Resource Management with Deep Reinforcement Learning". In: HotNets '16 (2016), pp. 50–56. DOI: 10.1145/3005745.3005750. URL: http://doi.acm.org/10.1145/3005745.3005750.

[73] Marco Matarese, Silvia Rossi, Alessandra Sciutti, and Francesco Rea. *Towards Transparency of TD-RL Robotic Systems with a Human Teacher*. 2020. arXiv: 2005.05926 [cs.RO].

[74] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. "Asynchronous Methods for Deep Reinforcement Learning". In: (2016). arXiv: 1602.01783 [cs.LG]. URL: https://arxiv.org/pdf/1602.01783.

[75] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. "Playing atari with deep reinforcement learning". In: *arXiv preprint arXiv:1312.5602* (2013). URL: https://arxiv.org/pdf/1312.5602.pdf.

[76] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (2015), p. 529. URL: https://storage.googleapis.com/deepmind-data/assets/papers/DeepMindNature14236Paper.pdf.

[77] Jean-Baptiste Mouret and Jeff Clune. "Illuminating search spaces by mapping elites". In: (2015). arXiv: 1504.04909 [cs.AI]. URL: https://arxiv.org/abs/1504.04909.

[78] T. Nathan Mundhenk, Barry Y. Chen, and Gerald Friedland. *Efficient Saliency Maps for Explainable AI*. 2019. arXiv: 1911.11293 [cs.CV]. URL: https://arxiv.org/pdf/1911.11293.

[79] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. *Deep Exploration via Bootstrapped DQN*. 2016. arXiv: 1602.04621 [cs.LG]. URL: https://arxiv.org/abs/1602.04621.

[80] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. *Curiosity-driven Exploration by Self-supervised Prediction*. 2017. arXiv: 1705.05363 [cs.LG]. URL: https://arxiv.org/abs/1705.05363.

[81] Thomas Pierrot, Guillaume Ligner, Scott Reed, Olivier Sigaud, Nicolas Perrin, Alexandre Laterre, David Kas, Karim Beguir, and Nando de Freitas. *Learning Compositional Neural Programs with Recursive Tree Search and Planning*. 2019. arXiv: 1905.12941 [cs.AI]. URL: https://arxiv.org/pdf/1905.12941.

[82] Rémy Portelas, Cédric Colas, Lilian Weng, Katja Hofmann, and Pierre-Yves Oudeyer. "Automatic Curriculum Learning For Deep RL: A Short Survey". In: *arXiv preprint arXiv:2003.04664* (2020). URL: https://arxiv.org/pdf/2003.04664.

[83] Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. "Quality diversity: A new frontier for evolutionary computation". In: *Frontiers in Robotics and AI* 3 (2016), p. 40. URL: https://www.frontiersin.org/articles/10.3389/frobt.2016.00040/full.

[84] Antonin Raffin, Ashley Hill, Kalifou René Traoré, Timothée Lesort, Natalia Díaz Rodríguez, and David Filliat. "Decoupling feature extraction from policy learning: assessing benefits of state representation learning in goal based robotics". In: *CoRR* abs/1901.08651 (2019). arXiv: 1901.08651. URL: http://arxiv.org/abs/1901.08651.

[85] Antonin Raffin, Ashley Hill, René Traoré, Timothée Lesort, Natalia Díaz-Rodríguez, and David Filliat. "S-RL Toolbox: Environments, Datasets and Evaluation Metrics for State Representation Learning". In: (2018). arXiv: 1809.09369 [cs.LG]. URL: https://arxiv.org/abs/1809.09369.

[86] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: (2016). arXiv: 1602.04938 [cs.LG].

[87] G. Rummery and Mahesan Niranjan. "On-Line Q-Learning Using Connectionist Systems". In: *Technical Report CUED/F-INFENG/TR 166* (Nov. 1994). URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.17.2539&rep=rep1&type=pdf.

[88] Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. *A simple neural network module for relational reasoning*. 2017. arXiv: 1706.01427 [cs.CL]. URL: https://arxiv.org/pdf/1706.01427.

[89] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. "The Graph Neural Network Model". In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 61–80. URL: https://persagen.com/files/misc/scarselli2009graph.pdf.

[90] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal Policy Optimization Algorithms". In: (2017). arXiv: 1707.06347 [cs.LG]. URL: https://arxiv.org/pdf/1707.06347.pdf.

[91] Harm van Seijen, Mehdi Fatemi, Joshua Romoff, Romain Laroche, Tavian Barnes, and Jeffrey Tsang. "Hybrid Reward Architecture for Reinforcement Learning". In: (2017). arXiv: 1706.04208 [cs.LG]. URL: https://arxiv.org/pdf/1706.04208.

[92] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *International Journal of Computer Vision* (Oct. 2019). ISSN: 1573-1405. DOI: 10.1007/s11263-019-01228-7. URL: http://dx.doi.org/10.1007/s11263-019-01228-7.

[93] Pedro Sequeira and Melinda Gervasio. "Interestingness Elements for Explainable Reinforcement Learning: Understanding Agents' Capabilities and Limitations". In: (2019). arXiv: 1912.09007 [cs.LG]. URL: https://arxiv.org/pdf/1912.09007.

[94] Pedro Sequeira, Eric Yeh, and Melinda T. Gervasio. "Interestingness Elements for Explainable Reinforcement Learning through Introspection". In: (2019). URL: https://explainablesystems.comp.nus.edu.sg/2019/wp-content/uploads/2019/02/IUI19WS-ExSS2019-1.pdf.

[95] Evan Shelhamer, Parsa Mahmoudieh, Max Argus, and Trevor Darrell. *Loss is its own Reward: Self-Supervision for Reinforcement Learning*. 2016. arXiv: 1612.07307 [cs.LG]. URL: https://arxiv.org/abs/1612.07307.

[96] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm". In: (2017). arXiv: 1712.01815 [cs.AI]. URL: https://arxiv.org/pdf/1712.01815.pdf.

[97] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, L Robert Baker, Matthew Lai, Adrian Bolton, Yutian chen, Timothy P. Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. "Mastering the game of Go without human knowledge". In: *Nature* 550 (2017), pp. 354–359. URL: https://www.nature.com/articles/nature24270.

[98] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps". In: (2013). arXiv: 1312.6034 [cs.CV]. URL: https://arxiv.org/pdf/1312.6034.pdf.

[99] Mateusz Staniak and Przemyslaw Biecek. "Explanations of model predictions with live and breakDown packages". In: *arXiv preprint:1804.01955* (2018). URL: https://arxiv.org/pdf/1804.01955.pdf.

[100] Mukund Sundararajan and Amir Najmi. "The many Shapley values for model explanation". In: (2019). arXiv: 1908.08474 [cs.AI]. URL: https://arxiv.org/abs/1908.08474.

[101] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic Attribution for Deep Networks". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. International Convention Centre, Sydney, Australia: PMLR, June 2017, pp. 3319–3328. URL: http://proceedings.mlr.press/v70/sundararajan17a.html.

[102] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Second. The MIT Press, 2018. URL: http://incompleteideas.net/book/the-book-2nd.html.

[103] Geraud Nangue Tasse, Steven James, and Benjamin Rosman. "A Boolean Task Algebra for Reinforcement Learning". In: (2020). arXiv: 2001.01394 [cs.LG]. URL: https://arxiv.org/pdf/2001.01394.pdf.

[104] Andreas Theodorou, Robert H. Wortham, and Joanna J. Bryson. "Designing and implementing transparency for real time inspection of autonomous robots". In: *Connection Science* 29.3 (2017), pp. 230–241. DOI: 10.1080/09540091.2017.1310182. eprint: https://doi.org/10.1080/09540091.2017.1310182. URL: https://doi.org/10.1080/09540091.2017.1310182.

[105] René Traoré, Hugo Caselles-Dupré, Timothée Lesort, Te Sun, Guanghang Cai, Natalia Díaz Rodríguez, and David Filliat. "DISCORL: Continual reinforcement learning via policy distillation". In: *arXiv preprint arXiv:1907.05855* (2019). URL: https://arxiv.org/pdf/1907.05855.

[106] René Traoré, Hugo Caselles-Dupré, Timothée Lesort, Te Sun, Guanghang Cai, Natalia Díaz-Rodríguez, and David Filliat. "DisCoRL: Continual Reinforcement Learning via Policy Distillation". In: (2019). arXiv: 1907.05855 [cs.LG]. URL: https://arxiv.org/abs/1907.05855.

[107] Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. "Shapley Q-value: A Local Reward Approach to Solve Global Reward Games". In: (2019). arXiv: 1907.05707 [cs.LG]. URL: https://arxiv.org/pdf/1907.05707.pdf.

[108] Tianfu Wu, Wei Sun, Xilai Li, Xi Song, and Bo Li. *Towards Interpretable R-CNN by Unfolding Latent Structures*. 2017. arXiv: 1711.05226 [cs.CV]. URL: https://arxiv.org/pdf/1711.05226v2.

[109] Ali Al-Yacoub, Yuchen Zhao, Niels Lohse, Mey Goh, Peter Kinnell, Pedro Ferreira, and Ella-Mae Hubbard. "Symbolic-Based Recognition of Contact States for Learning Assembly Skills". In: *Frontiers in Robotics and AI* 6 (2019), p. 99. ISSN: 2296-9144. DOI: 10.3389/frobt.2019.00099. URL: https://www.frontiersin.org/article/10.3389/frobt.2019.00099.

[110] Dian Yu, Chandra Khatri, Alexandros Papangelis, Andrea Madotto, Mahdi Namazifar, Joost Huizinga, Adrien Ecoffet, Huaixiu Zheng, Piero Molino, Jeff Clune, et al. "Commonsense and Semantic-Guided Navigation through Language in Embodied Environment". In: (). URL: https://vigilworkshop.github.io/static/papers/49.pdf.

[111] Vinicius Zambaldi, David Raposo, Adam Santoro, Victor Bapst, Yujia Li, Igor Babuschkin, Karl Tuyls, David Reichert, Timothy Lillicrap, Edward Lockhart, Murray Shanahan, Victoria Langston, Razvan Pascanu, Matthew Botvinick, Oriol Vinyals, and Peter Battaglia. "Relational Deep Reinforcement Learning". In: (2018). arXiv: 1806.01830 [cs.LG]. URL: https://arxiv.org/pdf/1806.01830.

[112] Amy Zhang, Harsh Satija, and Joelle Pineau. *Decoupling Dynamics and Reward for Transfer Learning*. 2018. arXiv: 1804.10689 [cs.LG]. URL: https://arxiv.org/abs/1804.10689.

[113] Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu, and Song-Chun Zhu. *Interpreting CNN Knowledge via an Explanatory Graph*. 2017. arXiv: 1708.01785 [cs.CV]. URL: https://arxiv.org/pdf/1708.01785.

[114] Quanshi Zhang, Ruiming Cao, Ying Nian Wu, and Song-Chun Zhu. *Growing Interpretable Part Graphs on ConvNets via Multi-Shot Learning*. 2016. arXiv: 1611.04246 [cs.CV]. URL: https://arxiv.org/pdf/1611.04246.

[115] Quanshi Zhang and Song-Chun Zhu. "Visual Interpretability for Deep Learning: a Survey". In: (2018). arXiv: 1802.00614 [cs.CV]. URL: https://arxiv.org/pdf/1802.00614.pdf.

[116] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Yuan, Xing Xie, and Zhenhui Li. "DRN: A Deep Reinforcement Learning Framework for News Recommendation". In: (Apr. 2018), pp. 167–176. DOI: 10.1145/3178876.3185994. URL: http://www.personal.psu.edu/~gjz5038/paper/www2018_reinforceRec/www2018_reinforceRec.pdf.

[117] Zhenpeng Zhou, Xiaocheng Li, and Richard N. Zare. "Optimizing Chemical Reactions with Deep Reinforcement Learning". In: *ACS Central Science* 3.12 (2017). PMID: 29296675, pp. 1337–1344. DOI: 10.1021/acscentsci.7b00492. eprint: https://doi.org/10.1021/acscentsci.7b00492. URL: https://doi.org/10.1021/acscentsci.7b00492.

[118] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. “Maximum entropy inverse reinforcement learning.” In: 2008. URL: https://arxiv.org/pdf/1507.04888.pdf.

6 Appendix

6.1 Glossary

- **A2C**: Asynchronous Actor Critic [74]
- **AI**: Artificial Intelligence
- **COMA**: Counterfactual multi-agent [33]
- **CNN**: Convolutional Neural Network [61]
- **DDPG**: Deep Deterministic Policy Gradient [66]
- **DL**: Deep Learning
- **DRL**: Deep Reinforcement Learning
- **DQN**: Deep Q Network [75]
- **GAN**: Generative Adversarial Network [40]
- **HER**: Hindsight Experience Replay [7]
- **HMM**: Hidden Markov Model
- **HRA**: Hybrid Reward Architecture [91]
- **HRL**: Hierarchical Reinforcement Learning [55]
- **IDDPG**: Independent DDPG [66]
- **MADDPG**: Multiagent DDPG [68]
- **MDP**: Markov Decision Process
- **Machine Learning**: Machine Learning
- **POMDP**: Partialy Observable Markov Decision Process
- **PPO**: Proximal Policy Optimization [90]
- **R-CNN**: Region Convolutionnal Neural Network [39]
- **RL**: Reinforcement Learning
- **SARSA**: State Action Reward State Action [87]
- **SRL**: State Representation Learning [63]
- **VAE**: Variational Auto-Encoder [59]
- **XAI**: Explainable Artificial Intelligence
- **XRL**: Explainable Reinforcement Learning