Explainable reinforcement learning

Blaž Dobravec[1][0000−0003−2567−8443] Jure Žabkar[2][0000−0002−9459−8562]

[1] Faculty of Computer and Information Science, University of Ljubljana, Slovenia
bd39148@student.uni-lj.si
[2] Faculty of Computer and Information Science, University of Ljubljana, Slovenia
jure.zabkar@fri.uni-lj.si

Abstract. Explainable reinforcement learning is an important sub-field of explainable artificial intelligence that aims to develop algorithms that can provide interpretable explanations for the actions and policies of agents in an environment. While explainability in artificial intelligence has seen rapid growth in recent years, there has on one hand lack of explanations in reinforcement learning and on the other hand a lack of quantifiable evaluations of explainability or interpretability. In this study, we introduce two methods for generating explanations in the context of reinforcement learning: one using an *alteration of Layer-wise Relevance Propagation* method that we call Smoothed-LRP and the other using hand-made prototypes of trajectories. We demonstrate the implemented methods in our own autonomous driving car environment. Our results show that these explanations can provide valuable insights into the decision-making processes of reinforcement learning agents and improve the interpretability of their actions.

Keywords: Explainable Reinforcement learning · Evaluation of explanations.

1 Introduction

Artificial intelligence (AI) has in recent years achieved notable advances in multiple research areas. Not only that, AI has integrated into many aspects of our everyday lives as well. For example, the use of AI in the financial domain for fraud detection and optimal trading is becoming a norm [5]. Advances in speech and text-domain is transforming the way we communicate with machines [24]. Research in the fields of image, video and 3D modelling recognition and generation has sprung new innovative tools for creatives and storytellers [15]. But in order to safely expand the use of AI to sometimes critical aspects of human life high degree of transparency is necessary. For that reason, a relatively young but widely acknowledged as crucial field eXplainable Artificial intelligence (XAI) has emerged [5]. A. Adadi and M. Berrada [2] indicate that the scope of XAI includes understanding, interpreting and explaining complex models used in Transportation, Healthcare, Legal, Military and Finance. Transparent by-design algorithms exist (e.g. decision trees, linear regression, rule-based models) but most of the current state-of-the-art algorithms and research, achieving the

best performances in many previously mentioned domains, are primarily using complex data-driven architectures (SVM, Tree Ensembles, Convolutional Neural Network, Graph Neural Network, Transformers, etc.). Many methods for explaining the models have been proposed in order to "uncover the black-box" models [22] or visualise their reasoning for the classification through techniques such as feature importance (e.g. Layer-wise Relevance Propagation [6]).

In this work, we propose 2 examples of explanations for reinforcement learning in the same environment. The first one is based on **feature attribution** and the second one is based on **prototypes**. We propose a new type of prototype, that we call **trajectory prototype**, which represents a typical trajectory (achieved by a series of actions and defined as a series of consecutive states) that an agent would have to to in our environment. We also introduce trajectory prototypes into the alteration of the existing learned policy by a classic reinforcement learning model.

Reinforcement learning (RL) algorithms have in the last years achieved human-like performance on multiple domains. A massive spike in interest in the field of reinforcement learning began when *DeepMind* released an extension of classic *Q*-learning called *Deep Q-learning* in 2014. A variation of the deep reinforcement learning (DRL) architecture was used in a famous *AlphaGo* [33] that beat a grand master at a game of chess. Reinforcement learning was and still is used to learn to play various computer games (e.g. Atari 2600 games), but has since then evolved to tackle real-life use cases as well. Researchers at Tesla are tirelessly working on car driving automation using DRL [35]. Similarly, has reinforcement learning been tested in complex real-life environments and systems (e.g. cooling data-server facilities [27]) as well as automation in robotics [30].

Despite many advancements and new developments in reinforcement learning, the field of machine learning may just be one of the least represented in real-world applications, especially in critical domains [10]. The inability to explain and justify agents' actions and decisions, makes us question the validity of the models and restricts wide-scale deployment of RL systems in some critical fields such as healthcare or transportation [10]. Those domains remain strict to the adoption of new technologies and methods, especially if the inner working of the technology is not transparent. Many times it is hard to construct interpretable models with the same accuracy as data-driven models, doe this has been counter-argued by C. Rudin [28]. The author has also collaborated on the construction of a prototype-based interpretable neural network architecture called *ProtoPNet* [9, 20]. A prototype is an instance (e.g. an image) from data that is representative of a prominent feature for a certain class. This research based on prototypes has recently ignited research in the use of prototypes in deep reinforcement learning [4] and Ragodos et al. [26].

explainability in RL

Interpretability vs explainability: Distinguishing between interpretability and explainability is an important topic to discuss. It is difficult to (mathematically) define either of them. A (non-mathematical) definition of interpretability presented by T. Miller [21] is that interpretability is the degree to which a human

can understand the cause of a decision. Kim et al. [19] defined interpretability as the degree to which a human can consistently predict the model's result. Similarly, Gilpin et al. [14] distinguished between interpretability and explainability, and proposed principles to evaluate ML interpretation methods. They argue that the main purpose of an explanation is to answer a *why* question and that the interpretability of the system is its ability to provide understandable explanations (answers to *how* questions) to the users.

Both interpretability and explainability are challenging to evaluate. Authors of the articles with explainability techniques focus on defining specific metrics and conducting specifically organised questionnaires based on which statistical tests are conducted to evaluate the explainability or interpretability (e.g. ANOVA test, Likert scale data of explanation metrics). Attempts to define quantifiable metrics of explainability have already been made, but many of them do not provide any examples of the evaluations.

2 Explainability in Reinforcement Learning

Explainability in reinforcement learning is in most survey papers [3, 16] divided into two main categories based on the type of explanation method used (depicted in Fig. 1):

- Transparent/Intrinsic based methods,
- Post-hoc explanation-based methods,

2.1 Transparent Explanations

Transparent algorithms are well known and used in Machine Learning (ML) (e.g., linear regression, decision trees or rule-based systems). Their strength lies in the fact that they are designed to have a transparent architecture that makes them directly explainable, without any additional processing. However, it is quite different for deep reinforcement learning, as standard DRL algorithms (e.g. DQN, PPO, DDPG, A2C...) are not transparent by nature [16]. Another approach is to transform the state space and/or actions to be more interpretable [13].

Explanation through representation learning focuses on learning abstract features that characterize data, in order to make it easier to extract useful information when building predictors [7, 12]. These learned features have the advantage of having low dimensionality, which generally improves training speed and generalization of Deep Learning models [25]. *Simultaneous learning of the explanation and the policy* is based on altering the existing architecture to incorporate explanations while training the agent. Juozapaitis et al. [17] introduced such a concept through *reward decomposition*, whose main principle is to decompose the reward function into a sum of meaningful reward types. An additional idea in reinforcement learning is to create a *sub-task decomposition* (also called Hierarchical RL) [8, 18, 31] that consist of dividing the bigger complex concept into smaller more manageable and understandable tasks. By learning what sub-goals

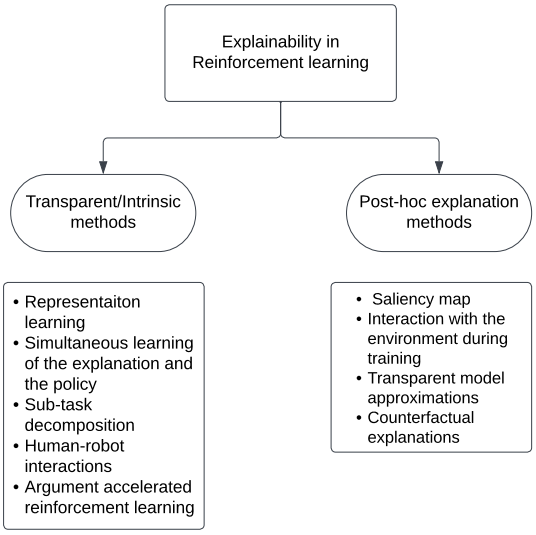

Fig. 1. Division of the field of explainability in reinforcement learning.

are optimal, the high-level agent is viewed as a representation of multiple puzzles that form an environment that is interpretable by humans.

Using *human-robot interactions* in the reinforcement learning domain has also been explored and exploited to provide more understandable and explainable policies. An *argument accelerated reinforcement learning* has been proposed by G. Yung and F. Toni [13] which uses human-defined argumentation to coordinate multi-agents and their actions through reward shaping which makes the coordination between agents interpretable. Another type of human-robot interaction includes human-guided reinforcement learning through text or voice. This technique was explored by researchers at DeepMind [1]. Authors first construct a database of robot interactions with the environment, they then annotate the time spans where agents are progressing or are regressing from the instructed goals. Based on this database of instructions and image trajectories authors propose a novel method that they call *Inter-temporal Bradley-Terry modelling* (IBT) to build a reward model that captures human judgments.

2.2 Post-hoc Explanations

Post-hoc explanations are a type of explanation that uses post-training interpretation approaches to provide explanations for complex data-driven models. Usually, this approach does not alter the model itself. A very commonly used explainability technique in image classifications is a *saliency map* [34] which is based on pixel-wise feature attribution. Another approach to explain the model is by gathering the information of the agent's *interactions with the environment*

during training and afterwards trying to explain actions using those interactions. This idea is exploited by Sequeira et al. [32] where interaction with the environment is the core of their *Interestingness Framework*. Authors use a Q-learning algorithm with a finite-state game called Frogger . *Transparent model approximations* [11] are also used in order to mimic the target agent's policy data-driven models and then derive explanations from the transparent model. Similarly, approximation techniques may be used to create an approximate model that replaces the original policy network. In order to get better insight into the reasons for actions in the domain translation to a knowledge graph was proposed by Zhao et al. [37]. Xian et al. [36] extend a similar architecture to include a policy-guided path reasoning. The latter approach using knowledge graphs is relatively restricted due to the need to construct such a knowledge graph, which is in some domains not trivial.

2.3 Evaluating Explanations

The general term 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 [16]. Evaluating explanations in XAI is quite a challenging task. There are two main reasons:

1. concept of explainability in RL is not well accepted and there is still al lack of formal (mathematical) definition,
2. quality of an explanation is many times qualitative and subjective where the same explanation is different in "strength" from person to person.

Due to the subjective nature of the explanations, user studies are a popular approach to evaluate explanations (e.g. asking questions like: "How good is the explanation that the agent has made a turn to the left" and giving quantitative answers). User explainability studies have in the past also been criticised as the participants' prior knowledge may play a crucial role. Generally in XAI, there is only a single instance 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?" [16]. Multiple attempts to define quantitative metrics of the explainability evaluations have been made. Santhanam et al. [29] propose a three metrics:

1. **Correctness** (sensitivity or fidelity in literature) refers to the ability of an explainer to correctly identify components of the input that contribute most to the prediction of the classifier.
2. **Consistency** is defined as the ability of the explainer to capture the same relevant components under various transformations to the input.
3. **Confidence** is concerned with whether the generated explanation and the masked input result in high confidence predictions.

[2] Available at https://github.com/pedrodbs/frogger.

Zhou et al. [38] divides the explainability methods into three categories where different metrics are defined for each category:

- Model-Based Explanations
- Attribution-Based Explanations
- Example-Based Explanations

Nguyen and Martinez [23] evaluate explanations using feature attribution define terms *monotonicity*, *non-sensitivity*, and *effective complexity*. To validate the proposed metrics authors test them on two tasks (transparent task and complex task). The authors also present metrics for the evaluation of example-based explainability. Experimental results are presented on a rotated MNIST dataset using CNN and three prototype explanation methods.

3 Experiments

Our two proposed types of explanations are implemented in our previously designed simple autonomous driving environment. First, we implemented a version of layer-wise relevance propagation algorithm [6] well known in classic XAI, that we adapted to work in the DRL setting and our environment. Secondly, we implemented a human-defined prototype explanation of our network which is inspired by previous work on prototypes in classification [20] and reinforcement learning setting [4]. Our agent is based on a Deep Q-network algorithm using experience replay with a simple three-layer fully connected neural network.

3.1 Environment

We created a simple autonomous driving environment where an agent is able to perform three simple actions:

- turning to the left for a fixed 3.5°,
- turning to the right for a fixed 3.5°,
- driving forward.

The speed of the car in our environment is constant. An agent controls the movement of the car and is able to view the environment using 5 sensors (lasers) arranged evenly 30° apart from each other as is depicted on Fig. 2. Additionally, we constructed the environment so that we are able to quickly construct different race tracks.

We defined the state in our environment defined as a vector $s = [s_1, s_2, s_3, s_4, s_5]$ where each s_i represents the length of the sensor (the length of the sensor is the length from the car to the first obstacle in the direction of the sensor as can be seen on Fig. 2).

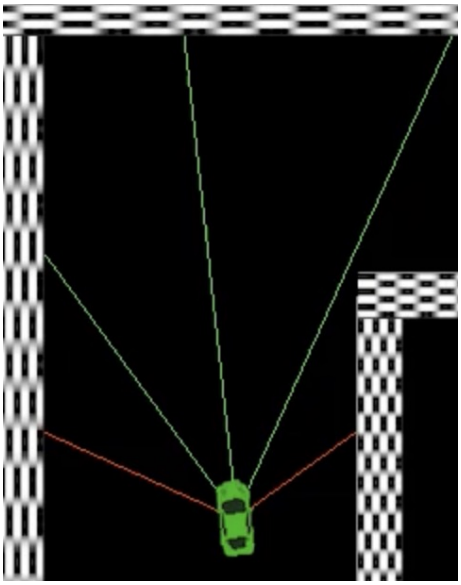

Fig. 2. Car and the five sensors in our environment.

3.2 Smoothed Layer-wise Relevance Propagation

We base our first explainability approach on the feature attribution method, more specifically LRP: A concept for generating saliency maps, which visualize how much each feature (pixel) of the input data (image) contributes to the output (classification). In contrast to most other approaches, LRP offers the benefit of conserving the certainty of the prediction at individual layers throughout the backward process. Additionally, since our approach is not vision-based we altered the LRP method to work on a general vector. LRP task uses an alternative forward and backward passing over the network in order to calculate individual feature (sensor values) importance attributions. Our method can be thought of as a wrapper around a classic deep reinforcement learning algorithm, where at each iteration of the testing process we calculate the attributions using backward passes over the network.

In our autonomous driving environment, we extract the attributions of each sensor for the selected action (turning left, turning right or driving forward) and present the attributions as vectors in the directions of the corresponding sensors. We are then able to, based on this approach and the chosen domain, sum up the directed vectors (using the sensor directions) and visualise the direction that the car is headed (heading vector).

Using LRP we can use different rules for attributions calculations also presented more in-depth in the works by Bach et al. [6]. We implemented multiple rules and visually analysed the results. The basic propagation rule is depicted in equation 1, where R_j and R_k are attributions of neurons j and k and z_{jk} is

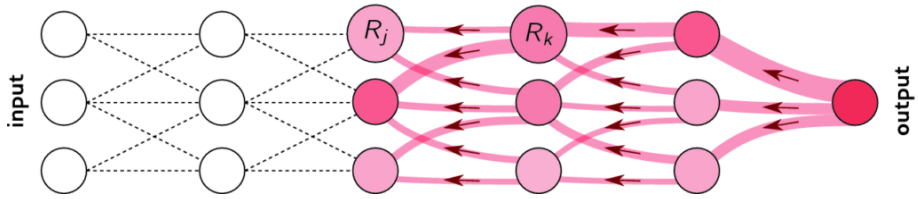

Fig. 3. Image representing LRP backpropagation of relevance values of each neuron (credited to https://neurosys.com/blog/visualizing-neural-networks-decision-making-lrp).

the activation of the neuron j multiplied by the weight between neuron j and neuron k. The visual representation of the process is displayed in Fig. 3.

$$R_j = \sum_k \frac{z_{jk}}{\sum_j z_{jk}} R_k \tag{1}$$

The rule that we found to work the best is an *alpha-beta* rule depicted in equation 2. It is an extension of an original rule 1 where negative and positive effects are divided. Parameters $\alpha = 1$ and $\beta = 0$ showed the most promising results, which means that we only take into account the positive weights in the network.

$$R_j = \sum_k \left(\alpha \cdot \frac{(x_j w_{jk})^+}{\sum_j (x_j w_{jk})^+} - \beta \cdot \frac{(x_j w_{jk})^-}{\sum_j (x_j w_{jk})^-} \right) R_k \tag{2}$$

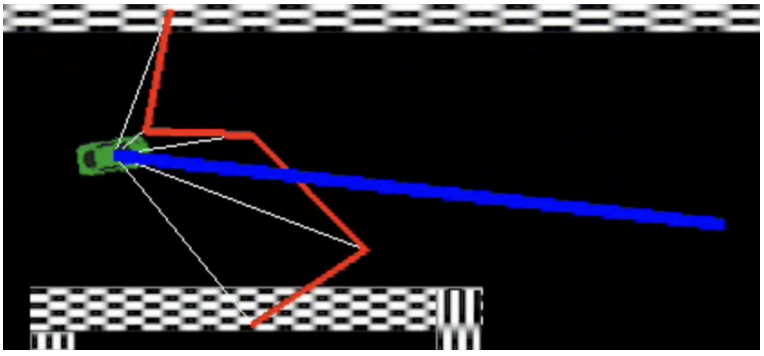

Fig. 4. A frame of a visualization of an explanation using modified LRP rules in an autonomous learning setting. White lines represent the feature importance attributes of individual sensors after the LRP procedure. The blue line represents the *header vector* (sum of the feature importance vectors) The connecting red lines are there only to better visualize the importance of the triangles between each sensor attribute value.

In order to better visualize the results we also smoothed the LRP (deriving from the naming Smoothed-LRP) individual attributions and the heading-vector (as depicted with blue colour on Fig. 4 and calculated as a sum of all attribution vectors). We used the smoothing effect as an average over a period of 10 steps in the environment, this produced the most prominent explanations. Making the smoothing bigger would result in the lag of the attributions (and consequently the *heading vector*).

3.3 Human-friendly trajectory prototypes

The original extensive work and the idea for the use of prototypes in explanations of complex models became possible when Chen et al. [9] introduced an alteration of the original neural network which integrated prototype learning in the learning process. The work has been recently extended for use in reinforcement learning [4, 26]. We used the idea of using human-friendly prototypes [4], but extended it to not only use a single state but rather use a trajectory (a series of states) as a representation of a prototype. Our idea was that while a single action in reinforcement learning might sometimes not be easily understandable, the trajectories are usually easier to understand. A similar idea has been proposed by Ragodos et al. [26], where prototypes are directly learned during training. Our proposed neural network architecture is depicted in Fig. 5.

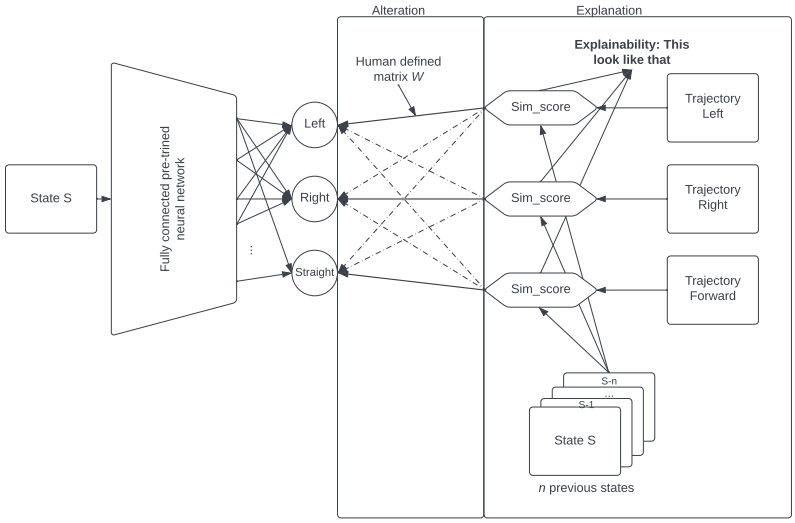

Fig. 5. A diagram of the explainability network after the learning stage using our proposed trajectory prototypes.

We train the normal fully connected neural network using the classic reinforcement learning technique using DQN algorithm and then in the explanation stage add an explainability phase at the end of the last fully connected layer. Our idea is based on the fact that previous n states to some degree influence the following states. This fact may not hold for all reinforcement tasks but does hold for many nature-inspired tasks (e.g. learning to walk).

The *explainability phase* is based on human-defined trajectory prototypes. Our trajectory prototypes each consist of 25 consecutive states representing the three main concepts (turning left, right and driving forward) in our environment. The similarity scores are calculated based on the corresponding first n states in the (normalized) trajectory prototypes and the n previously visited (normalised) states (n is referred as window). The similarity score was calculated based on the mean absolute error (MAE) metric for each tuple (first n steps in prototype, previous n states) and then normalised among all trajectory prototypes in order to gain the percentages of correlations of each prototype. We experimentally fixed the window at $n = 10$. The explanation for action is now that what we have in the last n states done looks like one of the trajectory prototypes, which indicated that the following steps may also be in the direction of the same prototype. The idea behind taking only the window (first n states) of the prototype is that there is a trade-off between the quickness of detection of a prototype and the agents' response time (window where possible alterations can be made), therefore the window n plays a crucial role. Changing the window n through time may also be beneficial.

The *alteration phase* is based on human-defined weight matrix W and similarity scores (which is in our example an identity matrix as we have the corresponding trajectory prototypes), and it alters the existing classifications of the learned neural network in order to guide the agent along the trajectory. The definition of this matrix is **not trivial in many domains** and depends on the choice of the prototypes. The values of a window n in each prototype now plays a crucial role. If window n is too big then an agent will be guided to follow the trajectory for too long, but if window n is too small then trajectory prototypes might be changing too frequently which may guide the agent to be less successful.

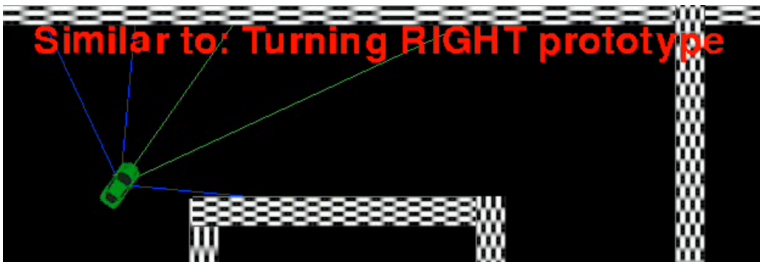

Fig. 6. The image represents a snapshot of the right turn trajectory prototype.

It is important to note that the trajectory prototypes may not be the same as the output actions (as they are in our example). If the number of prototypes is different and they do not correlate to the actions then the matrix W is harder to set.

We visualize our agent and explanation in Fig. 6 using a prompt which provides an explanation for the given state. We also provide supplementary material in a form of a video [3].

4 Discussion and Conclusion

In this report, we first present the field of explainable reinforcement learning, then we present our work on two explainability techniques: an RL variant of layer-wise relevance propagation and a prototype similarity algorithm. Through the autonomous driving task, we demonstrate the explanations of the agents' decision-making process. We provide explanations for the actions and multi-actions made by the agent however the evaluation of the explanations is still needed and is intended for future work. Another possible direction is the use of the LRP process or trajectory prototypes as arguments for or against actions made by the agent in the environment and therefore improve the learned policy this is already hinted at in the alteration phase of the trajectory prototype example. The limitation of our work may be that it is to some extent dependent on the specifics of the environment, meaning that in other domains where states are defined differently and have no time dependencies our approach would not work. Despite these limitations, our work represents a step forward in the development of explainable reinforcement learning algorithms.

References

1. Abramson, J., Ahuja, A., Carnevale, F., Georgiev, P., Goldin, A., Hung, A., Landon, J., Lhotka, J., Lillicrap, T.P., Muldal, A., Powell, G., Santoro, A., Scully, G., Srivastava, S., von Glehn, T., Wayne, G., Wong, N., Yan, C., Zhu, R.: Improving multimodal interactive agents with reinforcement learning from human feedback. ArXiv (2022)
2. Adadi, A., Berrada, M.: Peeking inside the black-box: A survey on explainable artificial intelligence (xai). IEEE Access **6**, 52138–52160 (2018)
3. Alharin, A., Doan, T.N., Sartipi, M.: Reinforcement learning interpretation methods: A survey. IEEE Access **8**, 171058–171077 (2020)
4. Anonymous: Towards interpretable deep reinforcement learning with human-friendly prototypes. In: Submitted to The Eleventh International Conference on Learning Representations (2023), under review
5. Arrieta, A.B., Rodríguez, N.D., Ser, J.D., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Inf. Fusion **58**, 82–115 (2020)

[3] Video of prototype explanations: https://youtu.be/cItgedx6mqw

6. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS ONE **10** (2015)
7. Bengio, Y., Courville, A.C., Vincent, P.: Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**, 1798–1828 (2012)
8. Beyret, B., Shafti, A., Faisal, A.: Dot-to-dot: Explainable hierarchical reinforcement learning for robotic manipulation. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) pp. 5014–5019 (2019)
9. Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., Su, J.K.: This looks like that: Deep learning for interpretable image recognition. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019)
10. Chia, H.X.: In machines we trust: Are robo-advisers more trustworthy than human financial advisers? Law, Technology and Humans (2019)
11. Coppens, Y., Efthymiadis, K., Lenaerts, T., Nowé, A.: Distilling deep reinforcement learning policies in soft decision trees. In: International Joint Conference on Artificial Intelligence (2019)
12. Danesh, M.H., Koul, A., Fern, A., Khorram, S.: Understanding finite-state representations of recurrent policy networks. In: International Conference on Machine Learning (2020)
13. Gao, Y., Toni, F.: Argumentation accelerated reinforcement learning for cooperative multi-agent systems. In: Proceedings of the Twenty-First European Conference on Artificial Intelligence. p. 333–338. ECAI'14, IOS Press, NLD (2014)
14. Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M.A., Kagal, L.: Explaining explanations: An overview of interpretability of machine learning. 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA) pp. 80–89 (2018)
15. Google: Google Research, Brain Team: Imagen. https://imagen.research.google, accessed: 2022-12-10
16. Heuillet, A., Couthouis, F., Rodríguez, N.D.: Explainability in deep reinforcement learning. Knowl. Based Syst. **214**, 106685 (2020)
17. Juozapaitis, Z., Koul, A., Fern, A., Erwig, M., Doshi-Velez, F.: Explainable reinforcement learning via reward decomposition (2019)
18. Kawano, H.: Hierarchical sub-task decomposition for reinforcement learning of multi-robot delivery mission. 2013 IEEE International Conference on Robotics and Automation pp. 828–835 (2013)
19. Kim, B., Koyejo, O., Khanna, R.: Examples are not enough, learn to criticize! criticism for interpretability. In: NIPS (2016)
20. Li, O., Liu, H., Chen, C., Rudin, C.: Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In: AAAI Conference on Artificial Intelligence (2017)
21. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Artif. Intell. **267**, 1–38 (2017)
22. Montavon, G., Samek, W., Müller, K.R.: Methods for interpreting and understanding deep neural networks. Digital Signal Processing **73**, 1–15 (2018)
23. phi Nguyen, A., Mart'inez, M.R.: On quantitative aspects of model interpretability. ArXiv (2020)
24. OpenAI: ChatGPT: Optimizing language models for dialogue. https://openai.com/blog/chatgpt/, accessed: 2022-12-10

25. Raffin, A., Hill, A., Traoré, K.R., Lesort, T., Rodríguez, N.D., Filliat, D.: Decoupling feature extraction from policy learning: assessing benefits of state representation learning in goal based robotics. ArXiv (2018)
26. Ragodos, R., Wang, T., Lin, Q., Zhou, X.: Protox: Explaining a reinforcement learning agent via prototyping. NeurIPS 2022 **abs/2211.03162** (2022)
27. Richard Evans, J.G.: Deepmind AI reduces google data centre cooling bill by 40%. https://www.deepmind.com/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-by-40, accessed: 2022-12-10
28. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence **1**, 206–215 (2018)
29. Santhanam, G.K., Alami-Idrissi, A., Mota, N., Schumann, A., Giurgiu, I.: On evaluating explainability algorithms (2019)
30. Schulman, J.: Optimizing Expectations: From Deep Reinforcement Learning to Stochastic Computation Graphs. Ph.D. thesis, EECS Department, University of California, Berkeley (Dec 2016)
31. van Seijen, H., Fatemi, M., Laroche, R., Romoff, J., Barnes, T., Tsang, J.: Hybrid reward architecture for reinforcement learning. In: NIPS (2017)
32. Sequeira, P., Gervasio, M.T.: Interestingness elements for explainable reinforcement learning: Understanding agents' capabilities and limitations. Artif. Intell. **288**, 103367 (2019)
33. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., Hassabis, D.: Mastering the game of Go with deep neural networks and tree search. Nature **529**(7587), 484–489 (jan 2016)
34. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. CoRR **abs/1312.6034** (2013)
35. Tesla: Tesla: Autopilot and full self-driving capability. https://www.tesla.com/support/autopilot, accessed: 2022-12-10
36. Xian, Y., Fu, Z., Muthukrishnan, S., de Melo, G., Zhang, Y.: Reinforcement knowledge graph reasoning for explainable recommendation. Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (2019)
37. Zhao, K., Wang, X., Zhang, Y., Zhao, L., Liu, Z., Xing, C., Xie, X.: Leveraging demonstrations for reinforcement recommendation reasoning over knowledge graphs. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (2020)
38. Zhou, J., Gandomi, A.H., Chen, F., Holzinger, A.: Evaluating the quality of machine learning explanations: A survey on methods and metrics. Electronics (2021)