Towards Future-Based Explanations for Deep RL Network Controllers

Sagar Patel[1], Sangeetha Abdu Jyothi[1,2], and Nina Narodytska[2]

[1]University of Califoria, Irvine [2]VMware Research

ABSTRACT

Lack of explainability is hindering the practical adoption of high-performance Deep Reinforcement Learning (DRL) controllers. Prior work focused on explaining the controller by identifying salient features of the controller's input. However, these feature-based methods focus solely on inputs and do not fully explain the controller's policy. In this paper, we put forward *future-based explainers* as an essential tool for providing insights into the controller's decision-making process and, thereby, facilitating the practical deployment of DRL controllers. We highlight two applications of future-based explainers in the networking domain: online safety assurance and guided controller design. Finally, we provide a roadmap for the practical development and deployment of future-based explainers for DRL network controllers.

1. INTRODUCTION

Deep Reinforcement Learning (DRL), in lab settings, offers state-of-the-art performance in increasingly more problems in the networking domain, such as load balancing, network traffic engineering, congestion control, and adaptive bitrate streaming. However, DRL controllers lack real-world deployment because operators cannot interpret, debug, or trust them [5].

The domain of eXplainable Reinforcement Learning (XRL) has emerged to address this lack of trust. At its core, XRL aims to explain the decision-making process of a learned controller to humans [1]. Prior work has interpreted the controller's actions by highlighting the important features given to the controller. Metis [5] applies the concepts of decision tree distillation and critical path identification to generate interpretations. Trustee [3] further builds on the process of distillation by introducing ways to improve fidelity and generating an associated trust report. We broadly categorize these works as *feature-based*.

Feature-based explainers have proven their effectiveness in a number of applications. They can identify issues with the feature set [3], dataset [3], and model architecture [5]. However, they do not capture the forward-looking objective of the controller's decision-making process and thus cannot provide a comprehensive understanding of the controller.

In this work, we present a new perspective on explainability that we define as *future-based*. This approach focuses on presenting a future-oriented perspective of the controller by capturing goals or rewards. In networking applications, future rewards, in particular, are human-designed and represent the key performance metrics of the network application. In this case, future-based explanations based on reward components can provide meaningful insights into future performance metrics, which are meaningful to network operators. For instance, by analyzing future rewards in congestion control, we can obtain insights into the upcoming performance of the controller in terms of throughput, latency, and loss.

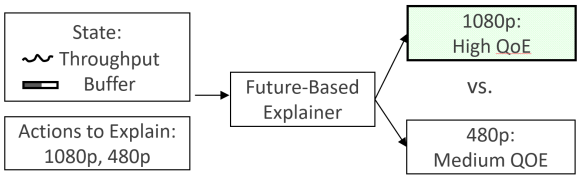
We highlight two key benefits of gaining a future-based understanding of DRL network controller behavior. First, it can provide insights for fine-tuning the algorithm parameters and the reward function during DRL controller design, which is a tedious and resource-inefficient process. Second, during the practical deployment of DRL controllers, future-based explainers can enable online safety assurance [6] by supporting network observability and preemptively triggering alerts for upcoming performance declines.

Recent work has introduced future-based explainers for gaming and robotic environments [4, 2]. However, these solutions are not adopted in practice since they either require accurately modeling the environment or require significant changes to the controller, which is often not feasible or detrimental to controller's performance. We outline the key research challenges towards developing practical future-based explainability frameworks in the networking domain. First, the forward-looking view of the controller contains a vast amount of information; capturing it succinctly and precisely is crucial for practical adoption. Second, future-based explainers must have low-latency explanations to spot safety violations before they happen, which is critical for tasks like online safety assurance. Third, they must function separately from the controller. This ensures they can be broadly applied without making extensive changes to the controller that could negatively affect the performance. Fourth, the explainers should be robust to malicious attacks, noise, and distribution shifts, thereby avoiding a false sense of security.

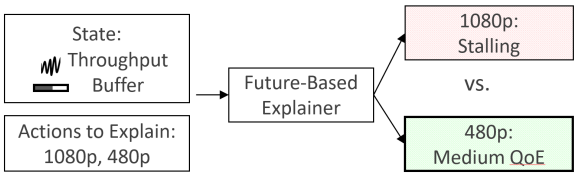
2. BACKGROUND

In this section, we provide a background of Reinforcement Learning and Adaptive Bitrate Streaming.

Reinforcement Learning. In Reinforcement Learning, an agent interacts with an environment. It is given a state s_t, and takes an action a_t according to its policy $\pi(A|s_t)$. The environment reacts to the agent's action and gives back to it the reward r_t, along with the next state s_{t+1}. The goal of the agent is to change its policy such as to maximize the reward over time, defined as the return $G = \sum_{t=0}^{\infty} \gamma^t r_t$.



(a) A contrastive future-based explanation for actions within state S_1



(b) The contrastive explanation for actions within state S_2

Figure 1: We illustrate how future-based explainers can provide insights across states and actions. We consider two seemingly similar states, S_1 and S_2, and seek to understand why the controller prefers different actions in them. We query the explainers with two actions: 1080p and 480p under both state S_1 (a) and S_2 (b). We can then peek into the future impact of these actions under both states and understand that 1080p is preferred in S_1 because it leads to high QoE, but it is not preferred in S_2 because it causes stalling.

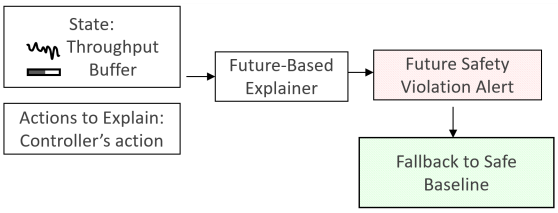


Figure 2: Online Safety Assurance: With the ability to capture the future performance of the controller, future-based explainers can be used to raise alerts about safety violations before they occur, falling back to a safe baseline and guaranteeing tail-ended performance.

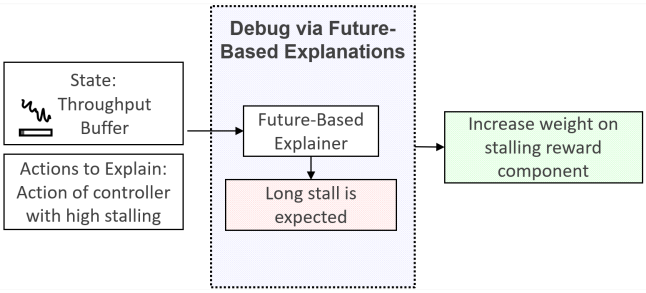


Figure 3: Guided Controller Design: Future-based explanations can help tune controller design. For a controller acting aggressively under poor network conditions, these explanations reveal stalling as an expected outcome. This insight helps the operator see that the reward function needs greater penalties for stalling.

Adaptive Bitrate Streaming. Adaptive Bitrate Streaming (ABR) works by dividing the video into chunks and encoding them at various discrete bit rates. During streaming, the most suitable bit rate for each chunk is chosen based on network conditions. The client also has a short buffer that can hold chunks yet to be seen. The ABR controller sequentially selects the bitrate to maximize the client's Quality of Experience (QoE), a numerical measure that awards high quality, and penalizes both changes in quality and stalling.

3. FUTURE-BASED EXPLAINERS

In this section, we first provide an overview of future-based explainers and how they can concretely enhance explainability. Next, we highlight two key applications of future-based explainers in the networking domain towards facilitating practical deployment of DRL controllers.

3.1 Overview

Future-based explainers shed light on the future goals or performance of the DRL controller. To train a future-based explainer, we take three inputs: the DRL controller, the simulation environment, and the training traces. We then roll out the controller, collecting the states, actions, and rewards the controller gets while interacting with the simulation environment. Finally, we use this interaction data to train the future-based explainer.

During inference, we can query the future-based explainer with a state and action to obtain a view into the impact of that action—getting explanations built around future states or rewards.

As a concrete example, let us consider a future-based explainer of an ABR controller that captures future performance through rewards and a scenario where the operator is looking to understand why the controller chooses different actions under seemingly similar network conditions. The operator selects the states representing this scenario and queries the explainer for the different actions it takes. In Figure 1, we visualize this scenario. We want to understand why the controller prefers to send a 1080p video chunk in S_1 while a 480p chunk in S_2, despite both of them having unstable throughput and similar buffer occupancy. We query a future-based explainer with both actions in both states. We can see that in S_2, the 1080p action is likely to lead to stalling and is thus avoided. Meanwhile, because the same is not expected in S_1, the 1080p action provides a higher quality of experience.

Thus, future-based explanations offer a medium to compare the impact of different actions from within and across multiple states.

3.2 Applications

Next, we discuss how insights offered by future-based explanations can be leveraged in the design and deployment of DRL controllers. We highlight two key applications: guided controller design and online safety assurance.

Guided Controller Design. Implementing practical DRL solutions demands various design choices. These range from selecting the feature set, picking the DRL algorithm and its hyperparameters to designing the reward function and learning parameters. Tuning of these design parameters is

a tedious and resource-inefficient process in practice, even for DRL experts. Typically, these parameters are tuned through a trial-and-error process.

Future-based explainers can aid in this design process. They offer insight into the exact factor the DRL algorithm optimizes: future performance. To demonstrate their utility in identifying DRL algorithm issues, we examine an example. In Figure 3, we debug an aggressive controller under poor network conditions. Using a future-based explainer, we find that the controller, despite anticipating long stalls due to its actions, still opts for them. This discovery suggests to the operator the need to increase the penalty for the stalling reward component.

Online Safety Assurance. Online Safety Assurance poses the challenge of detecting when the learning-based policy is likely to reach an unsafe state and avoiding it by falling back to a reliable and extensively tested baseline policy. This fallback mechanism acts as a "safety net" for learning-based systems, designed to facilitate high-performance outcomes under ideal circumstances while also offering minimum performance guarantees under less than perfect conditions [6]. Existing research has suggested that the problem can be addressed by quantifying uncertainty within the learning policy [6], where uncertainty serves as an indirect measure of potential unsafe states. This has been accomplished either through an ensemble method or novelty detection.

In this context, future-based explainers can be applied directly to foresee and alert for possible unsafe behavior without the need for a proxy. Figure 2 illustrates how such a system would work. The future-based explainer would receive the current state as input, predict the controller's future behavior, and issue warnings for potential safety violations. In response to these warnings, a fallback to a safe baseline can be triggered. This ensures the overall system maintains compliance with safety requirements and performance commitments.

4. KEY CHALLENGES

In this section, we describe the main research challenges in developing practical future-based explainers.

Concise Explanations. Future-based explainers must create their explanations by considering the future: a series of states, actions, and rewards. The challenge lies in converting this complex information into a format that humans can easily understand.

Low-Latency Inference. To support real-time applications such as online safety assurance, future-based explainers must provide explanations promptly. Moreover, this process should not disrupt the primary operations of the controller. In short, generating explanations must add minimal cost to the controller's decision latency. Fortunately, future-based explainers can offer insights beyond a single step in the future. Thus, they can function in parallel without being on the critical path of the controller.

Separation from the Controller. To ensure broad applicability, future-based explainers should not require significant modifications to the controller. Such changes can harm the controller's performance, introducing a performance and explainability trade-off. Instead, explainers should leverage the controller's inner workings, such as its learned features, without altering them.

Robustness of Explanations. The ability to create explanations that are robust to malicious attacks, noise, and shifts in distribution is a significant, unresolved challenge. Notably, even several widely-used feature-based explainers have proven susceptible to these threats [8]. However, building on early intuitions [7] and addressing this issue is critical to support trust-sensitive applications such as online safety assurance.

5. CONCLUSION

In this paper, we present an initial perspective on a new angle of explainability with future-based or forward-looking explainers. We highlight their ability to power guided controller design and enable online safety assurance. We then provided a road map for practically implementing future-based explainers by detailing key open research challenges. We envision this work to lay the foundation for a broad application of future-based explainers.

6. REFERENCES

[1] Nadia Burkart and Marco F Huber. A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70:245–317, 2021.

[2] Francisco Cruz, Richard Dazeley, Peter Vamplew, and Ithan Moreira. Explainable robotic systems: Understanding goal-driven actions in a reinforcement learning scenario. *Neural Computing and Applications*, pages 1–18, 2021.

[3] Arthur S Jacobs, Roman Beltiukov, Walter Willinger, Ronaldo A Ferreira, Arpit Gupta, and Lisandro Z Granville. Ai/ml for network security: The emperor has no clothes. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 1537–1551, 2022.

[4] Zoe Juozapaitis, Anurag Koul, Alan Fern, Martin Erwig, and Finale Doshi-Velez. Explainable reinforcement learning via reward decomposition. In *IJCAI/ECAI Workshop on explainable artificial intelligence*, 2019.

[5] Zili Meng, Minhu Wang, Jiasong Bai, Mingwei Xu, Hongzi Mao, and Hongxin Hu. Interpreting deep learning-based networking systems. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, pages 154–171, 2020.

[6] Noga H Rotman, Michael Schapira, and Aviv Tamar. Online safety assurance for learning-augmented systems. In *Proceedings of the 19th ACM Workshop on Hot Topics in Networks*, pages 88–95, 2020.

[7] Dylan Slack, Anna Hilgard, Sameer Singh, and Himabindu Lakkaraju. Reliable post hoc explanations: Modeling uncertainty in explainability. *Advances in neural information processing systems*, 34:9391–9404, 2021.

[8] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020.