Do LLMs Explain or Remember? Evaluating Language Model Explanations of Reinforcement Learning Failures

Nivetha Suntharamoorthy[1], Jørn Eirik Betten[2], Dennis Gross[2,3], Eirik Valseth[1,2] and Helge Spieker[2,*]

[1]Norwegian University of Life Sciences (NMBU), Ås, Norway
[2]Simula Research Laboratory, Oslo, Norway
[3]Institut für Kommunikations- und Prüfungsforschung gGmbH, Heidelberg, Germany

Abstract

We explore the capabilities of large language models (LLMs) to explain failures of reinforcement learning (RL) agents. Our study systematically investigates how prompt structure, model architecture, and environment representation affect the quality of Explainable RL (XRL) explanations. We conduct a multi-model evaluation using quantitative metrics to assess explanation coherence, helpfulness, and stability. To isolate interference from prior knowledge, we introduce an obfuscated RL environment that masks canonical identifiers while preserving underlying dynamics. This approach enables us to test whether LLMs ground their reasoning in observed sequential data or default to memorized assumptions about standard environments.

Keywords

Reinforcement Learning, Large Language Models, Explainable AI, LLM, Explainable RL, Trustworthy AI

1. Introduction

Reinforcement Learning (RL) has achieved remarkable success in sequential decision-making [1]. However, the resulting policies often remain opaque, making them challenging to deploy in safety-critical domains that require trust and accountability [2]. While Explainable AI (XAI) addresses model transparency, Explainable RL (XRL) introduces distinct challenges. Unlike supervised learning, RL involves temporal dependencies, delayed rewards, and exploration strategies, making long-term behaviors difficult to interpret post-hoc [3, 4]. In complex or stochastic environments, counter-intuitive but optimal actions, such as stability-driven high-thrust maneuvers to land a spacecraft, can easily be misinterpreted as failures without robust explainability tools [5, 6].

Large Language Models (LLMs) provide a simple interface for XRL by translating numerical state-action trajectories into natural-language explanations. However, integrating LLMs into XRL pipelines is non-trivial. LLM outputs are highly sensitive to prompt design [7, 8] and prone to hallucinations or over-reliance on prior knowledge acquired during pretraining. An open question is whether LLM-generated explanations are genuinely grounded in the specific trajectory data provided, or if they default to memorized assumptions about canonical RL environments [9].

To investigate this, we present an evaluation that compares GPT-5, Llama 3.2, and DeepSeek-R1 to assess how prompt structure, environment complexity, and architectural biases affect the quality of XRL explanations. We evaluate explanation stability under controlled perturbations, using multiple metrics and the LLM-as-a-judge approach. We further introduce an *obfuscated* version of the Lunar Lander environment [10] to isolate interference from prior knowledge. By hiding the environment's identity while preserving its dynamics, we test whether LLMs adapt to the observed sequential data or hallucinate behaviors based on pretraining priors.

TRUST-AI: The Second European Workshop on Trustworthy AI. Organized as part of the International Joint Conference on Artificial Intelligence - IJCAI/ECAI 2026. August 2026, Bremen, Germany.
*Corresponding author.
nivethasu@outlook.com (N. Suntharamoorthy); jorneirik@simula.no (J. E. Betten); dennis@artigo.ai (D. Gross); eirikva@simula.no (E. Valseth); helge@simula.no (H. Spieker)

This paper explores the relationship between prompt design, model choice, and environment representation in XRL, making the following key contributions:

1. **LLM-XRL Evaluation Framework:** We design a systematic evaluation framework for assessing LLM-generated explanations of RL agent failures.
2. **Multi-Model XRL Evaluation:** We perform a comparative analysis of explanation quality, coherence, and helpfulness across GPT-5, Llama 3.2, and DeepSeek-R1.
3. **Prompt Sensitivity Analysis:** We systematically evaluate how prompt design influences explanation consistency in sequential tasks.
4. **Prior Knowledge Interference Study:** We empirically test whether LLMs rely on explicit observation data or memorized pretraining priors, using environment obfuscation.

In the following, we first introduce the necessary background and an overview of related work (Sec. 2), then present our evaluation framework (Sec. 3), and finally discuss and analyze the results (Sec. 4). We conclude with a discussion of the findings, their limitations, and future perspectives (Sec. 5).

2. Background & Related Work

Reinforcement Learning Reinforcement Learning [1] is a branch of Machine Learning (ML) that addresses sequential decision-making tasks in which an agent interacts with an environment. In RL, these problems are formulated as a Markov Decision Process (MDP), which is defined by a four-tuple $\langle \mathcal{S}, \mathcal{A}, T, R \rangle$, where $\mathcal{S}$ is the state space of the environment, $\mathcal{A}$ is the action space of the agent, $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition function from one state to another (dependent only on the immediate previous state and the agent action in that state, hence Markovian), and $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function defining desired behavior of the agent. If the state is not fully observable to the agent, we have a *Partially-Observable Markov Decision Process* (POMDP) and the four-tuple becomes a five-tuple $\langle \mathcal{S}, \mathcal{O}, \mathcal{A}, T, R \rangle$, where $\mathcal{O}$ is the observation space defining what parts of the state space can be observed. In either case, the goal of RL is to learn desired behavior through trial and error by maximizing the cumulative rewards.

Explainable RL Although RL policies perform well on many challenging sequential decision-making tasks, for these models to be relevant for real-world deployment, they must be trusted by humans, which requires a rationale for *why* these agents act as they do. Therefore, the field of Explainable RL (XRL) focuses on enhancing the transparency of RL agents' decision-making processes, including the identification of critical states [11, 12, 13], where different decisions have a high impact on the outcome of the episode, and the explanation of failures, i.e., identifying why an agent failed, which is not necessarily obvious.

Machine learning models, like neural networks, are often considered *black-box models*, meaning their inner mechanisms are hidden from the user, and extensive analysis is required to understand *how* the model maps inputs to outputs. This opacity is compounded in deep RL, where failures emerge from sequences of such opaque decisions rather than single predictions.

LLMs for Explanation Many XAI and XRL methods are tailored towards specific types of technical explanations, which are not necessarily understandable to a (non-expert) human without additional context and information enrichment. Recently, LLMs have been prompted for explainability tasks, both to find explanations and to present and communicate explanations to a user. Bilal et al. [14] survey this emerging use of LLMs across XAI, spanning post-hoc explanation generation, translation of technical explanations into natural language, and interactive explanation dialogues. For RL specifically, Gross et al. [15] use LLMs to enrich formal analyzes of RL policies with natural-language context, to analyze and explain which alternative actions should have been selected in safety-critical states. These approaches treat the LLM as a trusted explanation interface. Whether that trust is warranted is largely untested, since LLM outputs are sensitive to prompt framing and can produce plausible but ungrounded

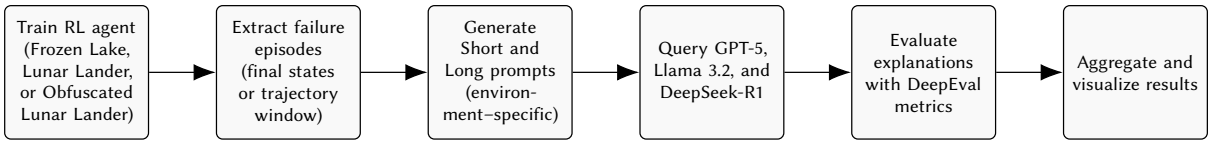


Figure 1: Overview of our experimental pipeline.

rationales [7]. Our study addresses this gap: rather than proposing a new LLM-based explanation method, we evaluate the explanations themselves, using environment obfuscation to isolate whether they are grounded in the observed trajectory or in memorized knowledge of the environment.

3. LLM Explanations of Reinforcement Learning Failures

3.1. Evaluation Framework

We present our evaluation method for analyzing LLM explanations of RL failures. The overall pipeline is outlined in Figure 1, starting from the RL agent training and failure extraction, over LLM prompting, to a final evaluation and analysis stage.

3.2. Research Questions

We structure our analysis alongside these four research questions (RQs):

RQ1 *Prompt Effect*: How do prompt length and contextual depth influence the quality of failure explanations?
RQ2 *Model Comparison*: Do different LLMs exhibit systematic differences in failure explanations?
RQ3 *Prior Knowledge*: Can LLMs explain failures without relying on memorized domain knowledge?
RQ4 *Task Complexity*: Do the environment characteristics influence the best explanation setup?

3.3. Environments

We use three environments in our experiments: Frozen Lake, Lunar Lander, and an obfuscated Lunar Lander version. We select these environments because they are well-known in the literature and of low complexity, but they still provide ample opportunities for the agent to make mistakes. Frozen Lake has very immediate failure scenarios, e.g., falling into water, whereas in Lunar Lander cause and effect are spread over time.

3.3.1. Frozen Lake

In Frozen Lake [16], the agent traverses a frozen lake with holes to avoid and a goal to reach. The state and action spaces are discrete: the state is an 8×8 grid, and the available actions are left, up, down, and right. The transition is stochastic in the sense that there is a chance that the agent 'slips' when moving in a direction, resulting in a movement in one of the orthogonal directions.

3.3.2. Lunar Lander

In the two-dimensional Lunar Lander environment [10], the agent is a landing aircraft that must land within two bounds on a one-dimensional surface without tipping over. The eight-dimensional state contains six continuous values: the $x-$ and $y-$coordinates of the landing aircraft's position and linear velocities, its angle, and its angular velocity. Additionally, it contains two boolean values representing contact with the surface of each leg. The action space is discrete, with four possible actions: do nothing, fire the left orientation engine, fire the main engine, and fire the right orientation engine.

In the *obfuscated Lunar Lander* variant, we reversed the state representation in the prompt to the LLM and removed all references mentioning "Lunar Lander".

Frozen Lake: Provide a short and precise analysis of why the agent failed in the Frozen Lake environment.
Map details: S = Start, F = Frozen, H = Hole, G = Goal, X = Agent's position
Second to last state: SFFF XHFH FFFH HFFG
Last action: RIGHT
Last state: SFFF FXFH FFFH HFFG
Please include specific factors that contributed to this failure.

Lunar Lander: Provide a short and precise analysis of why the agent failed in the Lunar Lander environment. State vector details: $[x, y, x\_velocity, y\_velocity, angle, angular\_velocity, left\_leg\_contact, right\_leg\_contact]$
Here are the last 50 steps of the trajectory (state, action, reward at each step):
Step 452: State=[0.12779226899147034, 0.10060523450374603, 0.4102094769477844, 0.04173511266708374, 0.5097846984863281, 0.8042332530021667, 0.0, 0.0], Action=Fire main engine, Reward=-1.6845600531330007
...
Step 501: State=[0.12647847831249237, 0.007400760427117348, -0.24154384434223175, -0.2548162043094635, 2.0596859455108643, 1.6378753185272217, 0.0, 0.0], Action=Fire main engine, Reward=-100.0
Total reward: -250.42298615586708
Please explain the main reasons for failure.

Figure 2: Short Prompt Examples for Frozen Lake and Lunar Lander. The long prompts include a detailed environment documentation to provide additional context to the LLM.

3.4. Large Language Models

We consider a selection of three LLMs, covering both open-weight and commercial models, and ranging from small to large model size: *Llama 3.2 11B* [17] from Meta is an LLM with 11 billion parameters released in September 2024. *DeepSeek-R1* [18] is a reasoning model from DeepSeek that contains 671B parameters and was released in January 2025. *GPT-5* from OpenAI is a commercial frontier LLM (with unknown parameter count) and was released in August 2025.

We acknowledge that the selection of models is a snapshot in time, and, as fast-paced as this field is, newer models will behave differently and are (presumably) more capable. Nevertheless, the selected models cover a variety of representative, state-of-the-art models with widespread usage, providing a broad perspective on the explanation capabilities of LLMs for RL failures.

3.5. Prompts

We designed two types of prompts: a short prompt with minimal context on the environment, and a longer prompt that included the environment's documentation.

Figure 2 shows two example prompts, one for Frozen Lake, where the task is to explain a specific failure that occurred between two states, and one for Lunar Lander, where the task is to explain a failure given a longer subtrajectory and describe the general causes for the overall failure. Both prompts are examples of *short* prompts. The longer prompt variants additionally include a detailed documentation of the environment, taken directly from Gymnasium's documentation [19].

3.6. Metrics

We select six metrics to analyze the explanation quality of the LLMs. Table 1 lists the metrics and their intended meaning. This selection of metrics was made to obtain a comprehensive picture of the explanations.

These metrics follow the LLM-as-a-judge pattern [22], where we queried GPT-5 to evaluate responses against them, using the DeepEval framework [23]. The use of GPT-5 to evaluate responses by GPT-5 among other models might introduce a self-preference bias [24], where a model scores responses from its own model family better without factual reasons. How to assess and mitigate the extent of this bias is an area of active research [25, 26, 27]. We acknowledge this potential threat and comparisons

Table 1
Evaluation metrics and their intention [20, 21]

Metric	Description
Prompt Alignment	Measures how well the generated explanation adheres to the instructions specified in the prompt.
Correctness	Evaluates the factual and logical accuracy of the explanation with respect to the agent's actual trajectory and actions; compared against a human-provided baseline explanation.
Helpfulness	Quantifies the extent to which the explanation supports human understanding of the agent's behavior.
Conciseness	Assesses how efficiently the explanation communicates relevant information.
Relevance	Measures whether the explanation focuses on the most important aspects of the trajectory and environment.
Coherence	Evaluates the logical structure, readability, and consistency of the explanation.

between models should be interpreted with this in mind. We chose GPT-5 as the judge because it was the most capable model available at the time of study design.

To further confirm the validity of the LLM-as-a-judge setup, one author manually reviewed all judged explanations, spanning all models, environments, and prompt types, and compared the metric scores against their own assessment. Since this validation involved a single rater, we report no inter-rater agreement and consider it a plausibility check rather than a formal annotation study; a systematic human evaluation remains future work. In our review, the judge proved mostly reliable. The most common issue was a failure to follow the prompt to be concise, resulting in long explanations. Similarly, the conciseness metric overly favored very short responses (as expected), even though this is sometimes at odds with the other metrics. However, this is a natural trade-off among the metrics and needs to be considered accordingly.

3.7. Data Collection

We collected failures by repeatedly executing DQN (Deep Q Networks) [28] policies for the three selected environments. In total, we collected 47 failures for Frozen Lake and 44 for Lunar Lander, giving 546 explanations across the three LLMs and two prompt types, plus 132 for the obfuscated environment with a single prompt. We ensured that failures were unique by requiring that terminal states be distinct. While we ensured that the policies perform well in the environments to avoid the most basic failures, we note that the quality of the policies is mostly irrelevant to the scope of our study, i.e., how well LLMs can *explain* failures by observing the policy's behavior in the environment immediately before failing does not depend on the quality of the policy. The accompanying source code for the analysis is available online[1].

4. Empirical Evaluation

In the following, we analyze the collected data from the perspective of the four research questions on *prompt effect* of short versus long prompt, *model comparison* among the LLMs, *prior knowledge* on the original Lunar Lander versus the obfuscated version, and the *task complexity* of explaining a single failure decision versus highlighting overall behavioral mistakes.

4.1. RQ1: Prompt Effect

In this section, we address RQ1: *How do prompt length and contextual depth influence the quality of failure explanations?* To investigate prompt effects across environments and LLMs, we aggregated the results of our collected data for the short and long prompts along three axes: the average score for each

[1]URL: https://github.com/ExplainableRL/Thesis-code

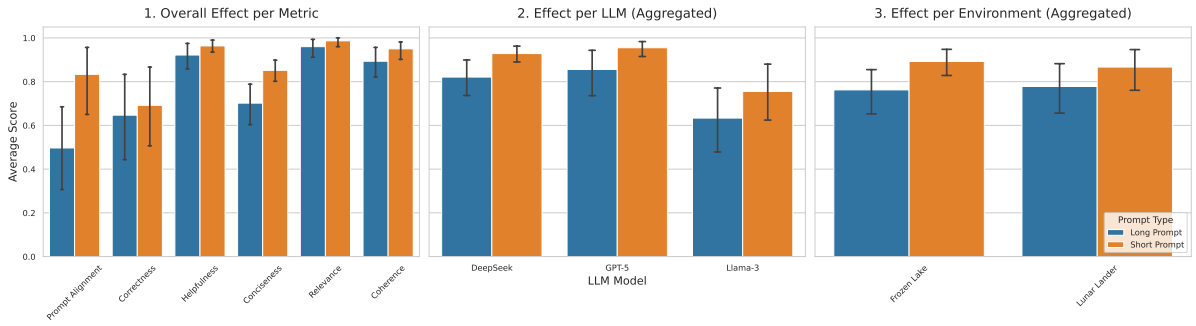


Figure 3: Evaluation of the two prompt types on individual metrics, LLMs, and environments.

metric across all LLMs in both environments, the average score for each LLM across all metrics and both environments, and the average score across all metrics and LLMs for each environment. These results are shown in Figure 3 from left to right, in that order.

Figure 3.1 shows that the short-prompt strategy consistently yields higher scores than the long-prompt strategy across all metrics, with the largest average-score gaps between them on the prompt alignment and conciseness metrics, suggesting a correlation between the verbosity of the prompt and the verbosity of the resulting LLM explanation. We observe minor drops in average scores for the other metrics, indicating that the LLMs' explanations are quite stable across prompt designs in terms of correctness, helpfulness, relevance, and coherence.

We present the aggregation across all metrics for each LLM in Figure 3.2, and we find similar gaps between prompt types across all LLMs, although GPT-5 and DeepSeek consistently achieve higher explanation quality than Llama 3.2. These results show that the short prompt design consistently provides better explanations across our LLM model selection.

In Figure 3.3, we grouped the results by environment and aggregated the scores across LLMs and metrics to separate how the short and long prompts affected explanation quality in each of them. The average score for each prompt type is very similar across environments, indicating that the LLMs provided approximately equally good explanations in both environments, with the same score gaps between the short and long prompts. This further indicates that a less contextual prompt provides better explanations, independent of the explanation task. Additionally, the consistent explanation quality across environments suggests that explanation tasks are of similar difficulty.

The explanations for the policy failures provided by the LLMs were consistently better when prompted with the short-prompt design than with the long-prompt design, particularly in terms of prompt alignment and conciseness. These results suggest that explanation quality degrades when contextual depth is excessive.

4.2. RQ2: Model Comparison

In this section, we provide our analysis of RQ2, namely, whether different LLMs exhibit systematic differences in their explanations of failures. To this end, in Figure 4 we present box plots for each explanation metric, separately for each LLM and prompt design.

Across all metrics, we observe that Llama 3.2 provides explanations of consistently lower quality than the other LLMs, most notably in terms of explanation correctness. DeepSeek and GPT-5 provide substantially better explanations than Llama 3.2, with GPT-5 scoring slightly higher than DeepSeek across all metrics. The smallest model, Llama 3.2, exhibits higher median correctness when provided with more context, whereas GPT-5 and DeepSeek produce more correct explanations when prompted with less context.

Although scores generally drop with the long prompt compared to the short prompt, GPT-5's explanations are stable across prompt types for the helpfulness, relevance, and coherence metrics. However, the prompt alignment score is substantially lower for GPT-5 than for DeepSeek when they

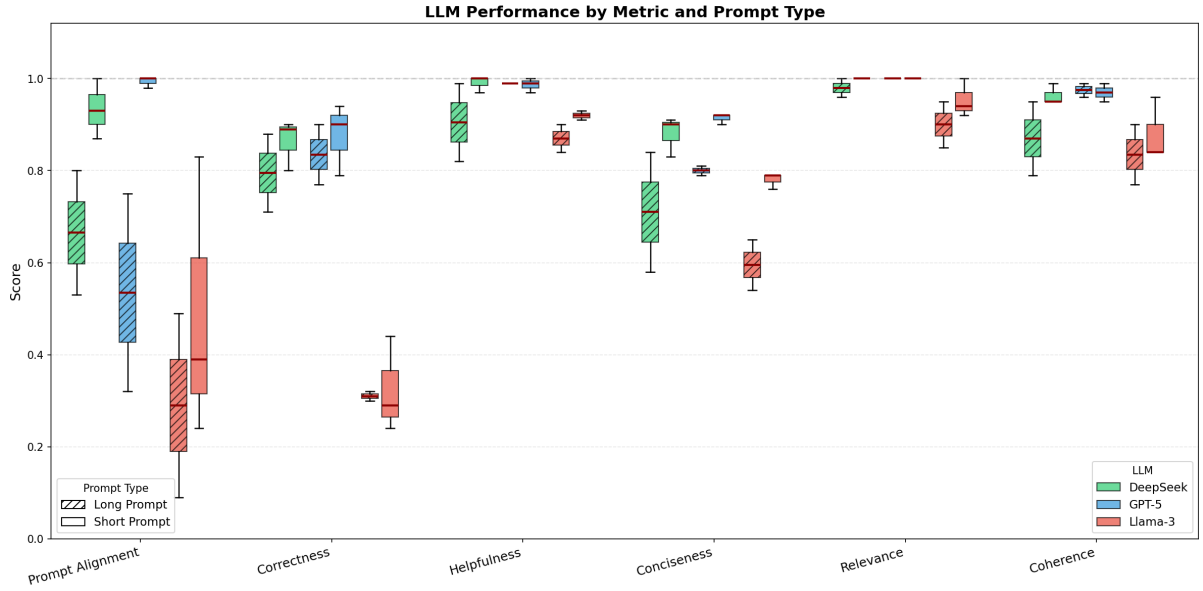
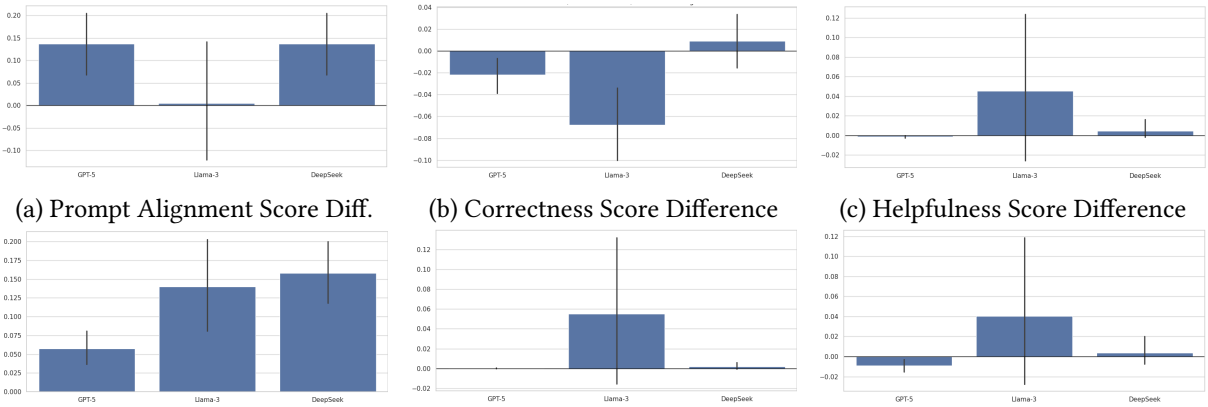


Figure 4: Boxplot distributions of each metric, reported per prompt type, and LLM.



(a) Prompt Alignment Score Diff. (b) Correctness Score Difference (c) Helpfulness Score Difference

(d) Conciseness Score Difference (e) Relevance Score Difference (f) Coherence Score Difference

Figure 5: Prior knowledge: Difference plots for Lunar Lander (Difference: Obfuscated - Original) for all metrics. x-axis: LLM; y-axis: Score Difference (Obfuscated Score - Original Score)

are given the long prompt, although GPT-5 explanations are consistently higher-scoring across all other metrics when the prompt type is taken into account.

Overall, we observe a drop in the score for all LLMs when prompted with more context, although we find some differences between the models. First, our most advanced model, GPT-5, exhibits a dramatic drop in prompt alignment as context increases, whereas our simplest model, Llama 3.2, yields approximately equally correct explanations under the same conditions. This might indicate that additional context misleads the model and that prior knowledge indeed has a stronger influence on the actual explanation quality.

4.3. RQ3: Prior Knowledge

In the following, we investigate RQ3: *Can LLMs explain RL failures without relying on memorized domain knowledge?* To answer this question, we added a third prompting strategy: removing all references to "Lunar Lander", while also transforming the trajectory representation to further obfuscate the environment's identity. The differences between the obfuscated and short prompt strategies on each explanation evaluation metric are shown in Figure 5.

In Figure 5, a positive difference in scores indicates that, on average across failures, the score was higher when the LLM was prompted with the obfuscated prompting strategy. Notably, the overall explanation quality remains stable, or even improves slightly, when the prompt obfuscates the environment's identity. The prompt alignment and conciseness scores substantially increase when references to Lunar Lander are removed from the prompt. When the environment's identity is hidden, the explanations generated by Llama 3.2 and GPT-5 achieve higher or similar evaluation scores across all metrics, except for correctness. Interestingly, Llama 3.2 and GPT-5 provide more concise, helpful, and to-the-point explanations when the environment is hidden, although they are more likely to produce incorrect explanations, suggesting they produce more fluent but less grounded explanations without environment context. DeepSeek's explanations improved across all metrics, including correctness, suggesting that removing context enabled DeepSeek to focus more on the information important to the explanation.

In Figure 5(b), we observe that the correctness of the explanations provided by GPT-5 and Llama 3.2 decreases when we hide the environment's identity, whereas the explanations given by DeepSeek are more correct under the same circumstances.

4.4. RQ4: Task Complexity

In this section, we investigate whether environment characteristics influence the choice of explanation setup (RQ4). In Figure 3, we observe that the long-prompt strategy yielded consistently worse explanations in both environments. Furthermore, the explanation quality generally increases when further context is removed by hiding the environment in Figure 5, with the caveat that the correctness may drop if too much information is withheld from the LLM. Together, these results suggest that limiting context when prompting LLMs to explain RL failures produces more useful explanations. However, we find no substantial differences between the explanation tasks of explaining failures in the Frozen Lake and Lunar Lander environments.

Our experiments, therefore, suggest that the best explanation setup is one that provides the LLM with the necessary, but not excessive, context to solve the explanation task.

5. Conclusion

We investigate whether LLMs can explain failures of RL agents and how explanation quality is affected by prompt design, model choice, environment complexity, and internalized prior knowledge. The results show that LLMs can generate meaningful, coherent, and often correct explanations of RL agent behavior. Across both environments, the models generally identified key failure mechanisms such as falling into holes in Frozen Lake or instability, velocity, and orientation errors in Lunar Lander. This shows that LLMs can serve as an effective interface between low-level numerical trajectories and human-understandable natural-language explanations.

The impact of prompt design was consistent rather than environment-dependent. Across both Frozen Lake and Lunar Lander, the short prompts produced better explanations than the long prompts, with the largest gaps on prompt alignment and conciseness, while correctness, helpfulness, relevance, and coherence remained comparatively stable. We found no substantial differences between the two environments in either absolute explanation quality or the size of the prompt-type gap. These results suggest that, at least for tasks of this complexity, adding full environment documentation to the prompt does not improve explanation quality and can degrade adherence to the instructions. Prompt design is therefore not only a practical detail but a central component of LLM-based XRL pipelines.

Model choice also mattered. GPT-5 achieved the highest scores across most metrics, closely followed by DeepSeek-R1, while Llama 3.2 produced explanations of consistently lower quality, most visibly in correctness. The models also differed in how they responded to added context: GPT-5 showed the largest drop in prompt alignment under the long prompt, whereas Llama 3.2 was the only model whose correctness improved with more context. Overall quality and robustness to prompt changes are thus separate properties; the strongest model in absolute terms was not uniformly the least sensitive to how the task was presented.

The obfuscated Lunar Lander experiment offered insight into interference from prior knowledge. Hiding the environment's identity left overall explanation quality stable or slightly improved for all three models, but the models diverged on correctness: GPT-5 and Llama 3.2 produced less correct explanations under obfuscation, while DeepSeek-R1 improved on every metric, including correctness. This pattern is consistent with GPT-5 and Llama 3.2 drawing on memorized knowledge of the canonical environment, and with DeepSeek-R1 grounding its explanations more directly in the provided trajectory. However, the relatively modest differences between the original and obfuscated conditions suggest that our modification, reversing the state representation and removing the environment name, may not fully suppress prior knowledge. Designing more radical environment transformations that cleanly separate memorization from trajectory-based reasoning is an important direction for future work.

From a broader perspective, the findings support the view that LLMs can play a meaningful role in XRL, but not as a standalone explainer. Automated evaluation frameworks, such as DeepEval, provide scalable evaluation, yet the study also indicated brittleness in the explanations and a potential overreliance on prior knowledge rather than an actual understanding of the task at hand. Explainability still depends on human perspectives, and the complete automation of its evaluation remains elusive, thereby opening several avenues for future research on guided prompting, and on structured, verifiable explanations.

Declaration on Generative AI

During the preparation of this work, the authors used Google Gemini and Anthropic Claude in order to: Paraphrase and reword. Grammar and spelling check. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] R. S. Sutton, A. G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.

[2] A. Mohamed, K. Abdelqader, K. Shaalan, Explainable artificial intelligence: A systematic review of progress and challenges, Intelligent Systems with Applications 28 (2025).

[3] A. B. Arrieta, et al., Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible ai, Information Fusion 58 (2020).

[4] Z. C. Lipton, The mythos of model interpretability, Queue 16 (2018).

[5] S. Milani, N. Topin, M. Veloso, F. Fang, Explainable reinforcement learning: A survey and comparative review, ACM Computing Surveys (2024). doi:10.1145/3616864.

[6] G. A. Vouros, Explainable deep reinforcement learning: State of the art and challenges, ACM Computing Surveys 55 (2022). doi:10.1145/3527448.

[7] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, J.-R. Wen, A survey of large language models, arXiv preprint arXiv:2303.18223 (2023).

[8] D. Zhou, et al., Large language models can be strong reasoners with proper prompting, arXiv preprint arXiv:2205.12287 (2022).

[9] A. Belouadah, M. L. Ruiz-Rodríguez, S. Kubler, Y. Le Traon, Evaluating the effectiveness of LLMs for explainable deep reinforcement learning, Machine Learning with Applications 22 (2025).

[10] Farama Foundation, Lunar lander environment — gymnasium, https://gymnasium.farama.org/environments/box2d/lunar_lander/, 2025. Accessed: 2025-12-06.

[11] S. H. Huang, K. Bhatia, P. Abbeel, A. D. Dragan, Establishing appropriate trust via critical states, in: IROS, 2018. doi:10.1109/IROS.2018.8593649.

[12] W. Guo, X. Wu, U. Khan, X. Xing, Edge: Explaining deep reinforcement learning policies, in: Advances in Neural Information Processing Systems, volume 34, 2021.

[13] H. Liu, M. Zhuge, B. Li, Y. Wang, F. Faccio, B. Ghanem, J. Schmidhuber, Learning to Identify Critical States for Reinforcement Learning from Videos, in: CVPR, 2023, pp. 1955–1965.

[14] A. Bilal, D. Ebert, B. Lin, LLMs for explainable ai: A comprehensive survey (2025). doi:10.48550/arXiv.2504.00125.
[15] D. Gross, H. Spieker, Enhancing rl safety with counterfactual llm reasoning, in: IFIP International Conference on Testing Software and Systems, Springer, 2024, pp. 23–29.
[16] Farama Foundation, Frozen lake environment — gymnasium, https://gymnasium.farama.org/environments/toy_text/frozen_lake/, 2025. Accessed: 2025-12-06.
[17] Llama Team, Llama 3.2 Model Card, https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD_VISION.md, 2025. Accessed: 2025-10-14.
[18] DeepSeek-AI, Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025. URL: https://arxiv.org/abs/2501.12948. arXiv:2501.12948.
[19] M. Towers, A. Kwiatkowski, J. Terry, J. U. Balis, G. De Cola, T. Deleu, M. Goulão, A. Kallinteris, M. Krimmel, A. KG, et al., Gymnasium: A standard interface for reinforcement learning environments, arXiv preprint arXiv:2407.17032 (2024).
[20] DeepEval Team, Prompt alignment metric documentation, https://deepeval.com/docs/metrics-prompt-alignment, 2025. Accessed: 2025-10-14.
[21] DeepEval Team, G-eval metrics: Correctness, helpfulness, conciseness, relevance, coherence, https://deepeval.com/docs/metrics-llm-evals, 2025. Accessed: 2025-10-14.
[22] D. Li, B. Jiang, L. Huang, A. Beigi, C. Zhao, Z. Tan, A. Bhattacharjee, Y. Jiang, C. Chen, T. Wu, K. Shu, L. Cheng, H. Liu, From generation to judgment: Opportunities and challenges of LLM-as-a-judge, in: EMNLP, 2025. doi:10.18653/v1/2025.emnlp-main.138.
[23] J. Ip, K. Vongthongsri, deepeval, 2026. URL: https://github.com/confident-ai/deepeval.
[24] D. Deutsch, R. Dror, D. Roth, On the limitations of reference-free evaluations of generated text, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, 2022. doi:10.18653/v1/2022.emnlp-main.753.
[25] K. Wataoka, T. Takahashi, R. Ri, Self-preference bias in LLM-as-a-judge, in: Neurips Safe Generative AI Workshop 2024, 2024. URL: https://openreview.net/forum?id=tLZZZIgPJX.
[26] Z.-Y. Chen, H. Wang, X. Zhang, E. Hu, Y. Lin, Beyond the surface: Measuring self-preference in LLM judgments, in: C. Christodoulopoulos, T. Chakraborty, C. Rose, V. Peng (Eds.), Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, 2025. doi:10.18653/v1/2025.emnlp-main.86.
[27] J. Yang, Z. Hu, C. Qiu, Z. Deng, X. Jiao, T. Zhou, Quantifying and mitigating self-preference bias of llm judges, arXiv preprint arXiv:2604.22891 (2026).
[28] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, et al., Human-level control through deep reinforcement learning, Nature (2015). doi:10.1038/nature14236.