Causal Evaluation of Planning Strategies in Large Language Models Through Interpretable Quality Prediction and Counterfactual Reinforcement Learning

Claudio Bryant
Department of Computer Science, University of North Texas, Denton, TX, USA.
claudio.work@unt.edu

Fernando Bennett
Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO, USA.
fernando487@missouri.edu

Vaibhav M. Chatterjee
Department of Computer Science, University of Central Florida, Orlando, FL, USA.
vmchatterjee@ucf.edu

Abstract

Large language models have demonstrated remarkable reasoning capabilities, yet their planning strategies remain opaque and difficult to evaluate systematically. This paper proposes a causal evaluation framework that combines interpretable quality prediction with counterfactual reinforcement learning to assess and improve the planning processes of LLMs. We argue that traditional evaluation metrics based solely on final outcome accuracy are insufficient for understanding the structural causes of planning failures. Instead, we introduce a quality prediction model grounded in interpretable machine learning techniques, such as SHAP-based feature attribution, which provides a causal proxy for the intermediate reasoning steps. This proxy enables the detection of planning deficiencies at a granular level. Subsequently, we employ counterfactual reinforcement learning to generate alternative planning trajectories and optimize the decision-making policy under causal constraints. The framework addresses critical system-level concerns including architectural trade-offs between planning depth and computational cost, governance of model deployment, robustness to distributional shift, fairness across diverse input populations, and policy implications for accountable AI. We illustrate the approach through conceptual case studies involving multi-step reasoning tasks and tool-use scenarios. The findings suggest that integrating causal reasoning into LLM evaluation not only enhances planning quality but also fosters transparency and alignment with human values. This work provides a foundational methodology for building interpretable, robust, and ethically governed LLM planning systems.

Keywords

causal evaluation, planning strategies, large language models, interpretable quality prediction, counterfactual reinforcement learning, SHAP, system architecture, governance, fairness, robustness.

1. Introduction

The rapid advancement of large language models has positioned them as central components in a wide range of socio-technical systems, from automated customer service to scientific hypothesis generation. These models are increasingly expected to exhibit deliberate reasoning and planning capabilities, often demonstrated through chain-of-thought prompting, hierarchical task decomposition, and tool use. However, the internal mechanisms by which LLMs arrive at a plan are largely hidden, and traditional evaluation methods that measure only the final correctness of an answer fail to capture the quality, efficiency, or reliability of the planning process itself. This gap creates significant risks when LLMs are deployed in high-stakes domains such as healthcare, finance, and public governance, where a flawed but seemingly reasonable plan can lead to harmful outcomes.

A growing body of research has begun to explore causal inference as a lens for understanding LLM behavior [1]. Causal evaluation moves beyond correlation-based metrics by asking what would happen if a particular reasoning step were altered, or whether a given planning strategy genuinely causes better outcomes. At the same time, interpretability techniques have matured to the point where we can attribute model predictions to specific input features or intermediate representations, providing a window into the model's reasoning [2,3]. When combined with reinforcement learning, these tools offer a powerful means to not only evaluate but also improve planning strategies. Counterfactual reinforcement learning, in particular, allows us to learn policies that are robust to distributional changes and that align with causal relationships discovered in the data [4,5].

This paper synthesizes these three strands of work into a unified framework for causal evaluation of planning strategies in LLMs. We propose that an interpretable quality prediction model can serve as a causal proxy, mapping observable planning steps to a quality score. This proxy is then used within a counterfactual reinforcement learning loop to iteratively refine the planning policy. Rather than focusing on algorithmic details, we emphasize system-level implications: how such a framework affects architectural decisions, computational overhead, governance protocols, robustness to adversarial perturbations, fairness across demographic groups, and the broader policy landscape for responsible AI deployment. We also discuss the trade-offs inherent in designing causal evaluation systems, including the tension between model complexity and interpretability, and the challenges of maintaining causal validity in large-scale deployments.

The remainder of the paper is organized as follows. Section 2 reviews relevant literature on LLM planning, interpretable machine learning, and causal RL. Section 3 lays out the conceptual framework for causal evaluation. Section 4 details the role of interpretable quality prediction as a causal proxy. Section 5 explains how counterfactual reinforcement learning leverages this proxy for strategy improvement. Section 6 examines architectural and governance considerations. Section 7 addresses robustness, fairness, and policy implications. Section 8 presents illustrative case analyses. Section 9 offers forward-looking perspectives, and Section 10 concludes.

2. Background and Related Work

The ability of LLMs to generate coherent plans has been studied extensively, with early work focusing on few-shot chain-of-thought prompting [6]. Subsequent research introduced tree-of-thoughts and graph-of-thoughts approaches that explore multiple reasoning paths simultaneously [7]. These methods have shown that explicit planning structures improve performance on complex tasks, yet they remain brittle to prompt variations and often fail to recover from early errors. The notion of planning as a search process through a state space is

well-established in classical AI, but its application to LLMs introduces unique challenges because the model's internal representation of state is not directly accessible.

Interpretable machine learning has emerged as a critical tool for opening the black box. SHAP (SHapley Additive exPlanations) provides a game-theoretic framework for feature attribution, assigning each input feature a contribution to the model's output [2,3]. In the context of LLMs, SHAP has been applied to explain why a model selects a particular next token or produces a specific chain-of-thought step. However, most applications focus on explanation rather than evaluation. A growing trend uses SHAP to predict the quality of model outputs, as demonstrated by recent work on API response quality prediction [8]. That approach uses a least-squares support vector machine combined with SHAP to identify which input features most influence output quality, offering a blueprint for our proposed quality prediction module.

Reinforcement learning has been used to fine-tune LLMs for reasoning tasks, often with a reward signal based on final correctness [9]. However, reward sparsity and delayed feedback make it difficult to learn effective planning strategies. Counterfactual reinforcement learning addresses this by considering not just observed outcomes but also hypothetical scenarios that could have occurred under different actions [4,10]. This is particularly relevant for planning, where a model may generate multiple plausible steps but only one leads to success. By reasoning about what would have happened if a different step were chosen, the model can learn more robust policies. Recent advances in high-level planning guidance reinforcement learning have integrated explicit plan decomposition into the RL loop, showing improved reasoning performance [11].

Causal inference provides the mathematical language for reasoning about interventions. Pearl's do-calculus and the potential outcomes framework underpin modern causal RL [5]. In LLMs, causal effects are often estimated by intervening on intermediate representations or input prompts. A key challenge is that LLMs are not causal models themselves; they learn statistical correlations from text, which may not reflect true causal relationships. Therefore, any causal evaluation must be accompanied by careful assumptions and validation.

Our work builds on these foundations by proposing a closed-loop system where an interpretable quality predictor (built using SHAP and a simple regressor) provides a causal proxy for planning quality. This proxy is then used as the reward signal in a counterfactual RL algorithm that updates the LLM's planning policy. The framework is designed to be modular, allowing replacement of the predictor or the RL algorithm as better methods emerge. Importantly, we frame the discussion at the system level, emphasizing the architectural and governance implications of deploying such a system in real-world infrastructures.

3. Conceptual Framework for Causal Evaluation of Planning Strategies

The central premise of this paper is that evaluating planning strategies in LLMs requires more than tracking final accuracy. A plan may be correct by chance, or it may be structurally sound but fail due to an unforeseen edge case. To capture the causal effect of planning, we need a counterfactual comparison: what would the outcome be if the model had chosen a different intermediate step? This is analogous to evaluating a human decision-maker by examining their reasoning process, not just the outcome.

We define a planning strategy as a mapping from a problem state to a sequence of actions or intermediate reasoning steps. In an LLM, this mapping is implemented by the model's parameters and the prompt structure. The quality of a plan can be decomposed into multiple dimensions: correctness, efficiency, coherence, and robustness. A causal evaluation

framework must be able to attribute variations in these dimensions to specific planning decisions. For example, if an LLM generates a flawed chain-of-thought because it incorrectly assumed a fact, the causal effect of that assumption on the final answer can be estimated by intervening on the assumption and observing the resulting answer.

Our framework consists of three components: a quality prediction module, a causal effect estimator, and a policy improvement module. The quality prediction module takes as input the plan (sequence of steps) and outputs a predicted quality score. This module is trained on a dataset of plans annotated with quality labels, and it uses SHAP to produce feature attributions that are interpretable. The causal effect estimator then uses these attributions as proxies for causal effects, under the assumption that the model captures relevant features. This assumption is strong but can be validated through controlled experiments. The policy improvement module employs counterfactual RL to generate new planning strategies that maximize the predicted quality, while also considering potential negative side effects.

A crucial aspect of this framework is its system-level design. The quality prediction module must be lightweight enough to run in real-time, especially in deployment scenarios where LLMs are serving many users. The counterfactual RL loop must be computationally feasible, perhaps operating offline or during periodic retraining. Furthermore, the entire pipeline must be auditable, so that stakeholders can understand which features drive quality predictions and how planning policies changed over time.

4. Interpretable Quality Prediction as a Causal Proxy

The quality prediction module serves as the bridge between observable planning behaviors and unobservable causal structures. We propose building this module using a simple yet interpretable model, such as a least-squares support vector machine or a sparse linear model, trained on features extracted from the LLM's planning trace. These features include the sequence of topics mentioned, the presence of certain keywords, the length of reasoning chains, the number of backtracking steps, and the confidence scores assigned to each step. SHAP values are then computed for each feature, providing a clear picture of which aspects of the plan most strongly influence predicted quality [2,3].

The use of SHAP is not merely for explanation; it also functions as a causal proxy because Shapley values satisfy a set of axioms that align with causal intuitions, including efficiency, symmetry, and dummy variable properties. Under the assumption that the feature set is causally sufficient (i.e., no unmeasured confounders), the Shapley value approximates the average marginal contribution of a feature to the prediction. In practice, this assumption rarely holds perfectly, but the proxy can still be useful for relative comparisons. For instance, if SHAP reveals that the length of the reasoning chain has a large negative contribution to quality, we can hypothesize that shorter chains cause better plans, and we can test this hypothesis through intervention experiments.

Recent work on API response quality prediction using least-squares SVM and SHAP demonstrates the feasibility of this approach in a similar context [8]. That study showed that SHAP not only identifies influential features but also provides insights into model behavior that can guide system improvements. Adapting this to planning strategies requires careful feature engineering: we must represent the hierarchical structure of a plan in a flat feature vector without losing critical information. One approach is to use a bag-of-steps representation, where each step is encoded by its semantic embedding and then aggregated. Another approach is to use a graph neural network to embed the planning graph, but that

would sacrifice interpretability. We advocate for a middle ground: a shallow representation that includes counts of step types, average step length, and a small set of prototype steps learned via clustering.

The quality prediction module is trained on a dataset collected from prior LLM deployments or from synthetic rollouts. Each planning instance is labeled with a quality score, which could be a composite of correctness, user satisfaction, and computational cost. The training process itself must be designed to avoid feedback loops: if the quality predictor is used to guide RL, it must be periodically revalidated to ensure it has not drifted. This is a classic challenge in online learning systems.

5. Counterfactual Reinforcement Learning for Strategy Improvement

With a causal proxy for planning quality in hand, we turn to improving the planning strategy via counterfactual RL. Traditional RL for LLMs uses the actual outcome (e.g., the final answer correctness) as the reward signal. This reward is sparse and often delayed, making it difficult to assign credit to individual planning steps. Counterfactual RL addresses this by considering not only the observed trajectory but also counterfactual trajectories that could have been sampled under a different policy [4,10].

In our framework, the RL agent learns a policy that maps a problem description to a planning strategy. At each step, the agent chooses an action (e.g., which subproblem to tackle next, which tool to call). The quality predictor then provides a dense, immediate reward estimate for the chosen action, based on the features of that action and the context. This reward is not the true outcome but a causal proxy that can be computed without waiting for the final answer. Moreover, counterfactual reasoning allows the agent to imagine what the reward would have been if it had chosen a different action, even if that action was not actually taken. This requires a model of the environment dynamics, which in the LLM case is the model's own response to prompts. We approximate this by using the LLM itself to generate hypothetical continuations given a changed action, a technique known as approximate counterfactual simulation.

The RL algorithm can be any that supports off-policy learning, such as Deep Q-Networks or policy gradient methods with counterfactual baselines. The key is that the reward signal is derived from the interpretable quality predictor, which itself may be a function of the same features used for SHAP. This creates a coherent loop: the quality predictor provides the reward, and the SHAP values from the predictor guide the interpretation of why certain actions are rewarded. Over time, the policy converges to a set of planning strategies that are not only high-quality according to the predictor but also robust to changes in the environment, because the counterfactual training encourages the agent to consider alternative scenarios.

A critical design consideration is the trade-off between exploration and exploitation. Because the quality predictor is a proxy, it may be biased, and the RL agent might exploit those biases to maximize predicted quality without actually improving true quality. To mitigate this, we recommend periodic validation against true quality measures and the incorporation of a small amount of random exploration. Additionally, the RL policy should be constrained by safety rules, especially in high-stakes domains. For example, a policy that learns to plan by skipping all factual verification steps to speed up inference could be harmful even if the quality predictor rewards speed.

6. Architectural and Governance Considerations

Deploying a causal evaluation framework for LLM planning requires careful architectural design. The system must support real-time quality prediction without adding prohibitive latency, and it must allow for periodic retraining of the quality predictor and the RL policy. One viable architecture is a microservice-oriented design where the LLM service, the quality predictor, and the RL policy engine are separate components that communicate via lightweight APIs. This separation allows independent scaling: the quality predictor can be deployed on CPU-only instances while the LLM service requires GPU clusters.

Governance becomes paramount when the system is used in production. The quality predictor's SHAP values should be logged for every planning interaction, creating an audit trail that regulators can inspect. The RL policy updates should be versioned and subject to human review before deployment. Moreover, the system should include monitoring for concept drift: when the distribution of input problems shifts, the quality predictor's accuracy may degrade, and the RL policy may become suboptimal. Automatic drift detection triggers a retraining pipeline that re-estimates the causal proxy using fresh data.

Another governance challenge is accountability. If a planning failure occurs, who or what is responsible? The LLM itself, the quality predictor, the RL policy, or the engineers who configured them? A causal evaluation framework explicitly links planning decisions to predicted outcomes, which can help attribute failures to specific features or steps. However, this attribution is only as reliable as the quality predictor's causal validity. To strengthen accountability, we recommend that all model updates be accompanied by a causal impact analysis report, showing how the change affected SHAP values for typical scenarios.

7. Robustness, Fairness, and Policy Implications

Robustness is a primary concern for any LLM-based system. Planning strategies learned via counterfactual RL may be brittle to adversarial inputs, such as prompts designed to exploit the quality predictor's blind spots. For instance, an adversary could craft a plan that receives a high predicted quality but yields a completely wrong answer. To guard against this, the quality predictor should be trained on adversarial examples and evaluated against a diverse set of stress tests. Moreover, the counterfactual RL loop can incorporate adversarial training by simulating adversarial perturbations to the planning steps.

Fairness is another critical dimension. The quality predictor might learn biased associations from the training data, for example, giving lower predicted quality to plans that use an uncommon reasoning style or that handle topics related to marginalized groups. SHAP values can help detect such biases by highlighting if certain demographic features are disproportionately influencing predictions. Once detected, fairness constraints can be added to the RL optimization, penalizing policies that lead to disparate quality predictions across groups.

Policy implications extend to national and international AI regulation. As governments consider frameworks for auditing high-risk AI systems, a causal evaluation methodology provides concrete evidence of a system's reasoning quality. Regulators can ask to see the SHAP values for a sample of planning decisions and verify that the RL policy was trained under appropriate constraints. This transparency could become a requirement for AI systems used in public services, healthcare, and criminal justice. Our framework offers a path toward such accountability, but it also raises questions about the resources needed to maintain it: smaller organizations may struggle to implement the necessary logging, retraining, and

auditing infrastructure. Policy makers should consider providing support for such capabilities, perhaps through standardized benchmarks and open-source reference implementations.

8. Illustrative Case Analyses

To ground the discussion, we consider two conceptual planning scenarios. The first involves a multi-step arithmetic reasoning task where an LLM must compute a compound interest formula. A naive planning strategy might attempt to compute the interest period by period, while a more efficient strategy would combine linear algebra. Our quality predictor, trained on a corpus of solution plans, might identify the presence of explicit intermediate recalculations as a feature that negatively correlates with quality. The counterfactual RL agent would then learn to prefer strategies that skip redundant steps. However, this preference could backfire on larger problems where intermediate verification is crucial for accuracy. The system must therefore balance efficiency with robustness.

The second scenario involves tool use: an LLM is asked to recommend a flight itinerary given constraints, and it must decide when to call an external API. A low-quality plan might call the API too early with insufficient context, leading to many wasted calls. A high-quality plan would first gather all constraints and then make a single, well-informed API call. Our quality predictor would capture the number of redundant API calls as a key negative feature. The counterfactual RL agent could then learn to defer API calls, but at the risk of omitting necessary information. By counterfactually simulating what would happen if an API call were made earlier, the agent can estimate the risk.

These examples illustrate the trade-offs inherent in any planning strategy and show how causal evaluation can surface them. The framework does not claim to produce optimal plans; rather, it provides a systematic way to explore the space of planning strategies with interpretable feedback.

9. Forward-Looking Perspectives

The integration of causal evaluation into LLM planning is still in its infancy. Several research directions promise to advance the field. One is the development of more sophisticated quality predictors that can handle long planning traces without losing interpretability. Hierarchical SHAP, which applies Shapley decomposition at multiple levels of abstraction, could be valuable. Another direction is the use of graph neural networks for the quality predictor, combined with GNN-explainer methods to maintain transparency.

On the RL side, model-based approaches that learn a causal dynamics model of the LLM's reasoning could enable more accurate counterfactual simulations. Instead of relying on the LLM itself to generate hypothetical continuations, a learned model would be faster and more controllable. This would also facilitate planning over longer horizons.

Finally, the ethical and societal implications of deploying such systems warrant deeper study. As LLMs become integral to critical infrastructure, the ability to causally evaluate their planning strategies becomes a matter of public safety. We call for interdisciplinary collaborations between computer scientists, social scientists, and policymakers to develop standards and best practices for causal evaluation in AI.

10. Conclusion

This paper has presented a comprehensive framework for causally evaluating planning strategies in large language models through the combined use of interpretable quality

prediction and counterfactual reinforcement learning. We have argued that traditional outcome-based metrics are insufficient and that a causal perspective is essential for understanding and improving the reasoning processes of LLMs. The proposed system architecture balances model transparency with computational efficiency, while addressing governance, robustness, fairness, and policy concerns through careful design choices. Although many technical challenges remain, the framework provides a solid foundation for building accountable and reliable LLM planning systems. As the deployment of large language models expands, the ability to causally evaluate and refine their internal decision-making processes will become increasingly important for ensuring alignment with human values and societal norms.

References

1. Pearl, J. (2009). Causality: Models, reasoning, and inference (2nd ed.). Cambridge University Press.
2. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems 30 (pp. 4765–4774).
3. Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., ... & Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. Nature Machine Intelligence, 2(1), 56–67.
4. Bottou, L., Peters, J., Quinonero-Candela, J., Charles, D. X., Chickering, D. M., Portugual, E., ... & Scholkopf, B. (2013). Counterfactual reasoning and learning systems: The example of computational advertising. Journal of Machine Learning Research, 14, 3207–3260.
5. Bareinboim, E., & Pearl, J. (2016). Causal inference and the data-fusion problem. Proceedings of the National Academy of Sciences, 113(27), 7345–7352.
6. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., ... & Le, Q. (2022). Chain-of-thought prompting elicits reasoning in large language models. In Advances in Neural Information Processing Systems 35 (pp. 24824–24837).
7. Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., & Narasimhan, K. (2023). Tree of thoughts: Deliberate problem solving with large language models. In Advances in Neural Information Processing Systems 36.
8. Gao, H., Zeng, W., Zhang, J., & Liang, Y. (2025, December). A large model API response quality prediction model based on least squares vector machine and SHAP interpretability analysis. In 2025 5th International Symposium on Artificial Intelligence and Big Data (AIBDF) (pp. 438-442). IEEE.
9. Stiennon, N., Ouyang, L., Wu, J., Shen, T., Zhuang, J., Schuurmans, D., ... & Christiano, P. (2020). Learning to summarize from human feedback. In Advances in Neural Information Processing Systems 33 (pp. 3008–3021).
10. Oberst, M., & Shalit, U. (2019). Action-context models for off-policy evaluation and learning. In Advances in Neural Information Processing Systems 32.
11. Dou, Z., Zhao, Q., Wan, Z., Zhang, D., Wang, W., Raiyan, T., ... & Biswas, S. (2025). Plan Then Action: High-Level Planning Guidance Reinforcement Learning for LLM Reasoning. arXiv preprint arXiv:2510.01833.

12. Shui, Y., Jin, R., Dou, Z., & Gao, Z. (2026). ProtoGuard-SL: Prototype Consistency Based Backdoor Defense for Vertical Split Learning. arXiv preprint arXiv:2604.03595.

13. Zhou, D. (2025, December). M-VP2: Microservice-Oriented Vulnerability Patch Planning-A Cost-Aware Approachusing Multi-Agent Reinforcement Learning. In 2025 5th International Conference on Computer, Internet of Things and Control Engineering (CITCE) (pp. 248-254). IEEE.

14. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems 30 (pp. 5998–6008).

15. Bengio, Y., Léonard, N., & Courville, A. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432.

16. Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., ... & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In Proceedings of the 33rd International Conference on Machine Learning (pp. 1928–1937).

17. Dulac-Arnold, G., Mankowitz, D., & Hester, T. (2019). Challenges of real-world reinforcement learning. arXiv preprint arXiv:1904.12901.

18. Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., & De Freitas, N. (2016). Dueling network architectures for deep reinforcement learning. In Proceedings of the 33rd International Conference on Machine Learning (pp. 1995–2003).

19. Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.

20. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135–1144).

21. Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608.

22. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete problems in AI safety. arXiv preprint arXiv:1606.06565.

23. Raji, I. D., Smart, A., White, R. N., Mitchell, M., Gebru, T., Hutchinson, B., ... & Barnes, E. (2020). Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (pp. 33–44).

24. Selbst, A. D., Boyd, D., Friedler, S. A., Venkatasubramanian, S., & Vertesi, J. (2019). Fairness and abstraction in sociotechnical systems. In Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency (pp. 59–68).

25. Zhang, J., & Bareinboim, E. (2022). Bounding causal effects on continuous outcomes. In Proceedings of the AAAI Conference on Artificial Intelligence 36 (pp. 10277–10285).