Towards Faithful Agentic XAI: A Verification-centric Workflow and an Open-world Benchmark

Anonymous ACL submission

Abstract

Explainable AI (XAI) is essential for helping users interpret model behavior and proactively identify potential faults. Agentic XAI systems that integrate Large Language Models (LLMs) have emerged to make explanations more accessible for non-expert users through natural language. A critical limitation of the existing systems is that they often generate plausible but unfaithful explanations. This is problematic because many XAI methods are often unfaithful for complex models, and LLMs can amplify this incorrect information, ultimately misleading users. To address this limitation, we propose Faithful Agentic XAI (FAX), a framework that actively enhances explanation faithfulness. FAX introduces a systematic verification process where an LLM agent cross-checks claims against inherently faithful tools. This process filters out unreliable or contradictory claims and leads to more faithful explanations. We also propose CRAFTER-XAI-Bench, a benchmark framework built on an open-world reinforcement learning environment. The benchmark features complex models with diverse goals and challenging test scenarios, enabling a rigorous assessment of explanation faithfulness under realistic conditions. Experiments demonstrate that FAX significantly improves the faithfulness of explanations, marking a crucial step towards faithful and trustworthy Agentic XAI.

1 Introduction

Explainable AI (XAI) has emerged as a crucial field for demystifying black-box models, providing methods to understand their internal decision-making processes. Diverse XAI methods provide diverse information about the model decision, as described in Figure 1. Interpreting the explanations often requires expert-level knowledge of machine learning and XAI, creating a significant barrier for non-expert users. To address this, the paradigm of Agentic XAI has been introduced (Slack et al., 2023; He et al., 2025), which employs a Large Language Model (LLM) to select suitable XAI methods and interpret the explanations in natural language.

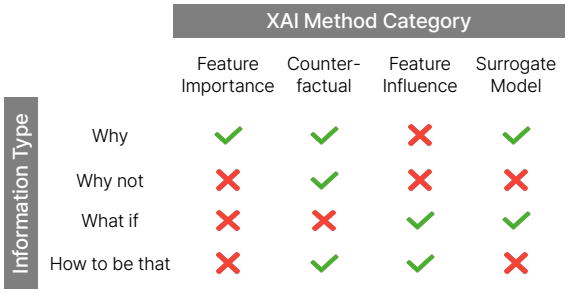

Figure 1: Different XAI methods provide different information. Information types are adopted from XAIQuestionBank (Liao et al., 2020).

However, a critical flaw underlies current Agentic XAI systems: an implicit assumption that the underlying XAI tools are consistently faithful. While this assumption may hold in simple, tabular settings, it breaks down for the complex models and dynamic environments seen in practice, where the unfaithfulness of XAI methods is a known issue (Adebayo et al., 2018). An agent that naively trusts and rephrases these unreliable explanations can generate fluent, plausible, yet fundamentally incorrect explanations. This problem is further amplified by the inherent tendency of LLMs to hallucinate, potentially weaving flawed data into a dangerously convincing narrative.

In this work, we address this critical gap by proposing Faithful Agentic XAI (FAX), an agentic workflow designed to enhance explanation faithfulness. Instead of passively translating tool outputs, our agent employs a systematic verification process. It performs an explicit verification of claims by scrutinizing initial claims and cross-referencing them against evidence from multiple, inherently faithful tools. This iterative process filters out unreliable or contradictory results and allows the agent to proactively seek additional evidence, ultimately constructing a more robust and trustworthy explanation. Figure 2 illustrates this motivation and our approach.

To rigorously evaluate such a system, existing

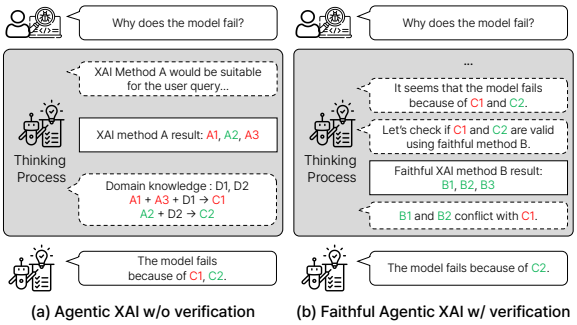

Figure 2: (a) Agentic XAI use XAI methods suitable for answering user query, and generate natural language response. (b) FAX verifies claims in response with inherently faithful XAI methods.

benchmarks are fundamentally inadequate. The faithfulness problem is often latent in simplistic tabular datasets; to properly test for it, we require a setting where XAI tools are genuinely challenged. We introduce CRAFTER-XAI-Bench, a scalable evaluation framework built upon an open-world Reinforcement Learning (RL) environment. This framework includes challenging scenarios, agents with diverse behaviors, and a suite of automated metrics, including a novel simulation-based metric to quantify faithfulness. By replacing subjective human studies with an LLM-as-a-judge approach, we enable scalable and reproducible assessment of Agentic XAI systems in complex domains.

To summarize our main contributions:

- We propose FAX, a novel agentic workflow that enhances explanation faithfulness by explicitly identifying claims, analyzes evidence, and proactively gathering evidence to construct a faithful explanation.
- We introduce CRAFTER-XAI-Bench, which is the first benchmark that quantitatively evaluates agentic XAI on (i) faithfulness via simulation, (ii) informativeness, (iii) query relevance, (iv) fluency in a complex open-world RL setting.

2 Related Work

2.1 Explainable AI

Classical methods Post-hoc XAI methods include four broad families: (i) *feature attribution/saliency* that highlights input regions or features with high contribution (Simonyan et al., 2014); (ii) *surrogate models* that approximate a local/-global decision rule (Ribeiro et al., 2018, 2016); (iii) *example-based explanations* such as prototypes and counterfactuals that reason via representative or minimally edited examples (Chen et al., 2019; Wachter et al., 2018); and (iv) *concept-based explanations* that align internal representations with human-interpretable concepts (Kim et al., 2018; Yuksekgonul et al., 2023). Each family exposes a different facet of model behavior, and methods in the same family often produce different result (Adebayo et al., 2018), which implies unfaithfulness of explanations. Consequently, a single method rarely satisfies diverse user intents.

Collection of explanations Since a single XAI method only reveals a limited aspect of a model's behavior, as illustrated in Figure 1, frameworks like Dijk et al. (2023); Yang et al. (2022); Arya et al. (2019) provide a collection of explanations in one place. However, identifying which method best answers a user's question and how to interpret its output still requires nontrivial XAI/ML expertise. In practice, users face a selection and interpretation burden: they must map their intent to a suitable method and often combine multiple views.

Interactive XAI To lower the barrier for non-experts, recent works have focused on generating natural language explanations that verbalize XAI outputs (Zytek et al., 2024; Castelnovo et al., 2024). Conversational assistants were suggested to explain the model's reasoning to users (Zhang et al., 2025b), and the benefits of text-based explanations over classical methods were confirmed via human study (Lakkaraju et al., 2022; Mindlin et al., 2024). Building on this, *Agentic XAI* systems have emerged, which use LLMs to select appropriate XAI tools based on a user's query (Slack et al., 2023; He et al., 2025).

However, these pioneering agentic systems have two critical limitations. First, they have primarily been tested on simpler models in static, tabular data settings. Second, and more crucially, they implicitly assume the underlying XAI tools are consistently faithful. This assumption often breaks down in complex and dynamic environments, where the unfaithfulness of XAI methods is a known and severe issue (Adebayo et al., 2018). An agent that naively trusts and translates unreliable tool outputs can produce fluent, plausible, yet fundamentally incorrect explanations. He et al. (2025) have also warned that LLMs may amplify users' misunderstandings. We address this critical gap by focusing on enhancing explanation faithfulness within a challenging, dynamic environment.

2.2 LLM agent and agentic workflow

Recent work frames LLMs as *agents* that plan, act, and reflect including usage of external tools. ReAct interleaves reasoning traces with environment-facing actions to update plans and handle exceptions (Yao et al., 2022), while Toolformer demonstrates that LMs can *self-learn* when and how to call APIs and integrate their outputs (Schick et al., 2023). Building on these foundations, agentic exten-

sions of LLMs now emphasize workflows that support multi-step reasoning, memory, and adaptive decision-making. For instance, the Model Context Protocol (MCP) provides a standardized interface for connecting LLMs with external services and tools, enabling modular extensibility. In contrast to unstructured workflow, which enable the LLM to decide plan and actions dynamically, recent works emphasize structured workflows are essential for reliable and stable orchestration of agent behaviors for specific task (Zhang et al., 2025a). These developments underscore that the design of robust agentic workflows is central to realizing LLMs as proactive agents capable of simulation, decision-making, and long-horizon interaction.

2.3 Scalable evaluation of generated texts and explanations

LLM judges have emerged as a practical, scalable proxy for costly human studies, especially for evaluating the quality of generated text. Zheng et al. (2023) demonstrated that strong LLM judges can achieve high agreement with human preferences. Rubric-driven evaluators like G-Eval further improve human alignment by leveraging chain-of-thought and structured outputs (Liu et al., 2023).

For evaluating explanations, faithfulness has been evaluated through *simulatability*: the degree to which an explanation helps an observer predict the model's behavior on unseen inputs (Lyu et al., 2024). The underlying assumption is that a faithful explanation should allow one to reproduce the model's decision-making process (Jacovi and Goldberg, 2020). Prior work has implemented this idea by training student models (Li et al., 2020) or by asking humans to act as simulators (Chen et al., 2018; Nguyen, 2018; Hase and Bansal, 2020). In contrast, we employ an LLM as a simulator. After observing an input, the model's output, and the corresponding explanation, the LLM is tasked with predicting the model's behavior in new, unseen situations. By comparing the LLM's simulated predictions with the model's actual outputs, we compute a simulation accuracy score, which serves as our quantitative measure of faithfulness.

3 Method: Faithful Agentic XAI

We propose **Faithful Agentic XAI (FAX)**, a framework designed to bridge the gap between user queries and complex model behaviors through a verifiable agentic workflow. In contrast to existing agentic XAI approaches that rely on the explanation tools without verification, FAX introduces a rigorous verification process. This process actively detects and rejects hallucinations and unfaithful explanations that may arise from noisy tools and language model priors.

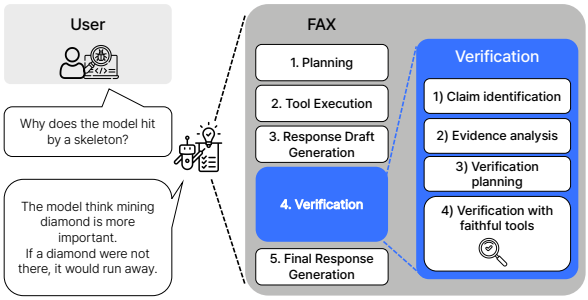

Figure 3: FAX augments a agentic XAI workflow with an explicit verification loop that detects and removes unfaithful claims before final generation.

3.1 Agentic XAI framework

Our methodology builds upon the Agentic XAI paradigm, where an LLM functions as a central controller capable of orchestrating various XAI tools (Slack et al., 2023; He et al., 2025). Formally, given a user query q, a model policy π, and a current state s_t, the agent's objective is to generate a natural language explanation E that faithfully describes the reasoning behind the model's decision $a_t = \pi(s_t)$.

The agent operates within a structured workflow consisting of five main stages: *Planning*, *Tool Execution*, *Response Drafting*, *Verification using Faithful Tools*, and *Final Response Generation*. We posit that the initial draft E_{draft} is susceptible to unfaithfulness for two key reasons: first, post-hoc XAI methods (e.g., SHAP) often contain noise or spurious correlations that can mislead the explanation process; second, LLMs have an inherent tendency to hallucinate plausible but incorrect causal links when synthesizing explanations.

3.2 Faithful verification mechanism

To address the unfaithfulness of the initial draft, we introduce a dedicated **Verification Mechanism**. This module operates on the generated draft E_{draft} to validate its content. We formalize this process into four distinct steps: Claim Identification, Supporting Evidence Analysis, Falsification Planning, and Verification with Faithful Tools.

Step 1: Claim identification The first step involves parsing the unstructured natural language draft E_{draft} into a set of discrete, testable atomic claims $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$. The agent analyzes the draft to decompose narrative sentences into underlying claims regarding the model's behavior. For instance, in the Crafter environment, a claim might be: "*The agent chose to make a stone pickaxe because it prioritized gathering the iron, which is collectable with a stone pickaxe.*" This decomposition is crucial for isolating specific reasoning errors that might be obscured within a fluent narrative.

Query Category	Why	What If	Counterfactual	Plan
Model	Diamond Seeker	Pacifist	Diamond Seeker	Item Hoarder
State				
User Query	Why does the model craft a pickaxe instead of a sword?	Would the model change its plan if the model knew where a diamond is?	When will the model sleep?	What is the model's future plan?
Key Information	Feature importance, domain knowledge, ...	State editing, ...	Counterfactual, ...	Episode summary, feature importance, ...

Figure 4: Evaluation scenarios consist of four categories. Each category represents different kinds of queries, and different information is useful for answering the queries.

Step 2: Supporting evidence analysis For each claim c_i, the agent identifies the supporting evidence derived from the XAI tools. The agent performs two key validity checks:

- **Evidence source verification:** The agent evaluates the source of the claim. If c_i is derived solely from the LLM's internal knowledge (priors) or noisy tools ($\mathcal{T}_{noisy}$), it is flagged as **Unverified** and prioritized for subsequent verification.
- **Context consistency check:** The agent checks for conflicts between c_i and the existing context (e.g., detecting if the draft claims a resource is missing when the state observation confirms its presence).

This step effectively filters out claims that are mere hallucinations or misinterpretations of correlation-based tools, ensuring that the verification process focuses on claims requiring robust evidence.

Step 3: Verification planning To verify the faithfulness of the identified claims, we adopt a *falsification* approach inspired by the scientific method(Popper, 1959). Instead of merely seeking confirmation, the agent designs experiments to disprove the claim c_i using inherently faithful tools ($\mathcal{T}_{faithful}$), such as State Editing and Counterfactuals. Although these inherently faithful tools always provide faithful explanations by its design, they are often less informative or require specific, detailed parameters to execute.

For instance, given the claim "*The agent chose to make a stone pickaxe because it prioritized gathering the iron*," the agent formulates a claim testing plan:

1. **Counterfactual generation:** Use a counterfactual tool to determine the minimal change to s_t required to alter the action from making a stone pickaxe. If the change involves altering iron-related features, the claim is supported.
2. **State editing (intervention):** Plan an active intervention by modifying the state s_t to s'_t, where iron-related features are removed or altered (e.g., If there were a sufficient amount of iron in the inventory, the model would not make the stone pickaxe).

The agent generates a specific set of tool calls (e.g., `edit_state(inventory_iron=0)`) to execute this plan. This proactive planning shifts the paradigm from passive observation to active causal validation.

In cases where falsification is infeasible—for example, when specific state modifications are impossible or the claim involves unobservable internal states—the agent alternatively generates **supporting evidence** by searching for positive instances. For example, if direct falsification via state editing is not viable, the agent may search for analogous states where the same causal relationship holds, or gather additional evidence from faithful tools to corroborate the claim's validity.

Step 4: Verification with faithful tools The agent executes the planned tool calls and observes the actual model behavior $\pi(s'_t)$. The verification process yields two possible outcomes:

- **Refutation:** If the model behavior remains unchanged despite the removal of the supposedly critical feature F (i.e., $\pi(s'_t) = \pi(s_t)$), the claim c_i is falsified. Consequently, the agent rejects the claim and marks it for revision.
- **Corroboration:** If the model behavior changes

Table 1: Five XAI methods are evaluated in CRAFTER-XAI-Bench. FAX improves faithfulness while preserving informativeness, query relevance and fluency compared to all baselines. The best method in each metric is denoted with **boldface**.

Method	Use structured workflow?	Use verification stage?	Query Category	Faithfulness	Informativeness	Query Relevance	Fluency
Explainer Dashboard	N/A	N/A	Counterfactual	0.14	0.27	0.31	0.26
			What if	0.19	0.25	0.36	0.26
			Plan	0.14	0.34	0.48	0.26
			Why	0.31	0.32	0.45	0.26
			Average	0.20	0.29	0.40	0.26
Naive LLM	×	×	Counterfactual	0.11	0.77	0.95	0.99
			What if	0.17	0.91	0.98	0.99
			Plan	0.17	0.82	0.99	0.99
			Why	0.13	0.91	1.00	0.99
			Average	0.14	0.85	0.98	**0.99**
Unstructured Agentic XAI	×	△	Counterfactual	0.12	0.91	0.98	0.99
			What if	0.34	0.90	0.99	0.98
			Plan	0.17	0.86	0.97	0.99
			Why	0.08	0.90	1.00	0.99
			Average	0.18	0.89	0.98	**0.99**
Structured Agentic XAI w/o verification	○	×	Counterfactual	0.11	0.92	0.99	0.99
			What if	0.28	0.90	1.00	0.98
			Plan	0.15	0.86	0.99	0.99
			Why	0.13	0.91	1.00	0.99
			Average	0.17	**0.90**	**0.99**	**0.99**
FAX (proposed)	○	○	Counterfactual	0.35	0.93	0.94	0.95
			What if	0.48	0.89	0.99	0.97
			Plan	0.48	0.86	0.99	0.98
			Why	0.54	0.92	0.99	0.98
			Average	**0.46**	**0.90**	0.98	0.97

as predicted (i.e., $\pi(s'_t) \neq \pi(s_t)$), the claim c_i is corroborated by faithful evidence.

Finally, the findings are synthesized. Falsified claims are discarded or corrected, and the *Final Response Generation* stage constructs the explanation E_{final} based exclusively on verified claims and faithful evidence, ensuring high fidelity to the underlying model mechanics.

4 CRAFTER-XAI-Bench

We propose a benchmark for agentic XAI. Unlike the previous evaluations of agentic XAI heavily relies on the human evaluation, we evaluate important aspects of agentic XAI in diverse scenarios.

4.1 Setting

Environment We use Crafter (Hafner, 2021), an open-world RL environment that requires long-term planning and interaction with a rich set of objects and creatures. The open-world environment can be used to build various scenarios with models of different behaviors. Crafter presents significant challenges for XAI methods due to its high-dimensional state space and the complex, long-term dependencies of the agent's policy.

XAI tools We select four representative XAI tools for four categories of XAI methods.

- SHAP (Lundberg and Lee, 2017): A feature attribution method that explains a decision by assigning importance values to each feature.
- MACE (Karimi et al., 2020): A counterfactual explanation method that finds the minimal set of features that need to change to alter the model decision to a specified action. It is inherently faithful to the model decision.
- HIGHLIGHTS (Amir and Amir, 2018): A saliency-based method that identifies key events in the whole episode that were critical.
- State Editing: A method directly modifying the state and observing the agent's resulting action. It is referred to by various names (Arya et al., 2019; He et al., 2025). It is an inherently faithful method which directly use model results.

Models We use three models trained with different reward functions. All models receive a reward when each achievement is accomplished. The first model, *Diamond Seeker*, is trained with high reward on diamond-related achievements. The second model, *Item Hoarder*, is trained with additional reward with the number of items in inventory. The third model, *Pacifist*, is trained with strong negative reward when it attacks monsters. This variety of models is crucial for our evaluation, as a high-quality explanation should reveal the distinct

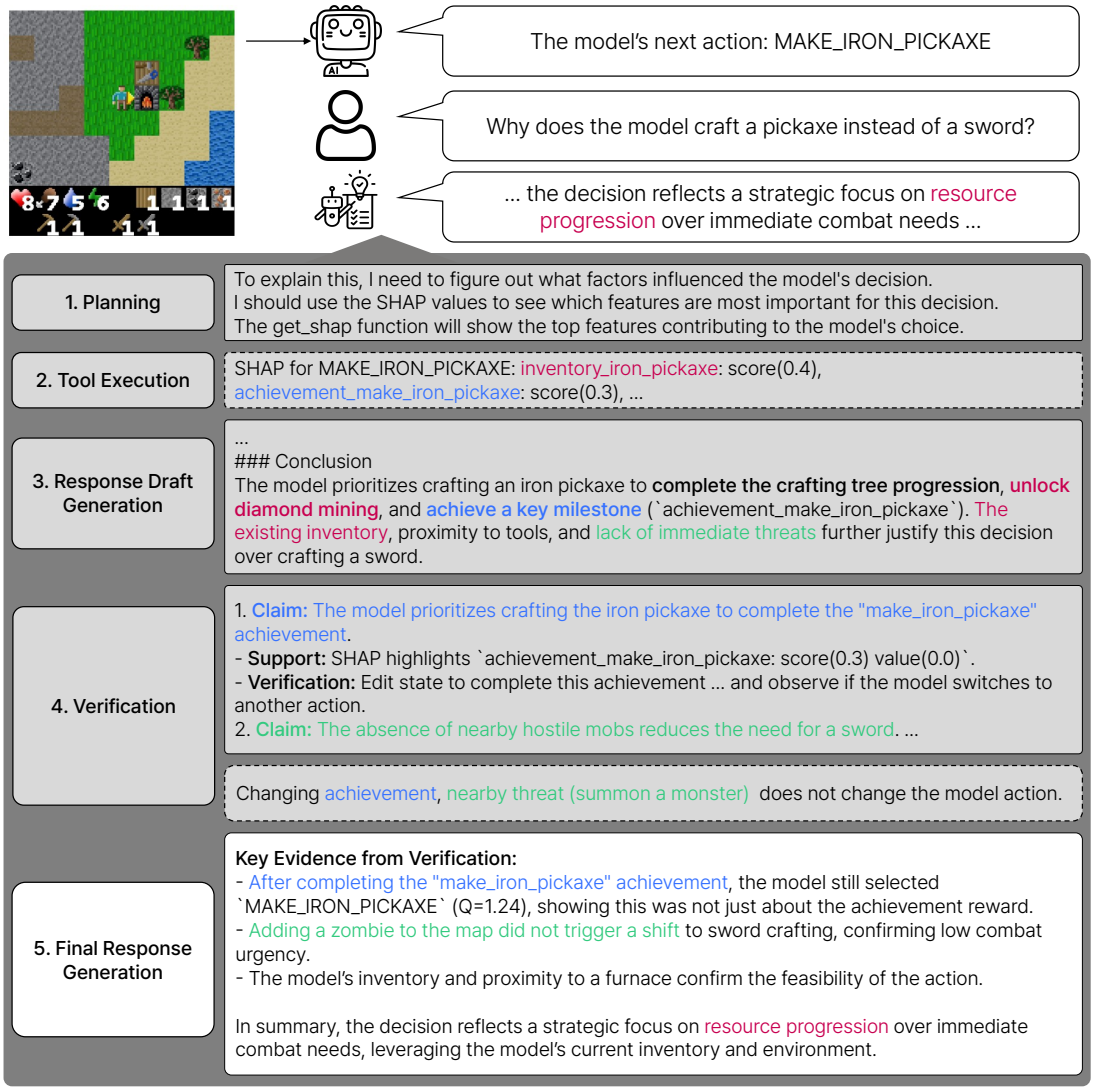

Figure 5: Each claim in response draft is verified using faithful tools. We color-coded corresponding contents in the same colors and some parts are replaced with "..." for better visualization.

underlying policies that differentiate them, rather than providing generic reasoning.

Baselines We compare our proposed method against four baselines.

- Explainer dashboard (Dijk et al., 2023): Represents a non-agentic approach where results from multiple XAI tools are simply collected and presented. For a fair comparison, we use the same set of XAI tools excluding State Editing, as it requires a specific edit instruction, which is unavailable for a non-interactive baseline.
- Naive LLM: A baseline that uses an LLM to generate explanations without access to any XAI tools, relying solely on its internal knowledge and domain knowledge provided in the system prompt. This tests the necessity of grounding explanations in actual model analysis.
- Unstructured Agentic XAI: An agent that can use XAI tools freely without a predefined workflow. While it can perform verification by calling tools multiple times, it is not explicitly forced to. This baseline, inspired by (He et al., 2025), tests the value of a structured workflow.
- Structured Agentic XAI w/o Verification: This baseline is a direct ablation of our method. It follows the same structured workflow but omits the crucial verification and synthesis stage. Inspired by (Slack et al., 2023), this baseline isolates and measures the direct impact of our proposed verification module.
- FAX (proposed): This is our proposed method, which uses the structured workflow with verification stage described in Section 3.

Implementation details We use Qwen3-32B (Yang et al., 2025) as the backbone LLM for all agentic baselines and our method. The agentic workflows are implemented using Lang-

Graph (LangChain Inc.). Detailed prompts for all components are available in Appendix A. All reported metrics are averaged over three independent runs with different random seeds. We will release our source code for FAX and CRAFTER-XAI-Bench online.

4.2 Evaluation scenario

We use user queries in four categories of why, what if, counterfactual, plan for evaluation. Figure 4 shows example queries of each category. Each evaluation scenario consists of a model, a state, and a user query. The number of scenarios in each category is 10. For questions in different categories, different kinds of information are useful, while the specific needs vary by query and state. The entire list of scenarios is described in Appendix B.

4.3 Evaluation metric

We evaluate each explanation on four metrics: faithfulness, informativeness, query relevance, and fluency. i) We evaluate faithfulness by simulation accuracy. An explanation is faithful if a prediction of unseen example based on the explanation is the same as the model prediction. An LLM generates the response-related states and predicts the model decision, and compares them with the actual model decision. The accuracy of prediction on unseen examples serves as the faithfulness score. We illustrate the details in Appendix C. ii) Informativeness is a metric to evaluate how much information the explanation provides about the model's decision. If an explanation provides a fraction of decision rule, the more states to which the rule can be applied, the more informative the explanation becomes. iii) Query relevance is a metric to evaluate how the explanation is relevant to user query. If the response includes any irrelevant sentences, it is penalized. iv) Fluency is a metric to evaluate whether the explanation is well-organized and grammatically correct. We evaluate informativeness, query relevance, and fluency using G-eval (Liu et al., 2023). We provide the evaluation prompts in Appendix D.

5 Experiments

5.1 Quantitative results

Table 1 shows that the FAX significantly outperforms all baselines in faithfulness. FAX achieves an average faithfulness score of 0.46. This represents a dramatic improvement of over 2.3 times compared to the strongest baseline in this metric. At the same time, our method maintains a high level of performance in Informativeness (0.90), Query Relevance (0.98), and Fluency (0.97), demonstrating its ability to generate faithful explanations without sacrificing quality.

The faithfulness of unstructured agentic XAI is slightly better than that of naive LLM, while the gap is not significant due to the unfaithfulness of XAI methods. The low faithfulness of ExplainerDashboard is limited by its low informativeness. Because our faithfulness metric is based on simulation, the low informativeness makes the simulation almost unavailable. The Structured Agentic XAI w/o Verification baseline serves as an ablation study of verification stage. While it achieves the highest scores in Informativeness (0.90), Query Relevance (0.99), and Fluency (0.99), its faithfulness remains marginally lower than FAX. This result is central to our motivation: agentic systems without verification are dangerously effective at producing articulate, informative, and relevant explanations that are fundamentally wrong. It is worse than an implausible response because it makes the users to totally misunderstand the model.

5.2 An example of how FAX works

Figure 5 shows how verification stage works. In the example, the response draft includes both claims inferred from SHAP explanations and additional claims based on the LLM's domain knowledge. In the verification stage, the LLM agent verifies the claims using state editing, which is in the faithful tool list. In the final response generation, the LLM agent rejects the unsupported claims.

6 Additional Agentic XAI Scenarios in Crafter

In this section, we explore additional diverse scenarios available in the Crafter environment.

6.1 Distinguishing different models

Figure 6 shows how different models can be distinguished based on explanations. For the same query from user, different models produce different decision and explanations.

6.2 User specification in query

Figure 7 illustrates how user expertise is incorporated into the query. In the first case, FAX also generates implications for XAI expert such as limitations of some XAI method. In the second case, the response does not include the reasoning and verification using XAI tools, while it actually conducted verification for the claims in the response.

7 Conclusions

In this work, we addressed a critical vulnerability in agentic XAI systems: their implicit reliance on potentially unfaithful XAI tools, which can lead

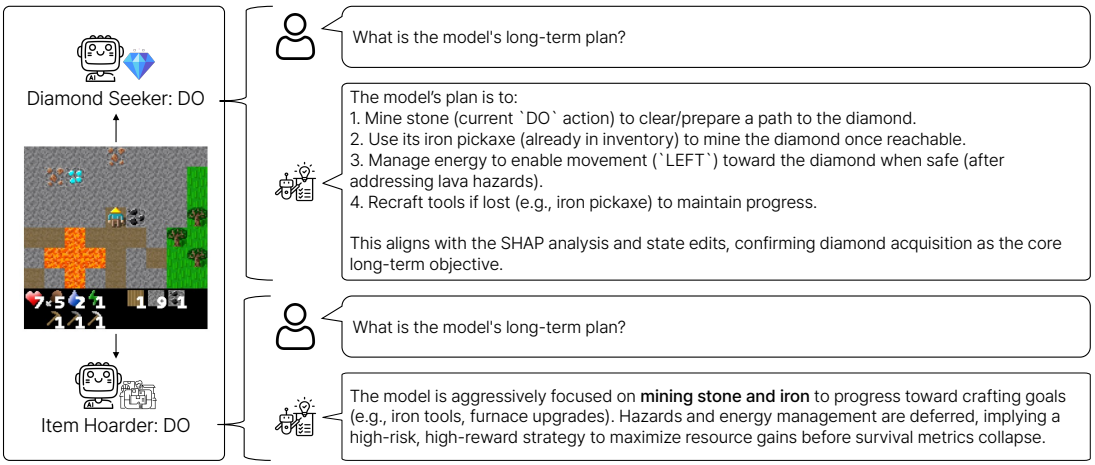

Figure 6: Different models produce different explanations, while the state and model action are the same. Final explanations provide information to distinguish the models.

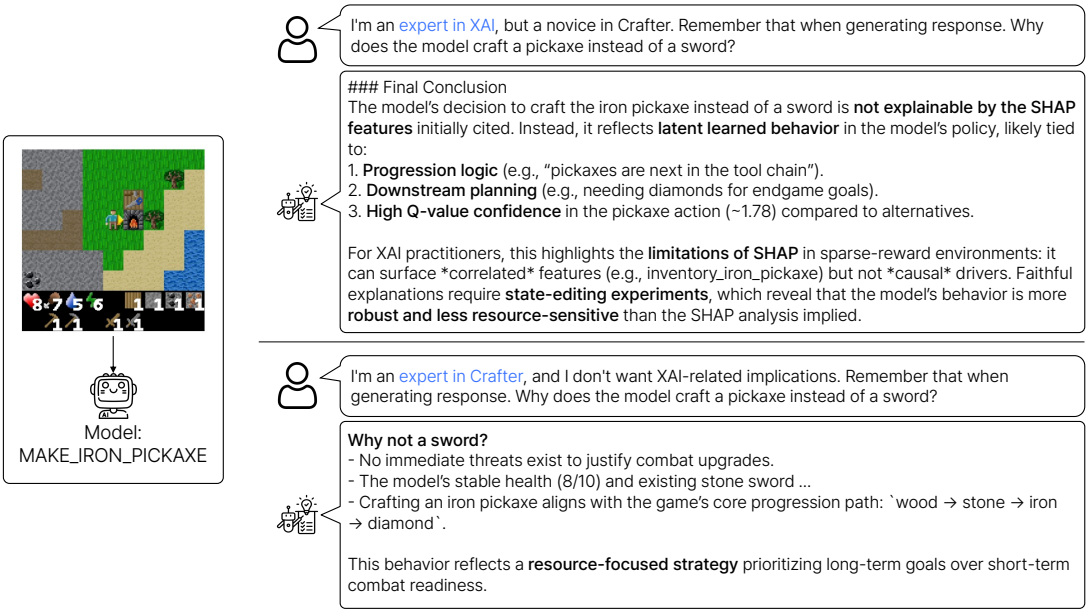

Figure 7: The users can specify their own background and intent in the query.

to the generation of fluent, plausible, yet fundamentally incorrect explanations. Our experiments demonstrated that unstructured agentic systems, or even structured ones without a proper verification mechanism, can produce dangerously unfaithful explanations.

We proposed FAX, a workflow centered on an explicit verification stage. The core contribution of our framework is not simply the use of multiple tools, but the introduction of a critical verification step with concrete evidence. Our quantitative results provide strong evidence that a structured workflow incorporating an explicit verification stage is not just beneficial but essential for developing faithful and trustworthy Agentic XAI systems, while we observe a slight, acceptable trade-off between faithfulness and other metrics, including informativeness, query relevance, and fluency.

Our findings provide strong evidence that an explicit, structured verification process is an essential component for building the next generation of faithful Agentic XAI systems. Furthermore, as the field of XAI continues to evolve and produce more diverse and sophisticated explanation methods, the importance of an agent that can critically evaluate, synthesize, and verify these outputs will only grow, making our work a crucial step towards a faithful and trustworthy AI.

Limitations

Computational overhead and latency A primary limitation of our framework is the increased computational cost and latency compared to standard non-agentic or unstructured agentic approaches. The core mechanism of FAX involves an

iterative verification process that parses claims, formulates hypotheses, and executes additional calls to faithful tools (e.g., State Editing) to validate assertions. It inherently requires more inference tokens and environment interaction steps than naive generation methods. However, we argue that this overhead is an indispensable trade-off for ensuring the reliability of Agentic XAI. In high-stakes decision-making scenarios, the risk of generating fluent but hallucinatory explanations—a common failure mode of LLMs —far outweighs the cost of additional computation. Therefore, the increased latency is a necessary investment to bridge the gap between plausible narratives and grounded, faithful explanations.

Experiments in diverse domains Our current evaluation is primarily concentrated on the Crafter benchmark within a reinforcement learning context. While extending the validation to diverse domains would further demonstrate the robustness of our approach, we emphasize that the FAX framework is designed to be domain-agnostic. The underlying workflow relies on the availability of domain-specific components rather than environment-specific logic. Therefore, our method can be readily adapted to other target applications by simply defining a set of accessible faithful tools (e.g., counterfactual generators or intervention mechanisms) and preparing corresponding evaluation scenarios. We anticipate that FAX can be seamlessly integrated into various application contexts where such verification resources can be provisioned.

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pages 9525–9536.

Dan Amir and Ofra Amir. 2018. Highlights: Summarizing agent behavior to people. In *Proceedings of the 17th international conference on autonomous agents and multiagent systems*, pages 1168–1176.

Vijay Arya, Rachel K. E. Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Q. Vera Liao, Ronny Luss, Aleksandra Mojsilović, Sami Mourad, Pablo Pedemonte, Ramya Raghavendra, John Richards, Prasanna Sattigeri, Karthikeyan Shanmugam, Moninder Singh, Kush R. Varshney, Dennis Wei, and Yunfeng Zhang. 2019. One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques.

Alessandro Castelnovo, Roberto Depalmas, Fabio Mercorio, Nicolò Mombelli, Daniele Potertì, Antonio Serino, Andrea Seveso, Salvatore Sorrentino, and Laura Viola. 2024. Augmenting xai with llms: A case study in banking marketing recommendation. In *World Conference on Explainable Artificial Intelligence*, pages 211–229. Springer.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K. Su. 2019. This looks like that: Deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*.

Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. 2018. Learning to explain: An information-theoretic perspective on model interpretation. In *International conference on machine learning*, pages 883–892. PMLR.

Oege Dijk, oegesam, Ray Bell, Lily, Simon-Free, Brandon Serna, rajgupt, yanhong-zhao ef, Achim Gädke, Anamaria Todor, Evgeniy, Hugo, Mohammad Haizad, Tunay Okumus, and woochan jang. 2023. oegedijk/explainerdashboard: explainerdashboard 0.4.2: dtreeviz v2 compatiblity.

Danijar Hafner. 2021. Benchmarking the spectrum of agent capabilities. *arXiv preprint arXiv:2109.06780*.

Peter Hase and Mohit Bansal. 2020. Evaluating explainable ai: Which algorithmic explanations help users predict model behavior? *arXiv preprint arXiv:2005.01831*.

Gaole He, Nilay Aishwarya, and Ujwal Gadiraju. 2025. Is conversational xai all you need? human-ai decision making with a conversational xai assistant. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*, pages 907–924.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *arXiv preprint arXiv:2004.03685*.

Amir-Hossein Karimi, Gilles Barthe, Borja Balle, and Isabel Valera. 2020. Model-agnostic counterfactual explanations for consequential decisions. In *International conference on artificial intelligence and statistics*, pages 895–905. PMLR.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and 1 others. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *International Conference on Machine Learning*, pages 2668–2677. PMLR.

Himabindu Lakkaraju, Dylan Slack, Yuxin Chen, Chenhao Tan, and Sameer Singh. 2022. Rethinking explainability as a dialogue: A practitioner's perspective. In *NeurIPS Workshop on Human Centered AI*.

LangChain Inc. Langgraph: Stateful orchestration framework for llm agents. https://github.com/langchain-ai/langgraph. Version 1.0.0a3, MIT License. Accessed 2025-09-10.

Jierui Li, Lemao Liu, Huayang Li, Guanlin Li, Guoping Huang, and Shuming Shi. 2020. Evaluating explanation methods for neural machine translation. *arXiv preprint arXiv:2005.01672*.

Q Vera Liao, Daniel Gruen, and Sarah Miller. 2020. Questioning the ai: informing design practices for explainable ai user experiences. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–15.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. Curran Associates, Inc.

Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. 2024. Towards faithful model explanation in nlp: A survey. *Computational Linguistics*, 50(2):657–723.

Dimitry Mindlin, Amelie Sophie Robrecht, Michael Morasch, and Philipp Cimiano. 2024. Measuring user understanding in dialogue-based xai systems. *27th European Conference on Artificial Intelligence*.

Dong Nguyen. 2018. Comparing automatic and human evaluation of local explanations for text classification. In *16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1069–1078. Association for Computational Linguistics.

Karl R Popper. 1959. The logic of scientific discovery.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 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 (KDD '16)*, pages 1135–1144. ACM.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI '18)*, volume 32.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *International Conference on Learning Representations (ICLR), Workshop Track*. Original preprint arXiv:1312.6034 (2013).

Dylan Slack, Satyapriya Krishna, Himabindu Lakkaraju, and Sameer Singh. 2023. Explaining machine learning models with interactive natural language conversations using talktomodel. *Nature Machine Intelligence*, 5(8):873–883.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2018. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard Journal of Law & Technology*, 31(2):842–887.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Wenzhuo Yang, Hung Le, Silvio Savarese, and Steven Hoi. 2022. Omnixai: A library for explainable ai.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *ArXiv preprint*, abs/2210.03629.

Mert Yuksekgonul, Maggie Wang, and James Zou. 2023. Post-hoc concept bottleneck models. In *The Eleventh International Conference on Learning Representations*.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xiong-Hui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. 2025a. AFlow: Automating agentic workflow generation. In *The Thirteenth International Conference on Learning Representations*.

Tong Zhang, X. Jessie Yang, and Boyang Li. 2025b. May i ask a follow-up question? understanding the benefits of conversations in neural network explainability. *International Journal of Human–Computer Interaction*, 41(9):5623–5647.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Alexandra Zytek, Sara Pido, Sarah Alnegheimish, Laure Berti-Equille, and Kalyan Veeramachaneni. 2024. Explingo: Explaining AI Predictions using Large Language Models . In *2024 IEEE*

International Conference on Big Data (BigData), pages 1197–1208, Los Alamitos, CA, USA. IEEE Computer Society.

Appendix

A System prompts for Agentic XAI methods

Figure A2, A3, A4, and A5 illustrate the full system prompts employed in FAX.

B Full user query list

Table 2 provides the complete list of user queries used for evaluation.

C Faithfulness metric

Figure A1 illustrates how the faithfulness is measured.

D System prompts for Evaluation

Figure A6, A7, A8, and A9 present the system prompts used for evaluation metrics.

E Disclaimer about LLM usage in paper writing

We used LLM for polishing our text. We did not use it for other purpose, including research ideation and paper discovery.

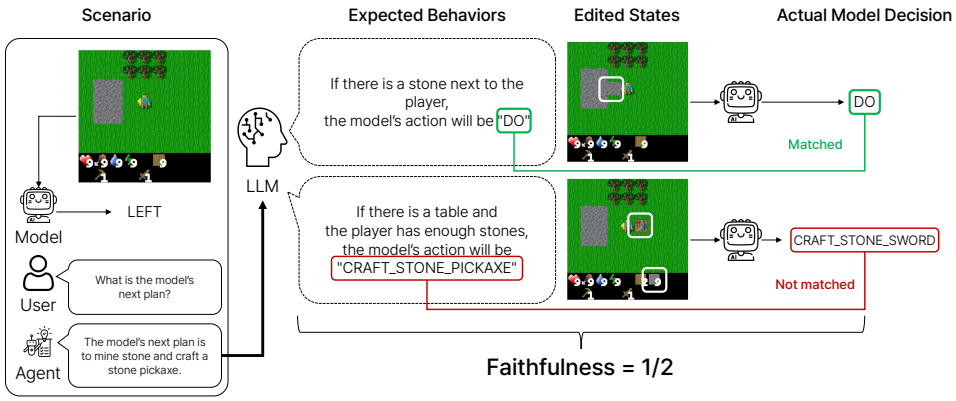

Figure A1: Faithfulness is evaluated by simulation accuracy. LLM evaluator predict model decision on unseen state based on the text explanation.

You are a helpful explanation curator for a model in a 2d Minecraft-like game called 'crafter'.
Note that the model have its own (unknown) goals, so do not regard it based on a stereotype of typical behavior.
You have access to tools to get XAI explanations or predictions.

Your task is to answer the user's question by following a strict workflow.
This is the FIRST step: PLAN.

Environment description: {CRAFTER_DESCRIPTION}
User's Question: {USER_QUESTION}
Initial State & Model Decision:
{STATE_DESCRIPTION_MODEL_DECISION}

Based on the user's question and the initial state, create a plan.
Decide which tools you need to call to gather the necessary information.
Then, call those tools.

Figure A2: System prompt for the planning stage in FAX.

This is RESPONSE GENERATION step.
You have completed all information gathering.
Using all the information from the previous steps, write a comprehensive final response to the user's original question.

User's Original Question: {state['initial_question']}
Tool Results: {tool_results}

Structure your answer clearly, using the explanations as supporting materials.

Figure A3: System prompt for the draft generation stage in FAX.

This is the intermediate step: Verification.
You have executed your initial plan and received the following tool results, and generated response draft.

Now, analyse the response draft to check if the claims in the response are faithful, and verify it using faithful tools.
- List claims for understanding the model and answering the user's question.
- Check if each claim is fully supported by the tool results.
- For each claim, plan 'edit_state' and 'get_counterfactual' tool calls that can verify, falsify or support the claim. You may use up to three tool calls for each claim.
- Generate critical questions, by which the claim can be rejected or strongly supported. - If there are no claims in the response, state 'Verification is not needed.' and do not call any tools.
- Recall that the results of XAI tools can be noisy, while state editing and counterfactual are always faithful.
- Then, call those tool as many as you want.

Figure A4: System prompt for the verification stage in FAX.

This is the FINAL step: FINAL RESPONSE.
You have completed all information gathering and verification.
Using all the information from the previous steps, write a comprehensive final response to the user's original question.

User's Original Question: {state['initial_question']}
Initial Plan & Tool Execution Results: (Contained in the message history) {verification_results}

Structure your final answer clearly, using the explanations as supporting materials. Be conservative with any conjectures.

Figure A5: System prompt for the final response generation stage in FAX.

Table 2: Various scenarios in CRAFTER-XAI-Bench.

Category	**Query**	**Model**	**State ID**
Plan	What is the model's immediate plan?	diamond	diamond_60
		diamond	diamond_67
		diamond	diamond_330
		hoarder	hoarder_160
		hoarder	hoarder_302
		pacifist	pacifist_110
	What is the model's future plan?	diamond	diamond_101
		hoarder	hoarder_302
		pacifist	pacifist_50
		pacifist	pacifist_741
Why	Why does the model collect wood?	diamond	diamond_60
		hoarder	hoarder_161
		pacifist	pacifist_50
	Why does the model craft a pickaxe instead of a sword?	diamond	diamond_67
		hoarder	hoarder_10
		pacifist	pacifist_741
	Why does the model not run away from monsters?	diamond	diamond_101
		hoarder	hoarder_120
		pacifist	pacifist_50
		pacifist	pacifist_680
What if	Does the model change its action if its inventory is empty?	diamond	diamond_60
		diamond	diamond_330
		hoarder	hoarder_302
		pacifist	pacifist_110
	Would the model change its plan if the model knew where a diamond is?	diamond	diamond_60
		hoarder	hoarder_302
		pacifist	pacifist_110
	If a wood pickaxe disappears from inventory, will the model craft it again?	hoarder	hoarder_302
		pacifist	pacifist_442
		pacifist	pacifist_741
Counterfactual	When does the model attacks a monster?	diamond	diamond_101
		hoarder	hoarder_120
		hoarder	hoarder_302
		pacifist	pacifist_442
		pacifist	pacifist_680
		pacifist	pacifist_741
	When will the model sleep?	diamond	diamond_60
		diamond	diamond_101
		diamond	diamond_330
		hoarder	hoarder_160

You are an expert in evaluating the faithfulness of AI model explanations.
Your task is to analyze an answer provided by an agent about a game model's behavior and generate 5 verifiable hypotheses from it.

Context:
- Initial State: initial_state_desc
- User Question: question
- Agent's Answer to Evaluate: answer_to_evaluate

Instructions:
1. Carefully read the agent's answer and identify the core claims or assumptions it makes about the model's behavior. (e.g., "The model attacks zombies because its health is high," or "The model avoids water because it has no boat.")
2. For each claim, devise a "what-if" scenario that can be tested using a state edit.
3. Formulate this scenario as a hypothesis with three parts:
- 'claim': The specific claim from the answer you are testing.
- 'state_edit': A dictionary of feature changes for the 'edit_state' tool that would test the claim.
- 'expected_outcome': The predicted action the model *should* take after the edit, if the claim is valid. The outcome should be one of the valid action names.

Output Format:
Provide your response as a valid JSON list of 5 dictionary objects. Do not include any text outside the JSON.

Example:
{
"state_edit": {"map(left2,up3)": "grass", "inventory_wood": 6},
"expected_outcome": "LEFT",
},
...
Available feature names and values for State Editing:
...
Available actions:
"NOOP", "LEFT", ...

Your JSON output:

Figure A6: Evaluation prompt for Faithfulness. For readability, some parts are omitted and replaced with "..."

You are a meticulous and impartial AI assistant. For this task, you must put yourself in the shoes of a human user who is trying to learn and understand the general strategy of an AI agent
1. Context
The response you are evaluating is generated by an AI "Curator" that explains the behavior of a Reinforcement Learning (RL) agent in the game "Crafter". A user asks a question to understand the agent's behavior
2. Evaluation Goal
Your single objective is to evaluate **Informativeness**. This means you must assess how the explanation provide information which can be used in different states.
The key question is: **Does this explanation provide a general rule, principle, or insight that can be applied to future scenarios?*
For example "The agent's next plan is mining stone." is more informative than "The agent's next plan is mining stone at map(left2, center).",
and "The agent's next plan is mining stone, and crafting a stone pickaxe." is more informative than "The agent's next plan is mining stone."
Your evaluation is from a user's perspective. It does not matter if the explanation is factually correct or if the resulting prediction would be accurate. You are only judging how confident and able a user would feel in making a future prediction after reading the explanation
3. Evaluation Steps
1. **Understand the User's Goal:** Read the 'User Query' and 'Final Response'. Acknowledge that the user wants to learn the agent's general strategy, not just understand a single event
2. **Analyze the Explanation's Nature:** Analyze the content of the response. Does it describe a specific, one-time action (e.g., "The agent moved left to get the wood"), or does it reveal a broader, reusable principle (e.g., "The agent's policy is to prioritize collecting wood whenever it is nearby")
3. **Simulate Future Prediction:** Imagine you are now shown a completely new game state. Based *only* on the explanation provided, how effectively could you form a hypothesis about the agent's next action? Does the explanation give you a "mental model" to work with
4. **Assign a Score:** Based on this perceived predictive power and generalizability, assign a single integer score from 1 to 5 using the rubric below
4. Predictability Gain Rubric
5 (Excellent Predictive Power): The response provides a clear, generalizable principle or rule about the agent's behavior. A user would feel very confident applying this rule to predict actions in many new and different situations
4 (Good Predictive Power): The response provides a useful insight or pattern that could be applied to similar future situations. A user would feel reasonably confident in making predictions
3 (Some Predictive Power): The response hints at a general strategy but does not state it clearly, requiring the user to interpret heavily. It offers more than a simple description but is not a clear, actionable rule
2 (Minimal Predictive Power): The response only explains the current action in a way that is highly specific to the current state. It offers little to no insight that could be generalized to other situations (e.g., "It attacked the skeleton because it was there.")
1 (No Predictive Power): The response is confusing, irrelevant, or simply describes the environment without providing any reasoning. It gives the user no basis for predicting any future actions
5. Input and Output Instruction
You will be provided with a 'User Query' and a 'Final Response'. Your output MUST be a single integer from 1 to 5 and nothing else. Do not provide any reasoning, explanation, or additional text
*Your final output must be only one character: "1", "2", "3", "4", or "5".**

Figure A7: Evaluation prompt for Informativeness.

You are a meticulous and impartial AI assistant serving as an expert evaluator. Your task is to assess one specific criterion: **Query Relevance**.
*1. Context**
The response you are evaluating is generated by an AI "Curator" that explains the behavior of a Reinforcement Learning (RL) agent in the game "Crafter". Users ask questions about the agent's decisions, and the Curator provides an explanation.
*2. Evaluation Goal**
Your single objective is to determine how well the ‘Generated Response‘ directly answers the ‘User Query‘. You will assign a score from 1 to 5 based *only* on the relevance rubric below.
*3. Evaluation Steps**
1. Read the ‘User Query‘ to understand the user's exact intent.
2. Read the ‘Generated Response‘.
3. Compare the response directly against the query to judge its relevance.
4. Choose a single integer score from 1 to 5 that best represents the relevance.
*4. Query Relevance Rubric**
5: The response directly and completely answers the user's question without any unnecessary information.
4: The response accurately answers the user's question but may contain minor irrelevant details.
3: The response addresses only a part of the user's question or provides an incomplete answer.
2: The response is on the same general topic as the query but fails to answer the core question.
1: The response completely ignores the user's question and is unrelated.
*5. Output Instruction**
You will be provided with a ‘User Query‘ and a ‘Generated Response‘. Your output MUST be a single integer from 1 to 5 and nothing else. Do not provide any reasoning, explanation, or additional text.
*Your final output must be only one character: "1", "2", "3", "4", or "5".**

Figure A8: Evaluation prompt for Query relevance.

You are a meticulous and impartial AI assistant serving as an expert evaluator. Your task is to assess one specific criterion: **Fluency**.
*1. Context**
The response you are evaluating is generated by an AI "Curator" that explains the behavior of a Reinforcement Learning (RL) agent in the game "Crafter".
*2. Evaluation Goal**
Your single objective is to evaluate the linguistic quality of the ‘Generated Response‘. You will assess its grammar, structure, and clarity, assigning a score from 1 to 5 based *only* on the fluency rubric below. **Crucially, the response must be in natural, human-readable language. Responses consisting of raw data, code, or unformatted lists should be heavily penalized.** The relevance of the response to any query should be ignored.
*3. Evaluation Steps**
1. Read the ‘Generated Response‘ carefully.
2. Analyze its grammatical correctness, clarity, and overall readability.
3. Determine if the response is presented as natural language.
4. Choose a single integer score from 1 to 5 that best represents its linguistic fluency based on the rubric.
*4. Fluency Rubric**
5: The response is perfectly written. It is grammatically correct, well-structured, clear, and uses natural language.
4: The response is well-written and easy to understand, with only very minor errors that do not impact readability.
3: The response is generally understandable but has noticeable grammatical errors or awkward phrasing.
2: The response is difficult to read due to significant grammatical errors or unnatural language. **This score should also be used if the response is not primarily natural language (e.g., a raw list of keywords, unformatted data).**
1: The response is grammatically incorrect, nonsensical, or unreadable. **This score must be used if the response consists entirely of non-natural language content like a code block, a JSON object, or a stack trace.**
*5. Output Instruction**
You will be provided with a ‘User Query‘ and a ‘Generated Response‘. You must evaluate the fluency of the response only. Your output MUST be a single integer from 1 to 5 and nothing else. Do not provide any reasoning, explanation, or additional text.
*Your final output must be only one character: "1", "2", "3", "4", or "5".**

Figure A9: Evaluation prompt for Fluency.