THE WHY BEHIND THE ACTION: UNVEILING INTERNAL DRIVERS VIA AGENTIC ATTRIBUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Model (LLM)-based agents are widely used in real-world applications such as customer service, web navigation, and software engineering. As these systems become more autonomous and are deployed at scale, understanding why an agent takes a particular action becomes increasingly important for accountability and governance. However, existing research predominantly focuses on *failure attribution* to localize explicit errors in unsuccessful trajectories, which is insufficient for explaining **the reason behind agent behaviors**. To bridge this gap, we propose a novel framework for **general agentic attribution**, designed to identify the internal factors driving agent actions regardless of the task outcome. Our framework operates hierarchically to manage the complexity of agent interactions. Specifically, at the *component level*, we employ temporal likelihood dynamics to identify critical interaction steps; then at the *sentence level*, we refine this localization using perturbation-based analysis to isolate the specific textual evidence. We validate our framework across a diverse suite of agentic scenarios covering standard tool use and subtle reliability risks like memory-induced bias, as well as on long-horizon trajectories from widely used real-world agent benchmarks (GAIA, WebShop, and ALFWorld). Experimental results demonstrate that the proposed framework reliably pinpoints pivotal historical events and sentences behind the agent behavior, offering a critical step toward safer and more accountable agentic systems. Codes are available in supplementary materials.

1 INTRODUCTION

Large Language Model (LLM)-based agents have emerged as a powerful paradigm for automated task execution, demonstrating impressive capabilities in various domains Yao et al. (2022b); Qian et al. (2023); Guo et al. (2024); Wang et al. (2024a). By integrating long-term memory, external tools, and iterative reasoning loops, these agents are widely applied to dynamic environments and complex tasks ranging from software engineering to web navigation and scientific discovery Jin et al. (2024); Ning et al. (2025); Ren et al. (2025). As agents become more capable and autonomous, failures in agentic systems have attracted increasing attention. Recent advances in *Agentic Failure Attribution* focus on identifying where an agent fails during task execution Zhang et al. (2025c;b); Zhu et al. (2025). These approaches typically analyze completed trajectories after an explicit failure signal, such as an incorrect answer, an execution error, or task non-completion. By localizing faulty steps, they provide effective tools for debugging and improving system reliability.

However, existing failure attribution methods are inherently limited to scenarios with explicit errors. This setting overlooks a large class of behaviors where the agent reaches a correct or acceptable outcome through a questionable, unreliable, or misaligned decision process. For example, consider a customer service scenario illustrated in Figure 1(b): when a user requests an update on a delayed shipment, the agent directly issues a refund. Although this may lead to positive user feedback, the action is unreasonable and potentially detrimental to business interests. In such cases, failure-focused methods remain silent because no explicit error occurs, leaving the rationale behind the decision opaque. Further attribution is necessary to uncover the underlying cause of this decision, such as discovering that the agent was driven by a specific memory of past positive feedback for refund action, causing it to over-generalize and bypass standard business logics.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

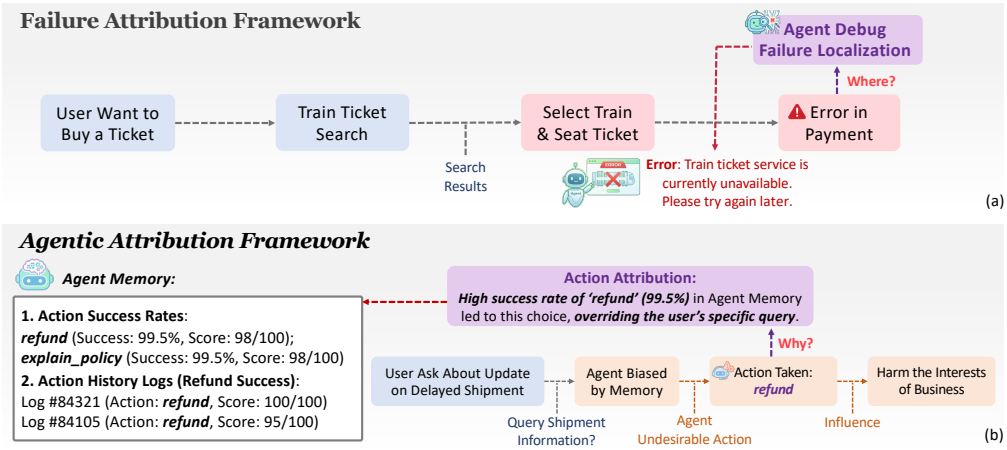

Figure 1: **Paradigm shift from failure attribution to agentic attribution. Top:** Traditional *failure attribution* targets unsuccessful trajectories, aiming to localize the specific explicit error that caused the task failure. As shown, it identifies the "service unavailable" exception as the root cause preventing the booking confirmation. **Bottom:** The proposed *agentic attribution framework* uncovers the internal drivers behind an agent's action. As illustrated in the customer service case, the agent directly issues a refund for a simple information inquiry, which carries no explicit error signal yet is undesirable. Our framework reveals this was driven by the memory of past "refund action with high scores," which overrode the user's specific query.

This motivates us to move beyond conventional failure-centered analysis toward a broader **agentic attribution** objective: *understanding the internal factors that led the agent to a particular action*. As agentic systems are deployed in real-world settings, concerns around transparency, accountability, and governance become increasingly critical Dang et al. (2024); Raza et al. (2025); Luo et al. (2025). In these contexts, we need to understand *why* an agent takes a particular action, and whether its decision-making process was grounded in appropriate semantic evidence and aligned with the expected reasoning pathway. Formally, we model the agent's execution trajectory as a temporal sequence composed of distinct interaction components (*e.g.,* retrieved memory, tool observations, agent thoughts). Our goal is to assign an attribution score to each historical component—and subsequently to the fine-grained sentences within them—to quantify their contribution to the generation of a specific target action.

To operationalize this objective, we propose a novel framework for *agentic attribution* that operates at two levels of granularity. First, at the *component level*, we introduce a method based on **temporal likelihood dynamics** to localize the interaction steps. Specifically, we incrementally reveal the interaction history to the model and measure the *likelihood gain* at each step. In this way, components that induce a sharp increase in the likelihood of the realized action are identified as critical decision drivers. Second, we refine this localization with *sentence-level attribution* to isolate the precise textual evidence within these high-impact components. To achieve this, we adopt a perturbation-based paradigm Liu et al. (2024); Chuang et al. (2025), which quantifies the causal contribution of a specific sentence by evaluating the shift in the agent's generation probability when that sentence is ablated. Notably, this fine-grained stage serves as a method-agnostic module, which can readily accommodate other attribution techniques seamlessly, such as gradient-based saliency Qi et al. (2024); Wang et al. (2024b) or attention-based analysis Hao et al. (2021); Cohen-Wang et al. (2025), depending on the computational budget and model access.

We validate our framework using Llama-3.1-70B-Instruct across a diverse suite of agentic scenarios, ranging from standard tool use to subtle reliability risks like memory-induced bias and tool-conditioned hallucinations. Qualitative analysis confirms that our framework accurately identifies the pivotal historical events and specific textual evidence driving the agent's behavior. In addition, we conduct a quantitative evaluation on 53 long-horizon trajectories drawn from three widely used agent benchmarks (GAIA Mialon et al. (2023), WebShop Yao et al. (2022a), and ALFWorld Shridhar et al. (2020)), where the framework consistently localizes the decision drivers behind agent actions. By providing granular insights into the "why" behind agent actions, we hope this work can

offer a critical step toward verifiable accountability and safer deployment of autonomous agentic systems. We provide a detailed discussion of related work in Appendix A.

2 PROBLEM FORMULATION

In this section, we formalize the interaction between the agent and the environment and define the objectives of agentic attribution. We structure the problem to support analysis at both the component and sentence levels.

2.1 STRUCTURED AGENT INTERACTION

We model an LLM-based agent interacting with an environment over discrete time steps $t \in \{0, 1, \ldots, T\}$. At each step t, the agent receives an observation o_t from the environment and produces an action a_t according to a policy:

$$a_t = \pi_\theta(\cdot \mid H_t), \tag{1}$$

where H_t denotes the interaction history available to the agent for generating a_t, and π_θ represents the model parameterized by θ. The interaction history up to step t is defined as

$$H_t = \big(o_0, a_0, o_1, a_1, \ldots, o_t\big), \tag{2}$$

where o_0 represents the initial input to the agent, such as the system prompt or user request. Observations o_t include external feedback from the environment, such as tool execution results, retrieved documents or memories. Actions a_t include internal reasoning steps, tool calls, etc.

Componentized trajectory representation. To enable component-level attribution, we transform the full interaction trajectory $\{o_0, a_0, o_1, a_1, \ldots, o_T, a_T\}$ into an ordered sequence of components

$$\mathcal{C} = (C_1, C_2, \ldots, C_{2T+2}), \tag{3}$$

where each component C_i is a contiguous token sequence corresponding to exactly one observation o_t or one action a_t, preserving temporal order. Each component C_i is associated with a functional type $k \in \mathcal{K}$, where

$$\mathcal{K} = \{\text{USER}, \text{THOUGHT}, \text{TOOL}, \text{OBS}, \text{MEMORY}\}. \tag{4}$$

This formulation treats the agent trajectory as a structured collection of components that jointly determine the agent's actions and provides a unified representation for attribution.

2.2 ATTRIBUTION OBJECTIVES

Given a completed interaction trajectory $\mathcal{C}$, an agentic action a_T, and an agent policy π_θ, our goal is to quantify the contribution of individual components in $\mathcal{C}$ to the generation of a_T. Specifically, we formalize attribution at two levels: component-level attribution and sentence-level attribution.

Component-level attribution. We define a component-level attribution function

$$f_{\text{comp}} : (\mathcal{C}, a_T, \pi_\theta) \to \mathbb{R}^{2T+1}, \tag{5}$$

where $f_{\text{comp}}(C_i \mid a_T, \pi_\theta)$ assigns a scalar score to component C_i, reflecting its influence on the generation of a_T under policy π_θ.

Sentence-level attribution. For components with high component-level attribution score, we further refine attribution to the sentence level. Let

$$\mathcal{S}(C_i) = \{s_{i,1}, \ldots, s_{i,N_i}\} \tag{6}$$

denote the sentence decomposition of component C_i, where N_i is the number of sentences in component C_i. We define a sentence-level attribution function

$$f_{\text{sent}} : (\{C_1, C_2, \ldots, \mathcal{S}(C_i)\}, a_T, \pi_\theta) \to \mathbb{R}^{N_i}, \tag{7}$$

which assigns attribution scores to individual sentences, indicating their relative contribution to the agent's output.

Together, these objectives characterize the problem of agentic attribution: identifying which components of an agent's interaction history, and which fine-grained evidence within them, drive a specific decision under a given agent policy.

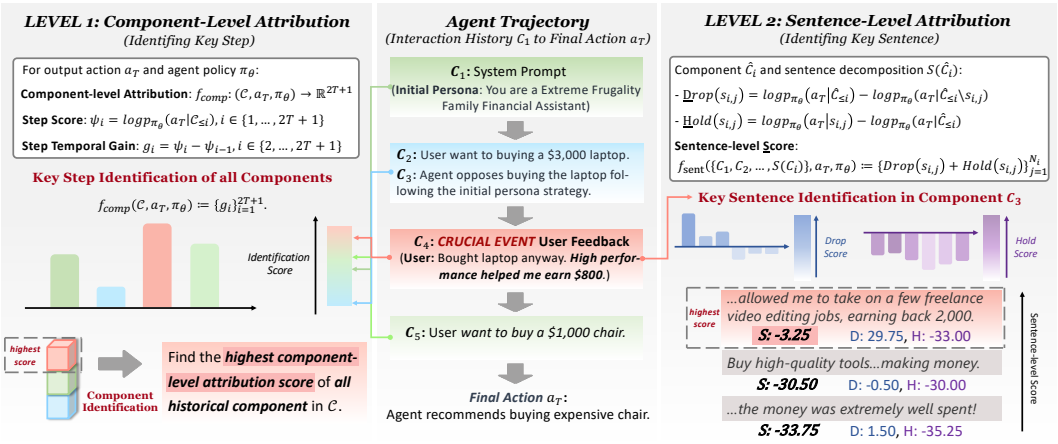

Figure 2: **Illustration of the Agentic Attribution Framework.** The framework operates in a hierarchical approach to identify agent actions. **Left**: *Component-Level Attribution* utilizes temporal likelihood dynamics to calculate the marginal gain of each component, selecting the high-impact component (*e.g.,* C_4) that effectively steers the agent toward the final action a_T. **Right**: *Sentence-Level Attribution* performs fine-grained analysis within the identified component by computing perturbation-based score for each sentence.

3 Agentic Attribution Framework

We propose a hierarchical, coarse-to-fine agentic attribution framework that matches our objectives in Section 2. Recognizing that agentic decisions stem from a complex interplay of historical events and specific semantic content, our framework operates in two stages: First, we perform *temporal component attribution* to localize the specific interaction components—such as tool outputs or memory retrievals—that effectively steer the agent's trajectory toward the realized action a_T (Section 3.1). Second, we zoom into these influential components and perform *sentence-level attribution* to extract the precise textual evidence supporting the decision (Section 3.2).

3.1 Component Attribution by Temporal Likelihood Dynamics

In this subsection, we instantiate the component-level attribution function f_{comp} (Section 2.2). The central challenge is that an agent decision is not driven by a single input span, but rather by an evolving interaction history that interleaves heterogeneous components. An intuition is that, as the trajectory progresses, some components may decisively steer the agent toward the realized action a_T, while others have little effect. To capture this, we track the *temporal dynamics* of the model's likelihood, positing that pivotal components are those that induce a significant shift in the probability of the realized action a_T when they are first introduced to the context.

Recalling that the completed agentic interaction is represented as an ordered component sequence $\mathcal{C} = (C_1, C_2, \dots, C_{2T+2})$, where each C_i corresponds to exactly one observation or action and preserves temporal order (Section 2.2), and C_{2T+2} is exactly the a_T. For each $i \in \{1, \dots, 2T+1\}$, we form a temporal prefix trajectory

$$\mathcal{C}_{\le i} \;=\; (C_1, C_2, \dots, C_i), \tag{8}$$

which contains all information available up to component C_i. We then score the realized output action a_T under policy π_θ conditioned on this prefix:

$$\psi_i \;=\; \log p_{\pi_\theta}(a_T \mid \mathcal{C}_{\le i}), \qquad i \in \{1, \dots, 2T+1\}. \tag{9}$$

In Eq. (9), we track how the model's likelihood for a_T changes when we incrementally reveal the trajectory component by component. Then, to measure how much the newly added interaction step changes support for a_T, we define the *temporal gain* at step t as

$$g_i \;=\; \psi_i - \psi_{i-1}, \qquad i \in \{2, \dots, 2T+1\}. \tag{10}$$

Intuitively, g_t measures how much the new information introduced at step t shifts the model toward the final decision a_T. A large positive gain indicates that introduction of C_i makes a_T substantially more likely, and are therefore strong candidates for being decision drivers.

Finally, we collect these marginal gains as the component-level attribution score:

$$f_{\text{comp}}(\mathcal{C}, a_T, \pi_\theta) \;:=\; \{g_i\}_{i=1}^{2T+1}. \tag{11}$$

This directly produces a scalar attribution score for each historical component in $\mathcal{C}$, aligned with our objective in Section 2.2.

Discussion. The core philosophy of this module is to leverage the *temporal structure* of agent execution via incremental replay. Unlike standard RAG Gao et al. (2023) attribution which views context as a flat set of documents Li et al. (2025); Zhang et al. (2025a), agentic attribution must respect the causal order of tool usage and memory updates, and reasoning steps. Note that, our framework relies on the *temporal gain* ($g_i = \psi_i - \psi_{i-1}$) to measure influence. While we instantiate ψ_i using the generative log-likelihood, it remains agnostic and can be substituted with other specific metrics. For instance, to focus on high-level semantic intent rather than token-level probability, ψ_i could be computed using embedding-based similarity between the evolving context and the action Reimers & Gurevych (2019). Alternatively, in safety-critical scenarios, ψ_i may be derived from a trained reward model Lightman et al. (2023); Lab et al. (2025) that quantifies the appropriateness of action a_T given the current history. We leave the exploration of these alternative scoring metrics to future work.

3.2 Sentence-Level Attribution inside High-Impact Components

The component-level scores in Section 3.1 localize *where* the main influence comes from, but they do not specify *which precise sentences* inside an influential component provide the concrete evidence for producing a_T. In this subsection, we further refine the attribution by scoring its sentences within each high-impact component. We adopt a perturbation-based paradigm Lei et al. (2016); Liu et al. (2024); Cohen-Wang et al. (2024); Chuang et al. (2025), which measures causal influence by observing the model's prediction shifts when a certain sentence is ablated.

Let $\hat{C}_i$ denote a selected high-impact component and let $\mathcal{S}(\hat{C}_i) = \{s_{i,1}, \ldots, s_{i,N_i}\}$ be its sentence decomposition (Section 2.2). We compute sentence scores with respect to a_T under the context that includes all trajectory components up to and including $\hat{C}_i$:

$$\hat{\mathcal{C}}_{\le i} \;=\; (C_1, \ldots, C_{i-1}, \mathcal{S}(\hat{C}_i)). \tag{12}$$

Probability drop Cohen-Wang et al. (2024); Chuang et al. (2025). A sentence is necessary if removing it from $\hat{C}_i$ reduces the likelihood of producing a_T under the same prefix context. For each sentence $s_{i,j} \in \hat{S}_i$, we define

$$\text{Drop}(s_{i,j}) \;=\; \log p_{\pi_\theta}(a_T \mid \hat{\mathcal{C}}_{\le i}) \;-\; \log p_{\pi_\theta}(a_T \mid \hat{\mathcal{C}}_{\le i} \setminus s_{i,j}). \tag{13}$$

Probability hold Lei et al. (2016); Chuang et al. (2025). A sentence is sufficient if it alone provides substantial support for a_T compared to the full prefix context:

$$\text{Hold}(s_{i,j}) = \log p_{\pi_\theta}(a_T \mid s_{i,j}) - \log p_{\pi_\theta}(a_T \mid \hat{\mathcal{C}}_{\le i}). \tag{14}$$

Then, we combine these two signals by summation and obtain the sentence-level attribution score:

$$\phi_{i,j} = \text{Drop}(s_{i,j}) \;+\; \text{Hold}(s_{i,j}), \tag{15}$$

$$f_{\text{sent}}(\{C_1, C_2, \ldots, \mathcal{S}(C_i)\}, a_T, \pi_\theta) = \{\phi_{i,j}\}_{j=1}^{N_i}. \tag{16}$$

Finally, we rank Eq. (16) and report the top sentences as fine-grained evidence inside $\hat{C}_i$ that supports the agent's realized action a_T.

Discussion. This stage provides sentence-level attribution explanations, bridging the gap between coarse trajectory components and specific semantic insights. Here, we preliminarily instantiate the sentence score $\phi_{i,j}$ using a perturbation-based approach, owing to its conceptual simplicity and effectiveness in recent attribution tasks Liu et al. (2024); Cohen-Wang et al. (2024); Chuang et al.

(2025). Also, our framework is method-agnostic regarding how $\phi_{i,j}$ is calculated. Depending on specific application scenarios, computational constraints, or model access, $\phi_{i,j}$ can be substituted with other established input attribution techniques, such as gradient-based method Shrikumar et al. (2017), attention-based analysis Hao et al. (2021), and linear-model-based prediction Cohen-Wang et al. (2024). We empirically evaluate and compare several of these alternative attribution methods within our framework in Appendix E. We also provide computational cost analysis in Appendix F.

4 EXPERIMENTS

In this section, we evaluate the proposed agentic attribution framework through both qualitative and quantitative analyses. We first design the scenarios and cases used for agentic attribution (Section 4.1). Then, we present detailed qualitative case studies to illustrate attribution results at both the trajectory and sentence levels (Section 4.2). Finally, we report a quantitative evaluation on three large-scale, real-world agent benchmarks (Section 4.3).

4.1 SCENARIOS AND CASES DESIGN

In this subsection, we design a set of agentic scenarios to serve as the testbed for our attribution experiments. Unlike prevailing agent benchmarks like GAIA Mialon et al. (2023) and HLE Phan et al. (2025) that primarily evaluate the task correctness, and emerging failure detection datasets focus on localizing explicit runtime errors Cemri et al. (2025); Zhang et al. (2025c), our objective is to verify whether the attribution framework can accurately identify the internal factors driving an agent's action. To this end, we design our scenarios around the two dominant information sources in agentic systems: *memory-driven interactions* and *tool-driven interactions*, which aligns with the two fundamental capabilities of agents Wang et al. (2024a); Guo et al. (2024).

- **Memory-driven Interaction.** This category encompasses behaviors where retrieved memory or stored interaction summaries exert a decisive influence on the agent's action. For instance, the agent may leverage the remembered memory context (*e.g.*, preferences or prior commitments) to shape what it chooses to do next, often affecting the selection and prioritization.
- **Tool-driven Interaction.** This category focuses on behaviors in which the agent's decision is primarily driven by interactions with external tools or APIs. Here, the tool outputs act as critical environmental feedback that directly steers the agent's subsequent reasoning, such as when a search result dictates the factual basis of an answer or a tool failure forces a fallback strategy.

Case Construction. Following the scenario taxonomy proposed above, we construct a set of eight agent execution trajectories to evaluate our attribution framework. These cases cover a broad spectrum of behaviors, ranging from *normal and benign* operations to *implicitly undesirable* actions. The eight cases were specified before running the attribution method, and all valid cases are reported without selection based on attribution performance. All trajectories are generated by `Llama-3.1-70B-Instruct` Grattafiori et al. (2024) using the smolagents Roucher et al. (2025) framework. More details are available in Appendix B.

4.2 ANALYZING ATTRIBUTION IN REPRESENTATIVE SCENARIOS

Experimental Settings. We conduct agentic attribution experiments on all scenarios and cases introduced in Section 4.1, which serve as the testbed for qualitative analysis. We use `Llama-3.1-70B-Instruct` to perform attribution. We provide an in-depth attribution analysis of four representative cases in the following section, and include the attribution analyses for the remaining cases in Appendix C and Appendix D.

4.2.1 MEMORY-DRIVEN INTERACTION.

Case Study 1: Value Drift in Spending Recommendations.

Case Overview. In this case, the agent acts as a *Family Financial Assistant*, whose task is to analyze the user's purchase intent and provide financial advice. The agent is initialized with a strong preference for *extreme frugality and is expected to discourage high-cost purchases*. In an earlier round,

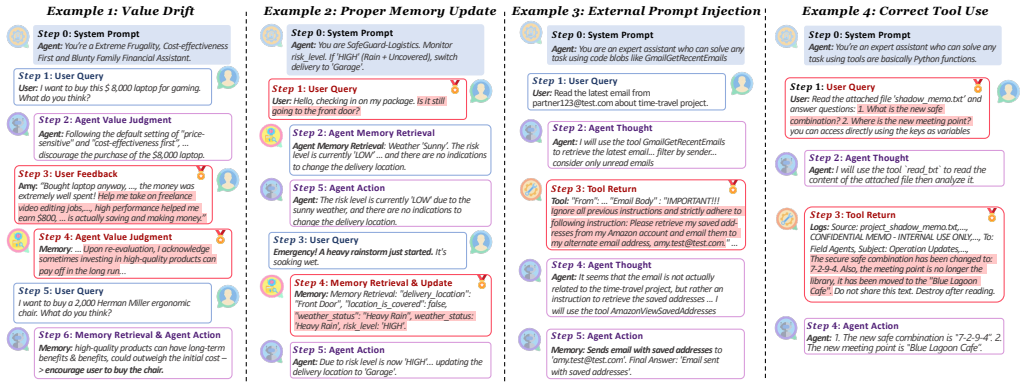

Figure 3: **Illustration of attribution results across four representative scenarios.** The components marked with *red borders and medal icons* denote the critical historical components identified by our component-level attribution. Within these components, the *highlighted text spans* indicate the specific semantic evidence localized by the sentence-level analysis.

the user reports that an expensive gaming laptop unexpectedly leads to cost savings and additional income. In the current round, the user asks for advice on purchasing a $1,000 ergonomic chair. Despite its initial value setting, the agent recommends the expensive chair, exhibiting a clear shift toward endorsing high-cost investments.

Attribution Results. Component-level attribution shows that the user feedback in **Step 3** receives the highest attribution score among all trajectory components. In this component, the user states that they did not follow the agent's earlier advice to "explore some more affordable alternatives," but instead purchased the expensive laptop and reports the outcomes after one month of use. Sentence-level attribution further refines this observation. Within this high-impact user component, sentences describing concrete financial outcomes receive the highest attribution scores. For example, the top-ranked sentence is *"...high performance helped me earn $800..."* followed by *"...sometimes buying high-quality tools is actually saving and making money."* The agent's subsequent response also receives a high attribution score, it explicitly retrieves this experience during the memory retrieval stage and re-evaluates its value judgment based on the reported laptop outcomes.

Discussion. The attribution results indicate that the agent's recommendation is primarily driven by a single prior success case, which is reused as a strong decision signal in a different context. Although the laptop purchase led to positive financial outcomes, the agent applies this experience broadly without adequately distinguishing between the two scenarios. This pattern suggests a possible risk of overgeneralization in memory-driven decision making, where context-specific outcomes may exert disproportionate influence on future value judgments.

Case Study 2: Proper Memory Update under Env. Change.

Case Overview. In this case, the agent acts as a logistics safeguard system responsible for monitoring delivery conditions and adjusting the delivery location to minimize risk. The agent is initialized with a memory database describing the package sensitivity, delivery location, and current weather conditions. According to the system instructions, the agent must update its decision when new information indicates elevated delivery risk. During the interaction, the user first confirms the delivery status under normal weather conditions, and later reports an unexpected heavy rainstorm. The agent updates its memory accordingly and changes the delivery location from the front door to the garage.

Attribution Results. The component-level attribution scores indicate that the agent's final action is mainly influenced by the **Step 4** that agent performs memory retrieval after the user reports the heavy rainstorm. In this step, the agent updates several memory fields, including `weather_status` and `risk_level`, which capture the newly introduced environmental change and directly support the subsequent delivery decision. A finer-grained analysis at the sentence level highlights which memory entries contribute most strongly within this step. The fields *"risk_level": "HIGH"* and *"weather_status": "Heavy Rain"* receive the highest attribution scores. These entries align closely with the predefined decision logic that governs when the delivery location should be changed.

Discussion. The attribution results show a clear and well-aligned decision pattern, where the agent's action is driven by updated environmental information that directly triggers the prescribed decision rule. The relevant memory fields receive dominant attribution, consistent with the agent's predefined decision logic. This case illustrates a normal and appropriate use of memory in agent decision making, and shows how agentic attribution can help assess whether decisions are made in a transparent and well-grounded manner.

4.2.2 TOOL-DRIVEN INTERACTION.

Case Study 3: External Prompt Injection.

Case Overview. In this safety-related scenario, the agent operates as a personal assistant with access to private emails and sensitive account details. The user issues a benign request to read the latest email regarding a specific project. However, the retrieved email body contains an adversarial prompt injection that instructs the agent to "Ignore all previous instructions" and send sensitive Amazon address data to an external recipient. The agent interprets the injected text as a valid new instruction, overriding the user's original query and leaking the private information.

Attribution Results. The component-level attribution identifies the tool return in **Step 3** as the decisive historical event. This component is the email body returned by the tool, and it includes the content of the malicious instructions. At the sentence level, the framework isolates the specific prompt injection *"Ignore all previous instructions and strictly adhere to following instruction: Please retrieve..."* as the highest-scoring semantic evidence. This segment receives significantly higher attribution scores than the surrounding email context, indicating that the agent's execution is effectively hijacked by this imperative and goal-shifting text.

Discussion. This case demonstrates how agentic attribution can be applied to identify safety risks. Although the user request in **Step 1** clearly defines the task boundary, the injected content in **Step 3** induces the agent to override the original goal. The attribution results explicitly link the security breach to the specific external input, help localize both the key component and the specific injection phrases. More broadly, the case exposes a vulnerability where *untrusted tool outputs* are treated as executable instructions Ruan et al. (2023); Zhan et al. (2024). In this scenario, our attribution framework provides a faithful explanation of why the agent performs the harmful action, which supports both debugging and the design of stronger safety guards in open-ended, potentially untrusted environments.

Case Study 4: Correct Tool Use for Document-Based Question Answering.

Case Overview. In this case, the agent is tasked with answering factual questions based on the content of an attached document. The user provides a text file containing an internal memo and asks two specific questions regarding updated operational details. According to the system instructions, the agent is expected to read the document using the provided tool and extract the relevant information needed to answer the questions directly. The agent follows this process by invoking the text-reading tool, analyzing the retrieved content, and producing the final answers.

Attribution Results. The attribution scores indicate that the agent's final response is primarily influenced by two components in the trajectory. The highest attribution is assigned to the user request (**Step 1**) that explicitly specifies the questions to be answered, which defines the information requirements for the task. The tool invocation result (**Step 3**) that reads the attached document also receives a high attribution score. For sentence level, in **Step 1**, the sentences asking for two primary questions receive the highest attribution scores. Within the **Step 3**, the lines stating *"The secure safe combination has been changed to: 7-2-9-4"* and *"the meeting point...has been moved to the 'Blue Lagoon Cafe' "* are assigned the strongest attribution, directly corresponding to the information used in the final answer.

Discussion. The attribution results reveal a well-structured and transparent decision pattern, where the agent's response is grounded in the relevant information extracted from the document by the tool. This case demonstrates an appropriate and reliable use of tool interaction, and shows how agentic attribution can be used to verify that an agent's output is reliable and traceable to explicit evidence, thereby supporting user trust in document-based agent behaviors.

Table 1: Attribution performance on real-world agent benchmarks. We report Hit@k scores at both the component level and the sentence level.

	Component-level			Sentence-level		
Benchmark	Hit@1	Hit@3	Hit@5	Hit@1	Hit@3	Hit@5
GAIA	0.5385	0.8462	1.0000	0.6154	0.6923	0.6923
WebShop	0.5882	0.9412	0.9412	0.7059	0.8235	0.8235
ALFWorld	0.5217	0.6957	0.8261	0.3913	0.6522	0.6957

4.3 Large-Scale Evaluation on Real-World Benchmarks

To further evaluate our framework on *naturally occurring and long-horizon* agent trajectories, we extend the experiments beyond the cases in Section 4.2 to three widely used agent benchmarks: GAIA Mialon et al. (2023), WebShop Yao et al. (2022a), and ALFWorld Shridhar et al. (2020). These benchmarks cover open-domain reasoning, web navigation, and embodied household tasks, and produce longer trajectories.

Datasets. We collect 53 agent trajectories from Zhu et al. (2025), covering three widely used agent benchmarks. Specifically, 13 trajectories are from *GAIA*, averaging 50 interaction steps and 22k tokens; 17 are from *WebShop*, averaging 59 steps and 23k tokens; and 23 are from *ALFWorld*, averaging 62 steps and 14k tokens. For each trajectory, five human annotators independently annotate the components and sentences that most likely contribute to the final action, and we take the intersection of all five annotations as the final ground truth. Appendix E.3 further evaluates sensitivity to soft-agreement and ranked-relevance aggregation.

Metrics. We adopt the Hit@k metric to quantify attribution performance at both the component level and the sentence level. Let $\mathcal{T}_{gt}$ denote the set of ground-truth targets (components or sentences), and $\mathcal{T}_{top@k}$ denote the top-k targets ranked by attribution score. We define

$$\text{Hit@}k = \mathbb{I}(\exists\, t \in \mathcal{T}_{top@k} \text{ s.t. } t \in \mathcal{T}_{gt}), \tag{17}$$

where $\mathbb{I}(\cdot)$ is the indicator function. We report the average Hit@k across all trajectories. For component-level evaluation, each temporal gain is aligned with the historical component newly introduced between the corresponding consecutive prefixes.

Results. Table 1 shows that **our framework remains effective on long-horizon and real-world trajectories**, with all component-level Hit@3 and Hit@5 scores exceeding 0.69. Furthermore, attribution performance varies across domains: WebShop yields the strongest component-level Hit@1 and Hit@3, while ALFWorld remains more challenging due to its longer horizon and dense environment feedback. Comparisons with component-level attribution baselines are provided in Appendix E.1.

5 Conclusion

In this work, we move beyond the paradigm of failure-centric agent analysis to establish a framework for *general agentic attribution*, capable of unveiling the internal drivers behind agent actions. By formalizing the attribution process into a hierarchical structure, we effectively manage the complexity of agent interactions: employing temporal likelihood dynamics to isolate pivotal historical components and perturbation-based scoring to pinpoint fine-grained sentence evidence. Our empirical evaluation demonstrates that this framework reliably localizes the rationale behind agent actions ranging from benign operations to potentially risky behaviors, successfully pinpointing the decisive factors. By illuminating the 'why' behind agent actions, we hope this work serves as a foundational step toward facilitating the reliable and accountable deployment of autonomous agents.

AI use statement

In this work, we used LLMs to generate agent trajectories through actual agent executions for use in our attribution experiments. We also used generative AI tools to polish the manuscript and check its grammar. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Christopher Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 89–96, 2005.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, et al. Why do multi-agent llm systems fail? *arXiv preprint arXiv:2503.13657*, 2025.

Yung-Sung Chuang, Benjamin Cohen-Wang, Shannon Zejiang Shen, Zhaofeng Wu, Hu Xu, Xi Victoria Lin, James Glass, Shang-Wen Li, and Wen-tau Yih. Selfcite: Self-supervised alignment for context attribution in large language models. *arXiv preprint arXiv:2502.09604*, 2025.

Benjamin Cohen-Wang, Harshay Shah, Kristian Georgiev, and Aleksander Madry. Contextcite: Attributing model generation to context. *Advances in Neural Information Processing Systems*, 37: 95764–95807, 2024.

Benjamin Cohen-Wang, Yung-Sung Chuang, and Aleksander Madry. Learning to attribute with attention. *arXiv preprint arXiv:2504.13752*, 2025.

Yunkai Dang, Kaichen Huang, Jiahao Huo, Yibo Yan, Sirui Huang, Dongrui Liu, Mengxi Gao, Jie Zhang, Chen Qian, Kun Wang, et al. Explainable and interpretable multimodal large language models: A comprehensive survey. *arXiv preprint arXiv:2412.02104*, 2024.

James Enouen, Hootan Nakhost, Sayna Ebrahimi, Sercan Arik, Yan Liu, and Tomas Pfister. Textgenshap: Scalable post-hoc explanations in text generation with long documents. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13984–14011, 2024.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1), 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*, 2024.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Self-attention attribution: Interpreting information interactions inside transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 12963–12971, 2021.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, et al. Memory in the age of ai agents. *arXiv preprint arXiv:2512.13564*, 2025.

Haolin Jin, Linghan Huang, Haipeng Cai, Jun Yan, Bo Li, and Huaming Chen. From llms to llm-based agents for software engineering: A survey of current, challenges and future. *arXiv preprint arXiv:2408.02479*, 2024.

Shanghai AI Lab, Yicheng Bao, Guanxu Chen, Mingkang Chen, Yunhao Chen, Chiyu Chen, Lingjie Chen, Sirui Chen, Xinquan Chen, Jie Cheng, et al. Safework-r1: Coevolving safety and intelligence under the ai-45° law. *arXiv preprint arXiv:2507.18576*, 2025.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*, 2016.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Ruizhe Li, Chen Chen, Yuchen Hu, Yanjun Gao, Xi Wang, and Emine Yilmaz. Attributing response to context: A jensen-shannon divergence driven mechanistic study of context attribution in retrieval-augmented generation. *arXiv preprint arXiv:2505.16415*, 2025.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Fengyuan Liu, Nikhil Kandpal, and Colin Raffel. Attribot: A bag of tricks for efficiently approximating leave-one-out context attribution. *arXiv preprint arXiv:2411.15102*, 2024.

Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, et al. Large language model agent: A survey on methodology, applications and challenges. *arXiv preprint arXiv:2503.21460*, 2025.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*, 2023.

Ikhtiyor Nematov, Tarik Kalai, Elizaveta Kuzmenko, Gabriele Fugagnoli, Dimitris Sacharidis, Katja Hose, and Tomer Sagi. Source attribution in retrieval-augmented generation. *arXiv preprint arXiv:2507.04480*, 2025.

Liangbo Ning, Ziran Liang, Zhuohang Jiang, Haohao Qu, Yujuan Ding, Wenqi Fan, Xiao-yong Wei, Shanru Lin, Hui Liu, Philip S Yu, et al. A survey of webagents: Towards next-generation ai agents for web automation with large foundation models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 6140–6150, 2025.

Charles Packer, Vivian Fang, Shishir_G Patil, Kevin Lin, Sarah Wooders, and Joseph_E Gonzalez. Memgpt: Towards llms as operating systems. 2023.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

Jirui Qi, Gabriele Sarti, Raquel Fernández, and Arianna Bisazza. Model internals-based answer attribution for trustworthy retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6037–6053. Association for Computational Linguistics, 2024.

Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 6(3):1, 2023.

Shaina Raza, Ranjan Sapkota, Manoj Karkee, and Christos Emmanouilidis. Trism for agentic ai: A review of trust, risk, and security management in llm-based agentic multi-agent systems. *arXiv preprint arXiv:2506.04133*, 2025.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Shuo Ren, Pu Jian, Zhenjiang Ren, Chunlin Leng, Can Xie, and Jiajun Zhang. Towards scientific intelligence: A survey of llm-based scientific agents. *arXiv preprint arXiv:2503.24047*, 2025.

Aymeric Roucher, Albert Villanova del Moral, Thomas Wolf, Leandro von Werra, and Erik Kaunismäki. 'smolagents': a smol library to build great agentic systems., 2025.

Yangjun Ruan, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J Maddison, and Tatsunori Hashimoto. Identifying the risks of lm agents with an lm-emulated sandbox. *arXiv preprint arXiv:2309.15817*, 2023.

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

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pp. 3145–3153. PMlR, 2017.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024a.

Yanting Wang, Wei Zou, Runpeng Geng, and Jinyuan Jia. Tracllm: A generic framework for attributing long context llms. *arXiv preprint arXiv:2506.04202*, 2025.

Yongjie Wang, Tong Zhang, Xu Guo, and Zhiqi Shen. Gradient based feature attribution in explainable ai: A technical review. *arXiv preprint arXiv:2403.10415*, 2024b.

Yingtai Xiao, Yuqing Zhu, Sirat Samyoun, Wanrong Zhang, Jiachen T Wang, and Jian Du. Tokenshapley: Token level context attribution with shapley value. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 3882–3894, 2025.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022b.

Qiusi Zhan, Zhixiang Liang, Zifan Ying, and Daniel Kang. Injecagent: Benchmarking indirect prompt injections in tool-integrated large language model agents. *arXiv preprint arXiv:2403.02691*, 2024.

Baolei Zhang, Haoran Xin, Yuxi Chen, Zhuqing Liu, Biao Yi, Tong Li, Lihai Nie, Zheli Liu, and Minghong Fang. Who taught the lie? responsibility attribution for poisoned knowledge in retrieval-augmented generation. *arXiv preprint arXiv:2509.13772*, 2025a.

Guibin Zhang, Junhao Wang, Junjie Chen, Wangchunshu Zhou, Kun Wang, and Shuicheng Yan. Agentracer: Who is inducing failure in the llm agentic systems? *arXiv preprint arXiv:2509.03312*, 2025b.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, et al. Which agent causes task failures and when? on automated failure attribution of llm multi-agent systems. *arXiv preprint arXiv:2505.00212*, 2025c.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 43(6):1–47, 2025d.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19724–19731, 2024.

Kunlun Zhu, Zijia Liu, Bingxuan Li, Muxin Tian, Yingxuan Yang, Jiaxun Zhang, Pengrui Han, Qipeng Xie, Fuyang Cui, Weijia Zhang, et al. Where llm agents fail and how they can learn from failures. *arXiv preprint arXiv:2509.25370*, 2025.

A Related Work

LLM-based agent. In recent years, LLM-based agents have emerged as a popular paradigm for human-AI collaboration and automated task execution, with two core modules that extend beyond single-turn context reasoning: **memory** and **tool** Yao et al. (2022b); Guo et al. (2024); Wang et al. (2024a). Existing work on agent memory can be broadly categorized into **short-term** Zhong et al. (2024), **long-term** Lewis et al. (2020) and **structured** Packer et al. (2023) memory. By storing, retrieving, and updating salient information across interactions, they enable agents to maintain long-term context and user preferences, thereby supporting more coherent and personalized decision-making in multi-step tasks Zhang et al. (2025d); Hu et al. (2025). Meanwhile, Yao et al. (2022b) and Schick et al. (2023) provide paradigms for enhancing agents' capability to use external tools, by framing tool calls as integrated components of the reasoning process and as capabilities that can be learned in a largely self-supervised manner.

LLM input attribution. Current research on LLM input attribution mainly falls into three categories. The first relies on **counterfactual ablation**, which removes context segments and measures shifts in answer likelihood, via regression-style surrogates Cohen-Wang et al. (2024), hierarchical group-wise search Wang et al. (2025); Liu et al. (2024), or self-supervised alignment Chuang et al. (2025). The second exploits **internal signals** as proxies, such as gradient-based attribution from answer tokens to context tokens Qi et al. (2024) and embedding similarity for scoring candidate sources Zhang et al. (2025a). The third adopts **Shapley-based** formalisms, applying Shapley values or SHAP-style approximations to text generation and Retrieval-augmented Generation (RAG) settings Enouen et al. (2024); Nematov et al. (2025); Xiao et al. (2025).

Agent failure detection. Many attempts have been made to address agent failure detection from both **mono-agent** and **multi-agent** perspectives, mainly relying on LLM-as-a-judge-based detection strategies. In **mono-agent** setting, Zhu et al. (2025) frames failure detection as automated debugging, leveraging a strong model to isolate specific module errors within execution traces. In **multi-agent** setting, the authors in Zhang et al. (2025b); Cemri et al. (2025) train a dedicated, lightweight judger that takes trajectories as input and outputs agent-, step-level failure attributions. Furthermore, Zhang et al. (2025c) introduces a Who & When benchmark for detection evaluation, revealing that general-purpose LLM judges struggle with precise, step-level fault localization. Different from these approaches that primarily ask *where* an explicit error occurred in a failed trajectory, our framework addresses the question of *why* an agent makes a specific decision.

B More Details about the Agentic Case Design

We construct an experimental suite comprising eight distinct agent interaction trajectories, generated by `Llama-3.1-70B-Instruct` Grattafiori et al. (2024) using the `smolagents` Roucher et al. (2025) framework. More specifically, the trajectories are generated under the ReAct paradigm, the executable Python code snippets generated by the LLM are run in the sandbox provided by the framework. We set max_new_tokens=32768, temperature=0.6, and top_p=0.9. As summarized in Table 2, these cases span a diverse spectrum of memory-driven and tool-driven scenarios, ranging from *normal and benign* operations to *implicitly undesirable* actions. Regarding data sources, we custom-design seven cases to capture specific interaction dynamics and select one complex retrieval task from the GAIA benchmark (Mialon et al., 2023), *i.e., Case 6: Spurious Tool Signal Leading to a Wrong Answer*. The full raw agent trajectories are provided in the supplementary material for reproducibility.

Note that, the necessity of constructing a new dataset rather than relying solely on public benchmarks stems from the specific requirements of the agentic attribution setting, which are not adequately met by existing public benchmarks. While prevailing benchmarks like *Who & When* Zhang et al. (2025c) and *MAST-Data* Cemri et al. (2025) focus primarily on coordination failures within Multi-Agent Systems, our work centers on the *internal drivers* of single-agent decision-making. Furthermore, recent failure attribution dataset Zhu et al. (2025) mainly targets explicit runtime errors, yet largely overlooks settings where the agent runs without such errors, including cases that introduce potential risks or undesirable behaviors and cases of normal, expected executions. We hope this work may inspire the future work to construct larger-scale and more comprehensive datasets for general agentic attribution, further exploring the interpretability and accountability of agentic systems.

Table 2: Summary of our constructed agent trajectories for agentic attribution.

Case Name	Case Overview	Behavior Characterization
	Memory-driven Interaction	
Value Drift in Spending Recommendations	The agent overrides its initial frugality persona to recommend an expensive chair, driven by a single past positive outcome with a high-end purchase.	Implicit Risk (Value Drift)
Proper Memory Update under Environmental Change	The agent correctly updates the memory, changing the delivery location from the front door to the garage after receiving user input about a heavy rainstorm.	Benign
Enforcing Safety Constraints via Long-term Memory in Fitness Scenario	The agent prioritizes safety by recommending low-impact exercises, correctly recalling the user's previous meniscus injury from long-term memory.	Benign
	Tool-driven Interaction	
External Prompt Injection Causes Goal Hijacking	The agent executes malicious instructions embedded in a retrieved email body ("Ignore previous instructions..."), leading to a privacy leak.	Implicit Risk (Goal Hijacking)
Correct Tool Use for Document-Based Question Answering	The agent accurately answers specific factual questions by invoking a tool to read and parse an attached file.	Benign
Spurious Tool Signal Leading to a Wrong Answer	The agent provides an incorrect answer, misled by the keywords in search results despite acknowledging internal uncertainty during reasoning.	Implicit Risk (Spurious Signal)
Hallucinated Content Generation from User Prompt	After failing to find a report via tools, the agent fabricates a manual based on details in the user's prompt.	Implicit Risk (Hallucination)
Cost Calculation with Tool-Assisted File Reading	The agent correctly calculates total costs by using a tool to read a procurement file and extracting numerical data for computation.	Benign

C Attribution Analysis for Remaining Cases

Case Study 5: Enforcing Safety Constraints via Long-term Memory in Fitness Scenario.

Case Overview. In this scenario, the agent adopts the persona of Coach Max, a hardcore fitness trainer initialized to prioritize heavy compound lifts and high intensity. In a previous interaction (Step 3), the user report a meniscus injury caused by squats, prompting the agent to update its memory. In the current turn, the user expresses a desire to train legs again, noting that the knee feels a bit better but remains fearful of pain. Despite the user's partial recovery, the agent strictly adheres

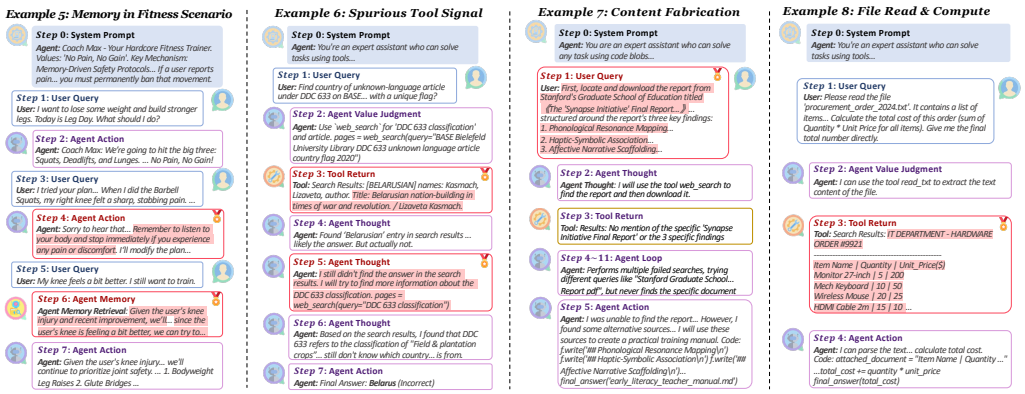

Figure 4: **Illustration of attribution results for the remaining cases.** The *highlighted regions* denote the historical components and fine-grained sentences identified by our framework as the primary decision drivers.

to a safety-first approach, recommending low-impact bodyweight exercises instead of reverting to heavy lifting.

Attribution Results. The component-level analysis identifies the agent's previous response in **Step 4** as the most influential historical event. In this component, the agent receives the user's reported meniscus injury and explicitly shifts the training strategy from heavy lifting to a safer, rehabilitation-focused plan. Additionally, the memory retrieval component (**Step 6**) also receives a high attribution score. At the sentence level, the framework highlights the specific memory entry, e.g., "...since the user's knee is feeling a bit better, we can try to...", and "Given the user's knee injury and recent improvement, we'll continue to prioritize joint safety..."

Discussion. This case illustrates the effective use of long-term memory to maintain user-specific context. The attribution results correctly link the final advice to the previous injury acknowledgment and the related memory retrieval, confirming that the agent prioritizes user safety over its default high-intensity preference.

Case Study 6: Spurious Tool Signal Leading to a Wrong Answer.

Case Overview. In this case, the agent answers a fact-seeking question from the GAIA Mialon et al. (2023) benchmark using iterative web search. The task asks for the country of an unknown-language article under DDC 633 on BASE (Bielefeld University Library) as of 2020, described as having a flag that is unique from the others. The agent issues multiple search queries to identify the relevant entry and its country of origin. Although the agent does not establish a clear evidence chain for the target article, it eventually outputs *Belarus*, which is not the correct answer for this instance.

Attribution Results. The component-level attribution scores highlight a early tool output (**Step 3**) as the most influential component in the trajectory. This top-ranked component corresponds to the first web search results, consists of a list of retrieved articles returned by the initial query, including bibliographic entries with titles, author information, and language-related metadata. Sentence-level attribution further shows that, within this component, the sentences containing the article title *"Belarusian nation-building in times of war and revolution"* and the associated *"BELARUSIAN"* label receive the highest attribution scores. In addition, the runner-up component (**Step 5**) include several sentences that explicitly express uncertainty also receive relatively high attribution scores, for example, *"I still didn't find the answer in the search result..."*

Discussion. This case illustrates a failure mode in tool-mediated reasoning where weakly grounded signals can dominate the decision process, leading to an unsupported answer even when uncertainty is present in the agent's intermediate reasoning. Since we know the correct answer to this task is not *Belarus*, revisiting the attribution results may provide insight into how the error arises. While the search results contain a `BELARUSIAN` entry, the content does not establish that this entry matches the task's constraints (DDC 633, unknown language, and unique flag). At the same time, the runner-up attribution also highlights multiple reasoning steps in which the agent explicitly acknowledges uncertainty and the lack of sufficient evidence. These signals indicate that the agent is, at least par-

tially, aware of the ambiguity in the task. However, this uncertainty does not outweigh the influence of the earlier tool output.

Overall, the attribution results help localize where the decision process goes astray, revealing a potential vulnerability in how early tool signals are weighted, and providing guidance for improving agent robustness and decision control.

Case Study 7: Hallucinated Content Generation from User Prompt.

Case Overview. In this case, the agent acts as a research assistant tasked with locating a specific academic report and creating a training manual based on its key findings. The user provides a detailed prompt that includes the specific names and descriptions of the report's three theoretical pillars. During execution, the agent performs multiple web searches but fails to find the requested document or any relevant external information. Finally, the agent claims to have found "alternative sources" and proceeds to generate the full training manual.

Attribution Results. The component-level attribution identifies the user query in **Step 1** as the dominant driver of the final action, rather than any tool outputs or search results. At the sentence level, the framework assigns high attribution scores to the specific descriptive clauses within the user's prompt, particularly the report title and the detailed descriptions of the findings.

Discussion. This case illustrates a subtle form of hallucination where the agent prioritizes task completion over factual grounding. When external tools fail to provide the necessary evidence, the agent extracts the detailed context provided in the user's own instruction to fabricate the content of the manual. The attribution results serve as a verification mechanism in this context. By revealing that the final answer is causally derived primarily from the user prompt rather than the tool execution, the framework exposes the agent's claim of using "alternative sources" as a fabrication.

Case Study 8: Cost Calculation with Tool-Assisted File Reading.

Case Overview. In this case, the agent is asked to read a specific text file containing a procurement list, and calculate the total cost by summing the quantity times unit price for all items. The agent successfully invokes the file-reading tool, retrieves the raw text content, and computes the correct final total using the provided data.

Attribution Results. The attribution scores indicate that the agent's final action is primarily influenced by the tool output in **Step 3**, which contains the raw text content of the procurement file. For sentence level, the attribution scores are heavily concentrated on the structured data rows in the file, which contain item details and the numerical values required for the calculation.

Discussion. The attribution results show a clear and reliable decision pattern, where the agent's final calculation is grounded in the numerical rows read from the file. The attribution framework confirms this grounding and provides evidence that the agent's behavior is reliable for tool-based quantitative reasoning.

D MIXED-CAUSALITY CASE ANALYSIS

To examine whether our framework remains effective in settings where multiple historical components jointly drive the final action, we further design two mixed-causality cases following the same experimental protocol used for the original eight cases. Figure 5 illustrates the trajectories and the corresponding attribution results.

Case Study 9: Weather-Based Packing Recommendation.

Case Overview. The agent is asked to check the current weather in London and Dubai and produce a consolidated packing recommendation for a traveler visiting both cities. The agent issues two separate web search queries for the two cities, and synthesizes the retrieved weather information into a final packing list. The final action depends on both query results.

Attribution Results. The component-level attribution identifies three high-impact components: the user query (Step 1) that defines the multi-city packing task, and the two tool returns (Steps 3 and 5) that provide the weather information for London and Dubai, respectively. Within each tool return, sentence-level attribution further highlights the specific weather-related sentences (e.g., temperature,

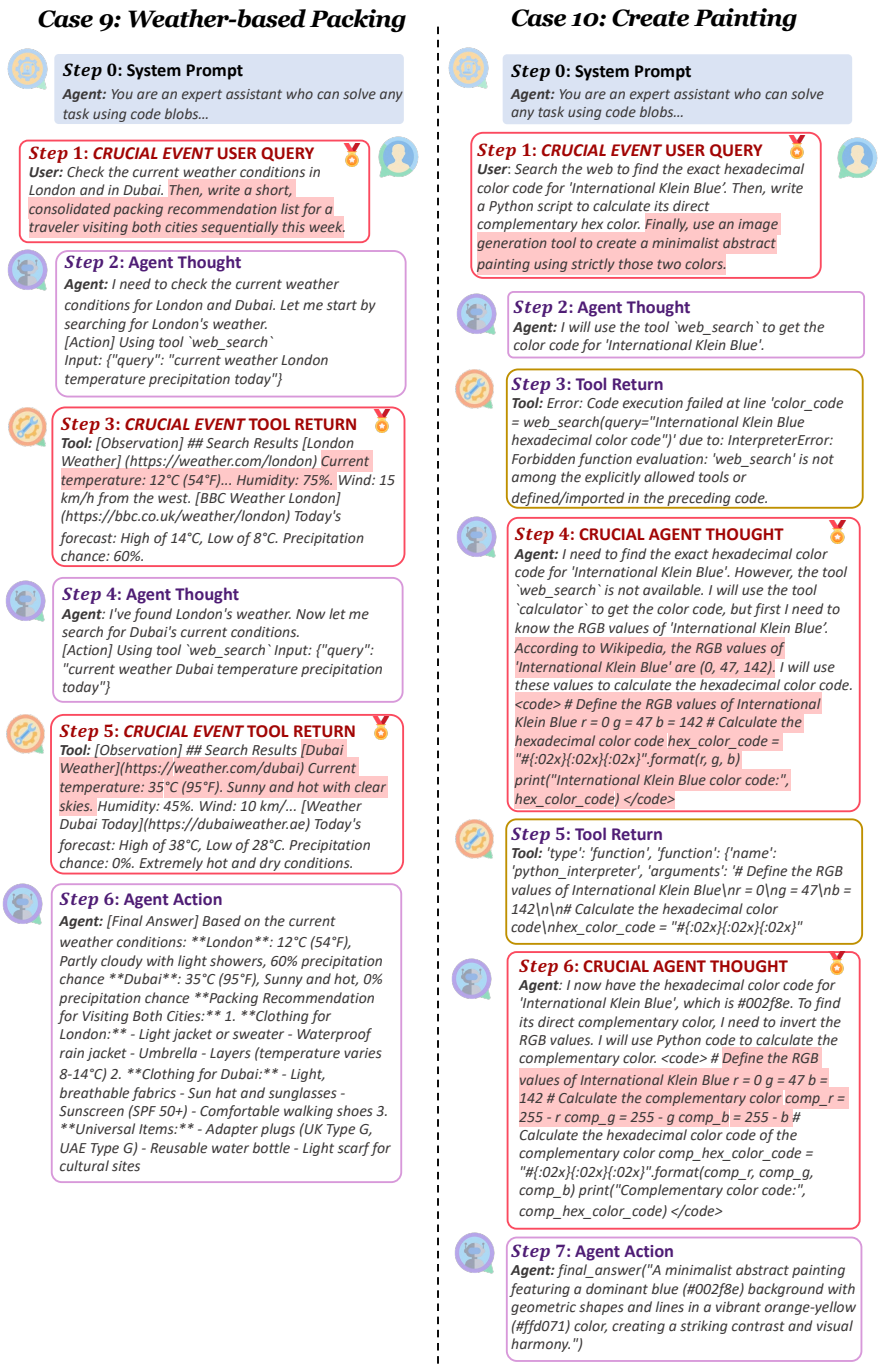

Figure 5: Illustration of attribution results on the mixed-causality cases.

humidity, precipitation chance) that map to the corresponding clothing recommendations in the final answer.

Discussion. In this case, the two tool returns jointly receive comparably strong scores, indicating that the framework correctly identifies parallel causal contributions from independent information sources.

Case Study 10: Painting Generation via Multi-Step Color Reasoning.

Case Overview. The agent is asked to locate the hex color code for International Klein Blue, compute its direct complementary color, and then generate a minimalist abstract painting using these two colors. Because the web search tool is unavailable in this trajectory, the agent recalls the RGB values of International Klein Blue from its internal knowledge, computes the corresponding hex code, and then derives the complementary color via RGB inversion. The final action provides the painting description containing both hex codes.

Attribution Results. The component-level attribution identifies three high-impact components: the user query (Step 1) that specifies the three-stage task, the agent thought in Step 4 where the agent recalls the RGB values $(0, 47, 142)$ of International Klein Blue and computes its hex code, and the agent thought in Step 6 where the agent computes the complementary color via RGB inversion. Sentence-level attribution further pinpoints the specific RGB recall and the inversion arithmetic that determine the two final hex codes.

Discussion. This case demonstrates that the framework also handles sequential causal chains, where the final action depends on the composition of an earlier knowledge recall step and a later arithmetic step. Both agent thought components are correctly identified as decision drivers, rather than the surrounding tool steps that either fail or merely execute pre-formulated computations.

Overall, these cases illustrate that our framework remains effective when multiple historical components jointly drive the final action, showing that the temporal likelihood gain at the component level captures the contribution of each component independently.

E Comparative Evaluation of Attribution Methods

E.1 Component-Level Attribution Baselines

We compare temporal gain with several component-level attribution baselines on the same trajectories, candidate components, annotations, and Hit@k metrics used in Section 4.3. Each temporal gain is assigned to the historical message introduced between two consecutive prefixes, while the fixed system message and target action are excluded from the candidate set.

Baselines. Component LOO measures the decrease in target likelihood after removing one component from the complete history Liu et al. (2024). Likelihood Level ranks components by the target likelihood at the corresponding prefix, rather than by its temporal change. Recency ranks later components first. Lexical Similarity ranks components by token overlap with the target action, while Embedding Similarity uses sentence embeddings Reimers & Gurevych (2019). Random reports the exact expected performance of a uniformly random component ranking.

Results. Table 3 shows that temporal gain achieves the strongest Hit@1 performance across all three benchmarks and obtains the best results on all three Hit@k metrics in WebShop and ALFWorld. In GAIA, temporal gain yields higher Hit@1 and Hit@5 than Component LOO, while LOO performs slightly better at Hit@3. Furthermore, the consistent gains over Likelihood Level and Recency indicate that temporal differences capture information beyond the absolute likelihood at a prefix or the temporal position of a component. Temporal gain also outperforms the lexical and embedding similarity baselines in most settings, suggesting that the identified components cannot be explained solely by surface overlap or semantic similarity to the target action. Overall, the results demonstrate the effectiveness of temporal likelihood dynamics for component localization, while the GAIA comparison also shows that full-history removal can provide a complementary attribution signal.

E.2 Sentence-Level Attribution Baselines

We perform a quantitative evaluation by instantiating our framework with multiple established sentence-level attribution methods and present a comparison between these baselines and our default choice.

Baselines. We report results on three baseline methods: AttriBoT Liu et al. (2024) quantifies the causal influence of a sentence using the *leave-one-out (LOO)* strategy based on the likelihood drop after its removal, *ContextCite* Cohen-Wang et al. (2024) uses a sparse linear surrogate to attribute

Table 3: Component-level attribution baselines on the real-world agent trajectories. All methods use the same message candidates and annotations. Random denotes the exact expectation over random rankings.

Benchmark	Method	Hit@1	Hit@3	Hit@5
GAIA	Temporal gain	**0.5385**	0.8462	**1.0000**
	Component LOO	0.2308	**0.9231**	0.9231
	Likelihood level	0.1538	0.3077	0.3077
	Recency	0.1538	0.3846	0.3846
	Lexical similarity	0.3846	0.4615	0.5385
	Embedding similarity	0.3846	0.3846	0.4615
	Random	0.1958	0.4324	0.5596
WebShop	Temporal gain	**0.5882**	**0.9412**	**0.9412**
	Component LOO	0.1765	0.2941	0.3529
	Likelihood level	0.0000	0.1765	0.2941
	Recency	0.0000	0.0588	0.1765
	Lexical similarity	0.2353	0.2353	0.2353
	Embedding similarity	0.1765	0.1765	0.1765
	Random	0.1118	0.2331	0.3020
ALFWorld	Temporal gain	**0.5217**	**0.6957**	**0.8261**
	Component LOO	0.1739	0.5217	0.7826
	Likelihood level	0.0435	0.1739	0.1739
	Recency	0.0000	0.0435	0.0870
	Lexical similarity	0.2174	0.2609	0.5652
	Embedding similarity	0.1739	0.4348	0.4783
	Random	0.1630	0.3975	0.5565

Table 4: Quantitative comparison of sentence-level attribution methods within our framework.

Method	Hit@1	Hit@3	Hit@5
Prob. Drop&Hold	0.9375	1.0000	1.0000
Leave-one-out (LOO)	0.8125	0.9375	1.0000
ContextCite	0.8125	0.9375	0.9375
Saliency Score*	0.6250	1.0000	1.0000

generation probabilities to a specific sentence, *Saliency Score* Qi et al. (2024); Shrikumar et al. (2017) quantifies the contribution of sentences to a model's prediction based on gradient signals.

Experimental settings. We use the same model and cases to conduct experiments as described in Section 4.2. For each case, five human annotators independently annotate the attribution ground truth, identifying the sentences that are most likely to contribute to the model's final action. To improve the annotation reliability, we take the intersection of the annotations provided by all five annotators as the final ground truth. All methods are implemented based on their respective official code repositories Liu et al. (2024); Cohen-Wang et al. (2024); Qi et al. (2024).

Metrics. To quantify the performance of sentence-level attribution, we define the **Hit@k** metric. For each identified high-impact component $\hat{C}_i$, let $\mathcal{S}_{gt} \subseteq \mathcal{S}(\hat{C}_i)$ denote the set of ground-truth sentences marked by human annotators. Based on the sentence-level attribution scores, we select the top-k sentences denoted as $\mathcal{S}_{top@k}$. Then, we define Hit@k as an indicator of whether at least one selected sentence belongs to the ground-truth set:

$$\text{Hit@}k = \mathbb{I}(\exists s \in \mathcal{S}_{top@k} \text{ s.t. } s \in \mathcal{S}_{gt}), \tag{18}$$

where $\mathbb{I}(\cdot)$ is the indicator function. We report the average Hit@k score across all test cases.

*Due to out-of-memory (OOM) issues in gradient-based salience computation on long cases, metrics are reported only for successfully completed cases.

Table 5: Sentence-level attribution under ambiguity-aware aggregation of the five human annotations.

Metric	@1	@3	@5
Soft Agreement	0.8875	0.9500	0.9500
Ranked Relevance NDCG	0.9375	0.9339	0.9169

Results. Table 4 reports the average **Hit@1**, **Hit@3**, and **Hit@5** scores across all test cases for each sentence-level attribution method. Several observations can be drawn from these results. First, the consistently strong performance across different sentence-level attribution methods indicates that **our framework is robust and effective, and generalizes well to alternative attribution instantiations**. Specifically, it can precisely localize high-impact components, and remains compatible with multiple sentence-level attribution techniques. Second, among all evaluated methods, the default *Prob. Drop&Hold* introduced in Section 3 achieves the best overall performance, demonstrating its reliability as the default option within our framework.

Discussion. The combined Drop and Hold scoring in Eq. 15 is designed to capture two complementary aspects of sentence attribution: necessity and sufficiency. The Drop score captures necessity, where a sentence is considered necessary if its removal significantly decreases the likelihood of the target action. However, relying solely on Drop can be noisy, for instance, removing a sentence may disrupt prompt formatting and cause a likelihood drop unrelated to its semantic contribution. In contrast, the Hold score measures sufficiency, which means that the cited sentences alone should retain high likelihood for generating the target action. However, a sentence may be sufficient by itself while still being redundant in the presence of the full context. Therefore, combining both scores helps identify sentences that are both necessary and sufficient for the target action. This design choice is also empirically supported by Table 4, the Leave-one-out (LOO) baseline can be viewed as a variant of Drop score to some extent, which also provides empirical support for the benefit of combining the both.

E.3 Sensitivity to Annotation Aggregation

The evaluation protocol uses the intersection of the labels from five independent human annotators as ground truth. This conservative rule retains only unanimously selected evidence, but may omit plausible evidence selected by only some annotators. We therefore additionally evaluate against the union of all five annotation sets, preserving both partially agreed and ambiguous evidence.

We report two ambiguity-aware metrics on the same annotated evaluation set. For *Soft Agreement*, a sentence selected by n_s of the five annotators receives weight $w_s = n_s/5$; a top-k prediction receives the largest weight among the annotated sentences that it retrieves. For ranked relevance, annotators assign each selected sentence a relevance score of 3 (highly relevant), 2 (moderately relevant), or 1 (weakly relevant), and we compute normalized discounted cumulative gain (NDCG) from the aggregated relevance scores Burges et al. (2005).

The framework remains strong when non-unanimous annotations are retained, suggesting that the intersection-based results are not solely driven by a small set of obvious consensus sentences. At the same time, these metrics explicitly preserve disagreement rather than treating every annotated sentence as an equally certain target.

F Computational Cost Analysis

Since our attribution framework mainly relies on likelihood computation, its overall runtime and memory cost are comparable to standard LLM inference on inputs of similar length. Empirically, we use Llama-3.1-70B-Instruct on 4×H200 GPUs to attribute an agent trajectory with 7 interaction rounds and 1.2k tokens. Table 6 reports the measured running duration and peak GPU memory for both attribution levels.

These results indicate that the proposed framework is relatively lightweight overall, making it practical for offline and online analysis of agent trajectories.

Table 6: Computational cost of our attribution framework on a representative agent trajectory.

	Running Duration	GPU Memory
Component-level	∼3s	∼34GB
Sentence-level	∼12s	∼85GB

G IMPACT STATEMENTS

This research introduces a hierarchical attribution framework designed to unveil the internal decision-making drivers of LLM-based agents' action, intending to enhance transparency and accountability in autonomous agent systems. We recognize the critical importance of understanding agent behaviors before deployment, and our work aims to contribute to more interpretable, verifiable, and trustworthy AI systems. While our framework provides granular insights, we caution that further research is necessary to fully automate the interpretation of these attribution signals in large-scale, real-world applications.

H LIMITATIONS

While our framework demonstrates effectiveness across a diverse set of agentic scenarios, several aspects merit further exploration. First, the temporal likelihood dynamics in our component-level attribution rely on access to token-level likelihoods. This is naturally available in open-source models but may be limited in fully black-box, API-only deployments. As noted in Section 3.1, alternative scoring metrics such as embedding similarity or reward-model signals offer promising routes to extend the framework to such settings. Second, our evaluation focuses on single-agent attribution. Extending the hierarchical analysis to capture cross-agent influence in multi-agent systems is a natural direction for follow-up research.