AgentHallu: Benchmarking Automated Hallucination Attribution of LLM-based Agents

Xuannan Liu[1*] **Xiao Yang**[2*] **Zekun Li**[3] **Peipei Li**[1†] **Ran He**[4]
[1] Beijing University of Posts and Telecommunications
[2] Department of Computer Science & Technology, Tsinghua University
[3] Minzu University of China
[4] Center for Research on Intelligent Perception and Computing, NLPR, CASIA
https://liuxuannan.github.io/AgentHallu.github.io/

Abstract

As LLM-based agents operate over sequential multi-step reasoning, hallucinations arising at intermediate steps risk propagating along the trajectory, thus degrading overall reliability. Unlike hallucination detection in single-turn responses, diagnosing hallucinations in multi-step workflows requires identifying which step causes the initial divergence. To fill this gap, we propose a new research task, **automated hallucination attribution** of LLM-based agents, aiming to identify the step responsible for the hallucination and explain why. To support this task, we introduce AgentHallu, a comprehensive benchmark with: (1) 693 high-quality trajectories spanning 7 agent frameworks and 5 domains, (2) a hallucination taxonomy organized into 5 categories (Planning, Retrieval, Reasoning, Human-Interaction, and Tool-Use) and 14 sub-categories, and (3) multi-level annotations curated by humans, covering binary labels, hallucination-responsible steps, and causal explanations. We evaluate 13 leading models, and results show the task is challenging even for top-tier models (like GPT-5, Gemini-2.5-Pro). The best-performing model achieves only 41.1% step localization accuracy, where tool-use hallucinations are the most challenging at just 11.6%. We believe AgentHallu will catalyze future research into developing robust, transparent, and reliable agentic systems.

1 Introduction

Large Language Models (LLMs) (OpenAI, 2025; Comanici et al., 2025) have been increasingly deployed into autonomous agents to tackle complex tasks (Mialon et al., 2024; Yang et al., 2024). However, such agents do not always achieve the intended task outcome, leading to agent failures. One type of agent failure arises from execution errors (e.g., turn limits), which prevent agents from producing valid responses. Since these failures are non-deceptive and readily observable, prior failure attribution works (Zhang et al., 2025b; Cemri et al., 2025; Rahardja et al., 2025) mimic human debugging processes to diagnose the failure origin given known fault labels. In contrast, hallucination errors occur when agents successfully respond to user queries but produces plausible yet non-factual content. Such deceptive outputs can easily mislead users into accept them as correct. Therefore, this underscores an urgent need for granular analyses of hallucinations in agentic workflows.

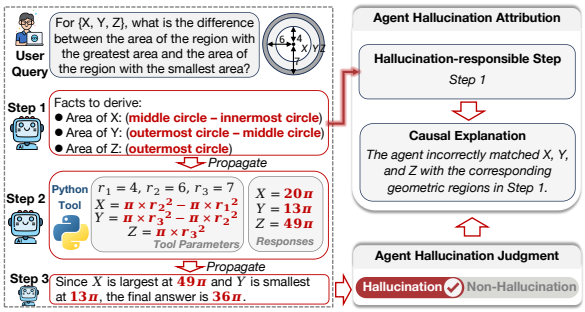

Figure 1: Illustration of hallucination attribution in LLM-based agents. **Left:** A misdefinition of regions X, Y, Z in Step 1 propagates to the tool call and leads to the incorrect final answer. **Right:** Beyond binary judgment, hallucination attribution aims to identify a hallucination-responsible step and a causal explanation.

Unlike LLM hallucinations confined to single-turn responses, agent hallucinations exhibit two distinct properties: **(1) Hallucination Propagation.** Due to the sequential nature of multi-step workflows, hallucinations in intermediate reasoning can propagate through subsequent actions and ultimately corrupt the final response (Zhou et al., 2025). As shown in Figure 1 **Left**, a planning hallucination misdefines "region X, Y, Z", which then propagates to downstream Python tool parameters and leads to an incorrect final answer. **(2) Multi-source Knowledge.** Agent outputs arise from integrating diverse functional modules, including long-horizon planning, multi-hop retrieval, iterative tool use, dynamic reasoning, and human

*Equal contribution. † Corresponding author.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 21290–21315
October 24-29, 2026 ©2026 Association for Computational Linguistics

interaction. These modules incorporate both LLM internalized and external environmental knowledge, which risk injecting non-factual information.

Current hallucination evaluations (Bang et al., 2025; Niu et al., 2024; Li et al., 2023b) primarily classify single-turn LLM responses as factual or hallucinated. While valuable, this binary paradigm fails to address concerns essential for building reliable agents: **where** and **why** hallucinations originate in agentic workflows. To fill this gap, we propose a novel research task of **automated hallucination attribution** for LLM-based agents. We define two key objectives: **(1) Hallucination-responsible Step Localization (Where):** identify the step responsible for the hallucinated result, (2) **Causal Explanation (Why):** provide an open-ended explanation of the underlying cause. As shown in Figure 1 **Right**, step attribution precisely identifies "Step 1" as the hallucination origin, while causal explanation provides a corresponding diagnostic analysis of "Step 1 incorrectly matched X, Y, and Z with their regions".

To support this task, we present AgentHallu, the first comprehensive benchmark tailored for automated hallucination attribution of multi-step agent trajectories. As shown in Table 1, the key highlights of the AgentHallu dataset include: *(1) Extensive Diversity.* We collect 693 trajectories from 7 popular agent frameworks with an average length of 7.6 steps. The dataset encompasses five distinct domains: world knowledge, science, math, general assistant, and tool use. *(2) High-quality Control.* We implement a rigorous three-stage filtering criterion to exclude non-deceive failures, overly short sequences, and trivial cases lacking diagnostic depth, thereby ensuring the benchmark's difficulty. *(3) Comprehensive Taxonomy.* We develop a hierarchical taxonomy of agent hallucinations via grounded theory (Glaser and Strauss, 2017), resulting in 5 primary categories (Planning, Retrieval, Reasoning, Human-Interaction, and Tool-Use) and 14 granular subcategories. *(4) Multi-level Annotation.* AgentHallu includes binary labels for judgment. For attribution, it specifies hallucination-responsible steps and explains the underlying cause in plain language. All annotations are manually curated through a labor-intensive process.

Using the AgentHallu, we develop an attribution evaluation framework along two dimensions: step localization accuracy as a measure of responsible-step identification, and G-EVAL scores (Liu et al., 2023) for assessing the quality of open-ended explanations. Leveraging this framework, we evaluate 13 leading LLMs, including 5 proprietary and 8 open-source models. Empirical results reveal several critical findings: (1) The best-performing model, Gemini-2.5-Pro, achieves only 41.1% accuracy in step localization, which drops to 11.6% accuracy on tool-use hallucinations. (2) Step-by-Step prompting improves attribution via incremental processing, but at the cost of higher token usage. (3) Increasing trajectory steps N_{step} poses a challenge to attribution, with GPT-5's accuracy dropping from 40.3% ($N_{\text{step}} \leq 5$) to 23.9% ($N_{\text{step}} \geq 11$).

Overall, our contributions include: (i) A novel task of automated hallucination attribution in LLM-based agents to understand where and why hallucinations originate. (ii) A comprehensive benchmark comprising 693 high-quality trajectories with broad diversity, a systematic taxonomy and multi-level annotations. (iii) Evaluation of 13 leading LLMs, revealing their strengths and limitations under varying conditions, including hallucination categories, prompting methods, and trajectory steps.

2 Related Work

2.1 Hallucination Detection Benchmarks

Hallucination detection aims to develop a framework or a model to automatically distinguish between hallucinated and factual content (Li et al., 2025; Ravichander et al., 2025; Qin et al., 2025; Zhang et al., 2025c,a). As shown in Table 1, a line of work assesses the model's factuality reasoning over diverse domains such as world knowledge (Li et al., 2023b; Wei et al., 2024b; Bang et al., 2025), science (Chen et al., 2023), and math (Chen et al., 2023). Moreover, RAGTruth (Niu et al., 2024) demonstrates that popular LLMs continue to hallucinate across tasks even with retrieval-augmented generation. To diagnose tool-use hallucinations, ToolBH (Zhang et al., 2024) collects 700 tool-call samples to perform solvability detection, solution planning, and missing-tool analysis. Different from prior works confined to binary judgment in single-turn responses, we introduce the first benchmark for automated hallucination attribution within multi-step agent trajectories.

2.2 LLM-based Agents

LLM-based agents (Yao et al., 2023; Wang et al., 2024) have showcased extraordinary capabilities in automating tasks across various fields. This growing capability is largely driven by emergent behav-

Table 1: Comparison of AgentHallu with existing hallucination detection datasets in terms of dataset statistics (sample size (**#Samp.**) and trajectory steps (**#Step**)), hallucination categories (planning hallucination (**Planning**), retrieval hallucination (**Retrieval**), reasoning hallucination (**Reasoning**), human-interaction hallucination (**Human**), and tool-use hallucination (**Tool**)), and task type (Hallucination **Judgment** and Hallucination **Attribution**).

Dataset	Dataset Statistic		Hallucination Category					Task Type	
	#Samp.	#Step	Planning	Retrieval	Reasoning	Human	Tool	Judgment	Attribution
HaluEval (Li et al., 2023b)	35,000	1	✗	✗	✓	✗	✗	✓	✗
FELM (Chen et al., 2023)	847	1	✗	✗	✓	✗	✗	✓	✗
SAC^3 (Zhang et al., 2023)	500	1	✗	✗	✓	✗	✗	✓	✗
FAVABench (Mishra et al., 2024)	902	1	✗	✗	✓	✗	✗	✓	✗
RAGTruth (Niu et al., 2024)	2,965	1	✗	✓	✗	✗	✗	✓	✗
ToolBH (Zhang et al., 2024)	700	1	✗	✗	✗	✗	✓	✓	✗
AgentHallu (Ours)	693	7.6	✓	✓	✓	✓	✓	✓	✓

iors that arise during chain-of-thought (Wei et al., 2022), in-context learning (Brown et al., 2020), and instruction following (Longpre et al., 2023). To extend agents' capabilities beyond their internal knowledge, function calling (Schick et al., 2023; Patil et al., 2024, 2025) has been proposed, enabling agents to interact with external tools and APIs in multi-step workflows.

Moreover, individual agents, each serving a specialized role, can be composed into multi-agent systems to solve complex tasks (Hong et al., 2024; Wang et al., 2025). Early multi-agents elicit stepwise reasoning through structured debate (Liang et al., 2024) or role-play dialogue (Li et al., 2023a). Recent works (Fourney et al., 2024; Hu et al., 2025) introduce central orchestrators that assign tasks to specialized agents. Beyond inter-agent interaction, agents are motivated to proactively seek human feedback to improve their decision-making (Feng et al., 2024). Despite this progress, hallucinations persist across operational stages in agent workflows (Lin et al., 2025), including planning, retrieval, reasoning, tool use, and human interaction, underscoring the need for reliability assessment.

3 Task Formulation

In this section, we formulate the task of automated hallucination judgment and attribution.

Background. LLM-based agents perform complex tasks with structured reasoning, where each interaction unit u_t interleaves a thought step c_t, an action step a_t, and an observation step o_t. The trajectory τ can be written as:

$$\tau = (u_1, u_2, \ldots, u_t), \;\; u_t = (c_t, a_t, o_t), \quad (1)$$

where c_t denotes the internal reasoning state, a_t specifies the invoked tool action, and o_t captures feedback from the tool responses.

Hallucination Judgment Objective. We classify a trajectory as hallucinated by determining whether its produced answer diverges from the ground-truth solution corresponding to the task:

$$\text{is_hallucination}(\tau) = \mathbb{1}_{\{y(\tau) \neq y^{\text{gt}}\}}, \quad (2)$$

where $y(\tau)$ denotes the result of a trajectory τ, y^{gt} denotes the task-specific ground-truth answer, and $\mathbb{1}_{\{\cdot\}}$ is the indicator function.

Hallucination Attribution Objective. Motivated by (Zhang et al., 2025b), we identify a hallucinated step u_t as the step whose correction increases the likelihood of producing the correct result. Specifically, we replace u_t with its correct counterpart and roll out subsequent steps to obtain counterfactual trajectories $\tau^{(t)}$. The set of hallucinated steps $\mathcal{H}(\tau)$ of a trajectory τ is defined as:

$$\mathcal{H}(\tau) = \{\, t \mid y(\tau) \neq y^{\text{gt}} \wedge \mathbb{E}\big[y(\tau^{(t)}) = y^{\text{gt}}\big] > \delta \,\}. \quad (3)$$

where $y(\tau^{(t)})$ denotes the result produced by the counterfactual trajectory $\tau^{(t)}$, and δ is the probability threshold for successful correction. To address scenarios with multiple hallucinated steps, we follow a **causality-aligned principle** and treat the initial error as the primary source of hallucination. We thus define an objective to determine:

$$t^{\star} = \arg\min_{t \in \mathcal{H}(\tau)} t. \quad (4)$$

In this study, we address the problem of automatically identifying the step $t^{\star}$ and providing an associated open-ended explanation.

4 AgentHallu Dataset

In this section, we first present an overview of our AgentHallu dataset in Sec. 4.1. Then, we detail the dataset development involving query collection in Sec. 4.2, trajectory construction in Sec. 4.3, and hallucination annotation in Sec. 4.4.

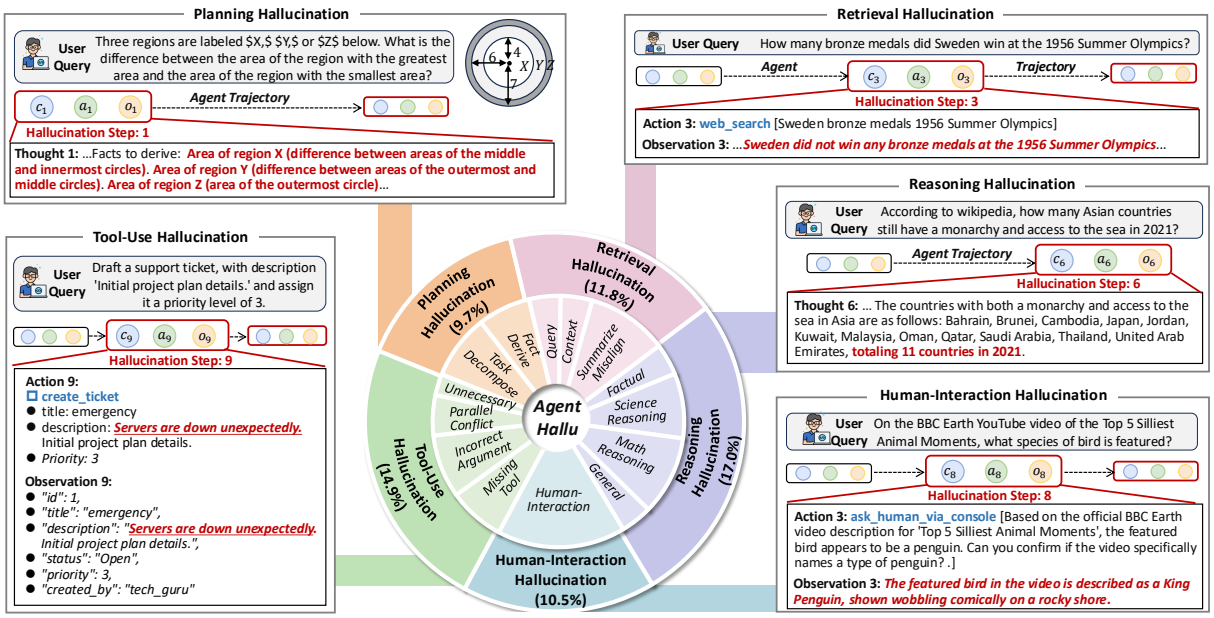

Figure 2: Overview of hallucination taxonomy in AgentHallu. The dataset includes 5 hallucination categories and 14 subcategories, where each trajectory step interleaves a thought step, an action step, and an observation step.

4.1 Overview

As shown in Table 1 and Figure 2, AgentHallu comprises 693 annotated agent trajectories, including 443 hallucinated instances and 250 non-hallucinated instances. Each instance in AgentHallu includes the following entries: **(1) Query:** A real-world question from 8 datasets, covering domains spanning world knowledge, science, math, general assistant, and tool use. **(2) Trajectory:** A trajectory generated to address the query, with each step standardized into a triplet of thought, action, and observation. These trajectories are collected from 7 mainstream LLM-based agents. **(3) Annotation:** Multi-level annotations curated by human labelers, comprising a binary label, a hallucination-responsible step, and a causal explanation. Detailed dataset statistics are provided in Appendix A.2.

4.2 Query Collection

To ensure comprehensive coverage of factuality, we curate a diverse set of queries spanning five realistic domains, as detailed below.

- *World Knowledge*: We incorporate queries from the SimpleQA dataset (Wei et al., 2024a), spanning ten topics such as politics, art, and sports to represent general world knowledge.
- *Science*: We include graduate-level scientific queries from the GPQA dataset (Rein et al., 2024), involving the disciplines of physics, chemistry, and biology.
- *Math*: We filter out difficulty Level 1 and Level 2 questions from MATH-500 (Hendrycks et al., 2021), retaining the harder subset. To integrate frontier-level challenges, we also include questions from the American Invitational Mathematics Examination (AIME) 2024 and AIME 2025.
- *General Assistant*: We include queries from the GAIA validation set (Mialon et al., 2024), which provides diverse and realistic instructions reflecting general assistant use.
- *Tool Use*: To mimic complex tool-use sequences in agentic workflows, we incorporate multi-turn and multi-step function-calling queries from BFCL V3 (Patil et al., 2025).

To extend coverage toward cutting-edge human knowledge, we also include a small subset of questions from HLE (Phan et al., 2025), spanning mathematics, humanities, and the natural sciences.

4.3 Trajectory Construction

Given the collected queries, we generate diverse and realistic trajectories by executing 7 widely used LLM-based agents (SmolAgents (Roucher et al., 2025), OpenDeepSearch (Alzubi et al., 2025), OpenManus (Liang et al., 2025), OctoTools (Lu et al., 2025), Magentic-One (Fourney et al., 2024), OWL (Hu et al., 2025), and Function-calling Agents (Patil et al., 2025)). Specifically, we partition queries from the four knowledge-intensive domains into six subsets and instantiate trajectories using the first six agents. In parallel, we utilize BFCL V3 to construct function-calling agent trajec-

Table 2: Taxonomy of agent hallucinations.

Category	Sub-category	Description
Planning Hallucination	Fact Derive	Introduce nonexistent or misleading facts.
	Task Decompose	Produce task-misaligned subgoals.
Retrieval Hallucination	Query Misalign	Formulate inaccurate retrieval queries.
	Context Misalign	Retrieve factually incorrect context.
	Summarize Misalign	Misrepresent context via summarization.
Reasoning Hallucination	Factual Reasoning	Incorrect factual inference over context.
	Math Reasoning	Incorrect math inference or computation.
	Science Reasoning	Incorrect science inference or computation.
	General Reasoning	Incorrect reasoning over general tasks.
Human-Interaction Hallucination		Incorrect messages propagated by the user.
Tool-Use Hallucination	Missing Tool	Miss required tool invocation.
	Incorrect Argument	Mis-specify arguments of invoked tools.
	Parallel Conflict	Trigger execution conflicts via parallel tools.
	Unnecessary Tool	Invoke irrelevant or incorrect tools.

tories. These agents are primarily built on the GPT series (GPT-4o and GPT-4.1). Details on agent configuration are provided in Appendix A.3.1.

To enhance the robustness and quality of our benchmark, we apply a three-stage filtering criterion to the collected trajectories:

- *(1) Exclude Failure Trajectories*: Since non-deceptive failures are easy to identify, we manually exclude trajectories that terminate without a task-completing response (e.g., early termination due to turn limits, token overflows, or tool-permission restrictions).
- *(2) Exclude Short Trajectories*: Excessively short agent trajectories degrade into native LLM responses, lacking sufficient reasoning depth for step localization. Therefore, we exclude trajectories with only one or two valid steps.
- *(3) Exclude Trivial Trajectories*: To select plausible and difficult samples, we retain trajectories with disagreement among LLM judges. Specifically, we use four independent LLMs (`GPT-5`, `Gemini-2.5-Pro`, `DeepSeek-V3.1`, and `Qwen3-32B`) to assign a binary label and a hallucination-responsible step for each trajectory. Then we exclude trajectories with full agreement across all four judges.

4.4 Hallucination Annotation

Through the multi-step filtering described above, we retain 693 agent trajectories. To ensure consistent and reproducible annotations across heterogeneous agent systems, we establish both an empirically grounded hallucination taxonomy and a standardized annotation protocol.

Empirically Grounded Taxonomy. To allow hallucination modes to emerge from empirical data, we apply grounded theory (Glaser and Strauss, 2017) to analyze a pilot set of 140 trajectories sampled from seven agent frameworks. Specifically, we first perform open coding (Khandkar, 2009) on the trajectory data to label observed hallucinated behaviors. Then we apply constant comparative analysis to refine the boundaries between different hallucination types. By merging and linking relevant behaviors, we organize the open codes into a structured taxonomy of hallucination categories. The taxonomy is finally refined through discussion and review until consensus is reached. The resulting taxonomy is presented in Table 2.

Table 3: Initial inter-annotator agreement on binary judgment (**Judgment**), categorization (**Category**), and hallucination-responsible step (**Step**). The results highlight the difficulty of manual hallucination attribution.

Annotation	World.	Science	Math	General	Tool	Overall
Judgment	98.4	98.0	100.0	100.0	98.8	98.9
Category	83.2	74.3	81.2	74.4	92.2	81.9
Step	80.4	72.5	76.5	69.2	85.4	77.9

Standardized Annotation Protocol. We introduce a hallucination annotation protocol, which progresses from binary judgment to fine-grained attribution and taxonomy classification. To ensure annotation rigor, we employ ten graduate-level annotators with specialized expertise in AI to perform iterative labeling and refinement.

- *(1) Construction of Oracle-guided Reasoning Paths*. Considering that hallucination attribution often requires domain-specific expertise, we leverage LLMs to construct detailed reasoning paths for question solving. Specifically, we condition the LLM on the question, the ground-truth answer, and, when available, dataset-provided solution annotations. Compared with question-only prompting, providing this additional information yields more faithful reasoning paths. To mitigate model-specific bias, each path is independently drafted by two different LLMs, GPT-5-Thinking and Gemini-2.5-Pro.
- *(2) Human Annotation*. Annotators first make a binary judgment of whether the agent trajectory is hallucinated by comparing it with the ground truth. For hallucinated cases, they further annotate the category, hallucination-responsible step, and causal explanation. To facilitate this process, LLMs are prompted with the reasoning path to generate attribution references, which are subsequently verified by annotators. Validation relies on two criteria: whether the candidate step introduces a factual error that directly distorts the outcome, or whether it propagates an error seeded in an earlier step. Upon detecting such propaga-

Table 4: Performance (%) of LLMs on AgentHallu under standard prompting, reporting hallucination judgment (F1/Recall/Acc) and hallucination attribution measured by step localization accuracy (Acc) and G-EVAL (GE).

Model Name	Judgment			Attribution											
	Overall			Planning		Retrieval		Reasoning		Human		Tool-Use		Overall	
	F1↑	Recall↑	Acc↑	Acc	GE	Acc	GE	Acc	GE	Acc	GE	Acc	GE	Acc↑	GE↑
Random	48.5	49.6	49.5	9.6	-	8.8	-	10.3	-	7.4	-	7.2	-	8.7	-
Proprietary Large Language Models															
GPT-5	**70.2**	**73.2**	**70.6**	**31.3**	**2.3**	26.8	1.7	57.6	3.0	39.7	2.6	4.9	0.6	32.7	2.0
GPT-5-mini	65.0	67.3	65.5	29.9	2.1	28.1	1.5	53.4	2.8	**61.6**	**3.3**	3.9	0.6	35.0	2.0
Gemini-2.5-Pro	64.6	64.2	68.8	25.4	2.2	**45.1**	**2.3**	**64.4**	**3.2**	50.7	2.8	14.6	1.3	**41.1**	**2.4**
Gemini-2.5-Flash	65.3	65.4	67.7	20.9	2.1	42.7	2.1	54.2	2.7	43.8	2.6	15.5	1.3	36.3	2.1
Claude-4.5-Sonnet	63.6	63.7	66.1	26.9	**2.3**	30.5	1.7	44.9	2.3	43.8	2.4	**19.4**	**1.4**	33.4	2.0
Average	65.7	66.8	67.7	26.9	2.2	34.6	1.9	54.9	2.8	47.9	2.7	11.6	1.0	35.7	2.1
Open-source Large Language Models															
DeepSeek-V3.1	**52.1**	52.1	**55.4**	**14.9**	**1.8**	22.0	**1.6**	27.1	**1.8**	21.9	1.9	7.8	**0.7**	19.0	**1.5**
Qwen3-32B	51.8	53.0	52.7	7.5	1.5	19.5	1.1	**28.8**	1.7	**41.1**	**2.1**	**8.7**	0.5	**21.2**	1.3
Qwen3-8B	49.5	54.2	49.5	4.5	1.2	**28.1**	1.3	17.0	0.9	23.3	1.2	3.9	0.2	15.1	0.9
Qwen2.5-72B	44.3	55.2	46.0	4.5	0.8	3.7	0.3	9.3	0.6	13.7	0.7	6.8	0.5	7.7	0.6
Qwen2.5-32B	49.3	**56.3**	49.6	4.5	1.1	1.2	0.5	15.3	0.9	12.3	0.8	6.8	0.6	8.6	0.8
Qwen2.5-7B	43.9	51.1	44.2	0.0	0.6	6.1	0.5	5.9	0.4	13.7	0.8	6.8	0.5	6.6	0.5
Llama3.3-70B	40.4	54.2	43.6	10.5	0.9	4.9	0.4	6.8	0.3	5.5	0.3	**8.7**	0.5	7.2	0.4
Llama3.1-8B	35.1	52.1	40.3	0.0	0.3	2.4	0.3	0.9	0.2	4.1	0.2	1.0	0.1	1.6	0.2
Average	45.8	53.5	47.7	5.8	1.0	11.0	0.8	13.9	0.8	17.0	1.0	6.3	0.4	10.9	0.8

tion, annotators trace the error chain backward and reassign attribution to the root cause.

- *(3) Consensus Resolution*. Inter-annotator agreement statistics are reported in Table 3. For disagreements, annotations are resolved through collaborative discussion, requiring all annotators to be convinced by the final rationale. For agreed cases, we employ a cross-validation protocol in which each annotator reviews peer annotations to ensure adherence to shared standards. Any detected inconsistency triggers discussion and re-annotation until consensus is achieved.

5 Experiments

5.1 Experimental Setup

Evaluated Models. We evaluate 13 frontier proprietary and open-source LLMs. The proprietary models include OpenAI's GPT-5 and GPT-5-mini; Google's Gemini-2.5-Pro and Gemini-2.5-Flash; and Anthropic's Claude-4.5-Sonnet. The open-source models include DeepSeek's DeepSeek-V3.1; Alibaba's Qwen3 (8B/32B) and Qwen-2.5 (7B/32B/72B); and Meta's Llama-3.3-70B and Llama-3.1-8B.

Prompting Methods. We evaluate two baseline prompting methods: Standard Prompting and Step-by-Step Prompting. In Standard Prompting, the model receives the query and the full trajectory and is asked to perform hallucination judgment and attribution. In Step-by-Step Prompting, the model receives the query and the trajectory incrementally and determines at each step whether a hallucination occurs, terminating immediately upon detection. More details can be found in Appendix B.2.

Evaluated Metrics. For hallucination judgment, we evaluate binary classification performance using standard metrics, including macro-F1, macro-recall, and accuracy. For hallucination attribution, we report step localization accuracy, defined as the proportion of hallucinated instances for which the model correctly identifies the responsible step. In addition, we use G-EVAL (Liu et al., 2023) with GPT-5 as the evaluator to score explanation quality. More details can be found in Appendix B.3.

5.2 Main Results

Comparison of Different LLMs. Table 4 reports the main results of different LLMs on AgentHallu. Our key findings are summarized as follows:

(1) Challenges of Attribution: A substantial performance gap remains between hallucination judgment and attribution tasks. While advances in proprietary models have boosted judgment performance to a peak F1 of 70.2% for GPT-5, the more demanding attribution task reaches only 41.1% localization accuracy and a 2.4 G-EVAL score for Gemini-2.5-Pro. These results indicate consider-

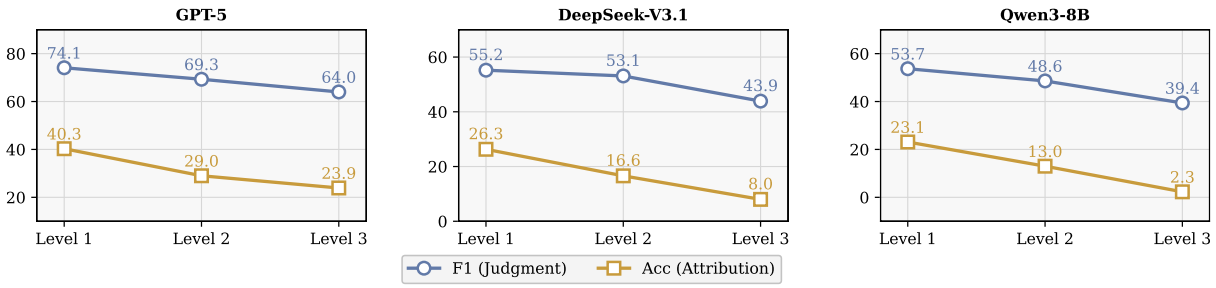

Figure 3: Comparison of hallucination judgment and attribution performance across LLMs under varying trajectory steps N_{step}. Level 1 spans trajectories with $N_{\text{step}} \leq 5$, Level 2 spans $6 \leq N_{\text{step}} \leq 10$, and Level 3 spans $N_{\text{step}} \geq 11$.

able room for attribution improvement and highlight the rigorous standards of this benchmark.

(2) Disparity between Proprietary and Open-source Models: Open-source models achieve an average localization accuracy of 10.9%, a level comparable to a random baseline and substantially below 35.9% of proprietary models. Even the strongest open-source model, DeepSeek-V3.1, attains only 19.2% localization accuracy. This performance gap may be attributed to the limited reasoning capabilities of open-source models.

(3) Category-level Analysis: Attribution accuracy varies substantially across hallucination categories. Reasoning hallucinations are comparatively easier to localize, with Gemini-2.5-Pro reaching 64.4% accuracy, whereas tool-induced hallucinations remain consistently the most difficult across all model families. This may be attributed to the challenge of verifying environmental state within action–observation loops, rather than purely linguistic factual errors. Further analysis on subcategories is provided in Appendix C.2.

Table 5: Comparison of prompting methods in terms of hallucination judgment (Judg.) F1, attribution (Attr.) step localization accuracy, and inference token cost.

Model Name	Prompting Method	Judg. F1↑	Attr. Acc↑	Efficiency Token Cost
GPT-5	Standard	**70.2**	32.7	7,426
	Step-by-Step	68.5	**42.7**	**25,454**
DeepSeek -V3.1	Standard	**52.1**	19.0	6,494
	Step-by-Step	51.1	**35.9**	**11,457**
Qwen3 -32B	Standard	51.8	21.2	5,017
	Step-by-Step	**52.2**	**31.2**	**15,630**

Comparison of Different Prompting Methods. We compare two prompting methods for hallucination judgment and attribution in Table 5. For hallucination judgment, the standard prompt remains consistently competitive and slightly outperforms the step-by-step variant, suggesting that binary decisions benefit from aggregating evidence over the full trajectory. In contrast, the step-by-step method substantially improves attribution, raising accuracy from 24.3% to 36.6% on average by incrementally processing context to enable more focused step localization. However, this method comes at a clear efficiency trade-off, increasing the average input token cost from 6,312 to 17,514 due to the additional decisions with multi-turn prompting.

Performance across Varying Trajectory Steps. To further examine the effect of trajectory steps on hallucination diagnosis, we partition the trajectory logs from AgentHallu into three levels based on the number of steps N_{step}. Level 1 includes trajectories with $N_{\text{step}} \leq 5$, Level 2 covers $6 \leq N_{\text{step}} \leq 10$, and Level 3 contains $N_{\text{step}} \geq 11$, resulting in 278, 274, and 141 samples, respectively. Both judgment and attribution performances for three LLMs across these levels are presented in Figure 3. The results show a consistent degradation in both tasks as trajectory length increases. Notably, attribution accuracy drops significantly on average, from 29.9% at Level 1 to 11.4% at Level 3, suggesting that the accumulation of distracting context can effectively obscure the hallucination-responsible step.

Table 6: Spearman and Kendall-Tau correlations between different metrics and human annotations.

Evaluation Metric	Spearman	Kendall-Tau
Rouge-L	0.44	0.34
BERTScore	0.32	0.24
G-EVAL-Qwen3-32B	0.78	0.62
G-EVAL-GPT5	**0.86**	**0.76**

Human Evaluation on Explanations. To examine alignment between causal explanation evaluation and human preference, we conduct a user study on 100 curated trajectory–explanation pairs from AgentHallu. The pairs are uniformly sampled across five hallucination categories and

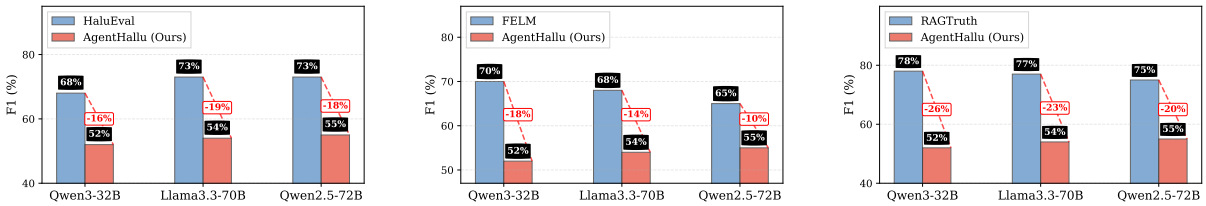

Figure 4: Comparison of hallucination judgment F1 (%) on AgentHallu against existing hallucination detection datasets across multiple LLMs.

span outputs from five models, including GPT-5, Gemini-2.5-Pro, DeepSeek-V3.1, Qwen3-32B, and Llama3.3-70B. Three annotators with AI expertise independently rate each pair on a five-point scale. In Table 6, LLM-based evaluator, G-EVAL, align more closely with human judgments than Rouge-L (Lin, 2004) and BERTScore (Zhang et al., 2020). In particular, GPT-5-based G-EVAL achieves 0.86 Spearman and 0.76 Kendall-Tau, indicating reliable assessment of hallucination explanations.

5.3 Experimental Analysis

Comparison against Hallucination Detection Datasets. To underscore the challenge of AgentHallu, we compare it against three existing hallucination detection datasets, HaluEval (Li et al., 2023b), FELM (Chen et al., 2023), and RAGTruth (Niu et al., 2024), using three advanced LLMs. As shown in Figure 4, all models consistently yield substantially lower performance on AgentHallu than on prior datasets, with an average degradation of about 18.2% binary F1. This consistent difficulty stems from the long-horizon nature of multi-step trajectories and the broader coverage of hallucination categories in AgentHallu.

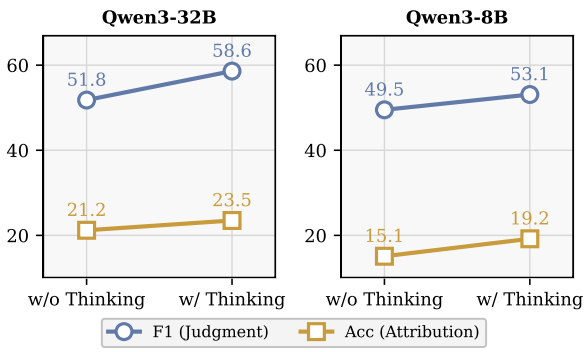

Figure 5: The performance of judgment and attribution with and without thinking mode on Qwen3.

Effect of Thinking Mode. We further study the effect of enabling thinking mode on automated hallucination judgment and attribution. Figure 5 shows consistent gains for both Qwen3 variants when thinking is enabled. For Qwen3-32B, judgment F1 improves from 51.8 to 58.6, while attribution accuracy increases from 21.2 to 23.5. The improvements are primarily attributable to enhanced self-verification under thinking mode, which better distinguishes plausible yet incorrect claims.

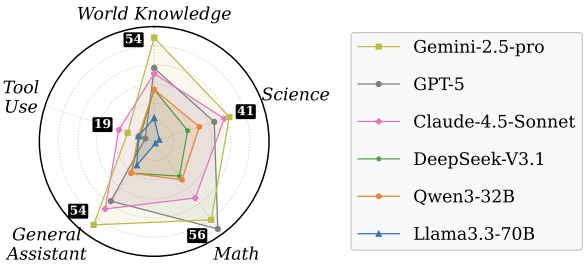

Figure 6: Step localization accuracy of six evaluated LLMs across five domains.

Performance across Domains. We finally report step localization accuracy across five domains in Figure 6. The results show that attribution remains challenging across all models and varies substantially by domain. For knowledge-intensive queries, performance peaks on Math at 56% accuracy but drops notably on Science to 41% accuracy. Tool Use is consistently the hardest, suggesting current LLMs are challenging to precisely track environment states under sequential tool interactions.

6 Conclusion

In this paper, we propose a novel task of automated hallucination attribution of LLM-based agents, aiming to identify the step where the initial hallucination originates and explain why it occurs. To advance this task, we present AgentHallu, a comprehensive benchmark comprising 693 high-quality trajectories featuring: (1) extensive diversity spanning 7 agent frameworks and 5 domains, (2) systematic coverage of 5 hallucination categories and 14 subcategories, and (3) multi-level human annotations of binary labels, hallucination-responsible steps and causal explanations. Evaluations on 13 leading LLMs highlight significant challenges, with performance varying across hallucination categories, prompting methods and trajectory lengths.

Limitations

While our AgentHallu marks a critical advancement in automated hallucination attribution for LLM-based agents, it is important to recognize several limitations. First, although AgentHallu spans 5 primary categories and 14 subcategories, it remains challenging to fully anticipate and represent emerging hallucination patterns as agent frameworks, tool ecosystems, and interaction protocols rapidly evolve. Therefore, the dataset should be continuously expanded to keep pace with new agent capabilities. Second, AgentHallu primarily targets text-based trajectories and does not consider multimodal agent settings grounded in images, audio, or other modalities. Given the growing adoption of multimodal agents, future work should explore extending the attribution framework to encompass these broader multimodal interactions.

Ethical Considerations

AgentHallu is strictly free of personally identifiable information and offensive content. The benchmark is exclusively sourced from publicly accessible datasets and repositories, as well agent trajectories generated under controlled settings, explicitly avoiding sensitive or restricted data sources. Designed for academic research, AgentHallu focuses on enhancing the reliability of autonomous agents. Through adherence to strict data integrity protocols and ethical standards, AgentHallu establishes a responsible foundation for the automated attribution of agent hallucinations.

Acknowledgments

This research is sponsored by National Natural Science Foundation of China (Grant No. 62306041) and Beijing Natural Science Foundation (L257008).

References

Salaheddin Alzubi, Creston Brooks, Purva Chiniya, Edoardo Contente, Chiara von Gerlach, Lucas Irwin, Yihan Jiang, Arda Kaz, Windsor Nguyen, Sewoong Oh, Himanshu Tyagi, and Pramod Viswanath. 2025. Open deep search: Democratizing search with open-source reasoning agents. arXiv preprint arXiv:2503.20201.

Yejin Bang, Ziwei Ji, Alan Schelten, Anthony Hartshorn, Tara Fowler, Cheng Zhang, Nicola Cancedda, and Pascale Fung. 2025. Hallulens: Llm hallucination benchmark. In ACL.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In NeurIPS.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya G. Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. 2025. Why do multi-agent llm systems fail? In NeurIPS.

Shiqi Chen, Yiran Zhao, Jinghan Zhang, I-Chun Chern, Siyang Gao, Pengfei Liu, and Junxian He. 2023. Felm: Benchmarking factuality evaluation of large language models. In NeurIPS.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261.

Xueyang Feng, Zhi-Yuan Chen, Yujia Qin, Yankai Lin, Xu Chen, Zhiyuan Liu, and Ji-Rong Wen. 2024. Large language model-based human-agent collaboration for complex task solving. In EMNLP Findings.

Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Erkang Zhu, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, Peter Chang, Ricky Loynd, Robert West, Victor Dibia, Ahmed Awadallah, Ece Kamar, Rafah Hosn, and Saleema Amershi. 2024. Magentic-one: A generalist multi-agent system for solving complex tasks. arXiv preprint arXiv:2411.04468.

Barney Glaser and Anselm Strauss. 2017. Discovery of grounded theory: Strategies for qualitative research. Routledge.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. In NeurIPS.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. Metagpt: Meta programming for a multi-agent collaborative framework. In ICLR.

Mengkang Hu, Yuhang Zhou, Wendong Fan, Yuzhou Nie, Ziyu Ye, Bowei Xia, Tao Sun, Zhaoxuan Jin, Yingru Li, Zeyu Zhang, Yifeng Wang, Qianshuo Ye, Bernard Ghanem, Ping Luo, and Guohao Li. 2025.

Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation. In NeurIPS.

Shahedul Huq Khandkar. 2009. Open coding. University of Calgary.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023a. Camel: Communicative agents for" mind" exploration of large language model society. In NeurIPS.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023b. Halueval: A large-scale hallucination evaluation benchmark for large language models. In EMNLP.

Qing Li, Jiahui Geng, Zongxiong Chen, Derui Zhu, Yuxia Wang, Congbo Ma, Chenyang Lyu, and Fakhri Karray. 2025. Hd-ndes: Neural differential equations for hallucination detection in llms. In ACL.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In EMNLP.

Xinbin Liang, Jinyu Xiang, Zhaoyang Yu, Jiayi Zhang, Sirui Hong, Sheng Fan, and Xiao Tang. 2025. Openmanus: An open-source framework for building general ai agents.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Text summarization branches out.

Xixun Lin, Yucheng Ning, Jingwen Zhang, Yan Dong, Yilong Liu, Yongxuan Wu, Xiaohua Qi, Nan Sun, Yanmin Shang, Pengfei Cao, Lixin Zou, Xu Chen, Chuan Zhou, Jia Wu, Shirui Pan, Bin Wang, Yanan Cao, Kai Chen, Songlin Hu, and Li Guo. 2025. Llm-based agents suffer from hallucinations: A survey of taxonomy, methods, and directions. arXiv preprint arXiv:2509.18970.

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. In EMNLP.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The flan collection: Designing data and methods for effective instruction tuning. In ICML.

Pan Lu, Bowen Chen, Sheng Liu, Rahul Thapa, Joseph Boen, and James Zou. 2025. Octotools: An agentic framework with extensible tools for complex reasoning. arXiv preprint arXiv:2502.11271.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2024. Gaia: a benchmark for general ai assistants. In ICLR.

Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. 2024. Fine-grained hallucination detection and editing for language models. In COLM.

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. Ragtruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In ACL.

OpenAI. 2025. Gpt-5. https://openai.com/zh-Hans-CN/index/introducing-gpt-5.

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E Gonzalez. 2025. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In ICML.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. 2024. Gorilla: Large language model connected with massive apis. In NeurIPS.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, Michael Choi, Anish Agrawal, Arnav Chopra, Adam Khoja, Ryan Kim, Richard Ren, Jason Hausenloy, Oliver Zhang, Mantas Mazeika, and 3 others. 2025. Humanity's last exam. arXiv preprint arXiv:2501.14249.

Chengwei Qin, Wenxuan Zhou, Karthik Abinav Sankararaman, Nanshu Wang, Tengyu Xu, Alexander Radovic, Eryk Helenowski, Arya Talebzadeh, Aditya Tayade, Sinong Wang, Shafiq Joty, Han Fang, and Hao Ma. 2025. Learning auxiliary tasks improves reference-free hallucination detection in open-domain long-form generation. In ACL.

Alfin Wijaya Rahardja, Junwei Liu, Weitong Chen, Zhenpeng Chen, and Yiling Lou. 2025. Can agents fix agent issues? In NeurIPS.

Abhilasha Ravichander, Shrusti Ghela, David Wadden, and Yejin Choi. 2025. Halogen: Fantastic llm hallucinations and where to find them. In ACL.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In COLM.

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

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. In NeurIPS.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. 2024. Executable code actions elicit better llm agents. In ICML.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang H. Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Yanjun Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, and 2 others. 2025. Openhands: An open platform for ai software developers as generalist agents. In ICLR.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024a. Measuring short-form factuality in large language models. arXiv preprint arXiv:2411.04368.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In NeurIPS.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. 2024b. Long-form factuality in large language models. In NeurIPS.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. Swe-agent: Agent-computer interfaces enable automated software engineering. In NeurIPS.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In ICLR.

Fujie Zhang, Peiqi Yu, Biao Yi, Baolei Zhang, Tong Li, and Zheli Liu. 2025a. Prompt-guided internal states for hallucination detection of large language models. In ACL.

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley Malin, and Sricharan Kumar. 2023. Sac3: reliable hallucination detection in black-box language models via semantic-aware cross-check consistency. In EMNLP Findings.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, and Qingyun Wu. 2025b. Which agent causes task failures and when? on automated failure attribution of llm multi-agent systems. In ICML.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In ICLR.

Yuxiang Zhang, Jing Chen, Junjie Wang, Yaxin Liu, Cheng Yang, Chufan Shi, Xinyu Zhu, Zihao Lin, Hanwen Wan, Yujiu Yang, Tetsuya Sakai, Tian Feng, and Hayato Yamana. 2024. Toolbehonest: A multi-level hallucination diagnostic benchmark for tool-augmented large language models. In EMNLP.

Zhenliang Zhang, Xinyu Hu, Huixuan Zhang, Junzhe Zhang, and Xiaojun Wan. 2025c. Icr probe: Tracking hidden state dynamics for reliable hallucination detection in llms. In ACL.

Jialong Zhou, Lichao Wang, and Xiao Yang. 2025. Guardian: Safeguarding llm multi-agent collaborations with temporal graph modeling. In NeurIPS.

Appendix

Contents

1 Introduction **1**

2 Related Work **2**
- 2.1 Hallucination Detection Benchmarks 2
- 2.2 LLM-based Agents 2

3 Task Formulation **3**

4 AgentHallu Dataset **3**
- 4.1 Overview 4
- 4.2 Query Collection 4
- 4.3 Trajectory Construction 4
- 4.4 Hallucination Annotation 5

5 Experiments **6**
- 5.1 Experimental Setup 6
- 5.2 Main Results 6
- 5.3 Experimental Analysis 8

6 Conclusion **8**

A More Details on AgentHallu **13**
- A.1 Dataset and Code Release 13
- A.2 Dataset Statistics 13
  - A.2.1 Category Statistics 13
  - A.2.2 Word Cloud 13
  - A.2.3 Trajectory Distribution . . 13
- A.3 Agent Configuration 13
  - A.3.1 Agent Description 13
  - A.3.2 Agent Distribution across Datasets 14
  - A.3.3 Agent Distribution across Models 14
- A.4 Source Dataset Licenses 15

B More Details on Evaluation **15**
- B.1 Model Configurations 15
- B.2 Prompting Method 15
- B.3 Evaluated Metric 16

C Additional Experiments **17**
- C.1 Model Bias Analysis in Explanation Evaluation. 17
- C.2 More Analysis on Subcategories . 17

D More Details on Prompt Templates **18**
- D.1 Templates for Question-solving Paths 18
- D.2 Templates for Standard Prompting 18
- D.3 Templates for Step-by-Step Prompting 18
- D.4 Templates for G-EVAL Evaluation 18

E Case Study **18**

F Broader Impact **18**

A More Details on AgentHallu

A.1 Dataset and Code Release

The dataset and code are distributed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0). This license permits sharing and adaptation with appropriate attribution for non-commercial use, provided that derivative works are distributed under the same terms.

A.2 Dataset Statistics

A.2.1 Category Statistics

AgentHallu includes 693 trajectories across one non-hallucination category and five hallucination categories and fourteen subcategories, with distribution statistics presented in Table 7. As shown in Table 7, the category distribution is intentionally balanced to ensure comprehensive and equitable coverage of diverse hallucination types.

Table 7: The statistics of AgentHallu over one non-hallucination category and five hallucination categories and fourteen sub categories.

Category	Samples	Ratio(%)
Non-Hallucination	**250**	**36.1**
Planning Hallucination	**67**	**9.7**
• Fact Derive	37	5.3
• Task Decompose	30	4.3
Retrieval Hallucination	**82**	**11.8**
• Query Misalign	18	2.6
• Context Misalign	18	2.6
• Summarize Misalign	46	4.6
Reasoning Hallucination	**118**	**17.0**
• Factual Reasoning	18	2.6
• Science Reasoning	37	5.3
• Math Reasoning	45	6.5
• General Reasoning	18	2.6
Human-Interaction Hallucination	**73**	**10.5**
Tool-Use Hallucination	**103**	**14.9**
• Missing Required Call	32	4.6
• Incorrect Tool Arguments	36	5.3
• Unnecessary Tool Call	23	3.3
• Parallel Call Conflict	12	1.6

A.2.2 Word Cloud

The query distribution in AgentHallu is visualized as a word cloud in Figure 7. Prevalent high-frequency terms highlight recurring linguistic patterns in agent queries, including tool invocations and information-seeking requests. The breadth of salient keywords indicates substantial topical diversity, suggesting that the benchmark covers a wide range of realistic user intents rather than a narrow set of prompt templates.

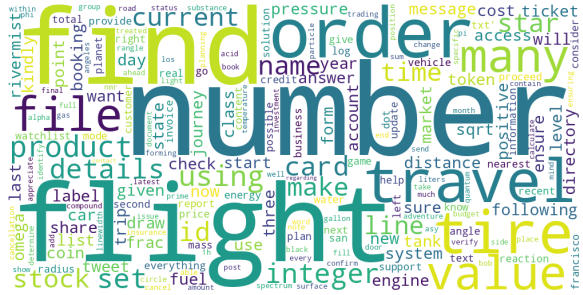

Figure 7: Word cloud of queries in AgentHallu dataset.

A.2.3 Trajectory Distribution

We present the trajectory length distribution of AgentHallu, measured by the number of steps per trajectory in Figure 8. As shown in Figure 8, AgentHallu excludes overly short trajectories with one or two steps. The longest trajectory contains 43 steps, and lengths are broadly distributed across step counts, indicating substantial diversity in interaction depth and long-horizon reasoning.

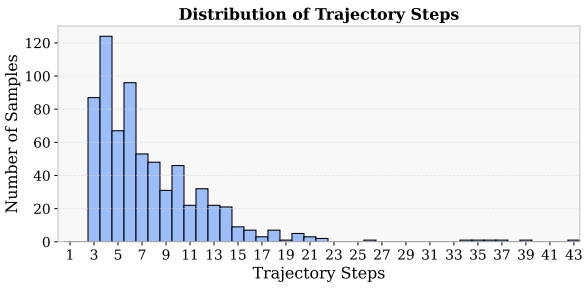

Figure 8: Distribution of trajectory lengths measured by the number of steps per trajectory.

A.3 Agent Configuration

A.3.1 Agent Description

We instantiate agent trajectories using seven representative LLM-based agents that span diverse reasoning paradigms and interaction patterns. We briefly summarize each framework below.

- *SmolAgents* (Roucher et al., 2025): SmolAgents is a lightweight agent framework that supports both CodeAct-style (Wang et al., 2024) and ReAct-style (Yao et al., 2023) agents, and we configure both agent types within our method.
- *OpenDeepSearch* (Alzubi et al., 2025): Built on SmolAgents as the reasoning agent, OpenDeepSearch integrates an advanced search tool that leverages an embedded LLM to refine retrieved context. We also configure both CodeAct-style and ReAct-style agent variants.
- *OpenManus* (Liang et al., 2025): OpenManus

Table 8: Distribution of query sources across agent frameworks in AgentHallu.

Agent Framework	SimpleQA	GPQA	MATH-500	AIME2024	AIME2025	GAIA	HLE	BFCL V3	Total
SmolAgents	27	20	21	1	2	14	6	-	91
OpenDeepSearch	55	9	13	3	3	10	6	-	100
OpenManus	42	36	8	7	7	4	0	-	104
Octotools	14	11	13	0	0	5	4	-	47
Magentic-One	28	25	8	3	5	20	5	-	94
OWL	17	36	13	2	7	15	3	-	93
Function-Calling Agent	-	-	-	-	-	-	-	164	164
Total	183	137	76	16	24	68	25	-	693

extends the planner–toolcaller architecture by incorporating explicit human-in-the-loop interactions, enabling the agent to solicit user input and integrate feedback during task execution.

- *OctoTools* (Lu et al., 2025): OctoTools provides over ten standardized tool cards that encapsulate diverse functionalities, enabling efficient multi-tool workflows for complex computational tasks.
- *Magentic-One* (Fourney et al., 2024): Magnetic-One employs a coordinator agent that collaborates with four specialized agents: a WebSurfer agent to browse the web, a FileSurfer agent to handle files, a Coder agent to write code, and a Computer Terminal agent to execute code.
- *OWL* (Hu et al., 2025): OWL includes a workforce-oriented framework with a Planner for task decomposition, a Coordinator for subtask management, and specialized Workers capable of domain-specific tool invocation.
- *Function-calling Agent* (Patil et al., 2025): A function-calling agent conditions an LLM on a set of tool or API specifications. The agent then emits a structured function call that selects the appropriate function and fills in its arguments. The tool output is fed back to the agent, enabling multi-turn execution and iterative reasoning.

A.3.2 Agent Distribution across Datasets

We present the distribution of query dataset sources across agent frameworks in AgentHallu, as shown in Table 8. The six general agent frameworks contribute trajectories across all seven knowledge-intensive datasets, yielding broad and relatively balanced coverage. In contrast, the function-calling agent is used exclusively for BFCL V3 queries to assess tool selection and argument filling behavior. Overall, we obtain 693 trajectories spanning heterogeneous agent designs and data sources, suggesting that AgentHallu captures diverse execution patterns and task contexts rather than artifacts of a specific agent implementation.

Table 9: Distribution of LLM backbones across agent frameworks in AgentHallu.

Model Backbone	Samples	Ratio(%)
SmolAgents	**91**	**13.1**
• GPT-4.1	45	6.5
• GPT-4o	30	3.2
• Claude-3.7-Sonnet	5	0.7
• Qwen2.5-Coder-32B	19	2.7
OpenDeepSearch	**100**	**14.4**
• GPT-4.1	36	5.2
• GPT-4o	9	1.3
• Claude-3.7-Sonnet	19	2.7
• Qwen2.5-Coder-32B	36	5.2
OpenManus	**104**	**15.0**
• GPT-4.1	40	5.8
• GPT-4o	40	5.8
• GPT-5	17	2.5
• Claude-3.7-Sonnet	7	1.0
Octotools	**47**	**6.8**
• GPT-4.1	13	1.9
• GPT-4o	34	4.9
Magentic-One	**94**	**13.6**
• GPT-4.1	50	7.2
• GPT-4o	15	2.2
• GPT-5	21	3.0
• Claude-3.7-Sonnet	8	1.2
OWL	**93**	**13.4**
• GPT-4.1	72	10.4
• GPT-4o	4	0.6
• GPT-5	17	2.5
Function-calling Agent	**164**	**23.7**
• GPT-4.1	60	8.7
• Qwen3-32B	60	8.7
• Llama3.3-70B	44	6.3

A.3.3 Agent Distribution across Models

To enrich behavioral diversity and mitigate backbone-specific bias, we instantiate the six general agent frameworks with five LLM backbones (`GPT-5`, `GPT-4.1`, `GPT-4o`, `Claude-3.7-Sonnet`, and `Qwen2.5-Coder-32B`). For BFCL V3 queries, we incorporate trajectories generated by function-calling agents based on `GPT-4.1`, `Qwen3-32B`, and `Llama-3.3-70B`. We summarize the backbone composition per framework in Table 9. The results reflect that this heterogeneous backbone mix-

Table 10: Configuration details of LLMs used for trajectory generation and hallucination evaluation in AgentHallu.

Organization	Model	Release	Version	Inference Pipeline
Proprietary LLMs				
Google	Gemini 2.5 Pro	2025-6	`gemini-2.5-pro-06-17`	API
	Gemini 2.5 Flash	2025-6	`gemini-2.5-flash-06-17`	API
OpenAI	GPT-5	2025-8	`gpt-5-2025-08-07`	API
	GPT-5-mini	2025-8	`gpt-5-mini-2025-08-07`	API
	GPT-4.1	2025-4	`gpt-4.1-2025-04-14`	API
	GPT-4o	2024-12	`gpt-4o-2024-11-20`	API
Anthropic	Claude-4.5-Sonnet	2025-09	`claude-4-5-sonnet-20250929`	API
	Claude-3.7-Sonnet	2025-02	`claude-3-7-sonnet-20250219`	API
Open-source LLMs				
DeepSeek	DeepSeek-V3.1	2025-8	`deepseek-v3.1-250821`	API
Alibaba	Qwen3-32B	2025-4	`Qwen3-32B`	Transformers
	Qwen3-8B	2025-4	`Qwen3-8B`	Transformers
	Qwen2.5-72B	2024-9	`Qwen2.5-72B-Instruct`	Transformers
	Qwen2.5-32B	2024-9	`Qwen2.5-32B-Instruct`	Transformers
	Qwen2.5-7B	2024-9	`Qwen2.5-7B-Instruct`	Transformers
	Qwen2.5-Coder-32B	2024-11	`Qwen2.5-Coder-32B-Instruct`	Transformers
Meta	LLama3.3-70B	2024-9	`Llama-3.3-70B-Instruct`	Transformers
	LLama3.1-8B	2024-7	`Llama-3.1-8B-Instruct`	Transformers

ture broadens AgentHallu's behavioral diversity across frameworks and reduces reliance on any single model family, making the benchmark more representative for attribution evaluation.

A.4 Source Dataset Licenses

The licenses for the source query datasets used in this paper summarized are as follows:

- *SimpleQA* (Wei et al., 2024a): MIT License.
- *GPQA* (Rein et al., 2024): MIT License.
- *MATH-500* (Hendrycks et al., 2021): MIT License.
- *AIME 2024 and AIME 2025*: MIT License.
- *GAIA* (Mialon et al., 2024): The dataset does not specify an explicit license.
- *BFCL V3* (Patil et al., 2025): Apache-2.0 License.
- *HLE* (Phan et al., 2025): MIT License.

B More Details on Evaluation

B.1 Model Configurations

Table 10 summarizes the configurations of the LLM backbones used to generate agent trajectories and hallucination evaluation. For trajectory generation, we adopt the default LLM settings provided by each agent framework. For hallucination evaluation, to ensure fair comparisons, we fix the sampling hyperparameters by setting "do_sample = False" or "Temperature = 0" to guarantee deterministic outputs, with the maximum output length set to 1024 tokens. All experiments are performed on eight NVIDIA GeForce A100 GPUs with PyTorch and are fully reproducible.

Algorithm 1 Standard Prompting

Require: Query Q, trajectory $\tau = (u_1, \ldots, u_n)$, llm evaluator $\mathcal{M}_\theta$

Ensure: Hallucination label $h \in \{0, 1\}$, responsible step s^*, causal explanation e^*

1: $h \leftarrow 0$; $s^* \leftarrow \varnothing$; $e^* \leftarrow \varnothing$
2: $(h, s^*, e^*) \leftarrow \mathcal{M}_\theta(Q, \tau)$
3: **return** h, s^*, e^* ▷ $h = 0$ indicates non-hallucination

B.2 Prompting Method

We provide more details on two baseline prompting methods, described as follows:

- *Standard Prompting Method:* Standard prompting feeds the query and the complete trajectory to an evaluator model in a single pass. The model is instructed to determine whether the trajectory contains a hallucination and, if hallucinated, to

identify the earliest responsible step and provide a causal explanation linking that step. The algorithm of the standard prompting is summarized in Algorithm 1.

- *Step-by-Step Prompting Method:* Step-by-Step prompting evaluates the trajectory in an incremental manner. It presents the query and trajectory prefixes step by step, and at each step determines whether the current prefix already contains a hallucination. The procedure terminates upon the first hallucination identification, assigning that step as the responsible step, along with the causal explanation provided by the evaluator. The algorithm of the Step-by-Step prompting is summarized in Algorithm 2.

Algorithm 2 Step-by-Step Prompting

Require: Query Q, trajectory $\tau = (u_1, \ldots, u_n)$, llm evaluator $\mathcal{M}_\theta$
Ensure: Hallucination label $h \in \{0, 1\}$, responsible step s^*, causal explanation e^*

```
h ← 0; s* ← ∅; e* ← ∅
for i ∈ {1, 2, ..., n} do
    τ_≤i ← (u_1, ..., u_i)
    (h_i, e_i) ← M_θ(Q, τ_≤i)
    if h_i = 1 then
        h ← 1
        s* ← i
        e* ← e_i
        return h, s*, e*
    end if
end for
return h, s*, e*                ▷ h = 0 indicates non-hallucination
```

B.3 Evaluated Metric

Hallucination Judgment. For hallucination judgment, we adopt the widely used macro-F1 metric, which balances precision and recall through a harmonic mean. The macro-F1 score is computed as follows:

$$\text{macro-F1} = \frac{1}{K}\sum_{k=1}^{K}\frac{2 \times \text{Precision}_k \times \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k}. \tag{5}$$

In this context, K denotes the number of classes, and we set $K = 2$ for binary classification. Precision_k denotes the class-k precision, defined as the proportion of samples predicted as class k that truly belong to class k:

$$\text{Precision}_k = \frac{TP_k}{TP_k + FP_k}. \tag{6}$$

Recall_k is the recall for class k, defined as the proportion of samples from class k that are correctly identified:

$$\text{Recall}_k = \frac{TP_k}{TP_k + FN_k}. \tag{7}$$

Beyond the F1 score, we also include macro-recall and accuracy. Macro-recall is defined as follows:

$$\text{macro-Recall} = \frac{1}{K}\sum_{k=1}^{K}\text{Recall}_k. \tag{8}$$

The accuracy score is defined as follows:

$$\text{Accuray} = \frac{N_{\text{correct}}}{N_{\text{total}}}, \tag{9}$$

where N_{correct} is the number of correctly classified samples, and N_{total} is the total number of evaluated samples.

Hallucination Attribution. Since a decisive hallucination step is well-defined only for hallucinated outputs, we compute attribution metrics on the subset of hallucinated samples. This restriction prevents non-hallucinated cases from dominating the score and keeps the metric aligned with responsible-step localization. For step localization, We report localization accuracy, defined as the proportion of samples for which the predicted step matches the ground-truth hallucination annotation. The localization accuracy is computed as follows:

$$\text{Acc}_{\text{step}} = \frac{1}{|\mathcal{H}_{hal}|}\sum_{i \in \mathcal{H}_{hal}} \mathbb{1}\{\hat{t}_i = t_i^*\}, \tag{10}$$

where $\mathcal{H}_{hal}$ denotes the subset of hallucinated samples, t_i^* denotes the ground-truth hallucination-responsible step for sample i, $\hat{t}_i$ denotes the step predicted by the model, and $\mathbb{1}\{\cdot\}$ is the indicator function.

To further assess the quality of the causal explanations produced by each model, we adopt G-EVAL (Liu et al., 2023) and use GPT-5 as the evaluator. For each instance i, GPT-5 assigns an ordinal score $s_i \in 1, 2, 3, 4, 5$. The score is determined by a fixed rubric that measures the explanation accuracy, guided by the human-annotated explanation and the trajectory. The full prompt template and scoring rubric are provided in Appendix D.4.

C Additional Experiments

C.1 Model Bias Analysis in Explanation Evaluation.

To probe potential hidden evaluator bias, we examine whether a GPT-5–based judge favors explanations generated by GPT-5 itself. As shown in Figure 9, we report the average scores of causal explanations assigned by human annotators and by G-EVAL based on GPT-5. These explanations are generated by five representative models. The results show that G-EVAL scores are consistently close to human ratings across all evaluated models. Notably, GPT-5 explanations are not favored by the GPT-5 judge, with only a 0.15-point difference between human annotations and G-EVAL, comparable to the discrepancies observed for other models. In contrast, Gemini-2.5-Pro receives higher human scores than G-EVAL, suggesting that the GPT-5 judge is more conservative when assigning high scores to explanations with complex writing styles.

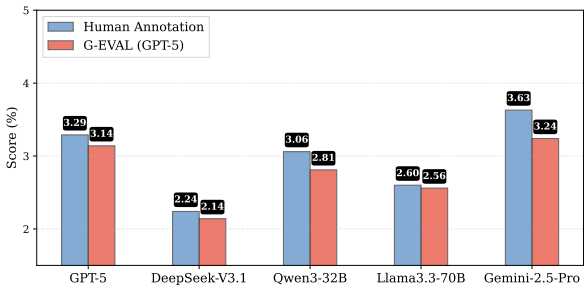

Figure 9: Comparison of average scores of causal explanation assigned by human annotators and the G-EVAL (GPT-5) judge across five models.

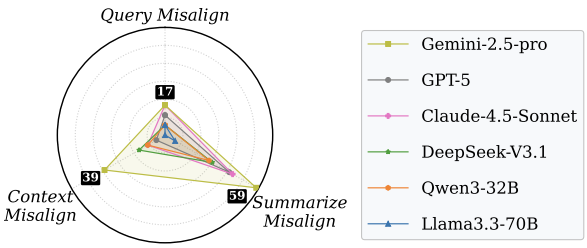

Figure 10: Step localization accuracy of six LLM judges across three retrieval hallucination subcategories.

C.2 More Analysis on Subcategories

We provide subcategory-level analysis for retrieval, reasoning, and tool-use hallucinations, described as follows:

- *Analysis on Retrieval Hallucination*. We report step localization accuracy for each retrieval hallucination subcategory in Figure 10. The results show that Gemini-2.5-Pro achieves the strongest attribution performance across the three retrieval-hallucination subcategories. Notably, Gemini-2.5-Pro shows a clear advantage in summarize misalign subcategory, indicating a superior ability to localize errors introduced during evidence aggregation and compression. In contrast, the query misalign subcategory remains the most challenging for all models, suggesting that hallucinations seeded by an incorrect retrieval intent are harder to diagnose and more likely to be confounded with later steps.
- *Analysis on Reasoning Hallucination*. We report step localization accuracy for each reasoning hallucination subcategory in Figure 11. Gemini-2.5-Pro consistently attains the highest accuracy across multiple hallucination reasoning subcategories. In contrast, open-source models perform substantially worse across all subcategories, indicating limited sensitivity to logical deviations. Overall, the results suggest that accurate reasoning-hallucination attribution requires fine-grained verification of intermediate claims and their dependencies, which remains a key bottleneck for current open-source models.
- *Analysis on Tool-Use Hallucination*. We report step localization accuracy for each tool-use hallucination subcategory in Figure 12. The results indicate that all models perform poorly on most subcategories, including incorrect ar-

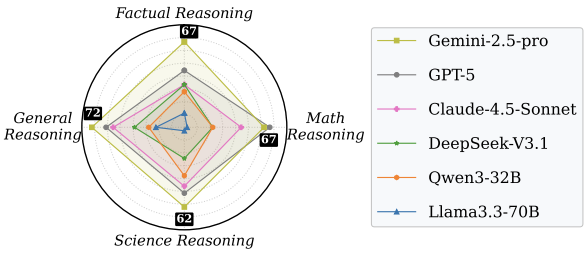

Figure 11: Step localization accuracy of six LLM judges across four reasoning hallucination subcategories.

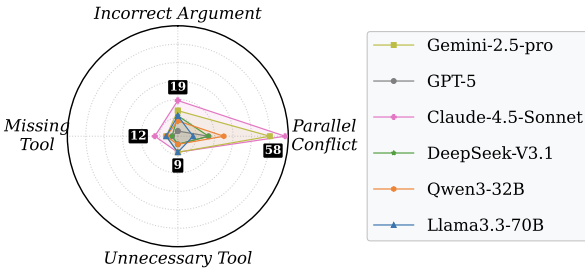

Figure 12: Step localization accuracy of six LLM judges across four tool-use hallucination subcategories.

gument, missing tool, and unnecessary tool. In contrast, Claude-4.5-Sonnet identifies parallel conflict hallucinations with notably higher accuracy, suggesting that inconsistencies from concurrent tool executions yield more explicit and verifiable contradictions.

D More Details on Prompt Templates

D.1 Templates for Question-solving Paths

Figure 13 illustrates the prompt template used to instruct LLMs to construct detailed question-solving paths based on the question, the ground-truth answer, and, when available, dataset-provided solution annotations.

D.2 Templates for Standard Prompting

Figure 14 shows the prompt template for standard prompting, where the LLM is provided with the query and the full trajectory and instructed to perform hallucination judgment and attribution.

D.3 Templates for Step-by-Step Prompting

Figure 15 illustrates the prompt template for step-by-step prompting, where the model processes the query and trajectory incrementally, determines whether a hallucination occurs at each step, and stops once a hallucination is detected.

D.4 Templates for G-EVAL Evaluation

Figure 16 illustrates the prompt template for G-EVAL evaluation. We evaluate each model's causal explanation, with reference to the trajectory and the human-annotated explanation. For each instance, G-EVAL assigns a five-point ordinal score under a fixed rubric that prioritizes explanation accuracy.

E Case Study

In this section, we provide qualitative case analysis of agent hallucination attribution in Figures 17, 18, 19, 20, 21. This analysis is essential for assessing both hallucination identification and the ability to explain where and why hallucinations arise in agentic workflows. To this end, we examine representative hallucination cases from five models, each illustrating a dominant hallucination pattern from one category: planning, retrieval, reasoning, human interaction, or tool use. For each case, we contrast the each model's predicted hallucination judgment, responsible step, and causal explanation with human annotations, and analyze where causal tracing breaks down.

F Broader Impact

AgentHallu aims to advance the reliability and transparency of LLM-based agents by enabling systematic hallucination diagnosis in multi-step workflows. By introducing a new task of automated hallucination attribution and providing a comprehensive benchmark with fine-grained annotations, AgentHallu enables researchers to better understand where and why hallucinations arise during agent execution. This capability is critical as LLM-based agents are increasingly deployed in high-stakes applications such as healthcare, finance, and decision support, where undetected error propagation can lead to severe downstream consequences.

We acknowledge the broader societal implications of releasing benchmarks for autonomous agents. AgentHallu is constructed exclusively from publicly available data and controlled agent executions, with all annotations carefully curated to avoid sensitive or harmful content. By emphasizing transparency, reproducibility, and ethical data practices, AgentHallu fosters responsible research and deployment of LLM-based agents, contributing to the long-term realization of reliable agentic systems.

Question-solving Path Prompting Template

You are an expert reasoning assistant. Based on the provided question, ground-truth answer, and solution guidance, your task is to produce a detailed, step-by-step solution that leads to the ground-truth answer. Restate the full question exactly as stated. Provide a clear, step-by-step reasoning process that shows how the correct answer is obtained, using the explanation as supporting evidence. Keep the logic explicit and the steps well organized.

<Question>**{question}**</Question>

<True Answer>**{true_answer}**</True Answer>

<Solution Guidance>**{solution_guidance}**</Solution Guidance>

Figure 13: Prompt template for constructing question-solving paths. The "question", "true_answer", and "solution_guidance" placeholders are replaced with the corresponding query, the ground-truth answer and the dataset-provided solution annotations.

Standard Prompting Template

Given a question and an agent trajectory that contains all reasoning and action steps taken by the agent to answer the question. Your task is to determine whether the agent's final answer contains hallucinated content that contradicts the real-world facts or the question's ground truth. If hallucinated, you need to further identify the first step that introduced the hallucination, and explain the cause.

<Question>**{question}**</Question>
<Agent Trajectory>**{trajectory}**</Agent Trajectory>

Based on the question and agent trajectory, please predict the following:
1. Examine the agent's answer in the given <Agent Trajectory> for the provided <Question>. If the answer contains hallucinated content, please set <is_hallucinated> to True, otherwise set <is_hallucinated> to False.
2. If <is_hallucinated> is True, identify the first step in the <Agent Trajectory> that introduces this hallucination and explain the cause. If <is_hallucinated> is False, set both hallucination_step and hallucination_reason to null.

ABSOLUTELY do not return your reasoning process. Return ONLY one JSON object that matches the format below and nothing else (in particular, never start your response with "JSON" or "Result"):
{
"is_hallucination": True/False: { "type": "boolean" },
"hallucination_step": Step Number { "type": ["integer", "null"], "minimum": 1 },
"hallucination_reason": Reason for Hallucination { "type": ["string", "null"], "maxLength": 1000},
}

Figure 14: Prompt template for automated hallucination judgment and attribution using standard prompting method. The "question" and "trajectory" placeholders are replaced with the corresponding query and agent trajectory to be evaluated.

Step-by-Step Prompting Template

Given a question and an agent trajectory up to the current reasoning and action steps taken by the agent to answer the question. Your task is to determine whether this most recent agent's step contains hallucinated content that could induce the question-solving process to produce an incorrect answer, and if so, explain the cause.

<Question>**{question}**</Question>
<Agent Trajectory>**{trajectory}**</Agent Trajectory>

Based on the question and agent trajectory, please predict the following:
1. Examine the agent's answer in the given <Agent Trajectory> for the provided <Question>. If the answer contains hallucinated content, please set <is_hallucinated> to True, otherwise set <is_hallucinated> to False.
2. If <is_hallucinated> is True, explain the reason that introduces or causes this hallucination. If <is_hallucinated> is False, set hallucination_reason to null.

ABSOLUTELY do not return your reasoning process. Return ONLY one JSON object that matches the format below and nothing else (in particular, never start your response with "JSON" or "Result"):

```
{
   "is_hallucination": True/False: { "type": "boolean" },
   "hallucination_reason": Reason for Hallucination { "type": ["string", "null"],
"maxLength": 1000},
}
```

Figure 15: Prompt template for automated hallucination judgment and attribution using step-by-step prompting method. The “question” and “trajectory” placeholders are replaced with the corresponding query and agent trajectory to be evaluated.

G-EVAL Template

You will be given one evaluation instance consisting of a question, an hallucinated agent trajectory, an expected hallucination explanation, and a generated hallucination explanation. The agent trajectory is a hallucinated attempt to answer the question, where the expected explanation is a human-annotated description of the earliest decisive cause underlying the hallucination, and the generated explanation is the evaluator model's predicted attribution.

<Question>**{question}**</Question>
<Agent Trajectory>**{trajectory}**</Agent Trajectory>
<Expected Explanation>**{expected_explanation}**</Expected Explanation>
<Generated Explanation>**{generated_explanation}**</Generated Explanation>

Your task is to evaluate the accuracy of generated explanation using the expected explanation as the gold reference.

Please make sure you read and understand following instructions carefully.

Evaluation Criteria:
Explanation Accuracy (1-5): the alignment between the generated explanation and the expected explanation in terms of error relevance, localization accuracy and causal correctness.
1. Score 1 (Fabricated): The explanation is distracted by irrelevant trajectory details and thus fails to attribute the error to the true hallucination cause, while also introducing fabricated evidence.
2. Score 2 (Mislocalized): The explanation references a genuine error in the trajectory but mislocalizes the hallucination by attributing it to a later step instead of the earliest decisive cause.
3. Score 3 (Wrong Cause at Correct Step): The explanation is grounded on the correct decisive step from the expected explanation, yet it misattributes the underlying cause of the hallucination.
4. Score 4 (Mostly Correct but Incomplete): The explanation matches the expected main cause with trajectory support, but is slightly incomplete or imprecise.
5. Score 5 (Exact Grounded): The explanation exactly matches the expected cause, is explicitly grounded in the trajectory, and adds no unsupported or contradictory content.

Evaluation Steps:
1. Read the question, agent trajectory, and expected explanation to establish the task intent and the trajectory segment where the hallucination becomes outcome-determining.
2. Indentify the key attribution claims in the generated explanation including the claimed error source, the described error mechanism, and the specific trajectory step it relies on.
3. Compare the generated explanation to the expected explanation for the specified evaluation criterion in terms of error relevance, localization accuracy and causal correctness.
4. Assign a single explanation accuracy score in <1,2,3,4,5> according to the criteria.
5. Ignore formatting mismatches and evaluate based on content-level correspondence only.

Answer the score from <1, 2, 3, 4, 5> and nothing else (in particular, never start your response with "I"):

Figure 16: Prompt template for explanation evaluation using G-EVAL method. The “question”, “trajectory”, “expected_explanation” and “generated_explanation” placeholders are replaced with the corresponding query, agent trajectory to be evaluated, the human-annotated expected explanation, and the explanation produced by the LLMs.

Planning Hallucination Attribution

Query: Which academic publisher published the paper "On Two Mathematical Representations for Semantic Maps"?
Answer: De Gruyter

Agent Trajectory

Step 1
Here are the facts: ...
Plan:
1. Search for the paper... prioritizing sources Elsevier...
5. select Elsevier as the academic publisher

Step 2
Web Search Tool
On Two Mathematical Representations for Semantic Maps
Elsevier
Parameters

•••

Step 4
The academic publisher of the paper "On Two Mathematical Representations for Semantic Maps" is Elsevier.

(Qwen3-32B) Answer:
Hallucination Judgment: **YES**
Responsible Step: **4**
Causal Explanation: The agent concluded that Elsevier is the academic publisher of the paper without sufficient evidence. The web search results did not confirm Elsevier as the publisher, and the search for the 'Journal of Mathematical Semantics' failed to provide any publisher information. The final answer was based on assumptions rather than verified data.

Ground Truth:
Hallucination Judgment: **YES**
Responsible Step: **1**
Causal Explanation: Step 1 Correctly understood the question, but guessed 'Elsevier' in advance in the task planning and wrote it into the plan, resulting in biased retrieval in the future, which belongs to Task Decomposition error and ultimately leads to incorrect answers.

Analysis:
The model correctly identifies the hallucination but misattributes its cause, blaming the final answer step rather than the earlier planning step where the error is introduced. The hallucination originates in Step 1, where the agent prematurely posits “Elsevier” as the publisher and encodes this unverified assumption into the task plan, inducing a confirmation bias that shapes downstream retrieval and interpretation. In contrast, the model focuses on Step 4, where the incorrect conclusion is stated, without tracing the causal chain back to the initial planning commitment that preconditions the trajectory.

Figure 17: Attribution example of planning hallucination category, with Qwen3-32B’s answers.

Retrieval Hallucination Attribution

Query: How many bronze medals did Sweden win at the 1956 Summer Olympics?
Answer: 6

Agent Trajectory

Step 1
Here are the facts:
...
Plan:
....

Step 3
Web Search Tool
Sweden bronze medals 1956 Summer Olympics.... *Parameters*
...Sweden did not win any bronze medals at the 1956 Summer Olympics.. *Responses*

Step 4
Sweden did not win any bronze medals at the 1956 Summer Olympics held in Melbourne, Australia.

(Claude-4.5-Sonnet) Answer:
Hallucination Judgment: **YES**
Responsible Step: **4**
Causal Explanation: The agent concluded that Sweden did not win any bronze medals at the 1956 Summer Olympics based on a search result snippet. However, this conclusion is incorrect. According to historical Olympic records, Sweden won 6 bronze medals at the 1956 Summer Olympics. The search result snippet appears to have been misinterpreted or was inaccurate.

Ground Truth:
Hallucination Judgment: **YES**
Responsible Step: **3**
Causal Explanation: The hallucination occurred in Step 3 because the agent relied on a misleading search engine snippet claiming Sweden won no bronze medals, instead of verifying the official medal table, leading to an incorrect conclusion.

Analysis:
The model correctly identifies the hallucination but misattributes its cause, assigning responsibility to the final response step rather than the earlier retrieval step. This failure arises from conflating error manifestation with error origin. The model focuses on the step that verbalizes the incorrect conclusion, without tracing the causal chain back to the retrieval decision that injected unverified evidence into the agent state.

Figure 18: Attribution example of retrieval hallucination category, with Claude-4.5-Sonnet's answers.

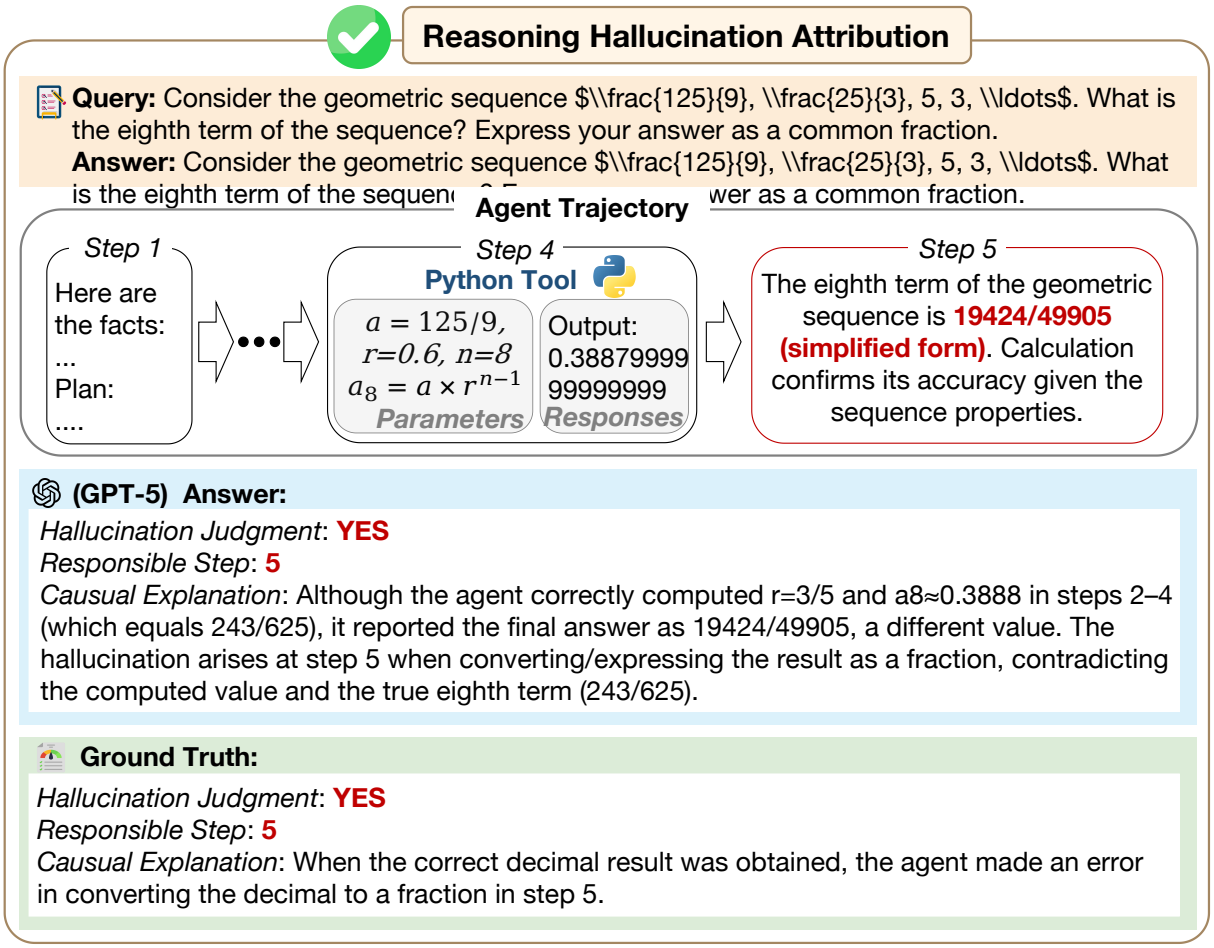

Figure 19: Attribution example of reasoning hallucination category, with GPT-5's answers.

Human-Interaction Hallucination Attribution

Query: On the BBC Earth YouTube video of the Top 5 Silliest Animal Moments, what species of bird is featured?
Answer: Rockhopper penguin

Agent Trajectory

Step 1
Here are the facts:
...
Plan:
....

Step 8
ask human via console Tool
Can you confirm if the video specifically names a type of penguin?
Parameters
...The featured bird in the video is described as a King Penguin...
Responses

Step 10
The species of bird featured in the BBC Earth YouTube video 'Top 5 Silliest Animal Moments' is the King Penguin

(DeepSeek-V3.1) Answer:
Hallucination Judgment: **YES**
Responsible Step: 10
Causal Explanation: Step 10 introduces hallucinated content by accepting the human-provided 'King Penguin' claim without independent verification. The agent's actual research (Steps 3-7) only found generic 'penguin' references, and the human response contains unverified specific species information that contradicts the verified sources available to the agent.

Ground Truth:
Hallucination Judgment: **YES**
Responsible Step: **8**
Causal Explanation: The agent accepted the user's incorrect claim that the bird was a King Penguin instead of verifying the actual species (Rockhopper penguin) through video observation or reliable sources, leading to a confidently stated wrong answer.

Analysis:
The model correctly identifies the hallucination but misattributes its origin to the final response step rather than the earlier human-interaction step where the incorrect information is introduced. The model blames Step 10, which verbalizes the incorrect conclusion, while overlooking Step 8, where where the human injects an unverified claim into the agent state.

Figure 20: Attribution example of human-interaction hallucination category, with DeepSeek-V3.1's answers.

Tool-Use Hallucination Attribution

Query: I've completed the maintenance on my car and ensured the doors are unlocked. Everything, especially the tires, seems in good condition. Would you kindly assist in securing the remaining doors and initiate the engine in START mode?
Answer: ["lockDoors(unlock=False, door=['driver', 'passenger', 'rear_left', 'rear_right'])", "pressBrakePedal(pedalPosition=1.0)", "startEngine(ignitionMode='START')"]

Agent Trajectory

Step 1 — **lockdoors Tool**

Step 2 — **press BrakePedal Tool**

Step 3 — **startEngine Tool**

Step 4 — **release BrakePedal Tool**

Step 5 — The doors are now locked, the engine is started, and the brake pedal is released.

(Gemini-2.5-Pro) Answer:
Hallucination Judgment: **NO**
Responsible Step: **NULL**
Causal Explanation: **NULI**

Ground Truth:
Hallucination Judgment: **YES**
Responsible Step: **4**
Causal Explanation: The hallucination occurred at step 4, where the agent issued an unnecessary releaseBrakePedal action not in the correct plan, adding an extraneous state change and deviating from the expected sequence.

Analysis:
The model classifies the trajectory as non-hallucinatory, despite an unjustified tool invocation. This error stems from not recognizing that the Step-4 releaseBrakePedal call is unnecessary and absent from the original plan. As a result, attribution collapses, since the model does not trace the causal chain to the extraneous tool call that introduces an invalid state transition, indicating limited sensitivity to redundant tool use in multi-step trajectories.

Figure 21: Attribution example of tool-use hallucination category, with Gemini-2.5-Pro's answers.