HALA-Bench: Hallucination Attribution and Localization Benchmark on Agentic RAG System

Anonymous ACL submission

Abstract

Agentic RAG systems interleave multi-step retrieval with iterative reasoning to solve complex multi-hop tasks, yet hallucinations can arise at diverse points along this chain—from malformed sub-queries to misleading documents—making it essential to attribute failures to their precise origin rather than merely flagging incorrect final answers. Existing benchmarks lack any framework for fine-grained, step-level attribution in agentic pipelines. We introduce HALA-BENCH, the first benchmark specifically designed for multi-step hallucination diagnosis in Agentic RAG. It is built on a principled taxonomy categorizing hallucinations into content-based (Query, Document, Evidence) and behavioral (Judgmental redundancy) types, and comprises 6,556 trajectories with step-level annotations. Evaluating ten prominent LLMs reveals a significant localization difficulty: while models can often detect whether a hallucination exists, joint accuracy for identifying both its type and exact step remains below 11% for most models, with document-level conflicts being especially elusive. Our code and data are available at https://anonymous.4open.science/r/hala-148B.

Question: *"Quando m'en vo' " is a soprano from an Opera composed by who ?*

TRAJECTORY

Step 0 [query_generation]
Query: *What opera is "Quando m'en vo'" from, and who originally performed it in the first production?*
(Original: *What opera is "Quando m'en vo'" from?*)

Step 0 [document_analysis]
Retrieved: *Opera; Mirette (opera); Carmen*
Evidence: *Mirette is an opéra comique in three acts composed by André Messager, first produced at the Savoy Theatre, London, on 3 July 1894.*

Step 1 [query_generation]
Query: *Who composed "Mirette", the opera containing "Quando m'en vo'"?*

Step 1 [document_analysis]
Retrieved: *Mirette (opera); Michel-Antoine Carré; Henri Miro*
Evidence: *"Quando m'en vo'" is a soprano aria from the opera "Mirette", which was composed by André Messager.*

Step final [answer_generation]
Final answer: `André Messager`

ANNOTATION:
`has_hallucination=true`, `type=query`, `step=0`,

Table 1: A hallucination example from HALA-BENCH. Query adds wrong constraint.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across knowledge-intensive tasks, yet their reliance on static parametric knowledge limits factual accuracy and adaptability to dynamic information needs (Lewis et al., 2020). Retrieval-Augmented Generation (RAG) mitigates this by grounding LLM outputs in external knowledge sources (Lewis et al., 2020; Gao et al., 2024). More recently, *Agentic RAG* has emerged as a paradigm shift: by embedding autonomous agents into the RAG pipeline, these systems interleave multi-step retrieval with iterative reasoning, enabling dynamic query reformulation, evidence validation, and adaptive workflow orchestration (Singh et al., 2026; Li et al., 2025).

LLMs are known to suffer from two fundamental categories of hallucination: *parametric hallucination*, where models generate factually incorrect content unsupported by their training data (Ji et al., 2023; Zhang et al., 2025), and *faithfulness hallucination*, where models produce outputs that contradict provided context (Huang et al., 2025; Sahoo et al., 2024). In Agentic RAG, retrieval and reasoning are interleaved across multiple iterations, making these failure modes especially consequential (Li et al., 2025; Singh et al., 2026). At the

query level, a malformed sub-question—or a miscalibrated model decision about when to retrieve—directs retrieval toward irrelevant evidence, giving rise to redundant and misdirected reasoning chains, as shown in Table 1; at the document level, a topically plausible but factually misleading document silently corrupts the reasoning context. There are more detailed examples in Appendix E. Without locating *where* and explaining *why* these hallucinations arise—whether from a flawed sub-query, a misleading document, or a reasoning drift—it becomes impossible to diagnose system failures or pinpoint inefficiencies. This underscores the need for fine-grained attribution that locates the precise source of failure at each step, rather than merely flagging an incorrect final answer.

Yet existing benchmarks fall critically short of this goal. Output-level benchmarks such as TruthfulQA (Lin et al., 2022) evaluate factual consistency entirely divorced from any retrieval process. RAG-specific benchmarks such as RAGTruth (Niu et al., 2024) incorporate retrieval but remain confined to single-turn settings, offering no visibility into multi-step error propagation. Agent benchmarks such as GAIA (Mialon et al., 2024) and τ-bench (Yao et al., 2025) evaluate multi-step tool use but treat hallucination as a binary outcome signal rather than a diagnostic target. No existing benchmark simultaneously supports **multi-step injection** of hallucination and **fine-grained attribution** along the full retrieval-reasoning chain in an Agentic RAG setting.

To bridge this gap, we introduce HALA-BENCH, the first benchmark specifically designed for hallucination attribution in Agentic RAG systems. At its core is a principled hallucination taxonomy that decomposes errors by their source along the retrieval-reasoning pipeline: Query, Document, Evidence and Judgement. To operationalize this taxonomy, we develop a suite of adversarial hallucination injection strategies and a multi-step injection framework that places hallucination at controlled positions along the reasoning trajectory, enabling step-level attribution evaluation.

Our main contributions are:

(i) We construct HALA-BENCH, the first benchmark for fine-grained hallucination attribution in Agentic RAG systems, containing 6,556 samples with type-level and step-level annotations across four hallucination dimensions.

(ii) We propose a principled multi-step hallucination injection framework with four adversarial strategies and controllable injection positions, enabling systematic evaluation of hallucination propagation along reasoning trajectories.

(iii) We benchmark multiple detection baselines and provide the first systematic analysis of how different hallucination types, injection positions, and agent behaviors interact in an Agentic RAG setting, yielding actionable insights for building more robust systems.

2 Related Work

2.1 Hallucination Detection Benchmarks for LLMs and RAG

Early benchmarking of hallucination in LLMs emphasized factual correctness of generated content, with TruthfulQA (Lin et al., 2022) exposing mimetic lies via adversarial questions and revealing that larger models are not necessarily more truthful. HaluEval (Li et al., 2023) expanded this effort to over 35,000 manually annotated samples across QA, dialog, and summarization, while FActScores (Min et al., 2023) enabled fine-grained evaluation by decomposing biographies into atomic statements validated against a reference corpus, and SelfCheckGPT (Manakul et al., 2023) offered a zero-resource alternative through multi-sample consistency checking. In RAG settings, models may contradict, ignore, or selectively reference provided documents, which RAGAS (Es et al., 2024) addresses through a suite of reference-free metrics measuring fidelity, answer relevance, and contextual utilization. RAGTruth (Niu et al., 2024) contributes the first large-scale naturally occurring hallucination corpus with word-level annotations, and stress-testing frameworks such as RGB (Chen et al., 2024) and CRUD-RAG (Lyu et al., 2025) further reveal failure modes involving noise, counterfactual conflicts, and incomplete retrieval that remain hidden in clean-environment benchmarks.

2.2 Agentic RAG Frameworks

The shift from single-turn RAG to agentic, iterative retrieval–reasoning systems has been driven by complementary advances in framework design and model-level training. On the framework side, ReAct (Yao et al., 2023; Singh et al., 2026) established the interleaving of reasoning traces and tool-use actions, which surveys later formalised as a closed loop of plan–retrieve–reason–act. On the model-training side, recent systems teach language models

Benchmark	Task (Hallucination)			Framework (Agentic RAG)		
	Judgement	Type Attr.	Step Loc.	Retrieval	Reasoning	Synergized
HaluEval (Li et al., 2023)	✓	×	×	×	×	×
RAGTruth (Niu et al., 2024)	✓	△	×	✓	×	×
HalluRAG (Ridder and Schilling, 2025)	✓	×	×	✓	×	×
CRAG (Yang et al., 2024)	✓	×	×	✓	×	×
AgentHallu (Liu et al., 2026)	✓	△	✓	✓	✓	△
τ-bench (Yao et al., 2025)	✓	×	×	✓	✓	✓
HALA-BENCH	✓	✓	✓	✓	✓	✓

Table 2: Comparison of hallucination and agent evaluation benchmarks. **Task** dimensions evaluate the granularity of hallucination diagnosis (Detection, Type, and Position). **Framework** dimensions characterize the system complexity. ✓ = fully supported; △ = partially supported; × = not supported.

to perform agentic search via reinforcement learning: Search-R1 applies outcome-based rewards for interleaving reasoning and search (Jin et al., 2025), and subsequent work progressively refines the training signal and decouples the search policy from the reasoner (Zhang et al., 2026; Xia et al., 2026), tracing a clear trajectory for autonomous, multi-turn search agents. Despite their impressive task performance, these systems optimise the retrieval–reasoning trajectory as a means to an end—whether via outcome or process rewards, the training signal targets performance gains rather than error localization.

2.3 Agent Evaluation and Hallucination Attribution

The closest line of work to ours evaluates LLM-based agents and their hallucination behavior in multi-step settings. GAIA and τ-bench (Mialon et al., 2024; Yao et al., 2025) assess multi-step reasoning with tools and agent-tool interaction, respectively, but both treat hallucination as a binary outcome with no diagnostic signal about *where* or *why* failure occurs. AgentHallu (Liu et al., 2026) advances the field by proposing step-level hallucination attribution—locating the earliest erroneous step in an agent's trajectory—yet it is designed for general-purpose agents and does not stratify hallucination by retrieval-specific sub-processes such as query formulation errors, document-level knowledge conflicts, or evidence extraction failures.

As summarized in Table 2, no prior benchmark simultaneously supports controlled, multi-step hallucination injection along the retrieval-reasoning chain, fine-grained attribution across retrieval-specific sources, and joint evaluation of hallucination detection accuracy.

3 Task Formulation

3.1 Agentic RAG Framework

An *Agentic RAG* system is a retrieval-augmented LLM agent that iteratively interleaves retrieval, evidence extraction, and reasoning to answer a given question x. Concretely, the agent operates over a sequence of discrete steps $t = 0, 1, \ldots, T$, where T is determined dynamically by the agent itself. At each step t, the agent performs the following:

Query generation. The agent formulates a retrieval query q_t based on the question x, previously retrieved documents $\{D_0, \ldots, D_{t-1}\}$, and its current reasoning state s_{t-1}:

$$q_t \sim \pi_\theta(\cdot \mid x, D_{<t}, s_{<t}) \tag{1}$$

Document retrieval. The query q_t is issued against an external knowledge corpus $\mathcal{K}$, returning a set of retrieved documents $D_t = \text{Retrieve}(q_t; \mathcal{K})$.

Evidence extraction. The agent extracts a structured evidence snippet e_t from the retrieved documents:

$$e_t \sim \pi_\theta(\cdot \mid x, q_t, D_t, s_{t-1}) \tag{2}$$

Reasoning update. The agent updates its internal reasoning state s_t by integrating the newly extracted evidence with prior knowledge and previous reasoning:

$$s_t \sim \pi_\theta(\cdot \mid x, e_t, s_{t-1}) \tag{3}$$

The agent repeats this cycle until it determines that sufficient information has been gathered, at which point it produces a final answer a at step T. We denote the complete **trajectory** as:

$$\tau = \big\{(q_0, D_0, e_0, s_0),\ (q_1, D_1, e_1, s_1),\ \ldots,\ (q_T, D_T, e_T, s_T);\ a\big\} \tag{4}$$

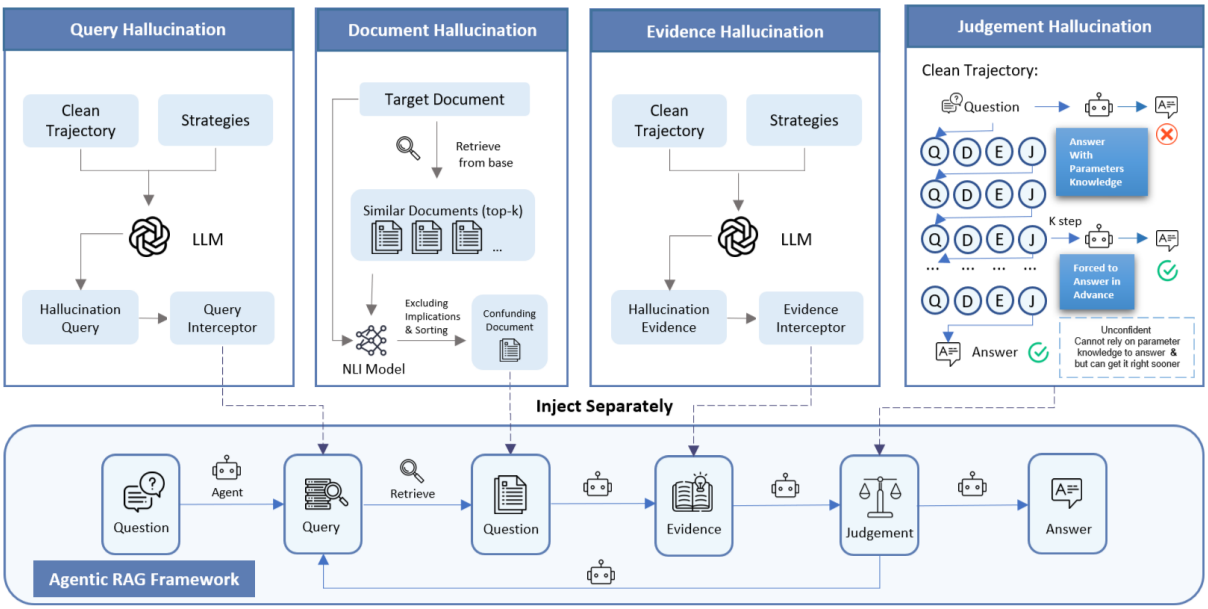

Figure 1: Overview of the HALA-BENCH construction framework. Q/D/E-type hallucinations are *injected* via adversarial strategies at their respective pipeline stages. J-type is annotated via early termination testing. The circles Q, D, E and J represent steps in the agentic RAG process. The four hallucinations are injected separately.

Type	Formal Condition
Q	$\text{Retrieve}(\tilde{q}_t) \not\subseteq \text{Retrieve}(q_t^*)$
D	$\exists\, d \in D_t$ conflicting with d' or ground truth f
E	$\text{Info}(e_t^*) \setminus \text{Info}(\tilde{e}_t) \neq \emptyset$
J	$k^* < T$, i.e., agent iterates beyond sufficient step

Table 3: Taxonomy of hallucination types in Agentic RAG.

3.2 Hallucination Taxonomy

Hallucinations can arise at any stage of the retrieval-reasoning pipeline. We define four types: query, document, evidence, and judgment. While the first three primarily involve information processing, judgment is distinct as it manifests as a behavioral hallucination characterized by agentic inefficiency. Table 3 presents the specific formal definitions.

Query hallucination (Q). The agent generates a query $\tilde{q}_t$ that is semantically misaligned with the information need of x—via entity substitution, constraint deletion, fabricated details, or unwarranted assumptions—causing retrieved documents to be off-target. The error is injected at step t but propagates through all downstream evidence and reasoning.

Document hallucination (D). Even with a well-formed query, the retrieval system may return documents that conflict with each other or with ground-truth facts (Longpre et al., 2021; Amiraz et al., 2025). The agent fails to detect or resolve these contradictions, incorporating false content into e_t or s_t.

Evidence hallucination (E). The agent extracts a snippet $\tilde{e}_t$ that omits key facts, focuses on irrelevant details, or over-generalizes from D_t, even when D_t contains all necessary information. The hallucination arises from *how* the agent processes retrieved documents, not from the documents themselves.

Judgment hallucination (J). The agent continues retrieving beyond the *sufficient step* k^*—the earliest step at which accumulated evidence $\{e_0, \ldots, e_{k^*}\}$ already supports a correct answer. The redundant steps $\{k^* + 1, \ldots, T\}$ constitute J-type hallucination, capturing a behavioral inefficiency rather than a content error.

4 Benchmark Construction

This section describes the construction of HALA-BENCH. We collect clean trajectories from a standard Agentic RAG system, inject three types of hallucination via controlled adversarial perturbation (Q, D, E), and annotate the fourth type (J) via early termination testing. Figure 1 illustrates the overall pipeline.

4.1 Baseline Trajectory Collection

We run the agentic pipeline using Reason-RAG (Zhang et al., 2026) (fine-tuned from Qwen-

2.5-7B-Instruct) on HotpotQA (Yang et al., 2018) dev split. We retain only trajectories with correct final answers and at least one non-empty, non-duplicate retrieval step, yielding clean trajectories as the foundation for all subsequent steps.

4.2 Hallucination Injection Overview

Q, D, and E hallucinations are injected at a designated step t within each clean trajectory; the agent then continues its normal loop from that point. All three types share the same construction pipeline: (1) generate hallucinated candidates at each retrieval step, (2) substitute the target component (query / retrieved documents / evidence snippet), and (3) apply type-specific quality filters.

4.3 Injection Details

Hallucinatory queries injection. Hallucinated queries are generated by GPT-5.4, GPT-5.4-mini, Gemini-3.1-Pro, and Claude-Sonnet-4.6 under a shared 2×2 perturbation taxonomy inspired by RAGTruth (Niu et al., 2024):

- *evident conflict* (QS1: Constraint Ablation).
- *subtle conflict* (QS2: Term Substitution).
- *evident baselessness* (QS3: Fabricated Detail Injection).
- *subtle baselessness* (QS4: Inferential Assumption).

Following the prompting methodology of HaluEval (Li et al., 2023), we design four corresponding strategies to elicit each hallucination type from LLMs during query construction.

Confounding documents injection. A confusion document is constructed by re-retrieving against the same Wikipedia corpus using the top-ranked document as a query, then selecting the highest-similarity candidate labeled *contradiction* or *neutral* by an NLI classifier (cross-encoder/nli-deberta-v3(He et al., 2021)). The confusion document is inserted together with the originals.

Hallucinatory evidence injection. Hallucinated evidence snippets are generated by GPT-5.4, Gemini-3.1-Pro, and Claude-Sonnet-4.6 under the same 2×2 perturbation taxonomy and prompting methodology described in Section 4.3, applied here at the document level: *direct contradiction* (evident conflict), *entity substitution* (subtle conflict), *fabricated detail* (evident baselessness), and *over-generalization* (subtle baselessness).

Judgment hallucination Annotation Judgement hallucination is annotated on clean trajectories rather than injected. For each trajectory of T steps, we conduct an **early termination test**: at each step $k \in [0, T-1]$, the agent is forced to answer using only evidence accumulated up to k. The earliest k^* at which the answer is correct is the *sufficient step*; if $k^* < T-1$, and the questions cannot be answered correctly by the parametric knowledge model alone, the trajectory exhibits judgment redundancy.

4.4 Human Verification

To validate our pipeline, we conducted a human study on 250 stratified samples (50 per type) across three dimensions: hallucination presence, type correctness, and step accuracy. As shown in Table 9 and Table 8, the pipeline's reliability is confirmed. Minor errors are confined to borderline cases with subtle signals, which reflect the inherent ambiguity of fine-grained hallucination detection.

4.5 Benchmark Statistics

The HALA-Bench comprises 6,556 trajectories drawn from 2,288 unique multi-hop questions, with 4,823 positive instances spanning four hallucination types and 1,733 negative controls. Table 7 summarizes the overall composition.

Class and Strategy Distribution Figure 2 presents the overall class composition of the dataset alongside the strategy-level breakdown within Q-type and E-type injections. Although all three types are constructed from the same trajectories with equal injection opportunities, their final proportions differ due to the yield of our validation pipeline. Evidence-type (E) dominates at 35.1%, as E-injections directly simulate model-internal hallucinations via LLM rewriting and most easily satisfy the formal definition. Query-type (Q) sits in the middle: a hallucinated sub-query must propagate through retrieval before affecting reasoning, and this buffer causes many candidates to dissipate before producing a verifiable effect. Document-type (D) is the smallest, since it does not fabricate content directly but introduces authentic yet misleading documents to emulate retrieval-induced hallucinations, and the strict requirement of retrieval divergence and valid NLI signals filters out most candidates. Despite their rarity, D-type cases prove the hardest to localize, as later experiments confirm. Negative controls account for 26.4%, providing a

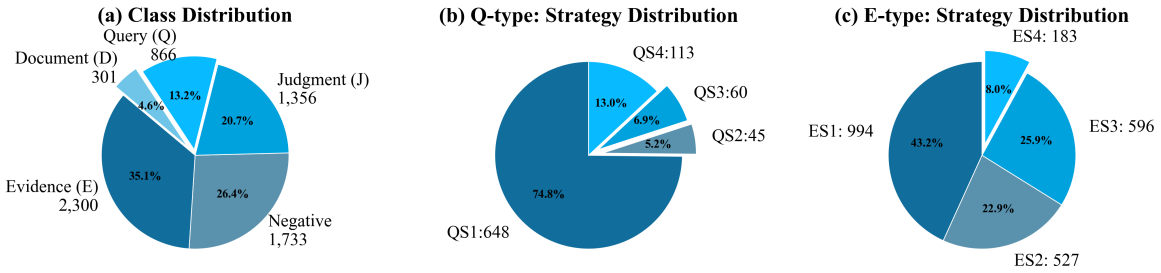

Figure 2: Composition of HALA-Bench. (a) Overall class distribution across five categories. (b) Q-type strategy distribution. (c) E-type strategy distribution.

balanced detection benchmark.

Strategy Distribution The strategy distributions in Figure 2(b–c) reflect the generating LLM's autonomous feasibility assessment rather than manual allocation. For Q-type, QS1 (Constraint Ablation) dominates because constraint deletion is universally applicable—every query contains scoping constraints that can be removed while preserving syntactic naturalness. QS4 (Inferential Assumption) ranks second, as embedding unwarranted presuppositions requires only a debatable implied premise. QS3 (Fabricated Detail) and QS2 (Term Substitution) are far less frequent, as both demand richer contexts that the generator often judges infeasible.

For E-type, ES1 (Direct Contradiction) dominates as a simple, low-risk negation of a key fact. ES3 (Entity Substitution) and ES2 (Fabricated Detail) are comparable, with the generator choosing between them based on whether nameable entities are present. ES4 (Over-Generalization) is least common, requiring answer-critical specifics to abstract away from—a condition many trajectories do not satisfy. This autonomous selection produces a non-uniform distribution that mirrors the practical difficulty landscape of adversarial hallucination generation.

5 Experiments

5.1 Experiments Setup

Evaluated Models. We evaluate ten instruction-tuned language models spanning different scales, families, and providers:

- `GPT-5.4, GPT-5.4-mini` (OpenAI, 2026a,b);
- `Qwen3-8B, 14B, 235B` (QwenTeam, 2025);
- `Llama-3.1-8B-Instruct` (Grattafiori et al., 2024).
- `Deepseek-V4-pro` (DeepSeek-AI, 2026) .
- `Grok-4` (xAI, 2024).
- `Gemini-3.1-pro` (Google, 2026).
- `Claude-Sonnet-4.6` (Anthropic, 2026)

All models are evaluated under a zero-shot prompting setup without fine-tuning.

Prompting Method. We adopt a structured prompting protocol that instructs the model to examine the full multi-step reasoning trajectory and output a JSON-formatted detection result. The prompt defines four hallucination categories (Q, D, E, J) and specifies that the model should identify the *earliest* step at which each hallucination type appears. For J-type redundancy, the model outputs the earliest step at which sufficient evidence was already acquired. Models receive the question and the full trajectory text.

Evaluation Metrics. We report **binary classification** metrics (accuracy, precision, recall, F1) for the coarse-grained *has_hallucination* prediction, plus the true negative rate (TNR) to quantify false-alarm behavior on clean trajectories. For **fine-grained attribution**, we separately report: (i) *type accuracy*—the proportion of positive samples for which the predicted hallucination type matches the ground truth, reported both overall and per-type (Q/D/E/J); and (ii) *step accuracy*—the proportion for which both type and step are simultaneously correct. Step accuracy is further broken down by the ground-truth injection position.

5.2 Comparison of Different LLMs

Table 4 reports the full detection results across nine LLMs and a random baseline, spanning three evaluation granularities. We analyze each dimension below.

Binary Detection Comparison. Overall, no model demonstrates a clearly dominant detection strategy. `Claude-Sonnet-4.6` achieves the highest F1 among all models, yet its TNR is the lowest, revealing an aggressive positive-bias strategy

Model	Binary Detection					Type Attribution						Step	
	Acc	Prec	Rec	F1	TNR	All	Macro F1	Q	D	E	J	Q+D+E	J
Closed-source Models													
Grok-4	70.4	**92.4**	65.2	76.4	**85.0**	39.5	29.2	**32.1**	2.0	69.9	1.1	**10.2**	0.0
Gemini-3.1-Pro	71.2	81.6	78.7	80.1	50.5	48.5	30.3	24.5	0.3	**90.8**	2.5	8.8	2.1
GPT-5.4	71.2	80.6	80.1	80.4	46.5	43.1	**32.6**	24.6	6.3	71.7	14.7	8.8	2.7
GPT-5.4-mini	60.9	88.2	54.1	67.0	79.8	30.9	29.6	22.4	3.0	55.7	0.4	7.5	0.0
Claude-Sonnet-4.6	**72.2**	73.9	**96.2**	**83.6**	5.3	**63.7**	26.8	4.9	1.0	88.3	**73.4**	2.6	**10.6**
Open-source Models													
DeepSeek-V4-pro	70.8	78.3	83.5	80.8	35.5	42.1	27.3	23.9	4.0	75.4	5.6	8.0	0.8
Qwen3-235B	69.6	79.6	78.9	79.3	43.9	23.9	26.3	27.1	27.2	30.7	9.5	6.9	2.7
Qwen3-8B	59.3	77.6	62.9	69.5	49.5	11.7	18.4	25.2	18.6	10.4	3.6	5.6	1.0
Llama-3.1-8B	71.0	75.1	90.7	82.2	16.2	12.5	12.8	31.1	**42.9**	9.0	0.0	4.8	0.0
Qwen3-14B	58.1	79.8	57.5	66.9	59.5	21.8	23.3	5.8	12.6	39.3	4.4	2.3	0.5
Random	50.0	73.6	50.0	59.5	50.0	25.0	18.4	25.0	25.0	25.0	25.0	6.3	6.3

Table 4: Detection results on HALA-BENCH. Type Acc. and Step are computed over positive samples only. Step is split by hallucination category: Q+D+E measures joint type-and-step accuracy on content-hallucination samples, while J measures joint type-and-step accuracy on redundancy samples (where the step label denotes the sufficient-evidence point rather than an injection step). All numbers in %. Best per column in **bold**. TNR: fraction of clean trajectories correctly classified as hallucination-free.

that approaches the degenerate predict-all-positive baseline under our label distribution. Conversely, `Grok-4` attains the highest precision and TNR, but at the cost of missing a substantial fraction of hallucinated trajectories. The tension between these two extremes suggests that models have not identified a robust decision boundary, likely because our adversarial injections preserve the surface-level coherence of valid agentic reasoning, making clean and hallucinated trajectories structurally difficult to distinguish by binary judgment alone.

Type Accuracy Comparison. The overall Type Acc. scores are moderate for closed-source models but drop sharply for open-source models, suggesting that smaller or open models struggle to reason about hallucination categories. However, per-type breakdowns reveal a more nuanced picture: all models perform well on **E**-type (evidence fabrication) while collapsing near zero on **D**-type (document distortion), indicating a systematic blind spot rather than uniform competence.

`Claude-Sonnet-4.6` appears as the sole exception at J-type accuracy, but this reflects a default-label bias rather than genuine discriminative ability: Claude predicts J for 74.7% of clean trajectories as well, and almost never abstains. The same bias explains its near-zero TNR, Q-type, and Step scores. Claude also no longer leads in Macro F1, with GPT-5.4 achieving the highest score.

Step Localization. We further assess whether models can jointly predict the correct type *and* pinpoint the exact step—a stricter criterion that requires both sub-tasks to be simultaneously correct.

Results are split by hallucination category. For content hallucinations (Q+D+E), scores are uniformly low: the best model (`Grok-4`) reaches only 10.2%, and most models cluster below 9%, barely above the random baseline. Performance on redundancy hallucinations (J) is even more constrained: `Claude-Sonnet-4.6` leads at 10.6%, but as discussed above this reflects default-label bias rather than genuine localization ability, and all remaining models score at or near zero. Taken together, these results show that current models lack the integrated reasoning capacity to jointly categorize and localize agentic hallucinations.

5.3 Step Localization by Injection Position

Figure 3 plots step accuracy as a function of the ground-truth injection step for six representative models on Q+D+E samples.

Two consistent patterns emerge. First, nearly all models peak at Step 1 and decline thereafter, suggesting a recency-independent bias toward early trajectory positions rather than genuine causal reasoning about where a hallucination was introduced. Second, performance at Step 0—where the hallucination is present from the very first action—is uniformly the lowest, indicating that models strug-

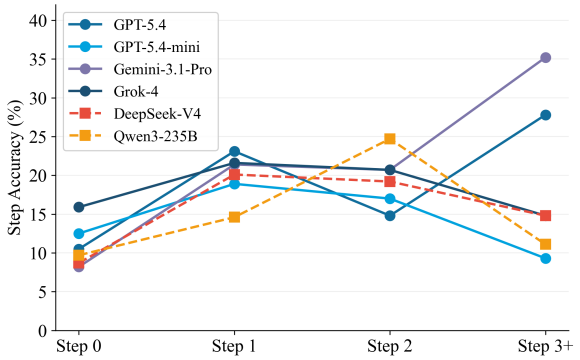

Figure 3: Step accuracy by injection step (Q+D+E only). Exact match between predicted and ground-truth hallucination step across six representative models.

gle most when there is no "clean" prefix to contrast against.

The divergence at Step 3+ is the most striking finding. `Gemini-3.1-Pro` and `GPT-5.4` improve sharply, higher than score observed across any model and any position. By contrast, `GPT-5.4-mini`, `DeepSeek-V4`, and `Qwen3-235B` all drop. This split suggests that longer trajectories amplify model-capacity differences: stronger models benefit from accumulating contextual evidence, whereas weaker models are overwhelmed by the increased context length and default to earlier positions.

Overall, no model achieves consistently high accuracy across all injection positions, confirming that precise step localization remains an open challenge regardless of trajectory length.

5.4 Strategy-Level Analysis

To validate whether detection difficulty varies by hallucination generation strategy, we break down Grok-4's Q-type and Gemini-3.1-Pro's E-type performance by the four injection strategies defined in our benchmark construction pipeline.

Q-type. As shown in Table 5, Grok-4's Q-type detection performance is dominated by strategy S1 (Constraint modification). On QS1, the recall and type accuracy Grok-4 achieves substantially below its already-modest overall Q-type figures. In contrast, strategies QS2–QS4, which involve more conspicuous manipulations such as entity injection, fabricated details, or unwarranted presuppositions, exhibit substantially higher recall and type accuracy. This sharp QS1–QS2/3/4 gap indicates that current detectors struggle primarily with *subtle constraint modifications* that preserve surface-level query plausibility while altering the retrieval target.

Strategy	#	Recall	Type	Step
QS1: Constraint modification	648	55.6	25.8	24.4
QS2: Entity injection	45	88.9	55.6	48.9
QS3: Detail fabrication	60	70.0	51.7	50.0
QS4: Assumption injection	113	78.8	48.7	31.9
All Q	866	61.3	32.1	28.4

Table 5: Grok-4 Q-type detection by hallucination strategy on HALA-BENCH. All numbers in %.

Strategy	#	Recall	Type	Step
ES1: Direct contradiction	994	99.8	93.6	0.0
ES2: fabricated detail	527	97.0	85.4	0.0
ES3: entity$_substitution$	596	98.5	91.9	10.9
ES4: over$_generalization$	183	98.9	87.4	23.5
All E	2300	98.7	90.8	4.7

Table 6: Gemini-3.1-Pro E-type detection by hallucination strategy on HALA-BENCH. All numbers in %.

E-type. Table 6 reveals a strikingly different pattern. Gemini-3.1-Pro detects evidence hallucination near-perfectly across all four strategies. However, step-level localization is severely deficient: on ES1, the most frequent strategy, step positioning failed completely, meaning Gemini correctly identifies that an evidence hallucination occurred and what type it is, but never pinpoints which step it was injected at. The sharp detection–localization dissociation underscores that recognizing the presence of evidence hallucination and locating its origin are fundamentally different capabilities.

6 Conclusion

This work shifts the evaluation paradigm from binary hallucination detection to stage-aware attribution. By decomposing errors into content-based and behavioral dimensions, HALA-Bench uncovers why and where the reasoning chain broke down. Our contribution lies in formalizing the necessity of traceability: the discovery of a pervasive localization gap across ten state-of-the-art LLMs—where detection capability fails to translate into precise step-level diagnosis—serves as a critical reality check for current agentic designs. Ultimately, this research provides the empirical evidence and the benchmarking framework required to move the field toward self-attributing agents that can not only recognize their mistakes but also pinpoint their origins to facilitate meaningful self-correction.

Limitations

Dataset and Domain Scope. Our benchmark is constructed from a single general-domain QA dataset, and the agent trajectories are generated by an opensource model fine-tuned via DPO specifically for Agentic RAG systems. Consequently, the generalization of our findings to specialized knowledge domains (e.g., biomedical, legal, or financial) and to agent models with different training objectives or architectures remains limited.

Reproducibility. Hallucination generation by LLMs and some LLMs evaluation in this work rely on proprietary LLM APIs. Because cloud-hosted models may be updated or deprecated without notice, exact reproduction of our results is not guaranteed. We mitigate this concern by releasing all prompts, annotations, and evaluation scripts, so that the benchmark itself remains stable even if specific model outputs change.

Implicit Assumptions. Our evaluation assumes that a well-formed agent trajectory has a single identifiable erroneous step, and that the injected hallucination does not interact with other steps in ways that produce cascading errors. In practice, real agent failures may involve distributed or compounding errors across multiple steps, which our current localization formulation does not fully capture. Results on Step-level accuracy should therefore be interpreted as a lower bound on localization difficulty rather than an exhaustive measure.

Ethical Considerations

HALA-Bench is derived entirely from publicly available QA datasets. It contains no personally identifiable information and no offensive or harmful content. We do not foresee direct harmful applications. Nevertheless, tools that evaluate hallucinations could hypothetically be repurposed to craft adversarial inputs that evade detection; we encourage the community to be mindful of this dual-use potential. This work is in full compliance with the Ethics Policy of the ACL.

References

Chen Amiraz, Florin Cuconasu, Simone Filice, and Zohar Karnin. 2025. The distracting effect: Understanding irrelevant passages in RAG. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18228–18258, Vienna, Austria. Association for Computational Linguistics.

Anthropic. 2026. Claude sonnet 4.6. https://www.anthropic.com/claude/sonnet. Large language model. Accessed: 2026.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17754–17762.

DeepSeek-AI. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence.

Shahul Es, Jithin James, Luis Espinosa Anke, and Steven Schockaert. 2024. RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Google. 2026. Gemini 3.1 pro. https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/gemini/3-1-pro. Accessed: 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2).

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training LLMs to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.

Yangning Li, Weizhi Zhang, Yuyao Yang, Wei-Chieh Huang, Yaozu Wu, Junyu Luo, Yuanchen Bei, Henry Peng Zou, Xiao Luo, Yusheng Zhao, Chunkit Chan, Yankai Chen, Zhongfen Deng, Yinghui Li, Hai-Tao Zheng, Dongyuan Li, Renhe Jiang, Ming Zhang, Yangqiu Song, and Philip S. Yu. 2025. A survey of RAG-reasoning systems in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 12120–12145, Suzhou, China. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Xuannan Liu, Xiao Yang, Zekun Li, Peipei Li, and Ran He. 2026. Agenthallu: Benchmarking automated hallucination attribution of llm-based agents. *Preprint*, arXiv:2601.06818.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuanjie Lyu, Zhiyu Li, Simin Niu, Feiyu Xiong, Bo Tang, Wenjin Wang, Hao Wu, Huanyong Liu, Tong Xu, and Enhong Chen. 2025. Crud-rag: A comprehensive chinese benchmark for retrieval-augmented generation of large language models. *ACM Trans. Inf. Syst.*, 43(2).

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2024. GAIA: a benchmark for general AI assistants. In *The Twelfth International Conference on Learning Representations*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

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 *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10862–10878, Bangkok, Thailand. Association for Computational Linguistics.

OpenAI. 2026a. Introducing GPT-5.4. https://openai.com/index/introducing-gpt-5-4/. Accessed: 2026-05-22.

OpenAI. 2026b. Introducing gpt-5.4 mini and nano. https://openai.com/index/introducing-gpt-5-4-mini-and-nano/. Accessed: 2026-05-22.

QwenTeam. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Fabian Ridder and Malte Schilling. 2025. The halluRAG dataset: Detecting closed-domain hallucinations in RAG applications using an LLM's internal states. In *AAAI 2025 Workshop on Preventing and Detecting LLM Misinformation (PDLM)*.

Pranab Sahoo, Prabhash Meharia, Akash Ghosh, Sriparna Saha, Vinija Jain, and Aman Chadha. 2024. A comprehensive survey of hallucination in large language, image, video and audio foundation models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11709–11724, Miami, Florida, USA. Association for Computational Linguistics.

Aditi Singh, Abul Ehtesham, Saket Kumar, Tala Talaei Khoei, and Athanasios V. Vasilakos. 2026. Agentic retrieval-augmented generation: A survey on agentic rag. *Preprint*, arXiv:2501.09136.

xAI. 2024. Grok-4. https://x.ai/news/grok-4. Accessed: 2024-05-20.

Tianle Xia, Ming Xu, Lingxiang Hu, Yiding Sun, wenwei li, Linfang Shang, Liqun Liu, Peng Shu, Huan Yu, and Jie Jiang. 2026. Search-p1: Path-centric reward shaping for stable and efficient agentic RAG training. In *The 64th Annual Meeting of the Association for Computational Linguistics – Industry Track*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel Gui, Ziran Will Jiang, Ziyu Jiang, Lingkun Kong, Brian Moran, Jiaqi Wang, Yifan Ethan Xu, An Yan, Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, and 8 others. 2024. Crag - comprehensive rag benchmark. In *Advances in Neural Information Processing Systems*, volume 37, pages 10470–10490. Curran Associates, Inc.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik R Narasimhan. 2025. {τ}-bench: A benchmark for \underline{T}ool-\underline{A}gent-\underline{U}ser interaction in real-world domains. In *The Thirteenth International Conference on Learning Representations*.

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 *The Eleventh International Conference on Learning Representations*.

Wenlin Zhang, Xiangyang Li, Kuicai Dong, Yichao Wang, Pengyue Jia, Xiaopeng Li, Yingyi Zhang, Derong Xu, Zhaocheng Du, Huifeng Guo, Ruiming Tang, and Xiangyu Zhao. 2026. Process vs. outcome reward: Which is better for agentic RAG reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2025. Siren's song in the ai ocean: A survey on hallucination in large language models. *Computational Linguistics*, 51(4):1373–1418.

A More Details on HALA-Bench

A.1 Benchmark Construction

HALA-Bench is constructed through a multi-stage pipeline over the HotpotQA development set. Starting from 2,687 questions with correct baseline trajectories (acc $= 1$ under the Qwen2.5-7B-Instruct-ReasonRAG agent), we first apply a data pre-filter to remove samples unsuitable for hallucination injection: (1) trajectories with zero retrieval steps (iteration_count $= 0$), where the model answers purely from parametric memory; and (2) trajectories where all retrieval queries are repeated yet return no evidence, indicating retrieval failure. This yields 2,357 clean trajectories with 3,827 injectable test cases across multiple retrieval steps.

For each hallucination type, we inject perturbations into the clean trajectories at a specific retrieval step. To ensure label validity, we apply a formal filtering stage that removes samples where the injection failed to produce a valid perturbation (e.g., hallucinated query identical to original, evidence unchanged after injection). Class sizes are then downsampled to prevent any single type from dominating evaluation, resulting in 6,556 entries across 2,288 unique questions (each question appears with multiple hallucination variants). Table 7 summarizes the overall statistics.

Statistic	Value
Total trajectories	6,556
Positive (hallucinated)	4,823 (73.6%)
Negative (clean)	1,733 (26.4%)
Unique questions	2,288
Avg. trajectory nodes	4.4 ± 2.1 (range 2–9)
Avg. retrieval steps	1.7 ± 1.0 (range 0–4)
Avg. answer length	14.0 chars (median=12)

Table 7: Overall benchmark statistics. Node count reflects the total number of reasoning steps in a trajectory; retrieval steps count only the rounds where an external retrieval action was taken.

A.2 Document Confusion Analysis

D-type injections require substitute documents that appear relevant yet contain conflicting facts. We quantify this via BGE cosine similarity and DeBERTa-v3 (He et al., 2021) NLI labels over the full candidate pool (3,795 pairs before filtering), summarized in Figure 4.

The similarity distribution is tightly controlled: no pair falls below 0.56 or exceeds 0.80, confirming surface overlap without near-duplication. On the NLI side, quality control removes all entailment pairs; the final 301 entries split into 87.4% neutral and 12.6% contradiction. Contradiction cases represent direct factual confusion, while the neutral majority presents a subtler challenge requiring relevance-mismatch detection. The mean NLI contradiction score of -0.38 (near the decision boundary) confirms that confusion signals are deliberately weak to moderate, mimicking the realistic ambiguity under which hallucinations naturally arise.

	Has Hall.	Type	Step	Joint
Correct	242	236	233	231
Total	250	250	250	250
Accuracy (%)	96.8	94.4	93.2	92.4

Table 8: Overall label accuracy across 250 human-verified samples. Joint accuracy requires hallucination presence, type, and step all correct simultaneously.

Type	#	Has Hall.	Type	Step	Joint
Q (Query)	50	100.0	100.0	98.0	98.0
D (Document)	50	100.0	96.0	92.0	92.0
E (Evidence)	50	92.0	84.0	84.0	80.0
J (Judgement)	50	100.0	100.0	100.0	100.0
Clean	50	92.0	92.0	92.0	92.0
Overall	250	96.8	94.4	93.2	92.4

Table 9: Per-type label accuracy (%). Joint: has_hall + type + step all correct.

B Agentic Framework Details

Retrieval. We use BGE-base-en-v1.5 (Xiao et al., 2023) as the retrieval model with FAISS flat indexing over a Wikipedia corpus of 1 million passages (wiki18_100w). For each query, the top-5 documents are retrieved via maximum inner-product search on BGE CLS embeddings. Retrieval is performed on GPU with FP16 precision. We use a query max length of 128 tokens and a retrieval batch size of 256.

Generation. The generator model is Qwen2.5-7B-Instruct-ReasonRAG, a 7B-parameter model DPO-fine-tuned on Agentic RAG trajectories. Inference is performed via the vLLM framework with greedy decoding (do_sample = False, max_tokens = 256). The model is deployed with tensor parallelism across 4 GPUs and a maximum model length of 32,768 tokens. The generation is constrained by stop tokens `<|im_end|>`,

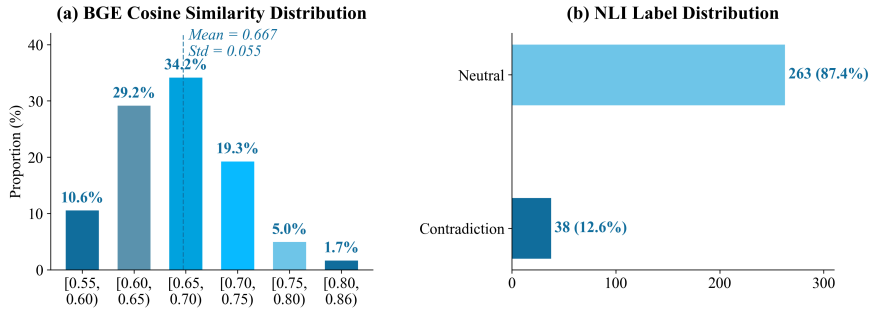

Figure 4: Document confusion analysis for D-type injections. (a) BGE cosine similarity between confounding and original top-1 documents. (b) NLI label distribution: 12.6% contradiction, 87.4% neutral. All entailment candidates are removed by quality control.

`</answer>`, `</query>`, and `</evidence>` to ensure clean structural outputs.

Evaluation Metrics. Following FlashRAG conventions, answer quality is evaluated using Exact Match (EM), F1, Accuracy, Precision, and Recall against the golden answers. For the benchmark, we use EM as the primary filter criterion: only trajectories where the baseline agent achieves acc $= 1$ (i.e., exact match with at least one golden answer) are retained for hallucination injection.

C Hallucination Detection Metrics

Given a trajectory, the detector outputs a prediction $\hat{y} \in \{0, 1\}$ (has_hallucination), a predicted type $\hat{t} \in \{\text{Q}, \text{D}, \text{E}, \text{J}\}$, and a predicted step $\hat{s} \in \mathbb{N}$. The ground truth consists of $y \in \{0, 1\}$, t, and s. We adopt a unified definition: if the detector predicts type $\hat{t} = \text{J}$, we override $\hat{y} = 1$ regardless of the raw has_hallucination output, since J-type redundancy is considered a hallucination by our benchmark definition.

Binary Detection. Let TP, FP, TN, FN be the standard counts over positive ($y = 1$, hallucinated) and negative ($y = 0$, clean) samples after applying the unified J-type rule. The binary metrics are:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

$$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{9}$$

TNR (True Negative Rate) specifically measures the fraction of clean trajectories correctly classified as hallucination-free, quantifying the detector's ability to avoid false alarms.

Type Accuracy. Type accuracy measures the detector's ability to correctly classify the hallucination category among positive samples. Let N_{pos} be the total number of ground-truth positive samples, and N_{pos}^t be the number of positives of type $t \in \{\text{Q}, \text{D}, \text{E}, \text{J}\}$. Let TypeCorrect be the count of positive samples where $\hat{t} = t$:

$$\text{Type Acc. (All)} = \frac{\text{TypeCorrect}}{N_{\text{pos}}} \tag{10}$$

$$\text{Type Acc. (Avg)} = \frac{1}{4} \sum_{t \in \{\text{Q,D,E,J}\}} \frac{\text{TypeCorrect}_t}{N_{\text{pos}}^t} \tag{11}$$

$$\text{Type Acc.}_t = \frac{\text{TypeCorrect}_t}{N_{\text{pos}}^t} \quad \text{(per-type)} \tag{12}$$

The weighted All metric reflects overall performance but is dominated by the most frequent class (E, 47.7% of positives). The macro-average Avg treats all four types equally, preventing large classes from masking poor performance on smaller classes.

Step Accuracy. Step accuracy requires both type and step to be simultaneously correct, jointly measuring the detector's ability to identify *what kind* of hallucination occurred and *where* it was injected:

$$\text{Step Acc.} = \frac{\sum_{i:y_i=1} \mathbb{I}[\hat{t}_i = t_i \wedge \hat{s}_i = s_i]}{N_{\text{pos}}} \tag{13}$$

We report step accuracy separately for two subsets, since their step labels carry different semantics:

- **Q+D+E step** ($s_{\text{Q+D+E}} \uparrow$): computed over Q, D, and E positive samples, where the step label denotes the injection step. Both type and step must match exactly.

- **J step**: computed over J positive samples only, where the step label denotes the sufficient-evidence point. $\hat{s}$ matches either the hallucination_step or sufficient_step field.

D More Details on Prompt Templates

D.1 Agentic RAG Prompts

As shown in Table 10, The Agentic RAG agent operates through four action-specific prompts. The user prompt for each action appends the current question and (for document analysis) retrieved documents to the system prompt below.

D.2 Hallucination Generation Prompts

Query Hallucination Generation. GPT-5.4-mini, GPT-5.4, Claude-Sonnet-4.6, and Gemini-3.1-Pro are prompted to rewrite queries. The generator receives the original question, retrieval step index, previous reasoning chain, and the current query or evidence, and outputs a JSON object with the chosen strategy and rewritten query or evidence. Table 11 shows the detailed prompt.

Evidence Hallucination Generation. GPT-5.4-mini, GPT-5.4, Claude-Sonnet-4.6, and Gemini-3.1-Pro are prompted to rewrite evidences. The generator receives the original question, retrieval step index, previous reasoning chain, and the current query or evidence, and outputs a JSON object with the chosen strategy and rewritten query or evidence. Table 12 shows the detailed prompt.

D.3 Hallucination Detection Prompt

The detector examines a complete reasoning trajectory and outputs a JSON-structured detection result.

E Case Study

We present one representative trajectory per hallucination type, focusing on how a single injected perturbation propagates through the reasoning chain and exerts persistent downstream influence. Injected content is **bolded**, and causal cascades are traced with $\rightarrow$.

Action	System Prompt
Begin Reasoning	You are an assistant for question answering with access to a retrieval tool. Upon receiving a question, your task is to: • **Analyze & Decompose:** Break the question into smaller, manageable sub-questions. • **Evaluate Knowledge:** Identify parts answerable from existing knowledge vs. parts requiring retrieval. • **Conciseness:** Use nouns or short phrases whenever possible. • **Response Format:** If sufficient: `So the answer is <answer>{answer} </answer>` If retrieval needed: `So the next query is <query>query </query>`
Document Analysis	You are an information retrieval assistant. Extract relevant evidence from the provided Wikipedia documents based on the latest query. • **Key Terms:** Identify key terms or concepts in the query. • **Evidence Search:** Search documents for supporting evidence. • **Response Format:** If found: `Based on the query, the relevant evidence is <evidence>evidence </evidence>` If none: `<evidence>None</evidence>`
Reasoning	You are a question-answering assistant with access to a retrieval tool. Provide concise and accurate reasoning. • **Error Reflection:** Identify and correct errors in previous thoughts if any. • **Sufficiency Check:** Evaluate whether current information is sufficient; if not, generate a new query (avoid repeating previous queries). • **Response Format:** If answering: `So the answer is <answer>{answer} </answer>` If more retrieval: `So the next query is <query>query </query>`
Answer Generation	You are a reasoning assistant with retrieval. Give a precise and very concise final answer for the given question, conclude with: `So the answer is <answer>{answer}</answer>` Keep your final answer brief and to the point, without any explanation.

Table 10: Agentic RAG system prompts for the four action states. User prompts append the current question and, for document analysis, the retrieved documents formatted as `Wikipedia Title: {title} \n{text}`.

Role	Content
System	You are an adversarial query rewriter for stress-testing multi-step Agentic RAG systems. Rewrite the current retrieval query into a *"hallucinated query"* that appears natural but causes the retrieval system to return wrong or misleading documents. Choose **ONE** strategy: **S1** **Constraint Ablation** — Remove or alter essential constraints (time, location, role, numerical thresholds). **S2** **Term Substitution** — Replace a precise term with a semantically close but factually different one. **S3** **Fabricated Detail Injection** — Add a plausible but unsupported specific detail. **S4** **Inferential Assumption** — Embed a presupposition that changes the factual framing. Output **ONLY**: `{"strategy": "S1|S2|S3|S4", "rewritten_query": "..."}`
User	**Original Question:** `{question}` **Step:** `{step_index}` *(0 = first retrieval)* **Previous Reasoning:** `{reasoning_chain}` **Current Query:** `{query}` **Output:**

Table 11: Prompt template for **Q-type** hallucinated query generation.

Role	Content
System	You are an adversarial test data generation expert. Given a question, golden answer, retrieved documents, and the original correct evidence, generate a **hallucinated evidence** that will mislead the Agent. Use **ONE** of four strategies defined by a 2×2 taxonomy: **S1** **direct_contradiction** [evident × conflict] — Reverse or negate a key fact. **S2** **fabricated_detail** [evident × baseless] — Insert a plausible but unsupported specific detail. **S3** **entity_substitution** [subtle × conflict] — Replace a key entity with a category-appropriate but incorrect one. **S4** **over_generalization** [subtle × baseless] — Replace specific details with vague descriptions. **Rules:** (1) The hallucinated evidence must **NOT** reveal or imply the golden answer. (2) It must look like a plausible extraction from the given documents. (3) For S1/S3: conflict must be with the correct answer. (4) For S2/S4: do not contradict documents. Output **ONLY**: `{"strategy": "S1|S2|S3|S4", "hallucinated_evidence": "..."}`
User	**Original Question:** `{question}` **Golden Answer:** `{golden_answers}` **Step:** `{step_index}` **Retrieved Documents:** `{documents}` **Original Evidence:** `{evidence}` **Output:**

Table 12: Prompt template for **E-type** hallucinated evidence generation.

Role	Content
System	You are a hallucination detector for Agentic RAG systems. An Agentic RAG system iteratively retrieves documents, extracts evidence, and reasons to answer a question. Examine a complete reasoning trajectory and determine: **1. External Hallucination (Q / D / E):** **Q** **Query Hallucination** — A retrieval query searches for wrong entities, missing constraints, or fabricated details. **D** **Document Hallucination** — Retrieved documents contain factually conflicting or fabricated information that the agent uncritically accepts. **E** **Evidence Hallucination** — Extracted evidence omits key facts, focuses on irrelevant details, or over-generalizes. **2. J-type Redundancy:** If no external hallucination exists, does the agent continue retrieval even though sufficient evidence was already acquired? Output the earliest step where evidence was sufficient (`sufficient_step`). **Rules:** (1) Identify the **earliest** step where hallucination appears (0-indexed). (2) Focus on retrieval and evidence content, **not** final answer correctness.

User	**## Question**
	`{question}`
	
	## Trajectory
	
	### Step 0 *[action_name]*
	`Query: {query}`
	`Response: {response}`
	`Retrieved documents:`
	`- [{title}] {text}`
	··· *(all subsequent steps)*
	
	Analyze the trajectory above and output your detection result in the following JSON format **ONLY**:
	
	`{`
	`"has_hallucination": true / false,`
	`"hallucination_type": "Q" / "D" / "E" / "J" / null,`
	`"hallucination_step": <0-indexed step or null>,`
	`"sufficient_step": <0-indexed or null>,`
	`"reasoning": "<brief explanation>"`
	`}`

Table 13: Prompt template for hallucination detection. The trajectory text is constructed by iterating over all intermediate nodes, each containing the action name, query, response, and truncated retrieved documents (max 150 chars per document).

Case Study: Query Hallucination

Question: The teaching method that Maggie Gripenberg was the first to introduce to Finland is used to teach what?
Answer: *music*

Injection: Step 0, **S1** (Constraint Modification) — Original query: *"What teaching method did Maggie Gripenberg introduce to Finland?"*

Step	Content
0 **[begin_reasoning]**	**Query:** *"What **dance training** method did Maggie Gripenberg introduce to Finland, and what subjects or skills does that method teach?"* ▶ **Spurious constraint injected:** "*dance training*" narrows scope to dance pedagogy rather than the actual method (Dalcroze Eurhythmics, a *music* education approach). The appended sub-question *"what subjects . . . does that method teach"* seeds an additional retrieval trajectory.
1 **[doc_analysis]**	**Response:** `<evidence>Dalcroze Eurhythmics</evidence>` → Despite the constraint, the retriever recovers the correct method name. However, the hallucinated sub-question now **commits the agent** to answering not just "what method" but "what is taught," forcing unnecessary further retrieval.
2 **[query_gen]**	**Query:** *"What subjects or skills does Dalcroze Eurhythmics teach?"* → **Detour propagates:** the spurious sub-question spawns an independent retrieval round. The agent abandons the original question and pursues a generic curriculum inquiry.
3 – 7 **[loop]**	→ **Retrieval deadlock:** Steps 3–7 alternate between the *same query* and the *same evidence* ("Dalcroze Eurhythmics"), with no document providing what the agent seeks. Four re-queries amplify the drift from the original question.
8 **[answer_gen]**	**Response:** `So the answer is <answer>musical concepts</answer>` ▶ **Wrong answer.** The spurious constraint at step 0 cascaded into **6 redundant retrieval rounds** and a vague, incorrect answer. The original question was answerable in **2 steps**.

Table 14: **Query-type propagation**. A spurious constraint at step 0 spawns a retrieval deadlock spanning 6 additional steps, culminating in a wrong answer.

Case Study: Document Hallucination

Question: Ray Roughler Jones single-handedly produced the magazine Roughler, but what Swedish singer-songwriter made contributions to the magazine? **Answer:** *Neneh Cherry*

Injection: Step 0, **D-type** (Real-corpus confound *"Ladbroke Estate,"* similarity = 0.568)

Step	Content
0 **[begin_reasoning]**	**Query:** *"Who made contributions to the magazine Roughler besides Ray Roughler Jones?"* → Correct initial query; no distortion at the query level. Hallucination originates in the *retrieved document set*.
1 **[doc_analysis]**	▶ **Confound "Ladbroke Estate" injected at position 0.** **Response:** `<evidence>None</evidence>` ▶ **Evidence vacuum:** the confound document obscures relevant content, yielding *no useful evidence*. This forces a re-query at the next step.
2 **[query_gen]**	**Query:** *"Which **Swedish singer-songwriter** contributed to the magazine Roughler?"* → **Re-query with category constraint:** the `None` result at step 1 propagates into a narrower query that bakes in the answer type from the original question, artificially constraining the search space.
3 **[doc_analysis]**	**Response:** `<evidence>Kjell Höglund</evidence>` → **Wrong evidence:** the narrower query retrieves *Kjell Höglund*, a Swedish singer-songwriter with **no connection** to Roughler. The confound-created vacuum at step 1 is filled by the first category-matching entity encountered in the corpus.
4 **[answer_gen]**	**Response:** `So the answer is <answer>Kjell Höglund</answer>` ▶ **Wrong answer.** Full causal chain: $\underbrace{\text{confound insertion}}_{\text{step 1}} \rightarrow \underbrace{\text{evidence vacuum}}_{\text{step 1}} \rightarrow \underbrace{\text{category-constrained re-query}}_{\text{step 2}} \rightarrow \underbrace{\text{wrong entity}}_{\text{step 3}} \rightarrow \underbrace{\textbf{wrong answer}}_{\text{step 4}}$ The confound was **never cited**, yet its presence at step 1 was the *root cause* of the failure.

Table 15: **Document-type propagation**. An obscure confound document creates an evidence vacuum at step 1, triggering a re-query that converges on a category-plausible but factually wrong entity.

Case Study: Evidence Hallucination

Question: Which satellite suburb of Melbourne is partially included in the Electoral district of Bass? **Answer:** *Pakenham*

Injection: Step 1, **S1** (Entity Replacement) — Original evidence: *"Pakenham"* → ***"Inverloch"***

Step	Content
0 **[begin_reasoning]**	**Query:** *"Which satellite suburb of Melbourne is included in the Electoral district of Bass?"* → Correct initial query; no distortion at the query level. Hallucination originates in the *extracted evidence* at step 1.
1 **[doc_analysis]**	**Response:** `<evidence>Inverloch</evidence>` ▶ **Wrong entity injected** into the agent's memory: *Pakenham* (correct) replaced with *Inverloch*, a different Victorian coastal town with no satellite-suburb relationship to Melbourne.
2 **[query_gen]**	Agent **senses inconsistency:** *"Inverloch is mentioned, but it is not a satellite suburb of Melbourne."* → **Self-correction attempted.** However, because the evidence was *injected* rather than retrieved, the same hallucinated entity will return on every subsequent re-query, rendering self-correction structurally impossible.
3 **[doc_analysis]**	**Response:** `<evidence>Inverloch</evidence>` → **Confirmation bias:** receiving *"Inverloch"* a second time reinforces the false belief that documents *consistently* support this entity, overriding the agent's prior skepticism.
4 – 7 **[loop]**	→ **Evidence trap — self-reinforcing loop:** *"Is Inverloch a satellite suburb?"* ⇒ `Inverloch` Re-asking original question ⇒ `Inverloch` Alternative phrasings ⇒ `Inverloch` Broader scope queries ⇒ `Inverloch` Every retrieval path returns the same wrong entity, progressively **eroding the agent's initial skepticism** through sheer repetition.
8 **[answer_gen]**	**Response:** `So the answer is <answer>Inverloch</antwer>` ▶ **Wrong answer after 8 failed escape attempts.** Full causal chain: $\underbrace{\text{entity replacement}}_{\text{step 1}} \rightarrow \underbrace{\text{memory corruption}}_{\text{step 1}} \rightarrow \underbrace{\text{failed self-correction}}_{\text{step 2}} \rightarrow \underbrace{\text{confirmation bias}}_{\text{step 3}} \rightarrow \underbrace{\text{evidence trap}}_{\text{steps 4–7}} \rightarrow \underbrace{\textbf{capitulation}}_{\text{step 8}}$ A single evidence-level injection overrode the agent's fact-checking instincts through **sheer repetition**.

Table 16: **Evidence-type propagation**. A hallucinated entity creates a self-reinforcing retrieval loop: every re-query returns the same wrong entity, eroding skepticism until the agent capitulates after 8 failed escape attempts.

Case Study: Judgement Hallucination

Question: Which team featured in both the 2012 and 2011 Copa del Rey Finals? **Answer:** *Barcelona*

Annotation: `sufficient_step = 1` — Truncated trajectory at step 1 answers correctly (`ACC=1`); at step 0, direct answer fails (`ACC=0`)

Step	Content
0 **[begin_reasoning]**	**Query:** *"Which teams featured in both the 2012 and 2011 Copa del Rey Finals?"* ▶ **Pre-retrieval answer check** (`ACC=0`): If forced to answer now, the agent outputs *"Atlético Madrid"* — **wrong**. The agent cannot answer from parametric knowledge alone; retrieval is necessary.
1 **[doc_analysis]** ★	**Response:** `<evidence>`**`Athletic Bilbao and Barcelona`**`</evidence>` ★ **Sufficient evidence point** (`sufficient_step = 1`): A single retrieval round provides enough information to answer correctly. Truncation test at step 1 yields *"Athletic Bilbao and Barcelona"* — **correct** (`ACC=1`). → Barcelona is identified as a team appearing in *both* finals. **All subsequent steps are redundant by definition.**
2 **[query_gen]**	Agent **fails to recognise sufficiency:** *"We need to confirm if both Athletic Bilbao and Barcelona featured in **both** years."* → **Redundancy trigger:** despite possessing sufficient evidence at step 1, the agent demands *"confirmation"* and launches an unnecessary retrieval round. This is the **root of the J-type redundancy**: not a corrupted document or injected entity, but a miscalibrated sufficiency judgment.
3 – 7 **[redundant loop]**	→ **Confirmation loop — 6 redundant rounds:** Query: *"Confirm participation of Athletic Bilbao in 2011 final…"* Evidence: `Athletic Bilbao and Barcelona` (identical) Query: *"Confirm participation of Barcelona in 2012 final…"* Evidence: `Athletic Bilbao and Barcelona` (identical) Every retrieval returns **identical information**. The agent is trapped re-verifying a fact already established at step 1. No new evidence is acquired; no reasoning progress is made.
8 **[answer_gen]**	**Response:** `<answer>Athletic Bilbao and Barcelona</answer>` → **Same answer, 6 steps too late.** The output at step 8 is *identical* to what could have been produced at step 1. The trajectory is entirely clean—no fabricated queries, no confound documents, no corrupted evidence—yet exhibits a clear failure: *the agent does not know when to stop.*

Table 17: **Judgement-type hallucination**. The sufficiency test reveals a sharp discontinuity: at step 0 the agent answers incorrectly from parametric memory (`ACC=0`), but at step 1 a single retrieval round already provides sufficient evidence for the correct answer (`ACC=1`). Yet the full trajectory runs for 9 steps—6 of which are redundant confirmation rounds that neither add evidence nor improve the answer.