

GroundRAG: Evidence-Verified Multi-Hop Reasoning on Lightweight Graphs

Jinjie Hao¹ Guixuan Zhang^{1,*} Ying Huang¹ Jie Liu¹ Shuwu Zhang¹

¹Beijing University of Posts and Telecommunications, China

{jinjie.hao, guixuan.zhang, ying.huang, ailj, shuwu.zhang}@bupt.edu.cn

Abstract

Multi-hop Retrieval-Augmented Generation can fail even when relevant evidence has been retrieved. The remaining difficulty lies in how evidence is connected across reasoning steps: an intermediate answer may be unsupported, attached to the wrong anchor, or propagated into a later step without valid evidence. We study this failure as *Reasoning Chain Hallucination* (RCH), where a fluent multi-hop reasoning process contains one or more ungrounded intermediate steps. To address RCH, we formulate Retrieve–Verify–Generate (R–V–G) for multi-hop RAG: retrieved evidence should support an explicitly verified reasoning structure before it is allowed to guide generation. GROUND-RAG is an evidence-verified framework that implements this requirement. GroundRAG first acquires the evidence needed to instantiate intermediate reasoning variables, then represents the reasoning process as a typed dependency chain. Each step is checked against retrieved passages using explicit grounding criteria; unresolved steps are repaired with queries formed from verified intermediate answers, while insufficiently grounded chains are discarded before generation. This design separates evidence acquisition from the decision of whether a reasoning chain is safe to use. On HotpotQA, 2WikiMultiHopQA, and MuSiQue, GroundRAG reaches 73.8%, 69.2%, and 44.3% LLM-judged accuracy, respectively, outperforming the strongest baseline in our evaluation.

Keywords: Retrieval-Augmented Generation , Multi-hop Question Answering , Grounded Chain Verification , Graph-based Retrieval , Large Language Models

1 Introduction

Retrieval-Augmented Generation (RAG) is widely used to mitigate factual hallucination in Large Language Models by grounding generation in query-relevant external evidence (Lewis et al., 2020; Ji

et al., 2023). As RAG increasingly relies on external knowledge bases, the integrity and ownership of retrieved corpora have also emerged as system-level concerns (Wang et al., 2026). On simple factoid questions the recipe is direct. On multi-hop questions—where the answer requires combining facts from two or more passages—it stops being so: the first hop’s retrieval may not surface the second hop’s evidence, the bridging entity may not appear in the question, and the model may produce a chain of reasoning that no retrieved passage actually supports.

Graph-based RAG (GraphRAG) addresses multi-hop retrieval by inducing structure over the corpus and propagating evidence across hops (Edge et al., 2024; Pan et al., 2024; Chen, 2024; Xiao et al., 2026). Early work such as RAPTOR (Sarthi et al., 2024) and Microsoft GraphRAG (Edge et al., 2024) organises knowledge through hierarchical summarisation, enabling coarse-to-fine retrieval. Building on this, GFM-RAG (Luo et al., 2025), G-Retriever (He et al., 2024), and LightRAG (Guo et al., 2025) integrate specialised encoders and indexing strategies for scalability and efficiency. HippoRAG (Gutiérrez et al., 2024) and HippoRAG 2 (Gutiérrez et al., 2025) draw on cognitive-science motivations to apply Personalised PageRank for multi-hop retrieval, and LinearRAG (Zhuang et al., 2026) further reduces graph-construction cost via a token-free tri-graph. A parallel line of work attacks the same problem through question decomposition: IRCoT (Trivedi et al., 2023) interleaves chain-of-thought with retrieval, ReD-RAG (Chen et al., 2025) trains a dedicated decomposer with rewriting markers, ChainRAG (Zhu et al., 2025) completes entities between hops, LAG (Xiao et al., 2025) uses logical-self-judgement terminators, and LogicRAG (Chen et al., 2026) constructs dependency DAGs at inference time.

Better retrieval alone does not make multi-hop

*Corresponding author.

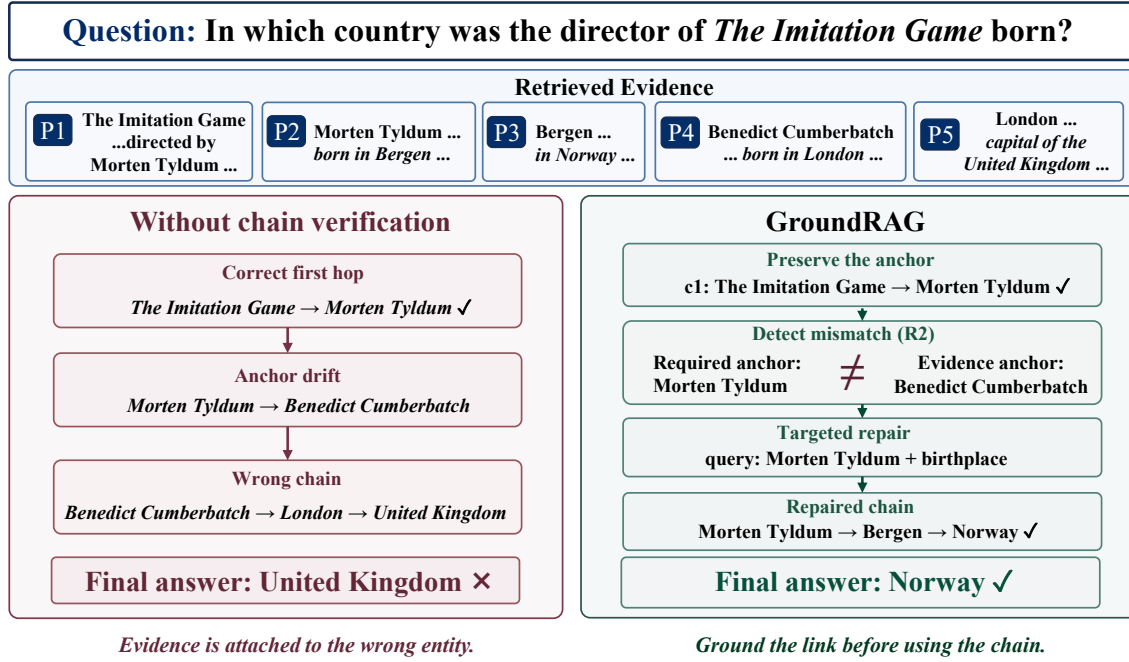


Figure 1: Evidence availability does not guarantee a grounded reasoning chain. In this illustrative bridge-entity example, all required facts are available alongside a plausible competing person. Without explicit dependency verification, the birthplace relation can drift from the film’s director to its lead actor, yielding a locally supported but globally incorrect chain. GroundRAG preserves the intermediate anchor in a typed slot, detects R2 (anchor-not-supported), repairs the unresolved dependency, and returns Norway.

RAG reliable. A system may retrieve passages containing the information needed for the final answer and still combine them incorrectly across reasoning steps. The key difficulty is therefore not only whether relevant evidence is available, but whether intermediate answers remain grounded as they are passed from one step to the next. A bridge entity recovered at an earlier step may become the anchor of a later relation; if that dependency is lost, the model can produce a fluent answer supported by a passage that is nevertheless attached to the wrong entity. We refer to this class of failures as *Reasoning Chain Hallucination* (RCH): one or more intermediate steps in an apparently coherent multi-hop chain are not supported by the evidence assigned to them. RCH is distinct from token-level hallucination (Ji et al., 2023): the surface chain may be fluent and factually plausible, while the dependency connecting its steps is ungrounded. Figure 1 shows that the required evidence can already be present while evidence-only generation still attaches a later relation to the wrong intermediate entity.

The distinction between evidence availability and chain correctness is important. Retrieving an answer-bearing passage does not establish that the model has followed the relations required by the

question. In multi-hop settings, evidence must be attached to the correct intermediate variables and used under the dependencies established by earlier steps. This motivates a Retrieve–Verify–Generate (R-V-G) formulation: retrieved evidence should not directly determine generation, but should first be organised into an explicit reasoning structure whose intermediate dependencies can be grounded and checked. Existing decomposition systems leave part of this problem open: IRCot and ReD-RAG do not ground every sub-answer against a cited passage; ChainRAG focuses on preserving the anchor entity between hops; LAG relies on LLM self-judgement; and LogicRAG prunes context without an explicit slot-level grounding decision.

Inspection of our system’s grounded-rejection signals on three benchmarks yields four recurring reason codes—*answer-not-supported* (437 cases on HotpotQA, 257 on MuSiQue), *anchor-not-supported* (225 on HotpotQA, 154 on 2WikiMultiHopQA), *evidence-not-in-passage* (111 on HotpotQA, 408 on MuSiQue), and *constraint-not-supported* (a few dozen across datasets). The dominant code shifts from answer support on HotpotQA to anchor support on 2WikiMultiHopQA and span alignment on MuSiQue, suggesting a structured,

dataset-dependent failure family rather than undifferentiated residual error.

We introduce GROUND-RAG, an evidence-verified framework built on a lightweight token-free tri-graph (Zhuang et al., 2026) of passages, sentences, and entities. Ground-RAG acquires evidence for latent intermediate variables, represents their dependencies as typed slots, and grounds, repairs, or rejects the resulting chain before generation. A chain guides final generation only when it has sufficient grounded coverage; otherwise the system falls back to generation from retrieved evidence alone. Our contributions are threefold:

- We identify and operationalise *Reasoning Chain Hallucination* as a failure mode of multi-hop RAG in which apparently coherent intermediate reasoning is not supported by the retrieved evidence. We characterise RCH through four grounding failure codes and analyse how their distributions vary across benchmarks.
- We introduce GROUND-RAG, an R-V-G framework for multi-hop reasoning that acquires evidence for intermediate steps, represents their dependencies with typed slots, checks each step against retrieved passages, repairs unresolved dependencies using verified intermediate answers, and falls back to evidence-only generation when the chain cannot be sufficiently grounded.
- We evaluate Ground-RAG on three multi-hop benchmarks and provide mechanism-level analyses of grounding rejection, targeted repair, retrieval coverage, and residual errors.

2 Related Work

Multi-hop RAG has been approached from two largely independent directions: building structured indexes over the corpus, and decomposing the question into sub-questions resolved one by one. We summarise both, then turn to self-verification, the closest neighbour to our own verification stage.

Graph-based RAG. Graph-based RAG induces a structure over the corpus and propagates evidence across hops. RAPTOR (Sarathi et al., 2024) clusters passages into a hierarchical summary tree, and Microsoft GraphRAG (Edge et al., 2024) organises knowledge through community-based summarisation. G-Retriever (He et al., 2024) learns

a graph neural network on text-graph alignments; LightRAG (Guo et al., 2025) reduces construction cost via dual-level retrieval; LinearRAG (Zhuang et al., 2026) further drops to a token-free tri-graph that incurs no LLM cost. HippoRAG (Gutiérrez et al., 2024) and HippoRAG 2 (Gutiérrez et al., 2025) apply Personalised PageRank with NER-anchored seeds, motivated by cognitive analogies; GFM-RAG (Luo et al., 2025) replaces the pipeline with a single graph foundation model. GraphRAG-Bench (Xiao et al., 2026) documents the brittleness of these systems on domain-specific multi-hop reasoning. Ground-RAG uses a lightweight token-free graph as an evidence substrate rather than treating graph traversal itself as the main contribution. Its main departure comes after evidence is acquired: intermediate variables are made explicit, their dependencies are preserved across hops, and each instantiated step is checked against its supporting passages. The graph therefore serves chain construction, while Ground-RAG asks whether the resulting chain is grounded well enough to guide generation.

Decomposition-based multi-hop RAG. A parallel line decomposes q into sub-questions and interleaves sub-question resolution with retrieval. IRCOT (Trivedi et al., 2023) interleaves chain-of-thought sentences with retrieval rounds, but does not explicitly ground each step against a cited passage. ReD-RAG (Chen et al., 2025) formalises decomposition with rewriting markers and trains a dedicated decomposer on a curated dataset. ChainRAG (Zhu et al., 2025) preserves anchor entities between hops through entity completion and sentence-graph retrieval. LAG (Xiao et al., 2025) adds a logical self-judgement terminator, while LogicRAG (Chen et al., 2026) constructs a dependency DAG at inference time and prunes context. Ground-RAG is complementary to these systems: it uses a lightweight graph as an evidence substrate, then combines typed dependencies, passage-addressed grounding reason codes, targeted repair, and an explicit fallback when the proposed chain remains unsupported.

Self-verification in RAG. Self-RAG (Asai et al., 2024) interleaves reflection tokens into generation, training the model to flag whether each retrieved chunk is relevant and whether each generated segment is supported. The mechanism is end-to-end and parametric—answer and verification come from the same model, so calibration of

the reflection tokens is an empirical property of that model. GroundRAG’s grounding routine differs in being rule-based, non-circular with the verifier (an unsupported *verified* status is rejected by the recheck even when the verifier is confident), and producing categorical reason codes that drive the targeted-repair loop. The two approaches operate at different layers and are not mutually exclusive; we leave their combination to future work.

3 GroundRAG: Evidence-Verified Multi-Hop Reasoning

3.1 Overview

GroundRAG treats multi-hop question answering as the construction and validation of evidence-grounded intermediate variables. Given a question q , the system first acquires candidate evidence that can instantiate the required reasoning steps. It then represents those steps as a typed dependency chain, where the answer of one slot may become the anchor of another. Each instantiated slot is checked against its cited passage. Unsupported slots return to evidence acquisition through queries formed from already verified answers, and only chains with sufficient grounded coverage are allowed to guide final generation. Figure 2 summarises this flow.

The evidence index is a Tri-Graph following LinearRAG (Zhuang et al., 2026). Given a corpus $\mathcal{P}$, sentences $\mathcal{S}$ are obtained by punctuation-based segmentation and entities $\mathcal{E}$ by lightweight NER (Hon-nibal et al., 2020). The graph $G = (V, E)$ has three node types and two adjacency matrices: passage-entity incidence $C \in \{0, 1\}^{|\mathcal{P}| \times |\mathcal{E}|}$ and sentence-entity weights $M \in \mathbb{R}_{\geq 0}^{|\mathcal{S}| \times |\mathcal{E}|}$,

$$\begin{aligned} C_{ij} &= \mathbf{1}[e_j \in p_i], \\ M_{\ell j} &= \text{tf-idf}(e_j, s_\ell), \end{aligned} \quad (1)$$

where $p_i \in \mathcal{P}$, $s_\ell \in \mathcal{S}$, and $e_j \in \mathcal{E}$; $\mathbf{1}[\cdot]$ is the indicator function. Construction uses no LLM calls and scales linearly with corpus size. The graph is an evidence index; the typed chain introduced below defines the reasoning structure.

3.2 Evidence Acquisition for Chain Construction

This stage combines complementary evidence signals, latent intermediate-variable recovery, and provenance-weighted propagation. Together they return the passages and graph-mapped intermediate variables needed to instantiate slots and form targeted repair queries.

Complementary evidence signals. We initialize the evidence state from explicit entity matches, semantic matches, and entities observed in the initially retrieved context. Explicit-match evidence E_{exp} is extracted from q via NER and matched to $\mathcal{E}$ by surface form, with optional top- K embedding expansion at threshold 0.7. Semantic-match evidence E_{sem} contains entities e satisfying $\cos(\phi(q), \phi(e)) \geq \tau_s$, where ϕ is the all-mpnet-base-v2 encoder (Song et al., 2020) and $\tau_s = 0.85$ is fixed across datasets. For context-derived evidence, we dense-retrieve the top-3 sentences under ϕ and collect their entities in E_{ctx} . Their union $E_{\text{init}} = E_{\text{exp}} \cup E_{\text{sem}} \cup E_{\text{ctx}}$ initializes propagation; the provenance weights below control the contribution of each source.

Latent intermediate-variable recovery. One evidence-conditioned expansion pass recovers intermediate variables that are absent from q . PPR with E_{init} returns the top- K passages, an LLM proposes entities needed to continue the reasoning path, and candidates that map to graph entities form E_{lat} . A variable previously supported only by semantic matching is promoted when it is recovered from context. The resulting variables are passed to typed chain construction rather than treated as final answers.

Provenance-weighted propagation. We assign provenance-dependent restart weights to the evidence signals:

$$\begin{aligned} w(e) &= \begin{cases} 1.0 & e \in E_{\text{exp}} \cup E_{\text{lat}}, \\ 0.4 & e \in E_{\text{sem}} \setminus (E_{\text{exp}} \cup E_{\text{lat}}), \\ 0.2 & e \in E_{\text{ctx}} \setminus (E_{\text{sem}} \cup E_{\text{exp}} \cup E_{\text{lat}}), \\ 0 & \text{otherwise,} \end{cases} \quad (2) \\ \pi(e) &= \frac{w(e)}{\sum_{e' \in \mathcal{E}} w(e')}. \end{aligned}$$

Here π is the PPR restart distribution. The three provenance levels preserve the implementation’s priority order, and all non-seed entities receive zero restart mass. Hyperparameters are fixed on a HotpotQA held-out subset and used unchanged on the other datasets. Together, complementary signals, intermediate-variable recovery, and provenance-weighted propagation produce the passages and entities used for chain construction.

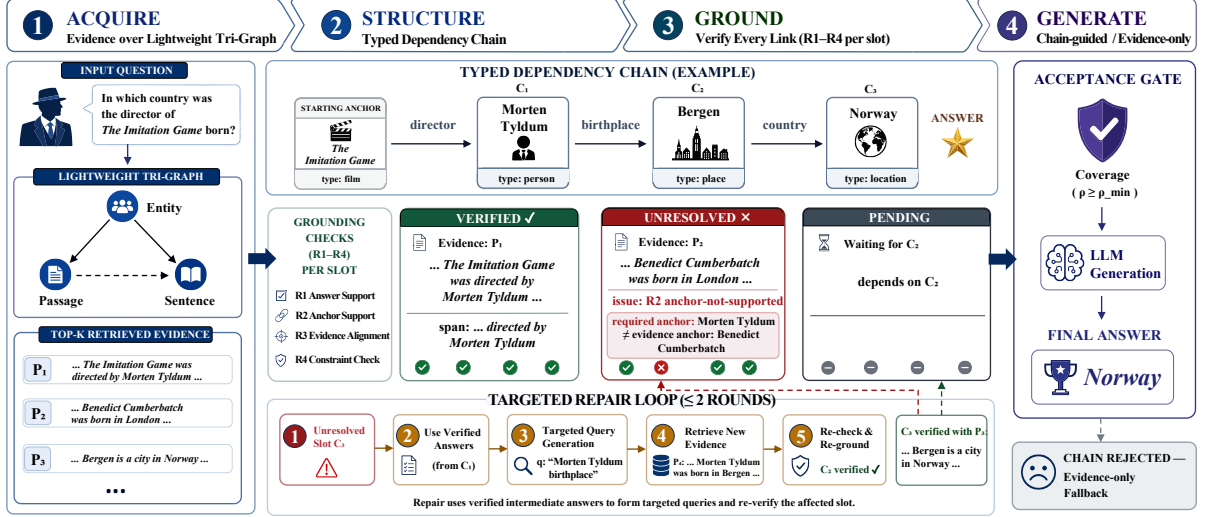


Figure 2: Overview of GroundRAG. Evidence acquisition over a lightweight tri-graph supplies passages and latent intermediate variables for a typed dependency chain. Grounding checks and targeted repair determine whether that chain can guide generation; otherwise GroundRAG generates from the retrieved evidence without the chain.

3.3 Constructing a Verifiable Reasoning Chain

Given the acquired evidence, a planner represents the reasoning process as typed slots $\mathbf{c} = (c_1, \dots, c_T)$. Here T is the actual number of planned slots, with $T \leq T_{\max}$, and each slot is

$$c_i = (a_i, r_i, \tau_i, \mathcal{D}_i). \quad (3)$$

Here a_i is the anchor, r_i the relation, τ_i the expected answer type, and $\mathcal{D}_i$ the set of predecessor slots. These correspond directly to the implementation fields anchor, relation, expected_type, and depends_on. The type vocabulary is finite, with $\tau_i \in \mathcal{T}$:

person	organization	place
location	date	number
entity	type	expanded-name

Dependencies are encoded with placeholders $\langle s_1 \rangle, \langle s_2 \rangle, \dots$ and parsed into a DAG. If $c_i \in \mathcal{D}_j$, the verified answer to c_i replaces the placeholder that anchors c_j . This makes the transfer of intermediate answers explicit and prevents a later step from silently switching to another entity in the context. The type field constrains the expected answer space and checks dependency substitution. The final slot is marked `is_final = True`, and its type must match the wh-form of q (e.g., “who” accepts $\{person, organization\}$); a mismatch triggers one plan-level repair. The chain is therefore an evidence-addressable structure rather than a free-form chain-of-thought trace.

3.4 Grounding, Repair, and Chain Acceptance

Slot verification. For each slot c_i , the verifier LLM receives candidate passages $\mathcal{C}_i$ from evidence acquisition and returns

$$v_i = (z_i, \hat{a}_i, d_i, \xi_i, \gamma_i), \quad (4)$$

where z_i is the status, $\hat{a}_i$ the proposed answer, d_i the cited passage identifier, ξ_i the evidence span, and γ_i the confidence. The status z_i is one of *verified*, *unresolved*, or *conflicting*. Whether the proposal is grounded is determined only by the subsequent deterministic recheck.

Deterministic grounding recheck. Every *verified* proposal is rechecked against its cited passage without another LLM call. Four reason codes identify failures. **(R1) answer-not-supported:** the proposed answer or required relation lacks lexical support in the cited text. **(R2) anchor-not-supported:** the concrete anchor tokens are not supported by the text. **(R3) evidence-not-in-passage:** the returned evidence cannot be aligned to the passage. **(R4) constraint-not-supported:** for applicable date or number slots, an explicit year constraint is not preserved by the supporting text. The implementation normalises case, whitespace, and punctuation and allows bounded token-overlap/fuzzy matching; it does not require raw-string identity. A failed slot is demoted to *unresolved* with its reason code. This local recheck is deterministic and non-circular with the verifier, but it is a structural grounding test rather than a general semantic-entailment model.

Targeted repair and re-verification. Let $\mathcal{U}$ be the unresolved slots. For each $c_i \in \mathcal{U}$, verified predecessor answers replace placeholders $\langle s_j \rangle$ in anchor_i and evidence_query_i . The resulting targeted query retrieves $k_{\text{tgt}} = 8$ additional sentences, after which the affected slots are verified and grounded again. We run at most $R_{\text{max}} = 2$ rounds. Substitution is type-checked: an answer of type *person*, for example, cannot fill a dependency that requires a *date*. Repair therefore follows dependencies already established by grounded steps instead of restarting from the original question.

Chain-coverage reranking. Let $\chi_i(p) = 1$ when passage p contains grounded evidence for slot c_i , and 0 otherwise. Candidate passages are re-ranked by

$$\text{score}(p) = w_{\text{base}} r_{\text{PPR}}(p) + w_{\text{cov}} \sum_{i=1}^T \chi_i(p), \quad (5)$$

where r_{PPR} is the converged evidence-acquisition score. We set $w_{\text{base}} = 1.0$ and $w_{\text{cov}} = 1.5$. This term favours a compact context that covers more of the chain while retaining the base ranking. The top $n = 8$ passages form the final context.

Conservative gate. After recheck, let $g_i = 1$ if slot c_i is both *verified* and grounded, and 0 otherwise. The grounded-slot rate is

$$\rho = \frac{1}{T} \sum_{i=1}^T g_i. \quad (6)$$

The implementation calls every planned slot whose dependency order is at or before that of the final slot a *critical dependency*; all such slots must be verified and grounded. A chain is accepted only when the final slot is grounded, every critical dependency passes this check, and $\rho \geq \rho_{\text{min}} = 0.67$. Otherwise the system falls back to generation from the evidence-acquisition context without the slot plan. The gate makes verification a condition on chain use rather than a demand that every proposed chain be accepted.

3.5 Verified Generation

When a chain passes the acceptance criterion, the generator receives q , the verified anchors, relations, types, intermediate answers, and the top- n supporting passages. The prompt renders the dependency chain as a numbered plan and constrains the response to the type of the final slot. If the chain

is rejected, the plan is omitted and the generator receives only q and the evidence-acquisition passages, matching the configuration without chain verification in Section 4.3. Verification thus gates chain-guided reasoning; it does not block evidence-only generation.

4 Experiments

We evaluate GroundRAG on three multi-hop QA benchmarks against baselines covering direct LLM inference, dense RAG, graph-based RAG, and decomposition-based RAG. The experiments address three questions. **Q1 (Overall accuracy):** How does GroundRAG compare with existing multi-hop RAG systems? **Q2 (Pipeline attribution):** How much do evidence acquisition for chain construction and grounded chain verification contribute? **Q3 (Verification necessity and behaviour):** Do errors remain when answer-bearing evidence has already been retrieved, and how does GroundRAG respond to unsupported intermediate steps? Cost and residual-error analyses provide supporting context. Full tables appear in Appendix A.

4.1 Experimental Setup

Datasets. We evaluate on HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (2Wiki) (Ho et al., 2020), and MuSiQue (Trivedi et al., 2022). The first two are mostly two-hop; MuSiQue contains up to four-hop chains and is substantially harder. Following HippoRAG (Gutiérrez et al., 2024) and LinearRAG (Zhuang et al., 2026), we sample 1,000 validation queries per dataset for a fair comparison. Corpus sizes are 1,311 / 2,309 / 1,354 passages.

Baselines. Four categories: (i) direct LLM inference (LLaMA-8B, LLaMA-13B, GPT-3.5-turbo, GPT-4o-mini); (ii) vanilla dense RAG with Top- $k \in \{1, 3, 5\}$; (iii) graph-based RAG—KGP (Wang et al., 2024), G-Retriever (He et al., 2024), RAPTOR (Sarathi et al., 2024), E²GraphRAG (Zhao et al., 2025), LightRAG (Guo et al., 2025), HippoRAG (Gutiérrez et al., 2024), HippoRAG 2 (Gutiérrez et al., 2025), GFM-RAG (Luo et al., 2025), and LinearRAG (Zhuang et al., 2026); and (iv) decomposition-based IR-CoT (Trivedi et al., 2023) and LogicRAG (Chen et al., 2026). We run the latter two with the same 1,000 questions, corpora, Top-5 evidence budget, and GPT-4o-mini backbone; IRCot uses its strict Top-5 profile on 2Wiki. We additionally evaluate GroundRAG without chain verification, retaining

Table 1: Main results (%). Best in **bold**, second-best underlined. † indicates GroundRAG significantly outperforms the strongest prior baseline at $p < 0.05$ ($n = 1,000$, two-proportion z -test).

Method	HotpotQA		2Wiki		MuSiQue	
	LLM	Contain	LLM	Contain	LLM	Contain
<i>Direct LLM inference</i>						
LLaMA-8B	27.3	31.1	16.2	33.6	8.1	7.4
LLaMA-13B	16.8	24.2	10.5	21.9	4.4	3.3
GPT-3.5-turbo	43.2	33.4	31.0	28.7	21.9	10.3
GPT-4o-mini	40.2	38.9	31.4	36.3	15.8	13.6
<i>Vanilla dense RAG</i>						
Top-1	49.1	46.3	31.7	36.6	21.1	17.8
Top-3	56.0	53.0	39.7	44.9	27.5	25.1
Top-5	58.6	55.7	43.0	48.6	29.6	26.1
<i>Graph / structured RAG</i>						
KGP	60.9	61.5	30.0	31.6	30.1	25.6
G-Retriever	40.6	42.2	27.1	46.6	15.5	14.4
RAPTOR	58.3	55.9	42.1	50.1	27.4	23.3
E ² GraphRAG	63.9	61.0	38.1	54.3	26.2	23.8
LightRAG	59.5	60.3	39.0	55.2	28.6	27.4
HippoRAG	59.3	57.0	59.9	66.1	24.1	29.3
GFM-RAG	65.6	62.7	59.6	66.8	34.6	29.9
HippoRAG 2	64.3	62.9	55.0	62.7	35.0	31.0
LinearRAG	<u>66.5</u>	64.3	<u>63.7</u>	<u>70.2</u>	<u>37.0</u>	<u>33.9</u>
<i>Decomposition-based RAG</i>						
IRCoT (Trivedi et al., 2023)	55.6	51.8	51.5	55.2	30.9	26.5
LogicRAG (Chen et al., 2026)	57.6	52.0	58.7	62.1	31.5	26.3
<i>Ours</i>						
GROUND-RAG	73.8 [†]	<u>63.1</u>	69.2 [†]	74.8	44.3 [†]	38.6

the same evidence-acquisition stage and evidence-only generation. Other published baseline numbers follow the shared HippoRAG/LinearRAG protocol.

Metrics. We report two metrics over $n = 1,000$ samples per dataset. **LLM-Acc** (primary): a judge LLM (GPT-4o-mini) decides semantic equivalence with the gold answer. **Contain-Acc**: the predicted answer string contains the gold after lower-casing and whitespace normalisation. For uncertainty and significance analyses, we estimate 95% confidence intervals by bootstrap with $N_{\text{boot}} = 10,000$ re-samples (seed 42) for our predictions and Wilson intervals for baselines without per-sample predictions. Significance between two methods is assessed by a two-proportion z -test; † marks results where GroundRAG significantly outperforms the strongest prior baseline at $p < 0.05$.

Implementation. GroundRAG uses all-mpnet-base-v2 (Song et al., 2020) for embeddings, GPT-4o-mini for latent-entity recovery, verification, and generation, and spaCy (Honnibal et al., 2020) for Tri-Graph NER. Hyperparameters are fixed across datasets: $\tau_s = 0.85$, $\alpha_{\text{hi}}/\alpha_{\text{md}}/\alpha_{\text{lo}} = 1.0/0.4/0.2$, $T_{\text{max}} = 4$, $R_{\text{max}} = 2$, $\rho_{\text{min}} = 0.67$, $w_{\text{cov}} = 1.5$, and $n = 8$. On 2Wiki, a lightweight GPT-4o-mini router predicts the four question types from question text alone with 97.4% accuracy. Questions predicted as compositional use GroundRAG, while the remaining types use the configuration without chain verification; this automatic routing yields the

reported 69.2% LLM-Acc.

4.2 Main Results (Q1)

Table 1 compares GroundRAG with four categories of baselines. Two observations follow.

Obs. 1: GroundRAG attains the highest LLM-Acc on all three datasets, with statistically significant gains over LinearRAG. GroundRAG reaches 73.8% / 69.2% / 44.3%, against LinearRAG’s 66.5 / 63.7 / 37.0, for gains of +7.3, +5.5, +7.3 ($p = 4 \times 10^{-4}$, 1×10^{-2} , 9×10^{-4}). The largest gain is on the hardest benchmark (MuSiQue), consistent with the hypothesis that RCH scales with reasoning depth: each additional hop is an additional opportunity for the planner or reader to drift, and verification matters more.

Obs. 2: HotpotQA Contain-Acc drops while LLM-Acc rises—a typed-answer length effect, not a regression. GroundRAG’s Contain-Acc on HotpotQA is 63.1, below the configuration without chain verification (69.5). Table 2 provides one explanation: typed slot generation shortens predicted answers from 26.3 tokens to 17.6, which is consistent with reduced incidental substring overlap. On MuSiQue, a similar shortening (40.9 $\rightarrow$ 24.3 tokens) co-occurs with a Contain-Acc *increase*, so answer length alone does not determine the metric change.

We therefore treat LLM-Acc as the more faithful indicator for multi-hop correctness; Contain-Acc

Table 2: Predicted vs. gold answer length (avg. tokens) and the corresponding Δ Contain-Acc relative to the configuration without chain verification.

Dataset	Evidence-only	GroundRAG	Δ	Gold	Δ Contain
HotpotQA	26.3	17.6	−33%	15.1	−6.4
2Wiki	27.6	23.5	−15%	14.1	+1.7
MuSiQue	40.9	24.3	−41%	17.0	+3.2

Table 3: Consecutive contributions to the grounded-chain pipeline (LLM-Acc, %).

Configuration	HotpotQA	2Wiki	MuSiQue
Basic graph evidence retrieval	62.2	60.5	33.2
+ Evidence acquisition for chain construction	70.1	66.5	40.3
+ Grounded chain verification (GroundRAG)	73.8	69.2	44.3

is reported for backward compatibility with prior work.

4.3 Component Attribution (Q2)

Obs. 3: Evidence acquisition and chain verification contribute consecutively. Starting from basic graph evidence retrieval, evidence acquisition raises LLM-Acc by 7.9/6.0/7.1 points on HotpotQA, 2Wiki, and MuSiQue. It supplies the passages and latent variables required to instantiate a multi-hop chain. Once that chain is formed, grounding, repair, and chain acceptance add a further 3.7/2.7/4.0 points. These increments should not be read as retrieval performance versus reasoning performance: the first determines whether the required intermediate variables can be instantiated, while the second determines whether those variables and their dependencies are grounded well enough to guide generation.

4.4 Additional Analyses

We first test whether verification remains useful after answer-bearing evidence has been retrieved, then report mechanism behaviour, retrieval coverage, cost, and residual errors.

Answer-bearing evidence does not eliminate chain errors. We retain questions for which at least one of the five passages supplied by the configuration without chain verification contains the normalised gold-answer string. This diagnostic is answer-string coverage, not official supporting-passage recall. It leaves 935 HotpotQA, 849 2Wiki, and 670 MuSiQue questions. Even on these subsets, the system without chain verification answers 29.1%, 31.4%, and 47.0% incorrectly. GroundRAG raises conditional LLM-Acc from 70.9% to 74.9% on HotpotQA and from 68.6% to

71.3% on 2Wiki; on MuSiQue the change is small (53.0% to 53.3%). Thus, answer availability alone does not ensure that the required dependencies are followed, although the size of the verification benefit varies by dataset.

Chain-verification behaviour. On HotpotQA, the LLM verifier marks 83% of slots *verified*, while 66% remain verified after deterministic grounding; the corresponding changes are 80% $\rightarrow$ 57% on the 2Wiki compositional subset and 70% $\rightarrow$ 49% on MuSiQue. The gap indicates that verifier confidence and structural evidence support are not interchangeable. The external recheck complements model self-evaluation by testing answer, anchor, span, and explicit-constraint support against cited text. The dominant reject reason also shifts across datasets: R1 leads on HotpotQA, R2 on 2Wiki, and R3 on MuSiQue. Appendix A.2 reports repair, coverage, fallback, and reason distributions.

Retrieval coverage. Under the implementation’s normalised gold-answer-string criterion, Recall@5 improves by +1.4, +1.1, and +4.7 points over the configuration without chain verification. The largest gain is on MuSiQue, where targeted repair issues more queries for deeper chains. Detailed Recall@{1, 3, 5, 10} values and the metric definition are in Appendix A.1.

Efficiency. *Obs. 6: The verification stage adds 16–19 $\times$ token cost over retrieval alone, translating to \$0.22–\$0.25 per newly-correct answer.* The trade favours applications where wrong answers carry downstream cost; the full cost decomposition is in Appendix A.3.

Residual errors and case study. *Obs. 7: A manual examination of 53 errors on MuSiQue indicates that roughly 1 in 5 stems from judge strictness*

(13.2%) or gold-label ambiguity (7.5%) rather than model behaviour, contextualising absolute accuracy numbers; the remaining 79.2% map onto the failure taxonomy of Section 3.4, with 39.6% falling under categories that grounded chain verification addresses. Appendix A.5 provides illustrative cases of two complementary chain-grounding failures: an anchor mismatch that propagates through dependent slots, and an unsupported intermediate answer that is rejected before downstream reasoning.

5 Conclusion

We introduced GroundRAG to address Reasoning Chain Hallucination in multi-hop RAG. GroundRAG follows R-V-G by requiring retrieved evidence to pass explicit chain-level grounding before it can guide generation. Evidence acquisition supplies the entities and passages needed to construct the chain; grounding and targeted repair determine whether the resulting chain is supported well enough to use. When it is not, GroundRAG generates from the retrieved evidence without forcing an unsupported reasoning path.

GroundRAG reaches 73.8%, 69.2%, and 44.3% LLM-judged accuracy on HotpotQA, 2WikiMultiHopQA, and MuSiQue. Our analysis identifies substantial errors even when answer-bearing passages are already available, as well as many verifier-accepted slots that fail explicit grounding checks. These results suggest that multi-hop RAG should be evaluated not only by whether it retrieves answer-bearing evidence, but also by whether the dependencies built from that evidence remain grounded.

Limitations

GroundRAG has three main limitations. **(1) Planner and verifier dependence.** The typed chain is produced by GPT-4o-mini, so an incorrect plan can prevent otherwise useful evidence from being connected. The deterministic recheck tests structural lexical support; it is not a general semantic-entailment model and may reject valid paraphrases or accept a passage with the right words but the wrong relation.

(2) Retrieval diagnostic. Our Recall@ k analysis treats a passage as relevant when it contains the normalised gold-answer string. This supports the answer-present analysis in Section 4.4, but it is not official supporting-passage recall and does not

show that every required fact has been retrieved.

(3) Verification cost. Planning, slot verification, and repair add substantial inference cost. Distilling reason-code prediction and grounding decisions into a smaller specialised verifier is a promising way to reduce this overhead while retaining an external check on the generated chain.

Acknowledgments

This work is supported by the National Key R&D Program of China (2024YFF0907000) and is also a research achievement of the Key Laboratory of Digital Right Services of the National Press and Publication Administration and the Key Laboratory of Interactive Technology and Experience System, Ministry of Culture and Tourism.

References

- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. [Self-rag: Learning to retrieve, generate, and critique through self-reflection](#). In *The Twelfth International Conference on Learning Representations*.
- Huajun Chen. 2024. [Large knowledge model: Perspectives and challenges](#). *Data Intelligence*, 6(3):587–620.
- Shengyuan Chen, Chuang Zhou, Zheng Yuan, Qinggang Zhang, Zeyang Cui, Hao Chen, Yilin Xiao, Jiannong Cao, and Xiao Huang. 2026. [You don’t need pre-built graphs for RAG: Retrieval augmented generation with adaptive reasoning structures](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40(36), pages 30270–30278.
- Xuanzhi Chen, Yajing Xu, Wei Han, Lei Luo, and Lei Yang. 2025. [ReD-RAG: Rewritable decomposition-guided iterative retrieval-augmented generation for open-domain multi-hop reasoning](#). In *Natural Language Processing and Chinese Computing*, volume 16104 of *Lecture Notes in Computer Science*, pages 120–132. Springer Nature.
- Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.
- Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2025. Lightrag: Simple and fast retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10746–10761. Association for Computational Linguistics.
- Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large

- language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. [From rag to memory: Non-parametric continual learning for large language models](#). In *Forty-second International Conference on Machine Learning*.
- Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37:132876–132907.
- Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. [Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625. International Committee on Computational Linguistics.
- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. [spacy: Industrial-strength natural language processing in python](#).
- Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards mitigating llm hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843.
- 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 *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS ’20*. Curran Associates Inc.
- Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Dinh Phung, Chen Gong, and Shirui Pan. 2025. Gfm-rag: Graph foundation model for retrieval augmented generation. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*. Curran Associates, Inc.
- Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiaapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.
- Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*.
- Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. [Musique: Multi-hop questions via single-hop question composition](#). *Trans. Assoc. Comput. Linguistics*, 10:539–554.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. [Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.
- Yu Wang, Nedim Lipka, Ryan A Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. 2024. Knowledge graph prompting for multi-document question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19206–19214.
- Zhuo Wang, Guixuan Zhang, Jiayang Yu, and Shuwu Zhang. 2026. [HoneyMark: Protecting data integrity in RAG via synthetic trap documents and post-hoc watermarking](#). *Data Intelligence*. Advance online publication.
- Yilin Xiao, Junnan Dong, Chuang Zhou, Su Dong, Qianwen Zhang, Di Yin, Xing Sun, and Xiao Huang. 2026. Graphrag-bench: Challenging domain-specific reasoning for evaluating graph retrieval-augmented generation. In *The Fourteenth International Conference on Learning Representations (ICLR 2026)*.
- Yilin Xiao, Chuang Zhou, Yujing Zhang, Qinggang Zhang, Su Dong, Shengyuan Chen, Chang Yang, and Xiao Huang. 2025. [LAG: Logic-augmented generation from a cartesian perspective](#). *arXiv preprint arXiv:2508.05509*.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [Hotpotqa: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380. Association for Computational Linguistics.
- Yibo Zhao et al. 2025. [E²GraphRAG: Streamlining graph-based RAG for high efficiency and effectiveness](#). *Preprint*, arXiv:2505.24226.
- Rongzhi Zhu, Xiangyu Liu, Zequn Sun, Yiwei Wang, and Wei Hu. 2025. [Mitigating lost-in-retrieval problems in retrieval augmented multi-hop question answering](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*.
- Luyao Zhuang, Shengyuan Chen, Yilin Xiao, Huachi Zhou, Yujing Zhang, Hao Chen, Qinggang Zhang, and Xiao Huang. 2026. Linearrag: Linear graph retrieval augmented generation on large-scale corpora. In *The Fourteenth International Conference on Learning Representations (ICLR 2026)*.

A Additional Experimental Analyses

This appendix reports the full tables and analyses summarised in Section 4.4.

A.1 Answer-String Retrieval Coverage

Recall@5 improves on all three benchmarks (+1.4, +1.1, +4.7), with the largest gain on the deepest dataset, consistent with targeted repair issuing more queries on harder chains. The Recall@1 drop on HotpotQA and MuSiQue is also consistent with chain-coverage reranking (Section 3.4), which can move a passage covering several slots ahead of a locally stronger single-slot passage. We use this metric only as an answer-bearing-context diagnostic; it does not establish that all gold supporting passages have been retrieved.

A.2 Grounded Chain Verification Behaviour

Obs. 4: Verifier approval and passage grounding differ substantially. On HotpotQA, 83% of slots receive a *verified* status from the LLM verifier, but 66% remain verified after grounding recheck. The gap widens on harder corpora (2Wiki compositional: 80 $\rightarrow$ 57; MuSiQue: 70 $\rightarrow$ 49). These rates show that a plausible answer and justification can still fail a structural check against the cited passage. The result motivates using an external deterministic check alongside model self-evaluation; it does not claim to replace semantic-entailment verification.

Obs. 5: The dominant grounding-reject reason shifts across datasets. On HotpotQA, R1 answer-not-supported dominates (437 occurrences), followed by R2 anchor-not-supported (225). On the 2Wiki compositional subset, R2 leads (154), ahead of R1 (61). On MuSiQue, R3 evidence-not-in-passage dominates (408), ahead of R1 (257). The changing modal reason suggests that RCH is a structured, dataset-dependent failure family rather than a single undifferentiated error category.

A.3 Cost Decomposition

Obs. 6: GroundRAG is 16–19 $\times$ more expensive than retrieval alone in raw tokens, but \$0.22–\$0.25 per newly-correct answer. The verification-and-repair pipeline runs additional LLM calls for planning, verification, and (when triggered) repair. We do not contest the absolute cost—it is the price of running an LLM as a verifier rather than only as a generator. The question is whether the cost is well-spent. On HotpotQA,

GroundRAG answers 37 additional questions correctly for an additional \$7.98, i.e. \$0.22 per newly-correct answer; on MuSiQue, 40 additional correct answers for \$10.10, i.e. \$0.25. For applications where wrong answers carry downstream cost, this trade is favourable; for throughput-bound deployments it is not.

A.4 Error Analysis on MuSiQue

Obs. 7: Roughly 1 in 5 remaining MuSiQue errors stems from sources outside model reasoning, informing how absolute accuracy numbers should be interpreted. We manually examined the 53 errors GroundRAG makes on a uniform random sample of 100 MuSiQue questions. Table 7 maps the annotated classes onto the failure taxonomy from Sections 3.4 and 3.2. Two patterns emerge. First, 20.8% of remaining failures arise from sources unrelated to the model: judge strictness on paraphrased answers (13.2%) and gold-label ambiguity in MuSiQue itself (7.5%). This proportion contextualises absolute numbers on MuSiQue—a substantial share of the gap to 100% reflects evaluation artifacts rather than model behaviour. Second, 39.6% of errors fall under an RCH category that grounded chain verification is designed to address, consistent with the gate’s fallback rate of $\sim 43\%$ when the MuSiQue chain cannot be grounded.

A.5 Illustrative Case Studies of Reasoning Chain Hallucination

A.5.1 Anchor Drift and Cascading Dependency Error

This illustrative case extends Figure 1 by one dependency. The question asks, “*What is the capital of the country where the director of The Imitation Game was born?*” The intended schema is Film $\rightarrow$ Director $\rightarrow$ Birthplace $\rightarrow$ Country $\rightarrow$ Capital, with gold answer *Oslo*. Table 8 shows how an incorrect intermediate binding can propagate through later relations even when every downstream fact is locally supported.

Anchor drift is not confined to the affected slot. Once an incorrect intermediate value is substituted into downstream dependencies, later relations can remain individually well supported while the overall reasoning chain becomes incorrect. Here c_3 directly depends on c_2 , and c_4 depends on it indirectly; repairing c_2 prevents the error from propagating through both.

Table 4: Recall@ k under the implementation’s normalised gold-answer-string criterion. A retrieved passage is counted as relevant when it contains the normalised gold answer; this is not official supporting-passage recall.

Dataset / Method		R@1	R@3	R@5	R@10
HotpotQA	Evidence-only	36.7	80.5	93.5	93.5
HotpotQA	GroundRAG	26.6	74.3	94.9	95.2
2Wiki	Evidence-only	42.1	74.6	84.9	84.9
2Wiki	GroundRAG	42.4	77.4	86.0	86.1
MuSiQue	Evidence-only	19.9	52.4	67.0	67.0
MuSiQue	GroundRAG	15.3	52.4	71.7	73.6

Table 5: Internal behavioural metrics of grounded chain verification. 2Wiki numbers are over the $n = 413$ compositional subset, since verification is invoked only on that question type. Other columns are over $n = 1,000$.

Metric	HotpotQA	2Wiki (comp.)	MuSiQue
Avg. slots per question	2.32	2.00	2.40
Verified slot rate (pre-grounding)	0.83	0.80	0.70
after grounding reject	0.66	0.57	0.49
Repair invocation rate	0.42	0.49	0.61
Avg. repair rounds (when invoked)	0.81	0.95	1.16
Final slot coverage rate	0.81	0.75	0.64
Conservative-gate fallback rate	0.23	0.38	0.43

A.5.2 Unsupported Intermediate Slot

A contrasting illustrative R1 case asks, “Which country is the composer of *The Imitation Game* from?”, with gold answer *France*. A cited film-context passage states that *The Imitation Game* is a 2014 historical drama directed by Morten Tyldum and starring Benedict Cumberbatch, so it supports the anchor but does not identify the composer. For $c_1 = (\text{film}, \text{composer}, \text{person})$, the model may propose the plausible name *Hans Zimmer* while citing this passage. Because neither that answer nor the required composer relation is supported, the deterministic recheck returns **R1: answer-not-supported**. Targeted retrieval then finds that the score was composed by *Alexandre Desplat*; a second passage identifies him as a French composer, grounding c_1 and $c_2 = (\langle s_1 \rangle, \text{nationality}, \text{location})$

and yielding *France* ✓.

Obs. 8: Reasoning-chain errors can arise either from incorrect dependency binding or from unsupported intermediate answers. In the anchor-drift example, locally valid evidence is attached to a competing person entity, and the resulting value propagates through downstream dependencies unless intercepted. In the R1 contrast case, the anchor is preserved but the proposed intermediate answer itself lacks support in the cited passage. GroundRAG addresses both at the slot level: R2 detects mismatched anchors, while R1 rejects unsupported answers or relations before they are allowed to guide later reasoning. These illustrative cases do not exhaust the RCH taxonomy; R3 and R4 remain defined and evaluated separately.

Table 6: Cost. Costs are calculated using the GPT-4o-mini API pricing available on May 21, 2026 (\$0.15/\$0.60 per million input/output tokens). “Cost / new-correct” = (cost difference vs. the configuration without chain verification) / (additional correct answers under LLM-Acc).

Dataset / Method		Prompt	Completion	Total \$	Cost / new-correct
HotpotQA	Evidence-only	2.85M	175K	0.53	–
HotpotQA	GroundRAG	53.6M	764K	8.51	\$0.22
MuSiQue	Evidence-only	2.87M	212K	0.56	–
MuSiQue	GroundRAG	67.4M	924K	10.66	\$0.25

Table 7: Failure taxonomy on MuSiQue (53 erroneous answers in a uniform random sample of 100 questions). “Addressable by” indicates which component of the system would resolve the failure if it worked perfectly.

Annotated class	Maps to	Count	% of errors
Incomplete evidence acquisition	Evidence acquisition	17	32.1
Reader hallucination	R1 answer-not-supported	10	18.9
Wrong bridge entity	R2 anchor-not-supported	4	7.5
Planning failure	Typed-slot / anti-compression	7	13.2
Partial recall / mis-ranked	Chain-coverage rerank	4	7.5
Evaluation noise	— (LLM-judge limitation)	7	13.2
Annotation ambiguity	— (dataset issue)	4	7.5
Model-addressable subtotal		42	79.2
Annotation / metric noise (non-model)		11	20.8

Table 8: Illustrative case study of cascading anchor error. A locally valid birthplace fact is initially attached to the wrong intermediate person; without dependency verification, the resulting value propagates through subsequent slots. GroundRAG detects the anchor mismatch, performs targeted repair, and prevents the error from contaminating downstream dependencies.

Question	<i>What is the capital of the country where the director of <i>The Imitation Game</i> was born?</i>
Gold Answer	Oslo
Retrieved evidence	P1: “... <i>The Imitation Game</i> was directed by <i>Morten Tyldum</i> and stars <i>Benedict Cumberbatch</i> ...” P2: “... <i>Morten Tyldum</i> was born in <i>Bergen</i> ...” P3: “... <i>Bergen</i> is a city in <i>Norway</i> ...” P4: “... <i>Benedict Cumberbatch</i> was born in <i>London</i> ...” P5: “... the capital of <i>Norway</i> is <i>Oslo</i> ...” P6: “... <i>London</i> is the capital of the <i>United Kingdom</i> ...”
Intended schema	Film → Director → Birthplace → Country → Capital
Without Chain Verification	
Dependency path	<i>The Imitation Game</i> → <i>Morten Tyldum</i> ✓ → anchor drift to Benedict Cumberbatch → London → United Kingdom → London
Prediction	× <i>London</i> ; the downstream facts are locally valid, but they inherit the wrong intermediate person
GroundRAG	
Typed plan	$c_1 = (\text{film}, \text{director}, \text{person}, \emptyset)$ $c_2 = (\langle s_1 \rangle, \text{birthplace}, \text{place}, \{c_1\})$ $c_3 = (\langle s_2 \rangle, \text{country}, \text{location}, \{c_2\})$ $c_4 = (\langle s_3 \rangle, \text{capital}, \text{place}, \{c_3\})$, with c_4 final
c_1 verified	<i>The Imitation Game</i> → <i>Morten Tyldum</i> ✓
c_2 initial proposal	$\langle s_1 \rangle = \text{Morten Tyldum}$, but proposes <i>London</i> from P4
Grounding decision	R2: anchor-not-supported ; P4 supports <i>Benedict Cumberbatch</i> → <i>London</i> , not <i>Morten Tyldum</i> → birthplace
Targeted repair	Substitute the verified c_1 answer and retrieve “ <i>Morten Tyldum</i> + birthplace”
c_2 verified	<i>Morten Tyldum</i> → <i>Bergen</i> ✓
c_3 verified	<i>Bergen</i> → <i>Norway</i> ✓
c_4 verified	<i>Norway</i> → <i>Oslo</i> ✓
Critical dependencies	c_3 directly and c_4 indirectly depend on c_2 ; all are grounded after repair
Gate (ρ)	$4/4 = 1.00 \geq \rho_{\min} = 0.67$; critical dependencies grounded; chain accepted
Prediction	✓ <i>Oslo</i>