The Bottleneck Is Retrieval, Not Reading: Diagnosing Evidence-based Generation with RAG for Long-Context Question Answering

Anonymous ACL submission

Abstract

Retrieval-augmented generation (RAG) is the dominant paradigm for grounding large language models in external corpora, but its similarity-driven, chunked pipeline has structural blind spots: it cannot anticipate which combination of passages a question requires, and it discards the document-level context that resolves cross-passage references. We introduce *evidence-based generation* (EBG), a framework that requires models to return both an answer and line-level citations, together with a joint `Ans+Evid` metric that scores correctness and grounding simultaneously. We reformulate two existing corpora into evidence-grounded multi-document QA benchmarks - MULTIHOP-RAG and QASPER-MP, a multi-paper adaptation of QASPER with realistic short paper cues, and pair them with a controlled synthetic two-hop haystack and an evidence-augmented version of NoLiMa spanning 10K–1M characters. Across ten frontier LLMs, EBG isolates two concrete RAG limitations that long-context models avoid: similarity retrieval surfaces *any* relevant chunk far more reliably than *all* required chunks for multi-fact queries, and chunking destroys document meta-information needed to resolve identity-bearing queries. EBG further exposes grounding failures invisible to answer-only evaluation, including incomplete citations in two-needle settings and a widening gap between `Ans` and `Ans+Evid`, when context contradicts parametric knowledge.

1 Introduction

Large Language Models (LLMs) can rapidly process vast amounts of text and identify the information most relevant to a given query. This capability suits applications such as legal discovery, medical record analysis, financial due diligence, literature review, and technical support, all of which require synthesizing targeted answers from evidence distributed across large document collections.

The predominant paradigm for grounding LLM outputs in external knowledge is retrieval-augmented generation (RAG; Lewis et al., 2020), which retrieves passages via embedding similarity and conditions the generator on those passages alone. While effective for *single-fact lookups*, this design has several structural limitations. First, each retrieved chunk is stripped of its surrounding context: the model cannot see where a paragraph falls within a document or how it relates to neighboring sections. Second, document-level metadata (authorship, date, section hierarchy) is typically lost during chunking. Third, and most critically, similarity-based retrieval cannot anticipate which combination of passages will be needed for questions requiring integration across multiple locations. This is particularly acute for *multi-hop questions*, where the answer depends on chaining facts that may appear in distant, semantically dissimilar passages (Karpukhin et al., 2020; Li et al., 2024).

Our goal in this paper is not to advocate replacing RAG with long-context (LC) models outright, but to make these structural limitations measurable. RAG systems are typically evaluated on answer correctness alone, which conflates two very different failure modes: the retriever fails to surface the right passages, or the reader fails to use them. To separate the two, we propose *evidence-based generation (EBG)*, a paradigm where the model must produce both an answer and explicit references to the input lines that support it. EBG inherits RAG's built-in verifiability (Lewis et al., 2020), but extends it to long-context settings where the model is given the full document neighborhood and must *select* its own evidence. The joint `Ans+Evid` metric pinpoints where a pipeline breaks: in retrieval, in grounding, or in reading. Recent work shows LC models can match or exceed RAG accuracy given sufficient context (Li et al., 2024; Laitenberger et al., 2025), but without provenance these comparisons are inconclusive; EBG closes that gap.

We make the following contributions:

- **EBG framework.** We define the evidence-based generation task and a joint `Ans+Evid` metric, and evaluate ten frontier LLMs under matched LC and RAG pipelines across 10K–1M character contexts.
- **Evidence-grounded benchmarks.** We reformulate two corpora into evidence-grounded multi-document QA benchmarks—a re-instrumented MULTIHOP-RAG with line-level evidence labels, and QASPER-MP, a multi-paper QASPER adaptation with realistic paper cues, and pair them with a controlled two-hop synthetic haystack and an evidence-augmented NOLIMA. The first two stress retrieval; the last two stress long-context reading.
- **Characterizing RAG's limitations.** Using EBG, we isolate two failure modes of similarity-based RAG that LC sidesteps: (i) retrieval recovers *any* relevant chunk far more reliably than *all* required chunks for multi-fact queries, and (ii) chunking discards document meta-information that identity-bearing queries depend on. On MULTIHOP-RAG, the retrieval ceiling at realistic top-k budgets, not the reader, is the binding constraint.
- **Diagnostic analysis of LC grounding.** Beyond the RAG comparison, EBG reveals that citation acts as a reasoning scaffold, that models cite incomplete evidence in two-needle settings, and that grounding degrades sharply when in-context evidence contradicts parametric knowledge.

2 Related Work

2.1 Long-Context Language Models

Advances in transformer architectures have expanded context windows from thousands to hundreds of thousands of tokens through improved positional encoding such as ALiBi (Press et al., 2022), RoPE extensions (Su et al., 2024), and memory-augmented attention (Wu et al., 2022). Training data matters equally, as pretraining corpora rarely contain long-range dependencies; Zhuang et al. (2025) construct data with long-distance references encouraging use of distant context. Yet larger windows do not automatically yield better reasoning. Du et al. (2025) show accuracy degrades with context length even under perfect retrieval, suggesting a processing bottleneck. Similarly, Liu et al. (2024) show performance drops when information appears mid-input rather than at boundaries.

2.2 Evaluating Long-Context Reasoning

Early long-context evaluations relied on needle-in-a-haystack tests (Kamradt, 2023), which mainly measure retrieval of planted information rather than reasoning. Subsequent work argues such evaluations set the bar too low, since strong performance often reflects shallow heuristics rather than genuine understanding (Hsieh et al., 2024; Liu et al., 2024; Schuster et al., 2025; Modarressi et al., 2025; Zhang et al., 2024; Yen et al., 2025). Recent benchmarks shift toward more demanding evaluations: ∞ Bench (Zhang et al., 2024) studies degradation beyond 100K tokens, HELMET (Yen et al., 2025) shows benchmark choice substantially alters rankings, and NoLiMa (Modarressi et al., 2025) removes lexical shortcuts by requiring semantic matching between queries and evidence. However, these benchmarks still score models on final answers alone, leaving open whether correct answers stem from actual use of relevant context or from priors, surface cues, or chance. Unlike these answer-only benchmarks, our work focuses on grounding by requiring models to return both answers and supporting input lines, distinguishing genuine reasoning from lucky guessing.

2.3 Retrieval-Augmented Generation

Retrieval-augmented generation (RAG) (Lewis et al., 2020) remains the dominant approach for grounding LLMs in external knowledge. Built on dense retrieval (Karpukhin et al., 2020) and multi-passage readers such as Fusion-in-Decoder (Izacard and Grave, 2021), RAG provides provenance by construction since retrieved passages serve as evidence. Recent work revisits RAG alongside long-context models, highlighting a trade-off between the two paradigms (Li et al., 2024; Laitenberger et al., 2025): long-context prompting generally improves accuracy but at substantially higher cost, motivating hybrid systems that route queries by difficulty (Li et al., 2024). Controlled comparisons also show well-tuned single-stage RAG can match elaborate multi-stage pipelines once token budgets are equalized (Laitenberger et al., 2025), while adaptive retrieval and self-reflection improve factuality and citation quality (Asai et al., 2024). Existing work primarily evaluates these approaches on accuracy or efficiency. We instead focus on grounding, distinguishing whether evidence was available from whether the model actually used it.

3 Evidence-based Generation (EBG)

We formalize **evidence-based generation (EBG)** as a grounded question-answering paradigm that extends standard answer-only evaluation by requiring models to generate both answers and explicit citations to the supporting input evidence.

3.1 Answer-only vs Evidence-based Generation

Existing long-context evaluations, including NoLiMa (Modarressi et al., 2025), use answer-only evaluation, omitting the grounding needed to validate model reasoning. EBG extends this formulation by requiring models to generate both explicit **evidence** references to supporting input lines and the final **answer**, yielding a stricter evaluation setting in which correctness depends on both accurate reasoning and faithful grounding. We evaluate this using the `Ans`, `Evid`, and `Ans + Evid` metrics, with `Ans + Evid` serving as the primary measure of evidence-based reasoning.

To illustrate the distinction, consider a haystack containing the sentence *"Then Stuart mentioned that he has been vegan for years."* at line 1,284 and the question *"Which character cannot eat Brandade?"* - a two-hop query requiring the inference that *Brandade* contains *fish* and is therefore incompatible with *veganism*. The answer-only and EBG settings differ in both the system instruction and the required output format.

Answer-only Prompt (User)

```
You will answer a question based on the following book snippet:
{haystack}
Use the information provided in the book snippet to answer the question. Your answer should be short and based on either explicitly stated facts or strong, logical inferences.
Question: Which character cannot eat Brandade?
Return only the final answer with no additional explanation or reasoning.
```

Model Response

```
Stuart
```

EBG Prompt (System + User)

```
[system] Your job is to answer the question entirely from the context and provide a reference. Your answer should cite all lines the answer is based on.
[user] <context>{haystack}</context>
Answer the question based on information only from the context. If the question is not answerable from the context, answer NA. Your response should comprise only the answer and all lines the answer is based on in json format. For example:
{"lines": [25, 412],"answer": "John"}
Question: Which character cannot eat Brandade?
```

Model Response

```
{"lines": [1284], "answer": "Stuart"}
```

Under answer-only evaluation, the response "`Stuart`" is simply marked correct. Under EBG, the same response is evaluated using three complementary metrics: answer accuracy (`Ans`), evidence accuracy (`Evid`), and their conjunction (`Ans + Evid`). While `Ans` preserves standard answer correctness, `Ans + Evid` additionally requires the model to cite the supporting evidence at line 1,284. This decomposition exposes silent failures hidden by answer-only evaluation: a model may arrive at the correct answer while citing irrelevant lines, or retrieve the correct evidence while producing the wrong answer. The joint `Ans + Evid` metric therefore serves as the strictest measure of evidence-based reasoning, requiring both accurate reasoning and faithful grounding.

3.2 RAG under EBG

In this work, we evaluate RAG systems *within the EBG framework* by requiring models to not only answer using retrieved chunks, but also to explicitly cite the supporting evidence. This yields two RAG variants:

- **Answer-only RAG:** Standard formulation requiring answer correctness only.
- **Answer + Evidence RAG:** EBG formulation requiring correct answer and correct citation.

This unified view allows us to decompose end-to-end performance into two factors:

- **Evidence availability:** whether the retriever surfaces the relevant needle(s) (Hit@k),
- **Evidence utilization:** whether the model correctly uses and cites the retrieved evidence (long-context accuracy).

Overall, our formulation shows that RAG and EBG are not competing paradigms but **complementary components**: retrieval determines *what evidence is available*, while EBG determines *whether that evidence is correctly used and cited*.

4 Benchmarks

To systematically evaluate EBG and RAG under EBG, we construct three complementary benchmarks: a fully synthetic benchmark, an evidence-augmented version of NoLiMa, and MultiHop-RAG. Together, they evaluate EBG under both controlled synthetic settings and realistic long-context scenarios.

4.1 Synthetic Dataset

The synthetic benchmark provides precise control over haystack content and evidence placement. It consists of a distractor fact pool, a curated needle set, and a haystack generation procedure.

Distractor fact pool. We construct a fixed pool of **66,834 unique single-sentence facts** using iterative LLM prompting followed by deduplication, ensuring that distractors are not confusable with inserted needles.

Needle set. We manually curate **10 two-hop needles**, each containing: i) two *atomic* supporting facts, ii) a compositional question requiring *both* facts, and iii) accepted answer surface forms. The needles span relations such as *compan–product* associations, *founder–specialty* links, and *birthplace* lookups.

Haystack assembly. For a target length L, distractor facts are shuffled and accumulated until reaching L characters, after which the two needle facts are inserted at independently sampled positions and recorded as gold evidence. Haystacks are formatted as line-numbered `i: fact` entries and evaluated under the EBG prompt template (Section A). Each trial uses a newly generated haystack and a uniformly sampled needle. Appendix B.1 provides a representative example.

Sanity check. At L=10K (≈2K tokens), all evaluated frontier models achieve at least 0.98 joint `Ans+Evid` accuracy over N=100 trials, confirming that the inserted needles are unambiguous. Any degradation at larger L therefore reflects failures in long-context processing.

4.2 NoLiMa-based Dataset

We adapt the NoLiMa haystack construction procedure to create long-context inputs ranging from L=10K to 1M characters and require models to both answer questions and cite supporting needles as evidence. This converts NoLiMa from an answer-only benchmark into an **evidence-grounded evaluation** that tests whether models truly use the provided context.

Needle set. We use the original **single-needle** (SN) setting from NoLiMa (Modarressi et al., 2025), specifically the `0402` and `0405` needle groups with both default and inverse variants under the *commonsense* category. Each needle contains a character name and a single factual attribute, and supports both one-hop and two-hop questions.

We further introduce a **two-needle** (*TN*) setting, in which the *explicit secondary needle* is consistent with world knowledge, and a **contradictory two-needle** (*CTN*) variant, in which it conflicts with world knowledge. Appendix C.1 provides additional details and templates for *SN*, *TN*, and *CTN* settings.

Haystack assembly. We adopt the random book generation procedure from NoLiMa (Modarressi et al., 2025) to construct haystacks ranging from 10K to 1M characters (10K–50K in 10K increments and 100K–1M in 100K increments), generating five random books $B_1^c, \ldots, B_5^c$ for each context length c. Because tokenization is model-dependent, we vary input length in characters rather than tokens. Each haystack is split into line-indexed entries using the '\n' delimiter. In the single-needle setting, the needle is inserted at one of four normalized depths: 0, 0.33, 0.67, or 1.0. In the two-needle setting, the primary needle is fixed at depth 0.5, while the secondary needle is inserted at one of the same four depths.

4.3 Multi-document QA Datasets

To stress the retrieval pipeline on the dimensions it is meant to handle, we reformulate two existing corpora into evidence-grounded multi-document QA benchmarks. Both are designed to make RAG's retrieval-vs.-reading split observable: gold evidence is labeled at the line (sentence or paragraph) level, so the same scoring code works for LC and any retriever-ordered context.

MULTIHOP-RAG (Tang and Yang, 2024). The official release contains **609 news articles** (4.6M characters, 1.2M tokens with the `cl100k_base` tokenizer) and 2,556 queries in four templated types: *inference*, *comparison*, *temporal*, and `null_query` (unanswerable). Each non-null query carries an `evidence_list` of **2–4 gold facts**, each consisting of a source article title and an exact textual

span. Removing the 301 `null_query` items leaves **2,255 usable queries**; for the pilot reported in this paper we uniformly sample 50 (seed 0), naturally stratified across the three answerable types (n_{inf}=13, n_{comp}=21, n_{temp}=16). For the LC condition we build per-query *mini-haystacks* containing every gold-evidence article plus 20 distractors sampled from the remainder of the corpus, concatenated in random order and split into sentence-per-line entries (average ∼241K characters). For the RAG condition we chunk the full corpus into 256-token sliding windows (6,296 chunks), embed with `text-embedding-3-large`, and use cosine similarity to produce top-K contexts at $K \in \{4, 10, 25, 50, 100, 200\}$. In both cases gold facts are re-mapped to line numbers inside the assembled context, so a fact whose source was not retrieved simply has no corresponding gold line. Appendix D reports the retrieval ceilings (Hits@k for *any* vs. *all* gold facts) and reproduces the answer-only RAG baselines from the original paper.

QASPER-MP. QASPER (Dasigi et al., 2021) contains 5,049 information-seeking questions over 1,585 NLP papers, each annotated by domain experts with paragraph-level evidence and a short free-form or extractive answer (including `Unanswerable`). The original task is single-document and therefore does not exercise the retrieval pipeline. We construct **QASPER-MP** (multi-paper) by pairing each gold paper with **ten distractor papers** sampled from the QASPER pool, yielding an 11-paper haystack per query, and we rewrite every question to include a short, realistic paper cue—in ∼50% of cases a first-author surname (e.g., "Smith et al."), in the remaining cases a 2–3 word verbatim fragment of the paper title—deterministically split by a hash of the question id. This forces the system to resolve document identity from a cue a user would actually type, rather than from the full paper title used in the original benchmark. For LC, papers are serialized in an XML-like format that preserves titles, abstracts, and section structure (`<paper>`, `<title>`, `<abstract>`, `<section>`); paragraphs are line-numbered, and the same paragraph-level gold labels are used for evidence scoring. For RAG, we evaluate both a *flat* variant (concatenate the top-K retrieved paragraphs in rank order) and a *structured* variant that re-emits the surrounding `<paper>`/`<title>`/`<abstract>` scaffolding and inserts "..." markers for discontinuities, so that retrieved fragments retain their document provenance. We report a 38-query pilot (the subset with non-yes/no, non-`Unanswerable` gold labels) under both an answerable-only and an aggregate breakdown. Construction details, prompts, and the LLM-judge protocol are in Appendix E.

5 Experiments

We organize our experiments around four research questions:

- **RQ1.** *What are the practical limitations of RAG, and can long-context EBG address them?*
- **RQ2.** *Does requiring explicitly citing evidence improve answer accuracy?*
- **RQ3.** *Can models correctly cite evidence lines even in a two-needle setting?*
- **RQ4.** *Can models remain context-faithful even when evidence contradicts their built-in knowledge?*

5.1 Experimental Setup

We evaluate 10 state-of-the-art proprietary models from three major providers across different experiments: `GPT-4O`, `GPT-4.1`, `o3-mini`, `GPT-5`, `GPT-5.5`, `Sonnet-4`, `Sonnet-4.5`, `Opus-4.6`, `Gemini-2.5-Fl`, and `Gemini-3-Fl`. We evaluate evidence-based generation (*EBG*) under both long-context (*LC*) and *RAG* settings. Prompt templates for the *answer-only* and *EBG* conditions are provided in Appendix A. Under *answer-only* prompting, we report answer accuracy (`Ans`*); under *EBG*, we report answer (`Ans`), evidence (`Evid`), and joint answer-plus-evidence (`Ans +Evid`) accuracies.

5.1.1 Multi-document QA setup.

For **RQ1**, we use two multi-document QA datasets - MULTIHOP-RAG and QASPER. Construction details, retrieval variants, and the LLM-judge protocol are in Appendices D and E.

5.1.2 EBG Setup

Synthetic Dataset. We sweep context lengths from 10K to 1M characters. For each length, we run N=100 trials: each trial draws a fresh distractor haystack and samples one needle uniformly from the 10 curated two-hop needles. Since all needles are two-hop by construction, we apply only the *EBG* prompt and report `Ans`, `Evid`, and `Ans+Evid` accuracy. A sanity check at L=10K confirms all models achieve ≥0.98 joint accuracy, establishing that degradation at longer lengths reflects context handling rather than needle ambiguity. This setup

addresses **RQ2** and **RQ3**; there are no contradictory needles, so RQ4 is not evaluated here.

NoLiMa-based Dataset. Under the NoLiMa setting, we use B_1^c as the haystack source across all context lengths to maintain a fully controlled evaluation setup.

There are different test arguments for the questions corresponding to each needle in the NoLiMa dataset. We use the `T01` test variant from NoLiMa throughout the controlled setup. `T01` is selected because, for each needle, it provides exactly one input argument for the one-hop question and one input argument for the two-hop question, enabling a consistent comparison between direct retrieval and commonsense composition while holding the underlying needle fixed. For example, for the needle `0405`, the one-hop template *"Which character cannot eat {1}?"* is instantiated with *fish-based meals*, whereas the two-hop template *"Which character cannot eat {2}?"* is instantiated with *Brandade*. Similarly, for `0402`, the one-hop and two-hop templates are instantiated with *milk* and *a cappuccino*, respectively. We test two characters, *Yuki* and *Stuart*, with insertion depths as mentioned in Section 4.2, yielding 16 needle-placement conditions per context length per model (4 needles $\times$ 4 depths).

We evaluate three settings: *single-needle* (one-hop and two-hop, addressing **RQ2**), *two-needle* (two-hop with explicit secondary needles, addressing **RQ3**), and *contradictory two-needle* (CTN, addressing **RQ4**). Results are reported separately for one-hop and two-hop settings and averaged across conditions.

5.1.3 NoLiMa-based RAG Setup

Under this setup, we conduct the *LC-EBG* vs. *RAG* comparison (**RQ1**) on the NoLiMa-based dataset. We evaluate two prompting settings: *answer-only* and *EBG* prompting with explicit evidence citation, and measure them using `Ans`* and joint `Ans+Evid` accuracy, respectively. We embed each haystack line after needle insertion and use the resulting representations to construct retriever-ordered chunks.

5.2 Results

We report results organized by the research questions introduced in Section 5.

RQ1: LC-EBG addresses two RAG limitations. On MULTIHOP-RAG, queries require 2–4 evidence facts spread across distinct articles,

Model	LC	flat@3	struct@3	ΔLC (best)
GPT-5	**0.605**	0.500	0.553	+0.05
GPT-5.5	**0.671**	0.500	0.539	+0.13
Opus-4.6	**0.697**	0.474	0.566	+0.13
Sonnet-4.5	**0.592**	0.513	0.474	+0.08
Gemini-3-Fl	**0.658**	0.487	0.539	+0.12

Table 1: **QASPER-MP answerable-only LLM-judge scores** (n=38**, EBG**). **LC** wins for all five models; the largest gaps are for the strongest readers (`Opus-4.6` and `GPT-5.5`). Higher (↑) values are better. Full breakdown in Appendix E.

so the retriever must surface every fact in its top-k budget and missing one collapses the answer. We first reproduce the answer-only RAG baselines from the original benchmark under our pipeline (`text-embedding-3-large` + 256-token windows; Appendix D), and measure the retrieval ceiling directly: at k=10, the retriever covers *any* gold fact for 82% of queries but covers *all* gold facts for only 19% (and just 8% on inference queries). **This all-vs-any gap is the structural limitation of similarity retrieval on multi-fact queries: standard RAG often retrieves some relevant evidence but fails to retrieve all required facts within the top-k budget, and missing a single fact collapses the answer.** This gap upper-bounds any RAG reader regardless of model strength. The reader numbers confirm it: at k=10 *RAG* answer accuracy is 60/48/44% (`GPT-5`/`Sonnet-4.5`/`Gemini-3-Fl`), while *LC-EBG* on the same queries reaches 86/82/78% (+26 to +34 pp); even at k=200, where the retrieved context already approaches the size of the *LC* haystack, *RAG* only ties *LC-EBG* for one of the three models. where the retrieved context already approaches the size of the *LC* haystack, *RAG* only ties *LC-EBG* for one of the three models.where the retrieved context already approaches the size of the *LC* haystack, *RAG* only ties *LC-EBG* for one of the three models. Consistent with this, our NOLIMA-based RAG analysis (Table 10) shows that **retrieval costs scale sharply with reasoning complexity and haystack length**. While one-hop full coverage remains relatively efficient, two-hop retrieval grows from 1.9K retrieved characters at 10K context to over 202K at 1M context, effectively pushing RAG toward brute-force long-context processing without guaranteeing complete evidence recovery.

On QASPER-MP, the query refers to a paper by an author or short title cue; answering requires

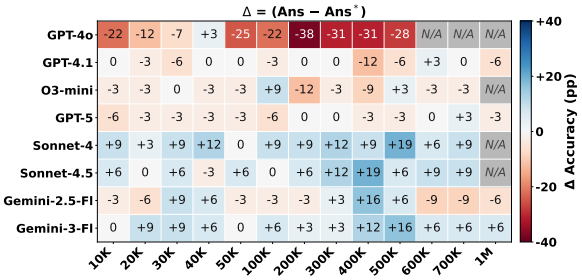

Figure 1: NoLiMa-based *LC* benchmark comparing answer accuracy under **EBG prompting (Ans)** and ***answer-only* prompting (Ans*)** across eight frontier models and context lengths ranging from 10K to 1M characters. Results are reported as the **accuracy difference** (Δ) between the two prompt settings. Cells marked *N/A* indicate inputs exceeding the model's maximum context window. Larger positive values (↑) indicate greater effectiveness of *EBG* over *answer-only* prompting.

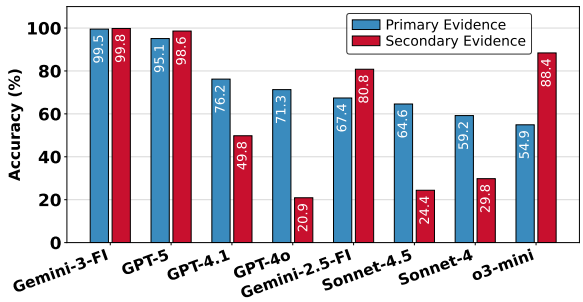

Figure 2: Two-needle citation breakdown: rate of correctly citing the **primary needle vs.** the **secondary needle** under NoLiMa-based *LC* benchmark. Higher (↑) values are better.

document-level identity, not just topical fragments. **This exposes a second limitation of standard RAG: chunk-level similarity retrieval struggles with document identity and provenance resolution when queries refer indirectly to a source document.** Table 1 reports answerable-only judge scores under the apples-to-apples comparison (*LC-EBG* and *RAG* share the same 11-paper pool). *LC-EBG* beats the best *RAG* variant for every one of five frontier models, with gaps of $+0.05$ to $+0.13$ judge points. Aggregate scores look tied only because *RAG* over-refuses on missed retrievals (24–40% refusal on answerable queries vs. 13–18% for *LC-EBG*), which accidentally matches QASPER's `Unanswerable` gold label on the 24% of pilot questions where Unanswerable is correct. With full-corpus retrieval (`flat_full@10`), all-needles coverage drops to 48% and *LC-EBG* beats the best *RAG* configuration by 5–13 judge points (Appendix E).

Across both datasets, **the bottleneck is retrieval (missed needles, induced refusals), not reading**: *LC-EBG* side-steps both by giving the model the full document neighborhood at once.

RQ2: Citation improves answer accuracy. On the synthetic benchmark (Fig. 4a, Appendix F.1), all three frontier models—GPT-5, Claude Sonnet-4.5, and Gemini-3-Flash—approach ceiling answer accuracy under both prompts (*answer-only* and *EBG*): GPT-5 and Gemini-3-Flash remain at $\geq 98\%$ across every length, and Sonnet-4.5 stays in the 93–100% band up to 700K characters. Consequently, the synthetic dataset does *not* exhibit a measurable citation-as-scaffold effect on the answer dimension: *Ans** and *Ans* are within sampling noise of each other for every cell. This is not a refutation of RQ2 but a consequence of the controlled construction: the two-hop facts are surrounded by lexically dissimilar distractors, so locating the answer is easy whenever the model can read the input at all.

The scaffolding effect becomes apparent in the NoLiMa-based *LC* benchmark (Fig. 1), where Sonnet and Gemini models exhibit substantial gains from *EBG* prompting, improving between $+9$ to $+19$ pp across context lengths ranging from 300K to 900K.

RQ3: Asymmetric Retrieval Behavior. As observed earlier, frontier LLMs attain near-ceiling answer accuracy on the synthetic benchmark under two of our primary prompt settings, limiting its ability to expose retrieval asymmetries. The underlying trend, however, becomes apparent in the NoLiMa-based *LC* benchmark.

As shown in Fig. 2, `Gemini-3-Fl` and `GPT-5` reliably recover both the primary and secondary needles across nearly all context lengths. `Gemini-2.5-Fl` and `o3-mini` also achieve strong results, but with a slight inversion in which citation of the secondary needle matches or surpasses that of the primary needle, suggesting that definitional evidence is easier to recover than character-specific constraints. By contrast, `GPT-4.1`, `GPT-4o`, and the `Sonnet` models exhibit a clear imbalance: they often identify the primary needle while missing the secondary one. Importantly, citation of the primary needle also falls short of perfect accuracy, indicating that retrieval errors affect both evidence types rather than only the secondary signal.

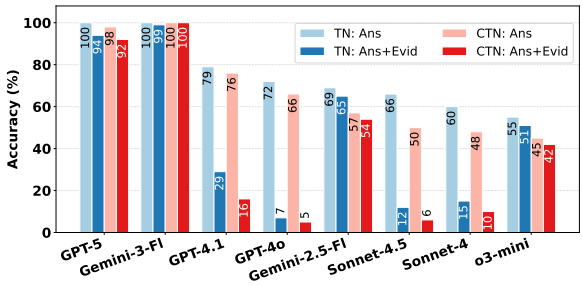

Figure 3: **TN vs. CTN**. TN uses a factual secondary needle; CTN uses a contradictory secondary needle, conflicting with world knowledge, requiring context-faithful reasoning and citation. Higher (↑) values are better.

Overall, these results point to asymmetric retrieval behavior: stronger models reliably recover both evidence lines, whereas others either under-retrieve the secondary needle or exhibit uneven retrieval across the two signals, resulting in incomplete attribution.

RQ4: CTN amplifies failure . Performance generally declines under contradiction, with substantial variation across models (Fig. 3). `GPT-5` and `Gemini-3-Fl` remain relatively context-faithful, preserving high answer accuracy, though their `Ans+Evid` still drops, indicating weaker grounding. In contrast, `GPT-4.1`, `GPT-4o`, and the `Sonnet` variants show sharper declines, especially in `Ans+Evid`, suggesting greater reliance on parametric knowledge over conflicting context. Overall, the widening gap between `Ans` and `Ans+Evid` under contradiction highlights a common failure mode: models often answer correctly but fail to justify responses using the provided evidence. Refer to Appendix G.3

6 Conclusion and Future Work

We introduced **evidence-based generation (EBG)**, a framework that requires models to return both an answer and line-level citations, together with a joint `Ans+Evid` metric that scores correctness and grounding simultaneously. EBG closes a long-standing gap in long-context evaluation—answer-only metrics conflate retrieval failures with reading failures—and lets us treat long-context (LC) models as a controlled probe of where retrieval-augmented generation (RAG) breaks. To exercise the RAG pipeline on the dimensions it is meant to handle, we reformulated two existing corpora into evidence-grounded multi-document QA benchmarks (MULTIHOP-RAG and QASPER-MP) and paired them with a controlled synthetic two-hop haystack and an evidence-augmented version of NoLiMa spanning 10K–1M characters.

Across ten frontier LLMs, EBG isolates two concrete RAG limitations that LC models avoid by construction: *(i)* similarity retrieval surfaces *any* relevant chunk far more reliably than *all* required chunks for multi-fact queries, capping any RAG reader at the retrieval ceiling at realistic top-k budgets; and *(ii)* chunking destroys the document meta-information that identity-bearing queries depend on, so RAG over-refuses on missed retrievals while LC succeeds on the same haystack. Beyond the RAG comparison, EBG reveals grounding failures invisible to answer-only evaluation: citation generation acts as a reasoning scaffold on naturalistic data, models cite incomplete evidence in two-needle settings, and the gap between answer and joint accuracy widens sharply when in-context evidence contradicts parametric knowledge.

The picture that emerges is not "LC replaces RAG" but rather that both pipelines have measurable failure modes, and EBG is the right instrument for locating them. We therefore argue that future long-context and retrieval-augmented evaluations should move beyond answer-only metrics and explicitly measure evidence attribution and grounding fidelity.

Several directions remain for future work. First, our pilots use small but stratified slices of MULTIHOP-RAG and QASPER-MP; running EBG over the full sets is straightforward with our infrastructure and would tighten confidence intervals. Second, we fix the retriever (`text-embedding-3-large` + sliding-window chunks) for clean apples-to-apples comparisons; richer pipelines—reranking, hybrid sparse–dense retrieval, query expansion, and document-aware chunking—are likely to mitigate (but not eliminate) the all-vs-any and meta-information failure modes, and EBG is the natural tool for evaluating them. Third, richer forms of evidence attribution (span-level alignment, hierarchical document references, chain-of-thought grounding) remain unexplored. Finally, our findings suggest that evidence generation itself improves reasoning quality, raising the broader question of whether grounding objectives should become part of model training rather than evaluation alone.

Limitations

Our evaluation focuses on synthetic and semi-synthetic long-context benchmarks designed to isolate grounding and evidence attribution behavior under controlled conditions. While these settings enable precise analysis, they do not capture the full complexity of real-world domains such as legal, scientific, or medical reasoning. In addition, our evidence metric operates at the line level and therefore measures coarse-grained provenance rather than fine-grained span-level attribution.

Our RAG analysis estimates end-to-end performance through a decomposition of retrieval and long-context accuracy rather than exhaustive evaluation of full production RAG pipelines across all configurations. Finally, because we evaluate proprietary frontier models through API access, results may vary as deployed systems evolve over time.

Ethics and Broader Impact

This work studies evidence grounding and provenance in long-context language models. Our benchmarks are constructed from synthetic or publicly available text and do not contain sensitive personal information. We do not train new foundation models; all experiments are conducted through inference-only evaluation of existing systems.

We believe evidence-based evaluation can improve transparency and verifiability in LLM-based systems by encouraging models to explicitly identify the context used to support their answers. Such capabilities may be particularly important in high-stakes applications such as legal analysis, scientific search, and medical document understanding, where unsupported or hallucinated answers can have significant consequences.

At the same time, citation generation should not be interpreted as a guarantee of faithful reasoning. A model may still cite irrelevant or incomplete evidence while producing a plausible answer. We therefore view evidence-based generation as a complementary evaluation framework rather than a complete solution to factuality or interpretability in language models.

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*.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610. Association for Computational Linguistics.

Yufeng Du, Minyang Tian, Srikanth Ronanki, Subendhu Rongali, Sravan Babu Bodapati, Aram Galstyan, Azton Wells, Roy Schwartz, Eliu A Huerta, and Hao Peng. 2025. Context Length Alone Hurts LLM Performance Despite Perfect Retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 23281–23298, Suzhou, China. Association for Computational Linguistics.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. 2024. RULER: What's the Real Context Size of Your Long-Context Language Models? In *First Conference on Language Modeling*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Greg Kamradt. 2023. Needle in a haystack-pressure testing llms.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Alex Laitenberger, Christopher D Manning, and Nelson F. Liu. 2025. Stronger Baselines for Retrieval-Augmented Generation with Long-Context Language Models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 32559–32569, Suzhou, China. Association for Computational Linguistics.

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*.

Zhuowan Li, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Retrieval Augmented Generation or Long-Context LLMs? A Comprehensive Study and Hybrid Approach. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages

881–893, Miami, Florida, US. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A. Rossi, Seunghyun Yoon, and Hinrich Schuetze. 2025. NoLiMa: Long-Context Evaluation Beyond Literal Matching. In *Proceedings of the 42nd International Conference on Machine Learning*, PMLR '25, pages 44554–44570.

Ofir Press, Noah Smith, and Mike Lewis. 2022. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. In *International Conference on Learning Representations*.

Thomas Schuster, Marian Lambert, Nico Döring, and Julius Trögele. 2025. Needle-in-the-Haystack Testing LLMs with a Complex Reasoning Task. In *Engineering Applications of Neural Networks*, pages 254–266, Cham. Springer Nature Switzerland.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomput.*, 568(C).

Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *arXiv:2401.15391*.

Qingyang Wu, Zhenzhong Lan, Kun Qian, Jing Gu, Alborz Geramifard, and Zhou Yu. 2022. Memformer: A Memory-Augmented Transformer for Sequence Modeling. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pages 308–318, Online only. Association for Computational Linguistics.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. 2025. HELMET: How to evaluate long-context models effectively and thoroughly. In *The Thirteenth International Conference on Learning Representations*.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024. ∞Bench: Extending Long Context Evaluation Beyond 100K Tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15262–15277, Bangkok, Thailand. Association for Computational Linguistics.

Yonghao Zhuang, Lanxiang Hu, Longfei Yun, Souvik Kundu, Zhengzhong Liu, Eric P. Xing, and Hao Zhang. 2025. Scaling Long Context Training Data by Long-Distance Referrals. In *The Thirteenth International Conference on Learning Representations*.

A Prompt Templates

The haystack is line-numbered (`[1]`, `[2]`, ..., `[N]`) and substituted into the `{haystack}` placeholder of each task template. We use two prompt families: an *answer-only* variant (no system prompt) and an *evidence-based (EBG)* variant that asks the model to also cite supporting line numbers in JSON.

Answer-only Prompt (User Message)

```
You will answer a question based on the following book snippet:
{haystack}
Use the information provided in the book snippet to answer the question. Your answer should be short and based on either explicitly stated facts or strong, logical inferences.
Question: {question}
Return only the final answer with no additional explanation or reasoning.
```

EBG System Prompt

```
Your job is to answer the question entirely from the context and provide a reference. Your answer should cite all lines the answer is based on.
```

EBG User Prompt

```
<context>{haystack}</context>
Answer the question based on information only from the context. If the question is not answerable from the context, answer NA. Your response should comprise only the answer and all lines the answer is based on in json format. For example:
{"lines": [25, 412], "answer": "John"}
Question: {question}
```

B Additional Details of Synthetic Dataset

We present a representative example from our synthetic dataset in this section.

B.1 Example

Consider facts *"Himalaya Inc. launched FitBand"* and *"Himalaya Inc. is based in Mumbai"*, with question *"Which smart wearable was launched by a company based in Mumbai?"* At L=10K the two facts land at lines 56 and 67 among random distractors:

Haystack excerpt and question (synthetic dataset)

```
...
54: Tech research group DataSynergy optimized machine learning algorithms.
55: A renowned sculptor in Rome crafted a public monument.
```

```
56: Himalaya Inc. launched a new smart
wearable called FitBand.
57: A notable charity, WarmHearts, distributed
winter supplies to families in need.
58: GreenWave unveiled a new system for
harnessing tidal energy.
. . .
65: Fitness startup VitalMotion developed
smart workout equipment.
66: GlobalVoyager published a feature on
little-known cultural festivals.
67: Himalaya Inc. is based in Mumbai.
68: Director James Park released a documentary
on endangered rainforests.
69: Education reformist Tara Singh promoted
progressive teaching methods.
. . .
Question: Which smart wearable was launched by
a company based in Mumbai?
```

Neither line alone is sufficient: line 56 identifies the product but not the company location, while line 67 identifies the location but not the product. The model must therefore compose evidence across both lines. Under the EBG prompt template, the expected output is a JSON object containing both the answer and the cited line numbers:

Expected EBG response

```
{"lines": [56, 67], "answer": "FitBand"}
```

Such a response is scored as correct on `Ans`, `Evid`, and `Ans + Evid`. The remaining nine needles follow the same structure with varied entities, relations, and answer forms.

C NoLiMa Datatset Adaptation Details

C.1 Needle settings

Table 2 presents the different needle templates used across different settings.

C.2 Single Needle (SN) setting

In this setting, `0402` and `0402Inv` encode the same *dietary* fact about a character (`{CHAR}`) using different surface forms with altered word order. We refer to the first variant as the *default* form and the second as the *inverse* form. Similarly, needles `0405` and `0405Inv` state that a character (`{CHAR}`) is *vegan*. These needles support both **one-hop** questions answerable directly from the needle (e.g., "*Which character is lactose intolerant?*") and **two-hop** questions requiring an additional world-knowledge inference (e.g., "*Which character cannot drink a cappuccino?*", which requires knowing that *cappuccino* contains *milk*).

C.3 Two-needle (TN) setting

As shown in Table 2, we introduce two explicit secondary needles under the two-needle (*TN*) setting: `TN-0402-S` and `TN-0405-S`. These needles explicitly provide the world-knowledge fact required for two-hop reasoning (e.g., *"CHAR said, 'Cappuccino is a classic Italian coffee made with espresso and steamed milk.',"*). Each secondary needle pairs with the corresponding primary needles. For example, `TN-0402-S` pairs with `0402` and `0402Inv`, while `TN-0405-S` pairs with `0405` and `0405Inv`. Answering a two-hop question, therefore, requires retrieving and jointly reasoning over both the primary and secondary needles.

C.4 Contradictory Two-needle (CTN) setting

Under this setting, we introduce secondary needles that explicitly conflict with world knowledge. For example, `CTN-0402-S` claims that *black coffee* is made with *steamed milk*, while `CTN-0405-S` claims that *falafel* contains *minced lamb*. Similar to the *TN* setting, each contradictory secondary needle pairs with its corresponding primary needles (e.g., `CTN-0402-S` with `0402` and `0402Inv`). Answering the resulting two-hop questions, therefore, tests whether models remain faithful to the provided context or instead override it using parametric knowledge.

D Long-context EBG vs. RAG on MultiHop-RAG

This appendix documents an additional set of experiments that compare long-context evidence-based generation (*LC-EBG*) against a retrieval-augmented generation (*RAG*) pipeline on the MULTIHOP-RAG benchmark (Tang and Yang, 2024). The motivation is direct: our synthetic two-hop benchmark and the NoLiMa-adapted benchmark are constructed haystacks, which leaves open the question of whether the trends we observe carry over to realistic, web-derived multi-document QA where *RAG* is the natural baseline. The filtered MULTIHOP-RAG subset (§4.3) of 2,255 multi-hop queries is a clean testbed for both *EBG* (requiring line-level evidence) and *RAG* (requiring chunked, embeddable corpus).

We run the following experiments on a 50-query pilot set (randomly sampled with seed 0 and naturally stratified across the three non-null question types in MULTIHOP-RAG: $n_{\text{inference}}$=13, $n_{\text{comparison}}$=21, and n_{temporal}=16). We use the

Variant	Needle Template
Original NoLiMa primary needles (under SN)	
0402	*"A message came in from {CHAR} saying, 'I'm lactose intolerant,' and nothing more."*
0402Inv	*"A message came in saying, 'I'm lactose intolerant,' from {CHAR}."*
0405	*"Then {CHAR} mentioned that he has been vegan for years."*
0405Inv	*"There was a vegan guest, named {CHAR}."*
Gender neutral rephrased primary needle (under SN)	
0405-R	*"Then {CHAR} mentioned being vegan for years."*
Explicit secondary needles (under TN)	
TN-0402-S	*"{CHAR} said, 'Cappuccino is a classic Italian coffee made with espresso and steamed milk.' "*
TN-0405-S	*"Then {CHAR} mentioned that Brandade is a fish-based dish made with salted cod."*
Explicit secondary needle (under CTN)	
CTN-0402-S	*"Black coffee is a rich and creamy beverage made by brewing coffee beans in steamed whole milk."*
CTN-0405-S	*"Falafel is a popular Middle Eastern dish made from ground chickpeas mixed with minced lamb."*

Table 2: **Needle templates** used in the evaluation. Original NoLiMa needles under the **single-needle** (*SN*) setting evaluate both one-hop and two-hop reasoning. We add 0405-R, a gender-neutral rephrased version of the 0405 primary needle template under SN, to examine the influence of gendered wording in the needle template. Explicit secondary needles under the **two-needle** (*TN*) setting directly provide the required world-knowledge fact and evaluate two-hop reasoning under fully contextualized evidence. The **contradictory two-needle** (*CTN*) setting instead introduces secondary needles that conflict with world knowledge, enabling evaluation of whether models remain strictly faithful to the provided context during two-hop reasoning.

same 50 queries throughout the appendix, making all reported results directly comparable.

Per-query mini-haystacks (long-context condition). For each query we assemble a mini-haystack consisting of (i) all gold-evidence documents and (ii) 20 additional distractor documents sampled (without replacement) from the remainder of the corpus. Documents are concatenated in random order, split into sentence-level lines, and rendered in the `{lineno}: {text}` format used by our synthetic benchmark. Gold evidence facts are re-mapped to line numbers inside the assembled mini-haystack so the EBG labels remain consistent with the prompt the model sees. Mini-haystacks average ~ 241K characters (~ 58K tokens), comfortably within every model's context window. This is the *LC (full)* condition.

RAG haystacks. For each query we also build a family of RAG-style contexts at $K \in \{4, 10, 25, 50, 100, 200\}$. We chunk the *full* 609-document corpus into 256-token sliding windows (6,296 chunks), embed every chunk with `text-embedding-3-large`, embed each query with the same model, and rank chunks by cosine similarity. The top-K chunks (in rank order) form the RAG context; we render them with the same sentence-per-line numbering as above, prefixing each chunk with a marker `[CHUNK <rank> | <title>]`. Gold evidence facts are again re-mapped to lines inside the RAG context (a fact whose chunks were not retrieved simply has no corresponding line).

D.1 Setup

Prompt modes. Each (query, context) pair is evaluated under two prompts that differ in a single sentence:

- *Answer-only (*`Ans`**).* The model is asked for a JSON object with only an `answer` field. We report `Ans`* accuracy.
- *EBG.* The model is asked for a JSON object with `answer` and `lines` fields. Scoring uses `Ans` (string match against the gold answer), `Evid (strict)` (every gold line appears in the prediction), and `Joint` $\equiv$ `Ans+Evid (strict)` (answer-and-evidence).

Both prompts share the same system message, instructions, and JSON example template, modulo the citation field. The system message and prompt templates are reproduced verbatim in our code release.

Models. We evaluate three frontier LLMs whose context windows comfortably fit under the *LC* condition: `GPT-5`, `Sonnet-4.5`, and `Gemini-3-Fl`. All calls use temperature 0 (or provider equivalent), default sampling parameters, and `parallel=5`.

Scoring. `Ans` matches the model's answer string against the gold answer using the same normalization used in the synthetic and NoLiMa experiments.

Model	*Ans** (ans-only)	*Ans* (EBG)	Δ
GPT-5	82.0%	86.0%	+4 pp
Claude Sonnet-4.5	68.0%	82.0%	**+14 pp**
Gemini-3-Flash	80.0%	78.0%	−2 pp

Table 3: **Ans* (answer-only) vs. Ans (EBG)** on the 50 pilot queries, full mini-haystack (∼ 241K characters, 20 distractors). Requiring citations raises answer accuracy for two of the three models, with a striking +14 pp improvement for `Sonnet-4.5`. This replicates the citation-as-scaffold effect that was invisible in our (saturated) synthetic benchmark. Higher (↑) values are better for `Ans`* and `Ans`.

Config	H@1	H@4	H@10	H@50	H@100	MAP@10
any gold						
small + sent.	0.29	0.52	0.67	0.90	0.95	0.22
small + tok256	0.27	0.59	0.80	0.97	0.99	0.24
large + sent.	0.32	0.55	0.71	0.94	0.97	0.24
large + tok256	**0.33**	**0.66**	**0.82**	**0.98**	**1.00**	**0.27**
all gold (faithful-RAG ceiling)						
small + sent.	0.00	0.06	0.14	0.34	0.48	—
small + tok256	0.01	0.07	0.18	0.57	0.79	—
large + sent.	0.00	0.07	0.16	0.38	0.51	—
large + tok256	**0.00**	**0.08**	**0.19**	**0.57**	**0.71**	**—**

Table 4: **Corpus-wide retrieval** metrics on MULTIHOP-RAG (N=2255). The *all gold* block is an upper bound on what any faithful RAG pipeline can achieve at top-k: if even one of the 2–4 gold facts is missing from the retrieved set, downstream answers cannot be faithful. The token-windowed `large` configuration dominates and is used for all RAG experiments below. Higher (↑) values are better.

`Evid (strict)` requires the predicted line-id set to contain *every* gold line for that query; partial matches are scored separately as `Evid (recall)` but only `Evid (strict)` is used for `Joint`. API failures (timeouts, safety refusals, transient connection errors) are excluded from the denominator; in practice this affected at most one cell (Sonnet-4.5, *RAG*@200 *EBG*, n=49/50).

D.2 Experiment A: Long-context EBG on MULTIHOP-RAG

We first run the LC condition in isolation to establish the base rates and to test RQ2 (*Does requiring explicitly citing evidence improve answer accuracy?*) on a realistic, non-synthetic benchmark. Each model sees the full mini-haystack (∼ 241K characters) and is prompted in *answer-only* mode or *EBG* mode. Table 3 summarizes the results.

Joint accuracy is much harder than *Ans*: under EBG, joint (*Ans*+*Evid*) is only 28% (GPT-5), 8% (Sonnet-4.5), and 28% (Gemini-3-Flash). The gap *Ans* − *Joint* ranges from 50 pp (Gemini) to 74 pp (Sonnet) and is concentrated on inference queries, mirroring the *asymmetric retrieval* finding from our synthetic two-needle setting: models routinely produce a correct answer while citing only one of the two required hops.

D.3 Experiment B: Retrieval analysis

Before running any LLM we measure how hard the retrieval problem itself is. For every query and every gold fact we record the rank of the first chunk that contains that fact; *Hits@k (all gold)* counts the fraction of queries for which *every* gold fact appears in the top-k. We compare two embedding models (`text-embedding-3-small/large`) and two chunkers (256-token sliding windows vs. sentence-level chunks); full results are in Table 4 (corpus-wide, N=2255), with our chosen configuration in bold.

The key qualitative finding is that retrieving *at least one* gold fact is easy (Hits@10 (any) = 0.82 for our chosen config), but retrieving *all* gold facts is hard (Hits@10 (all) = 0.19). Inference (multi-hop) queries are the most punishing: Hits@10 (all) is only 0.08 for that slice. For RAG to faithfully ground a multi-hop answer at k=10, more than 90% of inference queries are doomed before the LLM is even called.

D.4 Experiment B′: Needle coverage on the pilot

For the 50 pilot queries we resimulate this analysis at every K we will later evaluate. Table 5 reports the fraction of pilot queries for which every gold fact ("needle") is retrieved in the top-K, and the fraction of individual gold facts that appear anywhere in the top-K. These two numbers are the upper bound on, respectively, faithful joint accuracy and answer recall for RAG on the pilot set.

D.5 Experiment C: Head-to-head LC EBG vs. RAG

The central experiment of this appendix sweeps RAG at six values of K and compares against LC under both prompt modes. The same 50 queries, the same prompt templates, and the same scoring code are used throughout, so every cell in Table 6 is directly comparable.

The table tells three distinct stories, one per metric.

K	% queries w/ all needles	% gold facts retrieved
4	14.0%	36.3%
10	28.0%	49.6%
25	44.0%	68.1%
50	58.0%	76.3%
100	64.0%	84.4%
200	90.0%	96.3%

Table 5: **Needle coverage** on the 50 pilot queries (retriever: `text-embedding-3-large` + 256-token chunks). At realistic RAG budgets ($K \leq 25$) fewer than half of queries have all gold facts retrieved; only at K=200 does coverage approach saturation. Higher (↑) values are better.

Answer accuracy: LC strictly dominates RAG. Across both prompts and all three models, LC's *Ans* beats every RAG@K for $K \leq 100$. RAG@200 (which has 96% gold-fact recall and 90% all-needles coverage) finally ties Sonnet-4.5 (68% vs. 68%) but still trails GPT-5 by 4 pp (78 vs. 82%) and Gemini-3-Flash by 10 pp (70 vs. 80%). At a realistic RAG operating point (K=10), the LC advantage is large: +22 to +44 pp across models under the answer-only prompt and +26 to +34 pp under EBG. **If the goal is to answer multi-hop questions correctly, long context wins.**

Strict evidence: small-K RAG and LC are comparable. *Evid (strict)* is roughly flat across K for each model (10–40% absolute), with no clean monotone trend. This is because evidence correctness is a function of *whether all needles were retrieved* (coverage, Table 5) *and whether the model cited them all* (a behaviour). Small K helps the latter (the citation surface is small), large K helps the former (coverage rises), and the two effects roughly cancel.

Joint (faithful) accuracy: mixed. `Joint` = `Ans` ∧ `Evid` (also written as `Ans+Evid`) is the most diagnostic single number. Two trends are visible. First, joint *decreases* with K for every model: at large K the citation surface grows, the model loses track of which retrieved chunk it used, and joint collapses despite the higher answer recall. Second, the best RAG joint is always at small K (K=4 for all three models), where the citation problem is trivial and the only queries that score are the ones whose needles happened to be in the top 4 (14% of queries). Aggregate joint therefore over-rewards a regime that only works on the easy fraction of queries. LC's joint sits at 28%, 8%, 28% — competitive with RAG@4 for GPT-5 and Gemini, behind for Sonnet — and is computed over the full pilot set rather than the small subset where retrieval succeeded.

D.6 Discussion

We highlight four takeaways for the manuscript.

Takeaway 1: LC EBG dominates RAG for answer accuracy, at every realistic retrieval budget. The cleanest summary statistic. Under the answer-only prompt, LC beats RAG@10 by +22 to +44 pp and beats RAG@200 by 0 to 10 pp. Under EBG, the same ordering holds with similar magnitudes. This is the headline evidence that on a realistic multi-hop benchmark, long context is preferable to retrieval when the goal is correctness.

Takeaway 2: RAG's retrieval bottleneck is multi-hop coverage, not single-hop recall. Hits@10 (any gold) is 82% but Hits@10 (all gold) is only 19% (Table 4); on inference queries the all-gold number falls to 8%. The fundamental reason small-K RAG cannot match LC on MULTIHOP-RAG is that it almost never has the union of gold needles available.

Takeaway 3: Citations as a reasoning scaffold replicate outside the synthetic setting. *Ans* accuracy under EBG is higher than under the answer-only prompt for two of three models, with a striking +14 pp improvement for Sonnet-4.5 (Table 3). This is the RQ1 effect that was masked on our synthetic benchmark by ceiling performance.

Takeaway 4: The grounding gap is a real-data phenomenon. The *Ans* − *Joint* gap in the LC condition is 50–74 pp depending on model, and is largest on inference queries. Models routinely produce a correct multi-hop answer while citing only one of the two required hops — the same asymmetric-retrieval failure mode we documented on the synthetic benchmark, here observed on naturalistic web data.

D.7 Limitations and next steps

The pilot uses only 50 queries; sampling noise on small slices (*e.g.*, the 13 inference queries) is non-trivial. A full sweep on all 2,255 usable queries is straightforward with our infrastructure and would tighten the confidence intervals on Tables 3 and 6. We also fix the retriever (`text-embedding-3-large` + 256-token windows) for simplicity; reranking, hybrid sparse–dense retrievers, and longer chunks are likely to

Model	Metric	LC (full)	RAG@4	RAG@10	RAG@25	RAG@50	RAG@100	RAG@200
Answer-only prompt (*Ans** / Joint ≡ Ans)								
GPT-5	Ans	**82.0**	58.0	60.0	70.0	72.0	74.0	78.0
Sonnet-4.5	Ans	**68.0**	24.0	36.0	38.0	56.0	58.0	68.0
Gemini-3-Flash	Ans	**80.0**	38.0	44.0	50.0	58.0	58.0	70.0
EBG prompt: Ans / Evid (strict) / Joint								
GPT-5	Ans	**86.0**	58.0	60.0	64.0	72.0	76.0	76.0
	Evid	36.0	38.0	36.0	30.0	30.0	30.0	30.0
	Joint	28.0	**34.0**	26.0	22.0	20.0	22.0	22.0
Sonnet-4.5	Ans	**82.0**	42.0	48.0	62.0	70.0	68.0	74.0
	Evid	10.0	30.0	30.0	22.0	16.0	18.0	14.0
	Joint	8.0	**26.0**	22.0	18.0	16.0	12.0	12.0
Gemini-3-Flash	Ans	**78.0**	38.0	44.0	46.0	64.0	62.0	66.0
	Evid	40.0	28.0	18.0	24.0	24.0	24.0	28.0
	Joint	**28.0**	26.0	14.0	18.0	20.0	18.0	20.0

Table 6: **LC-EBG vs. RAG-EBG** on the 50 pilot queries. All numbers are percentages over n=50 (Sonnet-4.5 RAG@200 EBG: n=49). Boldface marks the per-row maximum within the corresponding mode block. Higher (↑) values are better.

improve RAG numbers but will not change the qualitative ordering at any K that is short enough to matter (*i.e.*, $K \leq 100$): the all-gold ceiling at K=10 is 19% *regardless of the LLM downstream*.

D.8 Reproducibility

All code, queries, and intermediate artifacts live under `baselines/multihop_rag/` in the experiment repository:

- `build_haystack.py`: builds per-query mini-haystacks (LC condition).
- `embed_and_rank.py`: chunks the corpus, embeds, ranks gold facts; produces `summary.json`, `fact_ranks.jsonl`, and `chunks.jsonl` under `retrieval_results/<config>/`.
- `build_rag_haystack.py`: assembles per-K RAG contexts from the retrieval artifacts.
- `run_baseline.py`: runs the LLM evaluation on either LC or RAG haystacks, supports `-answer-only`.
- `aggregate_pilot.py`, `aggregate_rag_sweep.py`: produce the tables above.

The 50-query pilot is reproducible with `-seed 0` on `processed/queries_d20.jsonl` (built from the official MULTIHOP-RAG HuggingFace release of the corpus and queries).

E Long-context EBG vs. RAG on QASPER-MP

This appendix documents a second comparison of long-context evidence-based generation (LC EBG) against retrieval-augmented generation (RAG), this time on a multi-paper adaptation of QASPER (Dasigi et al., 2021) that we call QASPER-MP. Whereas MULTIHOP-RAG (Appendix D) tests multi-document reasoning on news, QASPER-MP tests it on scientific papers, where the gold question–answer–evidence triples were written assuming a single known paper. To make the task non-trivial under retrieval, we (i) place each gold paper alongside ten distractor papers, (ii) rewrite the question so it identifies the target paper by a realistic short reference (an author name or a 2–3 word topic cue), and (iii) evaluate in both LC and RAG conditions over the same 50-query stratified pilot.

The headline finding is that the aggregate score (Answer-F1 or LLM-judge, averaged over all questions) makes LC and RAG look roughly tied, but this tie hides an answerability asymmetry: *LC EBG beats RAG by 5–22 judge points on answerable questions for every frontier model tested*, while RAG "wins" on unanswerable questions because it over-refuses. The over-refusal mechanically matches the QASPER `Unanswerable` gold label on the 24% of pilot questions where that is the correct answer, pulling the aggregate up.

E.1 Setup

Dataset. We use the QASPER dev split ($N = 1{,}005$ questions over 281 papers). Each question is annotated with an *answer type* (`extractive`, `abstractive`, `boolean`, or `none`—meaning unanswerable from the paper) and an *evidence set*: the list of paragraphs from the paper that support the answer. Up to three annotators contribute per question; we follow the official QASPER evaluator and take the max F1 across annotators.

50-query pilot. We sample 50 questions, stratified by answer type, so that the composition matches the dev-split distribution: 13 boolean, 13 abstractive, 12 extractive, 12 unanswerable. This split (38 answerable, 12 unanswerable) is the denominator for every condition below.

Multi-paper haystacks (LC condition). For each pilot query we build an 11-paper haystack consisting of the gold paper plus 10 distractor papers drawn uniformly from the dev split. Each paper is rendered with explicit document boundaries:

```
<paper>
<title>Paper title</title>
<abstract>Paper abstract</abstract>
<section>Section name</section>
Paragraph 1 of that section
Paragraph 2 of that section
...
</paper>
```

Lines are concatenated and numbered `i: <line>` so the model can cite specific lines as evidence (consistent with our EBG protocol). The median LC haystack size is ∼32K characters; the largest is ∼56K. The gold-evidence paragraph(s) for each query are re-mapped to line indices in the assembled haystack.

RAG condition. We embed every QASPER dev paragraph and every (disambiguated) query with OpenAI `text-embedding-3-large`, and retrieve the top-K paragraphs by cosine similarity. We evaluate four retrieval variants, factored as $\{\text{flat, struct}\} \times \{\text{pool11, full}\}$:

flat_pool11. Top-K paragraphs retrieved from the same 11-paper pool as the LC haystack (apples-to-apples corpus). The reader sees a flat list of retrieved paragraphs with a one-line `[CHUNK r title]|` marker before each.

struct_pool11. Same retrieval, but paragraphs are regrouped by their source paper and wrapped in the same XML layout used for LC. Within a paper, paragraphs are sorted by their original index, and discontinuities (gaps where intermediate paragraphs were not retrieved) are marked with a single `...` line.

flat_full / struct_full. Same as above but retrieving from the full dev corpus ($N \approx 6{,}000$ paragraphs) rather than the 11-paper pool. This is the *realistic* open-corpus RAG setting: the retriever must locate the target paper among hundreds of similar NLP papers using only the short cue in the rewritten question.

Models. We evaluate five frontier reasoning models: **GPT-5**, **GPT-5.5**, **Claude Opus 4.6**, **Claude Sonnet 4.5**, and **Gemini 3 Flash** (preview). All runs use the same query set, the same retrieval indices, and the same EBG prompt template. Citation rules in the prompt forbid citing structural lines (`<title>`, `<abstract>`, `<paper>`, `[CHUNK`, or `...`); these lines are present only as context.

Question disambiguation. QASPER questions were written to be unambiguous given the paper; many become ambiguous in a multi-paper haystack (e.g., "What evaluation metrics were used?" becomes pathological with 11 candidate papers). We disambiguate each pilot question by inserting a short *paper cue*, chosen deterministically 50/50 from two cue types:

- **Author cue** ($n = 24$): *"In Smith et al., ... "*, with surname(s) drawn from the arXiv metadata of the gold paper. Conventions: 1 author → *"Smith"*, 2 authors → *"Smith and Doe"*, ≥ 3 authors → *"Smith et al."*.
- **Title cue** ($n = 26$): *"In the ⟨cue⟩ paper, ... "*, where the cue is a verbatim 2–3 word substring of the paper title (e.g., *"Vietnamese Word Segmentation"*, *"Winograd Schemas"*, *"TF-IDF Features"*).

Cues are intentionally short and realistic, matching what a researcher would type into a search box. This makes the retrieval problem genuinely harder than indexing on full titles or descriptive paraphrases.

E.2 Aggregate Performance: LC ≈ RAG (and why this is misleading)

Table 7 reports per-condition Answer-F1, Evidence-F1, and LLM-judge (an automated 0/0.5/1 grader

using GPT-5; see Appendix E.5). Aggregating over all 50 pilot questions:

- LC and `flat_pool11@3` are within ± 0.06 judge points for every model.
- `struct_pool11@3` beats LC for GPT-5 and matches it for Gemini under the LLM judge.

On its face, this is the standard "LC is comparable to RAG" result.

The aggregate hides an answerability asymmetry. Splitting the 50 queries into the 38 answerable and 12 unanswerable subsets tells a sharper story (Table 8).

Three observations:

1. **LC EBG wins answerable questions cleanly.** For all five models, LC has the highest Judge score on the 38-question answerable subset, with absolute gains over the best RAG variant of $+0.05$ (GPT-5), $+0.13$ (GPT-5.5), $+0.13$ (Opus-4.6), $+0.08$ (Sonnet-4.5), and $+0.12$ (Gemini-3-Flash). These gaps are robust to the citation-rule and discontinuity-marker changes described in Appendix E.4.

2. **RAG over-refuses, and the refusals pay off on the unanswerable subset.** `flat_pool11@3` refuses 29–40% of answerable questions (vs. 16–18% for LC) because when retrieval misses the gold paragraph the reader sees only distractor content and sensibly says "Unanswerable." Those same refusal-prone retrieval failures happen to be the right answer on unanswerable queries (where "Unanswerable" *is* gold), so RAG wins that subset by 0.17–0.25 judge points.

3. **Realistic open-corpus retrieval makes LC win by a margin the aggregate cannot hide.** The `flat_full@10` row in Table 7 shows LC outperforming full-corpus RAG by 5–13 judge points on the aggregate already; the answerable-only gap is larger again (not shown). When the retriever must distinguish the gold paper from $\sim$280 distractors using a 2–3 word cue, paper-level retrieval coverage drops to 48% (all-needles@10), and downstream answer quality follows.

The aggregate score thus rewards RAG's high refusal rate. Once we condition on answerability, the LC $>$ RAG gap on *actually answerable* questions is consistent across models and retrieval variants.

E.3 Disambiguation Details and Cue Length Sensitivity

We initially experimented with longer "descriptive" cues generated by GPT-5 from each paper's title and abstract (e.g., *"In the cross-lingual zero-shot NMT paper"*). These cues are unrealistically informative: they often include domain phrases that do not appear in the title at all, which makes retrieval implausibly easy. Embedding-based retrieval coverage with descriptive cues was 60.0% (all-needles@10, full corpus). Switching to the short, realistic cues described in Appendix E.1 dropped coverage to 48.0%—a 12-point hit—reflecting the fact that *"In Smith et al."* or *"In the BERTSUM paper"* is a much weaker retrieval signal than a paraphrased domain summary.

We use the short, realistic cues throughout this appendix because they correspond to how researchers actually query systems in practice; the $+12$pp coverage gain on descriptive cues would inflate RAG numbers relative to what users would experience.

E.4 Structured RAG, Citation Eligibility, and Discontinuity Markers

A first iteration of the structured RAG layout produced a counterintuitive result: `struct_pool11@3` *underperformed* `flat_pool11@3` on Evidence-F1 across all models (gaps of 0.07–0.12), despite carrying *more* information than the flat baseline (abstract per paper). On inspection, $\sim$44% of structured-RAG citations pointed at `<title>` or `<abstract>` lines: when an abstract paraphrases the answer concisely, the model unsurprisingly cites it—but QASPER gold evidence excludes abstracts by construction, so abstract citations *cannot* match gold.

We therefore made two changes to the structured layout and prompt that we use throughout the main results:

1. **Citation eligibility.** The EBG prompt forbids citing structural lines (`<title>`, `<abstract>`, `<paper>`, `</paper>`, `[CHUNK`, and `...`). Models must cite at least one body paragraph line for any non-refusal answer. Titles and abstracts remain visible as context; they are simply not valid evidence.

2. **Discontinuity markers.** Within each paper, retrieved paragraphs are sorted by their original paragraph index, and any gap (the abstract followed by paragraph 2, or paragraph 5 followed by paragraph 9) is marked with a single

Model	Condition	n	Ans-F1	Evid-F1	Judge
GPT-5	LC	50	0.377	0.415	0.480
GPT-5	flat_pool11@3	50	0.357	0.543	0.460
GPT-5	**struct_pool11@3**	50	**0.437**	**0.607**	**0.550**
GPT-5.5	**LC**	50	0.378	0.433	**0.530**
GPT-5.5	flat_pool11@3	50	0.386	0.560	0.460
GPT-5.5	struct_pool11@3	50	**0.419**	**0.600**	0.490
Opus-4.6	**LC**	50	**0.537**	0.488	**0.650**
Opus-4.6	flat_pool11@3	50	0.471	**0.667**	0.500
Opus-4.6	struct_pool11@3	50	0.483	0.500	0.550
Sonnet-4.5	**LC**	50	0.462	0.351	**0.600**
Sonnet-4.5	flat_pool11@3	50	**0.550**	**0.653**	0.590
Sonnet-4.5	struct_pool11@3	50	0.465	0.567	0.510
Gemini-3-Fl	**LC**	50	0.493	0.507	**0.610**
Gemini-3-Fl	flat_pool11@3	50	0.493	**0.653**	0.560
Gemini-3-Fl	struct_pool11@3	50	**0.496**	0.610	0.570
Realistic open-corpus RAG (full dev corpus retrieval)					
GPT-5	flat_full@10	50	0.356	0.609	0.490
Sonnet-4.5	flat_full@10	50	**0.422**	0.636	0.480
Gemini-3-Fl	**flat_full@10**	25[†]	0.382	**0.715**	**0.500**

Table 7: **QASPER-MP aggregate scores** (50 pilot questions, EBG mode). Best condition per model in **bold**. Judge scores use the GPT-5 judge described in Appendix E.5; n reports how many queries returned valid output. Higher (↑) values are better.

[†]Gemini hit transient API errors on this condition; the remaining 25 responses are used.

. . . line. This gives the reader explicit signal that some material is missing between the visible paragraphs.

Under this revised protocol (Table 7), structured RAG recovers: Evidence-F1 jumps by +0.04 to +0.15 per model and `struct_pool11@3` beats or matches `flat_pool11@3` for two of three models. The intuition that paper-level structure is useful is correct—it had been masked by a metric artifact.

E.5 LLM-Judge Evaluation

To control for token-F1 brittleness on paraphrastic answers (especially boolean elaboration and abstractive paraphrase taxes), we additionally score every model output with an LLM judge:

- **Judge model:** GPT-5.
- **Inputs:** the original question, all annotator gold answers, and the model prediction.
- **Output:** a single number in $\{0, 0.5, 1\}$. The judge is instructed to be lenient about formatting and elaboration but strict about content; partial credit (0.5) is reserved for answers that capture some but not all required information. "Unanswerable" vs. "Unanswerable" scores 1.0; "Unanswerable" vs. a real answer (or the reverse) scores 0.0.

Judgments are cached by content hash so repeated scoring is free.

Across all conditions, Judge scores are systematically higher than Answer-F1 (the judge is lenient about phrasing), but the *ordering* of conditions is preserved. The LC vs. RAG gap on answerable questions is larger under the judge than under Answer-F1, because the judge does not penalize Sonnet-4.5 and Gemini-3-Flash for their characteristic boolean elaborations.

E.6 Summary

QASPER-MP corroborates the picture in Appendix D: under apples-to-apples conditions, long-context EBG and RAG end up at comparable aggregate scores, but the mechanism is asymmetric. RAG's higher refusal rate inflates its score on unanswerable questions; on the answerable subset—where retrieval quality actually matters—LC EBG beats every RAG variant for every one of the five frontier models we tested. The gap is largest for the strongest readers (Opus-4.6 and GPT-5.5 each show +0.13 judge points LC over best RAG on answerable questions), suggesting that LC EBG benefits disproportionately from stronger long-context comprehension. The gap widens further when the retrieval corpus expands from the apples-to-apples 11-paper pool to the realistic full-corpus setting,

Model	Condition	Answerable ($n = 38$) Judge	Ans-F1	Refuse %	Unanswerable ($n = 12$) Judge	Ans-F1	Refuse %
GPT-5	**LC**	**0.605**	0.469	15.8	0.083	0.083	33.3
GPT-5	flat_pool11@3	0.500	0.364	31.6	0.333	0.333	58.3
GPT-5	struct_pool11@3	0.553	0.390	31.6	**0.583**	0.583	66.7
GPT-5.5	**LC**	**0.671**	0.471	13.2	0.083	0.083	16.7
GPT-5.5	flat_pool11@3	0.500	0.402	23.7	0.333	0.333	41.7
GPT-5.5	struct_pool11@3	0.539	0.446	21.1	0.333	0.333	25.0
Opus-4.6	**LC**	**0.697**	0.548	13.2	0.500	0.500	41.7
Opus-4.6	flat_pool11@3	0.474	0.436	31.6	**0.583**	0.583	50.0
Opus-4.6	struct_pool11@3	0.566	0.477	23.7	0.500	0.500	41.7
Sonnet-4.5	**LC**	**0.592**	0.446	18.4	0.583	0.513	41.7
Sonnet-4.5	flat_pool11@3	0.513	0.461	36.8	**0.833**	0.833	83.3
Sonnet-4.5	struct_pool11@3	0.474	0.402	34.2	0.667	0.667	58.3
Gemini-3-Fl	**LC**	**0.658**	0.491	15.8	0.500	0.500	50.0
Gemini-3-Fl	flat_pool11@3	0.487	0.412	39.5	**0.750**	0.750	66.7
Gemini-3-Fl	struct_pool11@3	0.539	0.442	31.6	0.667	0.667	58.3

Table 8: **QASPER-MP results split by answerability** (EBG mode, LLM-judge). "Refusal" is the fraction of model outputs that started with `Unanswerable`. Best LC vs. best RAG variant per model in **bold** within each subset. Higher (↑) values are better.

where LC outperforms the best RAG configuration by 5–13 judge points.

F Long-context EBG: Synthetic and NoLiMa Benchmarks

This appendix collects the controlled long-context EBG results that support RQ2–RQ4: the synthetic two-hop benchmark (Sec. F.1) and the NoLiMa-based benchmark (Sec. F.2).

F.1 Ans-only vs EBG on Synthetic Dataset

Observations. Fig. 4 compares evidence-based genderation (**EBG**) against answer-only prompting on the synthetic two-hop benchmark, highlighting both the advantages and limitations of requiring explicit evidence attribution.

- **Performance rapidly saturates for all frontier models.** Across nearly the entire 10K–1M character range, all models operate close to ceiling performance, with accuracies clustering near 100% under both prompt settings. This indicates that the synthetic two-hop benchmark is largely solved by current frontier systems once the relevant evidence is successfully retrieved, and that increasing context length alone does not substantially reduce answer accuracy for the strongest models.
- **Answer-only and EBG prompting achieve nearly identical performance across all context lengths.** For `GPT-5` and `Gemini-3-Fl`, `Ans`* (answer-only prompting) and `Ans` (answer with evidence prompting) remain effectively indistinguishable from 10K to 1M characters, with almost all cells at or near 100% accuracy. The difference heatmap confirms that the vast majority of entries have zero difference, indicating that adding citation requirements does not materially affect answer correctness on this synthetic benchmark.

F.2 Ans-only vs EBG on NoLiMa-based Dataset

Observations. As shown in Table 9, *EBG* prompting is more beneficial for multi-hop queries than for simple one-hop queries. In one-hop settings, many strong models already perform near ceiling under answer-only prompting, leaving little room for explicit evidence organization to help. In some cases, the additional structure introduced by *EBG* even hurts performance, suggesting that it may interfere with models that already perform strong implicit retrieval or that are sensitive to extra reasoning overhead. The benefits of *EBG* become clearer in the two-hop setting, where models must retrieve and combine evidence across multiple pieces of context. The best long-context performers are `GPT-5` and `Gemini-3-Fl`, but for different reasons. `GPT-5` is highly robust regardless of prompting strategy, while `Gemini-3-Fl` gains substantially from *EBG* in two-hop settings.

The strongest evidence in favor of *EBG* comes from the `Gemini-3-Fl` two-hop results, where *EBG* improves performance consistently across al-

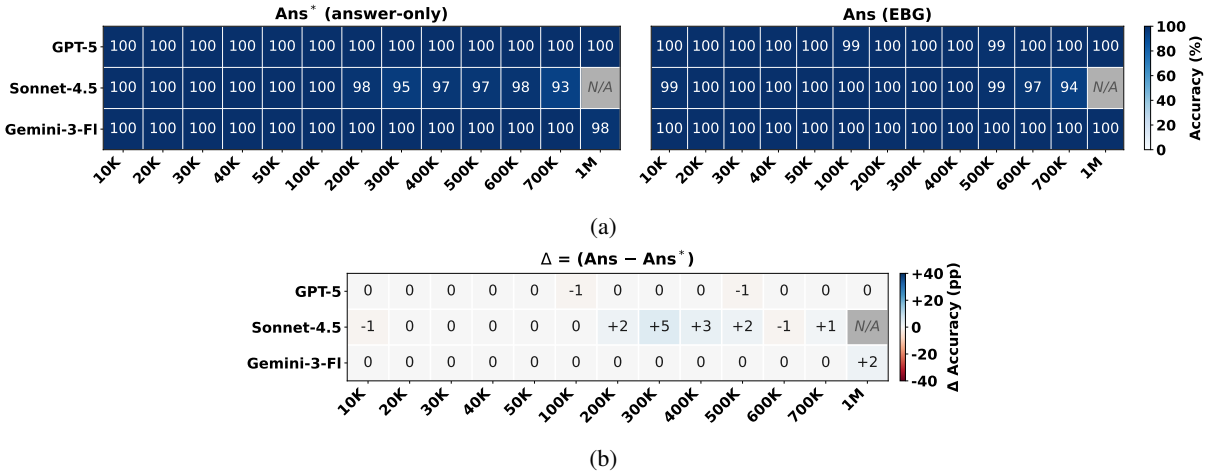

Figure 4: **Synthetic two-hop benchmark comparing Ans* (answer-only prompting) and Ans (under *EBG* prompting)** for three frontier models across 10K–1M characters. (a) Absolute Ans* and Ans accuracies. Higher (↑) values are better. (b) Accuracy difference between the two prompting settings. Cells report percent accuracy averaged over ≥ 50 trials per context length. Sonnet-4.5's 1M entry is marked *N/A* because the input exceeds its 200K-token context window. The two prompting settings produce nearly identical performance on this controlled synthetic benchmark, indicating that the citation requirement neither substantially helps nor hurts answer accuracy. Larger (↑) positive difference indicates effectiveness of *EBG* over *answer-only* prompting.

most all context lengths, including the 1M setting. The Sonnet models also show frequent gains from *EBG* in two-hop tasks, further supporting the view that explicit evidence organization is most useful when reasoning requires multiple retrieval steps. In contrast, the strongest evidence against universal *EBG* use comes from GPT-4o, where *EBG* substantially lowers performance in both one-hop and two-hop tasks.

Overall, the table suggests that *EBG* is especially valuable for long-context multi-hop reasoning in models, but it may be unnecessary for simple one-hop retrieval.

F.3 Strict EBG on NoLiMa-based Dataset

Observations. Fig. 5 reports the combined *Ans+Evid* accuracy across models, context lengths, and reasoning types. Since this metric requires both the final answer and the supporting evidence to be correct, it is a **stricter** measure than answer accuracy alone. The results show clear differences in how well models remain both accurate and grounded as context length increases.

One-hop performance. In the one-hop setting, most frontier models perform very strongly at short and medium context lengths. GPT-5, Sonnet-4, Sonnet-4.5, Gemini-2.5-FI, and Gemini-3-Fl frequently achieve near-perfect or perfect accuracy across many context lengths. This suggests that when only one relevant piece of evidence needs to be retrieved, strong long-context models can usually identify the evidence and produce the correct answer reliably.

Gemini-3-Fl is the strongest one-hop model overall, maintaining 100% accuracy across all evaluated context lengths, including 1M tokens. GPT-5 is also highly robust, achieving 100% accuracy across most context lengths and only dropping slightly to 94% at 1M. Gemini-2.5-Fl also remains strong, although it declines at longer contexts, reaching 81% at 600K and 700K before recovering to 94% at 1M.

GPT-4.1 performs well in one-hop reasoning, but its performance decreases at the largest context length, reaching 81% at 1M. GPT-4o and o3-mini show much weaker robustness. GPT-4o drops substantially after 100K tokens, while o3-mini degrades sharply beyond 200K tokens, falling to 31% at 500K and 19% at 700K.

Two-hop performance. The two-hop setting is significantly harder. Even models that perform very well in one-hop reasoning show larger performance drops when they must retrieve and combine multiple pieces of evidence. This indicates that long-context multi-hop reasoning requires more than simply locating one relevant span; the model must maintain multiple pieces of information and reason over their relationship.

Gemini-3-Fl is again the strongest model over-

Model	10K			20K			30K			40K			50K			100K		
	Ans*	Ans	Δ	Ans*	Ans	Δ	Ans*	Ans	Δ	Ans*	Ans	Δ	Ans*	Ans	Δ	Ans*	Ans	Δ
One-hop																		
GPT-4o	100.0	75.0	**-25.0**	100.0	87.5	**-12.5**	93.8	81.2	**-12.6**	81.2	87.5	**+6.3**	100.0	68.8	**-31.2**	100.0	87.5	**-12.5**
GPT-4.1	100.0	100.0	0.0	100.0	100.0	0.0	100.0	93.8	**-6.2**	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0
O3-mini	87.5	81.2	**-6.3**	87.5	81.2	**-6.3**	87.5	87.5	0.0	87.5	87.5	0.0	87.5	81.2	**-6.3**	68.8	75.0	**+6.2**
GPT-5	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0
Sonnet-4	100.0	100.0	0.0	100.0	100.0	0.0	93.8	100.0	**+6.2**	100.0	100.0	0.0	100.0	100.0	0.0	100.0	93.8	**-6.2**
Sonnet-4.5	100.0	100.0	0.0	100.0	100.0	0.0	87.5	100.0	**+12.5**	100.0	100.0	0.0	93.8	100.0	**+6.2**	100.0	93.8	**-6.2**
Gemini-2.5-Fl	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0
Gemini-3-Fl	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0
Two-hop																		
GPT-4o	87.5	68.8	**-18.7**	93.8	81.2	**-12.6**	87.5	86.7	**-0.8**	75.0	75.0	0.0	87.5	68.8	**-18.7**	87.5	56.2	**-31.3**
GPT-4.1	100.0	100.0	0.0	100.0	93.8	**-6.2**	100.0	93.8	**-6.2**	100.0	100.0	0.0	100.0	100.0	0.0	100.0	93.8	**-6.2**
O3-mini	68.8	68.8	0.0	68.8	68.8	0.0	62.5	62.5	0.0	62.5	56.2	**-6.3**	68.8	68.8	0.0	37.5	50.0	**+12.5**
GPT-5	100.0	87.5	**-12.5**	100.0	93.8	**-6.2**	100.0	93.8	**-6.2**	100.0	93.8	**-6.2**	100.0	93.8	**-6.2**	100.0	86.7	**-13.3**
Sonnet-4	68.8	87.5	**+18.7**	81.2	87.5	**+6.3**	68.8	81.2	**+12.4**	68.8	93.8	**+25.0**	81.2	81.2	0.0	62.5	87.5	**+25.0**
Sonnet-4.5	87.5	100.0	**+12.5**	100.0	100.0	0.0	93.8	93.8	0.0	100.0	93.8	**-6.2**	93.8	100.0	**+6.2**	81.2	87.5	**+6.3**
Gemini-2.5-Fl	100.0	93.8	**-6.2**	93.8	81.2	**-12.6**	68.8	87.5	**+18.7**	62.5	75.0	**+12.5**	68.8	62.5	**-6.3**	56.2	50.0	**-6.2**
Gemini-3-Fl	68.8	68.8	0.0	56.2	75.0	**+18.8**	62.5	80.0	**+17.5**	81.2	93.3	**+12.1**	93.8	93.3	**-0.5**	81.2	93.8	**+12.6**

(a) Performance from 10K to 100K.

Model	200K			300K			400K			500K			600K			700K		
	Ans*	Ans	Δ	Ans*	Ans	Δ	Ans*	Ans	Δ	Ans*	Ans	Δ	Ans*	Ans	Δ	Ans*	Ans	Δ
One-hop																		
GPT-4o	93.8	62.5	**-31.3**	87.5	68.8	**-18.7**	87.5	56.2	**-31.3**	93.8	62.5	**-31.3**	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
GPT-4.1	100.0	100.0	0.0	100.0	100.0	0.0	100.0	87.5	**-12.5**	100.0	93.8	**-6.2**	87.5	93.8	**+6.3**	93.8	93.8	0.0
O3-mini	87.5	75.0	**-12.5**	56.2	43.8	**-12.4**	62.5	37.5	**-25.0**	43.8	31.2	**-12.6**	37.5	25.0	**-12.5**	31.2	18.8	**-12.4**
GPT-5	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0
Sonnet-4	100.0	93.8	**-6.2**	87.5	93.8	**+6.3**	81.2	93.8	**+12.6**	68.8	87.5	**+18.7**	87.5	87.5	0.0	75.0	81.2	**+6.2**
Sonnet-4.5	100.0	100.0	0.0	81.2	87.5	**+6.3**	75.0	87.5	**+12.5**	75.0	75.0	0.0	75.0	87.5	**+12.5**	56.2	75.0	**+18.8**
Gemini-2.5-Fl	100.0	100.0	0.0	100.0	100.0	0.0	93.8	100.0	**+6.2**	87.5	100.0	**+12.5**	100.0	87.5	**-12.5**	100.0	87.5	**-12.5**
Gemini-3-Fl	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0
Two-hop																		
GPT-4o	87.5	43.8	**-43.7**	75.0	31.2	**-43.8**	75.0	43.8	**-31.2**	87.5	62.5	**-25.0**	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
GPT-4.1	93.8	100.0	**+6.2**	100.0	100.0	0.0	100.0	87.5	**-12.5**	87.5	81.2	**-6.3**	81.2	81.2	0.0	87.5	87.5	0.0
O3-mini	68.8	56.2	**-12.6**	31.2	37.5	**+6.3**	37.5	43.8	**+6.3**	18.8	37.5	**+18.7**	18.8	25.0	**+6.2**	12.5	18.8	**+6.3**
GPT-5	100.0	100.0	0.0	100.0	100.0	0.0	100.0	93.8	**-6.2**	100.0	93.3	**-6.7**	93.8	93.8	0.0	93.8	100.0	**+6.2**
Sonnet-4	43.8	68.8	**+25.0**	43.8	62.5	**+18.7**	50.0	56.2	**+6.2**	37.5	56.2	**+18.7**	68.8	81.2	**+12.4**	37.5	50.0	**+12.5**
Sonnet-4.5	50.0	62.5	**+12.5**	43.8	62.5	**+18.7**	50.0	75.0	**+25.0**	37.5	50.0	**+12.5**	68.8	75.0	**+6.2**	50.0	50.0	0.0
Gemini-2.5-Fl	62.5	56.2	**-6.3**	50.0	56.2	**+6.2**	50.0	75.0	**+25.0**	37.5	37.5	0.0	31.2	25.0	**-6.2**	31.2	25.0	**-6.2**
Gemini-3-Fl	81.2	87.5	**+6.3**	87.5	93.8	**+6.3**	62.5	86.7	**+24.2**	68.8	100.0	**+31.2**	81.2	93.8	**+12.6**	75.0	87.5	**+12.5**

(b) Performance from 200K to 700K.

Model	800K			900K			1M		
	Ans*	Ans	Δ	Ans*	Ans	Δ	Ans*	Ans	Δ
One-hop									
GPT-4o	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
GPT-4.1	100.0	81.2	**-18.8**	87.5	75.0	**-12.5**	87.5	81.2	**-6.3**
O3-mini	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
GPT-5	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0
Sonnet-4	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
Sonnet-4.5	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
Gemini-2.5-Fl	87.5	93.8	**+6.3**	93.8	75.0	**-18.8**	100.0	93.8	**-6.2**
Gemini-3-Fl	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0	0.0
Two-hop									
GPT-4o	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
GPT-4.1	93.8	81.2	**-12.6**	81.2	62.5	**-18.7**	75.0	68.8	**-6.2**
O3-mini	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
GPT-5	100.0	87.5	**-12.5**	86.7	80.0	**-6.7**	93.8	87.5	**-6.3**
Sonnet-4	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
Sonnet-4.5	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
Gemini-2.5-Fl	43.8	37.5	**-6.3**	37.5	18.8	**-18.7**	37.5	31.2	**-6.3**
Gemini-3-Fl	56.2	80.0	**+23.8**	73.3	91.7	**+18.4**	81.2	93.3	**+12.1**

(c) Performance from 800K to 1M.

Table 9: Performance across different context lengths, **comparing `Ans`* (answer-only) with `Ans` (EBG)**. Δ denotes the difference `Ans` − `Ans`*; positive values are shown in bold **blue**, negative values in bold **red**. Higher (↑) values are better. Positive values indicate that *EBG* prompting outperforms *answer-only* prompting, while negative values indicate that *answer-only* prompting improves over *EBG* prompting. Zero indicates no difference between the two prompting strategies. "N/A" denotes unavailable entries due to unsupported context lengths.

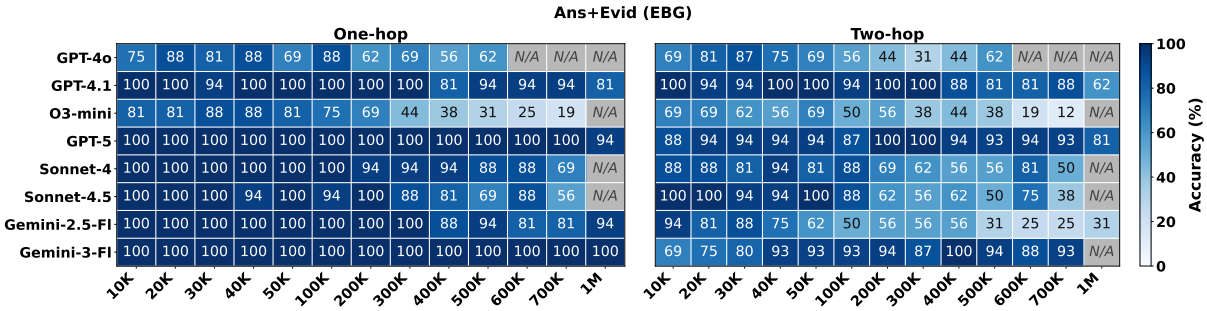

Figure 5: Performance of different models on **Ans+Evid (EBG)** accuracy for **one-hop and two-hop** questions across varying context lengths, averaged over needle depths. Cells marked with "N/A" indicate no available results for that model-context combination. Higher (↑) values are better.

	Context Length																		
	10K	**20K**	**30K**	**40K**	**50K**	**60K**	**70K**	**80K**	**90K**	**100K**	**200K**	**300K**	**400K**	**500K**	**600K**	**700K**	**800K**	**900K**	**1M**
# Vectors	66	124	173	247	297	381	395	470	513	584	1,179	1,739	2,275	2,852	3,468	4,034	4,658	5,247	5,730
Full-coverage (one-hop)																			
# Vectors	2	4	4	4	4	9	11	8	9	10	20	35	43	48	59	62	83	92	96
Chars	350	823	818	607	584	1,902	2,343	1,876	1,985	2,452	4,725	7,126	8,244	11,548	12,410	13,981	17,771	20,234	22,910
Full-coverage (two-hop)																			
# Vectors	10	21	25	36	42	63	71	82	78	92	169	276	348	444	528	635	750	821	890
Chars	1,956	4,756	5,875	6,771	9,906	12,600	15,902	16,841	17,236	20,567	39,052	63,009	77,792	99,519	117,251	136,473	164,002	177,879	202,412

Table 10: **Retrieval analysis of *RAG* under NoLiMa**, showing average total vector count; minimum full-coverage retrieval depth for one-hop and two-hop queries (i.e., the smallest top-k such that all required needles/evidence facts are contained within the retrieved set for every sample) and corresponding retrieved character budget across varying haystack context lengths, averaged over three books.

all in the two-hop setting. It maintains high accuracy across the full context range and achieves 93% at 1M tokens. `GPT-5` is also strong, especially from 10K to 700K, but drops to 81% at 1M. `GPT-4.1` performs well up to mid-range contexts but declines at 1M, reaching 62%.

G RAG on NoLiMa-based Dataset

We use `text-embedding-3-small` to generate embeddings for both the NoLiMa haystack lines (with inserted needles) and the corresponding queries, and perform retrieval using cosine similarity.

G.1 Retrieval Analysis

Table 10 summarizes how the retrieval requirements of a RAG system changes as the haystack context length grows from 10K to 1M. The first row reports the average number of vectors, or indexed chunks, produced from the full context. As expected, this number increases with context length: for example, the retriever searches over only 66 vectors at 10K context, but over 5,730 vectors at 1M context.

The remaining rows measure the retrieval depth required for *full-coverage*. Here, full coverage means that the retrieved top-k set contains all evidence facts needed to answer every query. Thus, the reported number of vectors is the smallest value of k such that no required evidence is missed.

For one-hop queries, where the answer depends on a single evidence fact, the required retrieval depth grows moderately with context length. At 10K context, retrieving only 2 vectors, corresponding to 350 characters, is sufficient for full coverage. However, at 1M context, full coverage requires retrieving 96 vectors, corresponding to 22,910 characters. This shows that even relatively simple queries require a larger retrieval budget as the haystack becomes longer.

For two-hop queries, where the answer depends on combining multiple evidence facts, the retrieval requirement grows much more sharply. At 10K context, full coverage requires retrieving 10 vectors, or 1,956 characters. At 1M context, however, it requires retrieving 890 vectors, or 202,412 characters. This indicates that multi-hop evidence is substantially harder to retrieve in long-context settings, since the relevant pieces of information may be distributed across many more chunks.

Several observations follow from these results. First, the total number of indexed vectors grows approximately linearly with context length, increasing from 66 vectors at 10K to 5,730 vectors at 1M. Second, the retrieval depth required for full

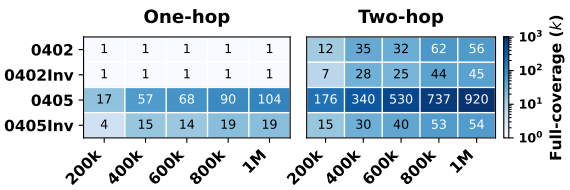

Figure 6: Minimum **full-coverage retrieval depth** (k) (for book B_1^c under NoLiMa). Lower (↓) values are better.

coverage is much smaller than the total number of indexed vectors, but it still grows substantially as the context becomes longer. Third, two-hop queries are consistently more demanding than one-hop queries: at 1M context, two-hop full coverage requires 890 vectors, compared with only 96 vectors for one-hop full coverage. This is more than a 9× increase in the number of retrieved vectors:

$$\frac{890}{96} \approx 9.27.$$

Fourth, the retrieved character budget also grows significantly. At 1M context, one-hop full coverage requires 22,910 characters, whereas two-hop full coverage requires 202,412 characters. Thus, two-hop retrieval requires nearly nine times more retrieved text:

$$\frac{202{,}412}{22{,}910} \approx 8.83.$$

Another important observation is that standard small retrieval budgets, such as top-5 or top-10, are unlikely to provide full coverage in long-context settings. For instance, at 100K context, one-hop full coverage already requires top-10, while two-hop full coverage requires top-92. At 1M context, the required two-hop retrieval depth increases to top-890, which is far beyond typical RAG configurations.

Overall, the table shows that standard small top-k retrieval settings may be insufficient for long-context RAG, especially for multi-hop queries. While one-hop retrieval remains relatively manageable, two-hop retrieval at 1M context requires retrieving a large fraction of the indexed chunks:

$$\frac{890}{5730} \approx 15.5\%.$$

Therefore, as context length increases, guaranteeing full evidence coverage can require a very large retrieved character budget, reducing the efficiency advantage usually associated with RAG.

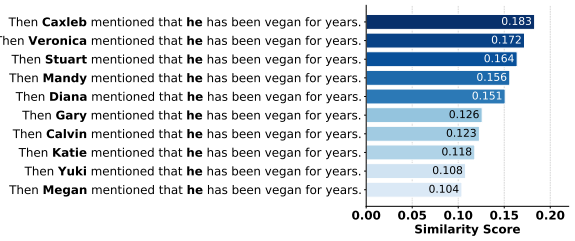

Figure 7: **Ranking of the `0405` needle instantiated with 10 different character names** based on cosine similarity to a one-hop test query. Although the sentence semantics remain unchanged, similarity scores vary across character tokens. Higher (↑) values are better.

G.2 Character and Pronoun Sensitivity in Needle Retrieval

Motivation. During retrieval analysis, we observed that the `0405` needle family requires substantially larger top-k budgets to achieve reliable retrieval compared to the `0402` family (Figure 6). This suggests that retrieval difficulty may be influenced not only by reasoning complexity but also by the *surface form of the needle itself*. The default `0405` template contains a gendered pronoun:

> *"Then {CHAR} mentioned that he has been vegan for years."*

Because the pronoun *he* is independent of the inserted character name, it may introduce additional semantic signals in the embedding representation that are not directly relevant to the query.

Character sensitivity analysis. To investigate this effect, we compute cosine similarity between the query embedding and needle sentences instantiated with different character names drawn from the NoLiMa corpus. Figure 7 shows the top retrieved variants of the needle and their similarity scores for the query *"Which character cannot eat Brandade?"*.

Although the sentence semantics remain unchanged, we observe noticeable variation in similarity scores across character substitutions. This indicates that character tokens themselves influence the embedding representation and can affect retrieval ranking.

Effect of pronoun removal. To isolate the role of the gendered pronoun, we construct a rephrased variant of the needle (`0405-R`, Table 2) that removes the pronoun while preserving the meaning:

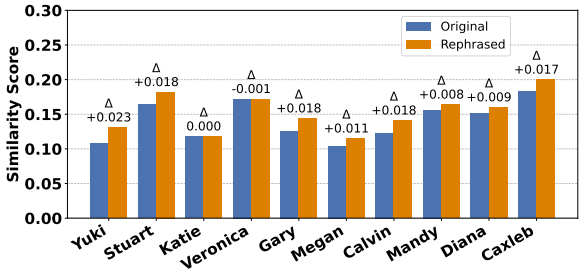

Figure 8: Cosine similarity **comparison between the original `0405` needle and the rephrased `0405-R` variant across 10 characters**. Removing the pronoun consistently increases similarity scores across different characters. Higher positive difference (↑) indicates greater effectiveness of the rephrased version compared to the original.

> *"Then {CHAR} mentioned being vegan for years."*

We recompute similarity scores for both the original and rephrased needles across multiple character substitutions. Figure 8 compares cosine similarity between the two formulations.

Across most characters, the rephrased template yields slightly higher similarity scores. While the improvement is modest for individual instances, it is systematic across characters, suggesting that removing the pronoun reduces semantic noise in the embedding space.

Implications for RAG retrieval. These results help explain the higher retrieval budget required for the `0405` needle family observed earlier. When the needle contains additional syntactic elements that are not directly aligned with the query semantics, embedding similarity becomes less discriminative. In large haystacks, even small similarity differences can significantly affect ranking, pushing the correct needle lower in the retrieval list and increasing the required top-k budget.

G.3 TN vs. CTN: Adherence to Contextual Evidence

We conduct an experiment on the NoLiMa-based dataset to evaluate whether LLMs remain faithful to contextual evidence in two-hop reasoning questions when presented with either factual secondary evidence that aligns with world knowledge (*TN*) or counterfactual secondary evidence that conflicts with world knowledge (*CTN*; e.g., "black coffee has milk").

Experiment. Under TN and CTN, each query is associated with two needles. We insert the primary needle is at the middle of the haystack (depth 0.5) and the secondary needle at one of the sampled depths $\{0.0, 0.33, 0.67, 1.0\}$. Since questions must be answered strictly from the provided context, success in CTN requires relying on the contradictory contextual evidence rather than parametric memory. We evaluate the performance of TN and CTN using `Ans` and `Ans+Evid` accuracy metrics.

Observations.

- **Contradiction reliably hurts performance.** Averaged across models, CTN reduces `Ans` accuracy by ≈**7.0 pp** (76.2% → 69.2%) and `Ans+Evid` by ≈**5.9 pp** (49.9% → 44.0%).
- **The Ans+Evid accuracy metric is much harder than Ans.** `Ans+Evid` is far below `Ans` in both *TN* and *CTN*, indicating many correct-answer cases still fail to produce a complete evidence trail.
- **Robustness varies sharply by model.** `Gemini-3-Fl` and `GPT-5` remain most stable; the largest `Ans+Evid` drops under *CTN* occur for `GPT-4.1` (≈**-13 pp**), `Gemini-2.5-Fl` (≈**-10 pp**), and `o3-mini` (≈**-8.6 pp**).
- **Depth sensitivity persists under contradiction,** with the CTN penalty tending to be larger when the secondary needle is deeper, consistent with degraded multi-evidence retrieval under harder placements, i.e., mostly at depths 0.33 and 0.67.
- **CTN remains below TN across context lengths**, indicating contradiction amplifies long-context brittleness as both conditions degrade with length.
- **Contradiction weakens evidence grounding, not just answers.** CTN lowers citation rates for both needles (primary ≈**-7.4 pp** and secondary ≈**-4.6 pp**), pointing to provenance failure as a primary failure mode under conflicting information.

G.4 Randomly Sampled Haystacks vs. Top-k Ranked Retrieval

Setup. We examine whether LLMs answer more accurately when we give them a randomly sampled long-context haystack or a compact context formed by top-k ranked retrieval. In the long-context setting, we provide the model with the Book 1 haystack at a target character length and insert the needle at predefined depths. This setting corresponds to randomly sampled haystacks:

LC (Chars)	Full-coverage (k) (Vectors)	Full-coverage (k) (Chars)	LC (matched) (Chars)
200K	176	45,742	50K
400K	340	72,493	70K
500K	422	93,694	90K
600K	530	113,052	100K
1M	920	217,152	200K

Table 11: **Character-budget matching procedure**. For each long-context (LC) haystack used during the retrieval stage on book B_1^c, we compute the minimum top-k such that the retrieved set contains all required needles and evidence facts for every sample. We then measure the character count of this retrieved set and match it to the LC condition with the closest character count. The columns report, from left to right, the original LC haystack size, the minimum top-k (in vectors), its character count, and the matched LC budget used for comparison.

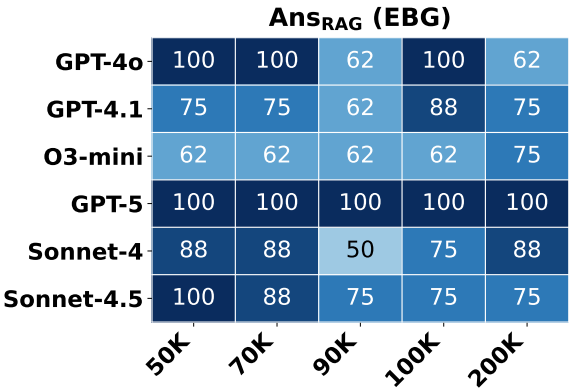

Figure 9: **Retrieval answer accuracy under the EBG prompting for the matched top-k retrieval contexts**. Columns denote the long-context character budgets we use for comparison after matching each retrieved top-k context to the nearest long-context haystack length. Rows denote the evaluated models. Each cell reports answer accuracy, $\mathrm{Ans}_{\mathrm{RAG}}$, after we condition the model on the retrieved top-k chunks rather than the randomly sampled long-context haystack. Higher (↑) values are better.

the long-context construction procedure embeds the relevant evidence within a larger context and determines the surrounding distractors and needle positions. In contrast, the retrieval setting replaces the full randomly sampled haystack with a retrieval-ranked context. For each query, we compute cosine similarity between the query embedding and each chunk embedding in the corresponding vector database, rank all chunks by similarity score, and select the minimum top-k prefix that contains all gold needle evidence for that condition. We call this value the best top-k.

Because long-context and retrieval inputs differ substantially in structure, we compare them under approximately matched character budgets rather than matched nominal context labels. For each Book 1 haystack, we first compute the total character count of the best top-k retrieved chunks. We then select the long-context condition whose haystack character count most closely matches that retrieval character count. For example, the best top-k retrieved context for the 1M-character haystack contains 217,152 characters; we therefore compare this retrieval condition against the 200K-character long-context condition, since the two inputs match most closely in character budget. Throughout, "context length" denotes character count rather than token count.

We run the retrieval prompting experiment using the same answer+evidence prompt template as the long-context EBG setting. We construct the retrieval prompt from the selected top-k chunks together with the assigned gold evidence metadata, and we require the model to produce both an answer and cited evidence. We evaluate answer accuracy for the retrieval setting, which we denote $\mathrm{Ans}_{\mathrm{RAG}}$, and compare it with answer accuracy from the matched long-context setting, which we denote $\mathrm{Ans}_{\mathrm{LC}}$. The primary comparison is the accuracy delta:

$$\Delta\mathrm{Ans} = \mathrm{Ans}_{\mathrm{RAG}} - \mathrm{Ans}_{\mathrm{LC}}.$$

A positive value means that top-k ranked retrieval outperforms randomly sampled long-context prompting under the matched character budget, while a negative value means that randomly sampled long-context prompting outperforms top-k ranked retrieval.

Table 11 summarizes the character-budget matching procedure. The table reports, for each original Book 1 haystack length, the best top-k value, the character count of the corresponding retrieved context, and the matched long-context character budget we use for comparison.

Figure 9 reports retrieval answer accuracy under the answer+evidence prompt, and Figure 10 reports the resulting $\Delta\mathrm{Ans}$ values across models and matched context lengths.

Observations Figure 10 shows that the relative advantage of top-k ranked retrieval versus randomly sampled long-context prompting depends strongly on the model. Since each cell reports

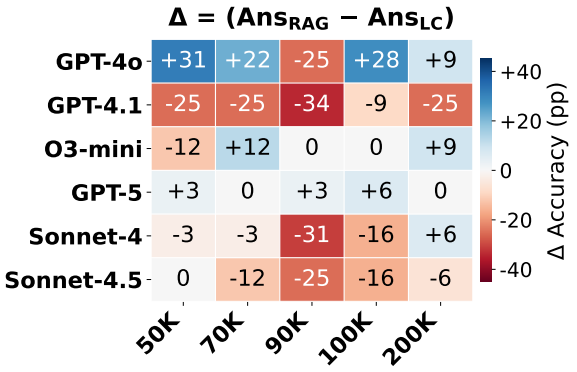

Figure 10: **Answer accuracy difference between RAG-EBG and LC-EBG**, compared under approximately matched character budgets (Table 11). Positive values indicate that RAG-EBG outperforms LC-EBG; negative values indicate that LC-EBG outperforms RAG-EBG. Higher (↑) positive values are better for RAG.

$\Delta\text{Ans} = \text{Ans}_{\text{RAG}} - \text{Ans}_{\text{LC}}$, positive values indicate that the top-k retrieved context performs better, whereas negative values indicate that the randomly sampled long-context haystack performs better.

Overall, top-k ranked retrieval competes strongly with long-context prompting under the matched character-budget setting. For GPT-4o, retrieval outperforms long-context prompting in four of the five matched budgets: 50K (+31 pp), 70K (+22 pp), 100K (+28 pp), and 200K (+9 pp). GPT-4o therefore benefits substantially when we replace the larger randomly sampled haystack with a compact retrieval-ranked context. GPT-5 also shows near parity between the two settings, with deltas close to zero across all budgets (+3, 0, +3, +6, and 0 pp). For GPT-5, once the retrieved top-k context contains the relevant evidence, the additional surrounding haystack context provides little or no answer-accuracy benefit.

In contrast, GPT-4.1 consistently favors the randomly sampled long-context haystack. Its deltas remain negative at every matched budget, ranging from −9 pp at 100K to −34 pp at 90K. GPT-4.1 therefore exploits information from the broader haystack context more effectively than from the compact top-k context, even when the retrieved context contains the necessary gold evidence. The Claude models show a similar but weaker long-context preference. Sonnet-4 improves under long-context prompting at 50K, 70K, 90K, and 100K, with the largest gain at 90K (−31 pp), but retrieval performs slightly better at 200K (+6 pp). Sonnet-4.5 also favors long-context prompting at most matched budgets, especially 90K (−25 pp) and 100K (−16 pp), while matching retrieval at 50K and showing only a small long-context advantage at 200K (−6 pp). O3-mini swings unpredictably across budgets in both directions: long-context prompting helps at 50K (−12 pp), retrieval helps at 70K and 200K (+12 pp and +9 pp), and the two settings tie at 90K and 100K.

A notable pattern emerges at the 90K matched budget, which favors long-context prompting most strongly. Four models show large negative deltas at 90K: GPT-4o (−25 pp), GPT-4.1 (−34 pp), Sonnet-4 (−31 pp), and Sonnet-4.5 (−25 pp). Evidence completeness alone therefore does not guarantee retrieval superiority. Even when the top-k context contains the required needles, the model may still benefit from the broader discourse, surrounding distractors, or positional structure of the original haystack. At the same time, the strong positive deltas for GPT-4o and the near-zero deltas for GPT-5 show that compact retrieval removes distractor burden without hurting answer accuracy for some models.

Overall, the results do not support a single universal winner between randomly sampled haystacks and top-k ranked retrieval. They instead reveal a model-specific tradeoff. Top-k retrieval outperforms long-context prompting when compact evidence-focused contexts help the model avoid distraction, as GPT-4o demonstrates most clearly. Long-context prompting outperforms retrieval when models benefit from the broader haystack context, as GPT-4.1 and the Claude models demonstrate. GPT-5 emerges as the most retrieval-robust model in this comparison, showing that a sufficiently strong model answers equally well from the compact retrieved context and the matched randomly sampled haystack.

Dataset	Source	License
Synthetic	Ours	CC By 4.0
NoLiMa-based	Modarressi et al. (2025)	ADOBE RESEARCH LICENSE
MultiHop-RAG	Tang and Yang (2024)	ODC-By
QASPER-MP	Dasigi et al. (2021)	CC By 4.0

Table 12: Datasets used in this work, along with their source references and licenses.

H Dataset Source Information

Table 12 summarizes all datasets used in this work along with their sources and licenses. We construct or adapt all datasets from publicly available sources. We construct the synthetic two-hop haystack en-

Model	Provider	Source URL	License
GPT-4o	OpenAI	https://platform.openai.com/docs/models/gpt-4o	Proprietary
GPT-4.1	OpenAI	https://platform.openai.com/docs/models/gpt-4.1	Proprietary
o3-mini	OpenAI	https://platform.openai.com/docs/models/o3-mini	Proprietary
GPT-5	OpenAI	https://platform.openai.com/docs/models/gpt-5	Proprietary
GPT-5.5	OpenAI	https://platform.openai.com/docs/models/gpt-5.5	Proprietary
Sonnet-4	Anthropic	https://www.anthropic.com/claude/sonnet	Proprietary
Sonnet-4.5	Anthropic	https://www.anthropic.com/claude/sonnet	Proprietary
Opus-4.6	Anthropic	https://www.anthropic.com/claude/opus	Proprietary
Gemini-2.5-Fl	Google	https://ai.google.dev/gemini-api/docs/models/gemini-2.5-flash	Proprietary
Gemini-3-Fl	Google	https://docs.cloud.google.com/gemini-enterprise-agent-platform/models/gemini/3-flash	Proprietary

Table 13: Models benchmarked in this work, along with their providers, source URLs, and licenses.

tirely using iterative LLM prompting for distractor generation and manually curated needles, and we release it under CC BY 4.0. The NoLiMa-based dataset adapts the haystack construction procedure and needle set from Modarressi et al. (2025), which is released under the Adobe Research License. We use MULTIHOP-RAG (Tang and Yang, 2024) as released under the ODC-By license, and re-map gold evidence facts to line-level labels for EBG evaluation. We derive QASPER-MP from the QASPER dev split (Dasigi et al., 2021), released under CC BY 4.0, and extend it to a multi-paper setting by pairing each gold paper with ten distractor papers and rewriting questions with short realistic paper cues.

I Model Information

We evaluate ten proprietary frontier models accessed via their respective APIs. Table 13 lists each model along with its provider, source URL, and license. All models were queried at temperature 0 (or the provider-equivalent default) with no additional fine-tuning or adaptation.

J Artifact Use and Intended Use

We use all existing datasets and models consistently with their intended use. MULTIHOP-RAG, QASPER, and NOLIMA are used strictly for research evaluation, consistent with their release conditions. All proprietary models are accessed via official APIs in accordance with provider terms of service. All derived datasets introduced in this work are intended exclusively for research use and do not conflict with the access conditions of the original artifacts.

K Disclosure of AI Use

We used AI assistants (large language models) at several stages of this work. We used them to review and debug code for our experimental pipelines and evaluation scripts, to polish and refine the writing of the manuscript by improving clarity, tightening phrasing, and catching grammatical errors, and to suggest alternative phrasings for titles, figure captions, and section headings. We developed all research ideas, experimental designs, benchmark constructions, analyses, and conclusions ourselves. We verified all AI-assisted code changes through testing, reviewed and edited all AI-assisted text, and take full responsibility for the final content of the paper.