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

Debopriyo Banerjee[1*], Owais Aijaz[2*], Ahmer Tabassum[2*], Zhuohan Xie[2]
Preslav Nakov[2], Veselin Stoyanov[2]
[1]Inception42 [2]MBZUAI
{deb.ban89, vesko.st}@gmail.com preslav.nakov@mbzuai.ac.ae

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 passages a question requires and discards document-level context that resolves cross-passage references. We introduce *evidence-based generation* (EBG), requiring models to return an answer with line-level citations, and a joint `Ans+Evid` metric that scores correctness and grounding simultaneously. We reformulate two 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 NoLiMa spanning 10K–1M characters. Across eight proprietary frontier and three open-weight LLMs, EBG isolates two 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 also 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. Data and code are available at https://github.com/debobanerjee/LC-EBG.

1 Introduction

Large Language Models (LLMs) can rapidly process large text volumes to identify information relevant to a query. This capability suits applications such as legal discovery, medical analysis, financial due diligence, literature review, etc., which require synthesis from 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.

[*]Equal contribution.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 47578–47613
October 24-29, 2026 ©2026 Association for Computational Linguistics

We make the following contributions:

- **EBG framework.** We define the evidence-based generation task and a joint `Ans+Evid` metric, and evaluate eight proprietary frontier and three open-weight LLMs under matched LC and RAG conditions 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, 2024), which mainly measure retrieval of planted information rather than reasoning. Subsequent work argues that such evaluations set the bar too low, since strong performance often reflects shallow heuristics rather than genuine understanding (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 characters, HELMET (Yen et al., 2025) shows that 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 and enabling a more faithful assessment of whether models use long contexts effectively and reliably.

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.

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
```

Under answer-only evaluation, the response "`Stuart`" is simply marked correct. But 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. Together, these metrics separate answer correctness from evidence-grounded correctness.

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"}
```

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.

We also evaluate vanilla Chain-of-Thought (CoT) prompting, following Modarressi et al. (2025), and a CoT-based EBG prompting variant. The prompt templates are in Appendix A.

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 four complementary benchmarks: a fully synthetic benchmark, an evidence-augmented version of NoLiMa, MultiHop-RAG, and QASPER-MP (an adapted version of QASPER). 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. This controlled construction allows us to vary context length independently of task difficulty while retaining exact ground-truth evidence locations.

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. The large pool also limits repeated distractors across independently generated trials and provides sufficient diversity for constructing long haystacks.

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 *company–product* associations, *founder–specialty* links, and *birthplace* lookups. By requiring composition across two independently placed facts, the benchmark tests both evidence retrieval and integration rather than simple single-fact recall.

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 (§ A). Each trial uses a newly generated haystack and a uniformly sampled needle. Consequently, performance differences cannot be attributed to a fixed ordering of distractors or evidence positions. 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 and readily solvable when context length is modest. Any degradation at larger L therefore reflects failures in long-context evidence retrieval or integration rather than intrinsic task ambiguity. This makes the synthetic benchmark a controlled probe of how model reliability changes as relevant evidence becomes increasingly sparse within the surrounding context.

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 three random books B_1^c, B_2^c, B_3^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 question answering (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 queries (seed 0), naturally stratified across the three answerable types (n_{inf}=13, n_{comp}=21, n_{temp}=16).

For the LC condition, we construct an *oracle-containing* per-query mini-haystack comprising every gold-evidence article plus 20 distractor articles sampled from the remainder of the corpus. The articles are concatenated in random order and split into sentence-per-line entries, producing contexts of approximately 241K characters on average. For the RAG condition, we perform *open-corpus retrieval* over the full 609-article corpus: we divide the corpus into 256-token sliding windows (6,296 chunks), embed the chunks with `text-embedding-3-large`, and rank them by cosine similarity to produce top-K contexts for $K \in \{4, 10, 25, 50, 100, 200\}$.

The two conditions are therefore not pool-matched: LC evaluates reading and evidence utilization when all gold source articles are guaranteed present, whereas RAG must first retrieve the required evidence from the full corpus. Accordingly, this experiment diagnoses the contribution of open-corpus retrieval failures to end-to-end performance; it is not a direct LC–RAG comparison under identical candidate pools. In both conditions, gold facts are re-mapped to line numbers inside the assembled context; in RAG, a gold fact that is not retrieved has no corresponding gold line. Appendix D reports the corresponding 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 rewrite every question to include a realistic paper cue - in ∼50% of cases a first-author surname (e.g., "Smith et al."), and otherwise a 2–3 word verbatim title fragment - split by a hash of the question id. This forces the system to resolve document identity from a realistic cue rather than the full paper title. The task jointly tests document identification, evidence localization, and answer generation. Since distractors come from the same NLP-paper collection, they provide domain-matched context. A successful system must identify the relevant paper and supporting paragraph(s) before answering, while retaining QASPER's expert-annotated answers and evidence.

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 evaluate QASPER-MP at two scales. For proprietary models, we conduct a pilot on N=50 queries, comprising 38 answerable and 12 unanswerable examples. For open-weight models, we expand the evaluation to N=200 queries, providing broader benchmark coverage and a more stable estimate of aggregate performance. Because the proprietary-model results are based on the smaller pilot, comparisons across the two model groups should be interpreted with the corresponding difference in sampling uncertainty in mind. Construction details, sampling procedures, prompts, and the LLM-judge protocol are provided 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 models to explicitly cite 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`. Additionally we evaluate three open-weight models served through Fireworks.ai[1] - `GPT-OSS-120B`, `GLM-5.2`, and `Nemotron-3.5-Lightning-30B-A3B` (or `Nemo-3.5-L` in short) - to test whether our findings hold beyond proprietary systems. 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.

5.1.1 Multi-document QA setup.

For **RQ1**, we use two multi-document QA datasets - MULTIHOP-RAG and QASPER-MP. 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. Here we apply the *answer-only* and *EBG* prompts and report `Ans*`, `Ans`, `Evid`, and `Ans+Evid` accuracies. A sanity check at L=10K confirms all models achieve $\geq$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 controlled evaluation setup. We use the `T01` test variant throughout, as it provides exactly one input argument for the one-hop question and one for the two-hop question associated with each needle. This enables a consistent comparison between direct retrieval and commonsense composition while holding the underlying needle fixed. For example, for `0405`, the one-hop question is instantiated with *fish-based meals*, whereas the two-hop question uses *Brandade*; similarly, `0402` uses *milk* and *a cappuccino*, respectively. We test two characters, *Yuki* and *Stuart*, at the insertion depths defined in § 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 evidence, addressing **RQ3**), and *contradictory two-needle* (CTN, addressing **RQ4**). Results are reported separately for one-hop and two-hop questions and averaged across the corresponding conditions.

5.1.3 NoLiMa-based RAG Setup

We compare long-context *LC* and *RAG* on the NoLiMa-based dataset under **RQ1** (Appendix G). We first analyze RAG retrieval performance for the single-needle and two-needle settings using one-hop and two-hop questions, examining how the retrieval depth required for full evidence coverage changes with haystack length and reasoning complexity. After needle insertion, each haystack line is embedded using OpenAI's `text-embedding-3-small` model and ranked by cosine similarity to the query.

We then compare *RAG* and *LC* under the same *EBG* prompting, reporting `Ans` accuracy. Since RAG contexts can be smaller than their source haystacks, we use matched character budgets. For each haystack length, we identify the minimum top-k providing full evidence coverage, compute its character count, and match it to the closest available LC context length. Table 15 summarizes this mapping. For example, full-coverage retrieval from the 1M-character haystack in the two-needle setting requires 920 vectors (217,152 characters) and is matched to the 200K LC condition. This approximately controls for the amount of text presented to the model and isolates retrieval-ranked versus long-context inputs under comparable budgets.

[1] https://fireworks.ai/

5.2 Results

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

RQ1: LC-EBG addresses two RAG limitations. On MULTIHOP-RAG, queries require 2–4 evidence facts spread across distinct articles, so faithful answering requires the retriever to surface all required facts within its top-k budget. For this benchmark, the two conditions use different evidence-access regimes: *open-corpus RAG* searches 6,296 chunks constructed from all **609 articles**, whereas the *oracle-containing LC* condition receives every gold-evidence article plus 20 distractor articles. The conditions are therefore not pool-matched; this comparison diagnoses the cost of open-corpus retrieval relative to reading when the gold source articles are guaranteed to be available.

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 annotated-gold retrieval coverage 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 exposes a limitation of the evaluated similarity-retrieval setup: it often retrieves some relevant evidence but fails to recover the complete evidence set required for multi-fact queries within a practical top-k budget.** When the gold annotations are exhaustive and every annotated fact is necessary, all-gold coverage bounds the faithfully grounded performance of the downstream RAG reader. The reader results corroborate this limitation (Appendix D.5, Table 6).

Under *EBG* prompting at k=10, open-corpus *RAG* achieves `Ans` accuracies of 60/48/44% for `GPT-5`, `Sonnet-4.5`, and `Gemini-3-Fl`, respectively, whereas the oracle-containing *LC-EBG* condition achieves 86/82/78% on the same queries. The resulting 26–34 pp gap includes the cost of retrieving the required evidence from the full corpus and should not be interpreted as a pool-matched estimate of an intrinsic LC–RAG difference. The gap remains at k=200, where the open-corpus RAG context approaches the size of the LC haystack despite comparable context sizes, but its `Ans` accuracy remains lower for all three models. Under *answer-only* prompting, open-corpus RAG at k=200 reaches parity with the oracle-containing LC condition only for `Sonnet-4.5` (68% vs. 68%).

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-subset 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.

Consistent with this, our NOLIMA-based RAG analysis (Table 13) 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 document-level identity, not just topical fragments. **This exposes a second limitation of RAG: chunk-level similarity retrieval struggles with document identity and provenance when queries refer indirectly to a source.** Table 1 reports answerable-subset judge scores for the pool-matched comparison, where *LC-EBG* and *RAG* use the same 11-paper pool. *LC-EBG* exceeds the best pool-matched RAG variant for all five frontier models by +0.05 to +0.13 judge points. Aggregate scores look tied only because *RAG* over-refuses on missed retrievals (24–40% on answerable queries vs. 13–18% for *LC-EBG*), accidentally matching QASPER's `Unanswerable` gold label on the 24% of pilot questions for which it is correct. Under the open-corpus condition (`flat_full@10`), all-needles coverage drops to 48%, and the judge-score gap relative to LC is 5–13 points (Appendix E).

Across both datasets, retrieval is a major error source despite different evidence-access regimes: MULTIHOP-RAG compares open-corpus RAG with oracle-containing LC, whereas QASPER-MP uses a pool-matched 11-paper comparison. This pattern persists for three open-weight readers on n=200 queries, with LC exceeding RAG by +13 to +36 pp in EBG `Ans` accuracy on MULTIHOP-RAG and by +0.07 to +0.24 answerable-subset judge points on QASPER-MP, alongside a large residual grounding gap (Appendices D.7 and E.6).

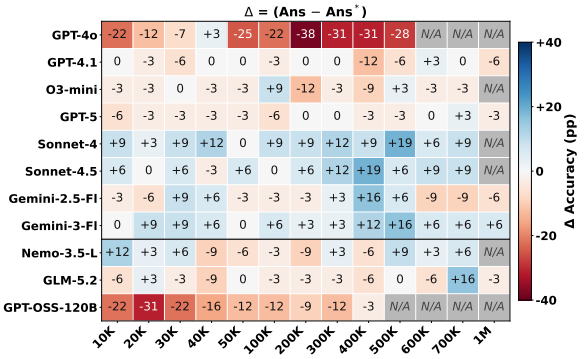

Figure 1: NoLiMa-based *LC* benchmark comparing **EBG (Ans)** and ***answer-only* (Ans*)** prompting across eight proprietary and three open-weight models over 10K–1M context lengths. Values report the **accuracy difference** (Δ), with larger positive values (↑) indicating greater EBG gains; *N/A* denotes inputs exceeding the model's context window.

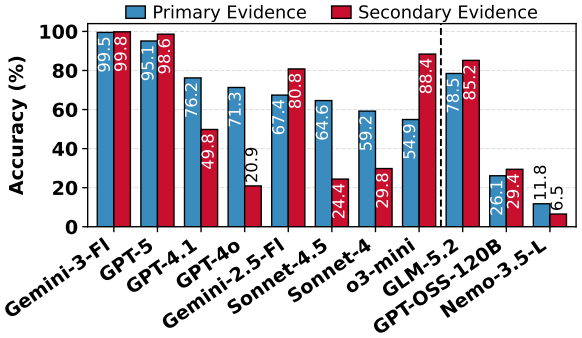

Figure 2: Two-needle citation breakdown: rate of correctly citing the **primary needle vs.** the **secondary needle** under NoLiMa-based *LC* benchmark. Higher (↑) values are better.

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 prompting conditions (*answer-only* and *EBG*). GPT-5 and Gemini-3-Flash maintain $\geq$ 98% accuracy across all tested context lengths, while Sonnet-4.5 remains within the 93–100% range up to 700K characters. Consequently, the synthetic dataset shows no measurable citation-as-scaffold effect on answer accuracy: *Ans** and *Ans* remain within sampling noise across all cells, **including at the longest contexts**. This does not refute RQ2 but reflects the controlled benchmark design: lexically dissimilar distractors make the two-hop facts relatively easy to isolate, leaving little headroom for EBG's benefits.

A different pattern emerges on the more challenging NoLiMa-based *LC* benchmark (Fig. 1), where the scaffolding effect becomes apparent **under long-context stress**. Sonnet and Gemini models gain between +9 and +19 pp from *EBG* prompting across context lengths from 300K to 900K. Among open-weight models, the smaller `Nemo-3.5-L` improves by +3 to +12 pp, while `GLM-5.2` reaches a maximum gain of +16 pp at 700K; in contrast, `GPT-OSS-120B` shows no benefit. Taken together, the contrast between the synthetic and NoLiMa settings suggests that **scaffolding is most useful when long-context retrieval is sufficiently challenging, while its effectiveness remains model-dependent and may not simply scale with model capacity**.

RQ3: Asymmetric Retrieval Behavior. As shown in Fig. 2, recovery of the two evidence types varies substantially across models. `Gemini-3-Fl` and `GPT-5` achieve near-perfect and balanced citation of both the primary and secondary needles (99.5/99.8% and 95.1/98.6%, respectively). `Gemini-2.5-Fl` also performs relatively well, but exhibits an inversion, with secondary-needle citation exceeding primary-needle citation (80.8% vs. 67.4%). A similar, more pronounced inversion appears for `o3-mini` (88.4% vs. 54.9%). These cases suggest that successful recovery of one evidence type does not necessarily imply comparable recovery of the other.

This asymmetry reverses for several other models, where primary-needle recovery substantially exceeds secondary-needle recovery. `GPT-4.1`, `GPT-4o`, and the `Sonnet` models exhibit this pattern, with the largest disparities observed for `GPT-4o` (71.3% vs. 20.9%) and `Sonnet-4.5` (64.6% vs. 21.4%). This disparity indicates that retrieving the primary evidence does not guarantee successful attribution of the complementary evidence required for the task. Moreover, primary-needle citation itself remains imperfect, showing that retrieval failures extend beyond the secondary signal alone.

The open-weight models show substantial variation in retrieval. `GLM-5.2` achieves balanced recovery (78.5% primary and 85.2% secondary), whereas `GPT-OSS-120B` records lower rates for both needles (26.1% and 29.4%). `Nemo-3.5-L` performs poorly on both (11.8% and 6.5%), indicating broader retrieval limitation rather than directional asymmetry. **Overall, multi-evidence retrieval varies across models and evidence types, reinforcing model-dependent patterns as in RQ1.**

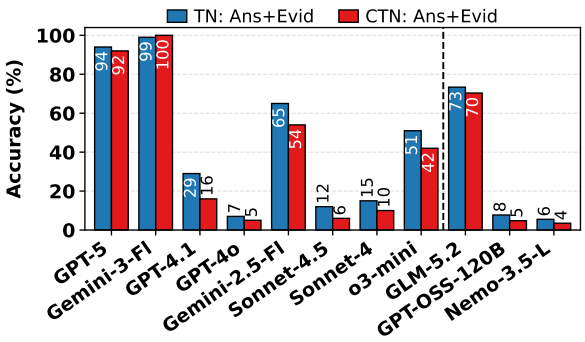

Figure 3: **TN vs. CTN**, `Ans+Evid`. TN uses a factual secondary needle; CTN uses a contradictory secondary needle, conflicting with world knowledge, requiring context-faithful reasoning and citation. Models to the right of the dashed line are open-weight. Higher (↑) values are better. `Ans` accuracy for the same runs is reported in Figure 6.

RQ4: CTN amplifies failure. Performance generally declines under contradiction, with substantial variation across models (Fig. 3); answer accuracy is reported separately in Figure 6. `GPT-5` and `Gemini-3-Fl` remain relatively context-faithful, preserving high answer accuracy, though their `Ans+Evid` still drops, indicating greater sensitivity of evidence grounding to contradiction. In contrast, `GPT-4.1`, `GPT-4o`, and the `Sonnet` variants show sharper declines, particularly in `Ans+Evid`, suggesting they can recover the expected answer while failing to reconcile contradictory evidence.

CTN robustness is also uneven among open-weight models. `GLM-5.2` is notably resilient: `Ans` decreases only from 79% to 72%, while `Ans+Evid` changes from 73% to 70%, largely preserving the small gap between the two metrics. By contrast, `GPT-OSS-120B` and `Nemo-3.5-L` start from substantially weaker TN baselines: the former falls from 28/8% (`Ans`/`Ans+Evid`) to 8/5% under CTN, and the latter from 12/6% to 5/4%. These models therefore already struggle to extract and ground answers under TN, with CTN further compounding this weakness.

Overall, `Ans+Evid` is generally more vulnerable to CTN than `Ans`, showing that **resolving contradiction while grounding the answer in supplied evidence is harder than producing the correct answer itself.** The widening `Ans`–`Ans+Evid` gap highlights a common failure mode: models reach the expected answer without faithfully using the context that should justify it, even when answers remain correct. See Appendix F.4 for more details.

6 Conclusion and Future Work

We introduced **evidence-based generation (EBG)**, a framework that requires models to produce both an answer and line-level citations, together with a joint `Ans+Evid` metric that measures correctness and grounding simultaneously. By separating retrieval from evidence-grounded reasoning, EBG exposes failures obscured by answer-only evaluation. We evaluate these failures using evidence-grounded versions of MULTIHOP-RAG and QASPER-MP, a controlled synthetic two-hop haystack, and an evidence-augmented NoLiMa spanning 10K–1M characters. **This evaluation suite spans both realistic and controlled settings, allowing retrieval, reasoning, and grounding errors to be examined under complementary conditions.**

Across ten frontier LLMs, EBG reveals two key RAG limitations: *(i)* similarity retrieval finds *any* relevant chunk more reliably than *all* evidence required for multi-fact queries, imposing a retrieval ceiling at realistic top-k budgets; and *(ii)* chunking can discard document-level meta-information needed by identity-bearing queries, causing RAG to over-refuse on retrieval misses. EBG further exposes grounding failures hidden by answer-only metrics: models often cite incomplete evidence, and the gap between answer and joint accuracy widens when in-context evidence conflicts with parametric knowledge. These findings show that correct answers alone can substantially overstate a model's ability to faithfully use long-context evidence. They also highlight evidence attribution as a useful diagnostic for distinguishing retrieval limitations from downstream reasoning failures. **Consequently, EBG provides a more informative basis for comparing RAG and long-context systems than evaluations that collapse retrieval, reasoning, and attribution into answer accuracy alone.**

These results do not imply that "LC replaces RAG"; rather, LC and RAG exhibit distinct failure modes that EBG helps localize, motivating evaluations that measure both answer accuracy and evidence fidelity. **The appropriate architecture may therefore depend not only on answer quality, but also on which sources of failure are acceptable for a given task and context regime.** Future work includes scaling EBG to the full benchmarks and testing document-aware chunking. Richer evidence attribution and the observed reasoning benefits of evidence generation also motivate investigating grounding objectives as part of model training.

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. The benchmarks also assume that the reference evidence is sufficiently complete and unambiguous, whereas real-world questions may admit multiple valid evidence sets or interpretations.

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. We therefore do not systematically explore retrieval variants such as query rewriting, hybrid search, or adaptive context selection, which could alter the observed retrieval ceilings. Finally, because we evaluate proprietary frontier and open-weight 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 cite irrelevant or incomplete evidence while producing a plausible answer. We therefore view evidence-based generation as a complementary 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 *Proceeding of the 12th International Conference on Learning Representations*, ICLR '24.

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*, NAACL '21, pages 4599–4610, Online.

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 '25, pages 23281–23298, Suzhou, China.

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*, EACL '21, pages 874–880, Online.

Greg Kamradt. 2024. Needle In A Haystack. Accessed: 2026-08-30.

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 '20, pages 6769–6781, Online.

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*, EMNLP '25, pages 32559–32569, Suzhou, China.

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 *Proceeding of the 2020 Conference on Advances in Neural Information Processing Systems*, NIPS '20.

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*, EMNLP '24, pages 881–893, Miami, Florida, USA.

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*, volume 267 of *Proceedings of Machine Learning Research*, pages 44554–44570. PMLR.

Ofir Press, Noah Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *Proceedings of the 10th International Conference on Learning Representations*, ICLR '22.

Thomas Schuster, Marian Lambert, Nico Döring, and Julius Trögele. 2025. Needle-in-the-Haystack testing LLMs with a complex reasoning task. In *Proceedings of the 26th International Conference on Engineering Applications of Neural Networks*, EANN '25, pages 254–266, Limassol, Cyprus.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 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 '22, pages 308–318, Online only.

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 *Proceedings of the 13th International Conference on Learning Representations*, ICLR '25.

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)*, ACL '24, pages 15262–15277, Bangkok, Thailand.

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 *Proceedings of the 13th International Conference on Learning Representations*, ICLR '25.

A Prompt Templates

The haystack is line-numbered (`[1]`, `[2]`, ..., `[N]`) and substituted into the `haystack` placeholder of each task template. We use four prompt variants: an *answer-only* variant (no system prompt), a *Chain-of-Thought (CoT)* variant (no system prompt), an *evidence-based generation (EBG)* variant that asks the model to cite supporting line numbers in JSON, and a *CoT-based EBG* variant (no system prompt) that requests both brief reasoning and the EBG JSON answer-and-evidence output.

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.
```

CoT 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. Be aware that some details may not be stated directly, and you may need to INFER the answer based on the given information. Begin with a brief explanation of your reasoning in NO MORE THAN THREE (3) sentences. Then, return the final answer on a new line.
Question: {question}
```

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:
```json
{"lines": [25, 412], "answer": "John"}
```
Question: {question}
```

CoT-based EBG Prompt (User Message)

```
You will answer a question based on the following book snippet:
<context>{haystack}</context>
Use the information provided in the book snippet to answer the question. Be aware that some details may not be stated directly, and you may need to INFER the answer based on the given information. Begin with a brief explanation of your reasoning in NO MORE THAN THREE (3) sentences. Then, answer the question based on information only from the context. If the question is not answerable from the context, answer NA. Provide the answer (just the character name) and all lines the answer is based on in json format. For example:
```json
{"lines": [25, 412], "answer": "John"}
```
Question: {question}
```

B Additional Details of Synthetic Dataset

We provide additional details on the construction and structure of our synthetic dataset and present a representative example to illustrate the resulting task format and evidence organization.

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*). The SN context supplies the entity-specific primary needle but not the complete inference chain required by the question. For example, a needle stating that a character is vegan must be combined with parametric commonsense knowledge that veganism excludes eating fish.

We therefore refer to the cited primary needle in SN as *anchor evidence*. For SN, evidence accuracy measures whether the model cites the correct contextual anchor, and joint accuracy measures answer correctness together with anchor citation; these metrics should not be interpreted as establishing a complete context-only evidence chain. This distinction is specific to SN because part of the information required to answer the query may reside in the model's parametric knowledge. Accordingly, SN citations cannot directly attribute the contribution of such internally stored information.

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.

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

This setting introduces secondary needles that conflict with world knowledge. For example, CTN-0402-S claims that *black coffee* contains *steamed milk*, while CTN-0405-S claims that *falafel* contains *minced lamb*. As in *TN*, each secondary needle pairs with its primary needles (e.g., CTN-0402-S with 0402 and 0402Inv), testing whether models follow provided context or override it with parametric knowledge. For both question types, the primary needle provides an entity-specific anchor while inference requires unstated bridge knowledge. One-hop questions require a commonsense bridge (e.g., veganism implies avoiding fish), while two-hop questions require an item-composition bridge (e.g., Brandade is fish-based).

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

This appendix analyzes an *oracle-containing long-context (LC)* reading condition and an *open-corpus retrieval-augmented generation (RAG)* condition on the MULTIHOP-RAG benchmark (Tang and Yang, 2024). The LC condition guarantees that every gold-evidence article is available to the reader together with 20 distractor articles, whereas the RAG condition must retrieve the required evidence from the full 609-article corpus. The purpose of this comparison is therefore to diagnose the contribution of open-corpus retrieval failures to end-to-end performance; it is not a pool-matched comparison of LC and RAG under identical candidate document sets. 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 queries tests both *EBG* (line-level evidence) and *RAG* (chunked corpus). We run experiments on a 50-query pilot set, sampled with seed 0 and stratified across three types: n_{inf}=13, n_{comp}=21, and n_{temp}=16. We use the same queries, prompts, and scoring throughout, while LC and RAG differ in candidate-document pools as described below.

LC haystacks. For each query, we assemble a mini-haystack of (i) all gold-evidence documents and (ii) 20 distractor documents sampled without replacement from the remaining corpus. Documents are randomly concatenated, 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 so EBG labels match the prompt. Mini-haystacks average ~ 241K characters (~ 58K tokens), within every model's context window. This is the *LC (full)* condition.

RAG haystacks. For each query, we construct RAG contexts at $K \in 4, 10, 25, 50, 100, 200$. The retriever searches the full 609-document corpus, divided into 256-token sliding windows (6,296 chunks). We embed chunks and queries with `text-embedding-3-large`, rank by cosine similarity, and use the top-K chunks as the RAG context. Each chunk is prefixed with `[CHUNK <rank> | <title>]`, and rendered with the same sentence-per-line numbering as LC. Gold evidence facts are re-mapped to lines in the retrieved context; unretrieved facts have no corresponding gold line.

The conditions are not pool-matched: LC has all gold source articles plus 20 distractors, whereas RAG must locate the required evidence across all 609 articles. This design deliberately preserves the native evidence-access setting of each approach in practice, enabling a realistic comparison between them under their respective retrieval assumptions.

D.1 Setup

Prompt modes. Each (query, context) pair is evaluated under two prompts differing in one sentence:

- *Answer-only (*`Ans`**).* The model returns a JSON object with only an `answer` field. We report `Ans`* accuracy.
- *EBG.* The model returns 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 accompanying public code release for full reproducibility.

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`.

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`. This setup keeps decoding conditions consistent across models and experiments.

Scoring. `Ans` matches the model's answer string against the gold answer using the same normalization used in the synthetic and NoLiMa experiments. `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). Thus, reported scores reflect successfully completed model calls only.

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

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

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); *any gold* reports the fraction of queries for which at least one annotated gold fact appears in the top-k retrieved set; *all gold* reports the fraction for which every annotated gold fact appears. The latter measures strict annotated-gold coverage, not an upper bound on answer accuracy. Under the stronger assumptions that the gold annotations are exhaustive and that every annotated fact is necessary, *all gold* upper-bounds 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.

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. This establishes a concrete retrieval ceiling on downstream RAG performance. Crucially, the bottleneck is not finding relevant evidence, but retrieving all pieces of evidence needed for a complete answer. Thus, many apparent reasoning failures downstream are already determined by missing context at retrieval time.

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.

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

For the 50 pilot queries, we repeat this analysis at every evaluated K. Table 5 reports the fraction of queries with all gold facts ("needles") retrieved in the top-K and the fraction of individual gold facts appearing anywhere in the top-K. These quantities upper-bound faithful joint accuracy and answer recall, respectively, for RAG on the pilot set.

D.5 Experiment C: Oracle-containing LC versus open-corpus RAG

The central experiment sweeps open-corpus RAG over six values of K and compares it with oracle-containing LC under both prompt modes. All conditions use the same 50 queries, prompts, models, and scoring code, but differ in candidate-document pools: LC receives every gold-evidence article plus 20 distractors, whereas RAG searches the 609-article corpus. Table 6 thus measures the end-to-end cost of open-corpus retrieval relative to gold-source access, not a pool-matched comparison. The table tells three stories, one per metric.

Answer accuracy: oracle-containing LC exceeds open-corpus RAG under the evaluated budgets. Across both prompts and all three models, oracle-containing LC outperforms open-corpus RAG for $K \leq 100$. At K=200, with 96% gold-fact recall and 90% all-needles coverage, RAG ties LC for `Sonnet-4.5` (68%) but remains 4,pp below for `GPT-5` (78% vs. 82%) and 10,pp below for `Gemini-3-Flash` (70% vs. 80%). At K=10, the LC–RAG gap is 22–44,pp under answer-only prompting and 26–34,pp under EBG. These gaps include full-corpus retrieval and thus compare open-corpus RAG with evidence-guaranteed LC, not an intrinsic LC advantage under identical candidate pools.

Model	Metric	LC (oracle)	RAG@4	RAG@10	RAG@25	RAG@50	RAG@100	RAG@200
Answer-only prompt (*Ans** / Joint $\equiv$ 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: **Oracle-containing LC versus open-corpus RAG on the 50-query MULTIHOP-RAG pilot.** LC receives every gold-evidence article plus 20 distractor articles; RAG retrieves from the full 609-article corpus. The conditions use the same queries, models, prompt templates, and scoring procedure but are not pool-matched. All values are percentages over n=50 (`Sonnet-4.5` RAG@200 EBG: n=49). Boldface marks the per-row maximum within the corresponding prompt-mode block. Higher (↑) values are better.

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 reflects two distinct requirements for evidence correctness: *all needles must be retrieved* (coverage, Table 5) and the model must *cite all of them correctly* (citation behaviour). Small K favors the latter because the citation surface is limited, whereas large K favors the former as retrieval coverage improves. Increasing K therefore trades easier citation selection for greater evidence availability, and these opposing effects roughly cancel. Consequently, small-K RAG can remain comparable to LC on strict evidence despite substantially lower retrieval coverage.

Joint (faithful) accuracy: mixed. `Joint` = `Ans` $\wedge$ `Evid` (also written as `Ans+Evid`) is the most diagnostic 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. Under the answer-only prompt, the oracle-containing LC condition exceeds open-corpus RAG@10 by 22–44 pp and exceeds RAG@200 by 0–10 pp. The same qualitative pattern holds under EBG. Because the two conditions do not share a candidate-document pool, these numbers quantify the combined cost of document discovery and passage retrieval relative to evidence-guaranteed reading; they are not a pool-matched estimate of an architecture-level LC–RAG effect.

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 principal source of the observed gap relative to the oracle-containing LC control is that, at small K, open-corpus RAG rarely retrieves the complete union of annotated gold facts.

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 Generalization to open-weight models

The preceding comparison uses three proprietary frontier LLMs. To test whether the observed pattern generalizes across model families, we repeat the same two evidence-access regimes, an oracle-containing LC condition and open-corpus RAG, with three open-weight models. To test whether the LC-EBG vs. RAG ordering is a property of the task rather than of a particular model family, we re-run the same conditions, using same mini-haystacks, same RAG contexts, same prompts and scoring code, with three open-weight models served through Fireworks.ai: `GPT-OSS-120B`, `GLM-5.2`, and `Nemotron-3.5-Lightning-30B-A3B` (or Nemo-3.5-L in short). To obtain tighter estimates we scale this comparison to a larger stratified sample of n=200 queries. We evaluate the two most informative RAG budgets, a realistic operating point (K=10) and the near-saturation budget (K=200, 90% all-needles coverage; Table 5), against the full LC condition. Table 7 reports the results.

Model	Metric	LC (oracle)	RAG@10	RAG@200
Answer-only prompt (*Ans**)				
GPT-OSS-120B	Ans	**75.0**	47.5	67.5
GLM-5.2	Ans	**88.0**	53.0	72.5
Nemo-3.5-L	Ans	**75.5**	43.5	51.0
EBG prompt: Ans / Evid (strict) / Joint				
GPT-OSS-120B	Ans	**54.0**	41.5	52.0
	Evid	3.0	**13.0**	2.0
	Joint	2.0	**9.0**	2.0
GLM-5.2	Ans	**88.5**	52.5	74.0
	Evid	**35.0**	21.5	22.0
	Joint	**30.0**	18.0	20.5
Nemo-3.5-L	Ans	**48.2**	28.5	28.5
	Evid	5.5	**9.0**	3.0
	Joint	5.5	**7.5**	2.0

Table 7: **LC-EBG vs. RAG-EBG for open-weight models** on a 200-query stratified sample. All numbers are percentages over n=200 (a handful of cells return n=198–199 where 1–2 queries exceed a provider's context limit). Boldface marks the per-row maximum within each mode block. Higher (↑) values are better. At n=200 the 95% bootstrap CI half-width on `Ans`/`Joint` is ≈6–7 pp. LC is the oracle-containing gold-article-plus-20-distractor condition; RAG@10 and RAG@200 use open-corpus retrieval over all 609 articles.

The qualitative picture from Table 6 carries over. **(i) Answer accuracy: oracle-containing LC exceeds the evaluated open-corpus RAG setup.** At the realistic K=10 operating point, LC's *Ans* exceeds RAG@10 for all three models under both prompts: by +28 to +35 pp under the answer-only prompt and by +13 (GPT-OSS-120B), +36 (GLM-5.2), and +20 pp (Nemotron) under EBG - the same +26 to +34 pp EBG gap we observe for the frontier models. At the near-saturation budget K=200 RAG closes some of the gap but overtakes LC for none of the three readers: LC *Ans* still leads by +2 (GPT-OSS-120B), +14.5 (GLM-5.2), and +19.7 pp (Nemotron), with the GPT-OSS-120B margin inside the bootstrap CI. The larger sample thus removes the one n=50 anomaly (GPT-OSS-120B RAG@200 nominally above LC) and confirms LC ≥ RAG@200 throughout. **(ii) The grounding gap is model-agnostic.** Under LC, the *Ans*−*Joint* gap is 52 (GPT-OSS-120B), 58.5 (GLM-5.2), and 42.7 pp (Nemotron) - the same 40–60 pp range as the frontier models. GPT-OSS-120B is the extreme case: it answers 54% of queries correctly but achieves only 2% joint accuracy, essentially never citing both required hops. **(iii) Citation-as-scaffold is uneven for weaker readers.** The *Ans** →*Ans* (EBG) change on LC is +0.5 pp for GLM-5.2 but −21 pp for GPT-OSS-120B and −27 pp for Nemotron: the weakest open readers, unlike the frontier models, struggle to answer and cite simultaneously, so the citation requirement can hurt rather than help. Overall, the pattern that oracle-containing LC achieves higher answer accuracy than the evaluated open-corpus RAG pipeline at practical retrieval budgets, while retaining a substantial grounding gap, generalizes across proprietary and open-weight readers. This result continues to include the cost of full-corpus retrieval and is not a pool-matched comparison.

D.8 Limitations and next steps

The frontier-model experiments use a 50-query pilot, so the reported estimates remain subject to sampling variation, particularly for small question-type slices such as the 13 inference queries. Evaluating all 2,255 usable queries would provide tighter confidence intervals. We also hold the retriever fixed to `text-embedding-3-large` with 256-token sliding windows. Reranking, hybrid sparse–dense retrieval,

query expansion, longer chunks, and document-aware indexing could improve the open-corpus RAG results. For the evaluated retriever, corpus-wide all-gold Hits@10 is 19% over the 2,255 usable queries; on the 50-query pilot, all-gold coverage at K=10 is 28%. Under the assumptions that the gold annotations are exhaustive and that every annotated fact is necessary, these coverage rates strongly constrain faithfully grounded performance at this retrieval budget.

A key design limitation is that the MULTIHOP-RAG LC and RAG conditions are not pool-matched. The LC condition is an *oracle-containing reading control*: every gold-evidence article is guaranteed to be present together with 20 distractor articles. The RAG condition instead performs *open-corpus retrieval* over the full 609-article corpus. Consequently, the reported LC–RAG gaps combine the effects of full-corpus document discovery, passage retrieval, and downstream reading; they should not be interpreted as an intrinsic ranking of LC and RAG under matched evidence access.

A complementary pool-matched RAG condition that retrieves from the same per-query gold-plus-20 candidate set would separate within-pool chunking and ranking effects from the cost of full-corpus document discovery. We leave this controlled comparison, together with evaluation of stronger retrieval pipelines and the full query set, to future work.

D.9 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 (pool-matched candidate corpus). The reader sees a flat list of retrieved paragraphs with a one-line `[CHUNK <rank> | <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 8 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.

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 8: **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.

The aggregate hides an answerability asymmetry. Splitting the 50 queries into the 38 answerable and 12 unanswerable subsets tells a sharper story (Table 9).

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 8 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 ∼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*

Model	Condition	Answerable ($n = 38$)			Unanswerable ($n = 12$)		
		Judge	Ans-F1	Refuse %	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 9: **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.

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 +12pp 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 structured RAG produced a counterintuitive result: `struct_pool11@3` *underperformed* `flat_pool11@3` on Evidence-F1 across models (gaps of 0.07–0.12), despite carrying *more* information than the flat baseline (abstract per paper). On inspection, ∼44% of structured-RAG citations pointed at `<title>` or `<abstract>` lines: when an abstract paraphrases the answer, the model cites it - but QASPER gold evidence excludes abstracts, so abstract citations *cannot* match gold.

We therefore made two changes to the structured layout and prompt used 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 `...` line. This gives the reader explicit signal that some material is missing between the visible paragraphs.

Under this revised protocol (Table 8), 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 five 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 paraphrases, we additionally score outputs 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 Generalization to open-weight models

We repeat the QASPER-MP comparison with three open-weight models served through Fireworks.ai - `GPT-OSS-120B`, `GLM-5.2`, and `Nemotron-3.5-Lightning-30B-A3B` - using the same retrieval indices, EBG prompt, and GPT-5 judge (Appendix E.5), scaled to a larger 200-query stratified sample (n=150 answerable, n=50 unanswerable) for tighter estimates. Table 10 splits the results by answerability, exactly as Table 9 does for the frontier models.

The answerability asymmetry we documented for frontier models reproduces on every open-weight reader. **LC EBG wins the answerable subset cleanly**: the LC judge score is highest for all three models, beating the best RAG variant by $+0.16$ (GPT-OSS-120B), $+0.24$ (GLM-5.2), and $+0.07$ (Nemotron) - gaps as large as, or larger than, the $+0.05$ to $+0.13$ we observe for the frontier models. **RAG over-refuses for the same mechanistic reason**: `flat_pool11@3` refuses 21–33% of answerable questions (vs. 7–12% for LC), because a missed retrieval leaves the reader with only distractor content; those refusals then mechanically boost RAG on the unanswerable subset, where "Unanswerable" is gold. **Realistic full-corpus retrieval widens the LC advantage**: `flat_full@10` answerable judge scores drop to 0.43–0.50, trailing LC by 0.09 to 0.24 judge points, as the retriever must find the gold paper among ∼280 distractors from a 2–3 word cue. The open-weight results thus corroborate the central QASPER-MP finding: once conditioned on answerability, long-context EBG beats every RAG variant, and the aggregate "tie" is an artifact of RAG's refusal behavior.

E.7 Strengthening RAG: metadata-in-chunk retrieval

The full-corpus RAG deficit above is partly a *retrieval* failure rather than solely a limitation of the downstream reader. After v2 disambiguation, each query carries only a short, typically 2–3 word cue identifying the relevant paper. In the baseline index, however, candidate chunks are embedded as raw paragraph text, without any explicit representation of document identity. The retriever must therefore infer the connection between a sparse paper-level cue in the query and a paragraph whose local content may never mention the paper title or otherwise reveal its source. This mismatch is especially consequential in the full-corpus setting, where evidence from the gold paper competes with paragraphs drawn from the entire ∼281-paper dev corpus. Even when the query contains enough information to identify the intended document, the embedding space has no direct document-level feature with which to exploit that signal, making it easy for the gold paper - and consequently the relevant evidence within it - to fall outside the retrieval budget.

A natural way to give RAG a fairer chance is therefore to expose document identity directly to the retriever by injecting metadata into the indexed chunks. We rebuild the retrieval index so that every paragraph is embedded together with its source paper's title, using the representation "`title\n\nparagraph`". This modification creates an explicit lexical and semantic bridge between the short paper cue supplied by v2 disambiguation and every paragraph originating from that paper, while preserving the paragraph text needed to rank evidence within the document. We then re-run the hardest retrieval condition, `flat_full@10`, at n=200 for all five readers. Importantly, this intervention is confined to retrieval: generation-side rendering, prompting, context construction, and reader settings are unchanged. Thus, any downstream improvement can be attributed to changing the representation used for ranking rather than to providing the reader with additional metadata or altering the generation procedure.

Metadata-in-chunk produces a large improvement in evidence retrieval. On the 200-query set, the fraction of queries for which the *full* gold-evidence set is recovered within the top-10 increases from 43.5% to 61.0%, a gain of 17.5 percentage points. Gold-paragraph recall rises even more substantially in relative terms, from 30.6% to 50.3% (+19.7 pp). These gains indicate that the original full-corpus deficit was not simply caused by an insufficient top-k budget: a substantial portion arose because the retriever lacked a representation of the very document-level signal introduced during query disambiguation. Prepending the title

Model	Condition	Answerable ($n = 150$) Judge	Ans-F1	Refuse %	Unanswerable ($n = 50$) Judge	Ans-F1	Refuse %
GPT-OSS-120B	**LC**	**0.701**	0.467	8.7	0.500	0.475	42.0
GPT-OSS-120B	flat_pool11@3	0.497	0.388	32.7	0.660	0.666	64.0
GPT-OSS-120B	struct_pool11@3	0.537	0.418	24.7	0.640	0.646	62.0
GPT-OSS-120B	flat_full@10	0.497	0.369	28.0	0.670	0.661	64.0
GLM-5.2	**LC**	**0.735**	0.553	6.7	0.500	0.457	36.0
GLM-5.2	flat_pool11@3	0.477	0.364	33.3	0.680	0.686	66.0
GLM-5.2	struct_pool11@3	0.477	0.360	33.3	0.660	0.665	64.0
GLM-5.2	flat_full@10[†]	0.497	0.385	30.4	0.673	0.668	65.3
Nemotron-3.5-30B	**LC**	**0.523**	0.375	12.0	0.530	0.513	42.0
Nemotron-3.5-30B	flat_pool11@3	0.397	0.316	21.3	0.330	0.326	32.0
Nemotron-3.5-30B	struct_pool11@3	0.450	0.349	20.7	0.390	0.387	38.0
Nemotron-3.5-30B	flat_full@10	0.433	0.329	22.7	0.420	0.406	40.0

Table 10: **QASPER-MP open-weight results split by answerability** (EBG mode, GPT-5 judge), over a 200-query stratified sample. Answerable subset is n=150 and unanswerable n=50; a handful of cells return 1–2 fewer queries where a provider error dropped a row ([†]`GLM-5.2` `flat_full@10` returned n=148/49). Best LC vs. RAG variant per model shown in **bold** within the answerable subset. "Refuse" is the fraction of outputs starting with `Unanswerable`. Higher (↑) values are better.

allows that signal to influence ranking directly, increasing both the probability that the correct paper is represented among the retrieved chunks and the amount of gold evidence that reaches the reader.

The retrieval gains propagate consistently to answer quality (Table 11). On the answerable subset, the judge score improves for *every* model. The four strongest readers - GPT-5, Sonnet-4.5, GPT-OSS-120B, and GLM-5.2 - gain between +0.10 and +0.13, while Nemotron improves by +0.03. The answerable-question refusal rate also falls by 5–8,pp, consistent with fewer queries reaching the reader with distractor or insufficient context. That the same retrieval-only intervention benefits readers with different capabilities strengthens the interpretation that evidence availability is a shared bottleneck across otherwise diverse reader architectures and levels of baseline answer quality. Part of the reader-level degradation under full-corpus RAG is therefore upstream: when document identity is represented during retrieval, more evidence survives the top-10 bottleneck, giving all five readers a better opportunity to answer. This shows that chunk representation can materially affect end-to-end RAG performance over a large, heterogeneous corpus.

Crucially, **a stronger retriever narrows but does not close the LC gap**. With metadata-in-chunk, `flat_full@10` trails LC on the answerable subset for all models - by 0.04 (Sonnet-4.5), 0.06 (Nemotron), 0.08 (GPT-OSS-120B), 0.14 (GLM-5.2), and 0.15 (GPT-5) judge points. Roughly half the full-corpus LC advantage is recoverable by fixing retrieval, but a residual remains: even when the right paragraphs are retrieved, removing document context costs answer quality relative to reading the full neighborhood. This points beyond evidence discovery: retrieval preserves passages but discards surrounding structure. That context supplies definitions, antecedents, methodological details, and cross-paragraph relationships needed to interpret evidence. The central finding remains robust to better RAG - LC EBG still wins.

Model	LC	full@10 plain Judge	Ref%	full@10 meta Judge	Ref%
GPT-5	0.757	0.473	27	**0.607**	20
Sonnet-4.5	0.623	0.490	29	**0.587**	24
GPT-OSS-120B	0.701	0.497	28	**0.623**	19
GLM-5.2	0.735	0.497	30	**0.597**	23
Nemotron-3.5-30B	0.523	0.433	23	**0.467**	17

Table 11: **Metadata-in-chunk retrieval on `flat_full@10`** (QASPER-MP, EBG, GPT-5 judge, answerable subset, n=200). "meta" prepends the paper title to each chunk before embedding. Judge improves over plain retrieval for every model (**bold**); "Ref%" is the answerable-question refusal rate. LC (full document neighborhood) is shown for reference. The 95% bootstrap CI half-width on the judge scores is ≈0.07.

E.8 Summary

QASPER-MP corroborates the picture in Appendix D: under the pool-matched `pool11` condition, long-context EBG and RAG end up at comparable aggregate scores, but the mechanism is asymmetric. RAG's higher refusal rate inflates its score

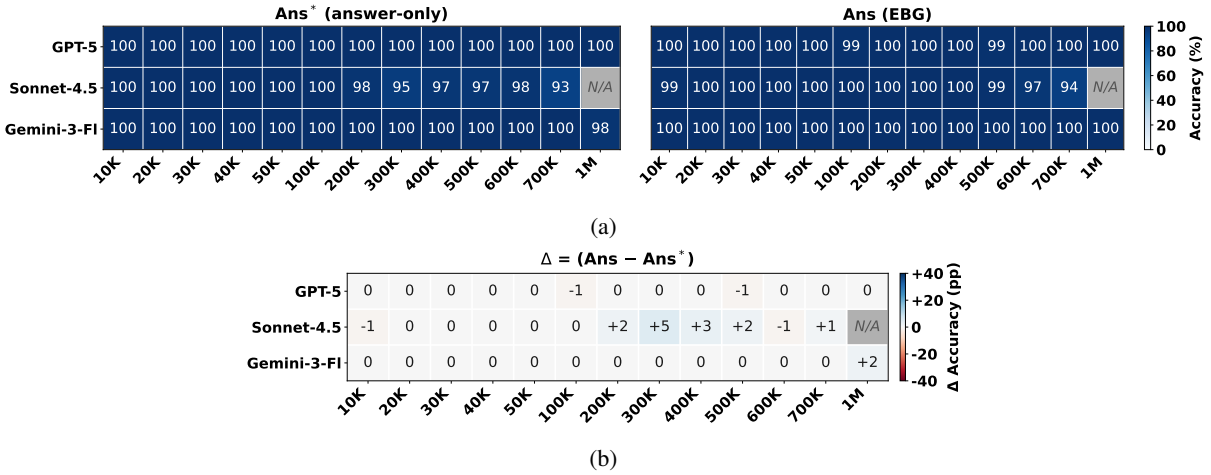

Figure 4: **Synthetic two-hop benchmark comparing `Ans`* (answer-only prompting) and `Ans` (under *EBG* prompting)** for three frontier models across 10K–1M characters. (a) Absolute `Ans`* and `Ans` accuracies. Higher (↑) values are better. (b) Accuracy difference between the two prompting settings. 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.

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

F Long-context EBG: Synthetic and NoLiMa-based Datasets

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

F.1 Ans-only vs EBG on Synthetic Dataset

Observations. Fig. 4 compares evidence-based generation (**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 12, *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

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	100.00	75.00	**-25.00**	100.00	87.50	**-12.50**	93.75	81.25	**-12.50**	81.25	87.50	**+6.25**	100.00	68.75	**-31.25**	100.00	87.50	**-12.50**
GPT-4.1	100.00	100.00	0.00	100.00	100.00	0.00	100.00	93.75	**-6.25**	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00
O3-mini	87.50	81.25	**-6.25**	87.50	81.25	**-6.25**	87.50	87.50	0.00	87.50	87.50	0.00	87.50	81.25	**-6.25**	68.75	75.00	**+6.25**
GPT-5	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00
Sonnet-4	100.00	100.00	0.00	100.00	100.00	0.00	93.75	100.00	**+6.25**	100.00	100.00	0.00	100.00	100.00	0.00	100.00	93.75	**-6.25**
Sonnet-4.5	100.00	100.00	0.00	100.00	100.00	0.00	87.50	100.00	**+12.50**	100.00	100.00	0.00	93.75	100.00	**+6.25**	100.00	93.75	**-6.25**
Gemini-2.5-Fl	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00
Gemini-3-Fl	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00
Nemo-3.5-L	75.00	93.75	**+18.75**	56.25	56.25	0.00	62.50	62.50	0.00	62.50	50.00	**-12.50**	56.25	50.00	**-6.25**	56.25	43.75	**-12.50**
GLM-5.2	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00	100.00	93.75	**-6.25**	100.00	100.00	0.00	87.50	100.00	**+12.50**
GPT-OSS-120B	75.00	75.00	0.00	75.00	37.50	**-37.50**	81.25	68.75	**-12.50**	62.50	62.50	0.00	75.00	68.75	**-6.25**	93.75	81.25	**-12.50**
Two-hop																		
GPT-4o	87.50	68.75	**-18.75**	93.75	81.25	**-12.50**	87.50	87.50	0.00	75.00	75.00	0.00	87.50	68.75	**-18.75**	87.50	56.25	**-31.25**
GPT-4.1	100.00	100.00	0.00	100.00	93.75	**-6.25**	100.00	93.75	**-6.25**	100.00	100.00	0.00	100.00	100.00	0.00	100.00	93.75	**-6.25**
O3-mini	68.75	68.75	0.00	68.75	68.75	0.00	62.50	62.50	0.00	62.50	56.25	**-6.25**	68.75	68.75	0.00	37.50	50.00	**+12.50**
GPT-5	100.00	87.50	**-12.50**	100.00	93.75	**-6.25**	100.00	93.75	**-6.25**	100.00	93.75	**-6.25**	100.00	93.75	**-6.25**	100.00	87.50	**-12.50**
Sonnet-4	68.75	87.50	**+18.75**	81.25	87.50	**+6.25**	68.75	81.25	**+12.50**	68.75	93.75	**+25.00**	81.25	81.25	0.00	62.50	87.50	**+25.00**
Sonnet-4.5	87.50	100.00	**+12.50**	100.00	100.00	0.00	93.75	93.75	0.00	100.00	93.75	**-6.25**	93.75	100.00	**+6.25**	81.25	87.50	**+6.25**
Gemini-2.5-Fl	100.00	93.75	**-6.25**	93.75	81.25	**-12.50**	68.75	87.50	**+18.75**	62.50	75.00	**+12.50**	68.75	62.50	**-6.25**	56.25	50.00	**-6.25**
Gemini-3-Fl	68.75	68.75	0.00	56.25	75.00	**+18.75**	62.50	81.25	**+18.75**	81.25	93.75	**+12.50**	93.75	93.75	0.00	81.25	93.75	**+12.50**
Nemo-3.5-L	37.50	43.75	**+6.25**	25.00	31.25	**+6.25**	18.75	31.25	**+12.50**	25.00	18.75	**-6.25**	25.00	18.75	**-6.25**	12.50	18.75	**+6.25**
GLM-5.2	62.50	50.00	**-12.50**	75.00	81.25	**+6.25**	68.75	62.50	**-6.25**	62.50	50.00	**-12.50**	50.00	50.00	0.00	50.00	31.25	**-18.75**
GPT-OSS-120B	50.00	6.25	**-43.75**	43.75	18.75	**-25.00**	31.25	0.00	**-31.25**	37.50	6.25	**-31.25**	31.25	12.50	**-18.75**	25.00	12.50	**-12.50**

(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	93.75	62.50	**-31.25**	87.50	68.75	**-18.75**	87.50	56.25	**-31.25**	93.75	62.50	**-31.25**	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
GPT-4.1	100.00	100.00	0.00	100.00	100.00	0.00	100.00	87.50	**-12.50**	100.00	93.75	**-6.25**	87.50	93.75	**+6.25**	93.75	93.75	0.00
O3-mini	87.50	75.00	**-12.50**	56.25	43.75	**-12.50**	62.50	37.50	**-25.00**	43.75	31.25	**-12.50**	37.50	25.00	**-12.50**	31.25	18.75	**-12.50**
GPT-5	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00
Sonnet-4	100.00	93.75	**-6.25**	87.50	93.75	**+6.25**	81.25	93.75	**+12.50**	68.75	87.50	**+18.75**	87.50	87.50	0.00	75.00	81.25	**+6.25**
Sonnet-4.5	100.00	100.00	0.00	81.25	87.50	**+6.25**	75.00	87.50	**+12.50**	75.00	75.00	0.00	75.00	87.50	**+12.50**	56.25	75.00	**+18.75**
Gemini-2.5-Fl	100.00	100.00	0.00	100.00	100.00	0.00	93.75	100.00	**+6.25**	87.50	100.00	**+12.50**	100.00	87.50	**-12.50**	100.00	87.50	**-12.50**
Gemini-3-Fl	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00
Nemo-3.5-L	56.25	56.25	0.00	50.00	50.00	0.00	25.00	12.50	**-12.50**	25.00	43.75	**+18.75**	31.25	37.50	**+6.25**	12.50	25.00	**+12.50**
GLM-5.2	87.50	93.75	**+6.25**	87.50	87.50	0.00	81.25	75.00	**-6.25**	87.50	81.25	**-6.25**	68.75	62.50	**-6.25**	75.00	93.75	**+18.75**
GPT-OSS-120B	75.00	68.75	**-6.25**	81.25	75.00	**-6.25**	56.25	50.00	**-6.25**	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
Two-hop																		
GPT-4o	87.50	43.75	**-43.75**	75.00	31.25	**-43.75**	75.00	43.75	**-31.25**	87.50	62.50	**-25.00**	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
GPT-4.1	93.75	100.00	**+6.25**	100.00	100.00	0.00	100.00	87.50	**-12.50**	87.50	81.25	**-6.25**	81.25	81.25	0.00	87.50	87.50	0.00
O3-mini	68.75	56.25	**-12.50**	31.25	37.50	**+6.25**	37.50	43.75	**+6.25**	18.75	37.50	**+18.75**	18.75	25.00	**+6.25**	12.50	18.75	**+6.25**
GPT-5	100.00	100.00	0.00	100.00	100.00	0.00	100.00	93.75	**-6.25**	100.00	93.75	**-6.25**	93.75	93.75	0.00	93.75	100.00	**+6.25**
Sonnet-4	43.75	68.75	**+25.00**	43.75	62.50	**+18.75**	50.00	56.25	**+6.25**	37.50	56.25	**+18.75**	68.75	81.25	**+12.50**	37.50	50.00	**+12.50**
Sonnet-4.5	50.00	62.50	**+12.50**	43.75	62.50	**+18.75**	50.00	75.00	**+25.00**	37.50	50.00	**+12.50**	68.75	75.00	**+6.25**	50.00	50.00	0.00
Gemini-2.5-Fl	62.50	56.25	**-6.25**	50.00	56.25	**+6.25**	50.00	75.00	**+25.00**	37.50	37.50	0.00	31.25	25.00	**-6.25**	31.25	25.00	**-6.25**
Gemini-3-Fl	81.25	87.50	**+6.25**	87.50	93.75	**+6.25**	62.50	87.50	**+25.00**	68.75	100.00	**+31.25**	81.25	93.75	**+12.50**	75.00	87.50	**+12.50**
Nemo-3.5-L	37.50	18.75	**-18.75**	0.00	6.25	**+6.25**	6.25	6.25	0.00	6.25	6.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GLM-5.2	68.75	56.25	**-12.50**	56.25	50.00	**-6.25**	56.25	50.00	**-6.25**	43.75	50.00	**+6.25**	50.00	43.75	**-6.25**	43.75	56.25	**+12.50**
GPT-OSS-120B	37.50	25.00	**-12.50**	43.75	25.00	**-18.75**	18.75	18.75	0.00	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*

(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	100.00	81.25	**-18.75**	87.50	75.00	**-12.50**	87.50	81.25	**-6.25**
O3-mini	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
GPT-5	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00
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	87.50	93.75	**+6.25**	93.75	75.00	**-18.75**	100.00	93.75	**-6.25**
Gemini-3-Fl	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00
Nemo-3.5-L	12.50	12.50	0.00	25.00	25.00	0.00	*N/A*	*N/A*	*N/A*
GLM-5.2	81.25	81.25	0.00	87.50	81.25	**-6.25**	50.00	43.75	**-6.25**
GPT-OSS-120B	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
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	93.75	81.25	**-12.50**	81.25	62.50	**-18.75**	75.00	68.75	**-6.25**
O3-mini	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
GPT-5	100.00	87.50	**-12.50**	87.50	81.25	**-6.25**	93.75	87.50	**-6.25**
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	43.75	37.50	**-6.25**	37.50	18.75	**-18.75**	37.50	31.25	**-6.25**
Gemini-3-Fl	56.25	81.25	**+25.00**	75.00	93.75	**+18.75**	81.25	93.75	**+12.50**
Nemo-3.5-L	0.00	0.00	0.00	12.50	12.50	0.00	*N/A*	*N/A*	*N/A*
GLM-5.2	43.75	62.50	**+18.75**	43.75	31.25	**-12.50**	25.00	25.00	0.00
GPT-OSS-120B	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*

(c) Performance from 800K to 1M.

Table 12: Performance of models across different context lengths, for one-hop and two-hop questions, averaged across $n = 16$ (4 needles $\times$ 4 depths) cases, **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. Here we present scores correct upto two decimal places.

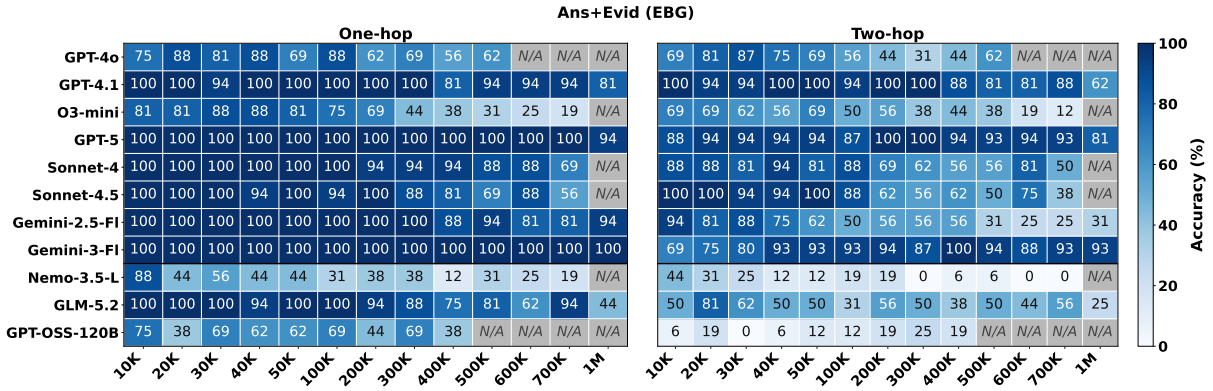

Figure 5: Performance of different models on **Ans+Evid (EBG)** accuracy for **one-hop and two-hop** questions across varying context lengths, averaged over needle depths. Cells marked with "N/A" indicate no available results for that model-context combination. Higher (↑) values are better).

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 almost 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, the reported joint `Ans+Evid` metric measures answer correctness together with citation of the primary contextual anchor, rather than a complete context-only reasoning chain, because for one-hop, part of the inference relies on *bridge knowledge* not stated in the haystack (such as *veganism* implying avoidance of *fish* or *lactose* intolerance implying avoidance of *milk*. Here, we observe that most frontier models perform very strongly at short and medium context lengths. `GPT-5`, `Sonnet-4`, `Sonnet-4.5`, `Gemini-2.5-Fl`, 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 characters. `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 characters, while `o3-mini` degrades sharply beyond 200K characters, 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 the strongest model in the two-hop setting. It maintains high accuracy across the context range and achieves 93% at 1M characters. `GPT-5` is strong from 10K to 700K, but drops to 81% at 1M. `GPT-4.1` performs well through mid-range contexts but declines at 1M, reaching 62%.

F.4 TN vs. CTN on NoLiMa-based Dataset: 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 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**).

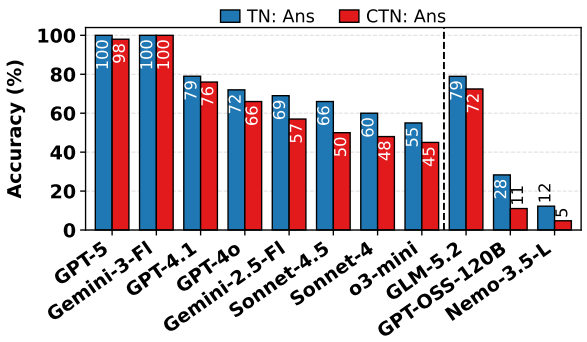

Figure 6: **TN vs. CTN** `Ans` accuracy under `EBG prompting`. We report an accuracy for the same runs whose `Ans+Evid` accuracy appears in Figure 3. Higher (↑) values indicate better performance.

- **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.
- **The answer dimension degrades too, but less than the joint metric.** Figure 6 reports `Ans` accuracy for the same runs. Every model except `Gemini-3-Fl` loses answer accuracy under contradiction, with the largest drops for `GPT-OSS-120B` (≈**-17 pp**), `Sonnet-4.5` (≈**-16 pp**), and `Gemini-2.5-Fl` and `Sonnet-4` (≈**-12 pp** each).

F.5 EBG vs. CoT Prompting on NoLiMa-based Dataset

Experiment. We compare answer accuracy under standard EBG and CoT prompting separately for one-hop and two-hop questions. We define $\Delta = \texttt{Ans} - \texttt{Ans}^c$, where `Ans` denotes answer accuracy under EBG prompting and $\texttt{Ans}^c$ denotes answer accuracy under CoT prompting. Positive values of Δ therefore indicate an advantage for EBG, whereas negative values indicate an advantage for CoT.

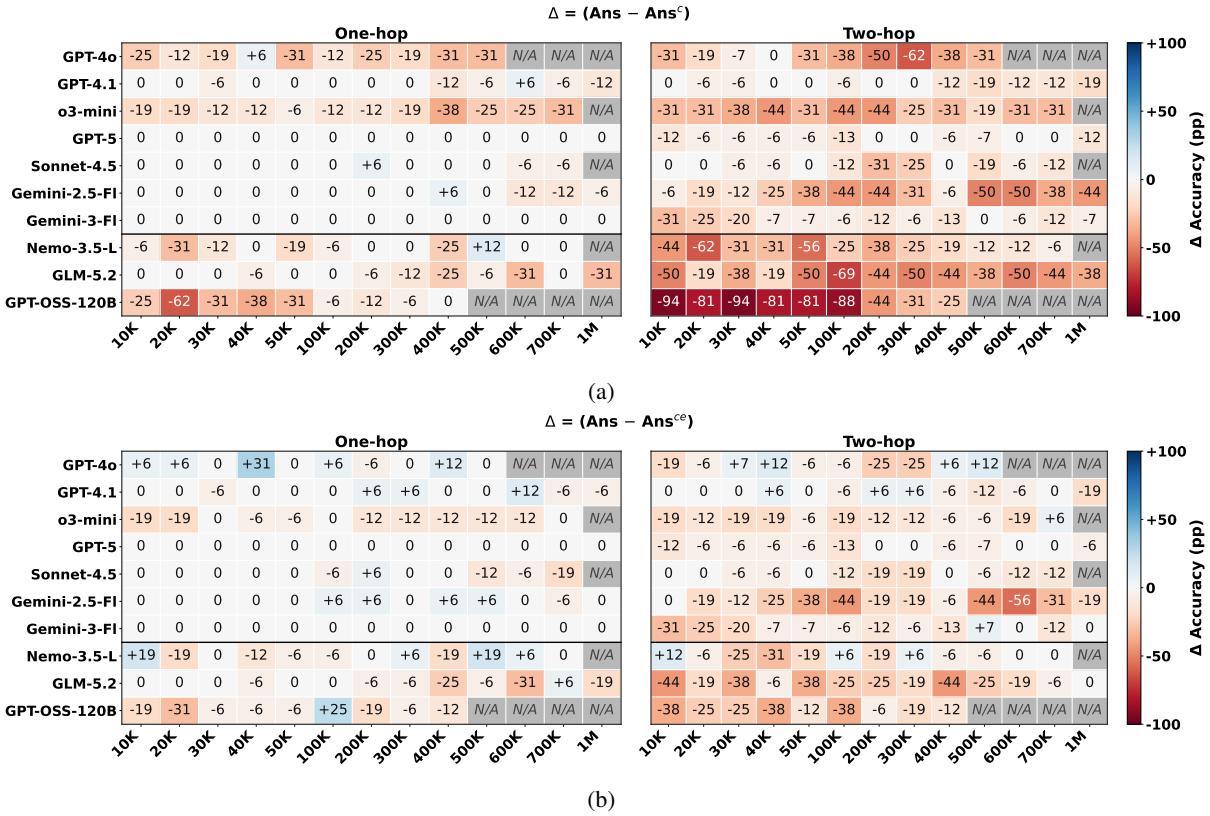

Figure 7: **Comparison of answer-accuracy differences between EBG vs. CoT and EBG vs. CoT-EBG prompting** on the NoLiMa-based dataset, with results reported separately for one-hop and two-hop questions. **(a) EBG vs. CoT**, where each cell reports $\Delta = \texttt{Ans} - \texttt{Ans}^{c}$ in percentage points; positive values favor EBG and negative values favor CoT. **(b) EBG vs. CoT-EBG,** where each cell reports $\Delta = \texttt{Ans} - \texttt{Ans}^{ce}$; positive values favor EBG and negative values favor CoT-EBG. Cells marked *N/A* indicate that no results are available for the corresponding model–context combination.

Observations. As shown in Fig. 7a, the effect of CoT is substantially larger for two-hop questions than for one-hop questions. On a model-macro average, CoT improves answer accuracy by approximately 8.5 pp for one-hop questions and 27.2 pp for two-hop questions. At the model level, CoT never reduces aggregate one-hop accuracy: it improves accuracy for eight models and matches EBG for `GPT-5` and `Gemini-3-Fl`. The largest one-hop gains occur for `GPT-OSS-120B` (+23.6 pp), `GPT-4o` (+20.0 pp), and `o3-mini` (+19.3 pp).

For two-hop questions, CoT achieves higher aggregate answer accuracy for every evaluated model. The improvements range from 5.9 pp for `GPT-5` to 68.8 pp for `GPT-OSS-120B`, with large gains also observed for `GLM-5.2` (+42.3 pp), `o3-mini` (+33.3 pp), `Gemini-2.5-Fl` (+31.3 pp), and `GPT-4o` (+30.8 pp). This larger two-hop effect suggests that explicitly eliciting a short reasoning step is particularly useful when answering requires an additional inference beyond direct needle retrieval. Because vanilla CoT also removes the citation and JSON-output requirements, however, this comparison should be interpreted as a comparison between complete prompting strategies rather than as an isolated estimate of the effect of reasoning alone.

F.6 EBG vs. CoT-EBG Prompting on NoLiMa-based Dataset

Setup. To examine whether explicit reasoning remains beneficial when evidence citation is retained, we compare standard EBG with CoT-EBG. We report `Ans` so that this comparison is directly commensurable with the vanilla-CoT comparison in Appendix F.5, which can only be scored on answer accuracy because the plain CoT prompt requests no citations. Holding the metric fixed across the two experiments isolates the effect of the citation requirement itself. We define $\Delta = \texttt{Ans} - \texttt{Ans}^{ce}$, where ce denotes CoT-EBG; negative values therefore favor CoT-EBG.

Observations. Fig. 7b shows a smaller and more model-dependent effect for one-hop questions. CoT-EBG improves aggregate answer accuracy for five models, matches EBG for two, and lowers it for three, with a model-macro average of 2.2 pp in favor of CoT-EBG. This mixed result is consistent with the near-ceiling performance of several models on direct one-hop retrieval, where an additional reasoning instruction has limited room to help and may introduce unnecessary output complexity. The two-hop result is more consistent: CoT-EBG improves aggregate `Ans` accuracy for every evaluated model, by 12.3 pp on a model-macro average. The gains range from 2.4 pp for `GPT-4.1` to 25.5 pp for `Gemini-2.5-Fl`. Substantial improvements are also observed for `GPT-OSS-120B` and `GLM-5.2` (+23.6 pp each), `o3-mini` (+12.0 pp), and `Gemini-3-Fl` (+10.4 pp). Thus, explicitly eliciting a brief reasoning step remains beneficial when the evidence-citation requirement and output schema are retained, particularly for multi-hop questions.

Taken together, the two experiments suggest that the gains from structured prompting are not attributable to citation formatting alone. **Explicit reasoning improves answer accuracy both under vanilla CoT and when the EBG citation requirement is retained, with the largest and most consistent effects appearing in the two-hop setting**.

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 13 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,

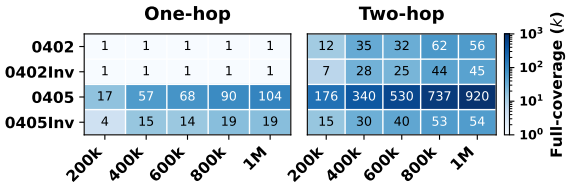

Figure 8: Minimum **full-coverage retrieval depth** (k) (for book B_1^c under NoLiMa). Lower (↓) values are better.

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 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,

	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 13: **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. Lower (↓) values are better.

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.

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 8). 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."*

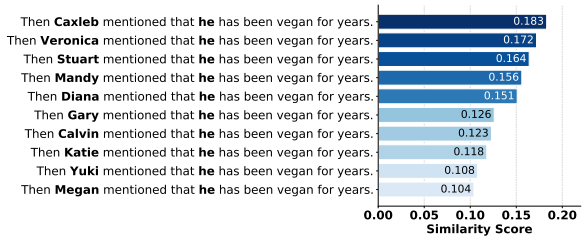

Figure 9: **Ranking of the `0405` needle instantiated with 10 different character names** based on cosine similarity to a one-hop test query. Although the sentence semantics remain unchanged, similarity scores vary across character tokens. Higher (↑) values are better.

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 9 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:

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

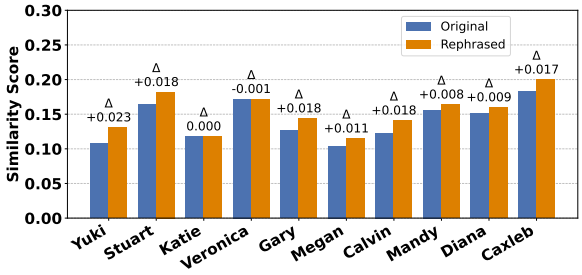

Figure 10: Cosine similarity **comparison between the original `0405` needle and the rephrased `0405-R` variant across 10 characters**. Removing the pronoun consistently increases similarity scores across different characters. Higher positive difference (↑) indicates greater effectiveness of the rephrased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

We recompute similarity scores for both the original and rephrased needles across multiple character substitutions. Figure 10 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 Retrieval Analysis under Embedding, BM25, and Reranker-RAG Pipelines

Motivated by the results and observations in Table 13, we further investigate whether the observed retrieval bottleneck is specific to embedding-based semantic retrieval or persists across alternative retrieval and ranking strategies.

Setup of Embedding-RAG. We first consider the embedding-based RAG pipeline from the preceding experiment (§ 5.1.3), restricting the evaluation to Book 1 (B_1^c). The pipeline follows the same retrieval setup as before: the haystack is divided into chunks, represented using dense vector embeddings, and ranked by semantic similarity to the query. We retain the original chunking, retrieval, and full-coverage evaluation protocol, and report the minimum top-k required to retrieve all evidence for one-hop and two-hop queries, together with the corresponding retrieved character budget. Thus, the Embedding-RAG results in Table 14 serve as the semantic-retrieval baseline against which Okapi BM25 and Qwen3-Reranker-8B are compared.

Setup of BM25-RAG. To examine whether the retrieval limitations observed with embedding-based RAG are specific to semantic similarity search, we replace the vector embedding-based retriever with Okapi BM25[2], a lexical retrieval method based on term matching and term-frequency statistics. We keep the evaluation protocol unchanged and measure the minimum retrieval depth required to achieve full evidence coverage for both one-hop and two-hop queries across increasing NoLiMa context lengths. As in our primary analysis, full coverage is defined as the smallest top-k for which all required evidence facts are present in the retrieved set for every sample. We report both this retrieval depth and its corresponding character budget in Table 14.

Setup of Reranker-based RAG. We further evaluate whether a stronger learned ranking stage can reduce the retrieval depth required for complete evidence coverage. Specifically, we use `Qwen3-Reranker-8B`, a fine-tuned reranking model, to reorder candidate passages (retrieved by the embedding-based RAG retriever) according to their relevance to the query. We then evaluate the resulting ranking using the same full-coverage criterion as the BM25-RAG and Embedding-RAG settings. This experiment tests the extent to which a learned reranker can move required evidence toward the top of the ranking and consequently reduce the context passed to the downstream model.

Observations. We observe that BM25-RAG consistently requires substantially larger retrieval budgets than Embedding-RAG, with the difference becoming more pronounced as context length increases. At 1M characters, full coverage of one-hop queries requires 307 vectors (102,475 characters) with BM25-RAG, compared with 104 vectors (23,589 characters) for Embedding-RAG. The same pattern holds for two-hop queries, where BM25-RAG requires 1,061 vectors (265,785 characters), compared with 920 vectors (217,152 char-

[2]https://pypi.org/project/rank-bm25/

	Context Length																		
	10K	**20K**	**30K**	**40K**	**50K**	**60K**	**70K**	**80K**	**90K**	**100K**	**200K**	**300K**	**400K**	**500K**	**600K**	**700K**	**800K**	**900K**	**1M**
# Vectors	71	135	169	226	290	392	407	453	515	542	1,161	1,775	2,319	2,858	3,502	4,086	4,641	5,229	5,671
Full-coverage (one-hop)																			
Embedding-RAG																			
# Vectors (↓)	2	4	4	3	1	11	13	4	7	14	17	41	57	48	68	45	90	94	104
Chars (↓)	118	850	758	363	62	1,556	2,258	1,415	1,721	3,284	4,985	7,879	10,185	11,106	12,182	9,589	18,310	22,216	23,589
BM25-RAG																			
# Vectors (↓)	5	8	6	9	13	18	31	20	37	33	59	110	147	168	214	195	251	285	307
Chars (↓)	1,607	2,919	1,238	2,438	4,858	3,603	9,067	6,816	12,364	10,194	21,656	35,511	49,102	56,184	65,122	70,084	80,559	94,943	102,475
Reranker-RAG																			
# Vectors (↓)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Chars (↓)	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62	62
Full-coverage (two-hop)																			
Embedding-RAG																			
# Vectors (↓)	11	24	18	45	41	72	69	65	70	101	176	287	340	422	530	630	737	823	920
Chars (↓)	1,681	3,706	3,518	9,442	10,675	12,552	13,877	15,742	18,841	24,965	45,742	66,739	72,493	93,694	113,052	136,498	167,983	180,487	217,152
BM25-RAG																			
# Vectors (↓)	13	33	24	49	50	78	82	74	90	118	202	337	407	510	640	742	855	971	1,061
Chars (↓)	2,877	6,085	5,582	11,192	13,144	14,521	18,093	17,787	26,198	30,901	55,545	84,908	98,295	123,608	150,872	178,140	210,240	235,357	265,785
Reranker-RAG																			
# Vectors (↓)	2	2	5	11	3	18	15	9	9	17	30	36	60	67	72	91	117	115	149
Chars (↓)	118	129	602	1,156	476	2,418	1,957	2,312	1,868	2,162	4,679	6,364	9,074	10,919	9,497	13,532	17,431	16,042	21,372

Table 14: **Retrieval analysis of three RAG pipelines under NoLiMa**, showing 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 for BM25-RAG, Embedding-RAG, and Reranker-RAG. Here all the results are computed only on Book 1 (B_1^c). Lower (↓) values are better.

acters) for Embedding-RAG. Thus, replacing semantic retrieval with Okapi BM25 does not remove the full-coverage bottleneck; instead, lexical retrieval generally requires retrieving more of the haystack to guarantee that all necessary evidence is available. This gap suggests that semantic similarity provides a stronger ranking signal for isolating relevant evidence as the distractor pool expands. Moreover, the persistent two-hop overhead across both retrievers indicates that jointly recovering multiple evidence pieces remains intrinsically more demanding than single-evidence retrieval.

As shown in Table 14, Qwen3-Reranker-8B improves retrieval efficiency over BM25-RAG and Embedding-RAG. For one-hop queries, evidence ranks first across all context lengths, yielding full coverage at top-1 with 62 characters. For two-hop queries, depth grows with context length: at 1M characters, Reranker-RAG reaches full coverage with 149 vectors (21,372 characters), compared with 920 vectors (217,152 characters) for Embedding-RAG and 1,061 vectors (265,785 characters) for BM25-RAG. Thus, a fine-tuned reranker dramatically reduces the retrieval budget for full evidence coverage. Nevertheless, increasing top-k for two-hop queries shows that stronger reranking does not eliminate the difficulty of jointly surfacing evidence as the haystack grows. The one-hop–two-hop contrast suggests that evidence multiplicity, not context length alone, drives the remaining burden: joint coverage becomes harder when multiple passages must survive the cutoff.

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: 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

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 15: **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.

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 use the same answer+evidence prompt as the long-context EBG setting, constructed from the selected top-k chunks and gold evidence metadata. Models produce an answer with cited evidence. We denote retrieval and matched long-context answer accuracy as $\mathrm{Ans}_{\mathrm{RAG}}$ and $\mathrm{Ans}_{\mathrm{LC}}$, respectively. The primary comparison is the accuracy delta:

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

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 15 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 11 reports retrieval answer accuracy under the answer+evidence prompt, and Figure 12 reports the resulting Δ difference values across models and matched context lengths.

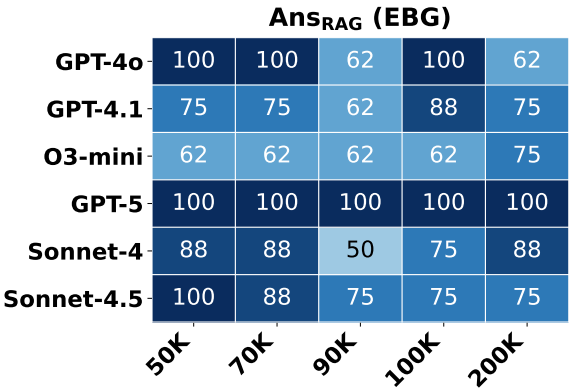

Figure 11: **Retrieval answer accuracy under the EBG prompting for the matched top-k retrieval contexts**. Columns denote the long-context character budgets we use for comparison after matching each retrieved top-k context to the nearest long-context haystack length. Rows denote the evaluated models. 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.

Observations. Figure 12 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 $\Delta = \mathrm{Ans}_{\mathrm{RAG}} - \mathrm{Ans}_{\mathrm{LC}}$, 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

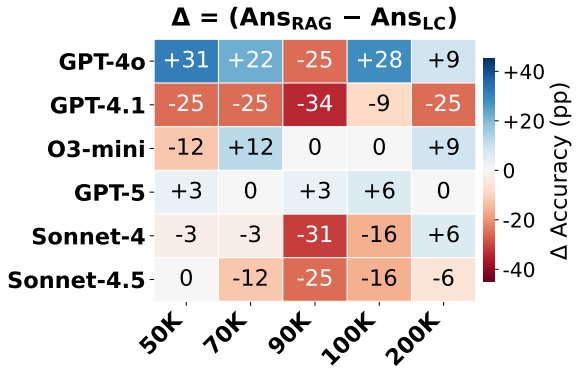

Figure 12: **Answer accuracy difference between RAG-EBG and LC-EBG**, compared under approximately matched character budgets (Table 15). Positive values indicate that RAG-EBG outperforms LC-EBG; negative values indicate that LC-EBG outperforms RAG-EBG. Higher (↑) positive values are better for RAG.

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 budget, which favors long-context prompting most strongly. Four models show large negative deltas: GPT-4o (-25,pp), GPT-4.1 (-34,pp), Sonnet-4 (-31,pp), and Sonnet-4.5 (-25,pp). Evidence completeness therefore does not guarantee retrieval superiority. Even when the top-k context contains the required needles, the model may benefit from broader discourse, surrounding distractors, or positional structure of the original haystack. Retrieval can therefore alter the contextual relationships through which evidence is interpreted, making compression effectiveness model-dependent. Conversely, strong positive deltas for GPT-4o and near-zero deltas for GPT-5 show that compact retrieval removes distractor burden without hurting 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.

G.5 Embedding-RAG vs Reranker-RAG under EBG Prompting

The preceding retrieval analysis (Table 14) shows that Qwen3-Reranker-8B can rank required evidence substantially closer to the top of the retrieved context than embedding-based similarity search. However, improved retrieval ranking does not necessarily translate into better end-to-end generation: even when the same evidence is available, its position within the retrieved context may affect a model's ability to identify, reason over, and correctly attribute that evidence. Motivated by this distinction, we evaluate whether the retrieval-order improvements from reranking translate into higher joint `Ans+Evid` accuracy when the amount of retrieved context is held fixed.

Setup. We compare Embedding-RAG and Reranker-FixedTopk-RAG across four LLMs. For each LC-matched budget in Table 15, Embedding-RAG retrieves full-coverage top-k chunks by cosine similarity. Reranker-RAG uses *the same chunks* and preserves the top-k and character budget, but re-orders them using `Qwen3-Reranker-8B` relevance scores. No chunks are added or removed; the pipelines differ only in evidence ordering. This comparison isolates whether placing relevant evidence earlier improves generation rather than conflating ranking quality with context volume. With evidence availability and context size fixed, differences can be attributed to ranking and reveal whether models benefit from prioritizing relevant chunks. We evaluate both pipelines with our EBG prompt over $n = 8$ examples, comprising four needles under one-hop and two-hop queries. Results appear in Table 16.

Model	50K		70K		90K		100K		200K	
	Emb-RAG	Reranker-RAG	Emb-RAG	Reranker-RAG	Emb-RAG	Reranker-RAG	Emb-RAG	Reranker-RAG	Emb-RAG	Reranker-RAG
Sonnet-4.5	100.0	100.0	87.5	100.0	75.0	100.0	75.0	100.0	75.0	87.5
GPT-4.1	75.0	100.0	75.0	87.5	62.5	87.5	87.5	75.0	75.0	87.5
GLM-5.2	62.5	50.0	62.5	75.0	75.0	75.0	50.0	75.0	50.0	75.0
GPT-OSS-120B	62.5	75.0	75.0	87.5	87.5	87.5	62.5	100.0	50.0	87.5

Table 16: **Comparison of Embedding-RAG and Reranker-RAG `Ans` accuracy under EBG prompting.** Columns are indexed by the LC-matched retrieved-context budgets from Table 15; for example, **200K** denotes the retrieval budget matched to the **1M** LC haystack. Each cell reports answer accuracy over $n = 8$ samples (4 needles $\times$ one-hop/two-hop). Emb: Embedding-RAG. Higher (↑) values indicate better performance.

Dataset	Source	License	Eligible / Constructed Pool	Samples Used
Synthetic	Ours	CC BY 4.0	66,834 distractor facts; 10 two-hop needles	100
NoLiMa-based	Modarressi et al. (2025)	Adobe Research License	4 needle variants $\times$ 4 insertion depths	16
MultiHop-RAG	Tang and Yang (2024)	ODC-By	2,255 answerable queries	50, 200
QASPER-MP	Dasigi et al. (2021)	CC BY 4.0	Dev split: 1,005 questions over 281 papers	50, 200

Table 17: **Dataset statistics.** For the synthetic benchmark, we construct 66,834 unique single-sentence distractor facts and curate 10 two-hop needles, evaluating 100 independently generated trials per context length. For the NoLiMa-based benchmark, we use four primary needle variants at four insertion depths, yielding 16 needle-placement conditions per context length per model. For MultiHop-RAG, removing 301 `null_query` examples leaves 2,255 answerable queries; we evaluate 50 queries with proprietary models and 200 with open-weight models. For QASPER-MP, experiments use the QASPER development split containing 1,005 questions over 281 papers. Each gold paper is paired with ten distractor papers. We evaluate 50 queries (38 answerable and 12 unanswerable) with proprietary models and 200 queries (150 answerable and 50 unanswerable) with open-weight models.

Observations. Table 16 shows that reranking improves end-to-end answer accuracy over embedding-based retrieval, with gains particularly pronounced at larger context budgets. Sonnet-4.5 benefits consistently from reranking, reaching 100% accuracy from 50K–100K and retaining an advantage at 200K. GPT-OSS-120B exhibits the largest gain at longer contexts, improving from 62.5% to 100% at 100K and from 50% to 87.5% at 200K. GPT-4.1 and GLM-5.2 also improve in most settings, although reranking is not uniformly beneficial: GPT-4.1 drops from 87.5% to 75% at 100K, while GLM-5.2 drops from 62.5% to 50% at 50K and remains unchanged at 90K.

Overall, across the 20 model–budget combinations, Reranker-RAG improves over Embedding-RAG in 15 cases, ties in three, and underperforms in only two, indicating that **improved evidence ordering can translate into better downstream answer generation even when the retrieved context budget is fixed.** The larger gains at higher budgets suggest that retrieval quality remains important even when more context is available, particularly as irrelevant passages increasingly compete for the reader's attention. Because both pipelines contain identical chunks, these gains isolate ordering rather than evidence availability, suggesting that readers are sensitive to where relevant evidence appears within the retrieved context.

H Dataset Source and Statistics

Table 17 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 entirely 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 eight proprietary frontier and three open-weight models via their APIs. Table 18 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.

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
GLM-5.2	Fireworks	https://fireworks.ai/models/fireworks/glm-5p2	MIT
GPT-OSS-120B	Fireworks	https://fireworks.ai/models/fireworks/gpt-oss-120b	Apache-2.0
Nemo-3.5-L	Fireworks	https://fireworks.ai/models/fireworks/nemotron-lightning-3p5-30b-a3b	OpenMDW-1.1
Qwen3-Reranker-8B	Fireworks	https://fireworks.ai/models/fireworks/qwen3-reranker-8b	Apache-2.0
text-embedding-3-large	OpenAI	https://developers.openai.com/api/docs/models/text-embedding-3-large	Proprietary
text-embedding-3-small	OpenAI	https://developers.openai.com/api/docs/models/text-embedding-3-small	Proprietary

Table 18: Models benchmarked in this work, along with their providers, source URLs, and licenses.

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.