ACE-RAG: Is Broader Retrieval Always Better for GraphRAG?

Anonymous ACL submission

Abstract

Graph-based RAG (GraphRAG) improves evidence coverage for multi-hop QA, but broad graph expansion can also introduce distractors and redundant context that weaken answerability. We frame GraphRAG retrieval as *answerable evidence-chain retrieval* and propose **ACE-RAG** (**A**nswerable **C**hain **E**vidence Retrieval for GraphRAG). ACE-RAG retrieves heterogeneous seeds from a multi-view graph and refines them into compact evidence chains through Local Refinement, which involves mention-edge expansion, cue selection, and anchor bundling. Across diverse settings, ACE-RAG achieves the best EM/F1 with the shortest retrieved context, suggesting that compact answerable chains are more effective than simply broadening retrieved context.

1 Introduction

Multi-hop question answering exposes a central tension in retrieval-augmented generation: the retriever must gather connected evidence, but overly broad retrieval can bury the answer path in distractors. Retrieval-Augmented Generation (RAG) grounds Large Language Models (LLMs) with external knowledge (Lewis et al., 2020), yet in multi-hop QA, the retrieval target is not a single relevant chunk but a connected reasoning path over multiple evidence units. GraphRAG addresses this need by structuring documents as graphs and retrieving relationally connected evidence through traversal or graph-aware retrieval (Edge et al., 2024; Gutiérrez et al., 2024; Ma et al., 2025).

A common strategy in GraphRAG and multi-hop retrieval is to expand evidence coverage to reduce the risk of missing necessary facts, for example by retrieving many passages, organizing documents hierarchically, or building graph-based memory structures (Sarthi et al., 2024; Guo et al., 2025; Gutiérrez et al., 2024). However, retrieval coverage is necessary but insufficient for answerability: broader graph expansion can also introduce redundant neighbors, weakly related facts, and distractors. In long contexts, useful evidence may be displaced or mixed with noise, so high retrieval recall does not necessarily make the context answerable (Liu et al., 2024; Yoran et al., 2024).

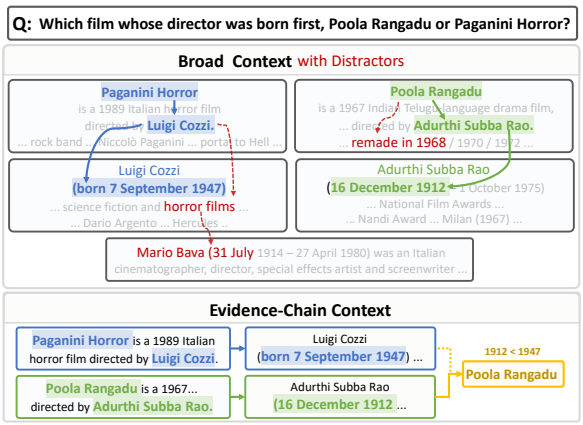

Figure 1: Motivating example. Broad retrieval contains distractors, while ACE-RAG exposes the evidence chain needed for comparison.

Figure 1 illustrates this coverage–answerability gap. Broad retrieval contains weakly related facts, whereas a compact evidence chain exposes the comparison path needed for answering. Recent RAG evaluation work similarly argues that retrieval quality should consider context precision, recall, and faithfulness together (Es et al., 2024), while adaptive and long-context RAG studies show that retrieved context should be controlled rather than indiscriminately expanded (Asai et al., 2024; Laitenberger et al., 2025). This motivates a retrieval objective that prioritizes compact evidence organization, not coverage

alone.

We propose **ACE-RAG** (**A**nswerable **C**hain **E**vidence Retrieval for GraphRAG), a compact evidence-chain retrieval framework. ACE-RAG builds a multi-view heterogeneous graph with doc-anchor, chunk, and sentence nodes, identifies heterogeneous seeds through Global Seed Search, and refines them into compact evidence chains through Local Refinement. This refinement stage involves mention-edge expansion, residual query cues, and anchor bundling to recover answer-supporting chains.

Our contributions are:

- We formulate the coverage–answerability gap in GraphRAG: high evidence recall can still yield weak QA performance when retrieved contexts are noisy, redundant, or poorly organized.
- We introduce ACE-RAG, a compact evidence-chain retrieval framework that combines multi-view seed search with Local Refinement.
- We show that ACE-RAG improves common-prompt context effectiveness, achieving higher EM/F1 with shorter retrieved contexts and exposing the limits of coverage-oriented GraphRAG.

2 Method

2.1 Overview

ACE-RAG performs seed-then-refine retrieval over a multi-view heterograph. Phase 1 selects heterogeneous access seeds across doc anchors, chunks, and sentences; Phase 2 refines them into compact evidence chains through mention-edge expansion, residual cue selection, and anchor bundling.

2.2 Multi-View Heterograph

The coverage–answerability analysis in Proposition 1 suggests that retrieval should preserve evidence roles rather than maximize flat coverage alone. ACE-RAG therefore organizes the corpus as a heterogeneous graph:

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}), \quad \mathcal{V} = \mathcal{V}_A \cup \mathcal{V}_C \cup \mathcal{V}_S, \tag{1}$$

where $\mathcal{V}_A$, $\mathcal{V}_C$, and $\mathcal{V}_S$ denote doc-anchor, chunk, and sentence nodes. Doc anchors provide entity-level access, chunks provide local context, and sentences provide fine-grained evidence.

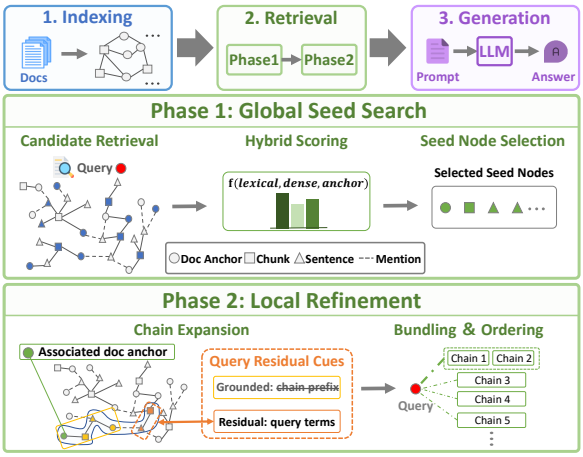

Figure 2: ACE-RAG framework. Phase 1 selects heterogeneous seeds, and Phase 2 refines compact evidence chains via mention-edge expansion, residual cues, and anchor bundling.

The edge set combines document hierarchy and mention links:

$$\mathcal{E} = \mathcal{E}_h \cup \mathcal{E}_m, \quad (s, a') \in \mathcal{E}_m, \tag{2}$$

where $\mathcal{E}_h$ preserves document structure and $\mathcal{E}_m$ links a sentence s to a mentioned doc anchor a'. A basic ACE-RAG chain is therefore

$$a_s \to s_s \xrightarrow{\text{mention}} a_m \to s_m,$$

which enables controlled chain refinement without arbitrary graph expansion.

2.3 Phase 1: Global Seed Search

Phase 1 selects heterogeneous seeds as access points for later chain refinement, rather than retrieving every required evidence unit directly. Given a query q, ACE-RAG builds u_q from lexical terms, dense semantics, and title/entity aliases, and retrieves candidates from all graph views:

$$\mathcal{C}_q = \mathcal{C}_A(u_q) \cup \mathcal{C}_C(u_q) \cup \mathcal{C}_S(u_q). \tag{3}$$

Doc anchors provide entity-level access, chunks provide local context, and sentences provide fine-grained evidence access.

Each candidate c_i is scored by lexical relevance, dense similarity, and doc-anchor matching:

$$r_i = \lambda_{\text{lex}} r_i^{\text{lex}} + \lambda_{\text{dense}} r_i^{\text{dense}} + \lambda_{\text{anchor}} r_i^{\text{anchor}}. \tag{4}$$

Method	HotpotQA			2Wiki			MuSiQue			PopQA		
	R@5	EM/F1	Tok.	R@5	EM/F1	Tok.	R@5	EM/F1	Tok.	R@5	EM/F1	Tok.
BM25	67.6	17.9/21.7	749	53.1	3.6/4.2	1111	28.4	1.5/2.2	835	38.1	26.7/35.0	773
Dense RAG	94.4	25.7/31.4	658	76.5	7.4/8.1	794	69.3	3.0/4.3	742	51.1	32.4/41.7	739
RAPTOR	41.0	12.3/15.0	3268	56.1	6.5/7.5	3098	42.3	3.8/4.9	3284	38.6	19.5/27.2	3264
HippoRAG2	95.1	25.9/31.3	668	**89.7**	8.0/8.9	845	72.2	4.1/5.5	749	51.6	32.9/41.6	743
LightRAG	92.2	26.1/32.3	641	80.2	8.3/9.5	784	54.0	3.0/4.5	705	35.5	30.5/39.1	741
ACE-RAG	**97.0**	**30.9/37.9**	**458**	88.1	**11.5/14.3**	**462**	**75.7**	**5.1/7.0**	**592**	**56.1**	**38.9/47.6**	**436**

Table 1: Main results under the common QA prompt. R@5, EM, and F1 are reported as percentages; Tok. denotes average retrieved-context tokens.

The weights are fixed across datasets. ACE-RAG selects seeds by relevance, using estimated rendering length $\ell(c_i)$ as a tie breaker:

$$\mathcal{Z}_q = \mathrm{TopK}_{K_s}(\mathcal{C}_q; r_i, -\ell(c_i)). \tag{5}$$

The seed budget K_s is distinct from the final rendering budget; Phase 1 improves seed access, while Phase 2 performs refinement.

2.4 Phase 2: Local Refinement

Phase 2 refines seed-grounded anchors into compact evidence chains without broad graph expansion. Its goal is to turn access seeds into answer-supporting chains by adding linked and unresolved evidence, rather than by increasing context breadth.

Chain Expansion. Each seed z is mapped to a doc anchor $a(z)$: anchor seeds map to themselves, while chunk and sentence seeds map to their parent anchor. ACE-RAG then follows a mention edge from an anchor-side sentence s_s to a mentioned anchor a_m, forming

$$t = (a_s, s_s, a_m), \quad a_s = a(z).$$

This creates a controlled multi-hop prefix through mention-linked anchors.

Residual query cues. To complete the mention side, ACE-RAG selects evidence under a_m using unresolved query information. Let $\tau(x)$ denote normalized lexical terms from x. For prefix t,

$$\begin{aligned} G_t &= \tau(a_s) \cup \tau(s_s) \cup \tau(a_m) \cup \Omega_{\text{stop}}, \\ \rho_t &= \tau(q) \setminus G_t. \end{aligned} \tag{6}$$

Here, Ω_{stop} is a stopword set, G_t is the grounded term set, ρ_t is the residual cue set, and $\mathcal{S}(a_m)$ is the set of sentences under anchor a_m. ACE-RAG selects the mention-side sentence

$$s_m = \mathrm{TopLex}_1\left(\mathcal{S}(a_m); \rho_t, u_q\right), \tag{7}$$

where TopLex_1 ranks candidates by residual cue coverage, lexical relevance, and rendering length. This targets query cues not yet grounded by the seed-side prefix.

Anchor bundling and rendering. For questions with multiple matched entities, ACE-RAG selects anchor-local evidence, orders chains by completeness, mention connectivity, query relevance, and length, and serializes deduplicated chains into a compact context. This compact rendering reduces the chance that required evidence is displaced by non-essential context. Thus, Local Refinement strengthens chain answerability through mention-edge expansion, residual-cue selection, and multi-anchor bundling without broad context expansion.

3 Experiments

3.1 Setup

Datasets and Baselines. We evaluate on HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (2Wiki) (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and PopQA (Mallen et al., 2023). We compare ACE-RAG against representative retrieval methods: BM25 (Robertson and Zaragoza, 2009), Dense RAG (Karpukhin et al., 2020), RAPTOR (Sarthi et al., 2024), HippoRAG2 (Gutiérrez et al., 2025), and LightRAG (Guo et al., 2025).

Metrics and Evaluation Protocol. All main results use the same QA prompt, generator, embedding model, and answer normalization to isolate retrieval-context quality. We report EM, token-level F1, Recall@5, average

Method	CtxTok↓	F1/1K↑	Ret. ms↓	Gen. ms↓
BM25	867	0.202	**31**	80
Dense RAG	733	0.300	33	**75**
RAPTOR	3228	0.042	73	174
HippoRAG2	751	0.302	762	76
LightRAG	718	0.304	312	76
ACE-RAG	**487**	**0.587**	278	83

Table 2: Average context-token and runtime efficiency under the common QA prompt.

Component	HotpotQA F1 (Δ)	2Wiki F1 (Δ)
ACE-RAG	37.9 (–)	14.3 (–)
w/o Mention Edge	30.2 (-7.7)	10.2 (-4.1)
w/o Residual Cues	36.6 (-1.3)	11.7 (-2.6)
w/o Chain Order	37.6 (-0.3)	14.2 (-0.1)
w/o Anchor Bundle	37.0 (-0.9)	10.7 (-3.6)

Table 3: Ablation study under the compact common-prompt setting.

context tokens, and per-query retrieval/generation runtime. Recall@5 is computed before generation with the same supporting-evidence evaluator for all methods. Further implementation details are provided in Appendix D.

3.2 Main Result

Table 1 shows that ACE-RAG achieves the best EM/F1 on all datasets while using the shortest retrieved contexts under the controlled common prompt. The clearest coverage–answerability gap appears on 2Wiki: HippoRAG2 obtains the highest R@5, but ACE-RAG improves F1 by 5.4 points while reducing context tokens from 845 to 462, suggesting that higher coverage alone is insufficient when the rendered context is less answer-focused.

On HotpotQA and MuSiQue, ACE-RAG achieves both the highest R@5 and the best EM/F1 with the shortest contexts, indicating that compact chain refinement can maintain strong coverage while reducing redundant context. PopQA further shows that this strategy also benefits entity-centric factual QA, improving F1 by 5.9 points over the strongest baseline with only 436 context tokens. Overall, the results suggest that ACE-RAG improves answer generation by organizing retrieved evidence into compact, query-grounded chains rather than simply expanding context.

3.3 Efficiency Analysis

Table 2 shows that ACE-RAG is optimized for token utility rather than minimum latency. Its retrieval stage is slower than lightweight sparse or dense retrieval because Local Refinement performs structured evidence-chain construction, including mention-edge-based Chain Expansion. RAPTOR is retrieval-efficient, but its long hierarchical-summary contexts and low F1 per 1K tokens show that low retrieval latency alone does not yield an efficient QA pipeline. Among structure-augmented methods, ACE-RAG is substantially faster than HippoRAG2 and LightRAG while producing the shortest context. Its F1 per 1K context tokens is nearly twice that of the strongest baseline, showing that the efficiency gain reflects a better quality–budget trade-off, not merely shorter contexts.

3.4 Ablation Study

Table 3 shows that Local Refinement components are critical for answer-supporting compact contexts. Removing Mention Edge causes the largest F1 drop, confirming that explicit bridge construction is central. Residual Cues improve mention-side evidence selection, while Anchor Bundle is especially important for 2Wiki comparison questions. The small effect of Chain Order suggests that ACE-RAG's gains come mainly from completing the right evidence chains rather than serialization order.

4 Conclusion

We proposed ACE-RAG, a compact evidence-chain retrieval framework for GraphRAG. Under controlled common-prompt evaluation, ACE-RAG achieves the best EM/F1 on all four datasets with the shortest retrieved contexts. The 2Wiki result highlights the coverage–answerability gap: despite HippoRAG2's higher Recall@5, ACE-RAG improves F1 by 5.4 points while reducing context tokens from 845 to 462. Across datasets, ACE-RAG achieves the highest token-normalized F1 of 0.587, and ablations confirm that Local Refinement, especially Mention Edge, is the key mechanism behind organizing usable evidence. Overall, GraphRAG retrieval should shift from broad evidence accumulation toward compact, answerable evidence-chain construction.

5 Limitations

This work emphasizes controlled retrieval-side evaluation under a shared QA prompt. Native-prompt results in Appendix B are reported as method-specific end-to-end analysis because prompt and rendering interfaces can affect final QA behavior.

ACE-RAG constructs compact chains using document anchors and mention edges. Future work can further examine robustness to noisy aliases, missing mention links, and broader document domains. ACE-RAG also adds structured local-refinement overhead during retrieval, trading minimal latency for shorter and more answer-focused contexts.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

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

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2025. LightRAG: Simple and fast retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10746–10761, Suzhou, China. Association for Computational Linguistics.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 59532–59569.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From RAG to memory: Non-parametric continual learning for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 21497–21515. PMLR.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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

Alex Laitenberger, Christopher D. Manning, and Nelson F. Liu. 2025. Stronger baselines for retrieval-augmented generation with long-context language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 32559–32569, Suzhou, China. Association for Computational Linguistics.

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

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.

Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo. 2025. Think-on-graph 2.0: Deep and faithful large language model reasoning with knowledge-guided retrieval augmented generation. In *The Thirteenth International Conference on Learning Representations*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations*.

A Theoretical Motivation

This section provides an idealized formal motivation for ACE-RAG. It is not an end-to-end optimality guarantee. Instead, it formalizes why multi-hop QA retrieval should prioritize compact chain answerability over broad evidence coverage, and why a seed-then-refine strategy is a natural approximation to this objective.

Definitions. For a query q, let

$$E_q^\star = \{e_1^\star, \ldots, e_L^\star\}$$

denote the indispensable evidence chain required to answer q. Let D_q be query-relevant but non-indispensable evidence, and define

$$\mathcal{R}_q = E_q^\star \cup D_q.$$

For a rendered context C, define

$$\mathrm{Cov}_q(C) = \frac{|C \cap \mathcal{R}_q|}{|\mathcal{R}_q|}.$$

We say that C is chain-answerable if

$$\mathrm{Ans}_{\mathrm{chain}}(q, C) = 1 \iff E_q^\star \subseteq C.$$

For the budget argument below, let B be a prompt-token budget and let Γ_B denote the effective evidence-unit capacity induced by B under a fixed rendering granularity.

Proposition 1 (Coverage does not imply answerability)**.** *Higher evidence coverage does not necessarily imply chain answerability.*

Proof. Consider a broad context

$$C_{\mathrm{broad}} = \{e_1^\star, \ldots, e_{L-1}^\star\} \cup D_q,$$

which retrieves all peripheral relevant evidence but misses the indispensable unit $e_L^\star$. Its coverage is

$$\mathrm{Cov}_q(C_{\mathrm{broad}}) = \frac{|D_q| + L - 1}{|D_q| + L}.$$

The compact chain context $C_{\mathrm{chain}} = E_q^\star$ has

$$\mathrm{Cov}_q(C_{\mathrm{chain}}) = \frac{L}{|D_q| + L}.$$

When $|D_q| > 1$,

$$\mathrm{Cov}_q(C_{\mathrm{broad}}) > \mathrm{Cov}_q(C_{\mathrm{chain}}).$$

However, C_{broad} is not chain-answerable because it misses $e_L^\star$, whereas C_{chain} contains the complete indispensable chain. Thus, higher coverage does not imply answerability. □

Proposition 2 (Seed access motivates local refinement)**.** *Let H be the event that the selected seed set contains at least one seed lying on or directly connected to the indispensable evidence chain $E_q^\star$. Let*

$$p_s = \Pr[H].$$

Let α be a lower bound on the conditional probability that local refinement produces a chain-answerable context given such a successful seed:

$$\Pr[\mathrm{Ans}_{\mathrm{chain}}(q, C) = 1 \mid H] \geq \alpha.$$

Then the probability of obtaining a chain-answerable context satisfies

$$\Pr[\mathrm{Ans}_{\mathrm{chain}}(q, C) = 1] \geq p_s \alpha.$$

If M independent seed proposals are considered and each hits or connects to the chain with probability p_0, then

$$p_s = 1 - (1 - p_0)^M,$$

and therefore

$$\Pr[\mathrm{Ans}_{\mathrm{chain}}(q, C) = 1] \geq \big(1 - (1 - p_0)^M\big)\alpha.$$

Proof. Let

$$A = \{\mathrm{Ans}_{\mathrm{chain}}(q, C) = 1\}.$$

By definition, $\Pr[H] = p_s$, and by assumption,

$$\Pr[A \mid H] \geq \alpha.$$

Therefore,

$$\begin{aligned}\Pr[A] &\geq \Pr[H]\Pr[A \mid H] \\ &\geq p_s\alpha. \end{aligned} \tag{8}$$

If M independent seed proposals are considered and each hits or connects to the indispensable chain with probability p_0, then the probability that none of them provides chain access is $(1-p_0)^M$. Hence,

$$p_s = 1 - (1-p_0)^M,$$

which gives the stated bound. □

Proposition 3 (Compact chains preserve answerability under budget)**.** *Under a fixed prompt-token budget, a compact context that contains the indispensable evidence chain preserves chain answerability more reliably than a broader over-budget context.*

Proof. Let B be a prompt-token budget and let Γ_B denote the effective evidence-unit capacity induced by this budget. Suppose a broad context contains T evidence units, including all L indispensable units, with $T > \Gamma_B$ and $\Gamma_B \geq L$. Under a uniform capacity model, the probability that all indispensable units survive budgeting is

$$\Pr[\mathrm{Ans}_{\mathrm{chain}}(q, C_{\mathrm{broad}}^{(\Gamma_B)}) = 1] = \frac{\binom{T-L}{\Gamma_B - L}}{\binom{T}{\Gamma_B}}.$$

This quantity decreases as the broad context grows while Γ_B and L are fixed, because indispensable units must compete with more non-essential evidence for the same budget.

In contrast, if a compact context C_{chain} satisfies

$$E_q^\star \subseteq C_{\mathrm{chain}} \quad \text{and} \quad \ell_{\mathrm{tok}}(C_{\mathrm{chain}}) \leq B,$$

then no indispensable evidence is removed by budgeting. Therefore,

$$\Pr[\mathrm{Ans}_{\mathrm{chain}}(q, C_{\mathrm{chain}}) = 1] = 1.$$

Thus, compact chain-preserving contexts are more reliable under a fixed prompt-token budget than over-budget broad retrieval. □

Connection to ACE-RAG. Proposition 1 motivates ACE-RAG's coverage–answerability framing. Proposition 2 motivates the seed-then-refine design: Phase 1 searches for heterogeneous access points that hit or connect to the indispensable evidence chain, while Phase 2 locally refines such seeds toward compact answer-supporting chains. Proposition 3 motivates compact chain rendering under a prompt budget. In ACE-RAG, Global Seed Search retrieves heterogeneous access points, and Local Refinement uses mention-edge-based Chain Expansion, residual query cues, and anchor bundling to construct compact evidence chains. These propositions are not guarantees of ACE-RAG's end-to-end behavior; rather, they formalize the retrieval objective that ACE-RAG approximates. In particular, ACE-RAG does not assume exact chain recovery, and the parameter α captures the approximate and implementation-dependent nature of local refinement.

B Native Prompt Experiments

B.1 QA Results

Table 4 reports native-prompt results, where each method uses its own prompt interface. Method-specific prompts substantially raise absolute QA scores and can change rankings. ACE-RAG uses the section-aware native-prompt interface described in Appendix D. Because this interface changes both the prompt and the rendering interface, it is not intended as a controlled comparison against the common-prompt evaluation. ACE-RAG obtains the strongest HotpotQA and PopQA F1 and remains competitive overall, while HippoRAG2 is stronger on 2Wiki and LightRAG is slightly stronger on MuSiQue F1. We therefore treat native-prompt results as method-specific end-to-end analysis. These results suggest that ACE-RAG's compact evidence-chain interface remains effective under its native prompt, while the common-prompt setting remains the primary controlled retrieval-side comparison.

C Common Prompt Experiments

C.1 Quality–Efficiency Trade-off

Figure 3 shows the quality–efficiency trade-off of ACE-RAG under the common QA prompt

Method	HotpotQA			2Wiki			MuSiQue			PopQA		
	R@5	EM/F1	Tok.	R@5	EM/F1	Tok.	R@5	EM/F1	Tok.	R@5	EM/F1	Tok.
BM25	67.6	34.0/43.4	1516	53.1	18.4/23.0	1680	28.4	7.1/12.3	1593	38.1	39.0/49.9	1535
Dense RAG	94.4	48.5/60.2	1432	76.5	34.9/40.6	1471	69.3	18.4/26.6	1509	51.1	46.4/58.3	1470
RAPTOR	41.0	17.1/21.4	3349	56.1	18.3/21.1	3349	42.3	5.3/7.9	3349	38.6	20.5/31.4	3349
HippoRAG2	95.1	49.8/61.2	1442	**89.7**	**40.2/47.2**	1508	72.2	19.1/28.2	1515	51.6	45.8/57.9	1476
LightRAG	92.1	48.9/60.5	1081	79.2	34.6/39.4	1222	61.7	18.9/**28.3**	1144	35.1	39.7/48.2	1171
ACE-RAG	**97.0**	**50.4/63.4**	**985**	88.1	36.0/43.4	**993**	**75.7**	**19.7**/27.7	1199	**56.2**	**46.8/59.8**	**1017**

Table 4: Native-prompt results with method-specific QA prompts. Tok. denotes average full input tokens, including retrieved context and prompt text.

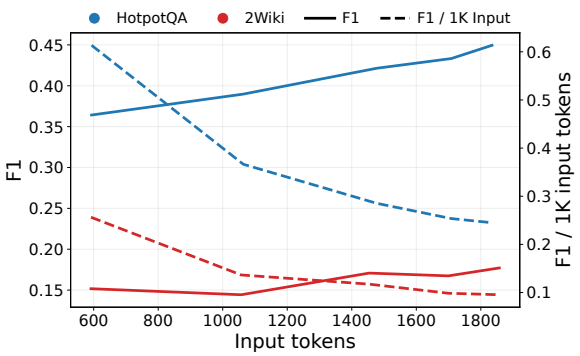

Figure 3: ACE-RAG quality–efficiency trade-off under the common QA prompt. Solid lines denote answer F1, while dashed lines denote F1 per 1K input tokens.

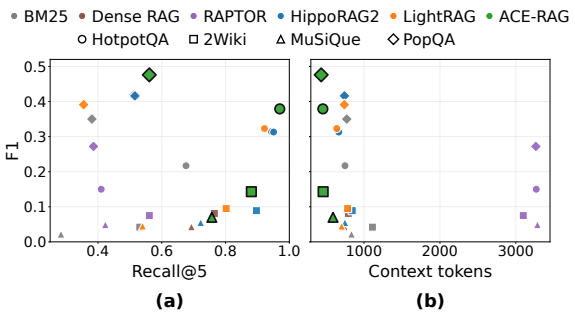

Figure 4: Coverage–answerability and token-efficiency scatter under the common QA prompt. (a) Recall@5 vs. F1 shows whether higher evidence coverage translates into answer quality. (b) Context tokens vs. F1 shows the quality–budget trade-off of retrieved contexts. Marker shapes denote datasets and colors denote methods.

on HotpotQA and 2Wiki, the two canonical multi-hop QA datasets used for sensitivity analysis. The main common-prompt result uses the compact operating point described in Appendix D, while larger evidence budgets provide a scaled setting for sensitivity analysis. As the input-token budget increases, answer F1 generally improves on the shown datasets, while the gain is not uniform across datasets. However, F1 per 1K input tokens decreases as more context is added. This indicates that compact evidence-chain contexts provide the most token-efficient operating point, while larger budgets can improve absolute answer quality at the cost of lower token-normalized efficiency.

C.2 Coverage–Answerability and Token Efficiency

Figure 4 complements the main results by visualizing the gap between retrieval coverage, answerability, and token budget. In Figure 4(a), higher Recall@5 does not consistently translate into higher F1. For example, on 2Wiki, HippoRAG2 achieves the highest Recall@5, but ACE-RAG obtains higher F1 with a slightly lower recall. This supports the view that retrieving more gold or relevant evidence is insufficient unless the retrieved context forms an answerable chain.

Figure 4(b) further shows that ACE-RAG occupies the favorable upper-left region, achieving higher F1 with substantially fewer context tokens. RAPTOR, in contrast, uses much larger contexts but does not achieve proportional answer-quality gains. This trend suggests that ACE-RAG's improvement is not merely due to retrieving more evidence, but to completing compact evidence chains that are more usable by the generator under the common prompt. This empirical pattern is consistent with Proposition 1: answer quality depends not only on coverage, but also on whether the rendered context compactly preserves the evidence needed for answering.

Item	Setting
Retrieval policy	Fixed across all datasets; anchor, chunk, and sentence candidates are used.
Controlled evaluation	Common QA prompt with compact chain rendering.
Native-prompt interface	Same retrieval policy with ACE-RAG's section-aware rendering interface and native prompt.
Evaluation data	HotpotQA, 2Wiki, MuSiQue, and PopQA; 1,000 queries per dataset.
Models	NV-Embed-v2 for dense retrieval; Qwen2.5-7B-Instruct for generation.
Environment	Python 3.10, PyTorch 2.2.0, CUDA 12.1, vLLM, and NVIDIA H100 GPU.
Runtime	Per-query retrieval and generation time; offline indexing time is excluded.
Artifacts	Public QA benchmarks and baseline methods are used only for research evaluation.

Table 5: Implementation and evaluation summary. All datasets use the same ACE-RAG retrieval policy and evaluation protocol.

D Implementation and Evaluation Details

ACE-RAG uses one frozen, dataset-independent retrieval policy across all reported experiments. Candidates are retrieved from anchor, chunk, and sentence views; seeds are selected by top-relevance ranking; and Local Refinement is applied with Chain Expansion over mention edges, residual cue selection, anchor-aware ordering, and multi-anchor bundling. No gold answers or supporting-fact labels are used during retrieval, chain refinement, rendering, or generation.

The common-prompt evaluation uses format-neutral compact chain rendering and serves as the main controlled comparison, since all methods share the same QA prompt. The native-prompt interface uses ACE-RAG's section-aware rendering interface and native prompt, and is reported only as method-specific end-to-end analysis. In common-prompt tables, *Tok.* denotes retrieved-context tokens; in native-prompt tables, *Tok.* denotes full input tokens, including retrieved context and method-specific prompt text.

We use publicly available QA benchmark datasets and baseline methods, cite their original creators in Section 3, and do not collect new data, recruit human annotators, or introduce additional personally identifying information. The released code archive contains scripts and configuration files, but does not include the original benchmark datasets by default; users should obtain the datasets from their original sources and place them in the expected data layout. All datasets, baseline implementations, and model artifacts are used according to their original licenses and release terms.

E Prompt Settings

We report the exact prompts used for the common-prompt evaluation and native-prompt interface. The common prompt is format-neutral and is shared by all methods. The native prompt is ACE-RAG-specific and assumes the section-aware evidence interface described in Appendix D.

E.1 Common QA Prompt

Common QA prompt

```
You are a QA assistant.
Answer the question using only the provided
Context.
When possible, use a concise answer explicitly
supported by the Context.
Return only the final short answer.
Do not output reasoning, explanations, citations,
Markdown, or prefixes.
For yes/no questions, output exactly yes or no in
lowercase.
If the answer cannot be found in the Context,
output exactly: insufficient information


Question: {question}
Context:
{context}
Answer:
```

E.2 ACE-RAG Native Prompt

ACE-RAG native prompt

```
---Role---
You are an expert reading-comprehension assistant
for ACE-RAG structured evidence.


---Goal---
Answer the question using ONLY the provided
Context.
Return only the final short answer span.


---Instructions---
1. The Context is organized into evidence
sections.
2. Section headers identify the evidence type and
```

```
anchor entity.
3. Earlier evidence sections are usually more
relevant, but use later sections when they
complete the answer.
4. Use Evidence Chain sections to connect a
question anchor to a bridge entity or answer
entity.
5. Use Multi-Anchor Evidence sections for
comparison or questions involving multiple
entities.
6. Use Supporting Evidence sections as direct
answer evidence when they answer the question.
7. Ignore source ids, relation_title fields, and
empty evidence sections.
8. Do not use outside knowledge.
9. If a supported answer span exists anywhere in
the Context, output that answer span.
10. Output insufficient information only when no
evidence in the Context supports an answer.
11. Output exactly one line with only the final
answer.
12. Do not output reasoning, explanations,
citations, Markdown, JSON, or prefixes.
13. For yes/no questions, output exactly yes or no
in lowercase.


Question: {question}
Context:
{context}
Answer:
```