PAR^2-RAG: Planned Active Retrieval and Reasoning for Multi-Hop Question Answering

Xingyu Li, Rongguang Wang, Yuying Wang, Mengqing Guo, Chenyang Li, Tao Sheng, Sujith Ravi, Dan Roth
Oracle AI

Abstract

Large language models (LLMs) remain brittle on multi-hop question answering (MHQA), where answering requires combining evidence across documents through retrieval and reasoning. Iterative retrieval systems can fail by locking onto an early low-recall trajectory and amplifying downstream errors, while planning-only approaches may produce static query sets that cannot adapt when intermediate evidence changes. We propose **Planned Active Retrieval and Reasoning RAG (PAR^2-RAG)**, a two-stage framework that separates *coverage* from *commitment*. PAR^2-RAG first performs breadth-first anchoring to build a high-recall evidence frontier, then applies depth-first refinement with evidence sufficiency control in an iterative loop. Across four MHQA benchmarks, PAR^2-RAG consistently outperforms existing state-of-the-art baselines, compared with IRCoT, PAR^2-RAG achieves up to **23.5%** higher accuracy, with retrieval gains of up to **10.5%** in NDCG.

1 Introduction

Recent LLMs have achieved expert-level performance in domains such as mathematics and coding (OpenAI, 2023; DeepSeek-AI, 2025; Team and DeepMind, 2024; OpenAI, 2025), yet they still struggle with multi-hop question answering (MHQA), where systems must gather evidence from multiple documents and compose those facts into a coherent answer (Ho et al., 2020; Trivedi et al., 2022; Schnitzler et al., 2024; Krishna et al., 2025). Retrieval-Augmented Generation (RAG) and related approaches were designed to address this gap, but they still fail frequently in multi-hop settings when evidence coverage is incomplete or reasoning trajectories drift (Lewis et al., 2020; Gao et al., 2023; Trivedi et al., 2023).

A common failure mode in MHQA is *premature commitment*, where retrieval approaches interleave reasoning with retrieval in a mostly greedy, depth-first manner (Yao et al., 2023; Trivedi et al., 2023). When early steps latch onto a distractor, later hops amplify that error and recovery becomes difficult. Meanwhile, planning-heavy methods can produce broad but static query sets that become misaligned when intermediate evidence changes (Cheng et al., 2025; Liu et al., 2025).

This motivates a central question: *can we improve multi-hop robustness by jointly controlling retrieval coverage and the timing of reasoning commitment?* We propose **Planned Active Retrieval and Reasoning RAG (PAR^2-RAG)**, an agentic and modular two-stage evidence search approach that separates *retrieval coverage* from *reasoning commitment*. Stage 1 performs *coverage anchor* to expand the evidence pool. Stage 2 performs *tterative chain refinement* within that anchored pool to construct a coherent retrieval evidence chain with intermediate sub-thoughts. The generation module then conditions on this structured evidence, which improves final answer quality by making synthesis more evidence-grounded.

We evaluate PAR^2-RAG on four MHQA benchmarks using multiple answer and retrieval quality metrics. PAR^2-RAG consistently outperforms strong training-free baselines on both retrieval coverage and answer accuracy, with 23.5% gain over IRCoT in accuracy and retrieval gains of10.3% in recall and 10.5% in NDCG.

We make three primary contributions:

- We introduce PAR^2-RAG, an agentic modular two-stage framework for MHQA through planned active retrieval and evidence-aware reasoning.

- We provide empirical study under both non-reasoning and reasoning settings, including reasoning-intensive MHQA benchmarks, showing consistent gains over strong training-free baselines.

- We present mechanism-level diagnostics that connect answer gains to retrieval behavior, together with robustness analyses.

2 Related Work

2.1 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) improves factual grounding by coupling generation with external evidence retrieval (Lewis et al., 2020; Guu et al., 2020). Follow-up work improves retrieval quality through dense encoders, re-ranking, and retrieval-aware prompting (Karpukhin et al., 2020; Izacard et al., 2021; Shi et al., 2023; Izacard et al., 2022; Asai et al., 2024). However, strong first-hop retrieval does not by itself solve multi-hop reasoning: systems can still fail when evidence is incomplete or poorly composed across steps (Gao et al., 2023; Tang and Yang, 2024).

2.2 Training-free Multi-Hop QA

Interleaving retrieval with reasoning is a dominant training-free strategy for multi-hop question answering (MHQA). ReAct alternates thought and action to iteratively collect evidence (Yao et al., 2023), and IRCoT interleaves chain-of-thought with retrieval to improve compositional QA performance (Trivedi et al., 2023). These methods are flexible and often strong, but retrieval and commitment are coupled in a single loop. Recent analyses of multi-hop behavior and retrieval difficulty indicate that errors in early hops can propagate and degrade later reasoning, especially when bridge evidence is fragile or missing (Biran et al., 2024; Zhu et al., 2025).

Other RAG-based approaches improve MHQA by decomposing questions into sub-queries and searching broader evidence frontiers before deep reasoning. Decomposition and structured retrieval can improve coverage, especially for bridge-fact questions, but purely static plans can become mismatched when intermediate evidence shifts the information need. Recent modular and structured pipelines (e.g., explicit retrieval-in-context search or retrieval–inference coupling) make this trade-off between breadth and commitment control explicit (Chen et al., 2025; Li et al., 2024; Cheng et al., 2025; Liu et al., 2025).

3 Methods

Multi-hop QA systems often fail when they commit to a reasoning path before evidence coverage is sufficient. This premature commitment can lock retrieval into low-recall trajectories and amplify downstream errors. Our goal is to address this failure mode by explicitly controlling *when* the system commits to reasoning and *how* retrieval is expanded beforehand. We therefore propose **Planned Active Retrieval and Reasoning RAG (PAR^2-RAG)**, a two-stage framework that separates *coverage acquisition* from *reasoning commitment*. The key design principle is *coverage first, commitment late*: first build a broad evidence frontier, then perform controlled chain refinement with explicit sufficiency checks.

3.1 PAR^2-RAG

To present PAR^2-RAG, we first formulate the multi-hop RAG. Let q denote the input multi-hop question and $D = \{d_1, \ldots, d_N\}$ the document corpus, where each d_n is a chunked passage. A multi-hop RAG process can be viewed as iterative sub-question planning, retrieval, and response generation. Within this setting, PAR^2-RAG uses five agents: (1) a Planner $\mathcal{P}(q)$ that generates coverage-oriented sub-queries to expand the evidence frontier; (2) a Retriever $\mathcal{R}(q, D, k)$ that returns top-k ranked passage evidence E_i for a query q_i; (3) a Query Formulator $\mathcal{Q}$ that rewrites follow-up queries for commitment-stage refinement; (4) an Evidence Sufficiency Controller (ESC) $\mathcal{E}$ that decides whether to continue retrieval via $\mathcal{Q}$ or stop and finalize the answer; and (5) Writer $\mathcal{W}$ to generate the intermediate response or the final answer with prompt-instructed LLM calls.

Figure 1 provides an overview of PAR^2-RAG. The architecture has two coordinated stages. Stage 1 (Coverage Anchor) performs breadth-first evidence expansion: the planner $\mathcal{P}(q)$ decomposes the query into complementary sub-queries, the retriever $\mathcal{R}$ gathers candidate passages, and the system merges them into an anchored context C_{anchor}. Stage 2 (Iterative

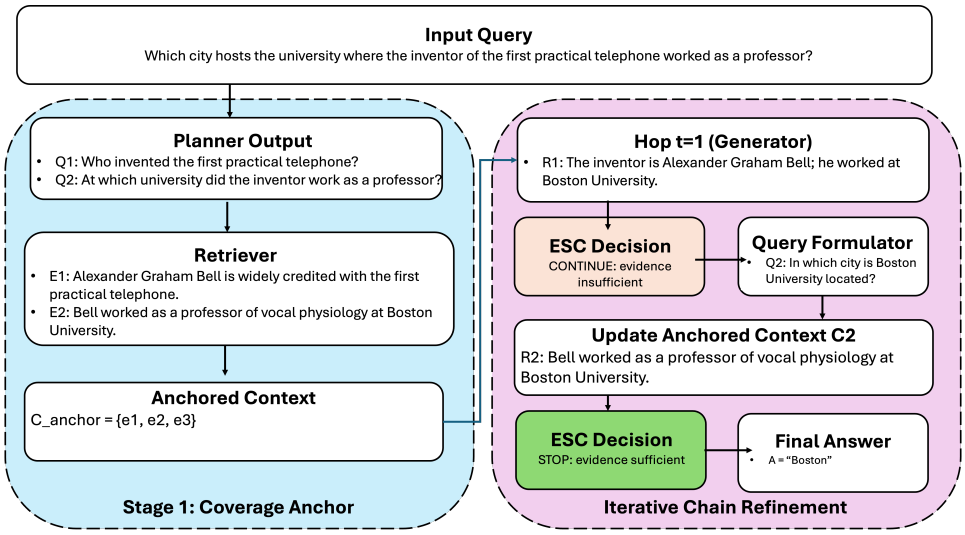

Figure 1: Overall architecture of PAR^2-RAG. Stage 1 (Coverage Anchor) expands evidence breadth to build C_{anchor}, and Stage 2 (Iterative Chain) performs ESC-gated refinement to either continue targeted retrieval or stop with the final answer.

Chain) performs depth-first refinement over this anchored context. At each hop, the writer $\mathcal{W}$ produces a step response and the ESC $\mathcal{E}$ decides whether to continue retrieval with a reformulated query or finalize the answer. This control loop makes retrieval adaptive to missing evidence while preventing premature commitment to an incomplete reasoning path.

Stage 1: Coverage Anchoring Stage 1 implements the **Coverage Anchor** module, whose objective is to construct a high-recall evidence frontier before the system commits to a narrow reasoning path. This stage uses two agents: a planner $\mathcal{P}(q)$ that produces diverse decomposed sub-queries, and a retriever–reranker $\mathcal{R}$ that executes these queries and returns ranked, deduplicated passages. Note that both agents are necessary: decomposed queries define retrieval intents, but they do not provide factual grounding by themselves. The retrieved chunks instantiate an anchor context C_{anchor} grounded in corpus evidence. This grounding reduces premature commitment, improves bridge-fact coverage, and provides a stable initialization for Stage 2 refinement.

Concretely, given q, the planner generates a diverse set of m sub-queries $\{q_i\}_{i=1}^{m}$ targeting complementary aspects of the information need. For each q_i, the system retrieves top-k candidates, reranks, deduplicates, and merges the retained passages into C_{anchor}. The resulting anchored context serves as a global evidence frontier, increasing the likelihood that required supporting facts are available before deep reasoning begins.

Stage 2: Iterative Chain Refinement Stage 2 corresponds to the **Iterative Chain** module, which aims to convert broad anchored evidence into a precise, answer-supporting reasoning chain. In contrast to Stage 1, this stage prioritizes commitment quality: deciding whether current evidence is sufficient for the next reasoning step, or whether targeted retrieval still needs refinement. The **Iterative Chain** stage uses three complementary agent: a generator $\mathcal{G}$ to produce step-level responses from the current context, a query formulator $\mathcal{Q}$ to propose focused follow-up query formulation when evidence is incomplete, and an ESC $\mathcal{E}$ to decide whether to query reformulation or answer generation. Considering at hop t, the generator $\mathcal{G}$ a step response r_t from the current context C_t (when t=1, C_1 is equal to C_{anchor}), then $\mathcal{E}$is invoked to decide the next step as:

$$\mathcal{E}(q, r_t, C_t) \rightarrow (\texttt{action}, q^{\star}_{t+1}), \qquad (1)$$

where $\texttt{action} \in \{\texttt{CONTINUE}, \texttt{STOP}\}$ and a prompt-instructed LLM makes the decision for $\mathcal{E}$. If $\mathcal{E}$ outputs `CONTINUE`, the system issues a reformulated query $q^{\star}_{t+1}$ and retrieves additional evidence for C_{t+1}; if $\mathcal{E}$ outputs `STOP`, the loop terminates and returns the answer.

Algorithm 1 summarizes the implementation of PAR^2-RAG. After stage 1, at each hop t,

Algorithm 1 PAR^2-RAG

Require: Query q, corpus D, hop budget H
Ensure: Final answer a

$\{q_i\}_{i=1}^{m} \leftarrow \mathcal{P}(q)$
Initialize C_{anchor}
for each q_i **do**
  $E_i \leftarrow \mathcal{R}(q_i, D, k)$
  $C_{\text{anchor}} \leftarrow C_{\text{anchor}} \cup E_i$
end for
$t \leftarrow 1$, $C_1 \leftarrow C_{\text{anchor}}$
while $t \leq H$ **do**
  $r_t \leftarrow \mathcal{G}(q, C_t)$
  $(\texttt{action}, q^{\star}_{t+1}, m_t) \leftarrow \mathcal{E}(q, r_t, C_t)$
  if $\texttt{action} = \texttt{STOP}$ **then**
    break
  end if
  $E_{t+1} \leftarrow \mathcal{R}(q^{\star}_{t+1}, D, k)$
  $C_{t+1} \leftarrow C_t \cup E_{t+1}$
end while
$a \leftarrow r_t$
return a

$\mathcal{G}$ produces $r_t = \mathcal{G}(q, C_t)$, then ESC returns $(\texttt{action}, q^{\star}_{t+1}, m_t)$. If $\texttt{action} = \texttt{CONTINUE}$, the generated $q^{\star}_{t+1}$ from $\mathcal{Q}$ invokes $\mathcal{R}$, and updates context to C_{t+1}; otherwise the loop terminates and returns the latest response as the final answer. This keeps retrieval adaptive while explicitly controlling continuation to reduce premature commitment and off-path reasoning.

3.2 Discussion

PAR^2-RAG is designed around a simple control principle: *retrieve broadly before committing narrowly*. Different from RAG methods such as ReAct and IRCoT, PAR^2-RAG separates coverage expansion from commitment-time refinement, so early low-recall evidence is less likely to lock the system into an off-path trajectory. Comparing to planning-only decomposition, PAR^2-RAG remains adaptive during refinement through ESC-guided continuation and query reformulation.

This agentic decomposition also clarifies the role of each component. Coverage Anchor increases evidence breadth and bridge-fact recall; Iterative Chain improves local trajectory quality; ESC decides when additional retrieval is necessary versus when evidence is sufficient to stop. Compared with single-module variants, the combined design offers complementary benefits: stronger initial recall, controlled correction when evidence is incomplete, and more stable behavior across multi-agents.

4 Experiments

4.1 Experimental Setups

Benchmarks We evaluate PAR^2-RAG on four benchmarks spanning two categories: 2WikiMultiHopQA (Ho et al., 2020) and MuSiQue (Trivedi et al., 2022), which emphasize connected multi-step evidence composition; and MoreHopQA (Schnitzler et al., 2024) and FRAMES (Krishna et al., 2025), which further stress generative reasoning and end-to-end retrieval-grounded factuality. For each dataset, we sample 500 queries from the validation split. Dataset details are provided in Appendix A.1.

Baselines To provide evaluation of PAR^2-RAG, we compare against three-types of training-free methods: (1) *Non-retrieval* baselines: Direct Inference and Chain-of-Thought (CoT) (Wei et al., 2022). (2) *Iterative retrieval–reasoning* baselines: ReAct (Yao et al., 2023) and IRCoT (Trivedi et al., 2023), which we treat as strong training-free RAG baselines. (3) *PAR^2-RAG module variants*: Coverage Anchor and Iterative Chain, which correspond to modular components of PAR^2-RAG rather than independent external methods.

Evaluating these two module variants separately helps clarify *why* PAR^2-RAG works: (a) it isolates gains from planning-oriented coverage expansion versus iterative chain refinement, (b) it reveals whether errors come from weak global evidence coverage or weak local chain updates, and (c) it quantifies the complementarity between planning and reasoning-chain control when both modules are combined.

Evaluation Metrics We evaluate both answer quality and retrieval quality. For answer quality, we use binary correctness judged against ground truth by OpenAI GPT-5-mini (Singh et al., 2025). For retrieval quality, we compare deduplicated retrieved chunk IDs against ground-truth documents using Recall@k, NDCG@k (Jeunen et al., 2024), and All-Pass (1 only when all required documents are retrieved). We report query-level averages and per-required-length diagnostics. Full metric definitions are provided in Appendix A.2.

Method	MuSiQue	2Wiki	MoreHopQA	FRAMES	Average
Direct Inference	0.204	0.418	0.144	0.294	0.265
CoT	0.338	0.724	0.594	0.631	0.572
ReAct	0.432	0.724	0.740	0.745	0.660
IRCoT	0.498	0.820	0.754	0.726	0.700
Coverage Anchor	0.539	0.835	0.753	0.774	0.725
Iterative Chain	0.516	0.868	0.800	0.788	0.743
PAR^2-RAG	**0.615**	**0.896**	**0.826**	**0.811**	**0.787**

Table 1: Non-reasoning generation answer-quality results on the considered MHQA benchmarks.

Method	MuSiQue	2Wiki	MoreHopQA	FRAMES	Average
Direct Inference	0.428	0.794	0.708	0.739	0.667
CoT	0.440	0.804	0.688	0.745	0.669
ReAct	0.575	0.864	0.832	0.833	0.776
IRCoT	0.590	0.830	0.742	0.754	0.729
Coverage Anchor	0.627	**0.912**	0.860	**0.860**	0.815
Iterative Chain	0.628	0.884	0.830	0.834	0.794
PAR^2-RAG	**0.639**	0.904	**0.864**	0.858	**0.816**

Table 2: Reasoning generation answer-quality results on the considered MHQA benchmarks.

Implementation Details Our pipeline follows ingestion, retrieval, and generation stages. During ingestion, each benchmark corpus is split into chunks and indexed in OpenSearch with 1024-dimensional embeddings. At inference time, retrieval uses two stages: broad hybrid candidate retrieval (TopK=500) followed by reranking with `e5-mistral-7b-instruct` (Wang et al., 2024) to produce a compact evidence set (TopK=5). We use GPT-4.1 for the Coverage Anchor and Iterative Chain agents, and GPT-o4-mini as the evidence controller. To test whether retrieval-policy gains persist under stronger generation-time reasoning, we evaluate both non-reasoning and reasoning generation settings using GPT-4.1 and GPT-o3, respectively. Detailed settings appear in Appendix A.4.

4.2 Main Results

Tables 1 and 2 show that PAR^2-RAG consistently delivers the strongest overall answer quality across benchmarks and generator settings. Under non-reasoning generation, PAR^2-RAG achieves the best performance on all four datasets. In particular, on MuSiQue, PAR^2-RAG yields its largest relative improvements: +42.4% over ReAct and +23.5% over IRCoT.

Under reasoning generation, PAR^2-RAG remains best on average. The largest relative gain over Coverage Anchor is +1.9% on MuSiQue, while the largest relative gain over Iterative Chain is +4.1% on MoreHopQA. The runner-up pattern also differs by regime: Iterative Chain is second-best under non-reasoning generation, while Coverage Anchor is second-best under reasoning generation. A plausible explanation is that the dominant bottleneck shifts with generator capability: with non-reasoning generation, stronger chain refinement helps more after retrieval, while with reasoning generation, broader evidence coverage becomes relatively more important.

Table 3 provides direct evidence for the mechanism. PAR^2-RAG achieves the best retrieval performance on Recall, NDCG, and All-Pass across all four datasets. These retrieval gains align with the answer-quality improvements. We hypothesize that ReAct and IRCoT are less competitive in this setting because iterative retrieval can suffer from *early commitment*: reasoning starts before evidence coverage is sufficiently broad, and early low-recall steps can bias later retrieval. In contrast, PAR^2-RAG explicitly separates coverage expansion and chain refinement, which provides more controllable behavior under fixed retrieval budgets.

4.3 Ablation Study

We conduct ablations to analyze how robustness and where PAR^2-RAG's gains come from.

Ablation on GPT-5.2 Settings We evaluate PAR^2-RAG and considered RAG-based methods with GPT-5.2 model under two reasoning configurations (`none` and `medium`) for both reasoning and generation. The experimental results in Table 4 show that PAR^2-RAG maintains clear gains over IRCoT and both module variants in both reasoning settings. Under reasoning=`none`, the largest relative gain over Coverage Anchor is +9.3% on FRAMES, and the largest gain over Iterative Chain is also +7.2% on FRAMES. Under reasoning=`medium`, the largest relative gain over Coverage Anchor is +3.5% on 2Wiki, while the largest gain over Iterative Chain is +4.8% on MuSiQue.

This extension provides a refined view of the runner-up hypothesis. At the average level, Iterative Chain remains the runner-up in both settings. However, when moving from reasoning=`none` to reasoning=`medium`, the gap between Iterative Chain and Coverage Anchor narrows substantially, and Coverage Anchor becomes competitive on multiple datasets. This trend is consistent with our hypothesis: as

Method	MuSiQue			2Wiki			MoreHopQA			FRAMES			Average		
	NDCG	Recall	All Pass	NDCG	Recall	All Pass	NDCG	Recall	All Pass	NDCG	Recall	All Pass	NDCG	Recall	All Pass
ReAct	0.596	0.581	0.240	0.724	0.683	0.386	0.896	0.891	0.782	0.717	0.671	0.355	0.733	0.707	0.441
IRCoT	0.611	0.692	0.420	0.750	0.795	0.622	0.822	0.884	0.771	0.804	0.806	0.565	0.747	0.794	0.595
Coverage Anchor	0.610	0.737	0.428	0.708	0.804	0.588	0.881	0.944	0.888	0.785	0.794	0.544	0.746	0.820	0.612
Iterative Chain	0.615	0.691	0.392	0.762	0.768	0.548	0.898	0.895	0.790	0.785	0.794	0.545	0.765	0.787	0.569
PAR^2-RAG	**0.636**	**0.747**	**0.460**	**0.784**	**0.877**	**0.728**	**0.908**	**0.954**	**0.908**	**0.834**	**0.874**	**0.677**	**0.791**	**0.863**	**0.693**

Table 3: Retrieval performance on the considered multi-hop QA benchmarks. We report NDCG, Recall, and All Pass for the retrieval stage. Best results are in bold and second-best are underlined.

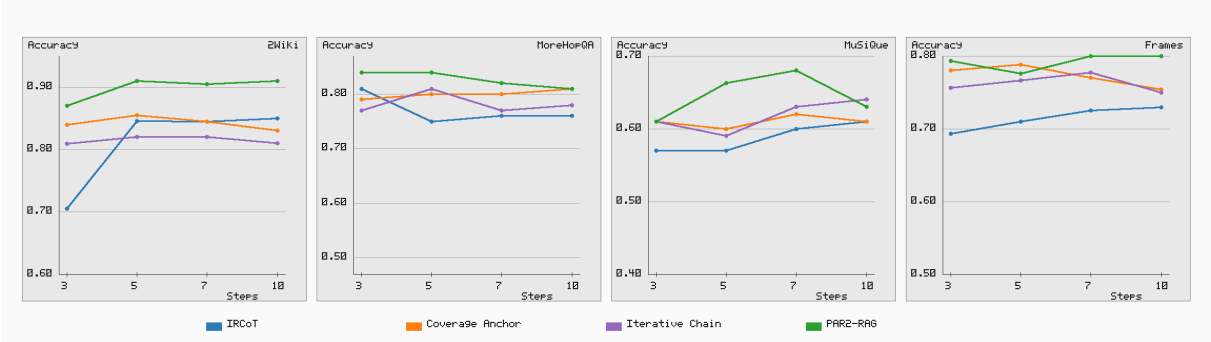

Figure 2: Step robustness results on answer quality, each panel compares IRCoT, Coverage Anchor, Iterative Chain, and PAR^2-RAG across step counts $3, 5, 7, 10$.

Method	MuSiQue	2Wiki	MoreHopQA	FRAMES	Average
OpenAI-GPT-5.2 Reasoning=None					
IRCoT	0.542	0.747	0.698	0.533	0.630
Coverage Anchor	0.603	0.785	0.757	0.707	0.713
Iterative Chain	0.607	0.827	0.782	0.721	0.734
PAR^2-RAG	**0.622**	**0.838**	**0.808**	**0.773**	**0.760**
OpenAI-GPT-5.2 Reasoning=Medium					
IRCoT	0.622	0.807	0.682	0.590	0.675
Coverage Anchor	0.661	0.859	0.845	0.849	0.803
Iterative Chain	0.652	0.879	**0.864**	0.829	0.806
PAR^2-RAG	**0.683**	**0.889**	0.862	**0.857**	**0.823**

Table 4: GPT-5.2 results across MHQA benchmarks with two reasoning configurations.

generation-time reasoning becomes stronger, broader evidence coverage contributes relatively more to final accuracy.

Ablation on Step Robustness Figure 2 shows step/sub-query scaling for all considered methods. The experimental results show the following observations: the performance of all methods generally improves from 3 to 5/7 steps/sub-queries, then saturates or declines at 10 steps on some datasets. We can tell that the proposed PAR^2-RAG remains the strongest overall performance, achieving the highest answer quality score at 12 of 16 benchmark-step points and the highest average performance across all benchmarks.

The results suggests that PAR^2-RAG is robust in terms of maintaining high quality under budget changes, even when deeper search does not help uniformly. Variance-based stability also highlights method differences: IRCoT and Iterative Chain exhibit larger step-wise fluctuations than Coverage Anchor, indicating that retrieval-first anchoring yields more stable behavior under budget variation. Overall, these patterns support the adaptive retrieval control view: additional depth helps only when newly retrieved evidence remains on-path.

5 Conclusions

Finding high-quality evidence across multi-source corpora for multi-hop question answering remains challenging. We proposed Planned Active Retrieval and Reasoning RAG (PAR^2-RAG), a two-stage framework that delays reasoning commitment until sufficient evidence coverage is achieved for MHQA. Across four MHQA benchmarks, PAR^2-RAG is consistently best for answer quality and achieves the strongest retrieval performance. Together with step-depth ablations, these results support a practical design rule for industry MHQA systems: use coverage-first anchoring before deep refinement, and prefer adaptive depth control to fixed large hop budgets.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection.

Eden Biran, Daniela Gottesman, Sohee Yang, Mor Geva, and Amir Globerson. 2024. Hopping too late: Exploring the limitations of large language models on multi-hop queries. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14113–14130.

Jiabei Chen, Guang Liu, Shizhu He, Kun Luo, Yao Xu, Jun Zhao, and Kang Liu. 2025. Search-in-context: Efficient multi-hop QA over long contexts via Monte Carlo tree search with dynamic KV retrieval. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 26443–26455, Vienna, Austria. Association for Computational Linguistics.

Rong Cheng, Jinyi Liu, Yan Zheng, Fei Ni, Jiazhen Du, Hangyu Mao, Fuzheng Zhang, Bo Wang, and Jianye Hao. 2025. Dualrag: A dual-process approach to integrate reasoning and retrieval for multi-hop question answering. *arXiv preprint arXiv:2504.18243*.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. 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.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot learning with retrieval augmented language models. *arXiv preprint arXiv:2208.03299*, 1(2):4.

Olivier Jeunen, Ivan Potapov, and Aleksei Ustimenko. 2024. On (normalised) discounted cumulative gain as an off-policy evaluation metric for top-n recommendation. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 1222–1233.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. 2025. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4745–4759.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Yuankai Li, Jia-Chen Gu, Di Wu, Kai-Wei Chang, and Nanyun Peng. 2024. Brief: Bridging retrieval and inference for multi-hop reasoning via compression. *arXiv preprint arXiv:2410.15277*.

Hao Liu, Zhengren Wang, Xi Chen, Zhiyu Li, Feiyu Xiong, Qinhan Yu, and Wentao Zhang. 2025. Hoprag: Multi-hop reasoning for logic-aware retrieval-augmented generation. *arXiv preprint arXiv:2502.12442*.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

OpenAI. 2025. Openai o3 and o4-mini. https://openai.com/index/introducing-openai-o3-and-o4-mini/. Accessed: 2026-02-08.

Julian Schnitzler, Xanh Ho, Jiahao Huang, Florian Boudin, Saku Sugawara, and Akiko Aizawa. 2024. Morehopqa: More than multi-hop reasoning. *arXiv preprint arXiv:2406.13397*.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec

Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, and 465 others. 2025. Openai gpt-5 system card. *Preprint*, arXiv:2601.03267.

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

Gemini Team and Google DeepMind. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

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.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Rongzhi Zhu, Xiangyu Liu, Zequn Sun, Yiwei Wang, and Wei Hu. 2025. Mitigating lost-in-retrieval problems in retrieval augmented multi-hop question answering. *arXiv preprint arXiv:2502.14245*.

A Appendix

A.1 Dataset

Dataset	Queries	Docs	Avg Text Length
MuSiQue	500	19990	489.52
2Wiki	500	10000	116.07
MoreHopQA	500	9942	132.44
FRAMES	500	9942	132.44
Average	**500**	**13311**	**246.01**

Table 5: Statistics of the multi-hop QA datasets. 'Query Number' and 'Docs Number' refer to the total count of queries and documents, respectively. 'Avg Text Length' is the average word count per document.

A.2 Evaluation Metrics

Correctness Evaluation To further evaluate the generated answer quality beyond string matching, we use a Correctness metric to measure the proportion of generated responses that are judged as correct by an LLM evaluator using a structured prompt template:

$$\text{Correct} = \frac{|\{\, r \mid \text{LLM-Judge}(q, p, g) = \text{“yes”} \,\}|}{|P|} \tag{2}$$

where q denotes the query, g denotes the ground-truth answer, and LLM-Judge(q, p, g) is a language model that evaluates whether the response p correctly answers query q given ground truth g. The evaluation uses GPT-4 with a predefined prompt template in Prompt 3, which outputs binary judgments, with responses containing "yes" (case-insensitive) scored as 1 and others as 0.

NDCG The Normalized Discounted Cumulative Gain (NDCG) metric evaluates the ranking quality of retrieved documents by considering both relevance and position:

$$\text{NDCG@k} = \frac{\text{DCG@k}}{\text{IDCG@k}} \tag{3}$$

where DCG@k is the Discounted Cumulative Gain at rank k, calculated as:

$$\text{DCG@k} = \sum_{i=1}^{k} \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)} \tag{4}$$

and IDCG@k is the ideal DCG@k obtained by sorting all relevant documents in descending order of relevance. Here, rel_i represents the

Prompt for Correctness judgment

You are a helpful research assistant. Your task is to evaluate an LLM's answer against a ground-truth answer and decide whether the ground-truth content is present in the model's response.

Instructions:

1. Carefully compare the *Predicted Answer* with the *Ground-Truth Answer*.
2. Judge based on substance and equivalence of meaning; do not require identical wording unless wording is crucial to meaning.
3. Make a binary decision on whether the vital facts of the ground-truth are contained in the predicted answer.

Input Data:

```
Question: {query}
Predicted Answer: {response}
Ground-Truth Answer: {answer}
```

Output Format:
Provide your final evaluation in the following format:

```
Explanation: <brief rationale for the decision>
Decision: <yes|no>
```

Output:

Figure 3: Prompt for correctness judgment.

relevance score of the document at position i, with binary relevance where documents containing ground-truth context receive a score of 1 and others receive 0.

Retrieval Recall The Retrieval Recall metric measures the proportion of queries for which at least one ground-truth context is successfully retrieved in the top-k results:

$$\text{Recall@k} = \frac{|q \mid \text{GT}_q \cap \text{Retrieved@k}_q \neq \emptyset|}{|Q|} \tag{5}$$

where Q denotes the set of all queries, GT_q represents the set of ground-truth for query q, and Retrieved@k_q represents the set of documents in the top-k retrieved results for query q. Each query receives a binary score of 1 if any ground-truth appears in the retrieval results, and 0 otherwise.

A.3 Prompts

A.4 Implementation Details

All main experiments use OpenSearch TopK=100 and rerank TopK=5. Unless otherwise stated, the step/sub-query budget is 5 and retrieval summary is disabled (`self.nosummary_3`). The retriever and reranker use `openai.o4-mini-2025-04-16`. Main settings are:

- Non-reasoning generation setting: reasoning model `openai.gpt-4.1-2025-04-14`, generation model `openai.gpt-4.1-2025-04-14`.
- Reasoning generation setting: reasoning model `openai.gpt-4.1-2025-04-14`, generation model `openai.o3-2025-04-16`.
- Extension setting: reasoning and generation models `openai.gpt-5.2-2025-12-11` with reasoning mode variations.

A.5 Ablation Details

Step/sub-query without reasoning generation (300-sample study) For PAR^2-RAG, correctness on MuSiQue increases from 0.610

Prompt for Planner

You are a helpful research assistant. Given a query, come up with a set of database searches to perform to best answer the query. Output 5 terms to query for.
Format your response as JSON (No code snippet) with a list of database searches needed to answer the query. Each search should include:

1. **Reason:** A brief explanation of why this search is necessary.
2. **Query:** The exact search term to use.

Example output:

```
{
  "searches": [
    {
      "reason": "Identify the best Caribbean destinations for surfing in April.",
      "query": "best Caribbean surfing spots April"
    },
    {
      "reason": "Find hiking trails in the Caribbean suitable for April vacations.",
      "query": "hiking trails Caribbean April"
    }
  ]
}
```

Figure 4: Prompt for planner.

(3 steps) to 0.680 (7 steps), then drops to 0.630 (10 steps). FRAMES increases from 0.793 (3) to 0.800 (7–10). Similar non-monotonic trends appear for IRCoT, Multi-step RAG, and Query Decomposition.

Step/sub-query with reasoning generation (300-sample study) With reasoning generation, PAR^2-RAG reaches 0.940 on 2Wiki at 7 steps and 0.700 on MuSiQue at 10 steps, while FRAMES peaks at 0.866 at 7 steps. The overall trend remains non-monotonic, reinforcing the need for adaptive stopping.

A.6 Additional Results

GPT-5.2 extension With reasoning mode set to none, PAR^2-RAG reports 0.838 (2Wiki), 0.808 (MoreHopQA), 0.622 (MuSiQue), and 0.773 (FRAMES). With reasoning mode set to medium, PAR^2-RAG reports 0.889, 0.862, 0.683, and 0.857, respectively.

Prompt for Searcher

You are a database search interface. Your ONLY responsibility is to return the FIVE database results exactly as provided, without any modification.

Strict Rules:

1. Output must consist ONLY of the raw database results.
2. Preserve the original wording, spelling, punctuation, capitalization, line breaks, and formatting of the database entries.
3. Do NOT summarize, rephrase, explain, interpret, or add commentary.
4. Do NOT add introductions, conclusions, or transitional text.
5. Do NOT merge, reorder, or alter results beyond their original sequence.
6. If multiple entries are returned, output them in exactly the same order and format as they are given by the database.
7. MUST output all five retrieved entries.

Example (for illustration only — do not generate similar text unless the database returns it):

```
[Document1.txt]
Document1 is an example entry from the database.

[Document2.txt]
Document2 is other entry in the database.

[Document3.txt]
Document3 is other entry in the database.

[Document4.txt]
Document4 is other entry in the database.

[Document5.txt]
Document5 is other entry in the database.
```

Figure 5: Prompt for searcher.

Prompt for Writer

You are a senior researcher tasked with providing a comprehensive answer to a research query. You will be provided with the original query, and initial research done by a research assistant.
DO NOT WRITE A SUMMARY, **directly provide a complete, accurate answer** to the original question.

Output Format
Format your response as **JSON (No code snippet)** with:

- `answer`: An answer that directly addresses the research query using all available evidence.

Example output:

```
{
  "answer": "Based on the research findings, the Caribbean in April offers exceptional
    conditions for outdoor activities..."
}
```

Figure 6: Prompt for Writer.