When to Refine: Dynamic Gates for Adaptive Retrieval in Multi-Hop QA

Anonymous ACL submission

Abstract

Multi-hop question answering (MHQA) challenges Large Language Models (LLMs) by requiring multi-step evidence synthesis, where early retrieval or reasoning mistakes can cascade and where unbounded context growth degrades coherence. Inspired by cognitive theories of working memory and chunking, as well as recent successes that achieve Olympiad-level problem solving through workflow design rather than retraining, we pursue a training-free, model-agnostic solution. We present an **Adaptive Agentic Pipeline for iterative MHQA**. A *RECAP* step compacts the evolving reasoning history to mitigate context overflow, and a *similarity-gated* query-refinement module monitors retrieval signals and intervenes only when context appears suboptimal. In controlled evaluations on MHQA benchmarks and across both proprietary and open-source LLMs, this pipeline yields consistent accuracy gains over a strong iterative baseline under the same model and retrieval/prompting settings, with a practical runtime cost.

1 Related Work

1.1 Iterative Retrieval for Multi-Hop QA

Traditional retrieval-augmented generation (RAG) frameworks typically perform a single retrieval pass based on the initial query (Guu et al., 2020; Karpukhin et al., 2020; Lewis et al., 2020), which is often insufficient for *MHQA* tasks that require progressive evidence gathering and multi-step inference. Iterative or interleaved paradigms address this limitation by decomposing a complex question into sub-questions and alternating reasoning with retrieval. Representative methods include *IRCoT* (Trivedi et al., 2022a) and the *Iterative QA* framework (Fei et al., 2024), as well as synergy-style loops that coordinate iterative retrieval and generation (Shao et al., 2023). These approaches extend stepwise reasoning trends such as chain-of-thought and least-to-most decomposition (Wei et al., 2022; Zhou et al., 2022). While these pipelines improve multi-hop performance, they introduce a cognitive limitation: as intermediate evidence accumulates, the model's context can exceed its effective working capacity, causing *context overflow* and loss of coherence (Laban et al., 2025). Our work adopts the strong iterative pipeline of Fei et al. (2024) and augments it with a *RECAP* mechanism inspired by chunking and coherence management (Sweller, 1988), yielding a compacted reasoning history that mitigates uncontrolled context growth.

1.2 Adaptive Reasoning and the Cost of Complexity

To further enhance robustness, recent research has explored adaptive controllers and search-based reasoning frameworks. Tree-style exploration methods (e.g., Tree-of-Thoughts) and Monte-Carlo Tree Search (MCTS) explicitly branch and evaluate multiple reasoning trajectories (Yao et al., 2023; Lee et al., 2024). Broader workflow-based systems achieve strong performance through structured plan–solve prompting without additional fine-tuning (Huang and Yang, 2025; Wang et al., 2023; Park et al., 2023). However, these approaches incur substantial computational overhead (e.g., latency, token usage, branching) and can amplify early retrieval errors if branching is guided by low-quality context. A complementary line of work introduces self-checking or critique mechanisms (Madaan et al., 2023; Gou et al., 2023), but these typically evaluate generated text rather than retrieval quality and may still propagate upstream errors. In contrast, we focus on *when* to intervene within an otherwise simple iterative loop. Our controller remains lightweight—requiring no search or retraining—while selectively correcting the information-gathering trajectory at the point where quality degradation first arises.

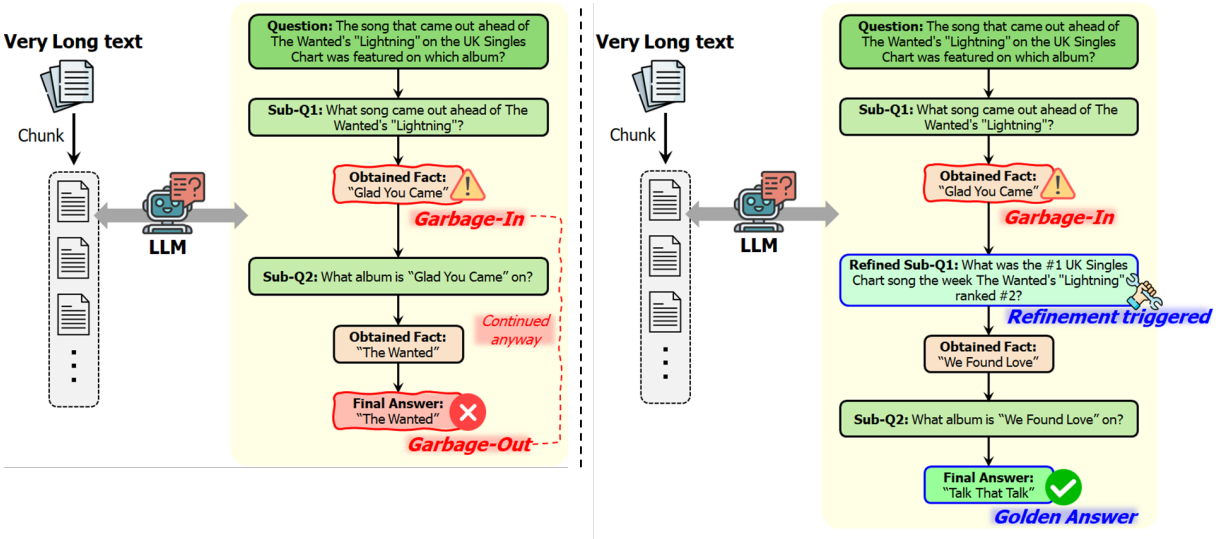

Figure 1: Overview of the proposed adaptive pipeline, illustrated with an example from the HotpotQA dataset. The task is to identify the album of the song that ranked higher than The Wanted's "Lightning". **Left (Standard Iterative Pipeline):** The non-adaptive baseline performs an initial off-target retrieval (*Garbage-In*), incorrectly identifying another song by the same artist ("Glad You Came"). Without a mechanism to detect or correct this error, the pipeline proceeds with the flawed context, which derails the reasoning trajectory and results in an incorrect final answer (*Garbage-Out*). **Right (Proposed Adaptive Pipeline):** The adaptive system detects the poor retrieval quality through its internal **Similarity Gate**, which triggers an autonomous **Query Refinement** process that formulates a more precise sub-question. This targeted self-correction enables the model to recover the correct reasoning path and successfully reach the gold answer.

1.3 Adaptive Control for Retrieval Quality

Another strand of work improves retrieval *inputs* adaptively. Several approaches rewrite queries or trigger retrieval conditioned on internal signals (Gao et al., 2023; Jiang et al., 2023; Ma et al., 2023). Learned controllers such as *Self-RAG* jointly optimize critique and retrieval through training objectives (Asai et al., 2023), and reinforcement-learning methods refine the retrieval–reasoning loop end-to-end (Jin et al., 2025). Although effective, these systems typically require task-specific training, model instrumentation, or specialized tokenization schemes (Schick et al., 2023). Our *Similarity-Gated Query Refinement* instead provides a training-free, model-agnostic controller that uses simple retrieval similarity statistics as uncertainty signals to decide when to refine a sub-query. This design is orthogonal to (and potentially complementary with) stronger multi-hop retrievers such as end-to-end beam retrieval (Zhang et al., 2023), and related to uncertainty-aware triggering explored in active retrieval (Jiang et al., 2023). Whereas prior work optimizes the retriever or learns an explicit policy, we keep the retriever fixed and intervene conditionally to prevent early *GIGO* cascades from suboptimal evidence. Finally, iterative interleaving frameworks such as *ReAct* (Yao et al., 2022) and *IRCoT* (Trivedi et al., 2022a) provide strong general baselines coupling reasoning and evidence acquisition. Our contribution adds a minimal adaptive component to such loops, explicitly addressing the question of *when to refine* using retrieval-side signals rather than expanding the search space or requiring additional training.

2 Methodology

Our methodology proceeds in two stages. First, we establish a strong baseline by augmenting an iterative QA framework with a cognitively inspired context-stabilization mechanism. We then introduce a lightweight adaptive controller that integrates into this baseline to proactively mitigate retrieval errors through conditional query refinement.

2.1 Baseline: A Robust Iterative QA Pipeline with RECAP

We build on the *Iterative QA* paradigm (Fei et al., 2024), which models MHQA as a sequence of sub-question generation, retrieval, and factual extraction. Let Q denote the main question. At each step t, the LLM proposes a sub-question q_t, a retriever R returns top-k passages E_t, and the LLM extracts a fact a_t from (q_t, E_t).

Cognitive challenges. As the loop progresses, intermediate notes and evidence accumulate from diverse contexts, risking loss of topical focus and coherence (*context overflow*). Early retrieval or generation errors can propagate (*GIGO*), reflecting the limits of human working memory and motivating *chunking*-style compaction (Miller, 1956; Baddeley and Hitch, 1974; Sweller, 1988).

RECAP as context stabilization. We maintain an *ordered* reasoning history H and append each (q_t, a_t) pair as textual entries. After the iterative loop terminates (via [STOP] or a step limit T), a single **RECAP** step summarizes the entire H into a compact, coherent synopsis S, which serves as the final context for answer generation. This "working-memory refresh" compacts dispersed notes into a stable representation that focuses attention and mitigates uncontrolled context growth.

Implementation notes. In our implementation, R is a sentence-transformer retriever (BAAI/bge-base-en-v1.5) using cosine similarity, and documents are chunked into ∼80-word spans. Unless stated otherwise, we use an initial k=10 for retrieval during the iterative loop and generate the final answer conditioned on the summarized representation S.

Algorithm 1 Iterative QA with Post-hoc RECAP (Strong Baseline)

Input: main question Q, retriever R, LLM L
Initialize ordered history $H \leftarrow$ "Main Question: " $||\, Q$
for $t = 1$ to T **do**
  Plan: $q_t \leftarrow L(Q, H)$ ▷ sub-question proposal; [STOP] allowed
  if q_t is [STOP] **then break**
  end if
  Retrieve: $E_t \leftarrow R(q_t, \text{top-}k{=}10)$
  Extract fact: $a_t \leftarrow L(q_t, E_t, H)$
  Update (order-preserving): $H \leftarrow H \,||\, [$"Sub-question: "q_t; "Answer: "$a_t]$
end for
RECAP (history-level): $S \leftarrow L(H)$
Final answer: $A_{\text{final}} \leftarrow L(Q, S)$
Output: A_{final}

2.2 Proposed Method: Adaptive Pipeline with Gated Query Refinement

The RECAP-enhanced baseline still follows a fixed linear workflow and does not detect suboptimal retrievals early. To address this limitation, we introduce a *Similarity-Gated Query Refinement* module inspired by metacognitive monitoring and attentional control (Nelson, 1990). The controller embeds self-monitoring into each step and intervenes only when retrieval similarity signals indicate poor context quality.

Similarity gate. At step t, after retrieving an initial set of k passages for q_t, let $S_t = \{s_i\}_{i=1}^{k}$ denote the cosine similarities between q_t and the retrieved chunks. We adopt a robust location-based criterion to determine whether the retrieval is suboptimal:

$$\text{Refine if } \ \bar{s} < \text{median}(S_t) + \delta,$$

where $\bar{s}$ is the mean of S_t, and δ is a small offset (empirically $\delta \approx 0.05$). The median term stabilizes the decision against outliers, while the offset provides a simple safety margin. This criterion flags retrievals whose similarity profile is both low and dispersed, signaling reduced contextual coherence.

Refinement action and consistency. When the gate is activated, the LLM refines q_t to q'_t conditioned on (q_t, E_t), and the retriever re-queries a narrower set (top-3) using q'_t. The fact extraction then conditions on the original sub-question q_t but uses the refined evidence E'_t. This design preserves the semantic intent of the planned question while improving evidence quality, preventing query drift and maintaining consistency across steps.

Overhead and integration. The controller introduces at most one refinement attempt per iteration: one additional LLM call for generating q'_t and one retriever call for obtaining E'_t. When the gate does not trigger, the loop proceeds without any extra cost. The final answer is derived from the RECAP summary S, ensuring full compatibility with the baseline pipeline.

Interfaces and prompts. Distinct prompt templates are used for each stage of the pipeline, including sub-question planning, fact extraction, history summarization, final answering, and query refinement. The complete prompt specifications are listed in the Appendix.

3 Experiments

3.1 Experimental Setup

Datasets. Experiments are conducted on three challenging full-wiki *MHQA* datasets from LongBench (Bai et al., 2023), each probing a distinct

Algorithm 2 Proposed Adaptive Pipeline with Similarity-Gated Refinement

Input: Q, R, L; step cap T; initial k=10; narrow k'=3; offset δ
Initialize ordered H
for $t = 1$ to T **do**
  $q_t \leftarrow L(Q, H)$;
  if q_t=[STOP] **then break**
  end if
  $E_t, S_t \leftarrow R(q_t, \text{top-}k{=}10)$ ▷ also collect similarity scores S_t
  $\theta \leftarrow \text{median}(S_t) + \delta$; **if** $\text{mean}(S_t){<}\theta$ **then**
    $q'_t \leftarrow L(q_t, E_t)$; $E'_t \leftarrow R(q'_t, \text{top-}k'{=}3)$ ▷ targeted re-query
    $E_t \leftarrow E'_t$ ▷ use refined evidence thereafter
  end if
  $a_t \leftarrow L(q_t, E_t, H)$ ▷ extract using *original* q_t with improved evidence
  $H \leftarrow H \,||\, [\text{"Sub-question: "}q_t\text{; "Answer: "}a_t]$
end for
$S \leftarrow L(H)$; $A_{\text{final}} \leftarrow L(Q, S)$
Output: A_{final}

aspect of multi-hop reasoning and retrieval robustness:

- **HotpotQA** (Yang et al., 2018): Predominantly 2-hop questions requiring synthesis across two supporting documents, assessing basic multi-step retrieval and reasoning.
- **2WikiMultihopQA** (Ho et al., 2020): Questions span 2–5 hops, testing coherence maintenance as hop count increases and providing a rigorous evaluation of adaptive refinement.
- **MuSiQue** (Trivedi et al., 2022b): Designed to minimize lexical shortcuts and enforce deeper inference over up to 4 hops, stressing reasoning coherence, attention control, and retrieval precision.

Rationale for benchmark selection. Public full-wiki *MHQA* corpora with standard evaluation scripts and widely reported baselines are selected for three reasons: (i) open-domain full-wiki settings stress retrieval under long-tailed evidence, making early errors consequential and suitable for testing *when-to-refine* controllers; (ii) the three datasets represent complementary difficulty dimensions (2-hop composition in HotpotQA; increased hop count and coherence pressure in 2Wiki; shortcut-resistant, deeper inference in MuSiQue); and (iii) their prevalence in prior work enables transparent comparison and reproducible evaluation protocols.

Evaluation metrics. Performance is measured using token-level **F1** and **average wall-clock inference time** (seconds per question) to capture both accuracy and computational efficiency.

Rationale for using F1. Token-level **F1** is used as the primary metric because multi-hop answers frequently exhibit legitimate surface variants (e.g., paraphrases, punctuation, casing). Exact-match metrics can be brittle, whereas F1 captures partial correctness and aligns more closely with human judgment on compositional answers. Efficiency is assessed via average runtime to quantify the accuracy–cost trade-off.

Implementation details. All methods are evaluated under identical retrieval and prompting configurations. We consider `gpt-4o-mini`, `gpt-4.1-mini`, `gpt-4.1-nano`, and an open-source model (*Llama 3.1–8B*) to assess generality across model families. The retriever is `BAAI/bge-base-en-v1.5` with cosine similarity over ∼80-word chunks; initial retrieval uses top-k=10. The gating controller employs a single margin parameter δ, fixed at δ=0.05 for all experiments.

Compared methods. Comparisons are structured to highlight incremental contributions:

- **Foundational methods:** *Standard RAG*; the original *Iterative QA* (Fei et al., 2024); and *Iterative QA + SNOWBALL* (Laban et al., 2025), which re-inserts the entire growing history at each iteration.
- **Strong baseline (Iterative QA + RECAP):** The baseline augments Iterative QA with a *post-hoc history-level RECAP* that summarizes the ordered reasoning history before final answering, improving stability and coherence.

3.2 Results and Analysis

3.2.1 Establishing a Strong Baseline.

Table 1 summarizes baseline performance. The *Iterative QA + RECAP* pipeline consistently outperforms foundational methods across all three

datasets. The *post-hoc, history-level* RECAP compacts the ordered reasoning history before final answering, mitigating uncontrolled context growth and improving coherence relative to both the raw iterative pipeline and the SNOWBALL variant. These empirical results support the cognitive hypothesis that structured compaction of history stabilizes multi-step reasoning. Accordingly, *Iterative QA + RECAP* is adopted as the strong baseline for subsequent adaptive enhancements.

Table 1: Baseline comparison under identical retrieval/prompting settings (`gpt-4o-mini`). F1 in percentage points.

Method	HotpotQA	2Wiki	MuSiQue	Avg.
Foundational Methods				
Standard RAG	12.0	12.0	9.0	11.0
Iterative QA	43.8	57.3	41.6	47.6
Iterative QA + SNOWBALL	44.6	48.2	44.4	45.7
Strong Baseline				
Iterative QA + RECAP	**50.7**	**58.5**	**45.2**	**51.5**

3.2.2 Main Results: Adaptive Refinement Enhances Performance Across Models.

Table 2: Main results: strong baseline vs. adaptive pipeline across models. F1 with absolute gains in parentheses; identical retrieval/prompting settings for all rows.

Model	Method	HotpotQA	2Wiki	MuSiQue
`gpt-4o-mini`	Baseline	50.7	58.5	45.2
	Adaptive	**56.2 (+5.5)**	**65.2 (+6.7)**	**52.4 (+7.2)**
`gpt-4.1-mini`	Baseline	51.8	60.1	46.3
	Adaptive	**55.2 (+3.4)**	**65.1 (+4.9)**	**51.3 (+5.0)**
`gpt-4.1-nano`	Baseline	50.2	57.4	46.5
	Adaptive	**53.9 (+3.7)**	**62.6 (+5.2)**	**53.4 (+6.8)**
`Llama-3.1-8B`	Baseline	48.1	39.8	28.9
	Adaptive	**50.2 (+2.1)**	**41.9 (+2.1)**	**30.1 (+1.2)**

Table 2 summarizes the main results. Incorporating the *Similarity-Gated Query Refinement* yields consistent gains over the strong baseline across all datasets and model variants (`gpt-4o-mini`, `gpt-4.1-mini`, `gpt-4.1-nano`). The observed improvements span MHQA benchmarks with diverse characteristics (different hop counts and degrees of lexical shortcut resistance), indicating that the gating-and-refinement mechanism remains effective under varying multi-hop conditions. All comparisons are performed under identical retrieval and prompting settings.

Open-source model (Llama 3.1–8B). Under the same experimental configuration, the open-source *Llama 3.1–8B* also benefits from the adaptive pipeline. The adaptive method increases F1 by +2.1, +2.1, and +1.2 percentage points on HotpotQA, 2Wiki, and MuSiQue, respectively (Table 2). Although the magnitude of improvement is smaller than that of the GPT-family models, the results demonstrate that the gating-and-refinement mechanism provides measurable benefits beyond a single model family. The reduced gains may be due to backbone-dependent generation quality, which influences the precision of refined sub

3.2.3 Efficiency and Gating Mechanism Analysis.

Table 3: Average inference times (seconds per question) for `gpt-4o-mini`. The adaptive pipeline incurs moderate additional runtime due to conditional refinements.

Method	HotpotQA	2Wiki	MuSiQue
Standard RAG	1.55s	1.56s	1.80s
Iterative QA	8.53s	7.85s	8.39s
Iterative QA + SNOWBALL	10.32s	7.79s	7.91s
Strong Baseline			
Iterative QA + RECAP	11.19s	10.05s	8.91s
Adaptive Pipeline (Ours)			
+ Gated Refinement	**13.95s**	**11.66s**	**12.31s**

Table 3 reports the runtime analysis. The substantial accuracy improvements of the adaptive pipeline are achieved with only a modest increase in computational cost. Average inference time rises moderately relative to baseline methods, as the *Similarity-Gated Query Refinement* is invoked conditionally—only when retrieval similarity profiles indicate low contextual coherence—thereby avoiding unnecessary computation. This balance between accuracy gain and runtime overhead demonstrates the practicality of the lightweight, training-free design.

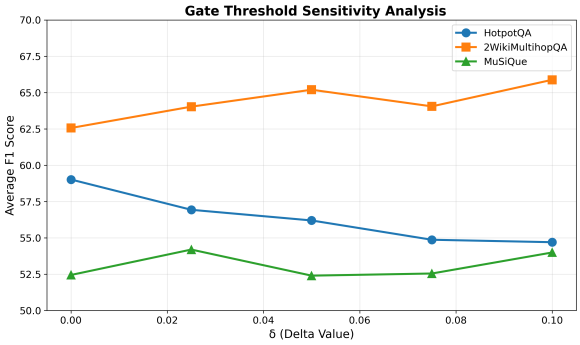

Figure 2: Development-set sensitivity sweep for the gate margin δ. The plotted curves show end-task F1 as δ ranges from 0.0 to 0.1; the empirical sweep supports selecting $\delta = 0.05$ as a pragmatic trade-off across datasets.

Choice of the gate margin δ. A single global gate margin of $\delta = 0.05$ is used for all experiments, determined from a development-set sensitivity sweep. Figure 2 plots end-task F1 as δ varies between 0.0 and 0.1; this empirical analysis forms the basis for selecting the default value. The gating rule and margin choice are treated as pragmatic heuristics, with the following clarifications:

- **Median–mean margin test.** The gate fires when the mean similarity is sufficiently lower than the median similarity:

$$\text{mean}(S) < \text{median}(S) + \delta.$$

  This median–mean margin flags retrievals whose similarity profiles are generally low or broadly dispersed—situations in which retrieved evidence is less likely to be coherently on-topic. Using the median as the reference reduces sensitivity to a few very high-scoring passages that could artificially raise the mean.

- **Role of δ.** The margin δ acts as a small buffer to avoid responding to trivial sampling noise or negligible fluctuations. In practice, $\delta = 0.05$ provided a pragmatic balance: it yields competitive F1 across datasets while preventing excessive refinement calls that would materially increase runtime.

- **Heuristic, not a formal statistical test.** We emphasize that this rule is a lightweight heuristic for signaling retrieval uncertainty. Alternative statistics (e.g., dispersion measures, entropy, or margin-based scores) are plausible and constitute promising directions for future work.

- **Practicality and generality.** We intentionally fix a single global δ across datasets and models to avoid per-dataset tuning and the attendant risk of overfitting to particular development/test splits; this choice favors reproducibility and practical deployment.

In short, $\delta = 0.05$ should be read as a pragmatic safety margin informed by the empirical sweep in Fig. 2, not as a theoretically derived threshold.

3.2.4 Ablation Study on Gating Condition.

Table 4: Ablation results comparing the Baseline (A), Adaptive Gate (B), and Gate-Always-On (B-AO) using `gpt-4o-mini`.

Method	Dataset	Avg F1	Avg Time (s)
A (Baseline)	HotpotQA	51.4	14.97
B (Adaptive Gate)	HotpotQA	**54.7**	**18.65**
B-AO (Gate-Always-On)	HotpotQA	55.6	19.30
A (Baseline)	2Wiki	59.0	10.55
B (Adaptive Gate)	2Wiki	**65.2**	**14.36**
B-AO (Gate-Always-On)	2Wiki	68.2	18.48
A (Baseline)	MuSiQue	47.0	11.33
B (Adaptive Gate)	MuSiQue	**53.8**	**13.05**
B-AO (Gate-Always-On)	MuSiQue	52.8	13.35

To further assess the adaptive logic of the gating mechanism, the conditional refinement approach (**B: Adaptive Gate**) is compared with a variant that unconditionally triggers refinement (**B-AO: Gate-Always-On**). Table 4 presents the results. The Gate-Always-On variant attains slightly higher F1 on simpler datasets but incurs greater computational cost and fails to yield consistent gains on more difficult benchmarks such as MuSiQue. In contrast, the adaptive gate retains most of the accuracy improvement while maintaining lower runtime overhead, indicating that selective intervention effectively balances accuracy and efficiency in a lightweight, retrieval-driven pipeline.

4 Conclusion

We addressed error propagation in iterative multi-hop question answering by introducing a lightweight, training-free adaptive controller that activates only when retrieval quality is low. Building on a strong iterative baseline augmented with a history-level **RECAP** step to stabilize context, we proposed a **Similarity-Gated Query Refinement** module that monitors retrieval similarity statistics and conditionally reformulates sub-queries when needed.

Evaluations across multiple MHQA benchmarks and model families—including proprietary GPT models and an open-source Llama variant—show that the adaptive pipeline achieves consistent accuracy gains under identical retrieval and prompting settings, with only modest runtime overhead from conditional refinements. A development-set sensitivity analysis informed a single global margin ($\delta = 0.05$), ensuring stable performance without per-dataset tuning.

These results highlight a general design principle of *minimal, efficiency-aware intervention*: targeted corrections to retrieval can substantially mitigate cascading errors in multi-step reasoning. By combining cognitively motivated history compaction with selective gate-driven control, the method enhances robustness in existing iterative frameworks without retraining or heavy search, offering a practical and portable approach to improving multi-hop question answering.

Limitations

Our study focuses on a lightweight, training-free controller evaluated in full-wiki MHQA settings, and several boundaries of scope remain. First, although we report consistent gains across multiple GPT-family models and an open-source `Llama 3.1-8B` configuration, broader architectural and scale coverage is limited; verifying behavior on stronger or differently trained open models and larger parameter counts is an important next step. Second, the gating rule is a median–mean margin heuristic with a single global margin (δ=0.05) chosen via a development-set sweep; while this worked robustly in our experiments, we do not claim theoretical optimality or provide per-step trigger-rate/correlation analyses, and alternatives such as dispersion- or entropy-based signals or learned calibration may yield finer control. Third, our efficiency accounting reports wall-clock latency under identical retrieval/prompting settings and shows practical overhead from conditional refinements, but we do not include budget-matched comparisons against heavyweight search/planning pipelines or a detailed token/latency breakdown across hardware and seeds. Finally, task generality beyond MHQA is not established here: portability to other multi-step domains (e.g., complex numerical or procedural reasoning, interactive settings, domain-specific or cross-lingual corpora) remains to be validated, and our experiments fix a single set of prompts, so robustness to prompt template variation is only indirectly evidenced by model and dataset diversity rather than by an explicit prompt-sensitivity study.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Self-reflective retrieval augmented generation. In *NeurIPS 2023 workshop on instruction tuning and instruction following*.

Alan D Baddeley and Graham Hitch. 1974. Working memory. In *Psychology of Learning and Motivation*, volume 8, pages 47–89. Elsevier.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, and 1 others. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Weizhi Fei, Xueyan Niu, Guoqing Xie, Yanhua Zhang, Bo Bai, Lei Deng, and Wei Han. 2024. Retrieval meets reasoning: Dynamic in-context editing for long-text understanding. *arXiv preprint arXiv:2406.12331*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2023. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*.

Kelvin Guu, Kenton Lee, Zora Tung, and et al. 2020. Realm: Retrieval-augmented language model pre-training. In *ICML*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Yichen Huang and Lin F Yang. 2025. Gemini 2.5 pro capable of winning gold at imo 2025. *arXiv preprint arXiv:2507.15855*.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei

Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, and et al. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP*.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2025. Llms get lost in multi-turn conversation. *arXiv preprint arXiv:2505.06120*.

Seongmin Lee, Jaewook Shin, Youngjin Ahn, Seokin Seo, Ohjoon Kwon, and Kee-Eung Kim. 2024. Zero-shot multi-hop question answering via monte-carlo tree search with large language models. *arXiv preprint arXiv:2409.19382*.

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.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

George A Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81.

Thomas O Nelson. 1990. Metamemory: A theoretical framework and new findings. In *Psychology of learning and motivation*, volume 26, pages 125–173. Elsevier.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. *arXiv preprint arXiv:2305.15294*.

John Sweller. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive science*, 12(2):257–285.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022a. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

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

Lei Wang, Wanyu Xu, Yihuai Lan, and et al. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*.

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.

Zhilin Yang, Wen tau Yih, Xiaodong He, Chris Meek, and Ming-Wei Chang. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*.

Shunyu Yao, Dian Yu, Thomas Griffiths, and et al. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.

Shunyu Yao, Jeffrey Zhao, Thomas Griffiths, and et al. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Yong Liu, and Shen Huang. 2023. End-to-end beam retrieval for multi-hop question answering. *arXiv preprint arXiv:2308.08973*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and 1 others. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

A Prompts and Templates

This section provides the exact, unedited prompt templates used to implement the baseline and proposed methods.

Standard RAG Prompt (01_rag_prompt)

```
Based on the following retrieved context, answer the user's question.

Context:
{retrieved_chunks}

Question: {question}

Answer:
```

Few-Shot Examples for Sub-question Planning (02_few_shot_examples)

```
# Example 1
Reasoning History:
Main Question: What is the birth day of Rainn Wilson's father?

Next Sub-question: Who is Rainn Wilson's father?

# Example 2
Reasoning History:
Main Question: Who is the paternal grandfather of John Smith?

Next Sub-question: Who is the father of John Smith?

# Example 3
Reasoning History:
Main Question: What is the capital city of the country of citizenship of Ivanka Trump's spouse?

Next Sub-question: Who is Ivanka Trump's spouse?
```

Plan Sub-question Prompt (03_plan_subquestion_prompt)

```
You are an expert at breaking down complex multi-hop questions into simpler sub-questions.

Given the reasoning history, generate the next sub-question that would help answer the main question.
If you think we have enough information to answer the main question, return "[STOP]".

{few_shot_examples}

# Current Task
Reasoning History:
{reasoning_history}

Next Sub-question:
```

Extract Fact Prompt (04_extract_fact_prompt)

```
Based on the given context, answer the question with specific factual information.

Context:
{retrieved_context}

Question: {question}

Provide a direct, factual answer based only on the information in the context.
If the context doesn't contain the answer, say "Information not found."

Answer:
```

RECAP Prompt (05_recap_prompt)

(This RECAP prompt is used once after the iterative reasoning loop completes; it summarizes the ordered reasoning history accumulated during the iterative steps.)

```
You are tasked with summarizing the reasoning history for a multi-hop question.

Create a concise summary that captures:
1. The main question
2. All sub-questions that were asked
3. All factual findings discovered so far
4. The logical flow connecting the findings

Reasoning History:
{reasoning_history}

Summary:
```

Final Answer Prompt (06_final_answer_prompt)

```
Based on the summarized reasoning history, provide the final answer to the main question.

Summary:
{recap_summary}

Main Question: {main_question}

Provide a direct, concise answer based on the reasoning chain. Return your response in JSON format:
{"final_answer": "your answer here"}
```

Refine Query Prompt (07_refine_query_prompt)

Note: This prompt is the core of our proposed adaptive mechanism (Method B) and is only triggered by the Similarity Gate.

```
The original query did not retrieve high-quality relevant documents.
Please refine the query to improve search results.

Original Query: {original_query}
Context: This is part of answering: {main_question}

Refined Query:
```

B Case Studies: From Failure to Success

Here we present concrete examples demonstrating how our proposed method (B) corrects reasoning failures encountered by the baseline (A).

Case Study 1: Identifying the Correct Band

Main Question: "For which band was the female member of Fitz and The Tantrums the front woman for ten years?"

Method A (Baseline) — Failure Trace

- **Step 1 (Sub-question):** Who is the female member of Fitz and The Tantrums?
- **Step 2 (Initial Retrieval):** Retrieves general documents about the band, but none specifically mention other bands associated with the female member, Noelle Scaggs.
- **Step 3 (Fact Extraction — Garbage In):** "Noelle Scaggs". While correct, the context lacks the necessary information for the next hop.
- **Step 4 (Next Sub-question):** What band was Noelle Scaggs the front woman for?
- **Step 5 (Fact Extraction):** The model, lacking specific context, hallucinates or returns "Information not found."
- **Final Answer (Incorrect):** Fails to produce the correct answer, "The Rebirth".

Method B (Proposed) — Success Trace

- **Step 1 & 2:** Same as Method A.
- **Step 3 (Similarity Gate Triggered):**
  - *Similarity Scores:* [0.45, 0.42, 0.38, 0.35, 0.33]
  - *Dynamic Threshold:* median(0.38) + 0.05 = 0.43
  - *Decision:* Gate is **activated** because mean(0.386) < 0.43.
- **Step 4 (Query Refinement):**
  - *Original Query:* "Fitz and The Tantrums female member"
  - *Refined Query:* "What band did Noelle Scaggs from Fitz and The Tantrums lead as front woman for ten years"
- **Step 5 (Improved Re-retrieval):** The refined query retrieves documents mentioning her role in "The Rebirth".
- **Step 6 (Fact Extraction — Corrected):** "The Rebirth"
- **Final Answer (Correct):** `{"final_answer": "The Rebirth"}`

Case Study 2: Film Actress Identification

Question: "Who starred in her final film role in the 1964 film directed by the man who also did Vera Cruz and Kiss Me Deadly?"

Method A (Baseline) — FAILURE (F1=0.0) Final Answer: `"Olivia de Havilland"`

- **Failure Analysis:**
  - **Incorrect Final Step:** Made a wrong assumption about which actress had her final role.
  - **Insufficient Information:** Retrieved documents did not clearly specify Mary Astor's final role.
  - **Logical Leap:** Made an incorrect conclusion without proper evidence.

Method B (Proposed) — SUCCESS (F1=1.0) Final Answer: `"Mary Astor"`

- **Success Analysis:**
  - **Gate Triggered:** Detected low-quality information regarding the actors' final roles.
  - **Targeted Refinement:** The query was refined to include all candidate names ("Mary Astor Bette Davis...").
  - **Corrected Understanding:** This targeted query retrieved precise documents confirming Mary Astor's final role.

Case Study 3: Biological Component Identification

Question: "The large subunit and small subunit that use two types of RNA are major components that make up what?"

Method A (Baseline) — FAILURE (F1=0.0) Final Answer: `"The ribosome"`

- **Failure Analysis:**
  - **Format Issue:** Used the singular "ribosome" instead of the required plural "ribosomes".
  - **Precision Error:** Added an unnecessary article "The".

Method B (Proposed) — SUCCESS (F1=1.0) Final Answer: `"Ribosomes"`

- **Success Analysis:**

- **Gate Triggered:** Detected format inconsistency in the retrieved documents.
- **Format-Aware Refinement:** The query was refined to include "plural form", which retrieved documents using the correct format.
- **Perfect Match:** The final answer matched the ground truth format and content perfectly.

Case Study 4: Geographic Location Specification

Question: "What city, located in Lincoln County, Montana, United States, is Vegepet based in?"

Method A (Baseline) — PARTIAL SUCCESS (F1=0.67) Final Answer: `"Troy"`

- **Failure Analysis:**
  - **Incomplete Information:** Found the correct city but missed the required state specification.
  - **Precision Error:** The answer was a partial match but not complete.

Method B (Proposed) — SUCCESS (F1=1.0) Final Answer: `"Troy, Montana"`

- **Success Analysis:**
  - **Gate Triggered:** Detected incomplete geographic specification.
  - **Completeness-Driven Refinement:** The query was refined to ask for the "full address".
  - **Complete Retrieval:** This led to the retrieval of the full "Troy, Montana" string.

Case Study 5: Choreographer Attribution

Question: "Who choreographed one who is said to be the last muse for the choreographer and starred in the Nutcracker?"

Method A (Baseline) — FAILURE (F1=0.0) Final Answer: `"Peter Martins"`

- **Failure Analysis:**
  - **Initial Misidentification:** Confused the original choreographer (George Balanchine) with a later one.
  - **Logic Chain Error:** Built the entire reasoning chain on this wrong foundation.
  - **Attribution Error:** Focused on the production choreographer rather than the original creator.

Method B (Proposed) — SUCCESS (F1=1.0) Final Answer: `"George Balanchine"`

- **Success Analysis:**
  - **Gate Triggered:** Detected an inconsistency in choreographer attribution.
  - **Disambiguation Refinement:** The query was refined to include "original creator" to resolve the ambiguity.
  - **Correct Attribution:** This led to the correct identification of George Balanchine.

C Prompt Statistics and Design Rationale

C.1 Prompt Length Statistics

Table 5: Summary of Prompt Template Characteristics.

Prompt Name	File Name	Tokens (approx.)	Core Purpose
RAG	01_rag_prompt	~35	Basic QA
Few-Shot Examples	02_few_shot_examples	~120	Sub-question examples
Plan Sub-question	03_plan_subquestion_prompt	~95	Sub-question generation
Extract Fact	04_extract_fact_prompt	~75	Fact extraction
RECAP	05_recap_prompt	~80	Summarization
Final Answer	06_final_answer_prompt	~70	Answer synthesis
Refinement	07_refine_query_prompt	~55	**Query improvement**

C.2 Critical Design Decisions

- **Dynamic Threshold (median + δ):** We use the median as a robust central tendency to reduce sensitivity to a few very high-scoring passages; adding a small margin δ helps avoid triggering on minor stochastic fluctuations. In our development settings this dynamic heuristic exhibited stable behavior and was practically easier to calibrate than a single global fixed threshold; broader comparative studies are outside our current scope.
- **Context-Aware Refinement:** Including the main question ({main_question}) in the refinement prompt keeps the reformulation aligned with the overall goal and reduces the risk of over-correction.
- **Structured JSON Output:** Enforcing a JSON output for final answers simplified automated parsing in our runs.