AMRAG: Adaptive Multi-step Retrieval-Augmented Generation for Complex Question Answering

Ning Zhang[1], Xinzhi Wang[1,*], Xiangfeng Luo[1], Jianqi Gao[1]
[1] Shanghai University, China
zhangning0904@shu.edu.cn, wxz2017@shu.edu.cn, luoxf@shu.edu.cn, jianqi_gao@shu.edu.cn

***Abstract*—Retrieval-augmented generation (RAG) is widely recognized as an effective paradigm for mitigating hallucinations in large language models (LLMs). For complex queries requiring multi-step reasoning, iterative retrieval is commonly employed to progressively gather evidence across reasoning steps. However, indiscriminate retrieval at every step not only introduces irrelevant or misleading information that impairs downstream reasoning, but also incurs substantial computational overhead. We address these challenges by leveraging a well-established observation: LLMs exhibit stronger factual accuracy on high-frequency knowledge from pretraining data, while low-popularity knowledge is more prone to hallucinations. Building on this insight, we propose AMRAG (Adaptive Multi-step Retrieval-Augmented Generation), a framework that selectively determines retrieval necessity at each reasoning step by estimating the popularity of key entities in the current sub-question. Retrieval is invoked only when the model is unlikely to possess the required knowledge. Retrieved documents are further filtered and reranked by a lightweight semantic reranker to retain only evidence closely aligned with the sub-question. Experiments on HotpotQA and other benchmarks demonstrate that AMRAG outperforms strong baselines while reducing computational costs.**

***Index Terms*—retrieval-augmented generation, large language models, multi-step reasoning, adaptive retrieval, knowledge hallucination.**

I. Introduction

Retrieval-Augmented Generation (RAG) mitigates hallucinations stemming from limited parametric knowledge by incorporating external evidence into the input context of large language models [1], [2]. In complex multi-step reasoning tasks, where information requirements evolve with intermediate reasoning states, existing RAG systems interleave retrieval with generation [3]. However, many current approaches employ fixed retrieval strategies that trigger retrieval at every reasoning step, regardless of whether the required knowledge already resides in the model's parameters [3]–[5]. This indiscriminate retrieval incurs dual costs: when knowledge is already encoded parametrically, it introduces distracting context and unnecessary computational overhead; when external evidence is genuinely needed, retrieved documents may contain redundant or noisy information that compromises reasoning accuracy. These limitations underscore the necessity for adaptive retrieval mechanisms capable of determining when external evidence is required during multi-step reasoning.

A pivotal challenge in developing such adaptive mechanisms lies in identifying reliable signals that indicate retrieval necessity [6]. Prior research demonstrates that LLMs' factual accuracy is strongly influenced by the frequency of relevant knowledge in pretraining data: a model's performance on fact-based questions correlates closely with the prevalence of corresponding documents during pretraining [7]–[10]. Consequently, high-frequency knowledge typically exhibits reliable parametric recall, whereas low-frequency facts present substantially elevated hallucination risk. This systematic asymmetry motivates a selective retrieval strategy wherein retrieval is triggered only when a reasoning step likely exceeds the model's reliable parametric knowledge. We posit that during iterative reasoning, retrieval necessity can be inferred from the popularity of key entities in the current reasoning step [11], [12]. Since reasoning steps are inherently fact-oriented, entity popularity serves as a reliable indicator of whether the requisite knowledge is already parametrically encoded in the model, thereby enabling fine-grained, adaptive retrieval decisions.

Building upon this observation, we propose AMRAG, an adaptive multi-step retrieval-augmented generation framework designed to eliminate unnecessary retrieval in multi-hop reasoning. At each reasoning step, AMRAG determines retrieval necessity based on query-derived signals. Specifically, we extract key entities from the current reasoning context and estimate their popularity using Wikipedia pageviews as a proxy for pretraining data prevalence. This approach leverages the correlation between Wikipedia pageviews and entity frequency in human-generated text, as LLM pretraining data is predominantly sampled from such distributions [13], [14]. This design circumvents reliance on model-internal states or self-assessed uncertainty signals, which are either inaccessible in black-box LLMs or introduce substantial inference overhead [15], [16]. To further refine retrieved evidence, we employ a semantic reranker to prioritize documents most relevant to the current sub-question, ensuring that only semantically aligned evidence informs the reasoning process.

In this paper, we make the following contributions:

- We propose AMRAG, an adaptive multi-step retrieval-augmented generation framework for complex question answering, which performs sub-question–level retrieval control to avoid unnecessary evidence acquisition.
- We empirically demonstrate that AMRAG improves multi-hop question answering performance on standard benchmarks while substantially reducing redundant retrieval via entity popularity–guided decisions, and achieves robust reasoning accuracy across varying rea-

soning depths and retrieval thresholds, with ablation studies validating the effectiveness of adaptive retrieval and document reranking.

II. Related Work

A. Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) has emerged as a mainstream technique for enhancing the factual accuracy and reliability of large language models (LLMs) by integrating external non-parametric knowledge bases [1]. A standard RAG system comprises an external retrieval module and a generative language model, designed to combine parametric knowledge with non-parametric evidence to augment question-answering capabilities. Upon receiving a query, the system first retrieves and extracts the most relevant knowledge fragments from large corpora. Subsequently, the generative model synthesizes and interprets this evidence to formulate the final response. The core value of this mechanism lies in its effective mitigation of hallucination issues stemming from the limited parametric knowledge of LLMs, while empowering models to rapidly adapt and update external knowledge.

B. Multi-Step Reasoning

Multi-step reasoning refers to answering questions via a step-by-step combination of multiple pieces of evidence, rather than relying on a single supporting fact. For instance, the Self-Ask framework decomposes complex problems into searchable sequences of subproblems [4], while Least-to-Most Prompting employs contextual examples to guide incremental reasoning [17]. While these methods offer strong interpretability, their decomposition processes are typically statically generated based on initial queries, making it difficult to dynamically adjust retrieval targets during reasoning based on acquired intermediate evidence. To address this, the IRCoT framework interleaves retrieval with chain-of-reasoning generation, enabling dynamic tracking of retrieval paths through reasoning [3]. Recent research further focuses on enhancing the adaptability of decomposition and retrieval processes. For instance, RQ-RAG directly optimizes subquery generation through a differentiable query rewriting module [18]; IterDRAG theoretically demonstrates the linear scaling of computational complexity with performance when retrieval and reasoning alternate, providing quantifiable performance guarantees for multi-step reasoning [19]; and GenDec attempts to mitigate error propagation through decoupled decomposition mechanisms [20]. These approaches have achieved significant progress in optimizing subproblem generation, enhancing computational efficiency, and strengthening robustness [21]. However, existing methods lack mechanisms to selectively trigger retrieval, creating a critical bottleneck in multi-step reasoning.

C. Adaptive Retrieval-Augmented Generation

Adaptive Retrieval-Augmented Generation aims to improve the fixed retrieval patterns in traditional retrieval-augmented generation frameworks. Its core lies in enabling systems to dynamically decide when and what to retrieve during answer generation. Self-RAG trains models to generate special control tokens that autonomously determine retrieval timing and critically evaluate retrieved content [22]. Adapt-LLM extends this paradigm by training models to generate a dedicated token for deciding when to retrieve, focusing on end-to-end learning of the retrieval decision itself [23]. FLARE proactively triggers retrieval when model-generated tokens exhibit low confidence, enabling verification and rewriting of subsequent content [24]. Similarly, SEAKR dynamically triggers retrieval and document reordering by monitoring token-level uncertainty during decoding [25]. In contrast, CtrlA introduces a control framework based on representation perspective, leveraging internal probing of the LLM's hidden states to make retrieval decisions [26]. Adaptive-RAG introduces a termination prediction-based mechanism enabling models to determine when to halt during multi-turn retrieval [27]. DRAGIN explicitly models the information needs of LLMs by dynamically triggering retrieval based on the current generation state, enabling fine-grained control over retrieval timing [28]. PLANxRAG guides retrieval steps by constructing explicit reasoning plans for complex queries [29].

III. Method

A. Iterative Controller

AMRAG employs an iterative reasoning process that progressively uncovers multi-hop knowledge dependencies within complex queries. This process is achieved through multiple rounds of alternating question decomposition and answer generation. At step i, the system directs the large language model to formulate a new sub-question $Q_i = \mathbf{LLM}(Q, Q_{1:i-1}, A_{1:i-1})$ that exhibits explicit factual directionality. This directionality is predicated on the original question Q, the generated sequence of sub-questions $Q_{1:i-1} = \{Q_1, \ldots, Q_{i-1}\}$, and the corresponding sequence of sub-answers $A_{1:i-1} = \{A_1, \ldots, A_{i-1}\}$.

This sub-question does not constitute a mere restatement of the original query; rather, it is dynamically generated based on the current reasoning trajectory. The purpose of the latter is to explicitly reveal the knowledge requirements that remain unsatisfied in the current state, thereby providing fine-grained query targets for subsequent retrieval. Subsequently, the system determines whether to introduce external knowledge based on an adaptive retrieval strategy. If retrieval is triggered, the system treats Q_i as a query, retrieves relevant documents D_i, which are then refined by a document reranker based on their semantic consistency with the current sub-question. Otherwise, the D_i is left empty. In light of the aforementioned data, the system once more activates the large language model in order to generate an intermediate response $A_i = \mathbf{LLM}(Q_i, D_i)$ by combining the sub-question and potential evidence.

B. Adaptive Retrieval Controller

The adaptive retrieval controller is a lightweight binary classifier designed to determine whether sub-question Q_i requires external knowledge. It is based on the assumption that the capacity of LLMs to memorize factual knowledge

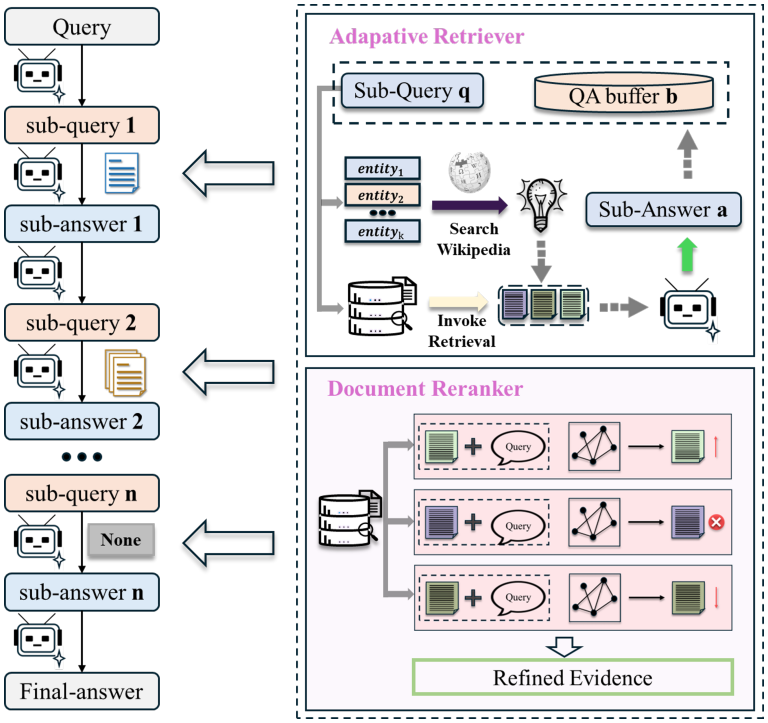

Fig. 1. Overview of AMRAG, illustrating an iterative reasoning framework with adaptive retrieval decisions that selectively incorporate external evidence, followed by document reranking and answer generation.

about entities strongly correlates with their frequency in the pre-training corpus [7]. Facts about high-frequency entities are often answerable without retrieval, whereas questions involving low-frequency entities are more likely to require external knowledge. As this decision is based solely on the popularity of entities that are accessible externally and does not involve the internal representations of language models, no model-specific adjustments are required.

Specifically, let $\mathcal{E}(Q_i)$ denote the set of named entities extracted from sub-question Q_i and $pop(e)$ denote the Wikipedia page views of entity e. The controller is defined as follows:

$$\text{Retrieve}(Q_i) = \mathbb{I}\left(\min_{e \in \mathcal{E}(Q_i)} \text{pop}(e) < \tau\right)$$

where $\mathbb{I}(\cdot)$ denotes the indicator function and τ is a preset threshold. When $Retrieve(Q_i) = 1$, retrieval is performed using Q_i; otherwise, retrieval is skipped and the system proceeds directly to answer generation.

C. Document Reranker

The document reranker aims to ensure that retrieved documents are aligned with the factual target of the current sub-question Q_i. In multi-hop reasoning, each sub-question targets a specific fact. However, documents retrieved at the initial stage are often only coarsely relevant and may include homonymous entities or peripheral context, which can introduce factual noise into subsequent reasoning. We apply a cross-encoder–based reranking model to the retrieved candidates, encoding each document–sub-question pair with a pre-trained language model to compute a relevance score based on deep interaction representations. The top-k documents are retained as refined evidence for sub-answer generation, mitigating topic drift and entity ambiguity.

IV. Experiments

A. Experiment Setup

We conduct experiments on two multi-hop question answering benchmarks: HotpotQA [30] and 2WikiMultiHopQA [31]. Both datasets require cross-document reasoning over multiple Wikipedia paragraphs under a standard distractor setting, where each question is associated with a set of candidate paragraphs containing both supporting evidence and irrelevant distractors. HotpotQA consists of approximately 113K manually constructed question–answer pairs focusing on chained factual reasoning, while 2WikiMultiHopQA includes 200K automatically generated and human-verified questions involving more complex multi-entity relations, attribute comparisons, and logical reasoning. This setting enables fair comparison with prior methods under an identical retrieval space.

Our framework is built on GPT-3.5-turbo, which is used for sub-question generation, sub-answer generation, and final answer synthesis. External knowledge is strictly limited to the candidate paragraphs provided by each dataset, without introducing additional documents. Initial retrieval ranks candidate paragraphs using BM25. An adaptive retrieval controller extracts named entities from sub-questions using dslim/bert-base-NER and leverages Wikipedia pageviews to decide whether retrieval should be triggered. Retrieved candidates are further reranked by a pre-trained semantic matching model,

retaining the top-k documents for generation. All components operate in a zero-shot setting without fine-tuning.

We evaluate model performance using Exact Match (EM) and F1, the standard metrics for multi-hop question answering.

B. Experiment Results

We compare AMRAG with representative multi-hop question answering methods spanning different reasoning and retrieval paradigms. Prompt-based reasoning approaches, such as Self-Ask and ReAct, employ explicit chain-of-thought decomposition but rely on static retrieval that does not adapt to intermediate reasoning steps. Iterative retrieval-augmented methods, including IRCoT [3], DRAG, IterDRAG [19], and ITER-RETGEN [21], interleave reasoning with step-wise retrieval throughout generation. Decomposition-based models, such as AMBER [32] and DecomP [33], explicitly generate sub-questions to guide multi-hop inference but adopt non-adaptive retrieval strategies. This comparison enables a systematic evaluation of AMRAG across diverse multi-hop reasoning paradigms and highlights the advantages of its adaptive, reasoning-aware retrieval mechanism.

TABLE I
EXPERIMENTAL RESULTS ON 2WIKIMULTIHOPQA AND HOTPOTQA. COMPARING REPRESENTATIVE MULTI-HOP QA METHODS ACROSS DIFFERENT REASONING AND RETRIEVAL PARADIGMS.

Models	2Wiki		HPQA	
	EM	F1	EM	F1
Prompt-based Reasoning				
3-shot	49.0	56.2	45.8	59.4
Self-Ask	37.3	48.8	36.8	55.2
ReAct	28.0	38.5	24.9	44.7
Iterative / Tool-based RAG				
DRAG	45.9	53.7	46.9	60.3
IterDRAG	44.3	54.6	38.3	49.8
ITER-RETGEN	35.5	47.4	45.1	60.4
Decomposition-based Methods				
AMBER	–	52.7	–	53.7
DecomP	–	59.3	–	50.0
AMRAG(Ours)	**58.2**	**64.9**	**49.1**	**62.4**

Note: Results for Self-Ask, ReAct, and ITER-RETGEN are obtained using text-davinci-003. DRAG and IterDRAG are evaluated with Gemini 1.5 Flash. 3-shot prompting is conducted with GPT-4o. AMBER is evaluated using Qwen2-7B. AMRAG is evaluated using GPT-3.5-turbo under the same experimental setting.

Results in Table I show that AMRAG achieves consistently strong performance on both 2WikiMultiHopQA and HotpotQA, attaining the highest or competitive EM and F1 scores among all compared methods. Compared with iterative retrieval-based approaches that retrieve at every reasoning step, AMRAG yields higher accuracy, suggesting that indiscriminate step-wise retrieval can introduce noise and hinder reasoning. AMRAG also outperforms decomposition-based methods, indicating that explicit sub-question generation alone is insufficient without effective retrieval control. Despite differences in backbone models and knowledge sources across baselines, AMRAG exhibits a consistent performance advantage, highlighting the effectiveness of its adaptive, reasoning-aware retrieval mechanism in zero-shot multi-hop question answering.

V. Analysis

To quantify the contributions of its core components to performance and efficiency, we conduct a systematic analysis of the AMRAG framework. Following standard practice, we randomly sample examples from the dataset to ensure statistical validity.

A. Ablation Study

We validate the effectiveness of each AMRAG component through ablation experiments, with results shown in Table II.

Ablating Iterative Controller. To assess the role of iterative control, we disable the iterative controller while keeping adaptive retrieval and document reranking intact, forcing the model to answer using the original question in a single step. EM drops sharply from 54.1 to 30.7 on 2WikiMultiHopQA and from 49.1 to 36.5 on HotpotQA, showing that iterative reasoning is crucial for multi-hop QA.

Ablating Adaptive Retrieval. We replace adaptive triggering with a fixed strategy, enforcing retrieval for every generated sub-question. EM declines by 2.8 points on 2Wiki and 0.6 points on HotpotQA. The modest degradation indicates that enforcing retrieval at every step introduces redundant evidence, supporting the advantage of selectively invoking retrieval only when external knowledge is required.

Ablating Document Reranking. In this variant, the system skips semantic reranking and directly uses the top-k BM25 documents as evidence. EM decreases by 3.9 points on 2Wiki and 5.8 points on HotpotQA, indicating that relying solely on keyword matching fails to ensure evidence aligns with sub-question facts, while document reranking substantially improves answer accuracy.

TABLE II
ABLATION STUDY OF AMRAG ON 2WIKIMULTIHOPQA AND HOTPOTQA

Models	2Wiki		HPQA	
	EM	F1	EM	F1
AMRAG (Full Model)	**54.1**	**62.6**	**49.1**	**62.4**
Ablating Iterative Controller (IC)				
AMRAG w/o IC	30.7	39.4	36.5	48.1
Ablating Adaptive Retrieval (AT)				
AMRAG w/o AT	51.3	59.0	48.5	60.3
Ablating Document Reranker (DR)				
AMRAG w/o DR	50.2	57.1	43.3	58.1

B. Hyper-parameter Search

To assess the robustness of AMRAG with respect to its hyperparameters, we analyze the effects of the maximum iteration limit L and the retrieval popularity threshold τ, which respectively control reasoning depth and retrieval triggering.

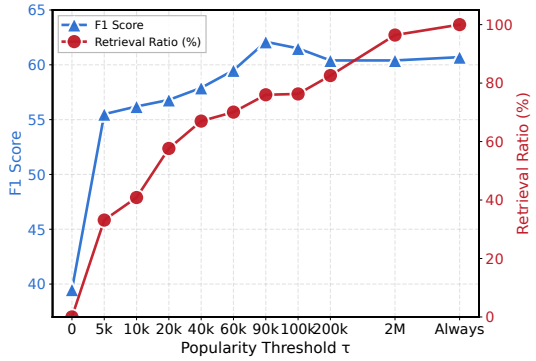

Fig. 2. F1 score (%) and retrieval ratio (%) of AMRAG on HotpotQA under different popularity thresholds τ. The figure shows that F1 performance remains stable across a broad range of thresholds, while adaptive retrieval substantially reduces the number of external documents accessed.

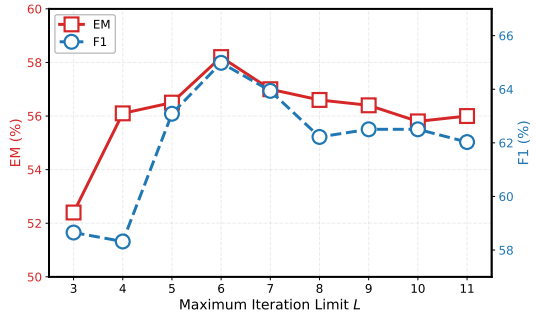

Fig. 3. Effect of the maximum iteration limit L on AMRAG's performance on 2WikiMultiHopQA. EM and F1 are reported across different values of L, showing that performance peaks at a moderate reasoning depth and remains stable thereafter.

As shown in Figure 2, AMRAG maintains high performance across a broad range of popularity thresholds, demonstrating robustness and showing that its effectiveness does not rely on fine-tuned hyperparameters. Within this stable regime, external retrievals are substantially reduced while reasoning accuracy remains high, confirming that adaptive retrieval minimizes redundant evidence without sacrificing performance. These findings validate the design principle that selective, context-aware retrieval suffices for practical deployment.

As shown in Figure 3, this experiment verifies that AMRAG benefits from multi-step reasoning only up to a moderate depth. Small values of L restrict the model's ability to resolve multi-hop dependencies, while increasing L beyond this range does not yield further improvements. This suggests that excessive iterations primarily introduce additional intermediate contexts that are weakly informative for the final answer, rather than contributing new reasoning signals. In practice, setting L to 6–7 achieves a good balance between reasoning coverage and context efficiency.

C. Component Contribution Analysis

To systematically evaluate the impact of the retrieval module on evidence quality, we conduct experiments on the 2WikiMultiHopQA development set. The knowledge base consists of all candidate paragraphs from all samples in the dataset. We used two metrics to assess retrieval effectiveness. Gold Ratio (evidence purity) is defined as the proportion of retrieved documents containing gold paragraphs relied upon by the ground-truth answers, reflecting retrieval accuracy. Gold Coverage (evidence completeness) is defined as the proportion of all gold paragraphs that are successfully retrieved, measuring the system's ability to cover critical information. The maximum number of retrieval passes, $L = 10$, is fixed to control for variables.

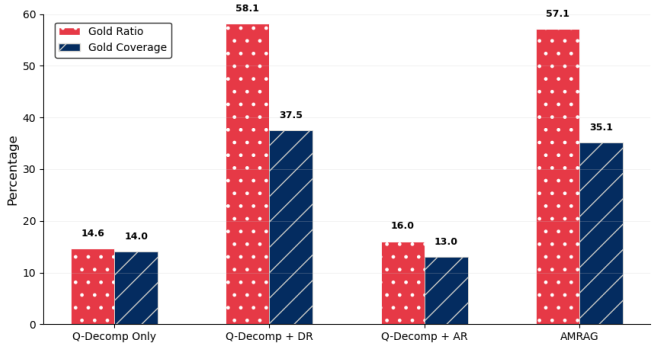

Fig. 4. Gold Ratio and Gold Coverage under different retrieval strategies on 2WikiMultiHopQA. DR substantially improves both metrics, while AR mainly limits irrelevant retrievals. AMRAG combines DR and AR to achieve high purity with good coverage.

As shown in Figure 4, under the Q-Iterate Only baseline, the Gold Ratio is only 14.6%, indicating that unconstrained retrieval retrieves many irrelevant documents. Introducing document reranking (DR) substantially increases the Gold Ratio to 58.1% and improves coverage from 14.0% to 37.5%, indicating that DR prioritizes factually relevant documents over superficially matched ones, thereby improving the utilization of a limited retrieval budget. In contrast, Adaptive Retrieval (AR) has a limited direct effect on retrieval metrics: Q-Iterate + AR achieves a Gold Ratio of 16.0% and coverage of 14.2%, close to the baseline. AR mainly serves to dynamically avoid retrieving clearly irrelevant sub-questions, reducing the injection of harmful noise. Although a few weakly relevant pieces of evidence may be missed, ablation experiments show minimal impact on final question-answering performance (see Section 5.1). Finally, AMRAG, which combines AR and DR, achieves high purity (57.1%) while maintaining good coverage (35.1%), highlighting the synergistic effect of structured retrieval control and semantic reranking.

VI. Conclusion

In this work, we propose AMRAG, an adaptive multi-step retrieval-augmented generation framework for multi-hop question answering. AMRAG coordinates iterative reasoning with adaptive retrieval to selectively incorporate external knowledge. Retrieval is guided by entity popularity signals to avoid unnecessary evidence access, while a lightweight semantic reranker refines retrieved documents for factual relevance. Experiments on HotpotQA and 2WikiMultiHopQA

show that AMRAG consistently outperforms strong baselines in complex multi-hop reasoning, and further analysis confirms the effectiveness of adaptive retrieval in balancing retrieval efficiency and reasoning accuracy.

Beyond empirical improvements, our results highlight that effective multi-step reasoning does not require retrieval at every step, and that retrieval timing can be reliably controlled using simple, model-agnostic signals without access to internal model states. We hope this work encourages further exploration of lightweight and interpretable retrieval control mechanisms for building more efficient and reliable retrieval-augmented generation systems.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.

[2] Y. Zhang, Y. Li, L. Cui, D. Cai, L. Liu, T. Fu, X. Huang, E. Zhao, Y. Zhang, Y. Chen *et al.*, " siren's song in the ai ocean: A survey on hallucination in large language models," *Computational Linguistics*, pp. 1–46, 2025.

[3] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "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)*, 2023, pp. 10 014–10 037.

[4] O. Press, M. Zhang, S. Min, L. Schmidt, N. A. Smith, and M. Lewis, "Measuring and narrowing the compositionality gap in language models," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 5687–5711.

[5] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," in *The eleventh international conference on learning representations*, 2022.

[6] Y. Abbasi Yadkori, I. Kuzborskij, A. György, and C. Szepesvari, "To believe or not to believe your llm: Iterative prompting for estimating epistemic uncertainty," *Advances in Neural Information Processing Systems*, vol. 37, pp. 58 077–58 117, 2024.

[7] A. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, and H. Hajishirzi, "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)*, 2023, pp. 9802–9822.

[8] N. Kandpal, H. Deng, A. Roberts, E. Wallace, and C. Raffel, "Large language models struggle to learn long-tail knowledge," in *International conference on machine learning*. PMLR, 2023, pp. 15 696–15 707.

[9] Y. Razeghi, R. L. Logan IV, M. Gardner, and S. Singh, "Impact of pretraining term frequencies on few-shot reasoning," *arXiv preprint arXiv:2202.07206*, 2022.

[10] R. T. McCoy, S. Yao, D. Friedman, M. D. Hardy, and T. L. Griffiths, "Embers of autoregression show how large language models are shaped by the problem they are trained to solve," *Proceedings of the National Academy of Sciences*, vol. 121, no. 41, p. e2322420121, 2024.

[11] S. Ni, K. Bi, J. Guo, and X. Cheng, "How knowledge popularity influences and enhances llm knowledge boundary perception," *arXiv preprint arXiv:2505.17537*, 2025.

[12] M. Marina, N. Ivanov, S. Pletenev, M. Salnikov, D. Galimzianova, N. Krayko, V. Konovalov, A. Panchenko, and V. Moskvoretskii, "Llm-independent adaptive rag: Let the question speak for itself," *arXiv preprint arXiv:2505.04253*, 2025.

[13] W. Arroyo-Machado, D. Torres-Salinas, and R. Costas, "Wikinformetrics: Construction and description of an open wikipedia knowledge graph data set for informetric purposes," *Quantitative science studies*, vol. 3, no. 4, pp. 931–952, 2022.

[14] W. Tang, Y. Cao, Y. Deng, J. Ying, B. Wang, Y. Yang, Y. Zhao, Q. Zhang, X.-J. Huang, Y.-G. Jiang *et al.*, "Evowiki: Evaluating llms on evolving knowledge," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 948–964.

[15] Z. Li, J. Xiong, F. Ye, C. Zheng, X. Wu, J. Lu, Z. Wan, X. Liang, C. Li, Z. Sun *et al.*, "Uncertaintyrag: Span-level uncertainty enhanced long-context modeling for retrieval-augmented generation," *arXiv preprint arXiv:2410.02719*, 2024.

[16] S. Liu, Z. Li, X. Liu, R. Zhan, D. F. Wong, L. S. Chao, and M. Zhang, "Can llms learn uncertainty on their own? expressing uncertainty effectively in a self-training manner," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 21 635–21 645.

[17] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. Le *et al.*, "Least-to-most prompting enables complex reasoning in large language models," *arXiv preprint arXiv:2205.10625*, 2022.

[18] C.-M. Chan, C. Xu, R. Yuan, H. Luo, W. Xue, Y. Guo, and J. Fu, "Rq-rag: Learning to refine queries for retrieval augmented generation," *arXiv preprint arXiv:2404.00610*, 2024.

[19] Z. Yue, H. Zhuang, A. Bai, K. Hui, R. Jagerman, H. Zeng, Z. Qin, D. Wang, X. Wang, and M. Bendersky, "Inference scaling for long-context retrieval augmented generation," *arXiv preprint arXiv:2410.04343*, 2024.

[20] J. Wu, L. Yang, Y. Ji, W. Huang, B. F. Karlsson, and M. Okumura, "Gendec: A robust generative question-decomposition method for multi-hop reasoning," *arXiv preprint arXiv:2402.11166*, 2024.

[21] Z. Shao, Y. Gong, Y. Shen, M. Huang, N. Duan, and W. Chen, "Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy," *arXiv preprint arXiv:2305.15294*, 2023.

[22] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-rag: Learning to retrieve, generate, and critique through self-reflection," 2024.

[23] T. Labruna, J. A. Campos, and G. Azkune, "When to retrieve: Teaching llms to utilize information retrieval effectively," in *Proceedings of the 15th International Conference on Recent Advances in Natural Language Processing-Natural Language Processing in the Generative AI Era*, 2025, pp. 623–632.

[24] Z. Jiang, F. F. Xu, L. Gao, Z. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, J. Callan, and G. Neubig, "Active retrieval augmented generation," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 7969–7992.

[25] Z. Yao, W. Qi, L. Pan, S. Cao, L. Hu, L. Weichuan, L. Hou, and J. Li, "Seakr: Self-aware knowledge retrieval for adaptive retrieval augmented generation," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 27 022–27 043.

[26] H. Liu, H. Zhang, Z. Guo, K. Dong, X. Li, Y. Q. Lee, C. Zhang, and Y. Liu, "Ctrla: Adaptive retrieval-augmented generation via probe-guided control," *arXiv e-prints*, pp. arXiv–2405, 2024.

[27] S. Jeong, J. Baek, S. Cho, S. J. Hwang, and J. C. Park, "Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity," *arXiv preprint arXiv:2403.14403*, 2024.

[28] W. Su, Y. Tang, Q. Ai, Z. Wu, and Y. Liu, "Dragin: dynamic retrieval augmented generation based on the information needs of large language models," *arXiv preprint arXiv:2403.10081*, 2024.

[29] P. Verma, S. P. Midigeshi, G. Sinha, A. Solin, N. Natarajan, and A. Sharma, "Plan* rag: Efficient test-time planning for retrieval augmented generation," *arXiv preprint arXiv:2410.20753*, 2024.

[30] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning, "Hotpotqa: A dataset for diverse, explainable multi-hop question answering," in *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2018, pp. 2369–2380.

[31] X. Ho, A.-K. D. Nguyen, S. Sugawara, and A. Aizawa, "Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps," *arXiv preprint arXiv:2011.01060*, 2020.

[32] Q. Qin, Y. Luo, Y. Lu, Z. Chu, X. Liu, and X. Meng, "Towards adaptive memory-based optimization for enhanced retrieval-augmented generation," in *Findings of the Association for Computational Linguistics: ACL 2025*, 2025, pp. 7991–8004.

[33] T. Khot, H. Trivedi, M. Finlayson, Y. Fu, K. Richardson, P. Clark, and A. Sabharwal, "Decomposed prompting: A modular approach for solving complex tasks," *arXiv preprint arXiv:2210.02406*, 2022.