Memory-Aware and Uncertainty-Guided Retrieval for Multi-Hop Question Answering

Yuelyu Ji
University of Pittsburgh
Pittsburgh, PA, USA
yuj49@pitt.edu

Rui Meng
Google Cloud AI Research
Palo Alto, CA, USA

Zhuochun Li
University of Pittsburgh
Pittsburgh, PA, USA

Daqing He
University of Pittsburgh
Pittsburgh, PA, USA
dah44@pitt.edu

Abstract

Multi-hop question answering (QA) requires models to retrieve and reason over multiple pieces of evidence. While Retrieval-Augmented Generation (RAG) has made progress in this area, existing methods often suffer from two key limitations: (1) fixed or overly frequent retrieval steps, and (2) ineffective use of previously retrieved knowledge. We propose MIND (Memory-Informed and INteractive Dynamic RAG), a framework that addresses these challenges through: (i) prompt-based entity extraction to identify reasoning-relevant elements, (ii) dynamic retrieval triggering based on token-level entropy and attention signals, and (iii) memory-aware filtering, which stores high-confidence facts across reasoning steps to enable consistent multi-hop generation.https://github.com/JoyDajunSpaceCraft/MIND.git.

Keywords

Retrieval-Augmented Generation, Multi-Hop Retrieval,

ACM Reference Format:
Yuelyu Ji, Rui Meng, Zhuochun Li, and Daqing He. 2018. Memory-Aware and Uncertainty-Guided Retrieval for Multi-Hop Question Answering. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Recent advances in large language models (LLMs) have significantly improved the performance of open-domain question answering (QA) systems, particularly when augmented with external knowledge retrieval [3, 8, 12, 15, 16, 31, 33]. However, many real-world questions require *multi-hop* reasoning—a process of sequentially combining information from multiple sources before arriving at the final answer [7, 29]. Traditional retrieval-augmented generation

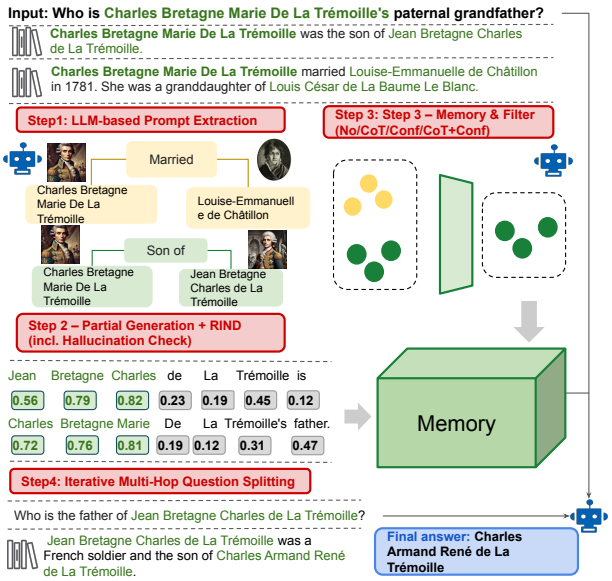

Figure 1: Overview of MIND. Given a multi-hop query (e.g., "Who is Charles Bretagne Marie De La Trémoille's paternal grandfather?"), Step 1 (§3.1) uses an LLM prompt to extract candidate entities/facts. Step 2 (§3.2) monitors partial generation with RIND and triggers retrieval when uncertainty rises. Step 3 (§3.3) stores high-confidence items in a memory module while discarding low-confidence ones (using either No Filter, CoT, Conf, or CoT+Conf). Step 4 (§3.4) repeats subquery refinement (e.g., "Who is Jean Bretagne Charles's father?") until no further retrieval is needed, yielding the final answer.

(*RAG*) methods often struggle with such tasks due to their inability to **adaptively retrieve information at the right moments**, sometimes retrieving too frequently or insufficiently [21, 30]. Moreover, these models lack mechanisms to robustly **carry forward** partially retrieved facts, leading to incomplete reasoning chains or redundant retrievals [11, 15, 19, 27, 28].

To address these challenges, recent studies have explored **dynamic retrieval**, where retrieval decisions are made adaptively during inference rather than following a fixed schedule. Notable

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

approaches include DRAGIN [21] and SEAKER [30], which trigger retrieval based on real-time uncertainty signals. Meanwhile, memory-based approaches, such as MemorAG [19], aim to track reliable facts to enhance reasoning consistency. Despite these efforts, models still struggle with **(1) Determining what to retrieve**: as chain-of-thought prompting [24] can introduce hallucinated entities, and purely confidence-based filtering may discard valuable but uncertain information; and **(2) Efficiently storing and reusing relevant facts**: without a structured memory mechanism, models risk inconsistencies in multi-step reasoning.

To address these limitations, we propose MIND (Memory-Informed & INteractive Dynamic RAG), a unified framework designed for multi-hop QA. As shown in Figure 1, MIND employs **dynamic thresholding** to monitor token-level entropy and attention patterns, determining when additional retrieval is required. This process is guided by **RIND (Retrieval-Integrated Neural Decision-making)**, which adaptively triggers retrieval based on real-time uncertainty signals. When retrieval is triggered, MIND generates a **sub-query**—a refined query derived from intermediate reasoning—to retrieve missing information while maintaining contextual relevance. Additionally, a **memory store** ensures retrieved entities remain accessible across reasoning steps, while a flexible **filtering strategy** balances recall and precision by integrating chain-of-thought reasoning with confidence-based ranking.

We evaluate MIND on four widely used multi-hop QA datasets: HotpotQA [29], 2WikiMultihopQA [7], StrategyQA [5], and IIRC [4]. Our experiments demonstrate that MIND significantly reduces unnecessary retrieval calls while improving answer quality, as measured by F1 score and Exact Match (EM). Furthermore, detailed analyses reveal how different filtering modes (e.g., chain-of-thought vs. confidence ranking) impact retrieval efficiency and correctness, offering insights into balancing efficiency with thorough multi-hop reasoning. Our main contributions are as follows:

- **Memory-aware dynamic retrieval**: We introduce a retrieval pipeline that adaptively triggers retrieval based on real-time uncertainty signals.
- **Entity-filtering strategies**: We propose multiple techniques to balance recall and precision, enhancing retrieval efficiency.
- **Extensive empirical validation**: We provide comprehensive experiments and ablation studies on four datasets, demonstrating the effectiveness of MIND for multi-step reasoning.

2 Related Work

2.1 Multi-Hop Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) has significantly improved open-domain QA by integrating external retrieval with language models [2, 9, 12–14, 17, 23, 25, 26, 32, 34]. Early approaches, such as RETRO [1] and ICRALM [20], adopt static retrieval schedules, triggering lookups at fixed intervals (e.g., every few tokens or sentences). More recent dynamic retrieval strategies, including DRAGIN [21], FLARE [10], and SEAKER [30], adaptively determine when additional retrieval is necessary, improving multi-hop reasoning efficiency.

Some of these dynamic retrieval approaches incorporate entity-based retrieval mechanisms to enhance sub-query generation. For instance, GraphRAG [6] structures knowledge into relational graphs, while KEPS [18] ranks extracted entities to improve retrieval precision. However, these methods often rely on static extraction thresholds and lack adaptive mechanisms to dynamically refine retrieval strategies. Our approach builds on these ideas by integrating a dynamic thresholding mechanism that refines entity selection based on real-time retrieval signals, ensuring sub-queries remain contextually relevant across reasoning hops.

2.2 Memory-Augmented Systems

Memory-augmented retrieval methods aim to enhance long-term context awareness by retaining high-confidence facts across multiple retrieval steps. Early memory networks [22] introduced end-to-end storage mechanisms, while more recent models like MemoRAG [19] refine retrieval by persistently storing extracted entities. However, these methods often lack adaptive filtering, leading to **redundant retrieval steps**. and inefficient memory utilization.

Our approach builds upon these foundations by integrating a dynamic memory mechanism that selectively retains and refines stored information based on real-time uncertainty signals. This enhances retrieval efficiency and ensures consistent reasoning across multi-hop QA tasks.

3 Methodology

We propose an integrated pipeline, **MIND** (Memory-Informed & Interactive Dynamic RAG), for multi-hop question answering. As shown in Figure 1, MIND interleaves generation with retrieval based on a dynamic confidence/attention estimator.

3.1 Prompt Extraction

Given a question Q, we prompt an LLM to extract potentially relevant entities and relations. For instance:

> *"Extract any names, events, or relationships that might be relevant to answering Q."*

The LLM output is parsed to produce a list of candidate entities $\{e_i\}$ and their relations $\{r_i\}$. Notably, we do not request confidence scores at this stage; these will be computed dynamically in later retrieval steps (see Section 3.3).

3.2 Retrieval-Integrated Neural Decision-making (RIND)

To determine when additional retrieval is required, we introduce **Retrieval-Integrated Neural Decision-making (RIND)**, a mechanism that adaptively triggers retrieval based on real-time uncertainty signals. RIND monitors two key uncertainty signals: **token-level entropy** and **attention influence**, which are formally defined below.

3.2.1 Entropy and Attention Influence for Retrieval. At each decoding step i, let $\{p(t \mid \text{context}_i)\}$ be the probability distribution over possible next tokens t. We define $\text{entropy}(t_i)$ as:

$$\text{entropy}(t_i) \;=\; -\sum_{t} p(t \mid \text{context}_i)\, \log p(t \mid \text{context}_i), \quad (1)$$

A larger $\text{entropy}(t_i)$ indicates greater uncertainty, suggesting that more external information may be needed.

We also measure the *attention influence* of token t_i, defined as:

$$\text{maxAttn}(t_i) = \max_{\text{future tokens}} \text{AttentionWeight}(t_i). \quad (2)$$

If $\text{maxAttn}(t_i)$ is high, then t_i strongly affects subsequent reasoning steps. We trigger retrieval if any token's uncertainty signal exceeds a dynamic threshold θ:

$$\theta = \alpha\,\text{mean}\big(\{\text{entropy}(t_i)\}\big) + \beta\,\text{mean}\big(\{\text{maxAttn}(t_i)\}\big), \quad (3)$$

where α and β are tunable parameters. If $\max_i S_{\text{RIND}}(t_i) > \theta$, retrieval is initiated.

3.3 Memory-Aware Entity Filtering

Once retrieval is triggered, we determine which extracted entities should be incorporated into the next sub-query. We employ three filtering strategies: **No Filtering**, **Chain-of-Thought (CoT) Filtering**, **Confidence-Based Filtering**, and **Hybrid Filtering**.

No Filtering (Baseline). This approach includes all extracted entities and relations in the sub-query without ranking or pruning. While maximizing recall, it risks incorporating irrelevant entities, reducing retrieval efficiency.

Chain-of-Thought (CoT) Filtering. This filter ensures that extracted entities remain logically consistent with the original query by validating them against structured reasoning steps.

Confidence-Based Filtering. We quantify each token's uncertainty and influence using entropy from Eq. 1 and $a_{\max}$ from Eq. 2. For an entity e spanning token indices $[t_s, t_e)$, we define:

$$\text{conf}(e) = \max_{t \in [t_s, t_e)} \left[\gamma \frac{1}{1 + \text{entropy}(t)} + \delta\,\text{maxAttn}(t)\right] \quad (4)$$

Entities with higher $\text{conf}(e)$ are preferred. We keep either the top-k or those above a threshold.

Hybrid: CoT + Confidence Filtering. To further enhance precision, we introduce a **hybrid filtering approach** that integrates CoT Filtering with Confidence-Based Filtering. First, CoT filtering removes logically inconsistent entities. Then, the remaining entities are ranked using the confidence-based scoring function. The final selection is determined using a predefined threshold or a top-k ranking strategy.

3.4 Iterative Multi-Hop Expansion

Many queries require multiple rounds of retrieval. Once new entities are identified, a refined sub-query is formed (e.g., "Who is the father of Jean Bretagne Charles de La Trémoille?"), and relevant facts are retrieved. The retrieved facts are stored in memory M, and the model iterates through retrieval and generation steps (using RIND) until no further retrieval is needed.

Final Processing. Once retrieval concludes, the model synthesizes retrieved information to generate the final answer. Figure 1 illustrates an example of this iterative process.

4 Experiments and Results

In this section, we will present our systematic evaluation of the proposed **MIND** framework on multi-hop QA tasks to verify its efficiency and effectiveness in retrieving and aggregating external knowledge. Specifically, we investigate three key aspects of MIND's performance.

First, we examine *whether MIND outperforms existing dynamic retrieval methods in terms of final answer accuracy* under complex multi-hop reasoning. Second, we evaluate the *effectiveness of our dynamic thresholding strategy*, which integrates attention and entropy signals to reduce unnecessary retrieval calls while maintaining correctness. Finally, we analyze *how the memory-aware design helps* maintain cross-hop consistency and mitigates the risk of dropping or misusing key entities. We primarily used LLaMA3.1–8B model or its distilled variant (DeepSeek R1 Distill LLaMA 8B). BM25 served as our external retriever.

4.1 Datasets and Baselines

We evaluate MIND on four widely used multi-hop QA benchmarks: **HotpotQA** (bridging reasoning across paragraphs), **2WikiMultihopQA** (multi-hop Wikipedia linking), **StrategyQA** (implicit reasoning in yes/no format), and **IIRC** (reasoning with incomplete context). We report **Exact Match (EM)** and **F1**, with **Accuracy** additionally used for yes/no tasks.

We compare MIND against two dynamic retrieval baselines: **DRAGIN** [21], which triggers retrieval based on a fixed confidence threshold but lacks entity-level memory filtering, and **SEAKER** [30], which generates partial sub-questions for retrieval but offers a less flexible filtering mechanism. Additionally, we include a *No Filter* baseline as a lower bound for comparison.

4.2 Overall Performance

As shown in Table 1, MIND consistently outperforms baselines across all datasets. On **HotpotQA**, it improves EM and F1 by 2–3%, indicating enhanced reasoning stability for bridging questions. On **2WikiMultihopQA**, it achieves gains of +3.0% EM and +3.5% F1, while on **StrategyQA**, its implicit reasoning capability leads to 2–4% higher accuracy. For **IIRC**, MIND reduces retrieval overhead and mitigates incorrect references by pruning spurious entities.

4.2.1 Retrieval Frequency and Efficiency. We measured average retrieval calls and total token usage as indicators of system efficiency. Table 2 shows that, compared with fixed-schedule retrieval (e.g., every n sentences), MIND's **dynamic thresholding** cuts unnecessary retrieval by around 10–15% in the Llama3.1-8B based results. The memory unit caches verified entities/facts across hops, preventing repeated entity retrieval calls and reducing cost.

4.2.2 Ablation Study. We further analyze the impact of different filtering strategies—*No Filter*, *CoT Filter*, *Confidence Filter (Conf)*, and the combined *CoT+Conf*—in Table 3. We find that **No Filter** tends to introduce noise, which lowers the overall accuracy. By contrast, **CoT Filter** removes off-topic reasoning, boosting performance on complex bridging questions. **Conf Filter** improves sub-query precision by ranking entities based on token-level entropy and attention. Finally, **CoT+Conf** achieves the best balance of precision

Table 1: Comparison of different ranking strategies on four multi-hop QA datasets (2Wiki, Hotpot, StrategyQA, IIRC), against two baseline models: DeepSeek R1 Distill LLaMA 8B (left) and Llama3.1–8B (right). We report Exact Match (EM) and F1 (in %).

Method	DeepSeek-R1-Distill-LLaMA-8B							Llama3.1–8B						
	2Wiki		Hotpot		Strategy	IIRC		2Wiki		Hotpot		Strategy	IIRC	
	EM	F1	EM	F1	ACC	EM	F1	EM	F1	EM	F1	ACC	EM	F1
Baseline														
DRAGIN	30.0	38.5	30.5	40.1	65.0	18.0	21.9	30.4	39.3	31.4	42.4	63.9	18.5	22.2
SEAKER	31.0	40.1	31.2	42.0	66.1	**18.8**	**22.5**	31.2	40.6	32.1	44.8	65.0	19.3	23.0
MIND														
No Filter	$24.0^{\pm 0.3}$	$32.8^{\pm 0.5}$	$25.1^{\pm 0.4}$	$37.3^{\pm 0.6}$	$62.0^{\pm 0.02}$	$16.2^{\pm 0.02}$	$19.9^{\pm 0.03}$	$25.0^{\pm 0.4}$	$33.5^{\pm 0.5}$	$27.0^{\pm 0.6}$	$38.1^{\pm 0.7}$	$60.0^{\pm 0.02}$	$17.8^{\pm 0.3}$	$21.5^{\pm 0.4}$
Confidence Filter	$29.5^{\pm 0.4}$	$38.0^{\pm 0.5}$	$30.2^{\pm 0.5}$	$39.9^{\pm 0.6}$	$\mathbf{67.0}^{\pm 0.02}$	$16.5^{\pm 0.02}$	$18.4^{\pm 0.03}$	$30.0^{\pm 0.4}$	$38.8^{\pm 0.5}$	$31.0^{\pm 0.6}$	$40.2^{\pm 0.7}$	$\mathbf{69.0}^{\pm 0.02}$	$18.3^{\pm 0.3}$	$22.8^{\pm 0.4}$
CoT Filter	$\mathbf{33.2}^{\pm 0.5}$	$\mathbf{42.3}^{\pm 0.6}$	$\mathbf{32.8}^{\pm 0.6}$	$\mathbf{45.2}^{\pm 0.7}$	$56.0^{\pm 0.02}$	$16.5^{\pm 0.02}$	$19.4^{\pm 0.04}$	$\mathbf{34.0}^{\pm 0.5}$	$\mathbf{43.0}^{\pm 0.6}$	$34.5^{\pm 0.6}$	$46.5^{\pm 0.7}$	$67.0^{\pm 0.02}$	$\mathbf{20.8}^{\pm 0.4}$	$\mathbf{25.0}^{\pm 0.5}$
Conf + CoT	$31.0^{\pm 0.6}$	$38.5^{\pm 0.7}$	$31.9^{\pm 0.7}$	$43.8^{\pm 0.8}$	$48.4^{\pm 0.02}$	$18.4^{\pm 0.01}$	$20.9^{\pm 0.04}$	$32.0^{\pm 0.4}$	$41.7^{\pm 0.5}$	$\mathbf{35.8}^{\pm 0.7}$	$\mathbf{47.2}^{\pm 0.8}$	$62.0^{\pm 0.02}$	$12.0^{\pm 0.05}$	$13.9^{\pm 0.06}$

and recall, with ($\gamma = 1.0, \delta = 0.2$) yielding the highest EM/F1 on HotpotQA.

Notably, in more straightforward queries (e.g. yes/no classification), certain baselines such as DRAGIN or SEAKER can occasionally match or exceed our method. We suspect these baselines are well-tuned for single-step retrieval on short questions, whereas *MIND* is designed for more complex multi-hop reasoning.

4.2.3 Fixed vs. Dynamic Thresholding. We also explore the effectiveness of our dynamic thresholding approach in deciding when to trigger retrieval. Table 4 compares a *fixed* threshold of 0.6 against our *dynamic* threshold on the HotpotQA dev set. Although the performance gap is modest (e.g. EM = 0.304 vs. 0.309), we observe a consistent improvement in both EM and F1 under the dynamic scheme. This indicates that adaptively adjusting the threshold based on token-level uncertainty can better handle questions of varying complexity than a single, fixed cutoff.

4.2.4 Limitations of CoT + Conf *Filtering.* Although combining CoT and Conf generally enhances retrieval, Table 1 shows that it does not always outperform using either filter alone. In simple queries (e.g., "Who is older, Annie Morton or Terry Richardson?"), chain-of-thought reasoning may introduce unnecessary elaboration, which the confidence filter repeatedly prunes—adding overhead. Excessive filtering can also remove low-certainty but necessary bridging entities, weakening multi-hop reasoning. Finally, while CoT expansion and Conf pruning can complement each other on complex queries, their interplay may be redundant or contradictory on straightforward tasks. As a result, **CoT+Conf** often excels on intricate bridging questions but can trail simpler approaches in more direct scenarios.

Table 2: Average retrieval calls (#Ret) across four datasets under different methods. "DS" = DeepSeek, "L3.1" = Llama3.1–8B.

Method	#Ret (DS / L3.1)			
	2Wiki	Hotpot	Strategy	IIRC
No Filter	4.25 / 3.10	4.15 / 2.80	4.36 / 3.39	4.56 / 3.13
Confidence Filter	4.20 / 3.04	4.05 / 2.75	4.30 / 3.20	4.35 / 3.00
CoT Filter	4.28 / 3.04	4.10 / 2.50	4.86 / 3.39	4.44 / 3.10
Conf + CoT	4.21 / 3.02	4.08 / 2.90	4.79 / 3.60	4.60 / 3.15
DRAGIN[21]	3.90 / 2.80	3.85 / 3.10	3.95 / 2.75	4.00 / 2.90
SEAKER[30]	3.80 / 2.60	3.75 / 2.90	3.85 / 2.70	3.90 / 2.85

Table 3: Effect of different aggregator hyperparameters (γ, δ) on HotpotQA dev set.

γ	δ	EM	F1	#Ret
0.5	0.1	0.290	0.382	3.4
1.0	0.2	**0.296**	**0.388**	**3.2**
1.5	0.3	0.293	0.386	3.3

Table 4: Comparison of fixed threshold = 0.6 vs. dynamic threshold on HotpotQA.

Threshold	EM	F1	Prec.
0.6 (Fixed)	0.304	0.393	0.395
Dynamic	**0.309**	**0.399**	**0.402**

5 Conclusion and Future Work

In this paper, we introduced a novel approach to enhance multi-hop retrieval-augmented generation by incorporating dynamic thresholding, prompt-based entity extraction, and memory-aware queries. Our experiments show that these enhancements significantly improve multi-hop reasoning, entity coverage, and final answer quality.

Future work will focus on extending this framework to **conversational AI systems**, where multi-turn interactions require robust retrieval strategies. Additionally, we aim to explore **cross-domain applications**, as our model's dynamic retrieval mechanism could be beneficial for tasks requiring adaptive reasoning across heterogeneous knowledge sources. Another important direction is **improving memory update mechanisms** to handle long-term dependencies, as our analysis suggests that entity retention plays a crucial role in maintaining cross-hop consistency.

Our experiments demonstrate that memory-aware retrieval and confidence-guided entity filtering significantly improve multi-hop

QA performance, particularly in reducing unnecessary retrievals while maintaining accuracy. Compared to existing baselines, MIND achieves **higher** entity coverage, more precise retrieval triggers, and improved final answer correctness across multiple datasets. Further optimizations in retrieval efficiency will be essential for scaling this approach to large-scale QA applications.

References

[1] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*. PMLR, 2206–2240.

[2] Xinwei Chen, Kun Li, Tianyou Song, and Jiangjian Guo. 2024. Mix of Experts Language Model for Named Entity Recognition. (2024), 502–506. https://doi.org/10.1109/CISCE62493.2024.10653372

[3] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130* (2024).

[4] James Ferguson, Matt Gardner, Tushar Khot, and Pradeep Dasigi. 2020. IIRC: A Dataset of Incomplete Information Reading Comprehension Questions. In *Conference on Empirical Methods in Natural Language Processing*. https://api.semanticscholar.org/CorpusID:226262208

[5] Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics* 9 (2021), 346–361.

[6] Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A Rossi, Subhabrata Mukherjee, Xianfeng Tang, et al. 2024. Retrieval-augmented generation with graphs (graphrag). *arXiv preprint arXiv:2501.00309* (2024).

[7] 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*, Donia Scott, Nuria Bel, and Chengqing Zong (Eds.). International Committee on Computational Linguistics, Barcelona, Spain (Online), 6609–6625. https://doi.org/10.18653/v1/2020.coling-main.580

[8] Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. 2024. GRAG: Graph Retrieval-Augmented Generation. *arXiv preprint arXiv:2405.16506* (2024).

[9] Tongzhou Jiang, Lipeng Liu, Junyue Jiang, Tianyao Zheng, Yuhui Jin, and Kunpeng Xu. 2024. Trajectory tracking using frenet coordinates with deep deterministic policy gradient. *arXiv preprint arXiv:2411.13885* (2024).

[10] 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. *arXiv preprint arXiv:2305.06983* (2023).

[11] Yihong Jin, Ze Yang, Xinhe Xu, Yihan Zhang, and Shuyang Ji. 2025. Adaptive Fault Tolerance Mechanisms of Large Language Models in Cloud Computing Environments. *arXiv preprint arXiv:2503.12228* (2025).

[12] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[13] Kun Li, Xinwei Chen, Tianyou Song, Hansong Zhang, Wenzhe Zhang, and Qing Shan. 2024. GPTDrawer: Enhancing Visual Synthesis through ChatGPT. (2024). arXiv:2412.10429 [cs.CV] https://arxiv.org/abs/2412.10429

[14] Kun Li, Xinwei Chen, Tianyou Song, Chengrui Zhou, Zhuoran Liu, Zhenyan Zhang, Jiangjian Guo, and Qing Shan. 2025. Solving Situation Puzzles with Large Language Model and External Reformulation. (2025). arXiv:2503.18394 [cs.LG] https://arxiv.org/abs/2503.18394

[15] Zijian Li, Qingyan Guo, Jiawei Shao, Lei Song, Jiang Bian, Jun Zhang, and Rui Wang. 2024. Graph Neural Network Enhanced Retrieval for Question Answering of LLMs. *arXiv preprint arXiv:2406.06572* (2024).

[16] Xueting Lin, Yuming Tu, Qingyi Lu, Jinghan Cao, Haowei Yang, et al. [n. d.]. Research on Content Detection Algorithms and Bypass Mechanisms for Large Language Models. *Academic Journal of Computing & Information Science* 8, 1 ([n. d.]), 48–56.

[17] Huanshuo Liu, Hao Zhang, Zhijiang Guo, Jing Wang, Kuicai Dong, Xiangyang Li, Yi Lee, Cong Zhang, and Yong Liu. 2024. CtrlA: Adaptive Retrieval-Augmented Generation via Inherent Control. https://api.semanticscholar.org/CorpusID:273163564

[18] Shuqi Lu, Zhicheng Dou, Chenyan Xiong, Xiaojie Wang, and Ji rong Wen. 2020. Knowledge Enhanced Personalized Search. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020). https://api.semanticscholar.org/CorpusID:220730253

[19] Hongjin Qian, Peitian Zhang, Zheng Liu, Kelong Mao, and Zhicheng Dou. 2024. Memorag: Moving towards next-gen rag via memory-inspired knowledge discovery. *arXiv preprint arXiv:2409.05591* (2024).

[20] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics* 11 (2023), 1316–1331.

[21] Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. Dragin: Dynamic retrieval augmented generation based on the real-time information needs of large language models. *arXiv preprint arXiv:2403.10081* (2024).

[22] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. *Advances in neural information processing systems* 28 (2015).

[23] Changyue Wang, Weihang Su, Yiran Hu, Qingyao Ai, Yueyue Wu, Cheng Luo, Yiqun Liu, Min Zhang, and Shaoping Ma. 2024. LeKUBE: A Knowledge Update BEnchmark for Legal Domain. In *SIGIR-AP*. https://api.semanticscholar.org/CorpusID:274596689

[24] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[25] Kehan Xu, Kun Zhang, Jingyuan Li, Wei Huang, and Yuanzhuo Wang. 2024. CRP-RAG: A Retrieval-Augmented Generation Framework for Supporting Complex Logical Reasoning and Knowledge Planning. *Electronics* (2024). https://api.semanticscholar.org/CorpusID:275103348

[26] Jinglan Yang, Jianghuai Liu, Zheng Yao, and Chaoqun Ma. 2024. Measuring digitalization capabilities using machine learning. *Research in International Business and Finance* 70 (2024), 102380. https://doi.org/10.1016/j.ribaf.2024.102380

[27] Ze Yang, Yihong Jin, and Xinhe Xu. 2024. HADES: Hardware Accelerated Decoding for Efficient Speculation in Large Language Models. *arXiv preprint arXiv:2412.19925* (2024).

[28] Ze Yang, Yihong Jin, Yihan Zhang, Juntian Liu, and Xinhe Xu. 2025. Research on Large Language Model Cross-Cloud Privacy Protection and Collaborative Training based on Federated Learning. *arXiv preprint arXiv:2503.12226* (2025).

[29] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600* (2018).

[30] Zijun Yao, Weijian Qi, Liangming Pan, Shulin Cao, Linmei Hu, Weichuan Liu, Lei Hou, and Juanzi Li. 2024. Seakr: Self-aware knowledge retrieval for adaptive retrieval augmented generation. *arXiv preprint arXiv:2406.19215* (2024).

[31] Yiming Zeng, Wanhao Yu, Zexin Li, Tao Ren, Yu Ma, Jinghan Cao, Xiyan Chen, and Tingting Yu. 2025. Bridging the Editing Gap in LLMs: FineEdit for Precise and Targeted Text Modifications. arXiv:2502.13358 [cs.CL] https://arxiv.org/abs/2502.13358

[32] Tianyao Zheng, Yuhui Jin, Haopeng Zhao, Zhichao Ma, Yongzhou Chen, and Kunpeng Xu. 2024. Deep Reinforcement Learning Based Coverage Path Planning in Unknown Environments. In *2024 6th International Conference on Frontier Technologies of Information and Computer (ICFTIC)*. 1608–1611. https://doi.org/10.1109/ICFTIC64248.2024.10913347

[33] Ting Zhong, Jienan Zhang, Zhangtao Cheng, Fan Zhou, and Xueqin Chen. 2024. Information Diffusion Prediction via Cascade-Retrieved In-context Learning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 2472–2476. https://doi.org/10.1145/3626772.3657909

[34] Zhui Zhu, Guangpeng Qi, Guangyong Shang, Qingfeng He, Weichen Zhang, Ningbo Li, Yunzhi Chen, Lijun Hu, Wenqiang Zhang, and Fan Dang. 2024. Enhancing Large Language Models with Knowledge Graphs for Robust Question Answering. *2024 IEEE 30th International Conference on Parallel and Distributed Systems (ICPADS)* (2024), 262–269. https://api.semanticscholar.org/CorpusID:274372990

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009