

Beyond Static Retrieval: Opportunities and Pitfalls of Iterative Retrieval in GraphRAG

Kai Guo¹, Xinnan Dai¹, Shenglai Zeng¹

Harry Shomer², Haoyu Han¹, Yu Wang³, Hui Liu¹

¹Michigan State University ²University of Texas at Arlington ³University of Georgia
{guokai1, daixinna, zengshe1, hanhaoy1, liuhui7}@msu.edu,
harry.shomer@uta.edu,
yu.wang6@uga.edu

Abstract

Graph-based RAG (GraphRAG) leverages entity–relation graphs to support multi-hop reasoning, but most systems still rely on static retrieval. When crucial evidence, especially bridge documents that connect disjoint entities, is absent, reasoning collapses and hallucinations persist. Iterative retrieval, which performs multiple rounds of evidence selection, has emerged as a promising alternative, yet its role within GraphRAG remains poorly understood. We present the first systematic study of iterative retrieval in GraphRAG by analyzing how different strategies interact with graph-based backbones and investigating when they are effective. Our findings reveal both opportunities and pitfalls which together highlight a central bottleneck, namely that GraphRAG’s effectiveness depends not only on recall but also on whether bridge evidence is consistently promoted into leading positions where it can support reasoning chains. To address this limitation, we propose *Bridge-Guided Dual-Thought-based Retrieval (BDTR)*, a simple yet effective framework that generates complementary thoughts and leverages reasoning chains to recalibrate rankings and bring bridge evidence into leading positions. BDTR achieves consistent improvements across diverse GraphRAG settings and provides guidance for the design of future GraphRAG systems. Our code is available at <https://github.com/KaiGuo20/BDTR>.

1 Introduction

In recent years, retrieval-augmented generation (RAG) has become a core paradigm for enhancing large language models (LLMs) on knowledge-intensive question answering (Xia et al., 2025; Lewis et al., 2020; Gao et al., 2023; Fan et al., 2024). By grounding generation in external evidence, RAG is able to effectively mitigate hallucination (Ayala and Bécard, 2024; Niu et al., 2024). Despite its tremendous success, standard

RAG systems often struggle with multi-hop reasoning, where multiple pieces of evidence are required to be linked across retrieval and inference (Tang and Yang, 2024; Saleh et al., 2024; Han et al., 2026).

To address this challenge, graph-based retrieval-augmented generation (GraphRAG) has emerged as a promising extension (Edge et al., 2024; Han et al., 2024; He et al., 2024; Wu et al., 2025; Jiang et al., 2024). By integrating entity–relation knowledge graphs into the retrieval pipeline, GraphRAG supports structured reasoning over multi-hop paths and has achieved strong performance on multi-hop QA tasks (Zou et al., 2025; Jimenez Gutierrez et al., 2024; Gutiérrez et al., 2025; Mavromatis and Karypis, 2025). However, most existing GraphRAG systems rely on single-shot static retrieval. If crucial evidence is absent from the selected candidates, the reasoning process collapses, and hallucinations persist (Luo et al., 2025; Guo et al., 2025).

Meanwhile, iterative retrieval has gained growing attention in the broader RAG literature (Trivedi et al., 2023; Jiang et al., 2025; Lee et al., 2025). Instead of committing to one retrieval step, iterative methods allow models to perform multiple retrieval rounds during reasoning, progressively refining or expanding the evidence set (Shao et al., 2023). This dynamic process can improve coverage, reduce hallucination, and has shown benefits in chain-of-thought and self-ask style frameworks (Trivedi et al., 2023). Recent systems such as GFM-RAG (Luo et al., 2026) and HippoRAG (Jimenez Gutierrez et al., 2024) include certain explorations using iterative retrieval. However, their analyses remain unsystematic, leaving open a fundamental question: *Can iterative retrieval reliably improve GraphRAG, and under what conditions is it effective?*

In this work, we present the first comprehensive study of iterative retrieval in GraphRAG. We inte-

grate various representative GraphRAG backbones with iterative retrieval strategies and evaluate them systematically across multi-hop QA benchmarks. Our analysis uncovers both *opportunities* and *pitfalls*:

- **Opportunities.** (1) Iterative retrieval substantially improves complex multi-hop questions, especially those requiring *bridge documents*—intermediate facts that connect otherwise disjoint entities. (2) Different iterative strategies exhibit complementary strengths, indicating potential for combination. (3) Iteration can act as an implicit re-ranking mechanism: by repeatedly updating scores, gold documents are progressively promoted into the leading positions, which leads to a sharp improvement in recall in the top ranks.
- **Pitfalls.** (1) Simply expanding the number of retrieved documents is not always beneficial: while recall may increase, the additional noise often dilutes precision and undermines QA accuracy. (2) For single-hop or simple comparison questions, iterative retrieval offers little to no benefit, and may even harm performance. (3) Even when gold bridge documents are retrieved, many remain buried beyond the leading positions, making them effectively unusable for reasoning.

Together, these findings highlight a central **bottleneck**: GraphRAG’s success depends not only on overall recall coverage, but on whether bridge-bearing evidence is consistently promoted into the leading positions where it can be used to complete reasoning chains. This perspective explains our observations: performance improves significantly on bridge-type questions once the required intermediate evidence is made available in the leading ranks.

Building on this insight, we propose *Bridge-Guided Dual-Thought-based Retrieval (BDTR)*, a simple yet effective iterative framework that explicitly targets this bottleneck. BDTR generates dual thoughts at each reasoning step to broaden coverage with complementary retrieval prompt, and leverages reasoning chains to recalibrate rankings and promote bridge evidence into the leading positions. This design consistently improves multi-hop QA across diverse GraphRAG backbones and datasets, offering practical guidance for future GraphRAG systems.

2 Related Work

Due to space limitations, this section provides only a brief summary of related work. A more comprehensive version is presented in the Appendix A.1.

GraphRAG. GraphRAG improves multi-hop QA by grounding reasoning in structured graphs. Existing methods integrate graph structure in different ways, including personalized PageRank (HippoRAG2 (Gutiérrez et al., 2025)), hierarchical summarization (RAPTOR (Sarathi et al., 2024)), graph neural networks (GFM-RAG (Luo et al., 2026)), and community detection (MS-GraphRAG (Edge et al., 2024)).

Iterative Retrieval. Iterative retrieval improves RAG through multi-round retrieval and reasoning. Representative methods include IRCOT (Trivedi et al., 2023), IRGS (Shao et al., 2023), TOG (Sun et al., 2024), and GCOT (Jin et al., 2024). While these methods have shown strong benefits in standard RAG, iterative retrieval remains underexplored in GraphRAG, where current systems only use it in a limited manner.

3 Preliminary Studies

To probe the effectiveness of iterative retrieval in GraphRAG, we begin by asking a question: *does iterative retrieval always improve performance, especially on multi-hop questions where GraphRAG is expected to shine?* To answer this, we conduct experiments across multiple datasets to establish the overall effectiveness. However, rather than stopping at the observation that iterative retrieval works, we seek to unpack the underlying mechanisms. Specifically, we investigate: (i) Which question types benefit most, and which do not? (ii) From a retrieval perspective, how does recall explain these improvements? (iii) How many rounds are necessary before the benefits saturate? (iv) Do different iterative strategies exhibit distinct behaviors?

These guiding questions structure our preliminary studies. By addressing them, we not only validate the effectiveness of iterative retrieval for GraphRAG, but also develop deeper intuitions that motivate the principled framework proposed in this work.

3.1 Experimental Settings

We begin by establishing the experimental setup, specifying the backbone model, iterative module, and evaluation datasets, so that subsequent analyses

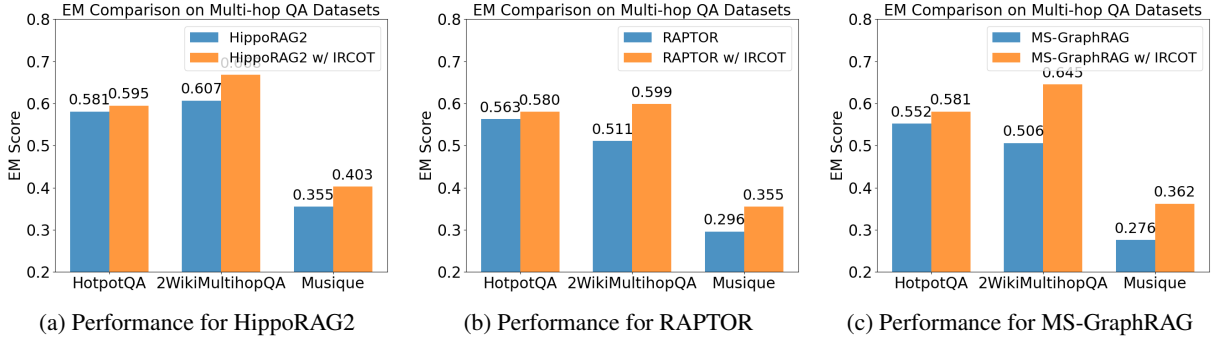


Figure 1: EM Comparison on Multi-hop QA Datasets.

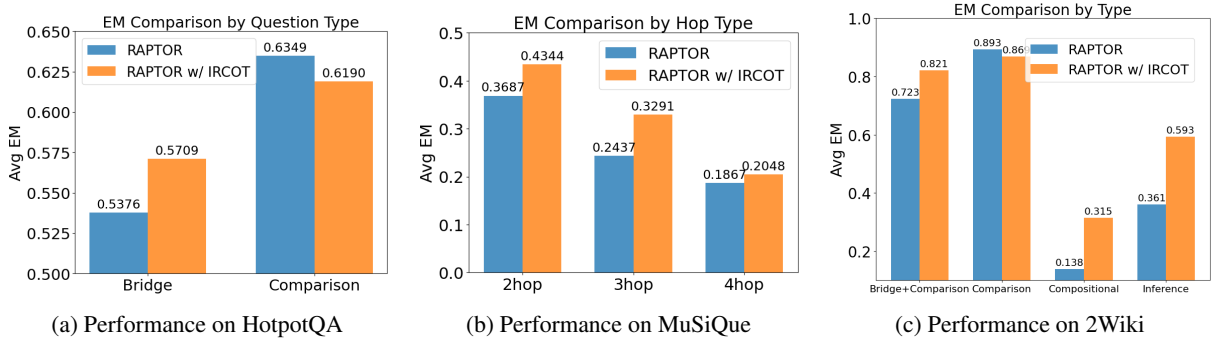


Figure 2: EM Comparison by Question Type with the RAPTOR Backbone.

can be interpreted under a consistent framework. Specifically, we adopt HippoRAG2, RAPTOR, and Microsoft’s GraphRAG (MS-GraphRAG) as backbone models, and integrate the iterative method IRCOT (Trivedi et al., 2023). Experiments are carried out on three widely used multi-hop QA datasets: HotpotQA, 2WikiMultiHopQA (abbreviated as 2Wiki), and MuSiQue. Following prior work (Gutiérrez et al., 2025), we use Exact Match (EM) and F1 as the evaluation metrics. This setup allows us to isolate the effect of iterative retrieval while controlling for other modeling factors.

3.2 Overall Effectiveness

We first verify whether iterative retrieval yields consistent gains in overall accuracy across datasets, providing a high-level validation of its effectiveness before delving into finer-grained analyses. Figure 1 shows that incorporating IRCOT consistently improves the EM performance of HippoRAG2, RAPTOR, and MS-GraphRAG across all three datasets. The corresponding F1 results are presented in Figure 17 in the Appendix. These results confirm that iterative retrieval is particularly valuable in settings that demand complex reasoning chains, as additional retrieval rounds help surface supporting evidence that static retrieval often overlooks. This

suggests that the current design of GraphRAG underutilizes its potential: while the graph structure provides a powerful foundation, its effectiveness is limited by the quality of retrieved evidence for multi-hop QA.

3.3 Question-Type Analysis

Beyond overall performance, it is crucial to identify which types of questions benefit most from iterative retrieval, since multi-hop reasoning demands vary across query categories. We therefore break down the results by question type to reveal when iterative retrieval is most beneficial. HotpotQA includes two types of questions: Bridge and Comparison. 2WikiMultiHopQA contains four types: Bridge+Comparison, Comparison, Compositional, and Inference. MuSiQue includes three types: 2-hop, 3-hop and 4-hop. Due to space limitations, we report the results for HippoRAG2 and MS-GraphRAG in Figures 19 and 20 in the Appendix.

From Figure 2, a clear pattern emerges: iterative retrieval yields the greatest benefit on *Bridge* and other multi-hop questions that require linking disparate pieces of evidence. These questions require identifying intermediate bridge entities that are rarely stated explicitly in the query, making

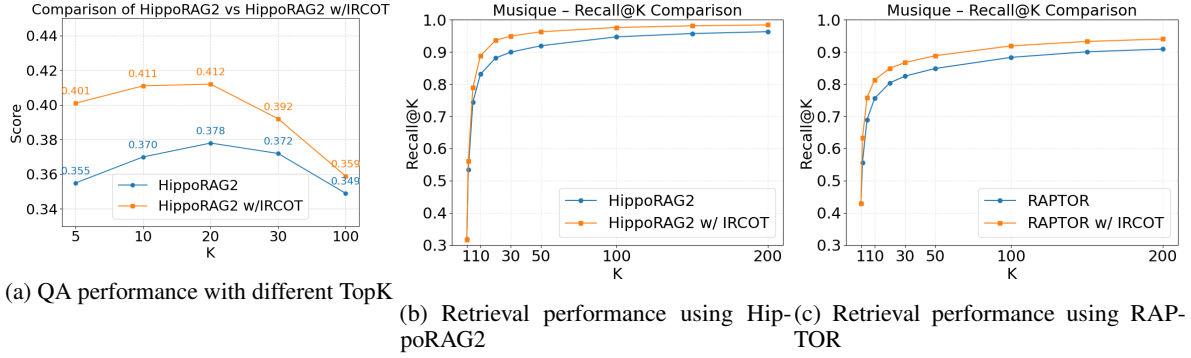


Figure 3: Impact of TopK on QA and retrieval performance. (a) EM performance across different TopK values. (b) Retrieval Recall@K comparison on HippoRAG2. (c) Retrieval Recall@K comparison on RAPTOR.

them particularly challenging for static retrieval. An illustrative example is provided in Figure 8 in the Appendix. In contrast, iterative retrieval offers little to no improvement on simple Comparison questions, and can even lead to slight performance drops due to over-retrieval and noise accumulation.

3.4 Understanding Improvements through Recall

The analysis of question types in Section 3.3 shows that iterative retrieval yields the largest gains on *Bridge*-style questions, which require identifying intermediate entities to connect disjoint evidence. This observation highlights a core challenge in GraphRAG: whether bridge-bearing documents can be ranked early enough to be effectively used for reasoning.

To better understand this issue, we analyze retrieval behavior through Recall@K. While high recall at large K indicates good coverage, coverage alone is insufficient for multi-hop QA. In practice, models operate under limited context budgets and primarily rely on top-ranked documents. Increasing K improves coverage but also introduces noise and can degrade performance, as shown in Figure 3a.

Iterative retrieval improves retrieval accuracy by progressively refining the query through interaction, thereby increasing the likelihood that relevant evidence is surfaced earlier. As shown in Figure 3b, HippoRAG2 already achieves high coverage at large K (Recall@100 $\approx$ 0.95), and iterative retrieval further improves recall at small K (e.g., Recall@5 and Recall@10), which aligns with the observed gains on bridge-type questions. However, this improvement remains limited. As illustrated in Figure 3c, even when most gold documents—including bridge documents—can even-

tually be retrieved, many still fail to appear in the leading positions, particularly for RAPTOR.

Taken together, these results clarify the central message of this work: the primary bottleneck in GraphRAG is not insufficient coverage or recall at large K , but the difficulty of reliably ranking important evidence within the leading positions under a constrained retrieval budget.

3.5 Complementarity of Iterative Methods

Another question is whether different iterative strategies capture distinct aspects of the evidence space. In that case, combining them could extend coverage and improve robustness. To investigate this, we compare two representative strategies: IRCOT (Trivedi et al., 2023) and GCOT (Jin et al., 2024).

As illustrated in Figure 4, each method succeeds on a different subset of questions, indicating that they uncover complementary evidence. This complementarity suggests that no single iterative method is univer-

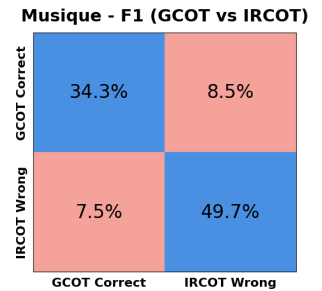


Figure 4: Complementary Study.

sally optimal. Instead, combining strategies—or adaptively selecting among them depending on the query—has the potential to achieve broader coverage and more reliable performance than any individual approach. The corresponding EM comparison and the complementarity analysis between IRGS and IRCOT are presented in the Figure 18 in the Appendix. In addition, we

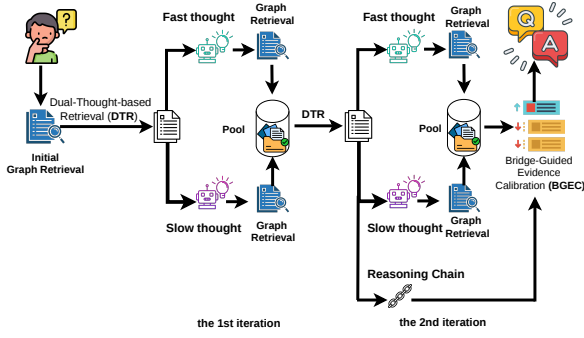


Figure 5: An illustration of our framework BDTR, shown with two iterations as an example. In each reasoning step, the model generates two thoughts to drive retrieval and constructs a reasoning chain that encodes intermediate bridge cues. The retrieved documents from the two thoughts provide diverse and complementary evidence, while the bridge-guided calibration module adjusts their ranking to ensure that critical bridge facts appear in leading position for reasoning.

explore the effect of iteration rounds. Due to space limitations, the details are provided in Appendix A.5.

Finally, we **summarize our observations**, with a more detailed discussion provided in Appendix A.2 due to space limitations.

4 Method

Based on our analysis, we design a new method guided by two key insights: **(1) Opportunity:** Different iterative methods exhibit complementary strengths. **(2) Pitfall:** While gold documents—particularly those containing bridge facts—can be retrieved, not all are ranked in the leading positions necessary for effective reasoning.

These insights motivate two core components: Dual-Thought-based Retrieval (DTR) and Bridge-Guided Evidence Calibration (BGEC). First, instead of relying on a single reasoning path, each step produces two thoughts with complementary. This design broadens coverage by combining distinct retrieval signals, ensuring that more gold evidence is captured. Second, we introduce a bridge-aware reranking mechanism that uses cues from the evolving reasoning chain to elevate bridge-bearing documents into the top ranks, where they can be effectively used. Together, these components improve both coverage and ranking quality, addressing the main bottlenecks identified in our preliminary analysis.

An overview of our framework, **Bridge-Guided Dual-Thought-based Retrieval (BDTR)**,

is shown in Figure 5. BDTR consists of two functional modules: **Dual-Thought-based Retrieval (DTR)** expands coverage by generating two retrieval signals per step, capturing evidence that a single strategy might miss. **Bridge-Guided Evidence Calibration (BGEC)** improves usability by promoting documents likely to contain bridge facts into the top ranks. These two modules operate jointly, first expanding evidence coverage and then filtering to prioritize the most relevant evidence.

4.1 Dual-Thought-based Retrieval (DTR)

Our analysis indicates that different iterative strategies capture complementary evidence. To exploit this property, each reasoning step generates two thoughts: one biased toward direct-answering passages and the other toward bridge-seeking relations. Each thought is issued as an independent query to the backbone graph retriever.

Initialization. Let Q denote the original question, f_{ret} the backbone graph retriever, and $d \in \mathcal{D}$ a candidate document from the corpus $\mathcal{D}$. In the first round, the retriever is applied to Q to obtain an initial set of documents $D_0 = f_{\text{ret}}(Q)$, where each document $d \in D_0$ is associated with a retrieval score $\hat{s}(d | Q)$. These scores are produced by GraphRAG backbones; and we take HippoRAG2 as a representative example and describe its scoring mechanism in Appendix A.13. The retrieved documents and their scores together form the initial candidate pool P_0 .

Iterative dual-thought retrieval. At iteration $t \geq 1$, two complementary thoughts are generated, each conditioned on the retrieved pool from the previous step, with P_0 used for the first iteration and P_{t-1} for subsequent ones. We denote them by q_t^{FT} and q_t^{ST} . Specifically, q_t^{FT} corresponds to the **fast thought** (FT), whereas q_t^{ST} corresponds to the **slow thought** (ST), as illustrated in Figure 14 in the Appendix. FT generates a focused query to find the next missing node in the reasoning chain, guiding the model to extend the path step by step rather than merely rephrasing the question. ST constructs a complete thought through graph-style neighborhood expansion. These thoughts are produced using distinct prompts to capture complementary facts, and are independently submitted to the retriever, yielding two sets of documents:

$$D_t^{\text{FT}} = f_{\text{ret}}(q_t^{\text{FT}}), \quad D_t^{\text{ST}} = f_{\text{ret}}(q_t^{\text{ST}}). \quad (1)$$

Pool update and sorting. The candidate pool is expanded as

$$P_t = P_{t-1} \cup D_t^{\text{FT}} \cup D_t^{\text{ST}}, \quad (2)$$

and the score of each document is updated by

$$s_t(d) \leftarrow \max\left(s_{t-1}(d), \hat{s}(d | q_t^{\text{FT}}), \hat{s}(d | q_t^{\text{ST}})\right), \\ \forall d \in P_t. \quad (3)$$

where $s_t(d)$ is the updated score of document d at iteration t and $\hat{s}(d | q)$ is the score assigned by the retriever to document d given query q . By taking the maximum in (3), each document is able to preserve its highest score. Afterward, all documents in P_t are **re-sorted** according to the updated scores.

In practice, combining the two trajectories enlarges the evidence frontier across iterations. By repeating this process, DTR improves the likelihood that gold documents are covered and retained with strong scores, paving the way for subsequent ranking calibration.

4.2 Bridge-Guided Evidence Calibration (BGEC)

Answering complex questions typically relies on bridge documents that provide the necessary connections between facts. However, such evidence is hard to surface, as it may not be explicitly reflected in the query. Furthermore, even if it is retrieved, it often appears far down in the ranking where it cannot be used. Without these bridge documents, the reasoning chain breaks, making it impossible to answer the question even if other supporting facts are present. This motivates a bridge-guided calibration step that explicitly identifies and promotes such overlooked documents.

Bridge-aware selection. In the final iteration, besides generating dual thoughts, the model also produces a reasoning chain (RC) that encodes potential bridging cues, as illustrated in Figure 14. We denote the pool of candidate documents after the last iteration as P_R . The pool P_R together with RC is sent to an LLM verifier. The verifier does not assign scores but directly selects documents that align with the reasoning chain:

$$\mathcal{G} = \{d \in P_R \mid \text{Verifier}(d, RC) = 1\}, \quad (4)$$

where $\text{Verifier}(d, RC) = 1$ indicates that the document d is judged to support RC . The concrete

prompt of verifier is shown in Figure 9 in Appendix. The selected documents $\mathcal{G}$ are then promoted to the top of P_R , ensuring that bridge-supporting evidence is made accessible to reasoning.

Final selection. After calibration, we further filter the pool to produce a compact and reliable context for QA. Let μ and σ be the mean and standard deviation of their scores. We define the final set as

$$\mathcal{D}_{\text{final}} = \{d \in P_R \mid s(d) \geq \mu + \sigma\}, \quad (5)$$

with the safeguard that at least 5 documents are always retained. This criterion guarantees that enough strong evidence remains to support the reasoning chain.

4.3 Routing Mechanism

Our method is mainly designed to address the bridge-evidence issue, and is therefore particularly beneficial for bridge-type questions. At the same time, since BDTR introduces two thought paths together with an additional verifier, it incurs higher token cost than standard iterative retrieval. To better balance effectiveness and efficiency, we adopt a lightweight routing strategy. Specifically, a model (e.g., GPT-4o-mini) first predicts whether a question is bridge-type. If so, the full BDTR procedure is applied; otherwise, only a single iteration is executed. This design allows us to concentrate the full reasoning budget on questions that are more likely to benefit from bridge-aware retrieval, while reducing unnecessary token overhead on other cases. The routing prompt is provided in Figure 16 in the Appendix, and the token cost analysis is reported in Appendix A.7. The algorithm description of our BDTR is shown in Appendix A.3.

4.4 Discussion: Graph-Guided Reasoning in BDTR

Although BDTR is expressed in natural language, its reasoning process is not free-form. Instead, it is grounded in the structured retrieval space induced by the GraphRAG framework, where the query and retrieved passages form an implicit local entity–relation graph. Within this space, we adopt a graph-guided reasoning process: RC enforces an explicit multi-hop entity–relation chain, FT targets the next missing node or fact along the current path, and ST justifies the necessity of a bridge entity and its connection to the answer. Therefore, while BDTR does not directly manipulate an explicit graph structure during generation, it still leverages structure to

Method	HotpotQA	2Wiki	MuSiQue
RAPTOR			
Original	0.563	0.511	0.296
IRCOT	0.580	<u>0.599</u>	0.355
GCOT	<u>0.588</u>	0.586	<u>0.358</u>
TOG	0.560	0.535	0.297
IRGS	0.581	0.592	0.352
Our	0.602	0.661	0.390
GFM-RAG			
Original	0.546	0.672	0.279
IRCOT	<u>0.562</u>	<u>0.697</u>	0.320
GCOT	0.556	0.690	<u>0.329</u>
TOG	0.558	<u>0.697</u>	0.321
IRGS	0.560	0.693	0.307
Our	0.587	0.728	0.369
MS-GraphRAG			
Original	0.552	0.506	0.276
IRCOT	<u>0.581</u>	0.645	<u>0.362</u>
GCOT	0.560	<u>0.652</u>	0.349
TOG	0.561	0.541	0.303
IRGS	0.566	0.580	0.324
Our	0.596	0.661	0.379
HippoRAG2			
Original	0.581	0.607	0.355
IRCOT	<u>0.595</u>	<u>0.668</u>	0.403
GCOT	0.597	0.662	<u>0.410</u>
TOG	0.590	0.630	0.374
IRGS	0.593	0.652	0.404
Our	0.608	0.679	0.422
Ave. Improvement	2.90%	4.32%	7.18%

Table 1: EM performance across multi-hop QA datasets. Highlighted are the results ranked **first** and **second** across all backbones. Within each backbone, the **best** and **second-best** results are indicated by boldface and underline, respectively.

guide relational path exploration, bridge validation, and evidence refinement.

5 Experiments

In our experiments, we aim to answer the following research questions: **RQ1**: How effective is the proposed BDTR framework when applied to state-of-the-art GraphRAG backbones for multi-hop QA? **RQ2**: How does BDTR compare with other iterative retrieval approaches? **RQ3**: What is the impact of the two core modules, DTR and BGEC, on overall performance?

5.1 Experimental Settings

Datasets. We evaluate on three multi-hop QA datasets—HotpotQA (Yang et al., 2018), 2WikiMultiHopQA, and MuSiQue (Trivedi et al., 2022)—sampling 1,000 queries from each following HippoRAG (Jimenez Gutierrez et al., 2024). To assess performance in the single-hop setting, we additionally sample 1,000 queries from PopQA (Mallen et al., 2023) using the same December 2021 Wikipedia corpus; results are reported in Appendix A.8.

GraphRAG Backbones. We integrate our method with four representative GraphRAG variants: HippoRAG2 (Gutiérrez et al., 2025), RAPTOR (Sarathi et al., 2024), GFM-RAG (Luo et al., 2026), and MS-GraphRAG (Edge et al., 2024).

Iterative Baselines. We compare against IRCOT (Trivedi et al., 2023), IRGS (Shao et al., 2023), TOG (Sun et al., 2024), and GCOT (Jin et al., 2024). Prompt templates are shown in figs. 10 to 13.

Implementation and Metrics. All methods use GPT-4o-mini for iterative reasoning, verification, and answer generation. We report EM and F1, and set the number of iterations to 2 for all experiments. Due to space limitations, we present the detailed experimental settings in Appendix A.4.

5.2 Main Results

In this section, we evaluate the performance of our proposed BDTR framework with various GraphRAG backbones and compare it against iterative retrieval baselines.

RQ1: QA Performance Comparison. We apply BDTR to different GraphRAG variants, including HippoRAG2, RAPTOR, GFM-RAG, and MS-GraphRAG. The results in Table 1 show that BDTR consistently improves performance across all backbones. Due to space limitations, the detailed results, particularly the F1 scores, are reported in Table 16 in the Appendix. On average, our method achieves an improvement of 11.8% in EM over HippoRAG2, 22.7% in EM over RAPTOR, and 15.93% in EM over GFM-RAG across three multi-hop QA datasets. These consistent gains demonstrate that BDTR is broadly effective across different retrieval paradigms, reinforcing its adaptability to diverse reasoning strategies in multi-hop QA.

RQ2: Comparison with other iterative methods. We further compare BDTR with other state-of-the-art iterative retrieval strategies, including

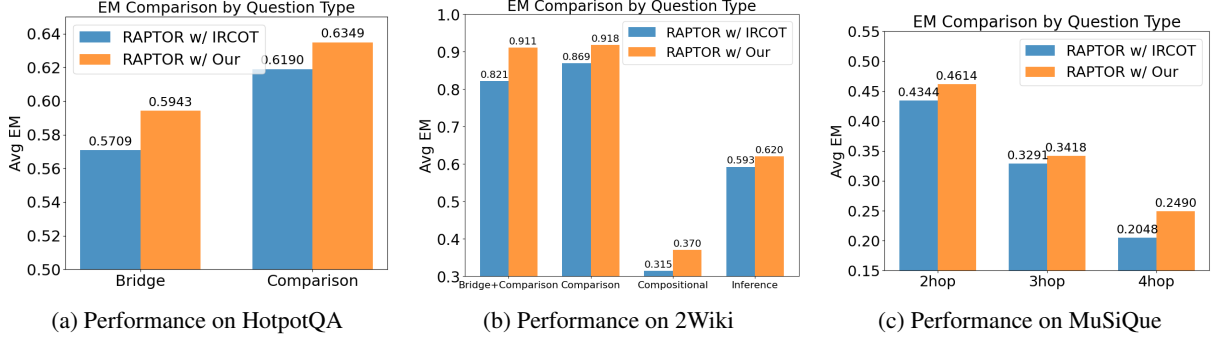


Figure 6: EM comparison on multi-hop QA datasets with different question type using the RAPTOR backbone.

IRCOT, IRGS, TOG, and GCOT. As shown in Table 1, BDTR consistently outperforms all baselines across datasets and backbones. For instance, compared to baselines, our method achieves an average improvement of 2.90% in EM on HotpotQA. Against baselines, BDTR improves by 4.32% in EM on 2WikiMultiHopQA and BDTR improves by 7.18% in EM on MuSiQue. These results highlight that BDTR not only inherits the benefits of iterative retrieval but also addresses its limitations through bridge-guided calibration, yielding more robust and reliable improvements.

Methods	Recall@2	Recall@5
HippoRAG2 w/ IRCOT	0.8300	0.9725
HippoRAG2 w/ Our	0.8925	0.9815

Table 2: Retrieval performance on HotpotQA

In addition, we analyze performance across different question types. As shown in Figure 6, with the RAPTOR backbone, our method achieves the largest gains on bridge-type questions and consistently improves performance across diverse multi-hop question types. Moreover, it alleviates the failure of standard iterative retrieval on comparison questions. The corresponding results with the HippoRAG2 backbone are reported in Figure 21 in the Appendix. Furthermore, we report the retrieval performance on HotpotQA in Table 2, while the results on 2WikiMultiHopQA and MuSiQue are provided in Tables 4 and 5 in the Appendix. The results show that our method achieves higher Recall@2 and Recall@5, demonstrating its ability to rank gold documents in the leading positions.

Remark. We postpone more results in Appendix as follows. The efficiency analysis is provided in Appendix A.7. Results on the single-hop dataset are reported in Appendix A.8. We additionally con-

duct experiments on a broader range of datasets, with results reported in Appendix A.9. To further validate the effectiveness of our approach, we additionally evaluate it using stronger models, including DeepSeek-V3.1 and GPT-5.2, beyond the GPT-4o-mini model. The corresponding results are presented in Appendix A.11. Moreover, we provide a deeper analysis of the differences between GraphRAG and standard RAG in Appendix A.12. We also present a case study in Figure 15. In addition, we report a sensitivity analysis of the $\mu + \sigma$ threshold in Appendix A.14 and an analysis of the verifier’s precision and recall in Appendix A.15.

5.3 Ablation Study

Methods	EM	F1
RAPTOR	0.296	0.418
RAPTOR w/ DTR	0.361	0.485
RAPTOR w/ BGEC	0.384	0.516
RAPTOR w/ BDTR	0.390	0.530

Table 3: Ablation Study.

To understand the contribution of each component and address **RQ3**, we conduct an ablation study on the two core modules of BDTR: Dual-Thought-based Retrieval (DTR) and Bridge-Guided Evidence Calibration (BGEC). The results are summarized in Table 3, showing that all components contribute to the overall performance, with detailed analyses provided in the Appendix A.6.

6 Conclusion

In this work, we presented the first systematic study of iterative retrieval in GraphRAG. Our analysis shows that iteration can promote bridge evidence and improve multi-hop reasoning, but it still leaves some gold documents buried too deep

to be effectively used. To overcome this bottleneck, we proposed *Bridge-Guided Dual-Thought-based Retrieval (BDTR)*, which consistently improves performance and provides guidance for future GraphRAG systems.

Limitations

While our method demonstrates strong performance across multiple benchmarks, several limitations remain. First, our approach relies on the quality of the underlying retrieval and reasoning backbone; errors in earlier stages may affect downstream performance. Second, how large language models internally utilize intermediate reasoning signals during iterative retrieval is still not fully understood. Moreover, our evaluation focuses on benchmark datasets, and assessing robustness under noisy or dynamically evolving corpora is an important direction for future research.

Acknowledgments

Kai Guo, Xinnan Dai, Shenglai Zeng, Haoyu Han and Hui Liu are supported by the National Science Foundation (NSF) under grant numbers CNS2321416, IIS2212032, IIS2212144, IIS 2504089, DUE2234015, CNS2246050, DRL2405483 and IOS2035472, the Michigan Department of Agriculture and Rural Development, US Dept of Commerce, Gates Foundation, Amazon Faculty Award, Meta, NVIDIA, Microsoft and SNAP.

References

- Orlando Ayala and Patrice Bécard. 2024. [Reducing hallucination in structured outputs via retrieval-augmented generation](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track, NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 228–238. Association for Computational Linguistics.
- Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitanaky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.
- Wenqi Fan, Yajuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 6491–6501.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).
- Kai Guo, Harry Shomer, Shenglai Zeng, Haoyu Han, Yu Wang, and Jiliang Tang. 2025. Empowering graphrag with knowledge filtering and integration. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25450–25464.
- Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*.
- Haoyu Han, Li Ma, Yu Wang, Harry Shomer, Kai Guo, Yongjia Lei, Zhisheng Qi, Zhigang Hua, Bo Long, Hui Liu, and 1 others. 2026. Rag vs. graphrag: A systematic evaluation and key insights. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 8966–8977.
- Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A Rossi, Subhabrata Mukherjee, Xianfeng Tang, and 1 others. 2024. Retrieval-augmented generation with graphs (graphrag). *arXiv preprint arXiv:2501.00309*.
- Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *Advances in Neural Information Processing Systems*, 37:132876–132907.
- Xinke Jiang, Rihong Qiu, Yongxin Xu, Yichen Zhu, Ruizhe Zhang, Yuchen Fang, Chu Xu, Junfeng Zhao, and Yasha Wang. 2024. Ragraph: A general retrieval-augmented graph learning framework. *Advances in Neural Information Processing Systems*, 37:29948–29985.
- Zhouyu Jiang, Mengshu Sun, Lei Liang, and Zhiqiang Zhang. 2025. Retrieve, summarize, plan: Advancing multi-hop question answering with an iterative approach. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 1677–1686.
- Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in Neural Information Processing Systems*, 37:59532–59569.
- Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Zheng Li, Ruirui Li, Xianfeng Tang, Suhang Wang, Yu Meng, and 1 others. 2024. Graph chain-of-thought: Augmenting large language models by reasoning on graphs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 163–184.

- Zhicheng Lee, Shulin Cao, Jinxin Liu, Jiajie Zhang, Weichuan Liu, Xiaoyin Che, Lei Hou, and Juanzi Li. 2025. Rearag: Knowledge-guided reasoning enhances factuality of large reasoning models with iterative retrieval augmented generation. *arXiv preprint arXiv:2503.21729*.
- 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.
- Haoran Luo, Guanting Chen, Qika Lin, Yikai Guo, Fangzhi Xu, Zemin Kuang, Meina Song, Xiaobao Wu, Yifan Zhu, Luu Anh Tuan, and 1 others. 2025. Graph-r1: Towards agentic graphrag framework via end-to-end reinforcement learning. *arXiv preprint arXiv:2507.21892*.
- Linhao Luo, Zicheng Zhao, Reza Haffari, Dinh Phung, Chen Gong, and Shirui Pan. 2026. Gfm-rag: graph foundation model for retrieval augmented generation. *Advances in Neural Information Processing Systems*, 38:36371–36405.
- Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 9802–9822.
- Costas Mavromatis and George Karypis. 2025. Gnn-rag: Graph neural retrieval for efficient large language model reasoning on knowledge graphs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16682–16699.
- Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. Ragruth: A hallucination corpus for developing trustworthy retrieval-augmented language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 10862–10878. Association for Computational Linguistics.
- Ahmmad OM Saleh, Gökhan Tür, and Yucel Saygin. 2024. Sg-rag: Multi-hop question answering with large language models through knowledge graphs. In *Proceedings of the 7th International Conference on Natural Language and Speech Processing (ICNLSP 2024)*, pages 439–448.
- Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*.
- 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. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9248–9274.
- Jiashuo Sun, Chengjin Xu, Luminyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *International Conference on Learning Representations*, volume 2024, pages 3868–3898.
- Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *arXiv preprint arXiv:2401.15391*.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 10014–10037.
- Junde Wu, Jiayuan Zhu, Yunli Qi, Jingkun Chen, Min Xu, Filippo Menolascina, Yueming Jin, and Vicente Grau. 2025. Medical graph rag: Evidence-based medical large language model via graph retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28443–28467.
- Peng Xia, Kangyu Zhu, Haoran Li, Tianze Wang, Weijia Shi, Sheng Wang, Linjun Zhang, James Y Zou, and Huaxiu Yao. 2025. Mmed-rag: Versatile multimodal rag system for medical vision language models. In *International Conference on Learning Representations*, volume 2025, pages 66188–66217.
- Zhishang Xiang, Chuanjie Wu, Qinggang Zhang, Shengyuan Chen, Zijin Hong, Xiao Huang, and Jinsong Su. 2026. When to use graphs in rag: A comprehensive analysis for graph retrieval-augmented generation. In *International Conference on Learning Representations*, volume 2026, pages 66145–66178.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.
- Deyu Zou, Yongqiang Chen, Mufei Li, Siqi Miao, Chenxi Liu, Bo Han, James Cheng, and Pan Li.

2025. Weak-to-strong graphrag: Aligning weak retrievers with large language models for graph-based retrieval augmented generation. *arXiv preprint arXiv:2506.22518*.

A Appendix

A.1 Related work

GraphRAG. Graph-based retrieval-augmented generation (GraphRAG) has emerged as a promising paradigm for improving large language models on multi-hop question answering by grounding reasoning in structured entity–relation graphs. A variety of methods have been proposed to integrate graph structure into retrieval. HippoRAG2 (Gutiérrez et al., 2025) adopts personalized PageRank (PPR) to expand retrieval around entity mentions and uncover distant yet relevant nodes. RAPTOR (Sarathi et al., 2024) introduces a tree-based hierarchical organization that recursively summarizes evidence at different levels, balancing efficiency with global context coverage. GFM-RAG (Luo et al., 2026) leverages graph neural networks (GNNs) to encode structural dependencies and propagate information across neighbors, thereby capturing higher-order relations that are crucial for multi-hop reasoning. Meanwhile, MS-GraphRAG (Edge et al., 2024) applies community detection to partition large knowledge graphs into semantically coherent clusters, reducing noise and emphasizing connections among related entities.

Iterative Retrieval. Iterative retrieval has been widely explored as a means to enhance RAG by dynamically refining the evidence set in multiple rounds of reasoning and retrieval. IRCOT (Trivedi et al., 2023) interleaves retrieval with chain-of-thought reasoning, allowing the model to iteratively issue new queries guided by intermediate reasoning steps. IRGS (Shao et al., 2023) emphasizes the synergy between retrieval and generation, where iterative refinement of both components leads to stronger evidence grounding. TOG (Sun et al., 2024) extends iterative retrieval to the graph setting, enabling LLMs to progressively navigate and reason over knowledge graphs. GCOT (Jin et al., 2024) integrates graph structures into the chain-of-thought process, combining step-by-step reasoning with structured retrieval to capture multi-hop dependencies more effectively. Collectively, these approaches demonstrate the importance of iterative retrieval as a general strategy to mitigate missing evidence and strengthen multi-step reasoning in complex QA tasks. Recent systems

such as GFM-RAG (Luo et al., 2026) and HippoRAG (Jimenez Gutierrez et al., 2024) incorporate only limited use of iterative retrieval. Nevertheless, how iterative retrieval strategies function within the GraphRAG framework remains largely unexplored.

A.2 Summary of Observations

Our preliminary studies reveal both the opportunities and the challenges of applying iterative retrieval in GraphRAG.

Opportunities. **1)** Iterative retrieval consistently enhances performance on complex multi-hop questions, particularly Bridge-type questions that require identifying implicit intermediate entities. **2)** The performance gains can be attributed to the reranking effect of iterative retrieval, which improves recall at small cutoffs (e.g., Recall@5, Recall@10) and thereby increases the likelihood that critical facts are utilized in reasoning. **3)** Different iterative strategies (e.g., IRCOT vs. GCOT) demonstrate complementary strengths, suggesting that combining or adaptively selecting among them could further improve coverage and robustness.

Pitfalls. **1)** Iterative retrieval offers little to no benefit on simple Comparison questions, and in some cases even reduces performance due to overthinking. **2)** Increasing the number of rounds beyond two generally leads to diminishing returns, reflecting an efficiency–effectiveness trade-off where additional complexity brings little incremental benefit. **3)** Although most gold documents—including bridge documents—can be retrieved, many fail to appear within the leading positions, thereby limiting their practical usefulness for reasoning.

A.3 Algorithm

The overall process of **Bridge-Guided Dual-Thought-based Retrieval (BDTR)** is summarized in Algorithm 1. It integrates the two key components introduced above: Dual-Thought-based Retrieval (DTR) for coverage expansion and Bridge-Guided Evidence Calibration (BGEC) for ranking calibration. In particular, Lines 4–8 correspond to DTR, where dual thoughts generate complementary retrieval signals and their results are merged into a shared pool. Lines 9–13 correspond to BGEC, where bridge-supporting documents are identified via the reasoning chain and re-ranked, followed by statistical filtering to form the final evidence set.

Algorithm 1: BDTR: Bridge-Guided Dual-Thought-based Retrieval

Input: question Q , backbone retriever f_{ret} , number of iterations R

Output: final document set $\mathcal{D}_{\text{final}}$

```
1:  $y \leftarrow \text{Router}(Q)$ ; // Predict bridge-type or not
if  $y = 0$  then
    2:  $R \leftarrow 1$ ; // Single iteration for non-bridge questions
3:  $P_0 \leftarrow f_{\text{ret}}(Q)$ ; // Initial retrieval
for  $t = 1$  to  $R$  do
    4: Generate two complementary queries  $q_t^{\text{FT}}, q_t^{\text{ST}}$ ;
    5:  $D_t^{\text{FT}} \leftarrow f_{\text{ret}}(q_t^{\text{FT}})$ ;
    6:  $D_t^{\text{ST}} \leftarrow f_{\text{ret}}(q_t^{\text{ST}})$ ;
    7:  $P_t \leftarrow P_{t-1} \cup D_t^{\text{FT}} \cup D_t^{\text{ST}}$ ;
    8: Update scores and resort:  $s_t(d) \leftarrow \max(s_{t-1}(d), \hat{s}(d | q_t^{\text{FT}}), \hat{s}(d | q_t^{\text{ST}}))$  for  $d \in P_t$ ;
9: Generate reasoning chain  $RC$  from final pool  $P_R$ ;
10:  $\mathcal{G} \leftarrow \{d \in P_R \mid \text{Verifier}(d, RC) = 1\}$ ; // Bridge docs selected by LLM
11: Promote  $\mathcal{G}$  to the top of  $P_R$ ;
12: Compute  $\mu, \sigma$  from top-30 scores in  $P_R$ ;
13:  $\mathcal{D}_{\text{final}} \leftarrow \{d \in P_R \mid s(d) \geq \mu + \sigma\}$ , ensuring  $|\mathcal{D}_{\text{final}}| \geq 5$ ;
return  $\mathcal{D}_{\text{final}}$ ;
```

A.4 Experimental Settings

Datasets. We evaluate our approach on three widely used multi-hop QA benchmarks: HotpotQA (Yang et al., 2018), 2WikiMultiHopQA, and MuSiQue (Trivedi et al., 2022). Following HippoRAG (Jimenez Gutierrez et al., 2024), we randomly sample 1,000 queries from each dataset. Since all of these benchmarks require reasoning across multiple evidence documents, they provide an appropriate setting for assessing the effectiveness of iterative retrieval strategies. In addition, we evaluate on a single-hop dataset to examine the effectiveness of both the baselines and our method. Specifically, we sample 1,000 queries from PopQA (Mallen et al., 2023), a **single-hop dataset**, using the corpus constructed from the December 2021 Wikipedia dump as in HippoRAG (Jimenez Gutierrez et al., 2024). The corresponding results are reported in the Appendix A.8.

Methods	Recall@2	Recall@5
HippoRAG2 w/ IRCOT	0.7935	0.9860
HippoRAG2 w/ Our	0.8622	0.991

Table 4: Retrieval performance on 2WikiMultiHopQA

Methods	Recall@2	Recall@5
HippoRAG2 w/ IRCOT	0.5468	0.7920
HippoRAG2 w/ Our	0.5571	0.8016

Table 5: Retrieval performance on MuSiQue

GraphRAG Backbones. We integrate our method with four representative GraphRAG variants: HippoRAG2 (Gutiérrez et al., 2025) (PPR-based), RAPTOR (Sarathi et al., 2024) (tree-based), GFM-RAG (Luo et al., 2026) (GNN-based), and MS-GraphRAG (Edge et al., 2024) (community-based), covering diverse retrieval paradigms.

Iterative Baselines. We compare BDTR against state-of-the-art iterative methods, including IRCOT (Trivedi et al., 2023), IRGS (Shao et al., 2023), TOG (Sun et al., 2024), and GCOT (Jin et al., 2024), to test whether our bridge-guided design offers advantages beyond existing strategies. The corresponding prompts are illustrated in figs. 10 to 13.

Implementation and Evaluation Metrics. For implementation, we adopt the official codebases of HippoRAG2 and GFM-RAG, and reimplement RAPTOR and MS-GraphRAG within the HippoRAG2 framework. Across all methods, GPT-4o-mini is used as the iterative reasoning engine, the verifier, and the generator for producing answers. We report two widely used metrics: Exact Match (EM) and F1. EM measures the percentage of predictions that exactly match the ground-truth answers, providing a strict indicator of correctness. F1 measures the token-level overlap between the predicted and gold answers, offering a softer evaluation that captures partial correctness. Together, EM and F1 provide a balanced view of QA performance in terms of both precision and recall. For all experiments, the number of iteration is set to 2.

A.5 Impact of the Number of Rounds

An important design choice in iterative retrieval is how many rounds to perform. On the one hand, additional rounds may surface evidence that is missed initially; on the other hand, excessive rounds risk

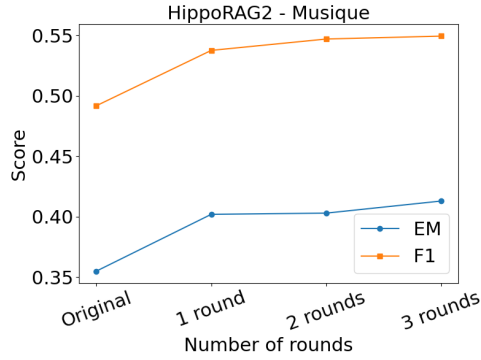


Figure 7: Effect of iteration rounds

introducing redundancy or irrelevant documents. We therefore examine how performance evolves as the number of rounds increases.

As shown in Figure 7, moving from a single round to two rounds of IRCOT yields substantial improvements, demonstrating that two additional passes is often sufficient to retrieve the missing bridge evidence. However, extending to three or more rounds provides only diminishing returns, suggesting that two iterations achieve the most favorable cost–benefit balance and nearly reach convergence.

A.6 Ablation Study

Methods	EM	F1
RAPTOR	0.296	0.418
RAPTOR w/ DTR	0.361	0.485
RAPTOR w/ BGEC	0.384	0.516
RAPTOR w/ BDTR	0.390	0.530

Table 6: Ablation Study.

To understand the contribution of each component and address **RQ3**, we conduct an ablation study on the two core modules of BDTR: Dual-Thought-based Retrieval (DTR) and Bridge-Guided Evidence Calibration (BGEC). The results are summarized in Table 6.

We observe that DTR alone brings a clear performance boost, improving EM by 21.9% and F1 by 16.0% on dataset MuSiQue. This demonstrates the effectiveness of generating complementary thoughts to enlarge the evidence frontier. BGEC further enhances performance by recalibrating the ranking based on the reasoning chain, improving EM by 29.7% and F1 by 23.6%. When both modules are combined, BDTR achieves the best results, with overall gains of 31.7% in EM and 26.5% in

2WikiMulHopQA	
IRCOT	0.85h
Our	1.30h
Our w/ parallelization	0.25h

Table 7: Comparison of running time on 2WikiMulHopQA with HippoRAG2.

2WikiMulHopQA	
IRCOT	3,391,372
Our	5,718,140

Table 8: Comparison of token cost on 2WikiMulHopQA with HippoRAG2.

F1. These findings validate that both DTR and BGEC are essential and complementary, jointly contributing to the strong performance of BDTR.

A.7 Efficiency Analysis

Table 7 reports the retrieval runtime on 2WikiMulHopQA in comparison with HippoRAG2.

Compared with IRCOT, our method introduces dual-thought generation and bridge-based evidence calibration, which increases the computational burden and leads to a modest rise in latency (1.30h vs. 0.85h). To address this overhead, we optimize the framework by parallelizing the retrieval pipeline across different queries. Concretely, each query is still processed sequentially to preserve reasoning consistency, but multiple queries are dispatched concurrently via a thread pool. This design reduces the latency of LLM calls and graph-based retrieval I/O, which are the dominant bottlenecks. As a result, the total running time is reduced from 1.3h to 0.25h, yielding a $5.2\times$ speedup, while the token cost is only $1.68\times$ that of IRCOT, as shown in Table 8.

A.8 Performance on single-hop dataset.

In addition to multi-hop datasets, we also evaluated on a single-hop dataset. The results in Table 9 show that, unlike in the multi-hop setting where iterative retrieval often yields significant improvements, these methods provide little to no benefit on single-hop questions. This finding further confirms our earlier observation that iterative retrieval is particularly suited for supporting multi-hop reasoning. On this single-hop dataset, our method achieves improvement over the baselines.

Methods	EM	F1
Original	0.419	0.5603
+IRCOT	0.425	0.5629
+GCOT	0.429	0.5662
+TOG	0.419	0.5615
+IRGS	0.426	0.5617
+Our	0.431	0.5725

Table 9: Results on Single-Hop dataset PopQA with HippoRAG2.

A.9 Results on More Datasets

We conduct additional experiments on the Medical datasets from GraphRAG-Bench (Xiang et al., 2026). These datasets cover several task types, including complex reasoning, contextual summarization, and fact retrieval. We use both LLM-as-a-judge and ROUGE-L to measure performance, following the same evaluation metric as the original paper. As shown in the Table 10, our method consistently outperforms IRCOT across all task categories.

A.10 Prompt Example

In this section, we provide the prompt examples for iterative methods: our method, IRCOT, ToG, GCOT, IRGS. They are illustrated in Fig 14, Fig 9, Fig 10, Fig 11, Fig 12 and Fig 13.

A.11 Results on DeepSeek-V3.1 and GPT-5.2

We conducted additional comparisons using the DeepSeek-V3.1 and GPT-5.2. The results in Table 14 and Table 13 show that our method consistently outperforms the baseline across different LLMs.

A.12 RAG VS. GraphRAG

We will conduct a deeper analysis of the distinctions between GraphRAG and standard RAG. We run experiments using embedding-based dense retrievers such as NV-Embed-v2. The results in the Table 11 show that our method also improves performance in this setting. On HotpotQA, the dense retriever (NV-Embed-v2 + BDTR) slightly outperforms GFM-RAG + BDTR, suggesting that dense embedding-based retrieval is sufficient for datasets with relatively linear reasoning chains. However, on 2WikiMultihopQA, GFM-RAG + BDTR achieves a substantially higher score than the dense retriever, indicating that graph-based re-

trieval remains highly advantageous for datasets requiring more complex multi-hop reasoning.

A.13 Retrieval Scoring in GraphRAG Backbones.

The retrieval scores in our framework are generated by GraphRAG-style backbones. We take HippoRAG2 as a representative example to illustrate the scoring mechanism. In this framework, the document relevance score is derived from a single Personalized PageRank computation.

Specifically, the input query assigns weights to entity nodes that appear in the most relevant facts, while passages that are semantically similar to the query are assigned Dense Passage Retrieval (DPR)-based weights. These two components jointly define the reset distribution for Personalized PageRank, which is then propagated over the entity–passage graph until convergence.

Formally, the weight assigned to an entity e is defined as

$$w^{(e)} = \frac{s_f}{c_e}, \quad (6)$$

where s_f denotes the fact–query similarity score, and c_e is the number of corpus chunks containing entity e .

Let $w^{(p)}$ denote the DPR-based weight assigned to passage p . The reset vector used in Personalized PageRank is then constructed as

$$r_i = w_i^{(e)} + w_i^{(p)}, \quad (7)$$

where r_i represents the reset probability associated with node i .

Given the reset distribution r , Personalized PageRank produces the final passage relevance score as

$$\text{score}(p) = \text{PPR}_p, \quad (8)$$

where PPR_p denotes the PageRank value of passage p under the personalized reset distribution.

A.14 Sensitivity Analysis

The use of $\mu + \sigma$ serves as a simple, distribution-aware threshold that adapts to the variance of confidence scores within each instance. It does not introduce additional hyperparameters and is easy to implement in practice. We chose this rule primarily because it is simple, stable, and works effectively without requiring additional tuning.

We conduct sensitivity analysis with alternative thresholds (e.g., $\mu + 0.5\sigma$, $\mu + \sigma$, $\mu + 1.5\sigma$). The results in Table 12 show only minor performance variation across settings.

Task	IRCOT LLM_ACC	Ours LLM_ACC	IRCOT ROUGE-L	Ours ROUGE-L
Complex Reasoning	0.8389	0.8566	0.3577	0.3600
Contextual Summarize	0.8166	0.8408	0.3569	0.3715
Fact Retrieval	0.8042	0.8342	0.3865	0.3933

Table 10: Comparison of LLM accuracy and ROUGE-L between IRCOT and our method across different task categories.

Method	HotpotQA EM	HotpotQA F1	2WikiMultiHopQA EM	2WikiMultiHopQA F1
NV-Embed-v2	0.590	0.7348	0.528	0.597
NV-Embed-v2+BDTR	0.602	0.7507	0.668	0.764
GFM-RAG+BDTR	0.585	0.7462	0.726	0.805

Table 11: Comparison of different retriever on HotpotQA and 2WikiMultiHopQA.

A.15 Precision and Recall of the LLM verifier

At each step, we provide 25 candidate documents to the verifier. Table 15 reports the precision and recall of the verifier in identifying bridge documents across datasets, showing consistently high recall and strong precision. The verifier prompt, including the reasoning chain and retrieved candidates, is shown in Figure 9.

Method	$\mu + 0.5\sigma$ (EM)	$\mu + 0.5\sigma$ (F1)	$\mu + \sigma$ (EM)	$\mu + \sigma$ (F1)	$\mu + 1.5\sigma$ (EM)	$\mu + 1.5\sigma$ (F1)
Our	0.425	0.5699	0.422	0.5683	0.425	0.5711

Table 12: Sensitivity analysis of different threshold settings.

Method	HotpotQA EM	HotpotQA F1	2Wiki EM	2Wiki F1	MuSiQue EM	MuSiQue F1
IRCOT	0.562	0.7473	0.571	0.6921	0.391	0.5458
Our method	0.570	0.7515	0.583	0.7083	0.473	0.6373

Table 13: Performance comparison on HotpotQA, 2WikiMultiHopQA, and MuSiQue using GPT-5.2.

Method	HotpotQA EM	HotpotQA F1	2Wiki EM	2Wiki F1	MuSiQue EM	MuSiQue F1
IRCOT	0.649	0.7794	0.694	0.782	0.419	0.5451
Ours	0.658	0.7882	0.710	0.796	0.469	0.5831

Table 14: Performance comparison on HotpotQA, 2WikiMultiHopQA (2Wiki), and MuSiQue using DeepSeek-V3.1.

Method	HotpotQA (P)	HotpotQA (R)	2Wiki (P)	2Wiki (R)	MuSiQue (P)	MuSiQue (R)
Our verifier	0.8503	0.9345	0.9057	0.8921	0.633	0.6451

Table 15: Precision (P) and Recall (R) of the LLM verifier in identifying bridge documents across different datasets.

Case Study	
Question:	<i>At what intersection was the former home of the wooden roller coaster now located at Six Flags Great America in Gurnee, Illinois located?</i>
Gold Answer:	North Avenue and First Avenue
Method	Retrieved Evidence → Prediction
HippoRAG2 w/ IRCOT	(Retrieved) <i>Little Dipper</i> : “relocated from Kiddieland Amusement Park.” (Retrieved) <i>Kiddieland Amusement Park</i> : “located at North Avenue and First Avenue.” → Correct : North Avenue and First Avenue
HippoRAG2	(Retrieved) <i>Little Dipper</i> : “relocated from Kiddieland Amusement Park.” (Missed) No document mentioning Kiddieland’s intersection (bridge fact missing). → Incorrect : Cannot infer the intersection

Figure 8: Case study showing the importance of retrieving *bridge facts*. With IRCOT (top), the retriever surfaces Kiddieland’s location, enabling the correct answer. Without it (bottom), the reasoning chain breaks.

Prompts

You are a careful reasoning verifier.

Reasoning Chain: {reasoning chain}

Candidate Documents: {listing}

Your task:

- Identify which documents provide direct evidence for one or more **critical nodes or bridges** in the reasoning chain (e.g., the key missing fact, entity–relation links, or decisive constraints).
- **Do not** bias toward the current rank. If a lower-ranked document strongly supports a critical node or a bridging relation, you **must** include it.
- Favor documents that:
 - explicitly answer the Direct Question node(s),
 - provide the bridge relation in the Graph Thought,
 - connect multiple nodes in the chain (multi-node coverage),
 - resolve ambiguities or disambiguate entities.
- Output **strict JSON only** with a single key "covered_doc_indices" as a list of 1-based integers.
- Order indices by **support/impact strength** (strongest first), not by rank.

Example (STRICT JSON): {{ "covered_doc_indices": [17,3,25] }}

Return STRICT JSON only. No explanations.

Figure 9: An example prompt of Bridge-based Evidence Calibration of our method.

Prompts

You serve as an intelligent assistant, adept at facilitating users through complex, multi-hop reasoning across multiple documents. This task is illustrated through demonstrations, each consisting of a document set paired with a relevant question and its multi-hop reasoning thoughts.

Your task: Generate **one reasoning thought for the current step**.

- Do not generate the entire reasoning chain at once.
- At each step, provide only a single intermediate thought that advances the reasoning.
- If you believe you have reached the final step, begin your response with: "So the answer is:".

Figure 10: An example prompt of IRCOT.

Prompts

You serve as an intelligent assistant, adept at facilitating users through complex, multi-hop reasoning across multiple documents. This task is illustrated through demonstrations, each consisting of a document set paired with a relevant question and its multi-hop reasoning thoughts.

Your task: Evaluate whether the given information is sufficient to answer the question.

- If the evaluation is positive, start the response with: "So the answer is:".
- Otherwise, explain what additional information would be required.

Figure 11: An example prompt ToG

Prompts

You serve as an intelligent assistant, adept at facilitating users through complex, multi-hop reasoning across multiple documents. This task is illustrated through demonstrations, each consisting of a document set paired with a relevant question and its multi-hop reasoning thoughts.

Your task: Think step by step about what additional information is required for the current step.

- Do not generate the full reasoning or the final answer at once.
- At each step, only articulate what is still missing or what bridge evidence should be retrieved next.
- If you reach what you believe to be the final step, begin your response with: "So the answer is:".

Figure 12: An example prompt of GCOT.

Prompts

Based on the following documents, answer the question with concise reasoning and a final answer. Keep the reasoning under 100 English words.

Documents: {context}

Question: {query}

Please provide:

1. Brief reasoning based on the documents
2. Final answer

Figure 13: An example prompt of IRGS

Prompt

You are a “Graph Reasoning Agent”. Your goal is to navigate a hidden Knowledge Graph to answer complex questions. You must identify the “Bridge Entities” that connect the User’s Question to the Answer.

For every turn, analyze the provided text and produce three specific outputs:

1. Reasoning Chain (The Map):

- Construct a valid entity path based on the context.
- Format: (Entity A) –[relation]–> (Entity B) –[relation]–> (Target).
- CONSTRAINT: Every entity in the chain must be explicitly related to the context. Do not hallucinate links.

2. Fast Thought (The Next Step):

- A specific, short question targeting the *next missing node* in your chain.
- Example: “Where is [Entity B] located?” (instead of “Where is the answer?”)

3. Slow Thought (The Topology):

- Explain why you are asking the Fast Thought based on graph structure.
- Format: “I have identified [Entity A]. To find [Target], I must first retrieve the neighbor [Entity B] which acts as a bridge.”

Rules:

- Output format: exactly three lines in this order:
 1. Reasoning Chain: ...
 2. Fast Thought: ...
 3. Slow Thought: ...

Figure 14: Prompt template of our method.

Framework	Method	HotpotQA		2WikiMultiHopQA		MuSiQue	
		EM	F1	EM	F1	EM	F1
RAPTOR	Original	0.563	0.7080	0.511	0.5726	0.296	0.4179
	IRCOT	0.580	<u>0.7266</u>	<u>0.599</u>	<u>0.6870</u>	0.355	<u>0.4905</u>
	GCOT	<u>0.588</u>	0.7261	0.586	0.6814	<u>0.358</u>	0.4882
	TOG	0.560	0.7041	0.535	0.6062	0.297	0.4229
	IRGS	0.581	0.7262	0.592	0.6833	0.352	0.4779
	Our	0.602	0.7480	0.661	0.7562	0.390	0.5297
GFM-RAG	Original	0.546	0.6820	0.672	0.7546	0.279	0.3982
	IRCOT	<u>0.562</u>	<u>0.7066</u>	<u>0.697</u>	0.7768	0.320	0.4440
	GCOT	0.556	0.6924	0.690	0.7711	<u>0.329</u>	<u>0.4484</u>
	TOG	0.558	0.7029	<u>0.697</u>	<u>0.7847</u>	0.321	0.4369
	IRGS	0.560	0.7184	0.693	0.7745	0.307	0.4380
	Our	0.587	0.7464	0.728	0.8103	0.369	0.4976
MS-GraphRAG	Original	0.552	0.6983	0.506	0.5757	0.276	0.4017
	IRCOT	<u>0.581</u>	<u>0.7283</u>	0.645	0.7533	<u>0.362</u>	<u>0.5060</u>
	GCOT	0.560	0.7041	<u>0.652</u>	<u>0.7582</u>	0.349	0.4917
	TOG	0.561	0.7080	0.541	0.6196	0.303	0.4309
	IRGS	0.566	0.7126	0.580	0.6842	0.324	0.4609
	Our	0.596	0.7458	0.661	0.7593	0.379	0.5300
HippoRAG2	Original	0.581	0.7372	0.607	0.7059	0.355	0.4917
	IRCOT	<u>0.595</u>	<u>0.7493</u>	<u>0.668</u>	<u>0.7683</u>	0.403	<u>0.5469</u>
	GCOT	0.597	0.7491	0.662	0.7652	<u>0.410</u>	0.5417
	TOG	0.590	0.7381	0.630	0.7330	0.374	0.5352
	IRGS	0.593	0.7484	0.652	0.7546	0.404	0.5436
	Our	0.608	0.7570	0.679	0.7731	0.422	0.5683
Ave. Improvement		2.90%	3.00%	4.32%	3.39%	7.18%	6.90%

Table 16: EM and F1 performance across multi-hop QA datasets. Each framework is evaluated with different iterative retrieval methods. Highlighted are the results ranked **first** and **second** across all backbones. Within each backbone, the **best** and second-best results are indicated by boldface and underline, respectively.

Successful Case	
Question	What year did the Governor of the city where the basilica named after the same saint as the one that Mantua Cathedral is dedicated to die?
Gold Answer	1952
Reasoning Chain	(Mantua Cathedral) –[dedicated to]–> (Saint Peter) –[associated city]–> (Vatican City) –[governed by]–> (Governor of Vatican City)
Prediction Answer	1952.
Find the bridges	Bridge 1: Confirmed that Mantua Cathedral is dedicated to Saint Peter. Bridge 2: Identified that St. Peter’s Basilica is located in Vatican City. Bridge 3: Determined that the Governor of Vatican City (Camillo Serafini) held the post until his death in 1952.

Failure Case	
Question	What is the place of birth of who does the voice of Stan on the series that <i>Insecurity</i> is part of?
Gold Answer	Denver
Reasoning Chain	(Insecurity) –[part of]–> (South Park) –[voiced by]–> (Trey Parker) –[born in]–> (Place of Birth)
Prediction Answer	Not specified in the text.
Missing Evidence	The reasoning chain is logically valid, but the retrieved evidence does not contain the final-hop fact needed to answer the question (Trey Parker’s place of birth). Without an explicit statement linking Trey Parker to “Denver,” the model cannot verify the answer and therefore outputs “Not specified.”

Figure 15: Success and failure cases of our method. Explicit modeling of bridge facts helps reduce reasoning failures.

Prompts

You are a multi-hop question type classifier.

Task: Classify the following question into exactly one of two categories:

- **bridge:** The question requires finding an intermediate (bridge) entity first, then using that entity to obtain the final answer. The reasoning follows a chain-like structure:

$$Q \rightarrow \text{entity A} \rightarrow \text{entity B} \rightarrow \text{answer}$$

Examples:

- “The fictional private detective that appears in ‘The Adventure of the Seven Clocks’ was written by whom?” (first identify the detective, then find the author)
- “What is one of the stars of The Newcomers known for?” (first identify the star, then determine what they are known for)
- “What government position was held by the woman who portrayed Nora Batty?” (first identify the actress, then find the government position)
- **comparison:** The question asks to compare properties or attributes of two or more explicitly named entities. The compared entities are already given in the question, and no bridge entity needs to be discovered. Examples:
 - “Are Colocasia and Coronilla both flowering plants?”
 - “Which is located in Cornwall, Southwell Minster or Truro Cathedral?”
 - “Which film has the director born first, Avatar or Titanic?”
 - “Are both X and Y from the same country?”

Question: {query}

Output requirement: Respond with **only one word**: bridge or comparison.

Figure 16: The prompt used for question-type routing in our method. A lightweight model first classifies each query as either bridge or **simple** comparison, and the prediction determines whether the full BDTR pipeline is applied.

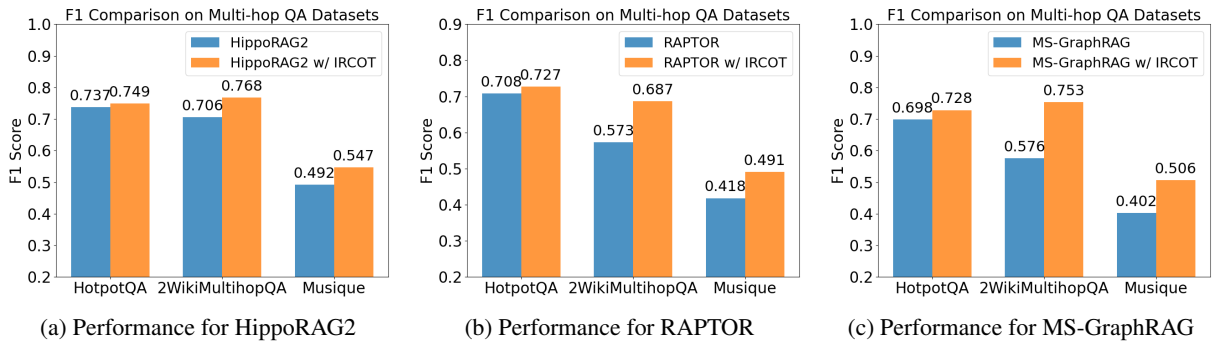
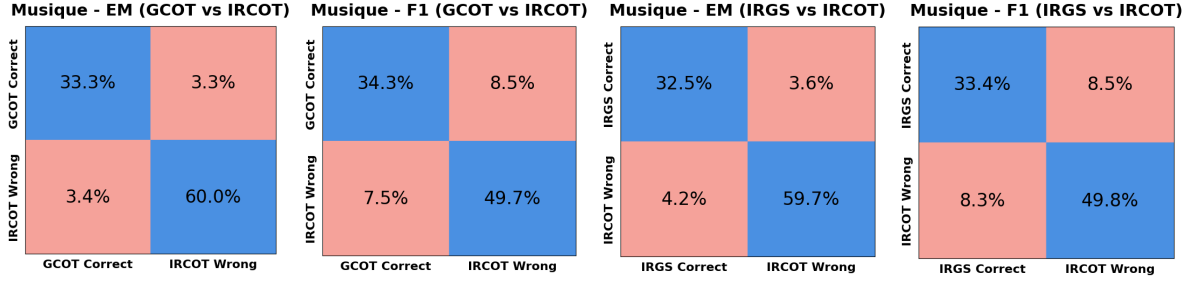


Figure 17: F1 Comparison on Multi-hop QA Datasets.



(a) Complementarity between GCOT and IRCOT under EM (b) Complementarity between GCOT and IRCOT under F1 (c) Complementarity between IRGS and IRCOT under EM (d) Complementarity between IRGS and IRCOT under F1

Figure 18: A complementary study of GCOT/IRGS and IRCOT under EM and F1 on multi-hop QA datasets.

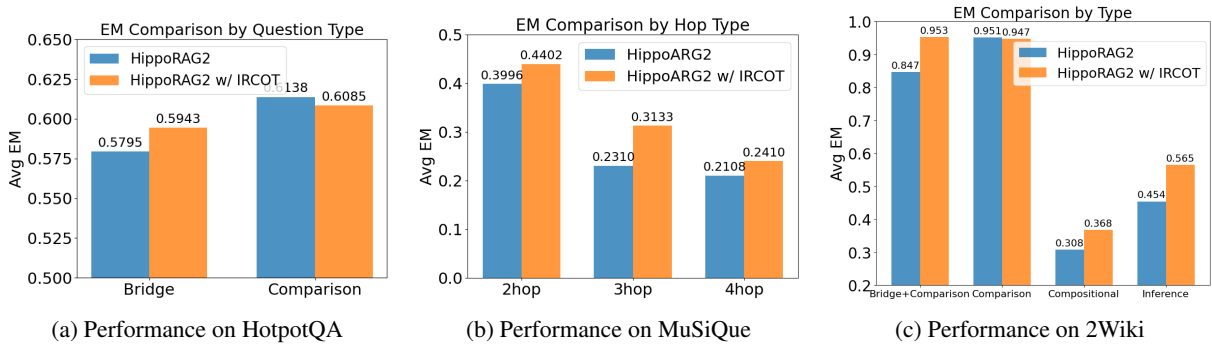


Figure 19: EM Comparison by Question Type with the HippoRAG2 Backbone.

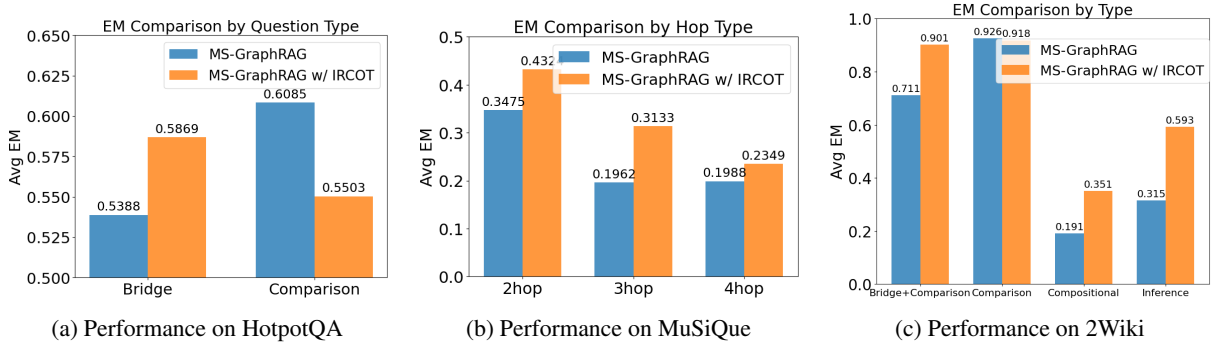


Figure 20: EM Comparison by Question Type with the MS-GraphRAG Backbone.

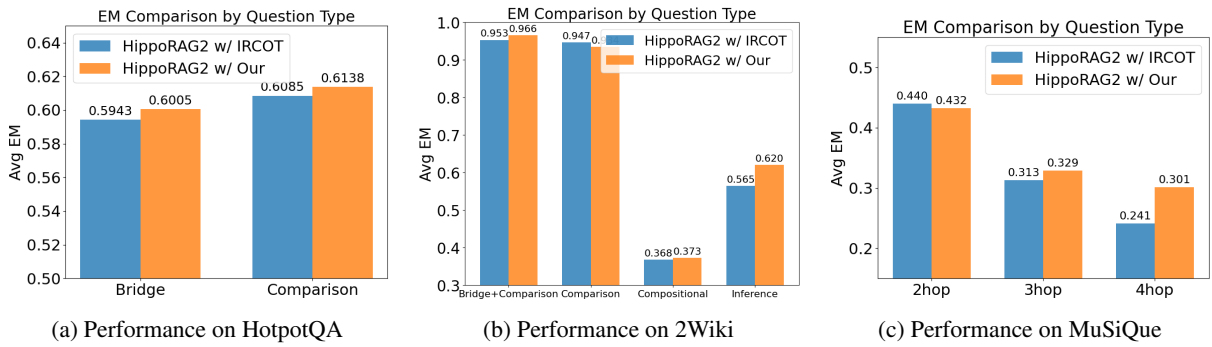


Figure 21: EM comparison on multi-hop QA datasets with different question type using the HippoRAG2 backbone.