PathCoRAG: Multi-step Reasoning with Path-Aware CoT Expansion for RAG

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have significantly advanced natural language understanding, yet they often struggle with complex, multi-step reasoning due to limitations in fixed-context knowledge access. Retrieval-Augmented Generation (RAG) frameworks address this by incorporating external knowledge, but conventional methods typically retrieve flat, chunk-level text without respecting the logical structure of reasoning, leading to fragmented and noisy contexts. We introduce PathCoRAG, a novel RAG framework that explicitly aligns multi-step reasoning with path-aware retrieval and context construction. Unlike prior methods, PathCoRAG performs step-wise query expansion and retrieves nodes and paths corresponding to each reasoning step. This produces a logic-preserving, sequential context structure that guides the LLM through a structured chain of thought during generation. Our approach consists of four tightly integrated components: (1) Chain-of-Thought-based Query Expansion, (2) Hierarchical Node Extraction per reasoning step, (3) Semantic Path Exploration and Scoring, and (4) Structured Context Prompting aligned with logical reasoning paths. Experimental results across diverse domains show that PathCoRAG consistently outperforms strong baselines. https://anonymous.4open.science/r/PathCoRAG-A1BB

1 Introduction

Large Language Models (LLMs) excel at natural language understanding and generation but struggle with complex, domain-specific queries that require real-time information retrieval and multi-step reasoning (Wei et al., 2022; Yao et al., 2023). To address this, Retrieval-Augmented Generation (RAG) systems have been developed (Lewis et al., 2020; Gao et al., 2023b; Peng et al., 2024), enabling LLMs to retrieve relevant information from large external document collections, improving factual accuracy and domain adaptability. However, conventional RAG approaches often rely on flat data structures, making it challenging to capture complex entity relationships (Izacard and Grave, 2021), which often results in responses where related evidence appears disjointed and lacks a coherent flow.

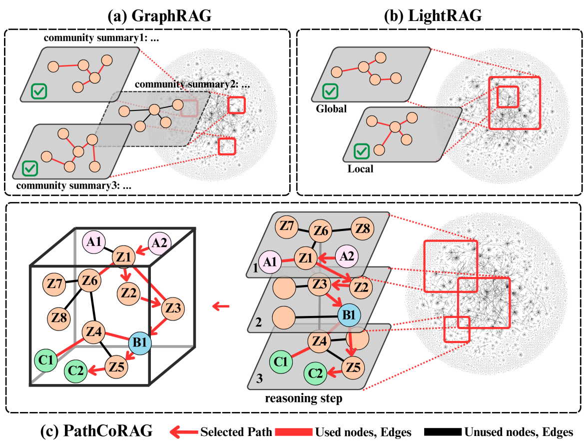

Figure 1: Comparison of information retrieval mechanisms in GraphRAG, LightRAG, and PathCoRAG. GraphRAG groups related entities into hierarchical subgraphs, LightRAG separates local and global contexts for more efficient retrieval, while PathCoRAG introduces path-aware multi-step reasoning to capture deeper semantic connections and reduce redundancy.

To address these challenges, graph-based RAG methods have been proposed (Edge et al., 2024; Liu et al., 2025), which explicitly model relationships between entities to enable more structured multi-step reasoning. As illustrated in Figure 1, GraphRAG (Edge et al., 2024) uses hierarchical subgraphs to group related entities.(Figure 1(a)) improving context aggregation but sometimes diluting critical relationships due to loosely connected nodes. LightRAG (Guo et al., 2024) further improves retrieval efficiency by separating local and global contexts.(Figure 1(b)) but it still relies on keyword-based retrieval, which can miss deeper semantic connections (Wei et al., 2022).

To overcome these limitations, we propose Path-

CoRAG, a novel graph-based RAG framework that explicitly integrates multi-step reasoning with path-aware retrieval and context construction (Figure 1(c)). PathCoRAG systematically decomposes complex queries into structured reasoning steps using Chain-of-Thought (CoT) prompting. For each step, it performs targeted node retrieval through hierarchical matching, capturing semantically relevant nodes while avoiding spurious aggregation. It then explores possible logical paths between these nodes and selects those that reflect coherent, step-aligned reasoning.

Unlike prior methods that retrieve and aggregate nodes in a flat or unstructured fashion, PathCoRAG transforms selected nodes and edges into structured context sequences such as: [Node A (desc)] $\rightarrow$ [Edge (desc)] $\rightarrow$ [Node B (desc)], thereby reducing redundancy and preserving the logical reasoning flow. This context structure is directly passed to the LLM, guiding it through a step-wise, semantically connected chain-of-thought for answer generation.

We evaluated PathCoRAG on diverse domains including Agriculture, Computer Science, Legal, and a mixed set, comparing it against strong baselines such as NaiveRAG (Gao et al., 2023b), HyDE (Gao et al., 2023a), LightRAG (Guo et al., 2024), GraphRAG (Edge et al., 2024), and PathRAG (Chen et al., 2025). Across multiple human evaluation metrics such as Comprehensiveness, Diversity, and Empowerment (Edge et al., 2024; Guo et al., 2024), PathCoRAG consistently achieved superior performance.

- **Step-wise Reasoning and Retrieval Integration.** PathCoRAG tightly couples CoT-based query decomposition with step-specific node and path retrieval, enabling logical precision and reducing context fragmentation.
- **Path-Aware Structured Context Prompting.** Retrieved paths are serialized into structured, logic-aligned text representations that maintain semantic flow while removing redundancy.
- **Superior Performance on Multi-Step Reasoning Tasks.** PathCoRAG consistently outperforms prior RAG systems across domains and metrics, demonstrating its strength in handling complex, reasoning-intensive queries.

2 Related Work

2.1 Traditional RAG

Traditional RAG approaches, such as those introduced by Lewis et al. (2020), typically leverage **dense retrieval** (e.g., DPR) (Karpukhin et al., 2020) and **sparse retrieval** (e.g., BM25) (Robertson and Zaragoza, 2009) to access relevant documents from large external corpora. While they improve factual grounding and domain adaptation (Gao et al., 2023b), these methods generally rely on flat structures that struggle to capture complex entity relationships (Tang and Yang, 2024), often resulting in fragmented or inconsistent responses. Additionally, they lack mechanisms to filter redundant or noisy information, which can impair contextual consistency and increase latency at scale. These limitations have motivated the development of more structured approaches like graph-based RAG systems.

2.2 Graph-Based RAG

Graph-based RAG frameworks model entities and their relations as graphs to enable structured reasoning and improved contextual coherence (Peng et al., 2024). GraphRAG (Edge et al., 2024) aggregates entities into subgraphs using community detection, while LightRAG (Guo et al., 2024) improves retrieval efficiency through a dual-level local-global strategy. However, both methods face a common limitation: they either include loosely related nodes that dilute logical connections or rely on keyword-based retrieval that fails to capture deeper semantic and reasoning structures.

2.3 Reasoning Path-Based RAG

Recent efforts aim to improve reasoning in RAG by incorporating multi-step query structures (Trivedi et al., 2023). HopRAG (Liu et al., 2025) extends retrieval through multi-hop traversal using LLM-generated pseudo-queries, and PathRAG (Chen et al., 2025) transforms retrieved paths into structured representations to highlight semantic links. While these methods emphasize reasoning-aware retrieval, they often lack explicit alignment between reasoning steps and retrieved content, resulting in disconnected or overly abstract context structures that can hinder faithful answer generation. These limitations highlight a gap between multi-step reasoning structures and the retrieval strategies used in prior RAG systems. PathCoRAG addresses this gap by explicitly aligning each reasoning step with semantically relevant nodes and their connecting paths, constructing a logic-preserving context that guides the LLM through a coherent, step-by-step chain of thought.

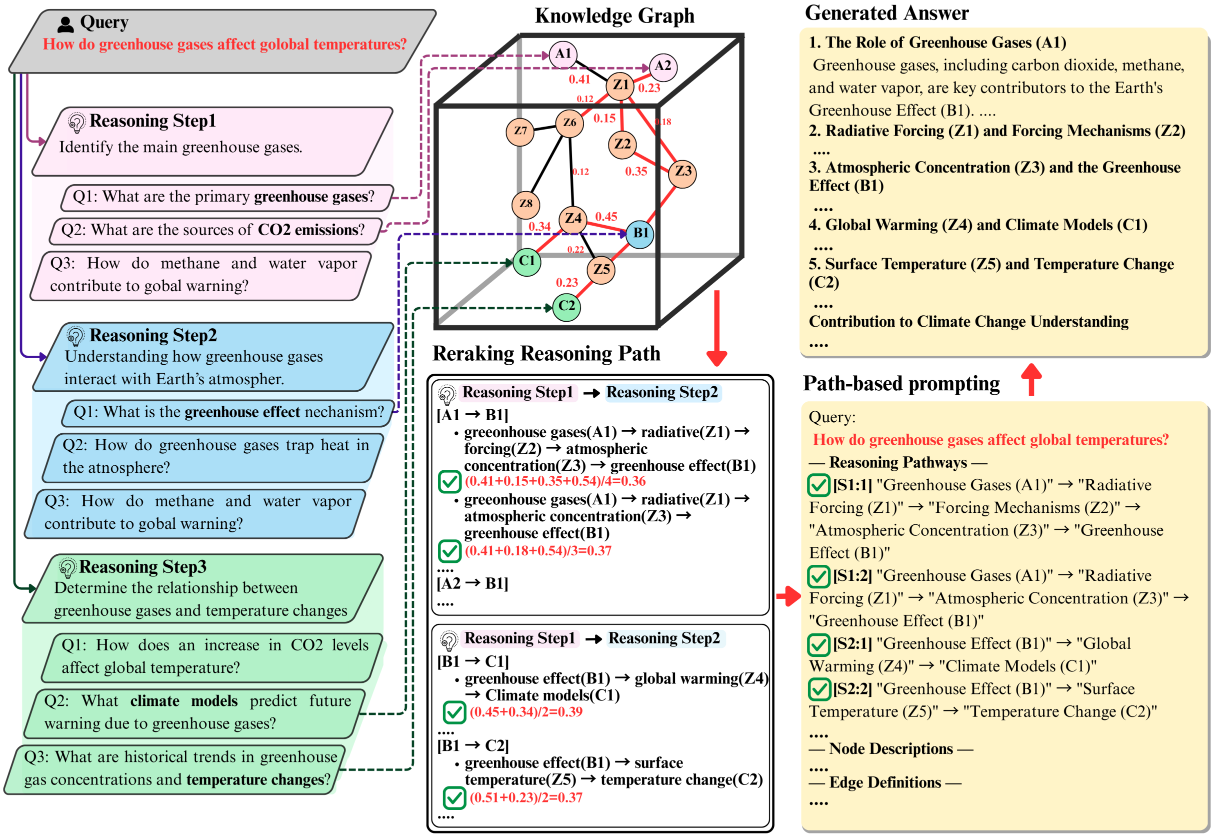

Figure 2: An illustration of the PathCoRAG methodology, including query decomposition into reasoning steps, node retrieval, reasoning path construction, and prompt assembly. This approach systematically expands complex queries into structured reasoning steps, extracts relevant nodes from the knowledge graph, identifies optimal reasoning paths using path scoring, and formats the final context for response generation.

3 Background

Graph-based RAG enhances LLM reasoning by modeling entity relationships in structured knowledge graphs, supporting complex multi-step interactions (Li et al., 2025). This section reviews the architecture and components of graph-based RAG, including graph construction, knowledge representation, and retrieval.

3.1 Graph-Based RAG Architecture

Graph-based RAG systems convert unstructured text into knowledge graphs, representing entities as nodes and relationships as edges (Chen et al., 2025). These structures are encoded into dense vectors to enable semantic search (Huang et al., 2025). Given a query, relevant nodes and paths are retrieved and scored to construct coherent contexts (Zhou et al., 2023).

3.2 Graph Construction

Graphs are constructed by extracting entities and their relationships from text (Edge et al., 2024). Each node $v \in V$ includes descriptive text t_v, and each edge $e \in E$ captures semantic connections. The graph is represented as $G = (V, E, T)$, where T contains textual descriptions (Li et al., 2024).

3.3 Knowledge Representation and Indexing

Nodes and edges are embedded using an embedding model and indexed in a vector database for fast semantic retrieval (Huang et al., 2025).

3.4 Graph-Based Retrieval

Relevant nodes and multi-hop paths are selected based on semantic similarity, forming coherent and compact contexts that reduce noise and improve response accuracy.

4 PathCoRAG

This section introduces PathCoRAG, a novel graph-based RAG framework designed to align retrieval with multi-step reasoning. Unlike prior systems that retrieve flat, chunk-level content, PathCoRAG performs step-wise reasoning-aware retrieval and constructs structured context based on explicit reasoning paths. As shown in Figure 2, PathCoRAG consists of four core components: (1) Chain-of-Thought-based Query Expansion, (2) Step-aligned Node Retrieval, (3) Semantic Path Search and Scoring, and (4) Path-structured Answer Generation.

4.1 Query Expansion

To support deep and interpretable reasoning, PathCoRAG decomposes the initial query into three explicit reasoning steps (RS_1, RS_2, RS_3), each reflecting a sub-goal in the overall inferential process. This structured breakdown encourages the model to focus on discrete aspects of the reasoning trajectory, reducing cognitive overload and ambiguity. For each step, the model generates three semantically diverse sub-queries using LLM prompting and few-shot demonstrations, resulting in a total of nine sub-queries per input query:

$$Q = \{RS_1, RS_2, RS_3\} \tag{1}$$

$$RS_i = \{q_{i1}, q_{i2}, q_{i3}\} \tag{2}$$

This CoT-style expansion enforces logical decomposition and semantic coverage, allowing later components to retrieve targeted evidence aligned with specific reasoning needs. Prompting examples are provided in Appendix A.

4.2 Node Retrieval

For each sub-query q_{ij}, we retrieve the top-N semantically relevant nodes from the knowledge graph using cosine similarity between embeddings:

$$\mathrm{sim}(q_{ij}, v) = \frac{\mathbf{q_{ij}} \cdot \mathbf{v}}{\|\mathbf{q_{ij}}\| \|\mathbf{v}\|} \tag{3}$$

To ensure step-aligned context, retrieved nodes are deduplicated across sub-queries and assigned to the reasoning step where they are most relevant:

$$\begin{gathered}\mathrm{step}(v) = \mathrm{argmax}_i(\mathrm{sim}(q_{ij}, v)) \\ \text{for } i \in \{1, 2, 3\}\end{gathered} \tag{4}$$

This step-aware allocation reduces noise and ensures that the retrieved context is both semantically focused and structurally aligned, which is essential for capturing complex logical dependencies across steps.

4.3 Reasoning Path Search

To connect the selected nodes across reasoning steps, PathCoRAG searches for top-k multi-hop reasoning paths (RS1→RS2, RS2→RS3) using Yen's algorithm (Yen, 1971). This path search explicitly encodes logical continuity between reasoning steps and filters out disconnected or semantically weak paths. Each candidate path is scored by its semantic alignment to the original query:

$$S(P) = \frac{1}{|P|} \sum_{e \in P} \mathrm{sim}(Q, t_e) \tag{5}$$

where t_e is the edge description. This scoring mechanism selects paths that are not only connected but also semantically meaningful in the context of the original question, enabling high-quality reasoning flow. Details are in Appendix B.

Datasets	Mix	CS	Agriculture	Legal
# of documents	61	10	12	94
# of tokens	619K	2.3M	2M	5M
# of nodes in KG	10K	20K	22K	20K
# of edges in KG	4.8K	13K	14K	24K

Table 1: Dataset Statistics for Mix, CS, Agriculture, and Legal Domains. KG denotes the indexed knowledge graph.

4.4 Answer Generation

Finally, PathCoRAG transforms the selected reasoning paths into a structured prompt, integrating node descriptions and edge semantics into a coherent chain of thought. This prompt is designed to reflect the underlying logic of the reasoning trajectory (RS1 $\rightarrow$ RS2 $\rightarrow$ RS3), guiding the LLM to generate answers that are not only factually grounded but also logically traceable and interpretable. The context formatting process is detailed in Appendix C.

Overall, these design choices address key limitations in prior work by bridging the gap between reasoning intent and retrieval behavior. PathCoRAG produces logically structured, semantically rich contexts that mitigate redundancy, prevent disjointed reasoning, and enable more faithful, multi-step answer generation.

5 Experiments

To evaluate the performance of PathCoRAG, we conducted extensive experiments to assess its effectiveness across various domains (Section 5.2), the impact of hyperparameter settings (Section 5.3), and the influence of reasoning structure on response quality (Section 5.4).

5.1 Experimental Settings

5.1.1 Datasets

We evaluate PathCoRAG using the UltraDomain benchmark (Qian et al., 2025), which consists of

	Mix		CS		Agriculture		Legal	
	NaiveRAG	**PathCoRAG**	NaiveRAG	**PathCoRAG**	NaiveRAG	**PathCoRAG**	NaiveRAG	**PathCoRAG**
Comprehensive	39.20%	**60.80%**	28.80%	**71.20%**	36.00%	**64.00%**	29.60%	**70.40%**
Diversity	24.00%	**76.00%**	23.20%	**76.80%**	27.20%	**72.80%**	11.20%	**88.80%**
Empowerment	36.00%	**64.00%**	22.40%	**77.60%**	29.60%	**70.40%**	24.00%	**76.00%**
Overall	36.80%	**63.20%**	21.60%	**78.40%**	31.20%	**68.80%**	25.60%	**74.40%**
	HyDE	**PathCoRAG**	HyDE	**PathCoRAG**	HyDE	**PathCoRAG**	HyDE	**PathCoRAG**
Comprehensive	37.60%	**62.40%**	39.20%	**60.80%**	43.20%	**56.80%**	35.20%	**64.80%**
Diversity	32.80%	**67.20%**	23.20%	**76.80%**	44.00%	**56.00%**	26.40%	**73.60%**
Empowerment	36.80%	**63.20%**	34.40%	**65.60%**	42.40%	**57.60%**	30.40%	**69.60%**
Overall	38.40%	**61.60%**	35.20%	**64.80%**	42.40%	**57.60%**	31.20%	**68.80%**
	GraphRAG	**PathCoRAG**	GraphRAG	**PathCoRAG**	GraphRAG	**PathCoRAG**	GraphRAG	**PathCoRAG**
Comprehensive	39.20%	**60.80%**	35.20%	**64.80%**	35.20%	**64.80%**	32.80%	**67.20%**
Diversity	45.60%	**54.40%**	34.40%	**65.60%**	28.00%	**72.00%**	40.00%	**60.00%**
Empowerment	42.40%	**57.60%**	36.80%	**63.20%**	32.00%	**68.00%**	31.20%	**68.80%**
Overall	40.80%	**59.20%**	37.60%	**62.40%**	33.60%	**66.40%**	32.80%	**67.20%**
	LightRAG	**PathCoRAG**	LightRAG	**PathCoRAG**	LightRAG	**PathCoRAG**	LightRAG	**PathCoRAG**
Comprehensive	35.20%	**64.80%**	37.60%	**62.40%**	38.40%	**61.60%**	31.20%	**68.80%**
Diversity	25.60%	**74.40%**	24.00%	**76.00%**	24.80%	**75.20%**	20.80%	**79.20%**
Empowerment	32.00%	**68.00%**	32.80%	**67.20%**	36.00%	**64.00%**	29.60%	**70.40%**
Overall	33.60%	**66.40%**	33.60%	**66.40%**	36.80%	**63.20%**	29.60%	**70.40%**
	PathRAG	**PathCoRAG**	PathRAG	**PathCoRAG**	PathRAG	**PathCoRAG**	PathRAG	**PathCoRAG**
Comprehensive	19.20%	**80.80%**	24.80%	**75.20%**	32.00%	**68.00%**	23.60%	**76.40%**
Diversity	14.40%	**85.60%**	9.60%	**90.40%**	9.60%	**90.40%**	17.60%	**82.40%**
Empowerment	16.80%	**83.20%**	20.80%	**79.20%**	26.40%	**73.60%**	22.80%	**77.20%**
Overall	16.00%	**84.00%**	21.60%	**78.40%**	27.20%	**72.80%**	22.80%	**77.20%**

Table 2: Main Results for PathCoRAG and Baseline Models

four diverse domains: Agriculture, Computer Science (CS), Legal, and Mixed. These datasets provide a broad and rigorous environment for assessing retrieval-augmented generation across domains. Table 1 presents the total token count, as well as the number of nodes and edges for each domain-specific graph. Further dataset statistics are available in Appendix D.

5.1.2 Metrics

We evaluated the quality of generated responses using a win-rate metric based on LLM pairwise comparisons (Zheng et al., 2023), with GPT-4o-mini as the evaluator. This metric assesses which model's output is preferred more often when compared directly. Following Guo et al. (2024), we evaluated responses across four dimensions: Comprehensiveness (coverage of essential query aspects), Diversity (range of perspectives), Empowerment (support for user understanding and decision-making), and Overall Quality (holistic evaluation of the other three). Further details on the evaluation criteria can be found in AppendixE.

5.1.3 Implementation Details

All LLM modules are implemented using GPT-4o-mini. For graph construction, we follow Guo et al. (2024) by chunking documents and extracting entities and relations. Through hyperparameter tuning, we set the number of top retrieved nodes to 50, number of k-shortest paths to 4, and the final selected reasoning paths to 15. For each query, we generate 3 reasoning steps, each with 3 sub-queries, forming the basis for step-wise node retrieval and reasoning path construction.

5.1.4 Baselines

For comparison, we evaluated PathCoRAG against several strong baselines, including NaiveRAG (Gao et al., 2023b), GraphRAG (Edge et al., 2024), LightRAG (Guo et al., 2024), HyDE (Gao et al., 2023a), and PathRAG (Chen et al., 2025). **NaiveRAG** adopts a basic RAG framework by retrieving text chunks based on vector similarity. **GraphRAG** constructs a knowledge graph and leverages community-based clustering for improved context selection. **LightRAG** integrates entity-level retrieval with lightweight community summarization for efficient response generation. **HyDE** generates hypothetical documents from the query to build a richer intermediate context before answer generation. **PathRAG** introduces path-centric retrieval by selecting top-k reasoning paths from a knowledge graph, aiming to improve multi-

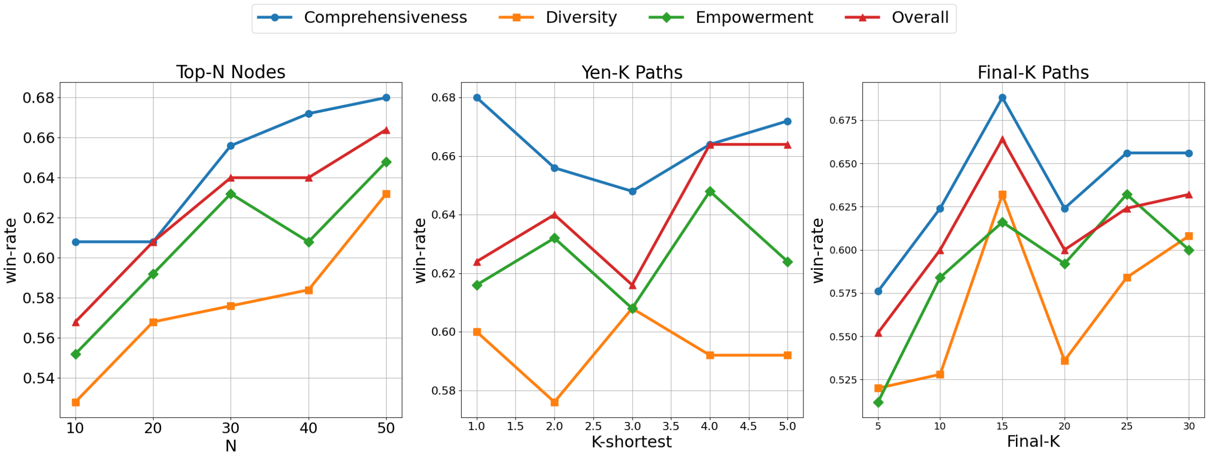

Figure 3: Effect of Top-N node selection, k-shortest path selection, and Final-K filtering on comprehensiveness, diversity, empowerment, and overall performance.

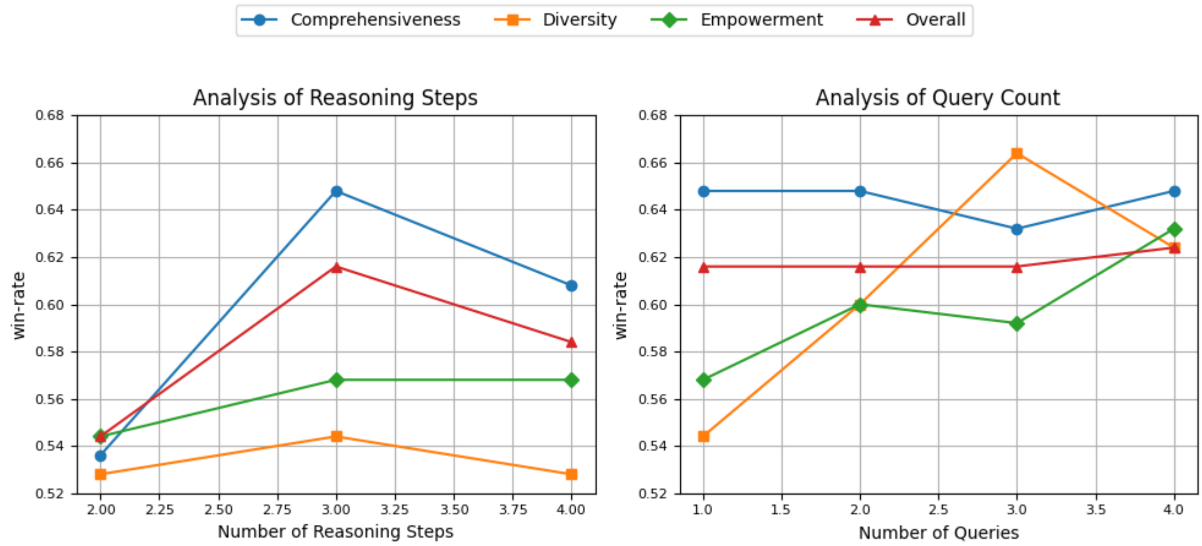

Figure 4: Effect of reasoning step depth (left) and number of query expansions per step (right) on PathCoRAG performance.

hop reasoning. Further details on these baselines are provided in Appendix F.

5.2 Main Results

Table 2 shows that PathCoRAG consistently outperforms all baseline models across four domains. The improvements are especially notable in Computer Science (78.4% overall win-rate) and Legal (77.2% overall win-rate), where deep multi-step reasoning is critical. These gains stem from step-wise query expansion, hierarchical node retrieval, and semantic path scoring, which allow PathCoRAG to isolate key concepts and maintain logical flow across reasoning steps. In contrast to flat or keyword-based retrieval, our method constructs more coherent and relevant context. A key challenge in multi-step frameworks is error propagation—where an inaccurate or irrelevant node retrieved in an early reasoning step can mislead subsequent path construction and context assembly, ultimately degrading answer quality. PathCoRAG mitigates this through redundant sub-queries and robust path scoring, ensuring stability even under noisy inputs. Additionally, PathCoRAG excels in Diversity and Empowerment metrics, reflecting its ability to retrieve nuanced and multi-faceted information. Overall, its integrated design effectively addresses core limitations of prior RAG systems—producing answers that are faithful, coherent, and interpretable.

5.3 Hyperparameter and Model Analysis

To investigate the contribution of key design choices in PathCoRAG, we conduct detailed ablation and sensitivity studies on the Mix domain, a heterogeneous dataset spanning various topics. Our analysis focuses on five components: top-N

	Mix		CS		Agriculture		Legal	
	w/o reasoning	reasoning path	w/o reasoning	reasoning path	w/o reasoning	reasoning path	w/o reasoning	reasoning path
comprehensive	44.80%	**55.20%**	48.00%	**52.00%**	47.20%	**52.80%**	47.20%	**52.80%**
Diversity	42.40%	**57.60%**	**54.40%**	45.60%	49.60%	**50.40%**	**52.00%**	48.00%
Empowerment	45.60%	**54.40%**	46.40%	**53.60%**	48.00%	**52.00%**	48.00%	**52.00%**
Overall	44.00%	**56.00%**	48.00%	**52.00%**	48.80%	**51.20%**	48.80%	**51.20%**

Table 3: Impact of Reasoning Path on Model Performance

node selection, k-shortest path selection, final path filtering, reasoning step depth, and the number of sub-queries per step.

5.3.1 Top-N Node Selection

We analyze the impact of varying the number of nodes retrieved per reasoning sub-query. As N increases from 10 to 50, we observe consistent improvements across all evaluation dimensions—Comprehensiveness, Diversity, Empowerment, and Overall—as illustrated in Figure 3 (left). This uniform gain is due to the expanded node pool providing a richer set of semantically relevant candidates, which enhances context without introducing significant redundancy. Unlike path selection, which may involve overlapping or semantically repetitive routes, increasing Top-N node selection uniformly enriches the retrieval base, leading to more stable and balanced improvements. Beyond $N = 50$, the gains plateau, suggesting diminishing returns. We adopt $N = 50$ as the optimal configuration balancing informativeness and precision.

5.3.2 k-Shortest Path Selection

We explore the impact of varying the number of reasoning paths (k) retrieved between node stages. As k increases, Comprehensiveness and Empowerment generally improve due to the inclusion of more reasoning routes. However, beyond $k = 4$, Diversity tends to decrease slightly, as additional paths often overlap or exhibit less semantic variation. This reflects a natural trade-off between precision and diversity: while more paths enrich logical connectivity, they may also introduce redundancy. As shown in Figure 3 (middle), $k = 4$ provides the best trade-off, capturing diverse yet semantically meaningful paths without overwhelming the context. Compared to node retrieval, where expansion consistently improves all metrics, path selection must be more carefully calibrated to balance informativeness with diversity.

5.3.3 Final Path Filtering (Final-K)

To filter and select the most informative reasoning paths for context construction, we experiment with varying the number of final paths (Final-K) included in the prompt. As shown in Figure 3 (right), performance across all metrics follows a bell-shaped curve.When Final-K is too small (e.g., 5), win-rates are low across all metrics due to insufficient context coverage. As Final-K increases to 15, all metrics—Comprehensiveness, Diversity, Empowerment, and Overall—improve significantly, reaching peak values. This suggests that moderate inclusion provides rich yet focused semantic cues for reasoning. However, further increasing Final-K beyond 15 leads to slight drops in some metrics, particularly Diversity and Empowerment. This indicates that excessive path inclusion introduces redundant or less relevant information, which can distract the model from the core reasoning flow. Therefore, we adopt Final-K=15 as the optimal configuration, balancing informativeness and focus.

5.3.4 Reasoning Step Depth

To explore the impact of logical decomposition depth, we varied the number of reasoning steps from 2 to 4. As illustrated in Figure 4 (left), moving from 2 to 3 steps results in significant gains across all evaluation metrics—particularly in Comprehensiveness and Overall performance. This suggests that a three-step decomposition provides a good balance between logical granularity and semantic coherence in the retrieved context. However, extending the chain to 4 steps leads to a slight decline in performance, likely due to over-fragmentation and increased noise. While Empowerment remains stable, other metrics show reduced effectiveness, indicating that deeper decomposition may complicate reasoning clarity.

5.3.5 Query Expansion per Reasoning Step

We investigate the impact of varying the number of expanded queries per reasoning step, ranging from 1 to 4. As shown in Figure 4 (right), performance steadily improves as the number increases, with three queries per step achieving the best overall balance across all four evaluation metrics (Comprehensiveness: 0.632, Diversity: 0.664, Empowerment: 0.592, Overall: 0.616). This configuration provides sufficient semantic coverage without overloading the retrieval module. While a fourth query yields a marginal improvement in Overall score, the gains are limited and come with increased risk of introducing noise and unnecessary computational overhead. These observations highlight that moderate expansion—specifically, three queries per step—offers a favorable trade-off between retrieval diversity, response quality, and efficiency.

5.4 Impact of Reasoning Path Information

To measure the value of path-aware prompting, we compare models with and without reasoning path structures included in the final prompt. As shown in Table 3, incorporating reasoning paths consistently improves performance across all four domains. Notably, in the Mix domain, we observe large gains in Comprehensiveness (+10.4%p), Diversity (+15.2%p), and Empowerment (+8.8%p). This indicates that modeling explicit reasoning paths helps the model better understand logical flow and retrieve more contextually relevant knowledge, leading to improved multi-hop generation.

5.5 Token Cost Analysis

We compare PathCoRAG with LightRAG, known for its efficiency in retrieval and token usage (Guo et al., 2024).

Metric	PathCoRAG	LightRAG
Keyword/Expansion Time	760.58 tokens / 5.43 s	410.58 tokens / 1.48 s
Path Search/Retrieval Time	7.06 s	0.83 s
In-Context Tokens	**8,830.52 tokens**	21,073.82 tokens

Table 4: Cost comparison: PathCoRAG vs. LightRAG.

As shown in Table 4, although PathCoRAG incurs 4–5 seconds more per query due to path search and expansion, it significantly reduces in-context tokens by over 50%. This compact context construction highlights PathCoRAG's practicality in resource-constrained scenarios while maintaining superior performance.

6 Further Discussion

6.1 Evaluation on Objective Benchmarks

We conducted additional evaluations using established QA benchmarks that provide ground-truth answers. These datasets allow for more objective comparisons based on standard evaluation metrics and help verify the effectiveness of PathCoRAG in diverse reasoning settings beyond our original test environment. Appendix G reports experimental settings and detailed results on the NovelQA(Wang et al., 2024), InfiniteQA(Zhang et al., 2024) and InfiniteChoice(Zhang et al., 2024) benchmark datasets.

PathCoRAG demonstrates strong performance across diverse benchmarks. On NovelQA, it excels in structured multi-hop reasoning with long-form answers. In InfiniteChoice, it performs competitively on contextual comparison tasks. Even in InfiniteQA, which favors concise factoid responses less suited to its design, PathCoRAG maintains solid performance. These results highlight its robustness and versatility in handling a range of reasoning challenges.

6.2 Evaluation on Multihop QA

To strengthen the empirical validation of PathCoRAG's reasoning capabilities, we include additional experiments on the widely-used HotpotQA(Yang et al., 2018) dataset. As HotpotQA is a standard benchmark for multi-hop question answering with diverse reasoning challenges, this evaluation allows us to assess the generalizability of our model beyond domain-specific settings and directly compare it with recent competitive baselines such as HippoRAG and HopRAG. Experimental settings and detailed results can be found in Appendix H.

PathCoRAG performs competitively on HotpotQA, despite not being tailored for short, extractive answers. Its strength in global reasoning and structured context construction proves effective for multi-hop inference, outperforming PathRAG and rivaling HippoRAG in real-world QA settings.

7 Conclusion

We presented PathCoRAG, a retrieval-augmented generation framework that explicitly aligns multi-step reasoning with structured retrieval through path-aware query expansion and graph-based context construction. By decomposing complex queries into step-wise reasoning units and retrieving semantically connected nodes and paths, PathCoRAG generates coherent, logically grounded answers while significantly reducing irrelevant context. Our experiments demonstrate consistent gains across diverse domains, validating the effectiveness of reasoning-aligned retrieval and structured context over traditional flat retrieval approaches.

Limitation

PathCoRAG adopts a fixed 3-step, 3-query-per-step expansion strategy to balance performance and efficiency. However, this uniform structure may not adapt well to queries of varying complexity, potentially leading to under- or over-decomposition. Future work will explore dynamic reasoning step selection and adaptive query expansion based on query semantics and reasoning depth.

References

Boyu Chen, Zirui Guo, Zidan Yang, Yuluo Chen, Junze Chen, Zhenghao Liu, Chuan Shi, and Cheng Yang. 2025. Pathrag: Pruning graph-based retrieval augmented generation with relational paths. *arXiv preprint arXiv:2502.14902*.

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*.

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

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

Yun Guo, Wei Wang, and Zhen Li. 2024. Lightrag: Efficient graph-based context retrieval for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yiqian Huang, Shiqi Zhang, and Xiaokui Xiao. 2025. KET-RAG: A Cost-Efficient Multi-Granular Indexing Framework for Graph-RAG. *arXiv preprint arXiv:2502.09304*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*.

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.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Angela Fan, Vishrav Chaudhary, Emily Payne, David Wendt, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Mufei Li, Siqi Miao, and Pan Li. 2025. Simple is Effective: The Roles of Graphs and Large Language Models in Knowledge-Graph-Based Retrieval-Augmented Generation. In *Proceedings of the 2025 International Conference on Learning Representations (ICLR)*.

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*.

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

Jian Liu, Xiaodong Liu, and Ming Zhao. 2022. Hyde: Hybrid dense and sparse retrieval for scalable qa systems. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921*.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. 2025. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. In *Proceedings of the ACM on Web Conference 2025*, pages 2366–2377.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. In *Foundations and Trends in Information Retrieval*, volume 3, pages 333–389. Now Publishers Inc.

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. 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, Toronto, Canada.

Cunxiang Wang, Ruoxi Ning, Boqi Pan, Tonghui Wu, Qipeng Guo, Cheng Deng, Guangsheng Bao, Xiangkun Hu, Zheng Zhang, Qian Wang, and 1 others. 2024. Novelqa: Benchmarking question answering on documents exceeding 200k tokens. *arXiv preprint arXiv:2403.12766*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

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.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Jin Y. Yen. 1971. Finding the k shortest loopless paths in a network. *Management Science*, 17(11):712–716.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and 1 others. 2024. bench: Extending long context evaluation beyond 100k tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15262–15277.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Wenxuan Zhou, Hongyu Wang, and Zhou Yu. 2023. Reducing hallucinations in retrieval-augmented generation with chain-of-thought prompting. *arXiv preprint arXiv:2305.13534*.

A Query Expansion Prompt

PathCoRAG decomposes complex queries into structured reasoning steps to enable more precise retrieval. Each query is divided into three reasoning steps, with each step generating multiple sub-queries to capture information from diverse perspectives. This hierarchical expansion separates intermediate concepts and strengthens semantic connections, allowing for more refined and contextually accurate retrieval. Please refer to Figure 5 for a detailed illustration.

B Yen's Algorithm for PathCoRAG

Yen's algorithm is an efficient method for finding the k-shortest paths from a single source node s to a single destination node t. It operates by first identifying the initial shortest path P_1 and then iteratively generating alternative paths $P_2, P_3, \ldots, P_k$ until the desired number of k paths are obtained.

B.1 Initial Shortest Path

The first shortest path, P_1, is typically computed using algorithms like Dijkstra or A*, defined as:

$$P_1 = \text{argmin}_P \left(\sum_{(u,v) \in P} w(u, v) \right) \tag{6}$$

where $w(u, v)$ represents the weight of each edge in the path. This initial path serves as the baseline for generating subsequent alternative paths.

B.2 Generating Alternative Paths

For each subsequent path P_i (where $i > 1$), Yen's algorithm selects a spur node from the previously identified shortest path and constructs a new path using this spur. This is defined as:

$$P_i = R_s + \text{spur}(n_s) \tag{7}$$

where R_s is the sub-path leading up to the spur node, and spur(n_s) is the remaining path from the spur node to the destination. To ensure path diversity, the algorithm partially blocks or removes overlapping segments from previously selected paths, allowing for the creation of unique alternative paths.

B.3 Path Selection

This process is repeated until a total of k shortest paths have been identified. Yen's algorithm is particularly well-suited for multi-step reasoning frameworks like PathCoRAG, as it efficiently explores a wide range of plausible reasoning paths while preserving critical relationships between nodes.

C Answer Generation Prompt

The answer generation prompt in PathCoRAG is designed to efficiently structure node and edge information associated with each reasoning path, minimizing redundancy and enhancing coherence. As illustrated in Figure 6, this approach integrates individual paths into a single, interconnected network, eliminating unnecessary node and edge repetitions. This unified representation allows the LLM to interpret complex, multi-step relationships more effectively, resulting in contextually consistent and accurate responses.

D UltraDomain Dataset Description

The UltraDomain benchmark is a comprehensive dataset constructed from 428 university textbooks, covering specialized domain documents across science, technology, and professional fields. It provides a diverse range of topics suitable for evaluating graph-based RAG systems. The UltraDomain dataset is licensed under the Apache License 2.0.

Mix

The Mix dataset contains diverse documents from various disciplines, including literature, history, philosophy, and social sciences. It is designed to test the generalization ability of models across a wide range of knowledge domains. This dataset presents a challenging environment for multi-hop reasoning, with complex narratives, abstract concepts, and interdisciplinary references. Key topics include literature, philosophy, history, sociology, political science, and anthropology.

Computer Science (CS)

The Computer Science dataset includes materials on algorithms, data structures, software engineering, machine learning, and artificial intelligence. It captures both theoretical concepts and practical applications, reflecting the rapid advancements in computer science. It also covers cutting-edge topics like quantum computing, distributed systems, and big data analytics. Key topics include algorithms, data structures, machine learning, artificial neural networks, distributed computing, software engineering, big data, and artificial intelligence.

Agriculture

The Agriculture dataset contains documents related to agricultural practices, crop management, livestock care, and agricultural technology. It covers technical topics such as soil management, irrigation techniques, pest control, sustainable farming, and agricultural economics, reflecting the complex, interconnected nature of agricultural knowledge. This makes it well-suited for multi-hop reasoning and graph-based retrieval tasks. Key topics include crop science, pest management, soil conservation, irrigation systems, agricultural economics, sustainable agriculture, and precision farming.

Legal

The Legal dataset consists of documents related to corporate law, intellectual property, contract law, regulatory compliance, and legal ethics. This domain is characterized by precise terminology and structured argumentation, making it suitable for evaluating search systems that need to capture subtle legal nuances. Key topics include corporate law, intellectual property, contracts, regulations, ethics, legal reasoning, and dispute resolution.

E Evaluation Metrics

E.1 Win-Rate Evaluation Framework

This section describes the evaluation metrics used to assess model performance. In this study, we adopt a win-rate-based evaluation framework, similar to those used in recent graph-based RAG studies like LightRAG and PathRAG. win-rate-based evaluation presents the win-rate results showing how often PathCoRAG's responses are preferred over those of baseline models across various domains. These scores are based on pairwise comparisons using an LLM evaluator (GPT-4o-mini), where each entry reflects the percentage of times PathCoRAG outperformed a specific baseline. Since win-rates are computed separately for each baseline, the same dataset and metric may yield different values depending on the comparison target (e.g., NaiveRAG vs. LightRAG). This explains the variation in PathCoRAG's reported scores within a single domain. We emphasize that these are not absolute performance scores, but relative win-rates from independent evaluation rounds.

E.2 Metric Definitions

Comprehensiveness

Measures how completely the response addresses all aspects of the input query, reflecting the model's ability to provide detailed and thorough answers.

Diversity

Evaluates the extent to which the response covers a wide range of perspectives and information, indicating the breadth of the response.

Empowerment

Assesses the degree to which the response helps the reader gain a deeper understanding of the topic and make informed decisions based on the provided information.

Overall Quality

Combines comprehensiveness, diversity, and empowerment (Edge et al., 2024) to provide an overall assessment of response quality.

E.3 Win-Rate Calculation

The evaluation follows a win-rate approach, where each response comparison is recorded as a win for the more preferred response. The final win-rate is calculated as the proportion of wins over the total number of comparisons. To reduce bias, the order in which responses are presented is randomized, and the final results are reported as the average of multiple experimental runs. For a detailed description of the prompt structures used in this evaluation, please refer to Figure 7 in this paper.

F Baseline Descriptions

NaiveRAG

NaiveRAG (Gao et al., 2023b) is the most basic form of RAG, where the text corpus is divided into fixed-size chunks and stored in a vector database. The chunks with the highest similarity to the input query are retrieved. While this approach is simple and efficient, it lacks the structured context provided by graph-based systems, potentially missing deeper relational information.

HyDE

HyDE (Liu et al., 2022) generates synthetic documents using an LLM based on the input query, and then retrieves relevant text chunks from the external database using these generated documents. While this method can capture more contextual information, it may introduce noise if the generated document does not accurately reflect the original query intent. For the HyDE baseline, the exact license details are not available. While the specific license for HyDE is not provided, its source code can be accessed at URL:`https://github.com/texttron/hyde`.

GraphRAG

GraphRAG (Edge et al., 2024) constructs an index graph by extracting entities as nodes and their relationships as edges from the text corpus. The graph structure enables multi-step reasoning by grouping texts into several interconnected communities, capturing complex dependencies between entities. However, this approach can be computationally expensive due to the graph construction and path search processes. GraphRAG is licensed under the MIT-License.

LightRAG

LightRAG (Guo et al., 2024) local and global keyword extraction for more efficient retrieval. It focuses on the immediate neighbors of relevant nodes, reducing computational costs while maintaining a balance between precision and recall. This dual-level search framework provides a more efficient yet context-rich retrieval mechanism. LightRAG is licensed under the MIT-License.

PathRAG

PathRAG (Chen et al., 2025) enhances graph-based RAG by explicitly constructing reasoning paths from source documents. It segments documents into structured graphs and performs step-by-step path traversal to simulate multi-hop reasoning. The framework focuses on aligning query intent with coherent paths of entities and relations, improving factual consistency in the generated responses. However, PathRAG heavily relies on the quality of the pre-constructed graph and reasoning path search, which may lead to incomplete or biased paths if the initial structure is sparse or fragmented. For the PathRAG baseline, the exact license details are not available. While the specific license for PathRAG is not provided, its source code can be accessed at URL:`https://https://github.com/BUPT-GAMMA/PathRAG`.

G Additional Evaluation on Objective Benchmarks

Datasets

We evaluated PathCoRAG and baselines on the following datasets:

- **NovelQA**: A QA benchmark designed for evaluating long-context understanding, with multi-hop reasoning and golden answers. For this study, we used a sampled subset of 22 queries with golden answers available.

- **InfiniteQA**: A benchmark that includes factoid-style open-ended questions. We sampled 125 queries for evaluation.

- **InfiniteChoice**: A multiple-choice benchmark containing diverse, multi-step reasoning questions. We sampled 100 queries and provided answer options for each.

Metrics

InfiniteChoice and InfiniteQA were evaluated as multiple-choice tasks, where the model selects the most likely answer from provided candidates.

- **Accuracy**: Used for NovelQA and InfiniteChoice, defined as the percentage of queries for which the selected or generated answer exactly matches the correct answer.

- **Rouge-L F1**: Applied to InfiniteQA. Measures the longest common subsequence between the generated and reference answers, focusing on factual overlap.

Method	NovelQA (Accuracy)	InfiniteQA (Rouge-L F1)	InfiniteChoice (Accuracy)
NaiveRAG	52.38	19.83	26.00
HyDE	59.09	**27.17**	40.00
GraphRAG	<u>63.63</u>	11.03	16.00
LightRAG	47.62	24.78	19.00
PathRAG	50.00	20.10	**47.00**
PathCoRAG	**66.67**	<u>25.07</u>	<u>43.00</u>

Table 5: Results on NovelQA, InfiniteQA, and InfiniteChoice

Results and Analysis

As shown in Table 5, PathCoRAG consistently demonstrates strong performance across diverse benchmarks aimed at evaluating multi-step reasoning capabilities. On NovelQA, which features long-form ground-truth answers and necessitates structured multi-hop inference, PathCoRAG achieves the highest accuracy (66.67), surpassing all other baselines. In the case of InfiniteChoice, a multiple-choice dataset requiring contextual judgment and comparison, PathCoRAG records 43.00 accuracy—closely trailing the best-performing method (PathRAG, 47.00) and showcasing highly competitive performance. On InfiniteQA, which emphasizes concise factoid-style answers and is less aligned with our model's design focus on step-wise reasoning, PathCoRAG achieves a Rouge-L F1 of 25.07. While this is slightly lower than HyDE's 27.17, it remains within a comparable range and highlights that PathCoRAG can still deliver high-quality outputs even in scenarios where its structured reasoning mechanism is underutilized. These results collectively validate the robustness and generalizability of PathCoRAG, particularly when applied to tasks demanding deep reasoning and structured information synthesis.

H Additional Comparison with PathRAG and HopRAG

Dataset

- **HotpotQA**: HotpotQA is a challenging question answering dataset designed to assess multi-hop reasoning across multiple documents. It contains Wikipedia-based questions that require combining evidence from different passages to answer a given question. For our evaluation, we randomly sampled 1,000 questions from the HotpotQA development set to conduct a comparative analysis across models.

Metrics

We report results using two standard evaluation metrics for HotpotQA:

- **Exact Match (EM)**: Measures whether the generated answer exactly matches the ground-truth answer.

- **F1 Score**: Computes the overlap between the predicted and ground-truth answers at the token level, considering both precision and recall.

Baseline

- **HopRAG** (Liu et al., 2025): is a retrieval-augmented generation framework that explicitly models the retrieval of reasoning steps using structured document graphs, optimizing path traversal and relevance scoring.

- **HippoRAG** (Jimenez Gutierrez et al., 2024): proposes a hybrid hierarchical retrieval architecture that combines short-term document context with long-term memory to improve long-context comprehension and answer recall.

Method	HotpotQA	
	EM	**F1**
PathRAG	41.40	55.10
HippoRAG	59.90	74.20
HopRAG	**61.30**	**78.30**
PathCoRAG	60.10	74.91

Table 6: Performance comparison on HotpotQA

Results and Analysis

As shown in Table 6, PathCoRAG demonstrates strong performance on the HotpotQA dataset, achieving results comparable to HopRAG despite not being explicitly optimized for short, extractive answers. While HopRAG slightly outperforms PathCoRAG in EM and F1 scores, PathCoRAG is designed for global reasoning and structured context construction, which aligns with tasks requiring multi-hop inference across distributed evidence. Additionally, PathCoRAG outperforms PathRAG and is competitive with HippoRAG, further validating its effectiveness in multi-step reasoning tasks under real-world QA settings.

I Case Study

Query	"How do the indemnification clauses vary among different contracts, and what risks do they present?"
GraphRAG	**Variation of Indemnification Clauses Among Contracts** Indemnification clauses are tailored to the specifics of each contract... Limited Partners may indemnify the General Partner... Borrowers may indemnify Lenders... Terms can vary widely, covering direct damages or broader liabilities... Some contracts provide comprehensive coverage, others impose strict limitations... **Risks Associated with Indemnification Clauses** The risks can be substantial, particularly if overly broad or ambiguous... Clauses may expose parties to unexpected liabilities... Disputes over what constitutes an indemnifiable event... Financial health of the indemnifying party is critical...
PathCoRAG	**Variations in Indemnification Clauses** Indemnification clauses can vary significantly... Scope of Coverage may include breach of contract, negligence, or third-party claims... For example, Section 10.04(b) covers Indemnitees, while Clause (C) provides exceptions... Types of Indemnified Parties can differ... including directors, officers, agents, or specific entities... Limitations and Exclusions are often included... such as Section 16.01(K)(II), which restricts coverage for certain breaches... Indemnified Amounts can vary... reflecting different compensation obligations... **Risks Presented by Indemnification Clauses** Indemnification clauses can also introduce risks... Unforeseen Liabilities - Broad definitions may expose a party to unexpected obligations... Legal Complexity - Disputes may arise over valid claims or covered losses... Financial Burden - Substantial costs for legal defenses and settlements... Dependence on Contractual Clarity - Vague terms can lead to enforcement issues...
LLM Decision	**Comprehensiveness** PathCoRAG (Answer 1) provides a comprehensive breakdown of various aspects of indemnification clauses... It covers scope of coverage, types of indemnified parties, limitations and exclusions, and methods for determining indemnified amounts... **Diversity** GraphRAG (Answer 2) highlights a more diverse perspective by addressing different contexts in which indemnification clauses might vary, such as partnerships and financial agreements... **Empowerment** PathCoRAG (Answer 1) empowers the reader by clearly outlining the variations and risks associated with indemnification clauses... It presents specific examples and scenarios that help the reader understand how different terms can affect the involved parties, supporting more informed decision-making... **Overall Winner** PathCoRAG (Answer 1) excels in both Comprehensiveness and Empowerment... It offers a richer and more informative response, effectively capturing the complexities and critical aspects of indemnification clauses...

Table 7: Comparison of GraphRAG and PathCoRAG responses to the indemnification clause query.

---Role---

You are a helpful assistant tasked with **expanding a given query using a Chain of Thought reasoning approach**.
Your goal is to break down the query into exactly three logical reasoning steps and generate exactly three extended queries at each step to improve document retrieval.

---Goal---

Given the query, **define a step-by-step reasoning process (with 3 steps) to reach an answer.**
For **each of the 3 steps**, **generate 3 extended queries that help retrieve relevant information.**

---Instructions---

- Analyze the query and break it down into **3 logical reasoning steps**.
- For **each step, generate 3 extended queries** to improve document retrieval.
- Output the reasoning process and queries in JSON format.
- The JSON should have the following structure:
  - reasoning_steps: A list of 3 reasoning steps required to answer the query.
  - extended_queries: A dictionary where each key is one of the reasoning steps and its value is a list of exactly 3 extended queries.
- All output must be in plain text, not unicode characters.
- Use the same language as the input `Query`.

Query Expansion Instruction Prompt

---Example---

Query: What are the most useful evaluation metrics for a movie recommendation system?
Output:

reasoning_steps:
  Identify types of recommendation systems., **Identify evaluation metrics for recommendation systems.**, **Compare evaluation metrics based on their characteristics.**

extended_queries:
  Identify types of recommendation systems.:
    What are the main types of recommendation systems?,
    What are the differences between collaborative filtering and content-based filtering?,
    How does a hybrid recommendation system work?
  Identify evaluation metrics for recommendation systems.:
    What are the key evaluation metrics for recommendation systems?,
    ...
  Compare evaluation metrics based on their characteristics.:
    Which is more important for recommendation systems: Precision or Recall?,
    ...

Query Expansion Example Prompt

Figure 5: Query expansion process in PathCoRAG, illustrating the hierarchical decomposition of complex queries into structured reasoning steps and sub-queries for more precise and contextually accurate retrieval.

---Role---

You are a helpful assistant responding to the user query based on the structured knowledge and reasoning information provided below.

---Goal---

Generate a concise and accurate response to the user's query by leveraging the reasoning paths and related knowledge presented in the context.
Do not use external knowledge.
Focus only on the information provided in the context.

---Knowledge Base---

{context_data}

---Response Rules---

- Target format and length: {response_type}
- Use markdown formatting with appropriate section headings if applicable.
- Respond in the same language as the user's question.
- Ensure the response maintains continuity with the conversation history.
- Do not include any information that is not present in the Knowledge Base.
- Prioritize the structure and reasoning captured in the provided context.

---Knowledge Base---

{query}

Response Prompt

---Reasoning Paths---
[S1:1] "LADY-WITCH" → "VISION"
[S1:2] "THE BOAT" → "THE IMAGE"
[S1:3] "THE WITCH OF ATLAS" → "MARY" → "SONG OF SOLOMON"
...

[S2:1] "SONG OF SOLOMON" → "MARY" → "THE WITCH OF ATLAS" → "PERCY BYSSHE SHELLEY"
[S2:2] "GARISH SUMMER DAYS" → "THE WITCH OF ATLAS" → "MARY"
[S2:3] "GARISH SUMMER DAYS" → "THE WITCH OF ATLAS" → "PERCY BYSSHE SHELLEY" → "MARY"
...

---Entities---
"id","entity","type","description"
0, LADY-WITCH, PERSON, The lady-witch is the central figure in the poem, embodying beauty and mystical power, and enchanting all creatures around her.
1, VISION, CONCEPT, Vision is illustrated as an ethereal presence or inspiration that engages with the lady-witch, symbolizing creativity and artistic expression.
2, THE BOAT, CATEGORY,The Boat serves as a symbolic vessel in the text, representing a transformative journey and the interplay of love and creativity, evolving from car to the lightest boat.
...

---Relationships---
"id","entity","type","description"
0, LADY-WITCH, PERSON, The lady-witch is the central figure in the poem, embodying beauty and mystical power, and enchanting all creatures around her.
1, VISION, CONCEPT, Vision is illustrated as an ethereal presence or inspiration that engages with the lady-witch, symbolizing creativity and artistic expression.
...

Response Context Data Prompt

Figure 6: Structured representation of nodes and edges for efficient answer generation, reducing redundancy and enhancing coherence by integrating individual paths into a unified network.

---Role---

You are an expert tasked with evaluating two answers to the same question based on three criteria: **Comprehensiveness**, **Diversity**, and **Empowerment**.

---Goal---

You will evaluate two answers to the same question based on three criteria: Comprehensiveness, Diversity, and Empowerment.

- **Comprehensiveness**: How much detail does the answer provide to cover all aspects and details of the question?
- **Diversity**: How varied and rich is the answer in providing different perspectives and insights on the question?
- **Empowerment**: How well does the answer help the reader understand and make informed judgments about the topic?

For each criterion, choose the better answer (either Answer 1 or Answer 2) and explain why. Then**, select an overall winner based on these three categories.**

Evaluation Instruction Prompt

Here is the question:{query}

Here are the two answers:
Answer 1: {answer1}
Answer 2:{answer2}

Evaluate both answers using the three criteria listed above and provide detailed explanations for each criterion.

Output your evaluation in the following JSON format:
Comprehensiveness: {Winner: [Answer 1 or Answer 2], Explanation: [Provide explanation here]},
Diversity: {Winner: [Answer 1 or Answer 2], Explanation: [Provide explanation here]},
Empowerment: {Winner: [Answer 1 or Answer 2], Explanation: [Provide explanation here]},
Overall Winner: {Winner: [Answer 1 or Answer 2], Explanation: [Summarize why this answer is the overall winner based on the three criteria]}

Evaluation Example Prompt

Figure 7: Evaluation prompt structure for assessing Comprehensiveness, Diversity, Empowerment, and Overall Quality