GRAPHRAG-BENCH: CHALLENGING DOMAIN-SPECIFIC REASONING FOR EVALUATING GRAPH RETRIEVAL-AUGMENTED GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph Retrieval-Augmented Generation (GraphRAG) has garnered increasing recognition for its potential to enhance large language models (LLMs) by structurally organizing domain-specific corpora and facilitating complex reasoning. However, current evaluations of GraphRAG models predominantly rely on traditional question-answering datasets. Their limited scope in questions and evaluation metrics fails to comprehensively assess the reasoning capacity improvements enabled by GraphRAG models. To address this gap, we introduce GraphRAG-Bench, a large-scale, domain-specific benchmark designed to rigorously evaluate GraphRAG models. Our benchmark offers three key superiorities: (i) Challenging question design. Featuring college-level, domain-specific questions that demand multi-hop reasoning, the benchmark ensures that simple content retrieval is insufficient for problem-solving. For example, some questions require mathematical reasoning or programming. (ii) Diverse task coverage. The dataset includes a broad spectrum of reasoning tasks, multiple-choice, true/false, multi-select, open-ended, and fill-in-the-blank. It spans 16 disciplines in twenty core textbooks. (iii) Holistic evaluation framework. GraphRAG-Bench provides comprehensive assessment across the GraphRAG pipeline, including graph construction, knowledge retrieval, and answer generation. Beyond final-answer correctness, it evaluates the logical coherence of the reasoning process. By applying nine contemporary GraphRAG methods to GraphRAG-Bench, we demonstrate its utility in quantifying how graph-based structuring improves model reasoning capabilities. Our analysis reveals critical insights about graph architectures, retrieval efficacy, and reasoning capabilities, offering actionable guidance for the research community.

1 INTRODUCTION

Retrieval-Augmented Generation (RAG) Lewis et al. (2020); Gao et al. (2024) has emerged as a key solution to ground large language models (LLMs) in external knowledge to mitigate both the hallucination problem and the lack of domain knowledge. By retrieving relevant text passages from corpora, RAG injects factual knowledge for a more reliable generation from LLMs. However, conventional RAG systems remain unsatisfactory when dealing with complex reasoning scenarios. The flat retrieval in RAG directly returns fragmentized chunks based on similarity matching, which limits their ability to model complex relationships between concepts to answer the questions requiring multi-hop reasoning Zhang et al. (2025); Dong et al. (2023), i.e., 'What was the impact of [event] the 2008 Lehman Brothers bankruptcy on [person] Elon Musk's Tesla?' or global comprehension, i.e., 'What is the main idea of the [event] Trade Policy Change?'.

To address these limitations, Graph Retrieval-Augmented Generation (GraphRAG) has been extensively studied to capture the structured knowledge among concepts in the form of graphs Edge et al. (2025); Peng et al. (2024); Zhou et al. (2025), where nodes represent concepts and edges are for the relations among them. Recent advances in GraphRAG can be categorized into three main directions. First, hierarchical graph construction methods like RAPTOR Sarthi et al. (2024) and Microsoft's GraphRAG Edge et al. (2025) organize knowledge through tree structures and community detection. Second, neural graph retrieval approaches, including GFM-RAG Luo et al. (2025) and G-Retriever He et al. (2024) employ graph neural encoders with specialized objectives for multi-

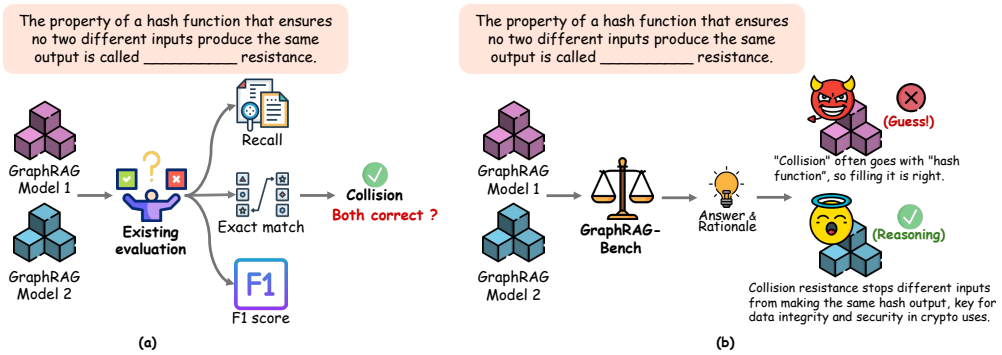

Figure 1: Comparison between existing evaluations (a) and our proposed GraphRAG-Bench (b). GraphRAG-Bench not only assesses the accuracy of generation but also evaluates the rationality of reasoning based on the challenging domain-specific questions.

hop reasoning. Third, dynamic knowledge integration systems such as DALK Li et al. (2024) and ToG Sun et al. (2024) develop adaptive graph construction and traversal mechanisms that are tightly coupled with LLMs. By structuring knowledge as graphs, GraphRAG enables LLMs to both traverse and reason over explicit relational paths, but also supports deeper reasoning by inferring implicit relations based on the graph structure Dong et al. (2025).

However, despite the promise, existing benchmarks for GraphRAG methods fail to reflect the performance of reasoning on graphs. They predominantly leverage the traditional QA dataset, e.g., HotpotQA Yang et al. (2018), 2WikiMultiHopQA Ho et al. (2020) and MuSiQue Trivedi et al. (2022), which only feature explicit factoid questions with limited complexity and short answers, e.g., 'Who is the grandchild of Dambar Shah?'. These datasets suffer from three critical limitations: (i) There are only commonsense questions that could be probably covered in the training corpus of LLMs. (ii) They typically require only single-hop or shallow multi-hop reasoning based on explicit connections, which inadequately probes the unique advantages of graph-structured knowledge. (iii) Narrow Answer Formats. Most answers are short (names, dates) or multiple-choice, which could hardly reflect the reasoning ability over graphs. To this end, we would like to ask a research question:

"Does graph augmentation truly enhance reasoning capabilities beyond simple retrieval?"

In this paper, we propose GraphRAG-Bench, the first challenging domain-specific benchmark particularly designed for GraphRAG. (i) Our dataset contains 1,018 college-level question spans 16 disciplines, e.g., computer vision, networks, human-computer interaction, etc, featuring the ability of conceptual understanding, e.g., "Given [theorem] A and B, prove [conclusion] C", complex algorithmic programming, e.g., coding with interlinked function calls) and mathematical computation, e.g., "Given [Input], [Conv1], [MaxPool], [FC], calculate the output volume dimensions." (ii) GraphRAG-Bench contains five types of diverse questions to thoroughly evaluate different aspects of reasoning, including multiple-choice (MC), multi-select (MS), true-or-false (TF), fill-in-blank (FB) and open-ended (OE). (iii) We offer a comprehensive multi-dimensional evaluation on each component of GraphRAG, including graph construction, knowledge retrieval, answer generation and rationale generation. We aim to provide unprecedented insights into how graph-structured knowledge enhances LLMs' reasoning capabilities compared to traditional RAG approaches. The major contributions are summarized hereunder:

- We propose the first challenging domain-specific benchmark, particularly concentrating on GraphRAG. It contains 1018 questions in 5 question types spanning 16 topics and a corpus of 7 million words from 20 computer science textbooks.
- A comprehensive evaluation protocol is designed to stress-test GraphRAG methods on graph construction, retrieval, and multi-hop answer generation and rationale generation.
- Extensive experiments have been conducted with nine state-of-the-art GraphRAG models. We make insightful observations and provide the insights that: 1) GraphRAG substantially enhances the reasoning capabilities of LLMs, and - to the best of our knowledge - we are the first to quantify this improvement using concrete evaluation metrics. 2) GraphRAG's impact varies by question types: it yields significant gains on some types but offers limited benefit for others.

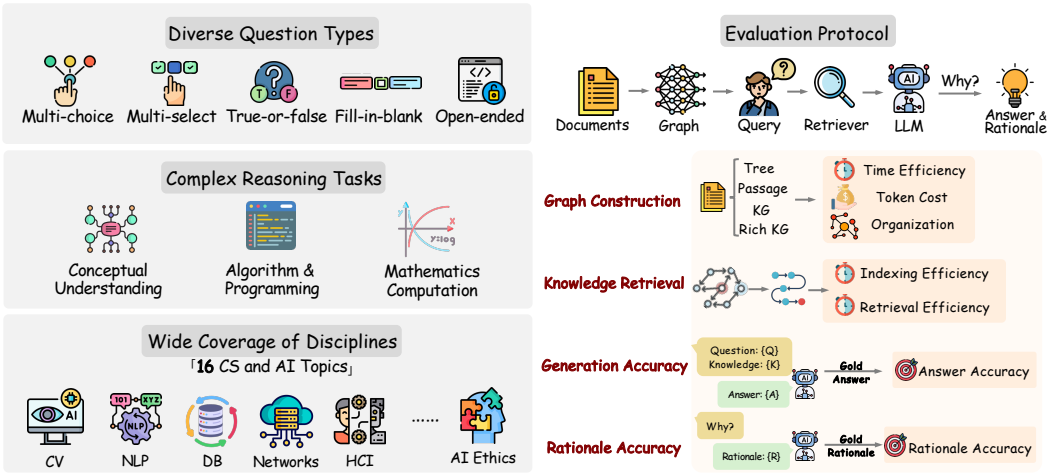

Figure 2: The overview of our benchmark `GraphRAG-Bench`, illustrating the contributions.

2 GraphRAG-Bench: Challenging Reasoning Benchmark

2.1 Question Design

To evaluate the GraphRAG framework on college-level reasoning, we first assembled an authoritative textbook corpus. Beginning with over 100 publications spanning 16 distinct subfields in computer science, we systematically identified the most representative 20 textbooks (the details of licenses can be found in ethics statement). We defined five types of questions, each targeting a different aspect of GraphRAG's reasoning capabilities, which are detailed in Tab. 1. After rigorous screening and refinement by several domain experts, we selected 1,018 high-quality challenging questions, covering a broad spectrum of topics.

By design, each question type is explicitly mapped to the core competencies of GraphRAG, with individual questions meticulously crafted for application in college-level instructional or assessment contexts. Should GraphRAG demonstrate improved performance on these tasks, it would establish itself as a highly effective tool in education, significantly enhancing teaching and learning efficiency.

Table 1: The description of different question types.

Question Type	**Description**
Fill-in-blank (FB)	Requires completing context-dependent statements with semantically precise terms. These assess the model's ability to generate coherent content by leveraging local semantic dependencies and entity grounding within graph-structured knowledge.
Multi-choice (MC)	Presents a question with 4 options, including linguistically plausible distractors. These assess the model's capacity to discern correct answers through discriminative reasoning, integrating entity information and edge relationships to reject semantically similar but factually incorrect options.
Multi-select (MS)	Demands selecting 2–4 correct answers from 4 options, often requiring reasoning over interconnected concepts. The inclusion of overlapping distractors tests the model's ability to handle complex query semantics, aggregating evidence from multi-hop graph paths and resolving conflicts between related but non-essential attributes.
True-or-false (TF)	Involves verifying the correctness of statements. These measure the model's factual accuracy, requiring logical inference over knowledge.
Open-ended (OE)	OE questions allow for a wide range of responses, requiring methods to formulate detailed and comprehensive answers. These evaluate the model's holistic knowledge synthesis, demanding the integration of multi-subfield knowledge to generate structured, logically coherent long-form responses.

2.2 CORPUS COLLECTION AND PROCESSING

Extracting accurate content from 20 PDF textbooks is challenging. We implement a multi-stage pipeline comprising preprocessing, content parsing, post-processing, and hierarchy construction, the details of which can be found in the Appendix. In preprocessing we separate text and scanned pages by text density and image-area ratio, extracting text with PyMuPDF or OCR as needed, and gather metadata (outline, page count, chapter/section ranges). For parsing we apply LayoutLMv3 Huang et al. (2022) to segment semantic regions (titles, paragraphs, figures, tables), detect formulas with a YOLO-based model Wang et al. (2024a) so formula images are handled separately, and transcribe scanned regions with PaddleOCR in reading order. Post-processing merges and reorders fragmented or overlapping regions into natural reading order using MinerU Wang et al. (2024b).

Finally, we organize the extracted content into a hierarchical textbook-tree structure. We map the textbook metadata (e.g., chapter titles, section divisions, and page ranges) to a four-level hierarchy: Book Title $\rightarrow$ Chapter $\rightarrow$ Section (Subchapter) $\rightarrow$ Knowledge Content Unit. Each node in this hierarchy is annotated with its contextual metadata and its structural role. This textbook-tree provides an intuitive navigation framework aligned with the textbook's organization. The resulting corpus, with its accurate content extraction, structural annotation, and hierarchical organization, forms a robust basis for evaluating GraphRAG's ability to leverage organized textbook knowledge for context-rich reasoning and retrieval-augmented generation.

2.3 EXPERT-CRAFTED RATIONALE

Existing benchmarks typically supply only final answers or explicit graph paths; by contrast, our dataset supplies expert-crafted rationales that articulate the complete logical progression necessary to solve each problem. These rationales go beyond mere corpus aggregation; they are structured narratives that (i) isolate prerequisite concepts, (ii) describe the relationships among these concepts, and (iii) specify the inferential operations applied during problem solving. By tracing each step of logical inference and knowledge interaction, we can assess whether GraphRAG models truly generate contextually grounded explanations or simply exploit surface-level patterns.

To enable fine-grained, topic-specific evaluation, each question in our dataset carries two hierarchical labels: a broad subfield (Level 1, e.g., "Machine Learning") and a more granular concept (Level 2, e.g., "Unsupervised Learning"). These annotations structure our post-hoc analyses. For each topic, we measure not only the accuracy of the model's answer but also the degree to which its generated rationale aligns with the gold one. In this way, we convert evaluation into a multidimensional process, requiring models to produce both correct solutions and faithful reasoning patterns.

3 EXPERIMENTS

We conduct experiments on each submodule following GraphRAG's pipeline, which includes the **graph construction** (or similar specialized structures), **knowledge retrieval**, and **generation**. Additionally, since our dataset contains a gold rationale for each query, we require the GraphRAG method to generate **rationales** during the generation phase to evaluate its reasoning capabilities.

Metrics. We provide a succinct introduction to the core ideas of each metric; the full evaluation protocol and details can be found in the Appendix.

- **Graph construction.** We evaluate graph construction across three aspects: 1) Efficiency: the time required to build a complete graph. 2) Cost: the number of tokens consumed during graph construction. 3) Organization: the proportion of non-isolated nodes within the constructed graph.
- **Knowledge retrieval.** We evaluate retrieval from two dimensions: 1) indexing time, defined as the duration required to construct the vector database for retrieval; 2) average retrieval time, representing the mean time consumed for retrieval per query. Additionally, we summarize the retrieval operators employed by each method to assess the complexity of their retrieval mechanisms.
- **Generation.** We argue that the existing exact match metric is inappropriate, as correct answering does not necessitate word-by-word correspondence. Therefore, this paper introduces a new metric, Accuracy, defined as follows: 1) For OE and FB questions, both the output and groundtruth are fed into an LLM via our designed prompt, which assigns a score based on semantic alignment and

correctness. 2) For MC and TF, 1 point for the correct answer, 0 points for otherwise. 3) For MS, 1 point for a fully correct answer; 0.5 points for a subset; 0 points for incorrect answers.

- **Rationale.** We designed a prompt to feed both the rationale generated by GraphRAG method and gold rationale into a LLM, which assigns a reasoning score R to evaluate their semantic correspondence and reasoning consistency. Simultaneously, we developed an additional assessment metric, namely the AR metric, to determine whether the model is able to provide correct reasoning when it answers the question accurately. This metric serves to distinguish whether the model has merely guessed the correct answer or has actually engaged in proper logical reasoning to reach the correct answer, thereby offering a more comprehensive understanding of the model's performance.

Experiment setups. In our experiments, we evaluated the performance of nine state-of-the-art GraphRAG methods, including: 1) RAPTOR Sarthi et al. (2024); 2) LightRAG Guo et al. (2024); 3) GraphRAG Edge et al. (2025); 4) G-Retriever He et al. (2024); 5) HippoRAG Gutiérrez et al. (2024); 6) GFM-RAG Luo et al. (2025); 7) DALK Li et al. (2024); 8) KGP Wang et al. (2024c); 9) ToG Sun et al. (2024). To ensure a fair comparison across all methods, we adopted the same GPT-4o-mini as the default large language model. We imposed no max token length to limit the performance of individual methods. For methods requiring top-k selection, we uniformly set k=5. Regarding text chunking, the chunk size was consistently set to 1200 tokens. Except for the parameters standardized for fair comparison, all other parameters were configured to the optimal values reported in the original papers.

Table 2: Comparison of graph construction process.

Method	Token cost of graph construction	Time cost of graph construction	Organization
RAPTOR (2024)	10,142,221	20396.49s	-
KGP (2024)	15,271,633	17318.07s	46.03%
LightRAG (2024)	83,909,073	12976.22s	69.71%
GraphRAG (2025)	79,929,698	11181.24s	72.51%
G-Retriever (2024)	32,948,161	5315.27s	89.95%
HippoRAG (2024)	33,006,198	5051.41s	89.58%
DALK (2024)	33,007,324	**4674.30s**	89.49%
ToG (2024)	33,008,230	5235.30s	89.95%
GFM-RAG (2025)	32,766,094	5631.10s	**89.97%**

3.1 Evaluation of graph construction

Graph construction aims to transform corpus into structured, storable objects, serving as the foundational step in GraphRAG. Current mainstream graph construction methods can be categorized into four classes: 1) Tree: RAPTOR leverages this structure, where each leaf node represents a chunk. By generating summaries via LLMs and applying clustering methods, parent nodes are iteratively created to form a tree structure. 2) Passage Graph: Adopted by KGP, this structure represents each chunk as a node, with edges established through entity linking tools. 3) Knowledge Graph: Used in G-Retriever, HippoRAG, GFM-RAG, and DALK, this structure extracts entities and relationships from chunks using open information extraction (OpenIE) tools to construct knowledge graphs. 4) Rich Knowledge Graph: Employed by GraphRAG and LightRAG, this structure enriches standard knowledge graphs with additional information (e.g., summarizing descriptions for nodes or edges).

Experimental results in Tab. 2 show that the tree structure incurs the lowest token count, as it only invokes LLMs for summary generation, but requires the longest time due to iterative clustering. The passage graph has suboptimal token cost, invoking LLMs only for summarizing entities or relationships, with the second-longest time consumption attributed to the time-intensive entity linking process. The knowledge graph has moderate token usage, requiring LLMs for both entity extraction from corpora and triple generation from entities, yet achieves the shortest time consumption due to rapid knowledge graph construction after triple acquisition. The rich knowledge graph consumes the most tokens, as it generates additional descriptions for entities and relationships via LLMs on top of standard knowledge graphs, leading to increased time costs. For evaluating graph construction quality, we use the non-isolated nodes ratio as the metric. Since the Tree structure contains no isolated nodes, this metric is inapplicable to it. Experimental results show that the Knowledge Graph

achieves the best performance, with its non-isolated nodes ratio maintained at approximately 90%. The Rich Knowledge Graph performs suboptimally; while it incorporates additional information, it inevitably introduces more noise. The Passage Graph exhibits the lowest non-isolated nodes ratio, indicating that entity linking tools fail to effectively establish edges between most entity pairs.

Table 3: Comparison of knowledge retrieval process.

Method	Retrieval operators	Indexing time	Average retrieval time
KGP	Node	204.10s	89.38s
ToG	Node+Relationship	1080.43s	70.53s
GraphRAG	Node+Relationship+Chunk+Community	1796.65s	44.87s
DALK	Node+Subgraph	407.10s	26.80s
G-Retriever	Node+Relationship+Subgraph	920.39s	23.77s
LightRAG	Node+Relationship+Chunk	1430.32s	13.95s
HippoRAG	Node+Relationship+Chunk	4695.29s	2.44s
GFM-RAG	Node	**93.55s**	1.96s
RAPTOR	Node	451.03s	**0.02s**

3.2 Evaluation of knowledge retrieval

As shown in Tab. 3. GFM-RAG incurs the shortest indexing time; it does not construct a traditional vector database to store entities but instead stores question-corresponding entities exclusively during graph construction. Among methods using vector databases, KGP, RAPTOR, and DALK exhibit lower costs due to minimal stored information; ToG, G-Retriever, and LightRAG have moderate costs, as relationship storage is inherently time-consuming; GraphRAG further increases indexing time by additionally storing community reports. HippoRAG demands the longest indexing time, attributed to its extra construction of entity↔relationship and relationship↔chunk mappings. Regarding average retrieval time, RAPTOR achieves the fastest speed, as its tree structure enables rapid information localization. GFM-RAG and HippoRAG follow, leveraging GNNs and PageRank for retrieval, respectively. G-retriever employs a prize-collecting Steiner forest algorithm, while LightRAG relies on relationship-based retrieval, both introducing additional latency. GraphRAG needs to utilize community information for retrieval, which leads to its time-consuming. KGP, ToG, and DALK incur substantial time costs due to their dependence on LLM invocations during retrieval.

Table 4: Comparison of generation process.

Method	Accuracy					
	Fill-in-blank	Multi-choice	Multi-select	True-or-false	Open-ended	Average
GPT-4o-mini	74.29	81.11	76.68	75.95	52.23	70.68
TF-IDF	75.71	77.88	72.52	84.17	50.18	71.71↑
BM-25	74.28	78.80	71.17	**84.49**	50.00	71.66↑
DALK	70.00	78.34	71.62	77.22	51.49	69.30↓
G-Retriever	70.95	77.42	71.62	78.80	52.04	69.84↓
LightRAG	65.24	78.80	73.42	82.59	53.16	71.22↑
ToG	70.48	78.80	**78.38**	79.75	54.28	71.71↑
KGP	74.29	79.26	74.77	82.28	51.49	71.86↑
GFM-RAG	72.38	80.65	72.07	82.59	52.79	72.10↑
GraphRAG	75.24	**81.57**	77.48	80.70	52.42	72.50↑
HippoRAG	70.48	80.18	74.32	81.65	**56.13**	72.64↑
RAPTOR	**76.67**	80.65	77.48	82.28	54.83	**73.58**↑

3.3 Evaluation of generation accuray

As shown in Tab.4. Given that GPT-4o-mini already exhibits strong question-answering capabilities, not all GraphRAG methods effectively enhance its performance. Notably, DALK and G-Retriever

degrade LLM performance; their over-reliance on structural information at the expense of semantic content introduces excessive noise during generation, impairing LLM judgment accuracy. LightRAG, ToG, and KGP achieve slight performance improvements, indicating their retrieved content provides marginal assistance for generation tasks. In contrast, GFM-RAG, GraphRAG, and HippoRAG significantly boost LLM performance by effectively integrating graph structural information with chunk-level semantics: GFM-RAG leverages large-scale pretraining to obtain a robust foundation model, GraphRAG optimizes retrieval using community-based information, and HippoRAG enhances retrieval efficiency via PageRank algorithm. The top-performing method in experiments is RAPTOR, which constructs a tree structure through iterative clustering, a design that aligns with the natural hierarchical organization of textbook data, enabling efficient retrieval of relevant information. Additionally, most GraphRAG methods outperform traditional RAG baselines such as BM-25 and TF-IDF, highlighting the utility of graph-based architectures in improving generation accuray.

Table 5: Comparison of reasoning capability.

Method	Reasoning											
	FB		MC		MS		TF		OE		Average	
	R	AR	R	AR	R	AR	R	AR	R	AR	R	AR
GPT-4o-mini	64.76	53.33	55.07	50.92	54.50	39.19	58.23	53.40	49.26	9.76	55.45	39.78
TF-IDF	68.09	52.61	52.76	49.19	56.30	43.02	64.08	61.23	50.37	10.50	57.61	42.38
BM-25	69.04	56.42	57.14	53.11	57.20	42.79	65.18	62.18	50.74	11.52	59.18	44.15
DALK	70.95	55.24	54.15	50.35	59.01	46.40	62.18	58.23	54.09	9.67	58.89	42.12
KGP	64.29	49.29	56.45	52.07	58.11	44.37	64.08	60.68	52.42	8.92	58.74	42.22
GraphRAG	**71.43**	55.24	56.22	52.42	57.66	45.72	63.61	60.13	53.16	10.50	59.43	43.30
G-Retriever	70.00	55.00	**57.60**	**53.46**	60.81	48.20	64.24	60.21	53.35	10.04	60.17	43.66
LightRAG	66.19	47.86	57.14	52.30	**61.71**	**49.10**	66.61	63.45	53.16	10.13	60.46	43.81
ToG	70.00	53.10	56.00	51.73	57.21	45.72	65.66	62.26	54.46	12.08	60.17	44.01
GFM-RAG	70.00	54.76	56.22	52.07	58.11	45.50	66.46	**63.69**	53.72	10.69	60.36	44.30
HippoRAG	66.67	50.48	56.68	52.30	59.91	47.52	**67.25**	63.61	**55.02**	12.36	**60.90**	44.55
RAPTOR	**71.43**	**57.86**	56.45	52.07	60.36	**49.10**	66.30	62.90	53.90	**13.57**	60.81	**45.53**

3.4 Evaluation of reasoning capability

As shown in Tab.5. In contrast to the high accuracy in generation tasks, GPT-4o-mini exhibits a notable decline in reasoning performance. The decrease in R score indicates that LLMs often fail to perform correct reasoning, instead selecting answers through conjecture or pattern matching in many cases. The drop in AR score suggests that even when LLMs provide correct answers, their reasoning processes may be flawed; alternatively, they might generate correct reasoning but choose incorrect answers. Importantly, all GraphRAG methods significantly enhance the reasoning capabilities of LLMs: through distinct designs, these methods retrieve not only semantically relevant corpus for questions but also identify multi-hop dependent corpus in the knowledge base, providing evidential support for LLM reasoning. This enables LLMs to reason based on external information rather than relying solely on internal knowledge for conjecture. In terms of algorithm performance, the distribution aligns with that of generation tasks: HippoRAG and RAPTOR remain the top performers, which is intuitive, since retrieving useful information is inherently correlated with enabling correct reasoning. Additionally, most GraphRAG methods still outperform traditional RAG baselines.

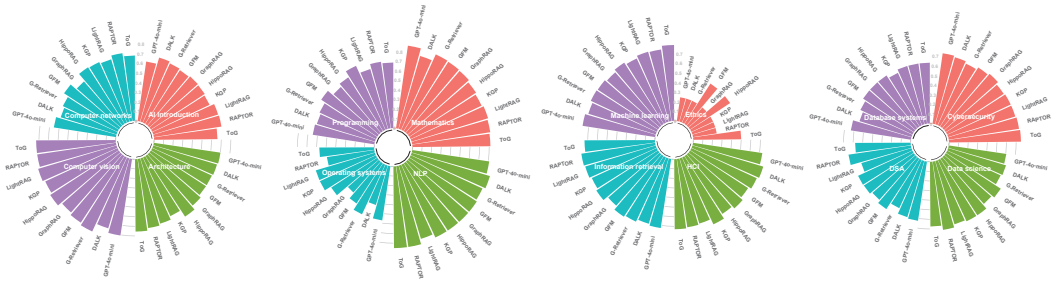

Figure 3: Comparison of Generation Accuracy by Topic.

3.5 TOPIC-SPECIFIC GENERATION ACCURACY ANALYSIS

Given our dataset spans 16 distinct topical domains, we conducted a fine-grained analysis of GraphRAG's impact on LLM generation accuracy. Overall, GraphRAG yields consistent improvements in most areas; However, several intriguing findings emerge: **1) Mathematics Domain.** All GraphRAG methods degrade the LLM's generation accuracy in mathematics. This is attributed to the critical reliance of mathematical problems on rigorous symbolic manipulation and precise reasoning chains; models must internally "compute" each deductive step rather than relying on keyword matching from external texts. Most documents retrieved through GraphRAG are explanatory or conceptual, with symbolic notation, formula layouts, and contextual structures often misaligned with the problem requirements, leading to ambiguities or loss of key steps during the extraction and transformation of information. **2) Ethics Domain.** Both GraphRAG and the LLM itself exhibit mediocre performance in ethics. We posit that ethical problems fundamentally involve subjective value judgments, whose meanings depend on dynamic contexts of moral trade-offs and social norms. The symbolic representations captured by LLMs through statistical learning struggle to accurately model ambiguous ethical constructs, introducing intrinsic limitations in reasoning. **3) Robustness.** Excellent GraphRAG approaches such as RAPTOR enhance LLM generation accuracy across most topics, demonstrating robust performance that validates their cross-domain effectiveness.

3.6 EXPERT-GUIDED POST VERIFICATION

For Open-ended and Fill-in-blank question types and reasoning, we believed that directly using string exact match was unreasonable. Many correct statements were judged incorrectly due to descriptions, capitalization, abbreviations. In such cases, we included LLM for auxiliary judgment, which is a widely adopted paradigm in generative tasks. During the actual post-processing, to verify the accuracy of LLM-as-a-judge, we have conducted expert-guided post verification. We sampled 500 questions from the dataset and asked 3 human experts (researchers with a computer science doctoral degree) to score them. The experts judged whether the LLM's judgment was reasonable based on the LLM's generation and the groundtruth and gave 1 score if yes. 3/3 points indicated reasonable and an aligned answer, 2/3 points indicated basically reasonable since the majority of experts agreed, and 1/3 or 0/3 point indicated unreasonable and needed to be reevaluted by the experts. The specific experimental results are shown in the Tab. 6, proving the reliability of LLM-as-a-judge.

Table 6: Expert majority voting-guided post verification for LLM-as-a-judge scenario.

Reliability	Counts	Percentage
3/3	470	94.0%
2/3	28	5.6%
1/3	1	0.2%
0/3	1	0.2%

4 OBSERVATION

'Can GraphRAG improve performance across all question types?'

Accuracy drop of MC questions. LLMs have internalized vast amounts of knowledge through extensive training on large corpora, enabling them to often correctly select answers in multiple-choice tasks. However, GraphRAG's retrieval-based augmentation may introduce redundant or loosely related information that does not precisely match the question context. Such retrieval noise can interfere with the model's decision-making ability, ultimately reducing its accuracy on MC questions.

Improvement in TF questions. TF questions require binary judgments about factual or logical statements. LLMs may contain blind spots or incomplete knowledge for certain facts, leading to incorrect answers. By retrieving relevant factual evidence, GraphRAG helps the model verify statements before answering. These supplementals improve the model's accuracy on TF questions.

Improvement in OE questions. Open-ended questions allow for expansive, detailed responses, which can be challenging for LLMs that rely solely on their internal knowledge. GraphRAG mitigates this challenge by providing additional context and facts from external corpora. The retrieved information enriches the model's responses, improves subject-matter detail and expressiveness, and reduces instances of hallucination by grounding answers in explicit evidence.

Different effects in FB & MS questions. Fill-in-blank questions demand precise contextual understanding to correctly predict missing words. GraphRAG's retrieved corpora often fail to match exact contexts, introducing noise that degrades the model's performance on FB questions. Multi-select questions require choosing multiple correct answers from a set and involve reasoning over complex combinations of options; if GraphRAG's retrieval omits relevant answer options or includes irrelevant details, it can confuse the model. As a result, these question types place high demands on retrieval precision; GraphRAG may have limited benefit unless its retrieval is highly accurate.

'Can GraphRAG effectively enhance LLMs' reasoning ability?'

Experiments demonstrate that GraphRAG enhances the reasoning capabilities of LLMs across diverse question types, increasing the probability of generating correct rationales alongside answers. This is attributed to their efficient retrieval mechanisms, which not only identify relevant corpora for questions but also provide robust evidential support for LLM reasoning processes. In particular, existing benchmarks lack systematic evaluation of GraphRAG's reasoning capabilities, an aspect of critical importance in real-world applications. For example, in the college-level educational context targeted in this document, users seeking professional knowledge expect not only correct answers, but also explicit rationales to facilitate understanding and knowledge acquisition. Similarly, in medical scenarios, patients require clear rationales for medication along with treatment recommendations to ensure transparency in decision-making. Thus, an effective GraphRAG approach should aim not only for high accuracy in answer generation but also for strong reasoning and explainability. Providing clear evidence-based justifications and reasoning chains is essential for meeting the requirements of explainability and transparency in real-world scenarios.

Case Study

Question: Why is it necessary for the server to use a special initial sequence number (ISN) in the SYN-ACK?

Multi-hop Reasoning:
1. <Server>→*sends*→ <SYN-ACK Packet> →*includes*→ < ISN > →*used to ensure*→ <Unique connection identification>
2. <Server>→*sends*→ <SYN-ACK Packet> →*includes*→ < ISN > →*used to ensure*→ < Proper packet sequencing >
3. < Special ISN > →*helps defend against*→ <SYN flood attack> →*exploits*→ <Predictability of ISN>

Rationale:
The server uses a special initial sequence number in the SYN-ACK to ensure unique connection identification and proper packet sequencing. This also mitigates SYN flood attacks by making it harder for attackers to predict ISNs and hijack sessions.

Figure 4: A case study in the topic of computer networks.

5 Case Study

As illustrated in Fig 4, we present a case study highlighting specific challenges within our dataset. Our questions span 16 core topics in undergraduate computer science; here, we focus on a sample from the Computer Networks section. This example demonstrates that (i) the questions demand specialized, college-level knowledge, and (ii) the correct answer cannot be retrieved through simple lookup. Instead, solving the problem requires synthesizing multiple reasoning steps to construct a coherent rationale before generating the final answer.

6 Conclusion

In this paper, we present GraphRAG-Bench, the first domain-specific benchmark designed for GraphRAG, comprising a 16-discipline dataset that challenges methods with multi-hop reasoning, complex algorithmic/programming tasks, mathematical computing, and varied question types. Our comprehensive, multi-dimensional evaluation, spanning graph construction, knowledge retrieval, generation and reasoning, quantifies the enhancement of LLM reasoning when augmented with structured knowledge. Extensive experiments on nine state-of-the-art GraphRAG methods reveal the significant role of graph integration in improving reasoning and generation performance. Our analysis reveals critical insights about graph architectures, retrieval efficacy, and reasoning capabilities, offering actionable guidance for the research community.

Ethics statement

License Information: As a pure research paper with no commercial purposes, we have already obtained full licenses to do research for 17 textbooks through the university platform among the 20 textbooks we adopted in total. For the remaining three textbooks, two of them are totally free with no copyright and one supports research with full textual resources online. For the processing of all the textbooks, we only keep the main content in the form of chunks with no tampering.

Our benchmark aims to fairly evaluate the existing open-sourced methods with the textbooks, which we have obtained the licenses. All data usage strictly adheres to the terms of the respective licenses and is confined to non-commercial research purposes. The processed textbook chunks are used solely for generating evaluation queries and will not be redistributed. We are committed to the responsible use of copyrighted materials and believe this work aligns with ethical research practices by promoting reproducible and fair comparisons.

Reproducibility statement

To ensure complete reproducibility of our work, we have elaborately detailed the construction, parameters, and evaluation metrics of our benchmark in the main text, including key implementation details. The evaluated baselines are all open-sourced. All relevant resources covering the complete dataset and well-documented source codes have been made publicly available and can be found in the supplementary materials.

References

Junnan Dong, Qinggang Zhang, Xiao Huang, Keyu Duan, Qiaoyu Tan, and Zhimeng Jiang. Hierarchy-aware multi-hop question answering over knowledge graphs. In *WWW*, pp. 2519–2527, 2023.

Junnan Dong, Siyu An, Yifei Yu, Qian-Wen Zhang, Linhao Luo, Xiao Huang, Yunsheng Wu, Di Yin, and Xing Sun. Youtu-graphrag: Vertically unified agents for graph retrieval-augmented complex reasoning. *arXiv preprint arXiv:2508.19855*, 2025.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization, 2025. URL https://arxiv.org/abs/2404.16130.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024. URL https://arxiv.org/abs/2312.10997.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. 2024.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 59532–59569. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/6ddc001d07ca4f319af96a3024f6dbd1-Paper-Conference.pdf.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 132876–132907. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/efaf1c9726648c8ba363a5c927440529-Paper-Conference.pdf.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th*

International Conference on Computational Linguistics, pp. 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. URL https://www.aclweb.org/anthology/2020.coling-main.580.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, pp. 4083–4091, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392037. doi: 10.1145/3503161.3548112. URL https://doi.org/10.1145/3503161.3548112.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Dawei Li, Shu Yang, Zhen Tan, Jae Young Baik, Sukwon Yun, Joseph Lee, Aaron Chacko, Bojian Hou, Duy Duong-Tran, Ying Ding, Huan Liu, Li Shen, and Tianlong Chen. DALK: Dynamic co-augmentation of LLMs and KG to answer Alzheimer's disease questions with scientific literature. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 2187–2205, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.119. URL https://aclanthology.org/2024.findings-emnlp.119/.

Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Dinh Phung, Chen Gong, and Shirui Pan. Gfm-rag: Graph foundation model for retrieval augmented generation, 2025. URL https://arxiv.org/abs/2502.01113.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546/.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5336–5358, Seattle, United States, July 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.naacl-main.391.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. Graph retrieval-augmented generation: A survey, 2024. URL https://arxiv.org/abs/2408.08921.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=GN921JHCRw.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=nnVO1PvbTv.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yolov10: Real-time end-to-end object detection. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 107984–108011. Curran Associates, Inc., 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/c34ddd05eb089991f06f3c5dc36836e0-Paper-Conference.pdf.

Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, Bo Zhang, Liqun Wei, Zhihao Sui, Wei Li, Botian Shi, Yu Qiao, Dahua Lin, and Conghui He. Mineru: An open-source solution for precise document content extraction, 2024b. URL https://arxiv.org/abs/2409.18839.

Yu Wang, Nedim Lipka, Ryan A Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. Knowledge graph prompting for multi-document question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19206–19214, 2024c.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Qinggang Zhang, Shengyuan Chen, Yuanchen Bei, Zheng Yuan, Huachi Zhou, Zijin Hong, Junnan Dong, Hao Chen, Yi Chang, and Xiao Huang. A survey of graph retrieval-augmented generation for customized large language models, 2025. URL https://arxiv.org/abs/2501.13958.

Yingli Zhou, Yaodong Su, Youran Sun, Shu Wang, Taotao Wang, Runyuan He, Yongwei Zhang, Sicong Liang, Xilin Liu, Yuchi Ma, and Yixiang Fang. In-depth analysis of graph-based rag in a unified framework, 2025. URL https://arxiv.org/abs/2503.04338.

A APPENDIX

A.1 THE USAGE OF LLMS

The use of LLM in this manuscript is fully in accordance with the regulations of the ICLR, and it is only used for detecting grammatical errors.

A.2 RELATED WORK

GraphRAG. Recent work in GraphRAG has focused on integrating structured knowledge and advanced retrieval strategies to overcome the limitations of vanilla RAG in handling large, noisy corpora and complex reasoning. For example, RAPTOR Sarthi et al. (2024) and Microsoft's GraphRAG Edge et al. (2025) both employ hierarchical clustering, RAPTOR via recursive tree construction with multi-level summarization, and GraphRAG via community detection with LLM-generated synopses, to support coarse-to-fine retrieval and diverse, high-coverage responses. GFM-RAG Luo et al. (2025), G-Retriever He et al. (2024), and LightRAG Guo et al. (2024) each combine graph neural encoders with specialized retrieval objectives, respectively a query dependent GNN trained in two stages for multi-hop generalizability, a Prize Collecting Steiner Tree formulation to reduce hallucination and improve scalability, and a dual level graph augmented index for efficient, incrementally updatable lookup, to enable accurate, scalable reasoning over document graphs. Inspired by hippocampal memory processes, HippoRAG Gutiérrez et al. (2024) leverages Personalized PageRank to achieve single-step multi-hop retrieval, delivering state-of-the-art efficiency and performance on both path following and path finding QA tasks. DALK Li et al. (2024) and KGP Wang et al. (2024c) introduce dynamic KG construction and traversal agents, using LLMs to build domain specific graphs and self aware retrieval policies, to inject structural context while reducing noise. ToG Sun et al. (2024) tightly couples LLMs with KGs via beam search exploration, enabling iterative graph reasoning and on the fly correction without additional training. Collectively, these methods exemplify the GraphRAG paradigm by uniting graph structures, generative language models, and novel retrieval formulations to enhance knowledge integration, scalability, and deep reasoning across diverse domains.

Prior benchmarks for GraphRAG. To date, no dataset has been specifically designed for GraphRAG tasks. Widely used datasets such as Quality Pang et al. (2022), PopQA Mallen et al. (2023), and HotpotQA Yang et al. (2018) are tailored for general question answering, where answers can often be directly extracted from corpora, failing to effectively measure the core capabilities of GraphRAG methods. Multi-hop QA datasets like MusiqueQA Trivedi et al. (2022) and 2WikiMultiHopQA Ho et al. (2020) contain questions artificially constructed via rules and logic, rather than natural queries from real-world scenarios. Additionally, their corpora are short and often derived from converting entities and descriptions of existing KGs, which deviates from practical application contexts. While DIGIMON Zhou et al. (2025) benchmarks some methods, it neither introduces new datasets nor evaluates the reasoning capabilities of GraphRAG. Critically, all aforementioned datasets neglect question type distinctions, focusing primarily on simple questions and thus unable to reflect GraphRAG's performance variations across different question categories. In summary, existing datasets lack long contexts and raw documents, mismatching real-world scenarios, and omit gold rationale, making it impossible to systematically evaluate GraphRAG's reasoning abilities.

A.3 ADDITIONAL EXPERIMENTS

A.3.1 TOPIC-SPECIFIC REASONING ANALYSIS

Given that our dataset encompasses 16 thematic domains, we conducted experiments to analyze the reasoning capabilities of GraphRAG across different topics. Results indicate that the large language model (LLM) based on GPT-4o-mini demonstrates significant improvements in reasoning through GraphRAG across most domains. However, the following intriguing observations were made:

Operating System Domain. The LLM exhibits suboptimal performance in this domain. While GraphRAG provides marginal improvements in reasoning capabilities, overall scores remain low. This is primarily attributed to the highly specialized, systematic, and logically complex nature of operating system knowledge, which involves multi-layered principles such as process scheduling, memory management, and file systems, requiring precise grasp of conceptual definitions, algorith-

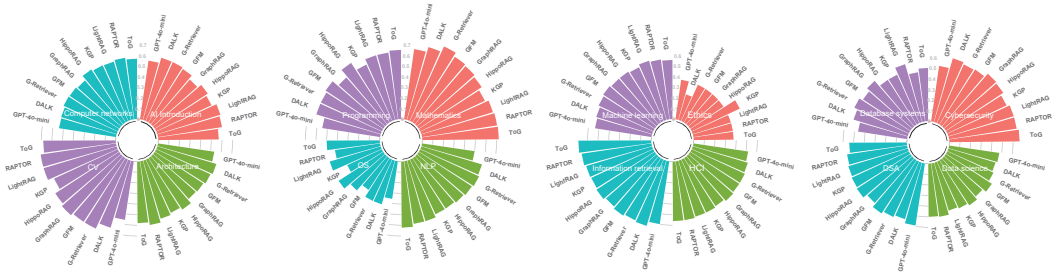

Figure 5: Comparison of R score by Topic.

mic workflows, and causal relationships between entities. General-purpose training data for LLMs often lack comprehensive coverage of such granular knowledge systems, and the models themselves have inherent limitations in structured logical reasoning.

Ethics Domain. Consistent with the generation accuracy results, LLMs face substantial challenges in reasoning about ethical questions. Ethical problems fundamentally involve subjective value judgments, whose meanings are rooted in dynamic contexts of moral trade-offs and social norms. The symbolic representations captured by LLMs through statistical learning struggle to accurately model ambiguous ethical constructs, leading to intrinsic difficulties in both generating correct answers and constructing valid reasoning chains.

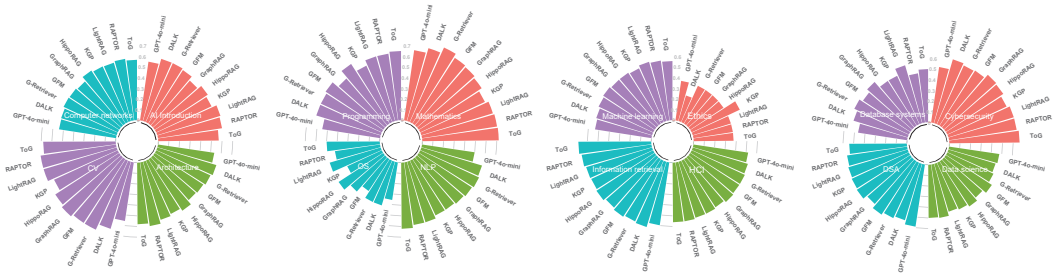

Figure 6: Comparison of AR score by Topic.

We further evaluated the AR scores of GraphRAG across different topics. Experimental results show that AR scores generally align with R scores in most cases. However, a notable observation emerges in the database systems domain: AR scores are significantly lower than R scores, indicating a high prevalence of "correct reasoning but incorrect answering" in LLMs, where reasoning steps diverge from final answer generation. This discrepancy arises because database system problems require models to reference specialized concepts such as relational algebra operations, transaction isolation levels, ACID properties, and query optimizer cost models, yet models do not perform formal computations or analyze critical factors like underlying data distribution and index selectivity. Although models may decompose processes like schema design or concurrency control according to human logical paradigms in chain-of-thought reasoning, their token selection during answer generation prioritizes statistical fluency from training corpora over contextual logical accuracy. The strict requirements for precise logical operations (e.g., cost estimation, deadlock detection) in database tasks create a fundamental mismatch with the model's learned fuzzy statistical patterns from general text, leading to reasoning chains that appear plausible in intermediate steps but produce erroneous conclusions at technical junctures, such as failing to execute physical query optimization calculations, due to the absence of real-world logical validation.

A.3.2 Evaluation of generation accuracy (with the deviations)

For all the generation results in the main text, they are the average values obtained after conducting five tests for each algorithm. Due to space limitations, we are unable to include all the details in the main text. Therefore, we are providing here the results with deviations, as shown in the Tab. 7:

Table 7: Comparison of generation process with the deviations.

Method	Accuracy					
	Fill-in-blank	Multi-choice	Multi-select	True-or-false	Open-ended	Average
DALK	70.00±0.32	78.34±0.85	71.62±0.58	77.22±0.92	51.49±0.25	69.30±0.58↓
G-Retriever	70.95±0.41	77.42±0.79	71.62±0.63	78.80±0.95	52.04±0.28	69.84±0.61↓
LightRAG	65.24±0.35	78.80±0.88	73.42±0.67	82.59±0.97	53.16±0.31	71.22±0.64↑
ToG	70.48±0.43	78.80±0.82	**78.38±0.71**	79.75±0.93	54.28±0.33	71.71±0.65↑
KGP	74.29±0.47	79.26±0.89	74.77±0.74	82.28±0.96	51.49±0.26	71.86±0.66↑
GFM-RAG	72.38±0.45	80.65±0.91	72.07±0.69	82.59±0.98	52.79±0.29	72.10±0.66↑
GraphRAG	75.24±0.50	**81.57±0.94**	77.48±0.76	80.70±0.90	52.42±0.27	72.50±0.68↑
HippoRAG	70.48±0.44	80.18±0.87	74.32±0.72	81.65±0.95	**56.13±0.35**	72.64±0.67↑
RAPTOR	**76.67±0.53**	80.65±0.93	77.48±0.78	82.28±0.97	54.83±0.32	**73.58±0.71**↑

A.3.3 Evaluation of reasoning capability (with the deviations)

For all the reasoning results in the main text, they are the average values obtained after conducting five tests for each algorithm. Due to space limitations, we are unable to include all the details in the main text. Therefore, we are providing here the results with deviations, as shown in the Tab. 8:

Table 8: Comparison of reasoning capability with the deviations.

Method	Reasoning											
	FB		MC		MS		TF		OE		Average	
	R	AR	R	AR	R	AR	R	AR	R	AR	R	AR
DALK	70.95±0.92	55.24±0.68	54.15±0.61	50.35±0.53	59.01±0.85	46.40±0.61	62.18±0.74	58.23±0.70	54.09±0.63	9.67±0.28	58.89±0.76	42.12±0.57
KGP	64.29±0.79	49.29±0.55	56.45±0.73	52.07±0.63	58.11±0.83	44.37±0.55	64.08±0.76	60.68±0.77	52.42±0.60	8.92±0.26	58.74±0.71	42.22±0.58
GraphRAG	**71.43±0.95**	55.24±0.69	56.22±0.70	52.42±0.65	57.66±0.80	45.72±0.58	63.61±0.74	60.13±0.73	53.16±0.62	10.50±0.33	59.43±0.77	43.30±0.62
G-Retriever	70.00±0.88	55.00±0.66	**57.60±0.83**	**53.46±0.71**	60.81±0.90	48.20±0.67	64.24±0.78	60.21±0.75	53.35±0.64	10.04±0.31	60.17±0.81	43.66±0.65
LightRAG	66.19±0.82	47.86±0.52	57.14±0.72	52.30±0.64	**61.71±0.93**	**49.10±0.69**	66.61±0.85	63.45±0.88	53.16±0.63	10.13±0.32	60.46±0.84	43.81±0.66
ToG	70.00±0.87	53.10±0.63	56.00±0.68	51.73±0.60	57.21±0.79	45.72±0.57	65.66±0.81	62.26±0.83	54.46±0.67	12.08±0.41	60.17±0.80	44.01±0.68
GFM-RAG	70.00±0.86	54.76±0.67	56.22±0.69	52.07±0.62	58.11±0.82	45.50±0.56	66.46±0.84	**63.69±0.89**	53.72±0.65	10.69±0.34	60.36±0.82	44.30±0.70
HippoRAG	66.67±0.83	50.48±0.58	56.68±0.71	52.30±0.63	59.91±0.87	47.52±0.63	**67.25±0.87**	63.61±0.88	**55.02±0.69**	12.36±0.42	**60.90±0.86**	44.55±0.71
RAPTOR	**71.43±0.94**	**57.86±0.75**	56.45±0.72	52.07±0.62	60.36±0.89	**49.10±0.68**	66.30±0.83	62.90±0.85	53.90±0.66	**13.57±0.47**	60.81±0.85	**45.53±0.76**

A.3.4 Comparisons with existing datasets.

Currently, most GraphRAG methods follow the experimental setting of HippoRAG, which is the most common and widely used setting in this field at present. It includes three public datasets: HotpotQA, 2WikiMultiHopQA, and MuSiQue which are all originally designed for multi-hop QA, not GraphRAG. They can hardly necessitate the use of graphs and showcase the use of graphs in complex QA scenarios.

In each of these data sets, 1000 questions were sampled. Therefore, following this standard and distribution, we selected 1018 of the most valuable questions as the final version. Meanwhile, grapragbench encompasses a wider range of question types (examining the model's robustness to various question formats), a larger corpus (making it more difficult to retrieve useful information), and covers multiple topics (requiring the model to understand different knowledge domains). We believe that the coverage scope is much broader from multiple perspectives than that of the current dataset. The specific comparison results are shown in the Tab. 9.

Dataset	**Question count**	**Corpus size**	**Question type**
HotpotQA	1000	6.16MB	1
2WikiMultiHopQA	1000	2.96MB	1
MuSiQue	1000	5.95MB	1
Ours	1018	41.30MB	5

Table 9: Comparisons with existing datasets.

A.3.5 COMPUTE RESOURCES

All code is done in Python, and experiments are conducted on H100*2 GPUs.

A.4 DETAILS OF CORPUS COLLECTION AND PROCESSING

Extracting accurate content from the 20 PDF-format core textbooks presents significant challenges. We implement a multi-stage pipeline comprising preprocessing, content parsing, post-processing, and hierarchy construction.

Textbook Preprocessing. 1) PDF Classification: To distinguish text-based pages from scanned (image-based) pages, we analyze each page's text density and image area proportion. Text-based pages are processed by extracting text directly using PyMuPDF, while scanned pages require optical character recognition (OCR) to extract their textual content. 2) Metadata Extraction: We extract metadata for each textbook, including its outline, total page count, and the page ranges for each chapter or section. This metadata supports the later construction of the document's logical structure.

Content Parsing. After preprocessing, we analyze each page's layout to extract textual and non-textual elements. 1) Layout Analysis: We apply LayoutLMv3 Huang et al. (2022) for multimodal document layout analysis. LayoutLMv3 is pre-trained with masked language modeling, masked image modeling, and cross-modal alignment, enabling it to learn rich representations of document pages. The model classifies page regions into semantic categories such as titles, paragraphs, figures, tables, or decorative/irrelevant elements. This segmentation yields coherent content blocks on each page. 2) Formula Recognition: Mathematical formulas embedded in text are often misrecognized by OCR. To prevent this, we first detect inline formulas using a pre-trained YOLO-based model Wang et al. (2024a) from PDF-Extract-Kit. This model identifies the bounding boxes of formula regions so that formula images can be extracted separately, ensuring that OCR does not garble the formula content. 3) OCR: In scanned PDFs, OCR is applied to recognize text regions. We use PaddleOCR to transcribe text from the regions labeled as titles and body paragraphs via layout analysis. This step produces the page's textual content in the correct reading order, while preserving non-text elements as separate objects.

Post-Processing. After parsing, the extracted elements (text blocks, formula, figures, tables, etc.) may be disordered due to overlapping bounding boxes or fragmented text lines. We resolve these issues by reordering and merging page regions according to human reading order. Concretely, we use MinerU Wang et al. (2024b) for post-processing, which partitions each page into logical reading regions and sequences them so that the final text flow matches the natural reading sequence.

Hierarchy Construction. Finally, we organize the extracted content into a hierarchical textbook-tree structure. We map the textbook metadata (e.g., chapter titles, section divisions, and page ranges) to a four-level hierarchy: Book Title $\rightarrow$ Chapter $\rightarrow$ Section (Subchapter) $\rightarrow$ Knowledge Content Unit. Each node in this hierarchy is annotated with its contextual metadata and its structural role. This textbook-tree provides an intuitive, pedagogical navigation framework aligned with the textbook's organization. The resulting corpus, with its accurate content extraction, structural annotation, and hierarchical organization, forms a robust basis for evaluating GraphRAG's ability to leverage organized textbook knowledge for context-rich reasoning and retrieval-augmented generation.

A.5 THE DETAILS OF METRICS

A.5.1 DETAILS OF AR SCORE.

The AR score is computed based on the combination of answer correctness (generation score) and rationale correctness (reasoning score), with the following evaluation rules:

- When both the answer and rationale are fully correct (generation score = 1 and reasoning score = 1), the AR score is 1.0.
- If the answer is correct but the rationale is partially correct (generation score = 1 and reasoning score = 0.5), the AR score is 0.5.
- When the answer is correct but the rationale is incorrect (generation score = 1 and reasoning score = 0), the AR score is 0.0.

- For incorrect answers with a fully correct rationale (generation score = 0 and reasoning score = 1), the AR score is 0.5.
- If both the answer is incorrect and the rationale is partially correct (generation score = 0 and reasoning score = 0.5), the AR score is 0.25.
- In all other cases (e.g., incorrect answer with incorrect or missing rationale), the AR score is 0.0.

This scoring scheme systematically captures the alignment between answers and their supporting reasoning, emphasizing the importance of both correctness and logical consistency in evaluating model performance.

A.5.2 Prompt of OE and FB questions.

Fig. 7 is the prompt used to generate the LLM-judge score for OE and FB questions.

Prompt of generation grading for OE and FB questions.

Instructions:
You are a strict evaluator. Compare the following two answers for correctness and completeness:

Predicted Answer:
<pred_answer>

Gold Answer:
<gold_answer>

Important Guidelines:
Please evaluate the predicted answer in comparison to the gold answer. Respond with a score between 0 and 1:
- 1: The predicted answer fully aligns with the gold answer.
- 0.5: The predicted answer is partially correct but lacks completeness or includes incorrect information.
- 0: The predicted answer is incorrect or completely misaligned with the gold answer.

Figure 7: Prompt of generation grading for OE and FB questions.

A.5.3 Prompt of reasoning grading.

Fig. 8 is the prompt used to evaluate the reasoning score R.

Prompt of rationale grading.

Instructions:
You are a strict evaluator. Compare the following two rationales for correctness and completeness:

Predicted Rationale:
<pred_rationale>

Gold Rationale:
<gold_rationale>

Important Guidelines:
Please evaluate the predicted rationale in comparison to the gold rationale. Respond with a score between 0 and 1:
- 1: The predicted rationale fully aligns with the gold rationale.
- 0.5: The predicted rationale is partially correct but lacks completeness or includes incorrect information.
- 0: The predicted rationale is incorrect or completely misaligned with the gold rationale.

Figure 8: Prompt of rationale grading.

A.6 LIMITATIONS OF EXISTING GRAPHRAG DATASETS

Through a systematic review of benchmark datasets used by contemporary Graph-RAG methods, we have identified four critical limitations that undermine both task suitability and the validity of evaluation results:

Superficial retrieval tasks. Most datasets pose questions that can be answered by straightforward text retrieval, without requiring deep integration of graph structure or sophisticated semantic reasoning. Consequently, models may achieve high scores by exploiting shallow keyword matching, offering no insight into their true capabilities in relational reasoning or entity-association modeling.

Synthetic and unrepresentative queries. Questions are typically generated via hand-crafted rules, yielding simplified language that lacks the domain-specific terminology, ambiguous intent, and syntactic variety found in real user queries. This synthetic distribution diverges sharply from natural problem settings, limiting the ecological validity of any conclusions about model generalization.

Cross-task misalignment. Many datasets are inherited from disparate tasks (e.g., knowledge-graph question answering) whose annotation schemes and answer formats do not align with the core objectives of Graph-RAG—namely, constructing and leveraging heterogeneous graph structures to guide multi-source information fusion. Transferring evaluation metrics across tasks therefore introduces inconsistencies that dilute the relevance of experimental findings for advancing Graph-RAG techniques.

Opaque reasoning evaluation. Existing benchmarks supply only final answers or explicit node sequences, but omit any structural or narrative annotation of the underlying inference process. Key decision points—such as why a particular graph subpath was selected or how evidence from multiple sources is reconciled—remain unexamined. Without annotated rationales, evaluation reduces to binary correctness checks and cannot assess a model's genuine reasoning competence.

These limitations collectively motivate the design of a dedicated benchmark that both challenges Graph-RAG models on core reasoning skills and provides richly structured annotations for fine-grained, interpretability-driven evaluation.

A.7 LIMITATIONS OF THIS PAPER

Despite the valuable contributions of this study, we acknowledge its limitations: (1) Our dataset currently only contains English content; more detailed research should be done in the future for different languages. (2) Other modal data such as images are not included in the current data set, and richer multimodal datasets can be considered in the future.