

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 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-

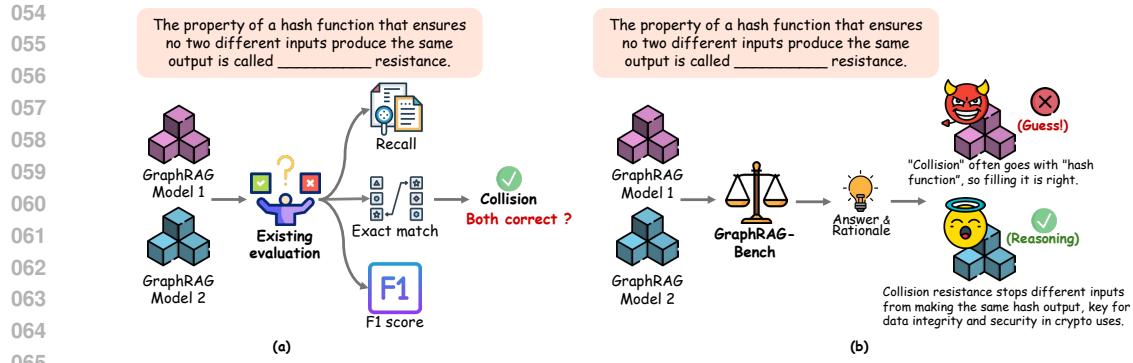


Figure 1: Comparison between existing evaluations (a) and our proposed GraphRAG-Bench (b). GraphRAG-Bench not only assesses the accuracy of generation but also evaluates the rationality of reasoning based on the challenging domain-specific questions.

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.

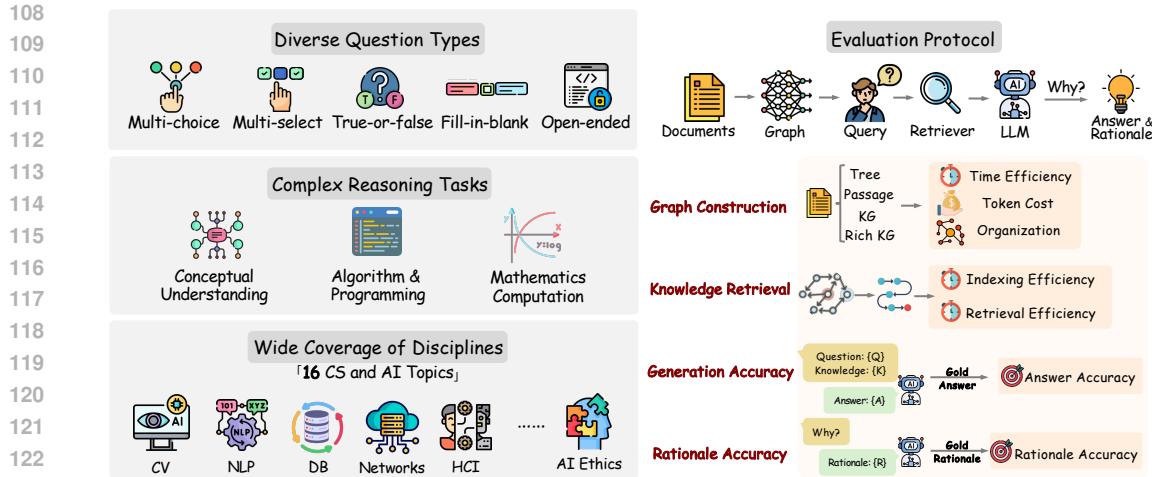


Figure 2: The overview of our benchmark GraphRAG-Bench, illustrating the contributions.

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. [The details of question design and selection can be found in the Appendix.](#)

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.

162 2.2 CORPUS COLLECTION AND PROCESSING
163

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

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

181 2.3 EXPERT-CRAFTED RATIONALE
182

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

190 To enable fine-grained, topic-specific evaluation, each question in our dataset carries two hierach-
191 ical labels: a broad subfield (Level 1, e.g., “Machine Learning”) and a more granular concept (Level
192 2, e.g., “Unsupervised Learning”). These annotations structure our post-hoc analyses. For each
193 topic, we measure not only the accuracy of the model’s answer but also the degree to which its gen-
194 erated rationale aligns with the gold one. In this way, we convert evaluation into a multidimensional
195 process, requiring models to produce both correct solutions and faithful reasoning patterns.

196 3 EXPERIMENTS
197

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

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

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

216 correctness. 2) For MC and TF, 1 point for the correct answer, 0 points for otherwise. 3) For MS,
 217 1 point for a fully correct answer; 0.5 points for a subset; 0 points for incorrect answers.
 218

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

226 **Experiment setups.** In our experiments, we evaluated the performance of nine state-of-the-art
 227 GraphRAG methods, including: 1) RAPTOR Sarthi et al. (2024); 2) LightRAG Guo et al. (2024);
 228 3) GraphRAG Edge et al. (2025); 4) G-Retriever He et al. (2024); 5) HippoRAG Gutiérrez et al.
 229 (2024); 6) GFM-RAG Luo et al. (2025); 7) DALK Li et al. (2024); 8) KGP Wang et al. (2024c);
 230 9) ToG Sun et al. (2024). To ensure a fair comparison across all methods, we adopted the same
 231 GPT-4o-mini as the default large language model. We imposed no max token length to limit the
 232 performance of individual methods. For methods requiring top-k selection, we uniformly set k=5.
 233 Regarding text chunking, the chunk size was consistently set to 1200 tokens. Except for the param-
 234 eters standardized for fair comparison, all other parameters were configured to the optimal values
 235 reported in the original papers.

236 Table 2: Comparison of graph construction process.

238 Category	239 Method	240 Token cost of graph construction	241 Time cost of graph construction	242 Organization
239 Tree	240 RAPTOR (2024)	241 10,142,221	242 20396.49s	243 -
240 Passage Graph	241 KGP (2024)	242 15,271,633	243 17318.07s	244 46.03%
241 Rich KG	242 LightRAG (2024)	243 83,909,073	244 12976.22s	245 69.71%
242 Rich KG	243 GraphRAG (2025)	244 79,929,698	245 11181.24s	246 72.51%
243 KG	244 G-Retriever (2024)	245 32,948,161	246 5315.27s	247 89.95%
244 KG	245 HippoRAG (2024)	246 33,006,198	247 5051.41s	248 89.58%
245 KG	246 DALK (2024)	247 33,007,324	248 4674.30s	249 89.49%
246 KG	247 ToG (2024)	248 33,008,230	249 5235.30s	250 89.95%
247 KG	248 GFM-RAG (2025)	249 32,766,094	250 5631.10s	251 89.97%

248 3.1 EVALUATION OF GRAPH CONSTRUCTION

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

260 Experimental results in Tab. 2 show that the tree structure incurs the lowest token count, as it only
 261 invokes LLMs for summary generation, but requires the longest time due to iterative clustering. The
 262 passage graph has suboptimal token cost, invoking LLMs only for summarizing entities or
 263 relationships, with the second-longest time consumption attributed to the time-intensive entity linking
 264 process. The knowledge graph has moderate token usage, requiring LLMs for both entity extraction
 265 from corpora and triple generation from entities, yet achieves the shortest time consumption due to
 266 rapid knowledge graph construction after triple acquisition. The rich knowledge graph consumes the
 267 most tokens, as it generates additional descriptions for entities and relationships via LLMs on top
 268 of standard knowledge graphs, leading to increased time costs. For evaluating graph construction
 269 quality, we use the non-isolated nodes ratio as the metric. Since the Tree structure contains no iso-
 270 lated nodes, this metric is inapplicable to it. Experimental results show that the Knowledge Graph
 271 achieves the best performance, with its non-isolated nodes ratio maintained at approximately 90%.

270 The Rich Knowledge Graph performs suboptimally; while it incorporates additional information, it
 271 inevitably introduces more noise. The Passage Graph exhibits the lowest non-isolated nodes ratio,
 272 indicating that entity linking tools fail to effectively establish edges between most entity pairs.
 273

274 Table 3: Comparison of knowledge retrieval process.
 275

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

286
 287 3.2 EVALUATION OF KNOWLEDGE RETRIEVAL
 288

289 As shown in Tab. 3. GFM-RAG incurs the shortest indexing time; it does not construct a traditional
 290 vector database to store entities but instead stores question-corresponding entities exclusively during
 291 graph construction. Among methods using vector databases, KGP, RAPTOR, and DALK exhibit
 292 lower costs due to minimal stored information; ToG, G-Retriever, and LightRAG have moderate
 293 costs, as relationship storage is inherently time-consuming; GraphRAG further increases indexing
 294 time by additionally storing community reports. HippoRAG demands the longest indexing time,
 295 attributed to its extra construction of entity \leftrightarrow relationship and relationship \leftrightarrow chunk mappings. Re-
 296 garding average retrieval time, RAPTOR achieves the fastest speed, as its tree structure enables rapid
 297 information localization. GFM-RAG and HippoRAG follow, leveraging GNNs and PageRank for
 298 retrieval, respectively. G-retriever employs a prize-collecting Steiner forest algorithm, while Light-
 299 RAG relies on relationship-based retrieval, both introducing additional latency. GraphRAG needs
 300 to utilize community information for retrieval, which leads to its time-consuming. KGP, ToG, and
 301 DALK incur substantial time costs due to their dependence on LLM invocations during retrieval.
 302

303 Table 4: Comparison of generation process.
 304

Category	Method	Accuracy					
		Fill-in-blank	Multi-choice	Multi-select	True-or-false	Open-ended	Average
Base LLM	GPT-4o-mini	74.29	81.11	76.68	75.95	52.23	70.68
Naive RAG	TF-IDF	75.71	77.88	72.52	84.17	50.18	71.71 \uparrow
	BM-25	74.28	78.80	71.17	84.49	50.00	71.66 \uparrow
	BERT-large	71.43	79.26	74.77	81.33	51.86	71.32 \uparrow
	BGE-M3	77.62	77.42	68.02	82.60	53.35	71.66 \uparrow
Passage Graph	KGP	74.29	79.26	74.77	82.28	51.49	71.86 \uparrow
Rich KG	LightRAG	65.24	78.80	73.42	82.59	53.16	71.22 \uparrow
Rich KG	GraphRAG	75.24	81.57	77.48	80.70	52.42	72.50 \uparrow
KG	DALK	70.00	78.34	71.62	77.22	51.49	69.30 \downarrow
KG	G-Retriever	70.95	77.42	71.62	78.80	52.04	69.84 \downarrow
KG	ToG	70.48	78.80	78.38	79.75	54.28	71.71 \uparrow
KG	GFM-RAG	72.38	80.65	72.07	82.59	52.79	72.10 \uparrow
KG	HippoRAG	70.48	80.18	74.32	81.65	56.13	72.64 \uparrow
Tree	RAPTOR	76.67	80.65	77.48	82.28	54.83	73.58\uparrow

319
 320 3.3 EVALUATION OF GENERATION ACCURAY
 321

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

Table 5: Comparison of reasoning capability.

Category	Method	Reasoning											
		FB		MC		MS		TF		OE		Average	
		R	AR										
Base LLM	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
Naive RAG	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
	BERT-large	67.62	52.14	56.22	52.07	55.41	44.59	65.66	62.26	52.60	10.97	58.99	43.27
	BGE-M3	67.14	53.10	54.61	51.04	58.11	44.14	63.77	60.36	52.97	11.34	59.01	43.37
Passage Graph	KG	64.29	49.29	56.45	52.07	58.11	44.37	64.08	60.68	52.42	8.92	58.74	42.22
Rich KG	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
Rich KG	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
KG	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
KG	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
KG	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
KG	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
KG	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
Tree	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.

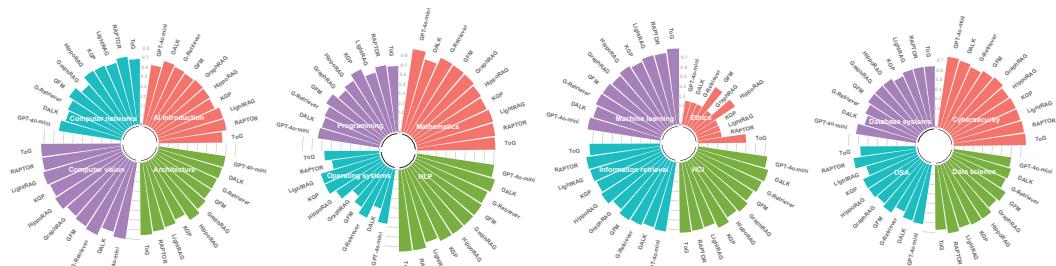


Figure 3: Comparison of Generation Accuracy by Topic.

378
379380 3.5 TOPIC-SPECIFIC GENERATION ACCURACY ANALYSIS
381

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

398
399400 3.6 COMPARISON ON DIFFERENT SCALES OF CORPORA
401

402 Given that corpus sizes can vary significantly in real-world scenarios, we conducted additional ex-
403 periments to further verify the effectiveness of different GraphRAG methods across corpora of vary-
404 ing scales. Specifically, we extracted two subsets from GraphRAG-Bench: a medium-sized subset
405 (comprising 3–4 textbooks and their corresponding questions) and a small-sized subset (consist-
406 ing of 1 textbook and its corresponding questions). Experimental results are presented in Tab. 6.
407 Our findings indicate that as corpus size decreases, performance discrepancies between different
408 methods become more pronounced. Nevertheless, more advanced GraphRAG methods (e.g., Hippo-
409 RAG, GFM-RAG, RAPTOR) consistently retain substantial performance advantages. This fur-
410 ther demonstrates that GraphRAG-Bench can effectively evaluate the performance of GraphRAG
411 methods across different scales.

412
413414 Table 6: Comparison on different scales of corpora
415

416 Category	417 Method	418 Medium-sized corpora			419 Smaller-sized corpora		
		420 Accuracy	421 R score	422 AR score	423 Accuracy	424 R score	425 AR score
426 Passage Graph	427 KGP	428 68.49	429 57.56	430 38.45	431 68.18	432 48.48	433 35.61
434 Rich KG	435 GraphRAG	436 60.92	437 54.62	438 35.08	439 65.15	440 47.73	441 29.55
442 Rich KG	443 LightRAG	444 67.65	445 57.14	446 38.87	447 68.94	448 46.97	449 32.58
450 KG	451 DALK	452 60.92	453 54.62	454 34.87	455 62.88	456 46.21	457 29.17
458 KG	459 G-Retriever	460 63.03	461 56.30	462 36.34	463 61.36	464 47.72	465 31.82
466 KG	467 ToG	468 68.07	469 56.72	470 38.45	471 67.42	472 50.00	473 34.47
475 KG	476 GFM-RAG	477 69.33	478 55.88	479 39.92	480 70.45	481 51.51	482 37.12
484 KG	485 HippoRAG	486 70.17	487 59.24	488 41.81	489 70.45	490 50.76	491 35.98
494 Tree	495 RAPTOR	496 71.43	497 58.40	498 41.18	499 71.97	500 50.76	501 38.26

502
503504 3.7 EXPERT-GUIDED POST VERIFICATION
505

506 For Open-ended and Fill-in-blank question types and rea-
507 soning, we believed that directly using string exact match
508 was unreasonable. Many correct statements were judged
509 incorrectly due to descriptions, capitalization, abbrevia-
510 tions. In such cases, we included LLM for auxiliary judg-
511 ment, which is a widely adopted paradigm in generative
512 tasks. During the actual post-processing, to verify the
513 accuracy of LLM-as-a-judge, we have conducted expert-

514 Table 7: Post verification results

515 Reliability	516 Counts	517 Percentage
518 3/3	519 470	520 94.0%
522 2/3	523 28	524 5.6%
526 1/3	527 1	528 0.2%
531 0/3	532 1	533 0.2%

432 guided post verification. We sampled 500 questions from the dataset and asked 3 human experts
 433 (researchers with a computer science doctoral degree) to score them. The experts judged whether
 434 the LLM’s judgment was reasonable based on the LLM’s generation and the groundtruth and gave
 435 1 score if yes. 3/3 points indicated reasonable and an aligned answer, 2/3 points indicated basically
 436 reasonable since the majority of experts agreed, and 1/3 or 0/3 point indicated unreasonable and
 437 needed to be reevaluated by the experts. The specific experimental results are shown in the Tab. 7,
 438 proving the reliability of LLM-as-a-judge.

439

440 4 OBSERVATION

441

442 ‘Can GraphRAG improve performance across all question types?’

443

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

449

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

454

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

460

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

468

469 ‘Can GraphRAG effectively enhance LLMs’ reasoning ability?’

470

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

484

485 **Regarding the analysis and observation we conducted:** Firstly, we obtained the phenomena through
 486 the experimental results. Secondly, the conclusions were derived by analyzing multiple instances in
 487 the specific QA process. The specific case study can be found in the Appendix.

488

489 5 CASE STUDY

490

491 As illustrated in Fig 4, we present a case study highlighting specific challenges within our dataset.
 492 Our questions span 16 core topics in undergraduate computer science; here, we focus on a sample

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

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–

540 2527, 2023.

541

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

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

549 Robert Friel, Masha Belyi, and Atindriyo Sanyal. Ragbench: Explainable benchmark for retrieval-
550 augmented generation systems, 2025. URL <https://arxiv.org/abs/2407.11005>.

551

552 Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng
553 Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey,
554 2024. URL <https://arxiv.org/abs/2312.10997>.

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

557

558 Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag:
559 Neurobiologically inspired long-term memory for large language models. In A. Globerson,
560 L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in
561 Neural Information Processing Systems*, volume 37, pp. 59532–59569. Curran Associates, Inc.,
562 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/6ddc001d07ca4f319af96a3024f6dbd1-Paper-Conference.pdf.

563

564 Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun,
565 Xavier Bresson, and Bryan Hooi. G-retriever: Retrieval-augmented generation for tex-
566 tual graph understanding and question answering. In A. Globerson, L. Mackey, D. Bel-
567 grave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural In-
568 formation Processing Systems*, volume 37, pp. 132876–132907. Curran Associates, Inc.,
569 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/efaf1c9726648c8ba363a5c927440529-Paper-Conference.pdf.

570

571 Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-
572 hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th
573 International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (On-
574 line), December 2020. International Committee on Computational Linguistics. URL <https://www.aclweb.org/anthology/2020.coling-main.580>.

575

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

580

581 Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman
582 Goyal, Heinrich Kütter, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel,
583 and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In
584 H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neu-
585 ral Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc.,
586 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

587

588 Dawei Li, Shu Yang, Zhen Tan, Jae Young Baik, Sukwon Yun, Joseph Lee, Aaron Chacko, Bojian
589 Hou, Duy Duong-Tran, Ying Ding, Huan Liu, Li Shen, and Tianlong Chen. DALK: Dynamic
590 co-augmentation of LLMs and KG to answer Alzheimer’s disease questions with scientific litera-
591 ture. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association
592 for Computational Linguistics: EMNLP 2024*, pp. 2187–2205, Miami, Florida, USA, November
593 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.119.
URL <https://aclanthology.org/2024.findings-emnlp.119/>.

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

596

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

603

604

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

611

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

614

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

618

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

623

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

627

628 Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding.
 629 Yolov10: Real-time end-to-end object detection. In A. Globerson, L. Mackey, D. Bel-
 630 grave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural In-
 631 formation Processing Systems*, volume 37, pp. 107984–108011. Curran Associates, Inc.,
 632 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/c34ddd05eb089991f06f3c5dc36836e0-Paper-Conference.pdf.

633

634 Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu,
 635 Yuan Qu, Fukai Shang, Bo Zhang, Liqun Wei, Zhihao Sui, Wei Li, Botian Shi, Yu Qiao, Dahua
 636 Lin, and Conghui He. Mineru: An open-source solution for precise document content extraction,
 637 2024b. URL <https://arxiv.org/abs/2409.18839>.

638

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

641

642 Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary,
 643 Rongze Daniel Gui, Ziran Will Jiang, Ziyu Jiang, Lingkun Kong, Brian Moran, Jiaqi Wang,
 644 Yifan Ethan Xu, An Yan, Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, Lei Chen, Nicolas
 645 Scheffer, Yue Liu, Nirav Shah, Rakesh Wanga, Anuj Kumar, Wen-tau Yih, and Xin Luna Dong.
 646 Crag - comprehensive rag benchmark. In A. Globerson, L. Mackey, D. Belgrave, A. Fan,
 647 U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*,
 648 volume 37, pp. 10470–10490. Curran Associates, Inc., 2024. doi: 10.52202/079017-0335.
 649 URL https://proceedings.neurips.cc/paper_files/paper/2024/file/

648 1435d2d0fca85a84d83ddcb754f58c29-Paper-Datasets_and_Benchmarks_
649 Track.pdf.
650

651 Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov,
652 and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question
653 answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun’ichi Tsujii (eds.), *Proceed-
654 ings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–
655 2380, Brussels, Belgium, October–November 2018. Association for Computational Linguistics.
656 doi: 10.18653/v1/D18-1259. URL <https://aclanthology.org/D18-1259/>.

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

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

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702
703 A APPENDIX704
705 A.1 THE USAGE OF LLMs706
707 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.708
709 A.2 RELATED WORK710
711 **GraphRAG.** Recent work in GraphRAG has focused on integrating structured knowledge and ad-
712 vanced retrieval strategies to overcome the limitations of vanilla RAG in handling large, noisy
713 corpora and complex reasoning. For example, RAPTOR Sarthi et al. (2024) and Microsoft’s
714 GraphRAG Edge et al. (2025) both employ hierarchical clustering, RAPTOR via recursive tree
715 construction with multi-level summarization, and GraphRAG via community detection with LLM-
716 generated synopses, to support coarse-to-fine retrieval and diverse, high-coverage responses. GFM-
717 RAG Luo et al. (2025), G-Retriever He et al. (2024), and LightRAG Guo et al. (2024) each combine
718 graph neural encoders with specialized retrieval objectives, respectively a query dependent GNN
719 trained in two stages for multi-hop generalizability, a Prize Collecting Steiner Tree formulation to
720 reduce hallucination and improve scalability, and a dual level graph augmented index for efficient,
721 incrementally updatable lookup, to enable accurate, scalable reasoning over document graphs. In-
722 spired by hippocampal memory processes, HippoRAG Gutiérrez et al. (2024) leverages Personalized
723 PageRank to achieve single-step multi-hop retrieval, delivering state-of-the-art efficiency and per-
724 formance on both path following and path finding QA tasks. DALK Li et al. (2024) and KGP Wang
725 et al. (2024c) introduce dynamic KG construction and traversal agents, using LLMs to build do-
726 main specific graphs and self aware retrieval policies, to inject structural context while reducing
727 noise. ToG Sun et al. (2024) tightly couples LLMs with KGs via beam search exploration, en-
728 abling iterative graph reasoning and on the fly correction without additional training. Collectively,
729 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.730
731 **Prior benchmarks for GraphRAG.** To date, no dataset has been specifically designed for
732 GraphRAG tasks. Widely used datasets such as Quality Pang et al. (2022), PopQA Mallen et al.
733 (2023), and HotpotQA Yang et al. (2018) are tailored for general question answering, where an-
734 swers can often be directly extracted from corpora, failing to effectively measure the core capa-
735 bilities of GraphRAG methods. Multi-hop QA datasets like MusiqueQA Trivedi et al. (2022) and
736 2WikiMultiHopQA Ho et al. (2020) contain questions artificially constructed via rules and logic,
737 rather than natural queries from real-world scenarios. Additionally, their corpora are short and often
738 derived from converting entities and descriptions of existing KGs, which deviates from practical
739 application contexts. Recently, some RAG benchmarks have emerged, such as RAGBench Friel
740 et al. (2025) and CRAG Yang et al. (2024). However, RAGBench has limitations in that it primarily
741 relies on factual queries and simple multi-hop queries, while CRAG focuses on queries in general
742 domains. Moreover, neither was specifically designed for GraphRAG, and thus neither can evaluate
743 the capabilities required at each stage of the GraphRAG pipeline. While DIGIMON Zhou et al.
744 (2025) benchmarks some methods, it neither introduces new datasets nor evaluates the reasoning
745 capabilities of GraphRAG. Critically, all aforementioned datasets neglect question type distinctions,
746 focusing primarily on simple questions and thus unable to reflect GraphRAG’s performance varia-
747 tions across different question categories. In summary, existing datasets lack long contexts and
748 raw documents, mismatching real-world scenarios, and omit gold rationale, making it impossible to
749 systematically evaluate GraphRAG’s reasoning abilities.750
751 A.3 CASE STUDY OF SECTION OBSERVATION752
753 Regarding the analysis and conclusions we conducted in the "Observation" section: Firstly, we
754 obtained the phenomena through the experimental results in the "section experiments". Secondly,
755 the conclusions were derived by analyzing multiple instances in the specific QA process. Each
analysis has supporting evidence behind it. As shown in the case study of Fig. 5 and Fig. 6.756
757 Apart from these two points, the observations in the rest of the content do not contain subjective
758 conclusions; rather, they are more about describing facts.

756
757
758
759
760

Case study for accuracy drop of MC questions.

761
762
763
764
765
766
767**Question:**

The number of nodes in a binary tree of height (depth) 7 cannot be?

Choices:

"A": "8", "B": "88", "C": "126", "D": "288"

Gold Answer:

"D"

LLM solution:

To solve this question, we first clarify the core constraint of binary tree node count: For a binary tree of height h (defined as the number of nodes on the longest root-to-leaf path), the number of nodes n satisfies $h \leq n \leq 2^h - 1$. For height 7, Minimum nodes = 7 (skewed binary tree), maximum nodes = $2^7 - 1 = 127$ (full binary tree).

Analyze the options:

A (8): $7 \leq 8 \leq 127 \rightarrow$ Valid.B (88): $7 \leq 88 \leq 127 \rightarrow$ Valid.C (126): $7 \leq 126 \leq 127 \rightarrow$ Valid.D (288): $288 > 127 \rightarrow$ Exceeds the maximum number of nodes.

Final Answer: D.

GraphRAG retrieved context:

1. Binary tree is a structure defined on a finite set of nodes that either contains no nodes (empty tree) or is composed of three disjoint sets of nodes: a root node . . .

2. In a binary tree, the maximum number of nodes at level k (root is level 1) is $2^{k-1} \dots$

3. A complete binary tree is a binary tree where all levels except possibly the last are fully filled, and the last level has nodes as left as possible. The number of nodes in a complete binary tree with height h (node-based) ranges from 2^{h-1} to $2^h - 1 \dots$

GraphRAG solution:

First, the definition of binary tree height is ambiguous. There are two standards (edge-based vs. node-based). If we adopt the edge-based definition (depth = number of edges), a height of 7 means the number of nodes on the longest path is 8. For this case: 1)Maximum nodes (full binary tree): $2^8 - 1 = 255$. 2)Minimum nodes (skewed binary tree): 8

If we adopt the node-based definition, the maximum nodes are $2^7 - 1 = 127$, and the minimum nodes are 7. Additionally, the retrieved context mentions that a complete binary tree with height h (node-based) has a minimum node count of ($2^{h-1} = 2^6 = 64$). Combining this with the ambiguous height definition, we can infer the "reasonable node range" for a binary tree of height 7 is between 64 (complete binary tree minimum) and 255 (edge-based full binary tree maximum).

Analyze the options:

A (8): Below the minimum node count of 64 (complete binary tree standard) \rightarrow Invalid.B (88): Within the range [64, 255] \rightarrow Valid.C (126): Within the range [64, 255] \rightarrow Valid.D (288): Exceeds the maximum node count of 255 \rightarrow Invalid.

Synthesizing all retrieved information: The ambiguous height definition and conflicting minimum node count standards (from different binary tree types) suggest that options A (8) and D (288) both exceed the possible range. But since the question is a single-choice question, we prioritize the complete binary tree property (more detailed in the retrieved context). Thus, the number of nodes cannot be A (8).

Final Answer: A.

804
805
806
807
808
809

Figure 5: Case study for accuracy drop of MC questions.

810
811
812
813
814

Case study for improvement in TF questions.

815
816
817
818
819
820**Question:**

"Prim's algorithm for computing the Minimum Spanning Tree (MST) only works if the weights are positive." Please judge: True (T) / False (F)

Gold Answer:

"False"

LLM solution:

Based on pre-trained knowledge, algorithms for graph optimization often have weight constraints. For example, Dijkstra's algorithm for shortest paths cannot handle negative weights. Prim's algorithm, as a classic algorithm for computing MST, is similar to Dijkstra's in its "greedy selection of minimum weight edges" logic.

It is reasonable to infer that Prim's algorithm also inherits the positive weight constraint, otherwise, negative weight edges might cause the algorithm to select inappropriate edges, leading to failure in finding the minimum spanning tree.

The statement claims that "Prim's algorithm only works if the weights are positive," which aligns with the understanding of greedy graph algorithms.

Final Judgment: True (T)

GraphRAG retrieved context:

1. Core principle of Prim's algorithm: Prim's algorithm is a greedy algorithm used to find the Minimum Spanning Tree (MST) of a connected, undirected graph. It has two sets of vertices (S) and (V-S): vertices which are already part of the MST and those which are not. The algorithm picks the edge in every iteration that connects a vertex in S with a vertex in V-S, and the weight of the edge is minimum, and then adds the edge to the MST and transfers the respective vertex in V-S to S. This is repeated until S absorbs all the vertices.

2. Weight constraint of Prim's algorithm: The Prim's algorithm does not expect the weights of the edges to be positive. The only conditions are: (1) the graph is connected; (2) the weights of the edges are not infinite (they could be positive, zero, or negative). Such negative weights do not break the logic of the algorithm since the aim of the algorithm is to minimize the total weight of the spanning tree, such a negative weight edge can actually be given priority to be added to the tree should it lower the total weight.

3. Distinction from Dijkstra's algorithm: Dijkstra's algorithm (for single-source shortest paths) requires non-negative edge weights. In the case of negative weight edges, the algorithm can erroneously think that a node is already processed and never causes the path to be updated to a shorter path. Nevertheless, the Prim's algorithm is concerned with the construction of the vertices to the MST set (not with cumulative path lengths), therefore, there are no logical failures with negative weights.

GraphRAG solution:

Based on the retrieved factual evidence, we verify the statement step by step:

First, confirm the core constraints of Prim's algorithm: The retrieved context clearly states that Prim's algorithm only requires the graph to be connected and edge weights to be finite. There is no requirement for weights to be positive. The statement's claim that "only works if the weights are positive" is inconsistent with the algorithm's actual constraints.

Second, distinguish from easily confused algorithms: The statement may confuse Prim's algorithm with Dijkstra's algorithm (context 3). Dijkstra's algorithm requires non-negative weights, but this constraint does not apply to Prim's algorithm. The algorithm's logic (selecting edges between two vertex sets) is not affected by negative weights.

Synthesizing the verified facts: The statement incorrectly imposes a positive weight constraint on Prim's algorithm, which has no such requirement.

Final Judgment: False (F)

858
859
860
861
862
863

Figure 6: Case study for improvement in TF questions.

864 A.4 REASONS FOR CHOOSING CS/AI DOMAIN
865866 We highly value CS and AI domain for three reasons:
867

- 868 • First, CS and AI naturally contain a variety of types of challenging tasks compared with all the
869 existing 'single-type' datasets, where the questions are extremely hard which require college-level
870 command of knowledge, compared with existing commonsense dataset. In CS and AI domain,
871 we have conceptual understanding tasks (algorithms, networks, databases, paradigms, machine
872 learning, NLP, CV), Math problems (Discrete mathematics, linear algebra, matrix computation,
873 calculus), Code generation problems (programming, optimization, code completion, error detec-
874 tion), etc.
- 875 • Second, the performance on LLMs on CS and AI is still very unsatisfactory or even bad. It is
876 valuable to benchmark and research.
- 877 • Third, it remains a blank in the benchmark of LLMs considering CS and AI. This also brings
878 two opportunities. (i) LLMs will have few chances to be fine-tuned based on our questions. (ii)
879 Following research on GraphRAG or LLMs themselves could benefit a lot from our benchmark.

880 Basic common knowledge in fields such as medicine and law (e.g., common legal provisions, basic
881 disease knowledge) is extensively covered in LLM pre-training corpora. However, university-level
882 computer knowledge (e.g., advanced algorithm design, underlying system principles, technical de-
883 tails in specialized sub-fields), due to its high professionalism and rapid updates, has extremely low
884 and fragmented coverage in pre-training. More importantly, such knowledge has extremely high
885 requirements for "the rigor of logical reasoning." Our experimental data shows that even if GPT-
886 4o-min provides correct answers in some computer tasks, its reasoning scores are still significantly
887 low. In-depth analysis reveals that these correct answers mostly stem from the model's "probabilis-
888 tic guessing of similar problems" rather than rigorous reasoning based on structured knowledge.
889 The core significance of GraphRAG lies in forcing the reasoning process to be bound to structured
890 knowledge, ensuring that answers originate from traceable and verifiable logical chains. This value
891 is particularly prominent in the computer field, where "pre-training coverage is insufficient and rea-
892 soning requirements are high," and it better reflects the uniqueness of GraphRAG in solving the
893 problem of "vague guessing."

894 Our corpus is derived from over 100 authoritative computer textbooks (with 20 selected in the end),
895 which naturally have a hierarchical structure of "chapters → sections → knowledge points → knowl-
896 edge content," and each knowledge content includes a complete logical chain of various knowledge
897 points. This is fully compatible with GraphRAG's technical path of "knowledge graph construc-
898 tion → retrieval and association → structured reasoning," making it the best carrier for evaluating
899 GraphRAG's "knowledge-driven reasoning" capability.

900 A.5 ADDITIONAL EXPERIMENTS
901902 A.5.1 COMPARISON ACROSS DIFFERENT LLMs
903

904 To further verify the effectiveness of GraphRAG-Bench and the robustness of different GraphRAG
905 methods for different LLMs, we conducted comprehensive experiments on two popular LLMs
906 (Deepseek-V3, Qwen3-32B). The specific experimental results are shown in Tab. 8, Tab. 9, Tab. 10
907 and Tab. 11. Based on the experimental results using two different LLMs as the base LLM, it can be
908 seen that all the main conclusions drawn in main text have been confirmed again.

909 A.5.2 TOPIC-SPECIFIC REASONING ANALYSIS
910

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

915 **Operating System Domain.** The LLM exhibits suboptimal performance in this domain. While
916 GraphRAG provides marginal improvements in reasoning capabilities, overall scores remain low.
917 This is primarily attributed to the highly specialized, systematic, and logically complex nature of
918 operating system knowledge, which involves multi-layered principles such as process scheduling,

Table 8: Comparison of generation process (Deepseek-V3 baseline).

Category	Method	Accuracy					
		Fill-in-blank	Multi-choice	Multi-select	True-or-false	Open-ended	Average
Base LLM	Deepseek-V3	74.54	81.39	76.94	76.21	52.01	70.92
Passage Graph	KGP	75.71	79.36	78.15	78.70	53.23	72.03↑
Rich KG	LightRAG	75.50	79.13	77.93	78.49	53.08	71.83↑
Rich KG	GraphRAG	75.50	81.57	76.12	79.72	52.68	73.12↑
KG	DALK	73.79	77.26	76.16	78.74	51.88	70.20↓
KG	G-Retriever	74.24	77.76	76.63	79.70	52.19	70.63↓
KG	ToG	74.07	79.31	76.60	77.26	51.55	72.16↑
KG	GFM-RAG	75.31	79.67	77.69	81.39	52.44	73.30↑
KG	HippoRAG	75.50	80.64	77.89	80.09	54.31	73.49↑
Tree	RAPTOR	77.77	80.76	78.48	80.98	53.14	73.99↑

Table 9: Comparison of reasoning capability (Deepseek-V3 baseline).

Category	Method	Reasoning											
		FB		MC		MS		TF		OE		Average	
		R	AR	R	AR	R	AR	R	AR	R	AR	R	AR
Base LLM	Deepseek-V3	65.23	54.37	55.72	51.89	55.41	40.27	59.35	54.62	50.18	10.33	56.40	40.31
Passage Graph	KGP	64.78	50.42	56.53	52.76	58.29	45.31	64.17	61.48	52.69	9.73	59.03	43.44
Rich KG	GraphRAG	71.53	55.27	56.19	52.46	57.64	45.72	63.81	60.19	53.26	10.53	60.32	44.80
Rich KG	LightRAG	66.28	48.17	57.23	52.38	61.76	49.19	66.83	63.42	53.29	10.26	60.99	44.92
KG	DALK	71.42	56.19	54.76	51.38	59.24	47.53	62.81	59.26	54.39	10.15	58.80	43.20
KG	G-Retriever	70.27	55.19	57.46	53.29	60.73	48.16	64.38	60.17	53.42	10.28	59.90	44.30
KG	ToG	70.23	53.19	56.07	51.64	57.18	45.62	65.81	62.27	54.53	12.26	61.02	44.87
KG	GFM-RAG	70.26	55.03	56.28	52.19	58.23	45.72	66.54	63.76	53.78	10.73	61.19	45.80
KG	HippoRAG	66.78	50.64	56.73	52.38	60.04	47.62	67.19	63.64	55.23	12.57	61.50	45.72
Tree	RAPTOR	71.73	58.19	56.54	52.17	60.53	49.16	66.54	62.87	53.96	13.72	61.55	46.20

memory management, and file systems, requiring precise grasp of conceptual definitions, algorithmic 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.

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

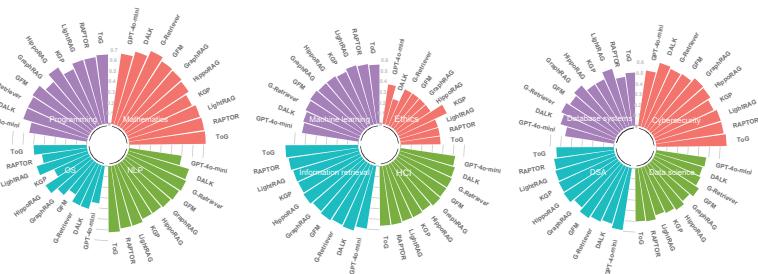


Figure 7: Comparison of R score by Topic.

Table 10: Comparison of generation process (Qwen3-32B baseline).

Category	Method	Accuracy					
		Fill-in-blank	Multi-choice	Multi-select	True-or-false	Open-ended	Average
Base LLM	Qwen3-32B	73.68	78.92	75.43	73.19	49.98	70.04
Passage Graph	KGP	73.28	77.54	75.67	77.36	51.75	71.11↑
Rich KG	LightRAG	72.59	76.94	75.21	76.81	51.28	70.57↑
Rich KG	GraphRAG	74.27	78.51	76.38	78.39	52.64	72.03↑
KG	DALK	71.59	76.18	74.59	75.98	50.71	69.80↓
KG	G-Retriever	70.79	75.37	73.98	75.21	50.15	69.10↓
KG	ToG	72.87	77.23	75.43	77.09	51.43	70.80↑
KG	GFM-RAG	75.61	79.37	76.98	79.49	53.29	72.95↑
KG	HippoRAG	74.85	78.91	76.68	78.79	52.97	72.44↑
Tree	RAPTOR	75.47	79.42	77.03	79.58	53.45	72.99↑

Table 11: Comparison of reasoning capability (Qwen3-32B baseline).

Category	Method	Reasoning											
		FB		MC		MS		TF		OE		Average	
		R	AR	R	AR	R	AR	R	AR	R	AR	R	AR
Base LLM	Qwen3-32B	64.35	53.17	55.92	50.41	54.18	39.85	58.73	53.26	49.83	9.34	56.01	40.03
Passage Graph	KGP	64.07	49.83	56.12	52.09	58.94	44.51	64.68	60.17	52.92	9.90	59.33	43.30
Rich KG	LightRAG	66.51	48.09	57.84	52.16	61.07	49.85	66.19	63.52	53.64	10.13	60.05	44.19
Rich KG	GraphRAG	71.09	55.68	56.57	52.61	57.53	45.17	63.94	60.09	54.08	11.07	60.11	44.72
KG	DALK	70.19	55.43	54.67	50.18	59.03	46.82	62.91	58.04	54.70	9.33	59.30	42.18
KG	G-Retriever	70.85	55.04	57.03	53.82	60.17	47.79	64.02	60.45	53.93	10.61	59.42	42.15
KG	ToG	70.09	53.84	56.72	51.03	57.94	45.01	65.13	62.67	54.40	12.90	60.68	44.33
KG	GFM-RAG	70.15	54.67	56.98	52.14	58.67	45.09	66.53	63.58	54.25	11.02	60.78	46.00
KG	HippoRAG	66.17	50.09	57.24	52.68	59.68	47.15	67.63	63.61	55.40	12.12	61.04	45.89
Tree	RAPTOR	71.76	57.15	56.09	52.54	60.58	49.07	66.12	62.69	54.47	13.11	61.22	46.33

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.

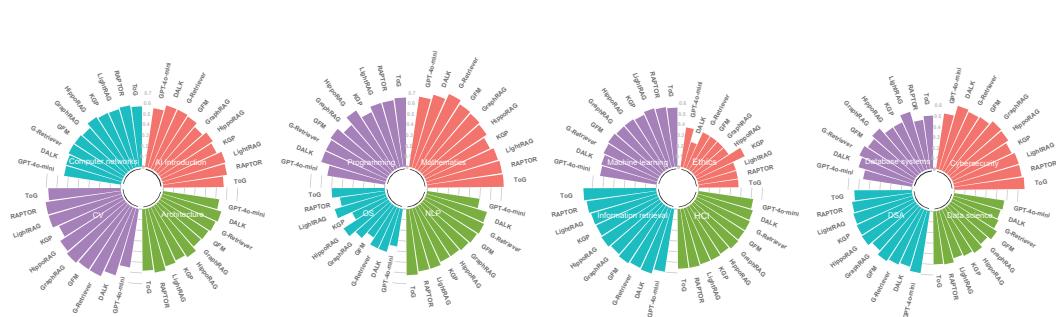


Figure 8: Comparison of AR score by Topic.

1026 A.5.3 EVALUATION OF GENERATION ACCURACY (WITH THE DEVIATIONS)
10271028 For all the generation results in the main text, they are the average values obtained after conducting
1029 five tests for each algorithm. Due to space limitations, we are unable to include all the details in the
1030 main text. Therefore, we are providing here the results with deviations, as shown in the Tab. 12:
10311032 Table 12: Comparison of generation process with the deviations.
1033

Method	Accuracy					
	Fill-in-blank	Multi-choice	Multi-select	True-or-false	Open-ended	Average
DALK	70.00 \pm 0.32	78.34 \pm 0.85	71.62 \pm 0.58	77.22 \pm 0.92	51.49 \pm 0.25	69.30 \pm 0.58 \downarrow
G-Retriever	70.95 \pm 0.41	77.42 \pm 0.79	71.62 \pm 0.63	78.80 \pm 0.95	52.04 \pm 0.28	69.84 \pm 0.61 \downarrow
LightRAG	65.24 \pm 0.35	78.80 \pm 0.88	73.42 \pm 0.67	82.59 \pm 0.97	53.16 \pm 0.31	71.22 \pm 0.64 \uparrow
ToG	70.48 \pm 0.43	78.80 \pm 0.82	78.38\pm0.71	79.75 \pm 0.93	54.28 \pm 0.33	71.71 \pm 0.65 \uparrow
KGP	74.29 \pm 0.47	79.26 \pm 0.89	74.77 \pm 0.74	82.28 \pm 0.96	51.49 \pm 0.26	71.86 \pm 0.66 \uparrow
GFM-RAG	72.38 \pm 0.45	80.65 \pm 0.91	72.07 \pm 0.69	82.59 \pm 0.98	52.79 \pm 0.29	72.10 \pm 0.66 \uparrow
GraphRAG	75.24 \pm 0.50	81.57\pm0.94	77.48 \pm 0.76	80.70 \pm 0.90	52.42 \pm 0.27	72.50 \pm 0.68 \uparrow
HippoRAG	70.48 \pm 0.44	80.18 \pm 0.87	74.32 \pm 0.72	81.65 \pm 0.95	56.13\pm0.35	72.64 \pm 0.67 \uparrow
RAPTOR	76.67\pm0.53	80.65 \pm 0.93	77.48 \pm 0.78	82.28 \pm 0.97	54.83 \pm 0.32	73.58\pm0.71\uparrow

1034 A.5.4 EVALUATION OF REASONING CAPABILITY (WITH THE DEVIATIONS)
10351036 For all the reasoning results in the main text, they are the average values obtained after conducting
1037 five tests for each algorithm. Due to space limitations, we are unable to include all the details in the
1038 main text. Therefore, we are providing here the results with deviations, as shown in the Tab. 13:
10391040 Table 13: Comparison of reasoning capability with the deviations.
1041

Method	Reasoning											
	FB		MC		MS		TF		OE		Average	
	R	AR										
DALK	70.95 \pm 0.92	55.24 \pm 0.68	54.15 \pm 0.61	50.35 \pm 0.53	59.01 \pm 0.85	46.40 \pm 0.61	62.18 \pm 0.74	58.23 \pm 0.70	54.09 \pm 0.63	9.67 \pm 0.28	58.89 \pm 0.76	42.12 \pm 0.57
KGP	64.29 \pm 0.79	49.29 \pm 0.55	56.45 \pm 0.73	52.07 \pm 0.63	58.11 \pm 0.83	44.37 \pm 0.55	64.08 \pm 0.76	60.68 \pm 0.77	52.42 \pm 0.60	8.92 \pm 0.26	58.74 \pm 0.71	42.22 \pm 0.58
GraphRAG	71.43\pm0.95	55.24 \pm 0.69	56.22 \pm 0.70	52.42 \pm 0.65	57.66 \pm 0.80	45.72 \pm 0.58	63.61 \pm 0.74	60.13 \pm 0.73	53.16 \pm 0.62	10.50 \pm 0.33	59.43 \pm 0.77	43.30 \pm 0.62
G-Retriever	70.00 \pm 0.88	55.00 \pm 0.66	57.60\pm0.83	53.46\pm0.71	60.81 \pm 0.90	48.20 \pm 0.67	64.24 \pm 0.78	60.21 \pm 0.75	53.35 \pm 0.64	10.04 \pm 0.31	60.17 \pm 0.81	43.66 \pm 0.65
LightRAG	66.19 \pm 0.82	47.86 \pm 0.52	57.14 \pm 0.72	52.30 \pm 0.64	61.71\pm0.93	49.10\pm0.69	66.61 \pm 0.85	63.45 \pm 0.88	53.16 \pm 0.63	10.13 \pm 0.32	60.46 \pm 0.84	43.81 \pm 0.66
ToG	70.00 \pm 0.87	53.10 \pm 0.63	56.00 \pm 0.68	51.73 \pm 0.60	57.21 \pm 0.79	45.72 \pm 0.57	65.66 \pm 0.81	62.26 \pm 0.83	54.46 \pm 0.67	12.08 \pm 0.41	60.17 \pm 0.80	44.01 \pm 0.68
GFM-RAG	70.00 \pm 0.86	54.76 \pm 0.67	56.22 \pm 0.69	52.07 \pm 0.62	58.11 \pm 0.82	45.50 \pm 0.56	66.46 \pm 0.84	63.69\pm0.89	53.72 \pm 0.65	10.69 \pm 0.34	60.36 \pm 0.82	44.30 \pm 0.70
HippoRAG	66.67 \pm 0.83	50.48 \pm 0.58	56.68 \pm 0.71	52.30 \pm 0.63	59.91 \pm 0.87	47.52 \pm 0.63	67.25\pm0.87	63.61 \pm 0.88	55.02\pm0.69	12.36 \pm 0.42	60.90\pm0.86	44.55 \pm 0.71
RAPTOR	71.43\pm0.94	57.86\pm0.75	56.45 \pm 0.72	52.07 \pm 0.62	60.36 \pm 0.89	49.10\pm0.68	66.30 \pm 0.83	62.90 \pm 0.85	53.90 \pm 0.66	13.57\pm0.47	60.81 \pm 0.85	45.53\pm0.76

1063 A.5.5 COMPARISONS WITH EXISTING DATASETS.
10641065 Currently, most GraphRAG methods follow the experimental setting of HippoRAG, which is the
1066 most common and widely used setting in this field at present. It includes three public datasets:
1067 HotpotQA, 2WikiMultiHopQA, and MuSiQue which are all originally designed for multi-hop QA,
1068 not GraphRAG. They can hardly necessitate the use of graphs and showcase the use of graphs in
1069 complex QA scenarios.1070 In each of these data sets, 1000 questions were sampled. Therefore, following this standard and dis-
1071 tribution, we selected 1018 of the most valuable questions as the final version. Meanwhile, graprag-
1072 bench encompasses a wider range of question types (examining the model’s robustness to various
1073 question formats), a larger corpus (making it more difficult to retrieve useful information), and cov-
1074 ers multiple topics (requiring the model to understand different knowledge domains). We believe
1075 that the coverage scope is much broader from multiple perspectives than that of the current dataset.
1076 The specific comparison results are shown in the Tab. 14.1077 A.5.6 COMPUTE RESOURCES
1078

1079 All code is done in Python, and experiments are conducted on H100*2 GPUs.

1080	Dataset	Question count	Corpus size	Question type
1081	HotpotQA	1000	6.16MB	1
1082	2WikiMultiHopQA	1000	2.96MB	1
1083	MuSiQue	1000	5.95MB	1
1084	Ours	1018	41.30MB	5

Table 14: Comparisons with existing datasets.

1090 A.6 DETAILS OF QUESTION DESIGN AND SELECTION

1092 We adopted a rigorous process to design questions covering a wide range of topics in computer science
 1093 and artificial intelligence. Experts manually constructed the question set based on a structured
 1094 ontology, which divided knowledge into three levels: 16 primary themes, 40 secondary themes, and
 1095 26 tertiary themes. Each level represents a progressive progression from a broad disciplinary field
 1096 to more specific sub-themes.

1097 For each concept among these n themes, experts were required to prepare at least two questions for
 1098 each type of question, totaling $2 \times 5 \times n$ questions, to ensure balanced content and comprehensive
 1099 coverage. To ensure high quality, experts referred to authoritative sources such as textbooks, course
 1100 materials, and widely recognized course outlines. Subsequently, each question was reviewed to
 1101 verify its relevance, clarity, and consistency with the corresponding concept. This process ensures
 1102 that the dataset not only conforms to real-world educational standards but also covers the full breadth
 1103 and depth of the field.

1104 Three experts (senior researchers in the RAG field with a doctoral degree) will rate questions from
 1105 three dimensions: "knowledge coverage, reasoning chain completeness, and compatibility with
 1106 GraphRAG" (1-5 points), and finally select the questions with an average score of ≥ 4 points.

1108 A.7 DETAILS OF CORPUS COLLECTION AND PROCESSING

1110 Extracting accurate content from the 20 PDF-format core textbooks presents significant challenges.
 1111 We implement a multi-stage pipeline comprising preprocessing, content parsing, post-processing,
 1112 and hierarchy construction.

1113 **Textbook Preprocessing.** 1) PDF Classification: To distinguish text-based pages from scanned
 1114 (image-based) pages, we analyze each page's text density and image area proportion. Text-based
 1115 pages are processed by extracting text directly using PyMuPDF, while scanned pages require optical
 1116 character recognition (OCR) to extract their textual content. 2) Metadata Extraction: We extract
 1117 metadata for each textbook, including its outline, total page count, and the page ranges for each
 1118 chapter or section. This metadata supports the later construction of the document's logical structure.

1119 **Content Parsing.** After preprocessing, we analyze each page's layout to extract textual and non-
 1120 textual elements. 1) Layout Analysis: We apply LayoutLMv3 Huang et al. (2022) for multimodal
 1121 document layout analysis. LayoutLMv3 is pre-trained with masked language modeling, masked
 1122 image modeling, and cross-modal alignment, enabling it to learn rich representations of document
 1123 pages. The model classifies page regions into semantic categories such as titles, paragraphs, figures,
 1124 tables, or decorative/irrelevant elements. This segmentation yields coherent content blocks on each
 1125 page. 2) Formula Recognition: Mathematical formulas embedded in text are often misrecognized by
 1126 OCR. To prevent this, we first detect inline formulas using a pre-trained YOLO-based model Wang
 1127 et al. (2024a) from PDF-Extract-Kit. This model identifies the bounding boxes of formula regions
 1128 so that formula images can be extracted separately, ensuring that OCR does not garble the formula
 1129 content. 3) OCR: In scanned PDFs, OCR is applied to recognize text regions. We use PaddleOCR to
 1130 transcribe text from the regions labeled as titles and body paragraphs via layout analysis. This step
 1131 produces the page's textual content in the correct reading order, while preserving non-text elements
 1132 as separate objects.

1133 **Post-Processing.** After parsing, the extracted elements (text blocks, formula, figures, tables, etc.)
 1134 may be disordered due to overlapping bounding boxes or fragmented text lines. We resolve these

1134 issues by reordering and merging page regions according to human reading order. Concretely, we
 1135 use MinerU Wang et al. (2024b) for post-processing, which partitions each page into logical reading
 1136 regions and sequences them so that the final text flow matches the natural reading sequence.
 1137

1138 **Hierarchy Construction.** Finally, we organize the extracted content into a hierarchical textbook-
 1139 tree structure. We map the textbook metadata (e.g., chapter titles, section divisions, and page ranges)
 1140 to a four-level hierarchy: Book Title → Chapter → Section (Subchapter) → Knowledge Content
 1141 Unit. Each node in this hierarchy is annotated with its contextual metadata and its structural role.
 1142 This textbook-tree provides an intuitive, pedagogical navigation framework aligned with the text-
 1143 book’s organization. The resulting corpus, with its accurate content extraction, structural annotation,
 1144 and hierarchical organization, forms a robust basis for evaluating GraphRAG’s ability to leverage
 1145 organized textbook knowledge for context-rich reasoning and retrieval-augmented generation.
 1146

1146 A.8 THE DETAILS OF METRICS

1147 A.8.1 DETAILS OF AR SCORE.

1149 The AR score is computed based on the combination of answer correctness (generation score) and
 1150 rationale correctness (reasoning score), with the following evaluation rules:
 1151

- 1152 • When both the answer and rationale are fully correct (generation score = 1 and reasoning
 1153 score = 1), the AR score is 1.0.
- 1154 • If the answer is correct but the rationale is partially correct (generation score = 1 and rea-
 1155 soning score = 0.5), the AR score is 0.5.
- 1156 • When the answer is correct but the rationale is incorrect (generation score = 1 and reasoning
 1157 score = 0), the AR score is 0.0.
- 1158 • For incorrect answers with a fully correct rationale (generation score = 0 and reasoning
 1159 score = 1), the AR score is 0.5.
- 1160 • If both the answer is incorrect and the rationale is partially correct (generation score = 0
 1161 and reasoning score = 0.5), the AR score is 0.25.
- 1162 • In all other cases (e.g., incorrect answer with incorrect or missing rationale), the AR score
 1163 is 0.0.

1165 This scoring scheme systematically captures the alignment between answers and their supporting
 1166 reasoning, emphasizing the importance of both correctness and logical consistency in evaluating
 1167 model performance.

1168 A.8.2 PROMPT OF OE AND FB QUESTIONS.

1169 Fig. 9 is the prompt used to generate the LLM-judge score for OE and FB questions.
 1170

1172 A.8.3 PROMPT OF REASONING GRADING.

1174 Fig. 10 is the prompt used to evaluate the reasoning score R.
 1175

1176 A.9 LIMITATIONS OF EXISTING GRAPHRAG DATASETS

1177 Through a systematic review of benchmark datasets used by contemporary Graph-RAG methods,
 1178 we have identified four critical limitations that undermine both task suitability and the validity of
 1179 evaluation results:
 1180

1181 **Superficial retrieval tasks.** Most datasets pose questions that can be answered by straightforward
 1182 text retrieval, without requiring deep integration of graph structure or sophisticated semantic rea-
 1183 soning. Consequently, models may achieve high scores by exploiting shallow keyword matching,
 1184 offering no insight into their true capabilities in relational reasoning or entity-association modeling.

1185 **Synthetic and unrepresentative queries.** Questions are typically generated via hand-crafted rules,
 1186 yielding simplified language that lacks the domain-specific terminology, ambiguous intent, and syn-
 1187 tactic variety found in real user queries. This synthetic distribution diverges sharply from natural
 1188 problem settings, limiting the ecological validity of any conclusions about model generalization.
 1189

1188 Prompt of generation grading for OE and FB questions.
1189
1190 **Instructions:**
1191 You are a strict evaluator. Compare the following two answers for correctness and completeness:
1192
1193 **Predicted Answer:**
1194 <pred_answer>
1195
1196 **Gold Answer:**
1197 <gold_answer>
1198
1199 **Important Guidelines:**
1200 Please evaluate the predicted answer in comparison to the gold answer. Respond with a score between
1201 0 and 1:
1202 - 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.
1203

Figure 9: Prompt of generation grading for OE and FB questions.

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 10: Prompt of rationale grading.

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.

1242 A.10 LIMITATIONS OF THIS PAPER
12431244 Despite the valuable contributions of this study, we acknowledge its limitations: (1) Our dataset
1245 currently only contains English content; more detailed research should be done in the future for
1246 different languages. (2) Other modal data such as images are not included in the current data set,
1247 and richer multimodal datasets can be considered in the future.

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

1259

1260

1261

1262

1263

1264

1265

1266

1267

1268

1269

1270

1271

1272

1273

1274

1275

1276

1277

1278

1279

1280

1281

1282

1283

1284

1285

1286

1287

1288

1289

1290

1291

1292

1293

1294

1295