Iterative Knowledge Graph Refinement and Integration for Medical Question Answering

Haochen Zou, Yongli Wang
Nanjing University of Science and Technology
{haochenzou,yongliwang}@njust.edu.cn

Abstract

Large Language Models (LLMs) are challenged by generating hallucinations and factually incorrect responses, particularly in complex and specialized medical question answering (QA). Integrating knowledge graphs (KGs) through retrieval-augmented generation (RAG) methods has emerged as a promising direction. However, existing graph-based RAG methods heuristically retrieve and refine question-relevant subgraphs, potentially introducing redundant and noisy factual information that is difficult for LLMs to process, ultimately limiting reasoning capability. To incorporate a concise yet informative evidence subgraph, we propose an iterative medical QA framework. It optimizes graph-based RAG methods by selectively retrieving focused knowledge from KGs to construct a precise evidence subgraph and progressively pruning it utilizing structured feature representations. The targeted KG integration maintains coherent and reliable inference. Experiments on three medical QA benchmark datasets demonstrate that the framework achieves state-of-the-art performance against representative baseline competitors, highlighting the importance of efficient KG integration.

1 Introduction

Large language models (LLMs) have demonstrated impressive performance in natural language processing (NLP) and have attracted significant research attention (Wan et al., 2025). However, in professional tasks from knowledge-intensive domains, such as medical question answering (QA), LLMs inadvertently suffer from hallucinations and generate unreliable responses that appear plausible but are factually incorrect, thus undermining trustworthiness (Sui et al., 2025b). Although developing domain-specific LLMs on specialized corpora can integrate parameterized prior knowledge, it is computationally expensive, difficult to update, limited in transferability, prone to catastrophic forgetting, and lacks interpretability (He et al., 2025).

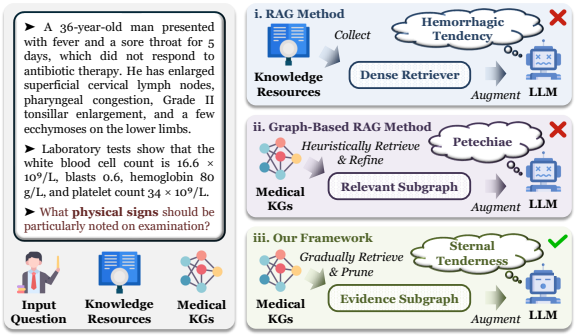

Figure 1: The comparison analysis of three categories of methods: (i) **RAG Method**, (ii) **Graph-Based RAG Method**, and (iii) **Our Framework**, which gradually retrieves and refines a concise and informative evidence subgraph sufficient for accurate answer prediction.

Retrieval-augmented generation (RAG) methods have been introduced to improve LLM reasoning by incorporating external trustworthy knowledge resources (Wan et al., 2025). To further enhance performance, structured knowledge from knowledge graphs (KGs) has been integrated through graph-based RAG methods (Rezaei et al., 2025). Different from conventional RAG methods that leverage sparse text content, graph-based RAG methods represent factual information from KGs in an interconnected structure, supporting explicit and traceable multi-step inference (Sui et al., 2025b). Nevertheless, as illustrated in Figure 1, both RAG methods and graph-based RAG methods struggle to handle complex medical QA, producing inaccurate responses and exhibiting limited effectiveness.

Existing graph-based RAG methods face notable challenges in medical QA, which can be categorized into three primary problems: (i) Graph-based RAG methods generally extract factual information from KGs utilizing heuristic retrieval strategies. Although they conditionally rank retrieved knowledge

Findings of the Association for Computational Linguistics: ACL 2026, pages 8358–8374

and selectively explore relevant n-hop neighbors (Wang and Yu, 2025), medical questions frequently contain extensive interconnected details that can overwhelm graph-based RAG methods in the beginning (Sohn et al., 2025). (ii) Graph-based RAG methods may retrieve redundant, overlapping, or substitutable factual information from KGs, leading to knowledge overload (He et al., 2025). It necessitates repeated refinement to construct a question-relevant subgraph, inadvertently introducing irrelevant or conflicting information and complicating effective KG integration for multi-step inference. (iii) Graph-based RAG methods lack a coherent structural overview of the retrieved subgraph. Subgraph descriptions generated from raw text content can be disorganized and semantically fragmented, which are difficult for LLMs to interpret (Wan et al., 2025). Although multiple graph-based RAG methods employ graph neural networks (GNNs) to encode structured feature representations (Sui et al., 2025a; Sohn et al., 2025), aligning the embeddings with the requirements of LLMs for effective implementation remains complicated.

We aim to discover a concise yet informative evidence subgraph for effective KG integration and improve LLM performance in complex medical QA. Inspired by the hierarchical organization of knowledge in encyclopedias, which incrementally incorporates factual information from primary to secondary knowledge resources, we propose an iterative medical QA framework. It begins by leveraging the top-ranked retrieved factual information and explores question-relevant neighbors hop by hop in KGs to construct a focused evidence subgraph. Subsequently, the framework progressively prunes the evidence subgraph utilizing its structured feature representations, avoiding the requirement to model entire KGs. The evidence subgraph is iteratively refined and encoded through a multi-level list, organizing structured information while preserving both the breadth information obtained from neighbor retrieval and the depth information captured through graph traversal during evidence subgraph construction, which facilitates LLM interpretation. The advantages can be summarized:

- We propose an innovative medical QA framework that iteratively retrieves and prunes to construct a compact evidence subgraph for KG integration.
- The framework selectively explores question-relevant neighbors in KGs and leverages structured feature representations to progressively prune the evidence subgraph, mitigating redundancy and sensitivity in KG integration.
- The framework incorporates the structured information from the evidence subgraph while preserving both the breadth and depth information during KG integration to enhance LLM performance in medical QA.
- To the best of our knowledge, this research represents the initial effort to concentrate on potential knowledge redundancy in graph-based RAG methods for medical QA. Experimental results on three medical QA benchmark datasets demonstrate its effectiveness.

2 Related Work

Medical QA aims to answer medical questions by leveraging domain-specific knowledge (Rezaei et al., 2025). We summarize existing approaches into four categories: (i) **General Domain LLMs**, (ii) **Medical Domain LLMs**, (iii) **RAG Methods**, and (iv) **Graph-Based RAG Methods**.

General Domain LLMs. Constructed on extensive corpora across diverse topics and supported by advanced architectures, general domain LLMs demonstrate impressive performance in multiple NLP tasks, including medical QA. Representative LLMs include the Generative Pre-trained Transformer (GPT) family models (Achiam et al., 2023), the Large Language Model Meta AI (LLaMA) family models (Grattafiori et al., 2024), and Mistral (Jiang et al., 2023). However, general domain LLMs may struggle in domain-specific and knowledge-intensive scenarios. They can produce hallucinations and factually incorrect outputs (Sui et al., 2025b). The unreliability significantly reduces the effectiveness of general domain LLMs in complex and specialized multi-step inference.

Medical Domain LLMs. Medical domain LLMs are comprehensively trained on specialized corpora to capture domain-specific terminology and knowledge. Representative medical domain LLMs include Meditron from LLaMA (Chen et al., 2023), BioMistral from Mistral (Labrak et al., 2024), and Meerkat-7B from Mistral (Kim et al., 2024). Medical domain LLMs demonstrate remarkable improvements in domain-specific NLP tasks (Rezaei et al., 2025). However, for complex medical QA, the internal static and parameterized prior knowledge of medical domain LLMs may struggle to support accurate multi-step inference.

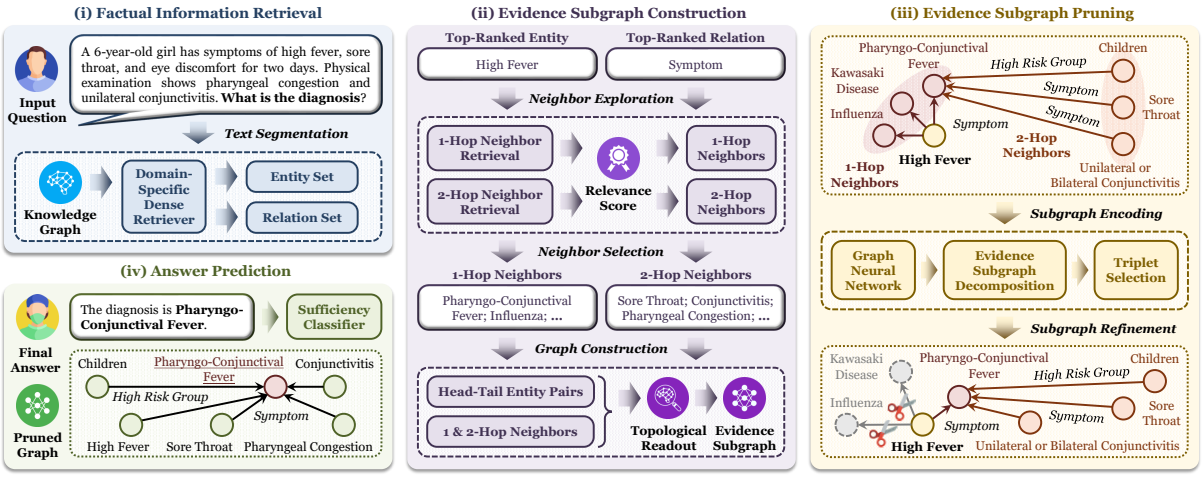

Figure 2: Overview of the framework for medical QA, illustrating the inference workflow for answer prediction through factual information retrieval, evidence subgraph construction, and evidence subgraph pruning.

RAG Methods. Enhance LLMs by integrating a dense retriever that supplies relevant and up-to-date knowledge during reasoning, RAG methods supporting LLMs mitigate limitations of internal static prior knowledge. Representative RAG methods for medical QA include MedRAG (Xiong et al., 2024), RGAR (Liang et al., 2025), and RAG^2 (Sohn et al., 2025). Prompting strategies such as chain of thought (CoT) can further improve performance (Rezaei et al., 2025). However, when dealing with complicated medical questions, RAG methods may retrieve redundant and sparse context, which can hinder accurate multi-step inference.

Graph-Based RAG Methods. By grounding multi-step inference of LLMs in structured knowledge, graph-based RAG methods progressively enhance performance by reducing hallucinations and supporting verifiable reasoning. Representative graph-based RAG methods include GraphRAG (Edge et al., 2024), ByoKG-RAG (Mavromatis et al., 2025), and AMG-RAG (Rezaei et al., 2025). However, existing graph-based RAG methods generally begin by extracting a broad set of factual information from the question. They heuristically retrieve and refine a relevant subgraph (Wang and Yu, 2025), which struggles to produce a concise evidence subgraph. They may introduce unnecessary and noisy factual information, ultimately undermining multi-step inference for medical QA.

3 Methodology

The framework, as presented in Figure 2 and summarized in Algorithm 1 from Appendix A.1, consists of four modules: (i) **Factual Information Retrieval** (Section 3.2), (ii) **Evidence Subgraph Construction** (Section 3.3), (iii) **Evidence Subgraph Pruning** (Section 3.4), and (iv) **Answer Prediction** (Section 3.5). Given a medical question, the framework progressively retrieves and ranks factual information from the KG to construct an evidence subgraph. The evidence subgraph is gradually refined through graph-based pruning. The framework iteratively continues the procedures until the evidence subgraph is estimated as sufficient, and it is incorporated into the answer prediction.

3.1 Formalization

Given an input medical question $\mathcal{Q}$ and a KG composed of multiple triplets, the KG can be formally defined as $\mathcal{G} = \{(h, r, t) | h, t \in \mathcal{E}, r \in \mathcal{R}\}$, where $\mathcal{E}$ and $\mathcal{R}$ denote the sets of entities and relations, respectively (Wang and Yu, 2025). Each triplet $\tau = (h, r, t)$ represents a relation r connecting a head entity h to a tail entity t. Both entities and relations are expressed in natural language (Lin et al., 2025). The objective is to design a framework $\mathcal{F}(\cdot)$ that predicts an answer $\mathcal{A}$ conditioned on the question $\mathcal{Q}$ and the KG $\mathcal{G}$, as illustrated:

$$\mathcal{A} = \mathcal{F}(\mathcal{Q}, \mathcal{G}; \gamma), \tag{1}$$

where γ represents the learnable parameters of the framework $\mathcal{F}(\cdot)$. The learning process aims to maximize the likelihood of generating the correct answer $\mathcal{A}$ (Gao et al., 2025), as formulated:

$$\mathbb{E}_{(\mathcal{Q},\mathcal{A})}[\log \mathbf{P}(\mathcal{A}|\mathcal{Q}, \mathcal{G})]. \tag{2}$$

The RAG paradigm of the framework $\mathcal{F}(\cdot)$ consists of two independent components (Gao et al.,

2025). The retriever explores an evidence subgraph $\mathcal{G}_{\text{sub}}$ from the KG $\mathcal{G}$ that is most semantically relevant to the question $\mathcal{Q}$, modeled by the probability $\mathbf{P}(\mathcal{G}_{\text{sub}}|\mathcal{Q}, \mathcal{G})$. Subsequently, the predictor generates the answer $\mathcal{A}$ with the probability $\mathbf{P}(\mathcal{A}|\mathcal{Q}, \mathcal{G}_{\text{sub}})$. Therefore, the answer prediction probability can be defined:

$$\mathbf{P}(\mathcal{A}|\mathcal{Q}, \mathcal{G}; \gamma) = \sum_{\mathcal{G}_{\text{sub}} \subseteq \mathcal{G}} \mathbf{P}(\mathcal{G}_{\text{sub}}|\mathcal{Q}, \mathcal{G}) \mathbf{P}(\mathcal{A}|\mathcal{Q}, \mathcal{G}_{\text{sub}}). \tag{3}$$

3.2 Factual Information Retrieval

To preliminarily extract entities and relations relevant to the question $\mathcal{Q}$ from the KG $\mathcal{G}$, we implement factual information retrieval as the initial stage of the framework. To mitigate internal bias and reduce noise propagation, instead of leveraging the backbone LLM of the framework, we employ a domain-specific dense retriever, BMRetriever[1] (Xu et al., 2024). It is optimized to capture sufficient factual information from the KG $\mathcal{G}$ for the reliable construction of the evidence subgraph $\mathcal{G}_{\text{sub}}$. Different from existing graph-based RAG methods that retrieve the evidence subgraph $\mathcal{G}_{\text{sub}}$ based on topic entities (Wang and Yu, 2025; Mavromatis et al., 2025), we simultaneously retrieve both entities and relations, enabling a semantically rich construction of the evidence subgraph $\mathcal{G}_{\text{sub}}$ for analysis.

The question $\mathcal{Q}$ is segmented into a sequence of n terms, denoted as $\mathcal{Q} = \{x_1, \cdots, x_n\}$. In parallel, the dense retriever encodes factual information from the KG $\mathcal{G}$, including entities $e_i \in \mathcal{E}$ and relations $r_j \in \mathcal{R}$. Each segmented term $x_p \in \mathcal{Q}$ is embedded utilizing the dense retriever to ensure semantic alignment. The top-k entities and relations are retrieved by computing semantic similarity scores $\mathbf{Sim}(\cdot)$ between the embeddings of segmented terms and factual information from the KG $\mathcal{G}$ (Sohn et al., 2025), as illustrated:

$$\begin{aligned} \mathbf{Sim}(x_p, e_i) &= \mathbf{E}(x_p)^\top \mathbf{E}(e_i), \\ \mathbf{Sim}(x_q, r_j) &= \mathbf{E}(x_q)^\top \mathbf{E}(r_j), \\ \mathcal{E}^{\text{ret}} &= \text{top}{-}k(\mathbf{Sim}\{(x_p, e_i)\}), \\ \mathcal{R}^{\text{ret}} &= \text{top}{-}k(\mathbf{Sim}\{(x_q, r_j)\}), \end{aligned} \tag{4}$$

where $\mathbf{E}(\cdot)$ refers to the embedding method of the dense retriever. $\mathcal{E}^{\text{ret}}$ and $\mathcal{R}^{\text{ret}}$ represent the retrieved entity and relation sets, respectively.

[1] https://huggingface.co/BMRetriever/BMRetriever-7B

3.3 Evidence Subgraph Construction

We construct the initial evidence subgraph $\mathcal{G}_{\text{sub}}$ using the top-ranked entity $e_{\text{top}} \in \mathcal{E}^{\text{ret}}$ and relation $r_{\text{top}} \in \mathcal{R}^{\text{ret}}$. To incorporate factual information, we gradually expand from the top-ranked entity e_{top} by exploring its neighbors in the KG $\mathcal{G}$. We first retrieve all 1-hop neighbors of the top-ranked entity e_{top}. Each neighbor e_{1h} is subsequently scored according to its relevance to the question $\mathcal{Q}$. The top-m neighbors are selected to prevent the inclusion of redundant factual information during the evidence subgraph construction. Relevance is predicted utilizing a concise two-layer MLP that performs binary classification (Wang and Yu, 2025), determining whether each neighbor e_{1h} is relevant or not. It takes the concatenated embedding of the question $\mathcal{Q}$ and each neighbor e_{1h}, expressed as $\mathrm{h}_{1h} = [\mathbf{E}(\mathcal{Q})||\mathbf{E}(e_{1h})]$, as input and outputs the relevance probability y_{1h}, as defined:

$$y_{1h} = \mathbf{Softmax}(\mathrm{W}_\alpha \sigma(\mathrm{W}_\beta \mathrm{h}_{1h} + \mathrm{b}_\beta) + \mathrm{b}_\alpha), \tag{5}$$

where W_α and W_β represent learnable weight matrices. b_α and b_β denote bias terms. $\sigma(\cdot)$ refers to the ReLU activation function. The relevance probability y_{1h} assigned to the relevant class signifies the normalized relevance score for each neighbor e_{1h}, with implementation details in Appendix A.2.

We apply the same procedure to the 2-hop neighbors of the top-ranked entity e_{top}. To maintain a focused scope, the search space of the 2-hop neighbors is restricted to the previously selected 1-hop neighbors from the KG $\mathcal{G}$. In parallel, we incorporate the head-tail entity pairs associated with the top-ranked relation r_{top} and entity e_{top}. Utilizing the retrieved entities, we perform a topological readout that implements a depth-first search to construct the connected evidence subgraph $\mathcal{G}_{\text{sub}}$ (Wan et al., 2025). It preserves both the breadth information obtained from neighbor retrieval and the depth information captured through graph traversal during evidence subgraph construction, with the algorithm provided in Appendix A.3.

3.4 Evidence Subgraph Pruning

The evidence subgraph $\mathcal{G}_{\text{sub}}$ is encoded utilizing a GNN to prune and retain the set of question-relevant triplets, denoted as $\mathcal{T}_{\text{sub}}$. For each triplet $\tau_i = (h_i, r_i, t_i)$, we construct a structured feature representation f_i by concatenating the embeddings of its constituent elements, as formulated:

$$\mathrm{f}_i = [\mathbf{h}_i || \mathbf{r}_i || \mathbf{t}_i] \in \mathbb{R}^{d_{\text{GNN}}}, \quad (6)$$

where $\mathbf{h}_i$, $\mathbf{r}_i$, and $\mathbf{t}_i$ represent the GNN-derived feature representations of the head entity h_i, relation r_i, and tail entity t_i within the triplet τ_i. They are obtained using the ReaRev-based GNN method (Gao et al., 2025; Mavromatis and Karypis, 2022). For each feature representation f_i, we compute the selection probability of the triplet τ_i employing a linear layer followed by the Sigmoid activation function, expressed as $\mathbf{P}(\tau_i) = \mathbf{Sigmoid}(\mathrm{W}_i \mathrm{f}_i + \mathrm{b}_i)$, where W_i denotes the learnable weight matrix and b_i refers to the bias term.

We gradually prune the evidence subgraph $\mathcal{G}_{\text{sub}}$ while preserving its connectivity by factorizing the refinement process into independent binary decisions, enabling each triplet $\tau_i \in \mathcal{T}_{\text{sub}}$ to be decomposed and evaluated individually. The probability of selecting the evidence subgraph $\mathcal{G}_{\text{sub}}$ is shown:

$$\mathbf{P}(\mathcal{G}_{\text{sub}}) = \prod_{\tau_i \in \mathcal{G}_{\text{sub}}} \mathbf{P}(\tau_i) \prod_{\tau_j \notin \mathcal{G}_{\text{sub}}} (1 - \mathbf{P}(\tau_j)). \quad (7)$$

Detailed implementations of the refinement process and the GNN method are provided in Appendix A.4 and Appendix A.5, respectively.

3.5 Answer Prediction

The backbone LLM of the framework is prompted to predict an answer utilizing both the structured knowledge within the evidence subgraph $\mathcal{G}_{\text{sub}}$ and its internal parameterized prior knowledge. To preserve the structured information in the evidence subgraph $\mathcal{G}_{\text{sub}}$, we perform a topological readout $\mathbf{Readout}(\cdot)$ that implements a depth-first search and augment the backbone LLM with the resulting feature representations for answer prediction. Starting from the top-ranked entity e_{top} as the root node, it gradually assigns hierarchical indices to each node. Therefore, the evidence subgraph $\mathcal{G}_{\text{sub}}$ is transformed into a multi-level list that summarizes factual information. The leading index of each node encodes the depth information, while the trailing index of each node encodes the breadth information. The multi-level list is an extensively used and effective way to organize structured information in text format (Wan et al., 2025; Wang et al., 2024), making the evidence subgraph $\mathcal{G}_{\text{sub}}$ informative and convenient for the backbone LLM to interpret. The answer $\mathcal{A}$ is predicted as presented:

$$\mathcal{A} = \mathbf{LLM}(\mathcal{Q}, \mathbf{Readout}(\mathcal{G}_{\text{sub}}); \gamma). \quad (8)$$

We evaluate whether the evidence subgraph $\mathcal{G}_{\text{sub}}$ contains sufficient factual information to answer the question $\mathcal{Q}$ by designing a sufficiency classifier based on the probability $\mathbf{P}(\mathcal{A}|\mathcal{Q}, \mathcal{G}_{\text{sub}}; \gamma)$. If the probability reaches maximum, we prompt the backbone LLM to provide the final answer $\hat{\mathcal{A}}$ to the question $\mathcal{Q}$. Otherwise, the framework iteratively reconstructs the evidence subgraph, which incorporates the next top-ranked entity and relation from the retrieved entity set $\mathcal{E}^{\text{ret}}$ and relation set $\mathcal{R}^{\text{ret}}$. The framework continues pruning the evidence subgraph and predicting the answer, repeating the process until sufficient factual information is accumulated and provides the final answer $\hat{\mathcal{A}}$.

4 Experiments

We evaluate the framework through extensive experiments, including the experimental setup, experimental results, and comprehensive analysis. We aim to address four research questions (**RQs**). **RQ1:** How does the framework perform against baseline methods? **RQ2:** How does the framework perform with different backbone LLMs? **RQ3:** How do individual modules contribute to effectiveness? **RQ4:** How does the retrieved knowledge influence the generated responses in medical QA?

4.1 Experimental Setup

We introduce four components of the experimental setup to evaluate the framework.

Benchmark Datasets. We evaluate the framework on three medical QA benchmark datasets: (i) **MedQA** contains medical questions created by examination experts from extensive professional question banks (Jin et al., 2021). (ii) **MedMCQA** includes medical questions from professional medical certification exams and evaluates specialized medical knowledge (Pal et al., 2022). (iii) **MMLU-Med** involves medical questions across six subject areas (Hendrycks et al., 2021). All benchmark datasets consist of multiple-choice questions. Additional details are provided in Appendix B.

Baseline Methods. We compare the framework against twelve representative baseline methods, including three general domain LLMs, three medical domain LLMs, three RAG methods, and three graph-based RAG methods. (i) **General Domain LLMs** involve Qwen-3-8B (Yang et al.,

Model	CoT	F-T	MedQA	MedMCQA	MMLU-Med	Average
Qwen-3-8B	✔	–	0.659	0.574	0.737	0.657
LLaMA-3.2-8B	✔	–	0.612	0.530	0.648	0.597
Mistral-7B	✔	–	0.590	0.518	0.665	0.591
Meditron-7B	✔	✔	0.607	0.522	0.655	0.595
BioMistral-7B	✔	✔	0.615	0.545	0.673	0.611
Meerkat-7B	✔	✔	0.644	0.557	0.701	0.634
MedRAG	✔	–	0.685	0.604	0.772	0.687
RGAR	✔	–	0.662	0.619	0.764	0.682
RAG^2	✔	–	0.699	0.641	<u>0.780</u>	0.707
GraphRAG	–	–	<u>0.707</u>	0.633	0.758	0.699
ByoKG-RAG	–	–	0.671	0.610	0.774	0.685
AMG-RAG	✔	–	0.694	<u>0.652</u>	0.777	<u>0.708</u>
Our Framework	✔	–	**0.733**	**0.670**	**0.805**	**0.736**

Table 1: Comparison of the framework with representative baseline methods on three medical QA benchmark datasets. The best results are highlighted in bold, and the second-best results are underlined. The abbreviation CoT refers to the chain of thought prompting, and F-T stands for the question-relevant fine-tuning procedure.

2025), LLaMA-3.2-8B (Grattafiori et al., 2024), and Mistral-7B (Jiang et al., 2023). (ii) **Medical Domain LLMs** contain Meditron-7B (Chen et al., 2023), BioMistral-7B (Labrak et al., 2024), and Meerkat-7B (Kim et al., 2024). (iii) **RAG Methods** consist of MedRAG (Xiong et al., 2024), RGAR (Liang et al., 2025), and RAG^2 (Sohn et al., 2025). (iv) **Graph-Based RAG Methods** comprise GraphRAG (Edge et al., 2024), ByoKG-RAG (Mavromatis et al., 2025), and AMG-RAG (Rezaei et al., 2025). Detailed information on the baseline methods is provided in Appendix C.

Evaluation Metric. Following prior studies (Sohn et al., 2025; Rezaei et al., 2025; Liang et al., 2025), we utilize accuracy as the primary evaluation metric, defined as the proportion of correctly answered questions. The generated outputs are extracted by applying regular expression matching to the entire generated response.

Experimental Configuration. We implement Qwen-3-8B as the backbone LLM for developing the framework. Following prior research (Wang and Yu, 2025; Mavromatis et al., 2025), we set $k = 5$ and $m = 5$. We configure the LLM temperature to 0.3. All experiments are conducted on a server equipped with two NVIDIA RTX A6000 GPUs, each featuring 48 GB of VRAM, an Intel Core i9-13900K CPU, and 128 GB of RAM. Consistent with previous research in the medical domain (Rezaei et al., 2025), we employ two existing medical knowledge graphs (MKGs), including Wiki-MKG and PubMed-MKG. The complete MKGs are stored in a Neo4j database, which provides an efficient graph query engine for analysis. Additional explanations regarding the experimental configuration are provided in Appendix D. The prompt strategies utilized in the experiments are shown in Appendix E.

4.2 Main Results

Table 1 summarizes the performance comparison of the framework and representative baseline methods on three medical QA benchmark datasets.

4.2.1 Experimental Results of the Framework and Baseline Methods

The experimental results addressing **RQ1** demonstrate that the framework significantly improves the performance of its backbone LLM, Qwen-3-8B (Yang et al., 2025), by approximately 7% to 9%. It consistently outperforms publicly available LLMs, RAG methods, and graph-based RAG methods, exceeding the second-best experimental results by roughly 2% to 3%. It is worth noting that Qwen-3-8B achieves the highest accuracy among general LLMs and medical LLMs, despite having fewer than 14 billion internal parameters. It highlights the impressive capability of Qwen-3-8B under practical constraints (Frisoni et al., 2024), making it well-suited for developing personal healthcare assistants on standard devices (Qiu et al., 2024). The selection of the backbone LLM can influence the performance of baseline methods, including RAG methods and graph-based RAG methods. In this research, Qwen-3-8B is employed as the backbone

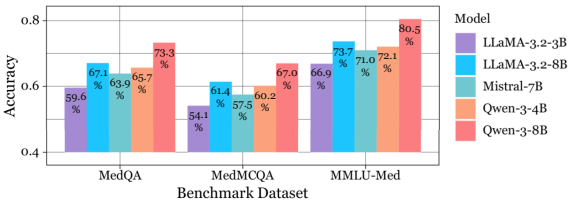

Figure 3: Comparison of the framework with five backbone LLMs on three medical QA benchmark datasets.

LLM for baseline methods under identical experimental configurations with the framework to ensure fairness. The superior experimental results of the framework validate its design and demonstrate its ability to enhance LLMs for medical QA.

The experimental results indicate that while most baseline methods and the framework utilize CoT prompting, non-CoT and graph-based RAG methods achieve the second-best performance on the MedQA and MedMCQA benchmark datasets. This finding suggests that the advantages of CoT prompting for complex medical QA may be limited compared to the advantages of incorporating domain-specific structured information from KGs. It is also noteworthy that over half of the baseline methods and the framework do not depend on computationally expensive or resource-intensive fine-tuning. Despite the development, they exhibit improved performance across the three medical QA benchmark datasets, reflecting strong robustness and scalability. This observation is further supported by the fact that all second-best experimental results are obtained without any fine-tuning on the benchmark datasets. Although fine-tuning effectively integrates domain-specific knowledge into LLMs, RAG methods remain a flexible and cost-efficient approach to enhancing medical QA.

4.2.2 Framework Performance with Different Backbone LLMs

To address **RQ2**, the framework is evaluated using five different LLMs: LLaMA-3.2-3B, LLaMA-3.2-8B, Mistral-7B, Qwen-3-4B, and Qwen-3-8B. The experimental results are presented in Figure 3.

The experimental results indicate that the scale of the backbone LLM and its released version contribute to performance improvements within the framework. Extended backbone LLM configurations with enriched internal parameters exhibit superior prior knowledge and enhanced capability for medical QA (Liang et al., 2025). Despite the advantages, the framework consistently outperforms

Ablation Setting	MedQA
w/o Medical Retriever	0.728
w/o Relevance Score	0.717
w/o Structured Feature	0.719
w/o Topological Readout	0.722
Our Framework	**0.733**

Ablation Setting	MedMCQA
w/o Medical Retriever	0.665
w/o Relevance Score	0.662
w/o Structured Feature	0.659
w/o Topological Readout	0.667
Our Framework	**0.670**

Ablation Setting	MMLU-Med
w/o Medical Retriever	0.799
w/o Relevance Score	0.789
w/o Structured Feature	0.791
w/o Topological Readout	0.796
Our Framework	**0.805**

Table 2: Experimental results of the ablation study on three medical QA benchmark dataset. The best results are highlighted in bold, and the second-best results are underlined. The abbreviation w/o denotes without.

baseline methods utilizing the same backbone LLM configurations, demonstrating the effectiveness of its iterative retrieval and pruning strategy for constructing concise evidence subgraphs. Moreover, the framework significantly improves the performance of multiple backbone LLMs across three medical QA benchmark datasets, highlighting its adaptability and transferability. We provide additional experiments in Appendix F.

4.3 Ablation Study

To address **RQ3**, we perform ablation studies on three medical QA benchmark datasets. The experimental results are illustrated in Table 2. From the experimental results of the ablation study, it can be observed that removing any key module from the framework results in a noticeable decline in its performance. However, the performance degradation does not exceed 2%, demonstrating the robustness and stability of the framework design.

The performance decline is noticeable on complex medical QA benchmark datasets with extended average question lengths, such as MedQA and MMLU-Med. In the aforementioned cases, the domain-specific dense retriever for factual information retrieval contributes less effectively to

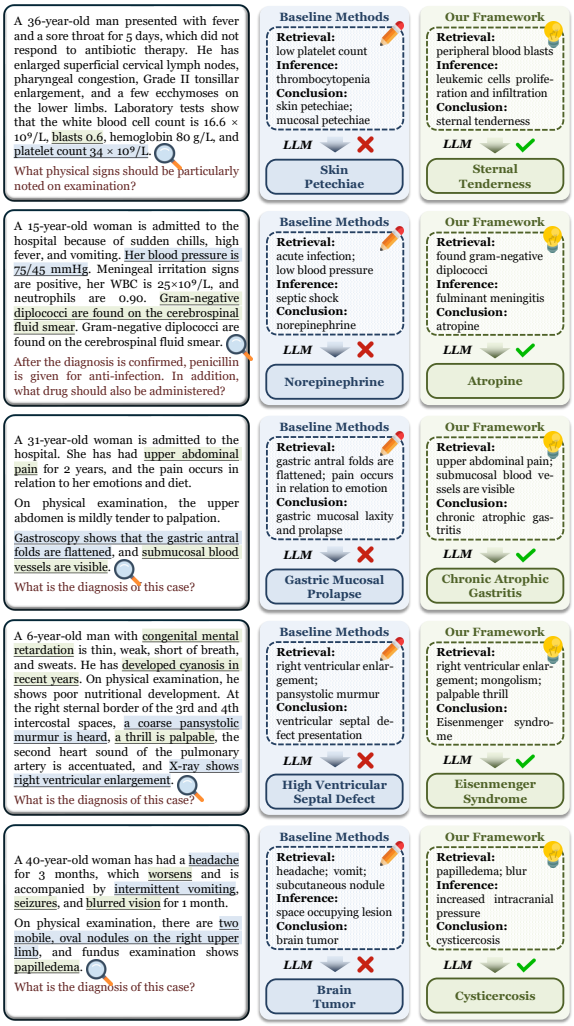

Figure 4: Comparison of the framework with the second-best-performing baseline methods on five cases.

the topological readout during answer prediction, particularly when the evidence subgraph becomes extensive. It is evident from the experimental settings **w/o medical retriever** and **w/o topological readout**, where the domain-specific dense retriever, BMRetriever, is replaced with a general domain dense retriever, LLM-Embedder[2] (Zhang et al., 2024), and the topological readout is substituted with a plain-text description of the refined evidence subgraph, respectively.

In the experimental settings **w/o relevance score** and **w/o structured feature**, the framework heuristically retrieves every 2-hop neighbor of the top-ranked entity during evidence subgraph construction and exclusively ranks the semantic relevance of each retrieved triplet to the question utilizing embeddings from the domain-specific dense retriever for evidence subgraph pruning, respectively. Under the aforementioned conditions, the performance of the framework drops significantly. The inclusion of relevance scores for selecting neighbors and structured feature representations for pruning retrieved triplets jointly enables the development of a concise yet informative evidence subgraph that remains closely aligned with the medical question. It underscores the importance of minimizing redundant and noisy factual information in graph-based RAG methods, not only during the refinement process but also at the stages of graph retrieval.

4.4 Case Study

To address **RQ4**, we conduct a complex case study that compares the summarized question-relevant factual information produced by the second-best performing baseline methods with the summarized evidence subgraphs generated by the framework. The analysis focuses on five cases from benchmark datasets where the baseline methods fail to predict accurate answers due to potential redundant and noisy factual information, as presented in Figure 4, whereas the framework successfully provides correct responses to medical questions, thereby illustrating that the quality of the retrieved factual information can influence medical QA.

5 Conclusion

In this paper, we introduce an iterative medical QA framework that enhances LLMs by progressively integrating precise factual information retrieval, focused evidence subgraph construction, and targeted evidence subgraph pruning from domain-specific KGs to reduce knowledge redundancy while maintaining coherent and reliable inference. The framework achieves state-of-the-art performance on three medical QA benchmarks and outperforms representative baseline competitors, demonstrating its effectiveness in domain-specific tasks.

Limitations

We acknowledge four limitations: (i) The development process is restricted to open-source LLMs with relatively limited internal parameters, which are resource-intensive and less scalable. It cannot be directly applied to closed-source LLMs based on APIs that generally offer stronger performance. Future research could design a flexible pipeline that supports a broader range of LLMs. (ii) The framework is augmented with KGs, indicating that its overall performance depends on the

[2] https://huggingface.co/BAAI/llm-embedder

quality and completeness of knowledge resources. Incomplete, inconsistent, or outdated factual information may hinder reliability. (iii) The framework captures factual information from 2-hop neighbors. While adequate for this research, various domain-specific KGs may require reasoning over deep relational paths. Future research could explore advanced and adaptive path-search strategies that incorporate multi-hop neighbors to leverage complex graph structures. (iv) The framework is developed and evaluated on three benchmark datasets consisting of multiple-choice questions for medical QA, which limit its reliability and generalizability to open-ended medical questions with broader contextual coverage. Extending the framework from multiple-choice questions to open-ended questions for medical QA could be a significant direction for future research. (v) The framework relies on a bag-of-words representation to retrieve question-relevant components from KGs during factual information retrieval and may not fully exploit the relations that are explicit or implicitly expressed in questions. Introducing efficient relation prediction mechanisms during factual information retrieval to enhance the informativeness of the framework could be an important direction for future research. (vi) The framework is designed for the medical domain, which limits its comprehensiveness and effectiveness in general or other targeted domains. Developing effective KG refinement and integration strategies to adapt the framework to complex scenarios in general or other targeted domains could be a persistent direction for future research.

Ethical Considerations

The development of the framework for medical QA necessitates careful ethical consideration due to the potential risks of inaccuracy and bias. This research strictly adheres to the ACL Code of Ethics. All benchmark datasets utilized are publicly available and sufficiently anonymized, which contain no personally identifiable information. Both benchmark datasets and baseline methods are employed in full compliance with the corresponding intended purposes and licenses for the term of usage, either under open-access terms or through authorized protocols. Both benchmark datasets and baseline methods are strictly used for research purposes only, with the consent of the creators or copyright holders, without any further changes or modifications. The objective of the framework is to improve the performance of LLMs in medical QA, and we unequivocally condemn any potential misuse.

While LLMs present noticeable benefits in extensive NLP applications within the medical domain, they also introduce potential risks. Even with retrieved factual information, responses generated from LLMs may contain errors or retain existing biases. Given the significant impact of medical information on healthcare decisions, we strongly advocate for a conservative implementation strategy. All outputs provided by the framework should be rigorously reviewed by qualified medical professionals before any practical application in real-world settings. The stringent validation is essential to maintaining the integrity of medical communication and preventing the spread of inaccurate or harmful information. Although the framework improves medical QA, we acknowledge that bias mitigation remains a considerable challenge. This research focuses on establishing the technical validity through standardized evaluations, while ethical and deployment considerations remain important directions for future research.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, and Shyamal Anadkat. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Zeming Chen, Alejandro Hernández Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. 2023. Meditron-70B: Scaling medical pretraining for large language models. *arXiv preprint arXiv:2311.16079*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Giacomo Frisoni, Alessio Cocchieri, Alex Presepi, Gianluca Moro, and Zaiqiao Meng. 2024. To generate or to retrieve? On the effectiveness of artificial contexts for medical open-domain question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9878–9919.

Guangze Gao, Zixuan Li, Chunfeng Yuan, Jiawei Li, Wu Jianzhuo, Yuehao Zhang, Xiaolong Jin, Bing Li, and Weiming Hu. 2025. D-RAG: Differentiable retrieval-augmented generation for knowledge graph question answering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 35386–35405.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, and Alex Vaughan. 2024. The LLaMA 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Bolei He, Xinran He, Run Shao, Shanfu Shu, Xianwei Xue, MingQuan Cheng, Haifeng Li, and Zhen-Hua Ling. 2025. Select to know: An internal-external knowledge self-selection framework for domain-specific question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10683–10703.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Hyunjae Kim, Hyeon Hwang, Jiwoo Lee, Sihyeon Park, Dain Kim, Taewhoo Lee, Chanwoong Yoon, Jiwoong Sohn, Donghee Choi, and Jaewoo Kang. 2024. Small language models learn enhanced reasoning skills from medical textbooks. *arXiv preprint arXiv:2404.00376*.

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. 2024. BioMistral: A collection of open-source pretrained large language models for medical domains. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5848–5864.

Sichu Liang, Linhai Zhang, Hongyu Zhu, Wenwen Wang, Yulan He, and Deyu Zhou. 2025. RGAR: Recurrence generation-augmented retrieval for factual-aware medical question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 4006–4033.

Can Lin, Zhengwang Jiang, Ling Zheng, Qi Zhao, Yuhang Zhang, Qi Song, and Wangqiu Zhou. 2025. RJE: A retrieval-judgment-exploration framework for efficient knowledge graph question answering with LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17288–17305.

Zhutian Lin, Junwei Pan, Shangyu Zhang, Ximei Wang, Xi Xiao, Shudong Huang, Lei Xiao, and Jie Jiang. 2024. Understanding the ranking loss for recommendation with sparse user feedback. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5409–5418.

Costas Mavromatis, Soji Adeshina, Vassilis N. Ioannidis, Zhen Han, Qi Zhu, Ian Robinson, Bryan Thompson, Huzefa Rangwala, and George Karypis. 2025. BYOKG-RAG: Multi-strategy graph retrieval for knowledge graph question answering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27869–27886.

Costas Mavromatis and George Karypis. 2022. ReaRev: Adaptive reasoning for question answering over knowledge graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2447–2458.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning*, pages 248–260.

Jianing Qiu, Kyle Lam, Guohao Li, Amish Acharya, Tien Yin Wong, Ara Darzi, Wu Yuan, and Eric J Topol. 2024. LLM-based agentic systems in medicine and healthcare. *Nature Machine Intelligence*, 6(12):1418–1420.

Mohammad Reza Rezaei, Reza Saadati Fard, Jayson Lee Parker, Rahul G Krishnan, and Milad Lankarany. 2025. Agentic medical knowledge graphs enhance medical question answering: Bridging the gap between LLMs and evolving medical knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 12682–12701.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, and Stephen Pfohl. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

Jiwoong Sohn, Yein Park, Chanwoong Yoon, Sihyeon Park, Hyeon Hwang, Mujeen Sung, Hyunjae Kim, and Jaewoo Kang. 2025. Rationale-guided retrieval augmented generation for medical question answering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12739–12753.

Yuan Sui, Yufei He, Zifeng Ding, and Bryan Hooi. 2025a. Can knowledge graphs make large language models more trustworthy? An empirical study over open-ended question answering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12685–12701.

Yuan Sui, Yufei He, Nian Liu, Xiaoxin He, Kun Wang, and Bryan Hooi. 2025b. FiDeLiS: Faithful reasoning in large language models for knowledge graph question answering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8315–8330.

Junhong Wan, Tao Yu, Kunyu Jiang, Yao Fu, Weihao Jiang, and Jiang Zhu. 2025. Digest the knowledge: Large language models empowered message passing for knowledge graph question answering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15426–15442.

Fengyi Wang, Guanghui Zhu, Chunfeng Yuan, and Yihua Huang. 2024. LLM-enhanced cascaded multi-level learning on temporal heterogeneous graphs. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 512–521.

Shuai Wang and Yinan Yu. 2025. iQUEST: An iterative question-guided framework for knowledge base question answering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15616–15628.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6233–6251.

Ran Xu, Wenqi Shi, Yue Yu, Yuchen Zhuang, Yanqiao Zhu, May Dongmei Wang, Joyce C. Ho, Chao Zhang, and Carl Yang. 2024. BMRetriever: Tuning large language models as better biomedical text retrievers. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22234–22254.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, and Chenxu Lv. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Peitian Zhang, Zheng Liu, Shitao Xiao, Zhicheng Dou, and Jian-Yun Nie. 2024. A multi-task embedder for retrieval augmented LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3537–3553.

Algorithm 1: Medical Question Answering

Input: Question $\mathcal{Q}$; Knowledge Graph $\mathcal{G}$; Entity Set $\mathcal{E}$; Relation Set $\mathcal{R}$.
Output: Final Answer $\hat{\mathcal{A}}$.
Initialization:
$\mathcal{G}_{\text{sub}} \leftarrow \emptyset$; $\mathcal{E}_{\text{sub}} \leftarrow \emptyset$; $\mathcal{R}_{\text{sub}} \leftarrow \emptyset$.
(i) Factual Information Retrieval:
$\mathcal{E}^{\text{ret}} \leftarrow \text{top}{-}k(\mathbf{Sim}(\mathcal{Q}, \mathcal{E}))$;
$\mathcal{R}^{\text{ret}} \leftarrow \text{top}{-}k(\mathbf{Sim}(\mathcal{Q}, \mathcal{R}))$;
for $i \leftarrow 1$ **to** k **do**
  (ii) Evidence Subgraph Construction:
  $\mathcal{E}_{\text{sub}} \leftarrow \mathcal{E}_{\text{sub}} \cup e_i \in \mathcal{E}^{\text{ret}}$;
  $\mathcal{R}_{\text{sub}} \leftarrow \mathcal{R}_{\text{sub}} \cup r_i \in \mathcal{R}^{\text{ret}}$;
    (a) Neighbor Retrieval:
    $\mathcal{E}_{1\text{h}} \leftarrow \mathbf{Score}(\mathbf{1\text{-}Hop}(e_i, \mathcal{G}), \mathcal{Q})$;
    $\mathcal{E}_{2\text{h}} \leftarrow \mathbf{Score}(\mathbf{1\text{-}Hop}(\mathcal{E}_{1\text{h}}, \mathcal{G}), \mathcal{Q})$;
    (b) Entity Pair Retrieval:
    $\mathcal{E}_r \leftarrow \mathbf{Head}(r_i, \mathcal{G}) \cup \mathbf{Tail}(r_i, \mathcal{G})$;
    (c) Subgraph Retrieval:
    $\mathcal{G}_{\text{sub}} \leftarrow \mathbf{Readout}(\mathcal{E}_{1\text{h}} \cup \mathcal{E}_{2\text{h}} \cup \mathcal{E}_r)$;
  (iii) Evidence Subgraph Pruning:
  $\mathcal{T}_{\text{sub}} \leftarrow \mathbf{Decompose}(\mathcal{G}_{\text{sub}})$;
  for each $\tau \in \mathcal{T}_{\text{sub}}$ **do**
    if $\mathbf{Select}(\mathbf{GNN}(\tau)) = \text{no}$ **then**
      if not affect connectivity **then**
        $\mathcal{G}_{\text{sub}} \leftarrow \mathbf{Prune}(\tau, \mathcal{G}_{\text{sub}})$;
      end
    end
  end
  (iv) Answer Prediction:
  $\mathcal{A} = \mathbf{LLM}(\mathcal{Q}, \mathbf{Readout}(\mathcal{G}_{\text{sub}}))$;
  if $\mathbf{Classifier}(\mathcal{G}_{\text{sub}}) = \text{Sufficient}$ **then**
    $\hat{\mathcal{A}} \leftarrow \mathcal{A}$;
    break
  end
end
return $\hat{\mathcal{A}}$

A Implementation Details

We provide implementation details for the methods used in the framework, including the algorithms, loss designs, and factorizations.

A.1 Algorithm for the Framework

The implementation of the framework is summarized in Algorithm 1.

Algorithm 2: Topological Readout

Input: Knowledge Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$;
Top-ranked Entity e_{top}.
Output: Graph Summary $\mathcal{Z}$.
Initialization:
Stack $\mathcal{S} \leftarrow \emptyset$; Index Dictionary $\mathcal{D} \leftarrow \emptyset$.
$\mathbf{Push}(e_{\text{top}}, \mathcal{S})$;
$\mathcal{D}[e_{\text{top}}] \leftarrow \varepsilon$;
while $\mathcal{S} \neq \emptyset$ **do**
  $t \leftarrow \mathbf{Pop}(\mathcal{S})$;
  if $\mathbf{Neighbors}(t) \neq \emptyset$ **then**
    $\mathcal{I} \leftarrow 1$;
    for each $e_{1h} \in \mathbf{Neighbors}(t)$ **do**
      if e_{1h} is not visited **then**
        $\mathcal{D}[e_{1h}] \leftarrow$
        $\mathcal{D}[t] \oplus \mathbf{String}(\mathcal{I}) \oplus' .'$;
        $\mathbf{Push}(e_{1h}, \mathcal{S})$;
        $\mathcal{I} \leftarrow \mathcal{I} + 1$;
      end
    end
  end
end
for each e_i in $\mathbf{Keys}(\mathcal{D})$ **do**
  $z_i \leftarrow \mathcal{D}[e_i] \oplus h_i$
end
return $\mathcal{Z} = \{z_i\}_{i \in \mathcal{E}}$

A.2 Loss Design for Relevance Prediction

We train the binary classifier of relevance prediction for each neighbor e_{1h} using cross-entropy loss with one-hot encoding, as formulated:

$$\mathcal{L}_{1h} = -\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{2} \tilde{y}_{i,j} \log(\hat{y}_{i,j}),$$

where n represents the number of neighbors. $\tilde{y}_{i,j}$ signifies the ground-truth label of the i-th neighbor for class j, and $\hat{y}_{i,j}$ denotes the predicted probability of the i-th neighbor for class j.

A.3 Algorithm for Topological Readout

The implementation of the topological readout is provided in Algorithm 2 (Wan et al., 2025).

A.4 Factorization for Subgraph Refinement

The factorization utilized for evidence subgraph refinement approximates the complex probability by evaluating each triplet $\tau_i \in \mathcal{T}_{\text{sub}}$, assuming that the selection of each triplet τ_i is independent (Gao et al., 2025). For an evidence subgraph $\mathcal{G}_{\text{sub}}$ containing $\mathcal{N}$ triplets, each triplet τ_i can be labeled as either selected or not, resulting in $2^{\mathcal{N}}$ possible evidence subgraphs. The sum of probabilities over all possible evidence subgraphs can be defined as:

$$\begin{aligned}
&\sum_{\mathcal{G}_{\text{sub}}} \mathbf{P}(\mathcal{G}_{\text{sub}}) \\
&= \sum_{\mathcal{G}_{\text{sub}}} \prod_{\tau_i \in \mathcal{G}_{\text{sub}}} \mathbf{P}(\tau_i) \prod_{\tau_j \notin \mathcal{G}_{\text{sub}}} (1 - \mathbf{P}(\tau_j)) \\
&= \sum_{\tau_i} \cdots \sum_{\tau_{\mathcal{N}}} \prod_{i=1}^{\mathcal{N}} \mathbf{P}(\tau_i)^{\mathbf{I}(\tau_i)} (1 - \mathbf{P}(\tau_i))^{1 - \mathbf{I}(\tau_i)} \\
&= \prod_{i=1}^{\mathcal{N}} \sum_{\mathbf{I}(\tau_i) \in \{0,1\}} \mathbf{P}(\tau_i)^{\mathbf{I}(\tau_i)} (1 - \mathbf{P}(\tau_i))^{1 - \mathbf{I}(\tau_i)} \\
&= \prod_{i=1}^{\mathcal{N}} (\mathbf{P}(\tau_i) + (1 - \mathbf{P}(\tau_i))) \\
&= 1,
\end{aligned}$$

where considering all subgraphs corresponds to evaluating both possibilities for each triplet τ_i independently. In the equation, $\mathbf{I}(\cdot)$ represents an indicator function equal to 1 if the triplet τ_i is included in the evidence subgraph and 0 otherwise.

To incorporate structured dependencies between triplets that may be overlooked by the independence assumption in the selection of each triplet, we implement the ReaRev-based GNN method (Gao et al., 2025; Mavromatis and Karypis, 2022). The inherent capability of the GNN method to systematically encode graph-based structured information enables it to capture correlations among triplets within the evidence subgraph, thereby mitigating the limitations of the independence assumption.

A.5 Loss Design for GNN Method

We adapt the ReaRev-based GNN method to encode graph-based structured information. Specifically, a domain-specific dense retriever (Xu et al., 2024) is first used to convert queries into instructions. Subsequently, the ReaRev-based GNN method initializes and updates node feature representations through message passing, incorporating interactions between the instructions and nodes. Based on the updated node feature representations and the predicted terminal node feature distributions, the ReaRev-based GNN method refines the instructions accordingly. The loss for triplet selection is presented as:

$$\mathcal{L}_\tau = -\sum_{\tau \in \mathcal{G}} [y_\tau \log \mathbf{P}(\tau) + (1 - y_\tau) \log(1 - \mathbf{P}(\tau))],$$

where $y_\tau = 1$ if $\tau \in \mathcal{G}_{\text{sub}}$, otherwise $y_\tau = 0$.

To mitigate the sparsity of positive examples in KG triplet classification, we integrate an additional ranking loss (Gao et al., 2025; Lin et al., 2024):

$$\mathcal{L}_{\text{rank}} = -\frac{1}{\mathcal{N}_+ \mathcal{N}_-} \sum_{i=1}^{\mathcal{N}_+} \sum_{j=1}^{\mathcal{N}_-} \log \sigma(\mathbf{P}(\tau_i) - \mathbf{P}(\tau_j)), \tag{9}$$

where $\mathcal{N}_+$ and $\mathcal{N}_-$ represent the numbers of positive and negative examples, respectively. τ_i and τ_j denote a positive and negative triplet, respectively. σ refers to the Sigmoid activation function. The ranking loss generates larger gradients on sparse examples, complementing the cross-entropy objective and improving classification performance.

The overall loss is a weighted combination of the aforementioned two losses, as presented:

$$\mathcal{L}_{\text{GNN}} = \eta \mathcal{L}_\tau + (1 - \eta) \mathcal{L}_{\text{rank}},$$

where $\eta = 0.7$ is empirically set in this research to balance the cross-entropy loss and the rank loss (Gao et al., 2025; Lin et al., 2024).

B Benchmark Datasets

We evaluate the framework using three benchmark datasets within the medical domain, including **MedQA**, **MedMCQA**, and **MMLU-Med**.

- **MedQA** is a multiple-choice medical QA dataset collected from professional medical board examinations. It requires retrieving relevant evidence and applying multi-step inference to answer medical questions accurately. Each question within MedQA includes several answer options that require a solid understanding of medical concepts and logical inference, generally grounded in medical textbook knowledge. MedQA spans a maximum of 872 tokens per question, with an average length of 197 tokens per question and a minimum of 50 tokens per question. Following the standardized experimental setting (Jin et al., 2021), we utilize 1,272 samples for model development, 10,178 samples for model training, and 1,273 samples for model testing.

- **MedMCQA** is a multiple-choice medical QA dataset that covers both foundational and medical knowledge across extensive medical specialties. The medical questions span approximately 2,400 healthcare topics and 21 medical subjects, providing comprehensive coverage of the medical domain. Answering medical questions within MedMCQA requires retrieving medical knowledge relevant to each specific scenario. Due to its exam-oriented design, MedMCQA contains filtered and concise question statements. It includes questions with a maximum length of 207 tokens, an average length of 41 tokens, and a minimum of 11 tokens. Following the standardized experimental setting (Pal et al., 2022), we utilize 4,183 samples for model development, 182,822 samples for model training, and 6,150 samples for model testing.

- **MMLU-Med** known as the Massive Multitask Language Understanding (MMLU) benchmark dataset, which evaluates the multitask learning capabilities of LLMs across 57 diverse subjects (Hendrycks et al., 2021). We select a subset of six tasks that are related to the medical domain (Xiong et al., 2024), including anatomy, clinical knowledge, professional medicine, human genetics, college medicine, and college biology. The subset is collectively referred to as MMLU-Med (Singhal et al., 2023). MMLU-Med contains questions with a maximum length of 961 tokens, an average length of 87 tokens, and a minimum of 17 tokens. Consistent with previous work (Xiong et al., 2024), we evaluate on the test sets of six tasks, involving 1,089 samples in total.

The questions in the three benchmark datasets are recorded in English. The sample questions from the three benchmark datasets within the medical domain are illustrated in Figure 5.

C Baseline Methods

We compare the framework with twelve representative baseline methods, including general domain LLMs, medical domain LLMs, RAG methods, and graph-based RAG methods. Specifically, the evaluation focuses on baseline methods with **fewer than 14 billion** internal parameters, reflecting practical constraints for developing a personal healthcare

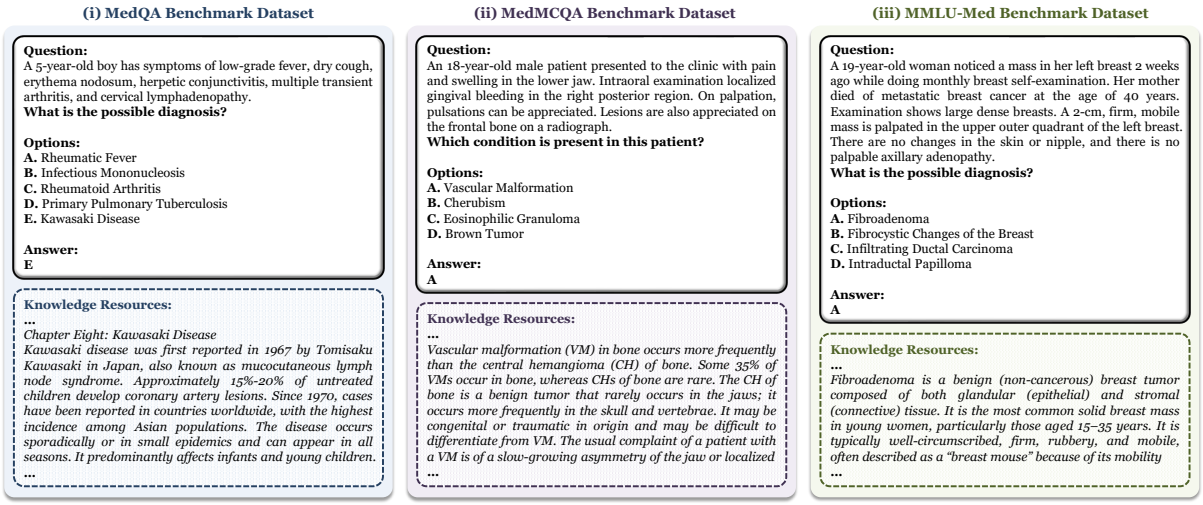

Figure 5: The sample questions from the three benchmark datasets within the medical domain.

assistant on standard devices (Qiu et al., 2024), and aligning with typical definitions of resource-constrained environments (Frisoni et al., 2024).

For general domain LLMs, we evaluate **Qwen-3-8B** (Yang et al., 2025), **LLaMA-3.2-8B** (Grattafiori et al., 2024), and **Mistral-7B** (Jiang et al., 2023). To fully leverage the reasoning capabilities of general domain LLMs, we incorporate CoT prompting (Wei et al., 2022) and integrate domain-specific knowledge retrieved by the domain-specific dense retriever from the provided knowledge resources (Xu et al., 2024), ensuring a fair comparison.

- **Qwen-3-8B**[3] (Achiam et al., 2023) is a Transformer-based general domain LLM trained on a large-scale high-quality corpus. It integrates supervised fine-tuning and multistage reinforcement learning in the post-training phase, complementing its extensive pre-training to enhance performance.
- **LLaMA-3.2-8B**[4] (Grattafiori et al., 2024) is a decoder-only Transformer-based general domain LLM with extensive internal parameters and an extended context window, which supports enhanced reasoning performance. It is trained through an iterative post-training process that combines supervised fine-tuning with direct preference optimization.
- **Mistral-7B**[5] (Jiang et al., 2023) is a Transformer-based general domain LLM that leverages grouped-query attention for fast inference and sliding-window attention to handle long sequences efficiently with a reduced computational cost.

For medical domain LLMs, we evaluate **Meditron-7B** (Chen et al., 2023), **BioMistral-7B** (Labrak et al., 2024), and **Meerkat-7B** (Kim et al., 2024). To ensure a fair comparison, we apply CoT prompting (Wei et al., 2022) and incorporate domain-specific knowledge retrieved by the domain-specific dense retriever from the provided knowledge resources (Xu et al., 2024).

- **Meditron-7B**[6] (Chen et al., 2023) is a domain-specific adaptation of LLaMA, achieved through continued pre-training on a comprehensive corpus in the medical domain.
- **BioMistral-7B**[7] (Labrak et al., 2024) is a domain-specific adaptation of Mistral. It is pre-trained on extensive text content from a corpus in the medical domain and systematically refined through supervised fine-tuning.
- **Meerkat-7B**[8] (Kim et al., 2024) is developed by extracting CoT reasoning paths from multiple medical textbooks and combining reasoning paths with diverse instruction-following datasets in the medical domain. Following prior research (Sohn et al., 2025), Meerkat-7B is initialized with Mistral-7B weights.

[3] https://huggingface.co/Qwen/Qwen3-8B
[4] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
[5] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3
[6] https://huggingface.co/epfl-llm/meditron-7b
[7] https://huggingface.co/BioMistral/BioMistral-7B
[8] https://huggingface.co/dmis-lab/meerkat-7b-v1.0

For RAG methods, we evaluate **MedRAG** (Xiong et al., 2024), **RGAR** (Liang et al., 2025), and **RAG**^{2} (Sohn et al., 2025). We apply the provided knowledge resources from benchmark datasets and adhere strictly to the experimental configurations reported in the original references.

- **MedRAG**[9] (Xiong et al., 2024) is a unified framework for optimizing RAG with process supervision, applicable to medical QA. It includes domain-specific corpora from four knowledge resources, four retrieval algorithms, and six LLMs. We report experimental results from the best-performing configuration across different options.
- **RGAR**[10] (Liang et al., 2025) is a recurrence generation-augmented retrieval framework that integrates both domain-specific factual information and conceptual knowledge for medical QA. We report experimental results from the best-performing configuration across different options.
- **RAG**^{2}[11] (Sohn et al., 2025) is a framework designed to improve the reliability of RAG in the medical domain. It integrates optimized techniques for query formulation, retrieval, and filtering in medical QA. We report experimental results from the best-performing configuration across all experimental settings.

For graph-based RAG methods, we evaluate **GraphRAG** (Edge et al., 2024), **ByoKG-RAG** (Mavromatis et al., 2025), and **AMG-RAG** (Rezaei et al., 2025). We use the knowledge resources provided by benchmark datasets as context and the same medical KGs to support Graph-based RAG methods, following the experimental configurations reported in the original references.

- **GraphRAG**[12] (Edge et al., 2024) is a graph-based RAG method designed to enhance LLM with structured knowledge from KGs. GraphRAG leverages the topology and relationships in KGs to provide precise and contextually relevant information.
- **ByoKG-RAG**[13] (Mavromatis et al., 2025) is a framework that improves QA with custom KGs utilizing multi-strategy graph linking and retrieval. LLMs generate key graph artifacts, which are linked to KGs and used to retrieve graph context. The graph context enables LLMs to iteratively refine graph linking and retrieval before generating the final answer.
- **AMG-RAG**[14] (Rezaei et al., 2025) is a graph-based RAG framework that automates KG construction and updating, incorporates reasoning, and retrieves up-to-date external evidence from KGs for medical QA.

D Experimental Configuration

We employ Chroma as the vector database, which is persistently stored on disk to enable efficient reusability. The input length to the backbone LLM is set to 512 tokens per turn, with a 100-token overlap between consecutive turns to maintain contextual consistency (Rezaei et al., 2025). The framework adopts distinct training strategies for optimization on the training samples of benchmark datasets (Gao et al., 2025), where negative samples are generated through random negative sampling. During evidence subgraph construction and pruning, the relevance classifier and the GNN are fully trained with a learning rate of 2e-5 and 5e-5 for 30 epochs, respectively. The domain-specific dense retriever is configured with a hidden dimension of 4096. The GNN adopts a hidden dimension of 128. The framework employs the AdamW optimizer with a weight decay of 0.001, a batch size of 16, and a cosine learning rate scheduler.

E Prompt Strategies

We provided the prompt strategies implemented in the experiments.

The prompt strategy employed to instruct the backbone LLM of the framework to predict answers using the evidence subgraph and its internal parameterized prior knowledge is presented in Table 3, where the summary outlines represent the topological readout of the evidence subgraph.

The prompt strategy utilized to instruct the baseline LLMs to predict answers through CoT prompting and the provided knowledge resources is presented in Table 4, where context refers to the domain-specific knowledge retrieved by the dense retriever from the knowledge resources.

[9] https://github.com/Teddy-XiongGZ/MedRAG
[10] https://github.com/dbsxfz/RGAR
[11] https://github.com/dmis-lab/RAG2
[12] https://github.com/microsoft/graphrag
[13] https://github.com/awslabs/graphrag-toolkit
[14] https://github.com/MrRezaeiUofT/AMG-RAG

Question Answering Prompt
Prompt: You are an experienced medical expert, and your task is to answer a multiple-choice medical question. Based on the given factual information and your own knowledge, please think step-by-step and predict the single best option from the given options as the final answer to the question. Please be informed that your responses will be used for research purposes only. **Question:** <Question> **Factual Information:** <Summary Outlines>

Table 3: The prompt strategy to predict answers.

Baseline LLMs with CoT Prompt
Prompt: You are an experienced medical expert, and your task is to answer a multiple-choice medical question. Based on the given knowledge resources and your own knowledge, please think step-by-step and predict the single best option from the given options as the final answer to the question. Please be informed that your responses will be used for research purposes only. **Question:** <Question> **Knowledge Resources:** <Context>

Table 4: The prompt strategy of baseline LLMs.

Model	Total Time	Avg. Time
RAG^2	0.38h	1.08s
GraphRAG	0.18h	0.51s
AMG-RAG	0.57h	1.61s
Our Framework	0.30h	0.85s

Table 5: Comparison of the framework with the second-best performing baseline methods on the MedQA benchmark dataset for execution time. The abbreviation avg denotes average.

Model	Total Time	Avg. Time
RAG^2	0.60h	0.35s
GraphRAG	0.46h	0.27s
AMG-RAG	0.72h	0.42s
Our Framework	0.51h	0.30s

Table 6: Comparison of the framework with the second-best performing baseline methods on the MedMCQA benchmark dataset for execution time. The abbreviation avg denotes average.

Model	Total Time	Avg. Time
RAG^2	0.16h	0.54s
GraphRAG	0.10h	0.32s
AMG-RAG	0.24h	0.79s
Our Framework	0.15h	0.48s

Table 7: Comparison of the framework with the second-best performing baseline methods on the MMLU-Med benchmark dataset for execution time. The abbreviation avg denotes average.

F Additional Experiments

We present additional experiments to compare the performance of the framework for execution time and knowledge coverage.

F.1 Execution Time Comparison

We evaluate the execution time of the framework and the second-best performing baseline methods, including RAG^2 (Sohn et al., 2025), GraphRAG (Edge et al., 2024), and AMG-RAG (Rezaei et al., 2025), on three medical QA benchmark datasets. The experimental results are illustrated in Table 5, Table 6, and Table 7.

From the experimental results, the framework achieves an effective balance between execution time and performance. Specifically, as the average token length per question increases, both the framework and the second-best performing baseline methods require extended execution time for each question. Although GraphRAG demonstrates slightly faster execution time, the performance comparison summarized in Table 1 indicates a clear performance advantage for the framework. For real-time medical QA applications, the framework offers an optimal trade-off between accuracy and efficiency. In contrast, other baseline methods lag behind both the framework and GraphRAG, which makes baseline methods less suitable for medical QA applications (Liang et al., 2025), where reliability and computational efficiency are critical.

Specifically, the experimental results indicate that the input length provided by RAG methods to the backbone LLM substantially influences execution time, followed by the internal operations of RAG methods, such as encoding, retrieval, and refinement. AMG-RAG identifies domain-specific terminology and progressively retrieves contextual information to construct a KG for traversal (Rezaei et al., 2025). However, since the retrieved information is not refined and only the reasoning paths are pruned, its input length remains extensive, which

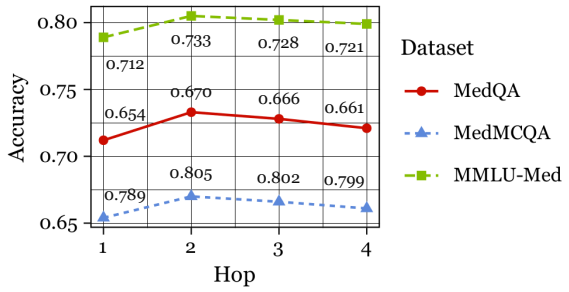

Figure 6: Comparison of the framework across different depths of knowledge coverage in the evidence subgraph on three medical QA benchmark datasets.

results in the longest execution time from the experimental results. Although the framework, RAG^2, and GraphRAG incorporate refinement strategies after domain-specific knowledge retrieval (Sohn et al., 2025; Edge et al., 2024), GraphRAG achieves the shortest execution time through three stages of refinement that significantly improve efficiency compared to conventional RAG methods. Nonetheless, the framework demonstrates superior effectiveness within the medical domain, achieving enhanced performance while maintaining flexibility for medical QA.

Since embeddings generated by different embedding methods are pre-stored, the retrieval process generally occurs within a limited part of execution time. Consequently, multiple retrieval operations have a negligible impact, and employing different retrieval strategies may result in comparable execution times. Meanwhile, as presented in Table 1, GraphRAG does not utilize CoT prompting, suggesting that CoT prompting can produce unstable input lengths, which may in turn increase the execution time of RAG methods in medical QA.

F.2 Performance Comparison

We evaluate the performance of the framework across different depths of knowledge coverage in the evidence subgraph, which ranges from 1-hop to 4-hop neighbors, on three medical QA benchmark datasets to emphasize its capability for step-wise inference. The experimental results are illustrated in Figure 6.

From the experimental results, it can be observed that increasing the depth of knowledge coverage in the evidence subgraph from 1-hop to 4-hop neighbors enhances the completeness and exploration of reasoning paths, leading to improved accuracy, particularly in the 1-hop and 2-hop experimental settings for medical QA. While complex medical questions may indeed require extended depth of knowledge coverage for multi-step inference, excessive expansion of the evidence subgraph for all questions in the benchmark datasets increases the complexity of the retrieved factual information, resulting in redundant and noisy structured knowledge in the medical domain. The effect is evident in the performance differences obtained across the 3-hop and 4-hop experimental settings. Although the framework introduces evidence subgraph pruning to refine the retrieved evidence subgraph by comparing structured feature representations, the 2-hop configuration, which is adopted as the default experimental setting of the framework, achieves an optimal balance between knowledge coverage and performance for medical QA. Future research may focus on dynamic hop selection during factual information retrieval to enhance adaptability.