MedJudgeRAG: Option-Wise Evidence Judgment with Dynamic Knowledge Graphs for Medical MCQA

Seongwon Seo [1] **Seung Hwan Cho** [1] **Young-Min Kim** [1 2]

Abstract

In medical multiple-choice question answering (MCQA), Retrieval-Augmented Generation (RAG) can supplement the domain knowledge of language models (LMs). However, since vanilla RAG indiscriminately utilizes retrieved documents, it can degrade LM performance. To address this, we propose MedJudgeRAG. Our framework represents retrieved documents as a dynamic knowledge graph (KG) composed of entities and relations. For each option, the model judges an evidence verdict from the retrieved documents and the KG. Based on the verdict combination, the model determines a knowledge utilization strategy to reason toward the final answer. These capabilities are trained via supervised fine-tuning using structured reasoning traces generated by a teacher LM. The training employs a weighted cross-entropy loss that differentially weights the KG and reasoning segments. Experiments on two medical MCQA benchmarks demonstrate that MedJudgeRAG consistently outperforms both vanilla RAG and parametric baselines. Furthermore, ablation analysis reveals that the dynamic KG is more effective as graph-conditioned supervision at training time than as an explicit output at inference time. Our code is available at https://github.com/hyu-amllab/medjudgerag, and the generated reasoning traces are released at https://huggingface.co/datasets/youarethewon/medjudgerag.

[1]Department of Industrial Data Engineering, Hanyang University, Seoul, South Korea [2]School of Interdisciplinary Industrial Studies, Hanyang University, Seoul, South Korea. Correspondence to: Young-Min Kim <yngmnkim@hanyang.ac.kr>.

The Workshop on Graph Foundation Models at the 43rd International Conference on Machine Learning (ICML 2026), Seoul, South Korea. Copyright 2026 by the author(s).

1. Introduction

Medical question answering (QA) has been studied across a wide range of evaluation settings, including medical examinations and biomedical research (Singhal et al., 2025). Recent benchmarks that demand expert-level medical knowledge and advanced reasoning have further expanded medical QA toward assessing domain knowledge-grounded reasoning capabilities (Zuo et al., 2025). Medical multiple-choice question answering (MCQA) benchmarks likewise include clinical vignette-based questions, with answer options designed to require fine-grained medical understanding (Jin et al., 2021; Pal et al., 2022). In such settings, simple fact recall is insufficient to distinguish the correct answer. Instead, acquiring additional domain knowledge or providing useful external evidence for reasoning becomes essential.

Pre-trained language models (LMs) rely primarily on parametric knowledge acquired during training, which limits their performance on tasks that require domain-specific reasoning. To address this limitation, Retrieval-Augmented Generation (RAG) has been proposed (Lewis et al., 2020; Guu et al., 2020). RAG retrieves relevant documents from an external corpus and provides them as input to the LM. This supplements the model's limited domain knowledge and has demonstrated potential for improving performance on domain-specific QA tasks. However, when applying RAG to medical MCQA, vanilla RAG, which simply prepends retrieved documents to the input, still exhibits notable limitations. When the evidence for the correct answer is scattered across multiple documents, the model struggles to structurally connect these fragments. Moreover, it lacks a mechanism to systematically judge whether the retrieved information constitutes sufficient evidence for each option. Finally, it lacks an explicit intermediate decision process to bridge retrieved evidence and final answer selection. As a result, these limitations can cause the reasoning accuracy of retrieval-augmented LMs to fall below that of LMs relying solely on parametric knowledge. To overcome these challenges, the LM must be able to structure scattered evidence, explicitly judge whether sufficient evidence exists for each option, and determine how to utilize retrieved information based on these judgments.

In this work, we introduce MedJudgeRAG, a novel frame-

work in which the LM learns to leverage a dynamic knowledge graph for option-wise evidence judgment to reason toward the final answer. MedJudgeRAG trains the LM through supervised fine-tuning (SFT) to acquire two capabilities. In Step 1, the LM learns to extract key entities and relations from retrieved documents that are discriminative for the given question and options. This structures the external knowledge into a dynamic KG. In Step 2, the LM learns to identify option-wise evidence based on the retrieved documents and the KG. It then judges whether this evidence supports each option. Based on these judgments, the model determines how to leverage external knowledge for answer reasoning . Specifically, when sufficient evidence supports a particular option, the model reasons directly from that evidence. When evidence only contradicts certain options, the model applies elimination to narrow down candidates. When the evidence is insufficient for all options, the model reverts to parametric knowledge. To train these capabilities, we leverage structured reasoning traces generated by a teacher LM as SFT data. The fine-tuned model takes a question, options, and retrieved documents as input. It then performs dynamic KG construction, option-wise evidence judgment, knowledge utilization strategy selection, and final answer prediction in an end-to-end manner.

We evaluate MedJudgeRAG on the MedQA and MedMCQA benchmarks (Jin et al., 2021; Pal et al., 2022), with parametric inference and vanilla RAG as baselines under the same backbone and identical retrieval results. A notable finding is that MedJudgeRAG mitigates the known problem of performance degradation caused by indiscriminately utilizing retrieved documents. This improvement does not stem from learning additional domain knowledge. Instead, the model learns dynamic knowledge construction and a reasoning strategy tailored to multiple-choice questions. Furthermore, through ablation analysis, we find that the dynamic KG functions primarily as structural training-time supervision rather than a required inference-time output. Accordingly, we explore both explicit and implicit KG decoding and analyze the trade-off between them. Our main contributions are as follows:

- We propose MedJudgeRAG, a framework that structures retrieved documents into a dynamic KG and performs MCQA-focused evidence reasoning. Experiments on two medical MCQA benchmarks demonstrate that MedJudgeRAG overcomes the limitations of vanilla RAG under the same retrieval conditions.
- Through ablation analysis, we reveal that the dynamic KG contributes primarily as graph-conditioned supervision at training time rather than as an explicit output at inference time. We further analyze the trade-off between explicit and implicit KG generation, showing that each mode has complementary strengths.

2. Related Work

2.1. Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) can supplement the domain knowledge deficiency of generator models by augmenting the prompt of pre-trained models with retrieved documents (Lewis et al., 2020; Guu et al., 2020). Recently, advanced RAG frameworks have been proposed. These frameworks enable the generator LM to discriminately accept retrieved documents. Jiang et al. (2023b) proposed a framework that determines when and what to retrieve during generation through an active retrieval strategy that triggers retrieval only when the LM generates low-probability tokens. Lin et al. (2023) introduced a two-stage fine-tuning approach that enables the retrieval-augmented LM to better utilize retrieved information while aligning the retriever with the LM's preferences to yield more contextually relevant results. Asai et al. (2023) employed special tokens to determine whether external knowledge retrieval is necessary and to self-criticize the LM's outputs, thereby selecting the optimal generation. Wei et al. (2024) demonstrated that LMs can explicitly remove noise from retrieved documents by generating denoising rationales. Our work discriminately incorporates retrieved documents by leveraging a dynamic KG to guide option-wise evidence judgment.

2.2. Knowledge Graphs in Retrieval-Augmented Generation

A knowledge graph (KG) is a structured representation of facts composed of entities, relationships, and semantic descriptions (Ji et al., 2021). Recently, KGs have been adopted in RAG as a means of structuring external knowledge. Related work can be categorized into approaches that structure external knowledge as a graph before inference and those that do so at inference time. Methods in the former category convert a corpus or multiple documents into a pre-structured knowledge source. Wang et al. (2024) construct a KG from passages using nodes that represent passages across multiple documents and edges that represent connections between nodes based on similarity and lexical overlap. GraphRAG leverages LLMs to construct a KG from a large text corpus (Edge et al., 2024). HippoRAG transforms a corpus into a schemaless knowledge graph (Gutiérrez et al., 2024). Methods in the latter category convert retrieved evidence into a KG tailored to the current question. REANO proposes a module that generates a KG composed of a set of knowledge triples from retrieved passages (Fang et al., 2024). RAS constructs a question-specific KG through iterative retrieval (Jiang et al., 2025). In the medical domain, MedGraphRAG proposes a knowledge graph construction method based on triple graph construction along with an efficient retrieval method called U-Retrieval (Wu et al., 2025). However, since MedGraphRAG pre-constructs the graph

and retrieves from this structured knowledge, its KG construction and utilization approach differs from that of our work. TAdaRAG constructs an on-the-fly KG using prompt templates tailored to the question domain, but represents only entities explicitly (Zhang et al., 2026). In contrast, our KG explicitly represents both entities and relationships, and is leveraged as training data.

2.3. LLMs for Medical Question Answering

Medical question answering often requires selecting the correct answer from multiple candidates, typically in a multiple-choice or selection-based format (Jin et al., 2019; Hendrycks et al., 2021). To improve LLM performance on medical QA, prior work has primarily pursued two directions, namely training additional medical domain knowledge or leveraging external knowledge through RAG. A representative approach for the former is adapting general-purpose LLMs to medical corpora or clinical data (Labrak et al., 2024; Chen et al., 2023; Kim et al., 2025; Christophe et al., 2024). For the latter direction, RAG^2 focuses on the retriever side of medical QA and introduces rationale-guided filtering to mitigate noisy retrieval and poorly targeted queries (Sohn et al., 2025). Self-BioRAG is a framework designed for reliable biomedical text processing that generates explanations, retrieves domain-specific documents, and self-reflects on generated responses (Jeong et al., 2024). However, these medical RAG methods presuppose adaptive retrieval or modifications to the retrieval pipeline. Our work aims to better utilize retrieved external knowledge within a fixed retrieval setting by explicitly training the model on the process of judging an evidence verdict for each option.

3. Method

We propose MedJudgeRAG for medical MCQA. MedJudgeRAG represents retrieved documents as a dynamic knowledge graph. The model judges an evidence verdict for each option and reasons toward the correct answer. Figure 1 illustrates the overall pipeline of MedJudgeRAG. Our end-to-end training first enables the LM $\mathcal{M}$ to generate a dynamic knowledge graph from the question, options, and retrieved documents. The model then generates an `<ANALYSIS>` delimiter to initiate option-wise evidence analysis. After judging whether each option is supported by the external knowledge or the KG, the model determines its reasoning strategy based on the judgment results.

3.1. Problem Formulation

LM $\mathcal{M}_\theta$ has access to an annotated dataset for the Multiple Choice Question Answering task $\mathcal{T} = \{\langle q, O, a\rangle\}$, where q is a question, $O = \{o_1, \ldots, o_n\}$ is an option set, and a is the answer. In the standard RAG setting, the model additionally has access to an external knowledge base via a retriever $\mathcal{R}$. Given a question q, the retriever $\mathcal{R}$ returns a set of relevant documents $D = \{d_1, \ldots, d_k\}$ from the external knowledge base. The model selects the most appropriate o_i from the option set O for the given question q based on D and its own parametric knowledge, denoted as $p_\theta(a|q, O, D)$.

MedJudgeRAG trains $\mathcal{M}_\theta$ to sequentially generate a single generation sequence $y = [g; r]$ from $x = (q, O, D)$. Here, g is a dynamic knowledge graph extracted from D based on q and O, consisting of entity tuples and relation tuples. The MCQA-focused reasoning sequence r denotes the sequential generation of $[j; c; a]$. j denotes the evidence judgment, which assigns a verdict $v \in \{\text{supported}, \text{contradicted}, \text{insufficient}\}$ for each option. The decision $c \in \{\text{grounded}, \text{elimination}, \text{parametric}\}$, which determines the knowledge utilization strategy, is conditioned on j. The final answer a is then produced according to c.

3.2. Dynamic Knowledge Graph Construction

Retrieved documents can be noisy, and evidence for the correct answer is often scattered across multiple documents. When such documents are directly fed to the LM, the model may be distracted by irrelevant information or fail to connect dispersed evidence. To address this, we employ a predesigned extraction template to extract only the information relevant to the question and options from the retrieved documents and structure it as a dynamic knowledge graph.

KG Schema and Format. The KG in MedJudgeRAG consists of two types of elements, entities and relations, each represented as a structured tuple:

- Entity tuple: (`"Entity"`, Name, Type, Description, Evidence)
- Relation tuple: (`"Relation"`, Source, RelationType, Target, Description, Evidence)

Entity types follow the 15 Unified Medical Language System (UMLS)[1] Semantic Groups, while relation types are adapted from UMLS relations with task-specific adjustments. These schema constraints serve to reduce redundancy and improve entity relevance (Zhang et al., 2026). The Evidence field lists the source document indices for each element (e.g., [1] or [1, 2]).

KG Data Generation. To construct training data, we provide the teacher LM $\mathcal{M}_\phi$, GPT-5.1 (OpenAI, 2025), with (q, O, D). We then prompt it to generate a KG following the KG Generation Principles specified in the instruction, along with a one-shot example.

- **Discriminative extraction.** Rather than extracting all medical concepts appearing in the retrieved documents, only information that contributes to distinguishing among the four options is selectively extracted.
- **Document-grounded.** All entity names and descrip-

[1] https://www.nlm.nih.gov/research/umls

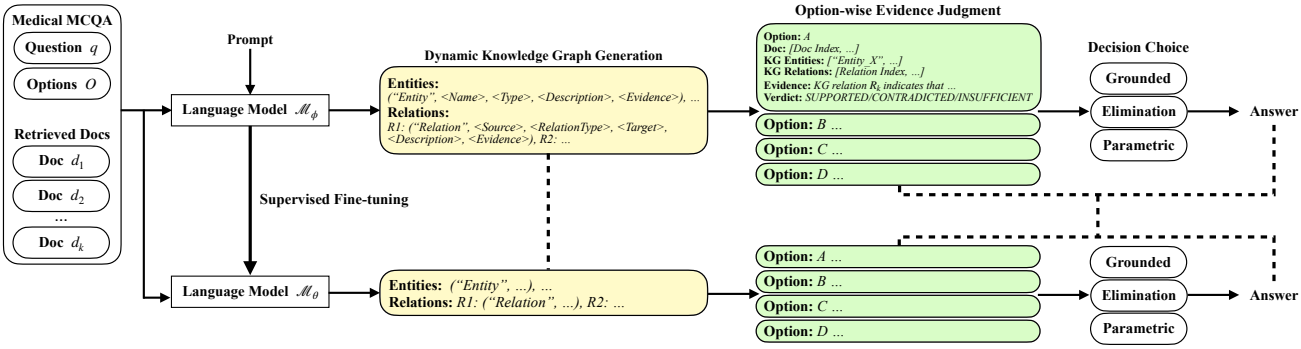


Figure 1. Overview of the MedJudgeRAG framework. **Top**: the teacher LM $\mathcal{M}_\phi$ (GPT-5.1) generates structured reasoning traces consisting of a dynamic knowledge graph, option-wise evidence judgments, and a decision. **Bottom**: the student LM $\mathcal{M}_\theta$ (Mistral/Llama) is trained via supervised fine-tuning on the teacher's outputs and produces the same structured output at inference time. Dashed lines indicate that the teacher's outputs serve as SFT training targets.

tions must be grounded in the retrieved documents. This prevents the extraction of hallucinated entities that rely on the model's own parametric knowledge. Relations are extracted only when explicitly supported by the documents.

- **Evidence-traced.** All entities and relations specify their source document indices. This enables tracing the basis of verdicts in the subsequent judgment stage, strengthening evidence-grounded reasoning.
- **Empty graph allowed.** When the retrieved documents are irrelevant to the question or contain only information unhelpful for answer determination, the model may output an empty KG. This prevents amplifying noise by forcing extraction from irrelevant documents.

To ensure that the KG is generated under the same conditions as $\mathcal{M}_\theta$ at inference time, the gold answer is not included in the teacher prompt. The generated KG undergoes the following schema-aware post-processing steps before being adopted as training data: (1) normalizing entity types to the pre-defined 15 categories, (2) filtering invalid relation types, and (3) removing relations whose source or target entities do not exist among the extracted entities.

3.3. Option-wise Evidence Judgment

To address the limitations of vanilla RAG discussed in Section 1, we train the model with an explicit generation structure that judges an evidence verdict for each option, determines a reasoning strategy based on the verdict combination, and then selects the answer.

Option-wise Verdict Generation. The model is trained to first output an `<ANALYSIS>` delimiter after KG generation. `<ANALYSIS>` delimiter signals the beginning of MCQA-focused reasoning generation and is included in r. Following this delimiter, the model sequentially generates a verdict block for each option $o_i \in O$. Each verdict block describes the supporting evidence for the option by referencing relevant document indices, entity names, and relation indices. A verdict is then selected based on the written evidence. This evidence-first design suppresses unsupported verdicts by requiring the model to articulate evidence before committing to a judgment. The verdict consists of three labels: (1) **SUPPORTED**: The documents or KG provide direct evidence supporting the option as the correct answer. (2) **CONTRADICTED**: The documents or KG explicitly indicate that the option is incorrect. (3) **INSUFFICIENT**: The documents or KG alone do not provide sufficient basis to determine whether the option is correct or incorrect.

Three-Way Decision Choice. Once the option-wise verdicts are determined, the model selects one of three knowledge utilization strategies based on the combination of verdicts. This decision explicitly branches how the retrieved information is utilized. The decision consists of three labels:

- **Grounded**: When a SUPPORTED verdict exists. The model selects the SUPPORTED option as the answer based on the retrieved evidence.
- **Elimination**: When no SUPPORTED verdict exists but at least one CONTRADICTED verdict is present. The model eliminates the CONTRADICTED options as incorrect and reasons over the remaining options with the aid of parametric knowledge.
- **Parametric**: When all options receive an INSUFFICIENT verdict. The retrieved information is deemed insufficient for answer determination. The model selects the answer based on its own parametric knowledge.

3.4. Training

Teacher Data Generation. The training data consists of complete sequences $s^* = [g^*; r^*]$ generated by $\mathcal{M}_\phi$ via the OpenAI Batch API with greedy decoding (`temperature`=0), producing one trace per example without resampling. $\mathcal{M}_\phi$ receives only (q, O, D) as input; the

gold answer is excluded from the prompt and used only during post-hoc validation. Following instructions that specify the KG generation principles (Section 3.2) and the MCQA-focused reasoning output structure (Section 3.3), $\mathcal{M}_\phi$ sequentially generates a KG, option-wise evidence judgment, decision, and answer. We draw 2,750 training samples from each of the MedQA and MedMCQA training splits with a fixed random seed. We evaluate on the held-out MedQA test split and the MedMCQA validation split. The MedMCQA validation split, disjoint from its training split, serves as the test set following the MedRAG convention since the official test labels are withheld. Each generated trace undergoes two stages of validation. Format validation checks structural constraints such as tag completeness, presence of required fields, and consistency of verdict-decision combinations. Semantic validation checks rule-based consistency between evidence and verdicts, adherence to decision rules, internal consistency between the final answer and the teacher's analysis, and agreement with the gold label. We further remove samples containing hint-leakage expressions that directly reveal the answer (e.g., "the correct answer is", "obviously option A"). Of the 5,500 generated traces, 3,562 (64.8%) passed validation. After removing samples exceeding 8,192 tokens under the Mistral-7B-Instruct-v0.3 (Jiang et al., 2023a) and Meta-Llama-3-8B-Instruct (Grattafiori et al., 2024) tokenizers, 3,479 (3,148 train / 331 val) and 3,559 (3,222 train / 337 val) samples were adopted for each backbone, respectively.

Weighted Loss Fine-tuning. MedJudgeRAG is trained to generate the generation sequence $y = [g; r]$ for all instruction-target pairs (x, y) in the training set $\mathcal{D}_{\text{train}}$. However, since g targets accurate KG extraction while r targets the reasoning sequence, the two subsequences have different learning objectives. We therefore allocate more training focus to the reasoning segment and reduce the per-token gradient for the KG segment, which serves as an intermediate representation for reasoning, through loss weight adjustment. To this end, we employ a weighted cross-entropy loss that differentially weights tokens based on their position within the sequence. Let $\mathcal{T}_g, \mathcal{T}_r \subseteq \{1, \ldots, |y|\}$ denote the index sets of token positions belonging to the KG segment and the reasoning segment, respectively ($\mathcal{T}_g \cap \mathcal{T}_r = \emptyset$). Their respective counts of valid tokens are denoted by $|\mathcal{T}_g|$ and $|\mathcal{T}_r|$. Prompt and padding tokens are excluded from the loss. The boundary between the two segments is determined by <ANALYSIS> delimiter. The training objective is defined as follows:

$$\mathcal{L}(\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\frac{1}{Z}\left(\lambda_g \sum_{t\in\mathcal{T}_g} \log p_\theta(y_t \mid y_{<t}, x) + \sum_{t\in\mathcal{T}_r} \log p_\theta(y_t \mid y_{<t}, x)\right)\right] \tag{1}$$

where $Z = \lambda_g|\mathcal{T}_g| + |\mathcal{T}_r|$.

The hyperparameter $\lambda_g \in [0, 1]$ controls the loss weight for the KG segment, while the loss weight for the reasoning segment is fixed at 1. Normalizing by the weighted sum $\lambda_g|\mathcal{T}_g| + |\mathcal{T}_r|$ fixes the per-token contribution ratio of the two segments (KG : reasoning) at $\lambda_g : 1$, independent of sample length.

4. Experiments

4.1. Experimental Setup

Datasets & Metrics. We evaluate on two medical MCQA benchmarks: the test set of MedQA (1,273 questions) and the validation set of MedMCQA (4,183 questions). Performance is measured by accuracy, where the predicted answer is extracted from the model's generation via regular expression parsing.

Baselines. All comparisons are conducted under the same backbone and the same retrieval results, varying only how the retrieved information is utilized. Parametric provides only the question and options without retrieved documents. Vanilla RAG appends the top-5 retrieved documents to the prompt via simple concatenation. Both Parametric and Vanilla RAG follow the chain-of-thought prompt from Xiong et al. (2024). The maximum generated output length is set to 1,024 tokens for Parametric and 8,192 tokens for Vanilla RAG to accommodate our experimental setting.

Implementation Details. To ensure a fair comparison, vanilla RAG and MedJudgeRAG use identical retrieval results. The retrieval corpus consists of PubMed[2] for biomedical abstracts and medical textbooks for domain-specific knowledge (Jin et al., 2021). The retriever is Contriever (Izacard et al., 2021), retrieving the top-5 documents per question. At inference time, $\mathcal{M}_\theta$ receives (q, O, D) as input and operates in one of two decoding modes. In **Explicit decoding**, the model outputs KG, option-wise evidence judgment, decision, and answer in a single generation step. In **Implicit decoding**, KG generation is skipped and only option-wise evidence judgment, decision, and answer are generated. The comparison between Explicit and Implicit decoding is presented in Section 5 to isolate the effect of KG generation. The maximum generated output length is 8,192 tokens, and we use vLLM to accelerate inference (Kwon et al., 2023). We adopt Mistral-7B-Instruct-v0.3 (hereafter Mistral) and Meta-Llama-3-8B-Instruct (hereafter Llama) as backbones. Training is performed via QLoRA-based SFT with ZeRO-2 (Dettmers et al., 2023). The maximum input sequence length is 8,192 tokens, per-GPU batch size is 1, and gradient accumulation is set to 16 steps. We train for 3 epochs with bfloat16 mixed precision, an initial learning rate

[2] https://pubmed.ncbi.nlm.nih.gov/

Table 1. Accuracy on MedQA and MedMCQA. Mistral denotes Mistral-7B-Instruct-v0.3; Llama denotes Meta-Llama-3-8B-Instruct. All *Ours* results use $\lambda_g = 0.0$. Explicit decoding generates the KG before reasoning; Implicit decoding skips KG generation. Underline denotes the best result per backbone.

Backbone	Method	MedQA	MedMCQA	Avg
Mistral	Parametric	52.55	46.26	49.41
	Vanilla RAG	50.59	41.57	46.08
	Ours (Explicit)	59.54	50.37	54.96
	Ours (Implicit)	60.88	50.16	55.52
Llama	Parametric	60.64	54.86	57.75
	Vanilla RAG	49.88	45.61	47.75
	Ours (Explicit)	63.94	55.39	59.67
	Ours (Implicit)	66.93	57.33	62.13

of 1e-4, AdamW optimizer, and cosine scheduler. We experiment with KG loss weight $\lambda_g \in \{0.0, 0.1, 0.3, 0.5, 1.0\}$ and select the best-performing λ_g based on average accuracy. Training takes 10–12 hours on 2 NVIDIA GeForce RTX 3090 (24 GB) GPUs.

4.2. Results

Main Results. Table 1 reports the accuracy of Parametric, Vanilla RAG, and MedJudgeRAG (Explicit/Implicit) for both backbones. The primary comparison is Vanilla RAG vs. Ours, measuring the effectiveness of MedJudgeRAG over indiscriminately utilizing retrieved information. Additionally, we compare Ours (Explicit) and Ours (Implicit) to analyze how explicitly generating the KG affects the reasoning chain. As shown in Table 1, Vanilla RAG degrades accuracy relative to Parametric on both backbones: by 2.0 percentage points (pp) and 4.7pp on MedQA and MedMCQA for Mistral, and by 10.8pp and 9.3pp for Llama. This confirms that simply prepending retrieved documents can degrade performance for these 7-8B-scale backbones. However, MedJudgeRAG overcomes this limitation using the same retrieved documents, achieving improvements over Vanilla RAG of up to +10.3pp (MedQA) and +8.8pp (MedMCQA) for Mistral, and +17.1pp (MedQA) and +11.7pp (MedMCQA) for Llama. Furthermore, MedJudgeRAG consistently improves over Parametric, demonstrating that it fulfills the original goal of RAG, enhancing LM performance by incorporating retrieved documents.

Explicit vs. Implicit Decoding. We compare Explicit decoding, which outputs the KG before reasoning, with Implicit decoding, which skips KG generation. For Mistral, Implicit decoding achieves higher accuracy on MedQA while Explicit decoding slightly outperforms on MedMCQA, showing that the advantage varies by benchmark. For Llama, Implicit decoding consistently outperforms Explicit decoding on both benchmarks. We analyze the trade-off between Explicit and Implicit decoding in detail in Section 5.

Table 2. Ablation on KG loss weight per benchmark. *w/o KG* is trained without the KG segment (independent of λ_g). *Exp./Imp.* denote Explicit/Implicit decoding. **Bold**: best within each decoding for each (backbone, benchmark). Underline: best per (backbone, benchmark).

Bench.	Method	KG loss weight (λ_g)					w/o KG
		0.0	0.1	0.3	0.5	1.0	
Mistral							
MedQA	Exp.	**59.54**	53.73	54.05	57.34	55.15	55.15
	Imp.	60.88	**61.19**	60.17	59.23	60.49	
MedMCQA	Exp.	**50.37**	42.96	44.68	46.52	46.12	47.57
	Imp.	**50.16**	49.39	48.96	49.06	48.82	
Llama							
MedQA	Exp.	**63.94**	57.89	58.05	60.49	59.94	62.29
	Imp.	**66.93**	65.59	64.65	63.32	63.16	
MedMCQA	Exp.	**55.39**	53.14	53.77	53.65	52.09	55.87
	Imp.	**57.33**	56.13	56.01	55.77	54.82	

5. Ablation Study

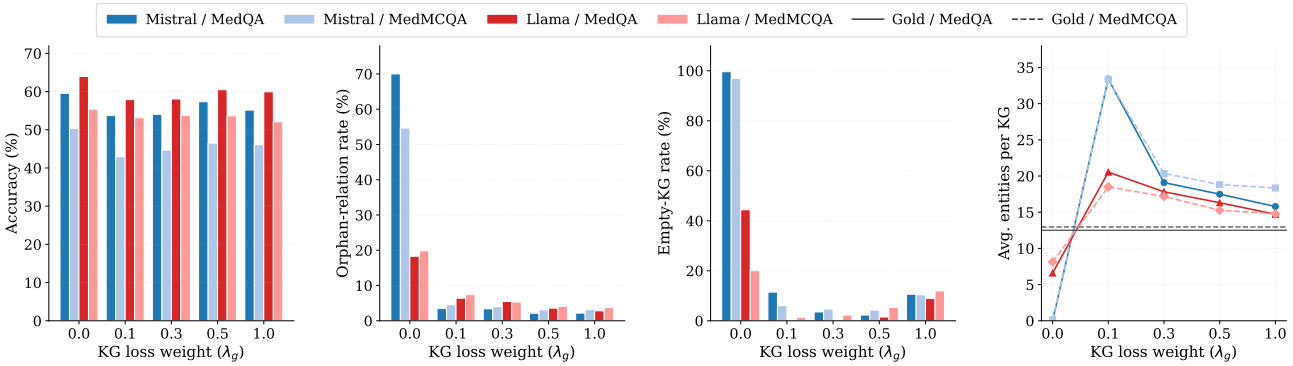
This section examines performance changes when KG generation is directly optimized ($\lambda_g > 0$) versus not ($\lambda_g = 0$). We also compare **Explicit** and **Implicit** decoding. Through these analyses, we show that the KG serves more effectively as graph-conditioned supervision at training time than as an explicit output at inference time. Furthermore, while KG learning intensity can be controlled via λ_g, improving KG surface-form quality does not directly translate to improved answer accuracy.

5.1. Effect of KG Loss Weight

Table 2 reports accuracy on the MedQA and MedMCQA benchmarks for each λ_g setting. In Explicit decoding, both Mistral and Llama achieve the highest average accuracy at $\lambda_g = 0.0$. In Implicit decoding, Mistral achieves the best performance at $\lambda_g = 0.1$ on MedQA and at $\lambda_g = 0.0$ on MedMCQA, while Llama achieves the best at $\lambda_g = 0.0$. This demonstrates that directly optimizing the model to reconstruct KG tokens does not necessarily lead to higher answer accuracy. However, $\lambda_g = 0$ does not mean that the KG is entirely unlearned. At $\lambda_g = 0$, although there is no direct training signal for KG tokens, the model learns to predict the analysis conditioned on the gold KG given as a prefix during training. In other words, the model indirectly learns to reason conditioned on the KG prefix. The non-monotonic relationship between λ_g and performance can be attributed to two factors: (1) Objective mismatch: KG surface-form reconstruction accuracy and final answer accuracy are not the same objective. As λ_g increases, more training focus is allocated to KG reconstruction at the expense of the answer objective. (2) Excessive output length in Explicit decoding: as λ_g increases, the model is trained to generate longer and more elaborate KGs. This causes transi-

Table 3. Comparison between Explicit and Implicit decoding across 5 λ_g settings. Imp. wins(Implicit decoding wins): number of λ_g settings where Implicit outperforms Explicit. Δ Acc: average accuracy gap (Implicit − Explicit) in percentage points. Invalid: average number of samples where the model failed to select an answer candidate. Exclusive correct: average number of questions correctly answered by one decoding but not the other.

		Accuracy		Invalid		Exclusive correct	
Backbone	**Bench.**	**Imp. wins**	**Δ Acc**	**Exp.**	**Imp.**	**Exp.**	**Imp.**
Mistral	MedQA	5/5	+4.43	36.4	0.4	143.6	200.0
	MedMCQA	4/5	+3.14	148.2	4.2	479.4	611.0
Llama	MedQA	5/5	+4.67	31.6	0.0	125.2	184.6
	MedMCQA	5/5	+2.41	75.0	1.6	427.8	528.4



(a) Answer Accuracy (b) Orphan-relation Rate (c) Empty-KG Rate (d) Average Entity Count

Figure 2. Effect of KG loss weight (λ_g) on KG quality and answer accuracy under Explicit decoding. All results are reported across 4 (backbone × benchmark) combinations and 5 λ_g values (0.0, 0.1, 0.3, 0.5, 1.0).

tion failures to the reasoning segment after <ANALYSIS> delimiter. It also encroaches on the token budget within the maximum output sequence length. Table 3 summarizes the aggregated comparison between Explicit and Implicit decoding across all λ_g settings. The fact that the average Invalid count is substantially higher for Explicit decoding than Implicit decoding in Table 3 supports this interpretation. Detailed output lengths for Invalid samples are provided in Appendix A.

5.2. Effect of Graph-Conditioned Supervision

To assess the practical contribution of graph-conditioned supervision, we compare against a model trained on data from which all KG-related signals have been removed. This control model (w/o KG) reduces the fine-tuning target to $y_{\text{ctrl}} = [r_{\text{ctrl}}]$. The reduction is achieved by removing the KG segment g from y and removing all KG references from the reasoning sequence r (e.g., entity/relation fields, relation identifiers, and KG vocabulary). The retrieval and reasoning chain are kept identical to isolate the contribution of graph-conditioned supervision. This control model still outperforms Vanilla RAG, demonstrating that the MCQA-focused reasoning structure alone can improve performance. Comparing against the graph-conditioned model, we find that it outperforms w/o KG in 18 out of 20 Implicit decoding settings, with only 2 settings (MedMCQA at $\lambda_g = 0.5$ and $\lambda_g = 1.0$) showing small reversals. Therefore, the role of the graph in our framework is better interpreted as a structural element that provides graph-conditioned supervision at training time, rather than a component that must be explicitly output at inference time. Empirically, the balance point with the downstream answer objective is observed in the $\lambda_g \approx 0 - 0.1$ range.

5.3. Explicit vs. Implicit KG Decoding

Table 3 shows that Implicit decoding achieves higher accuracy than Explicit decoding in 19 out of 20 settings. The advantage reflects a trade-off rather than strict inferiority of Explicit decoding. Explicit decoding correctly answers some questions that Implicit decoding misses, but it also misses more questions that Implicit decoding correctly answers. In our experimental setting, the regressions outweigh the improvements. We therefore adopt Implicit decoding as the main result in Table 1.

5.4. KG Quality Analysis with Respect to λ_g

All analyses in this subsection are based on output sequences from Explicit decoding.

KG quality vs. answer accuracy. Figure 2 disentangles the effects of λ_g on the structural quality of the KG and on answer accuracy. Figure 2b shows that the orphan-relation ratio generally decreases as λ_g increases across all four settings, confirming that λ_g operates as intended, i.e., as a hyperparameter controlling KG structural consistency. However, Figure 2a shows that improvements in KG structural quality do not translate to higher answer accuracy. Instead, $\lambda_g = 0.0$ achieves the highest accuracy across all four settings. This indicates that precisely optimizing KG surface-form quality at inference time does not directly lead to improved accuracy.

Backbone-specific mechanisms. Figure 2c and 2d show that at $\lambda_g = 0.0$, Llama and Mistral exhibit qualitatively different KG generation behaviors. Llama still generates short KGs with an average of 6–8 entities, while Mistral largely ceases generating KGs altogether. Both models achieve accuracy improvements but through different pathways. Llama proceeds to analysis with its self-generated noisy KG prefix intact, whereas Mistral effectively converges to Implicit decoding by skipping the KG stage.

Effect of λ_g on KG length. Figure 2d shows that the average number of KG entities peaks at $\lambda_g = 0.1$ and generally decreases as λ_g increases. Under weak supervision, the model tends to over-generate, but as supervision strengthens, token-level accuracy becomes a direct training signal, causing the model to generate entity counts closer to the gold average. This suggests that the mechanism of generating unnecessarily long KGs (Section 5.1) can be mitigated as λ_g increases. Nevertheless, the lack of accompanying accuracy improvement reaffirms that surface-level accuracy of KG output does not directly translate to answer accuracy.

6. Conclusion

In medical MCQA, vanilla RAG can degrade LM performance by indiscriminately utilizing retrieved documents. To address this, we proposed MedJudgeRAG, a framework that represents retrieved documents as a dynamic KG and judges an evidence verdict for each option. Based on the verdict combination, the model determines a knowledge utilization strategy to reason toward the final answer. Experimental results on MedQA and MedMCQA demonstrate that MedJudgeRAG achieves consistent improvements over vanilla RAG under the same backbone and identical retrieval results. It also improves over parametric inference, fulfilling the original goal of RAG. However, Explicit decoding, which generates the KG at inference time, exhibits lower accuracy than Implicit decoding, which omits KG generation. Ablation analysis suggests that this is because the dynamic KG contributes primarily as graph-conditioned supervision at training time rather than as an explicit output at inference time. Notably, this numerical gap does not imply that Explicit decoding is strictly inferior. Each decoding correctly answers some questions that the other misses, but Implicit decoding does so for more questions on average. Future work includes optimizing the relationship between KG generation quality and answer accuracy via methods such as reinforcement learning. Another promising direction is exploring complementary utilization of Explicit and Implicit decoding. Validating MedJudgeRAG on additional medical MCQA benchmarks to assess broader generalization is also worth pursuing. Finally, extending the framework to non-medical domains remains an important direction.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2026-25492127).

References

Asai, A., Wu, Z., Wang, Y., Sil, A., and Hajishirzi, H. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2023.

Chen, Z., Cano, A. H., Romanou, A., Bonnet, A., Matoba, K., Salvi, F., Pagliardini, M., Fan, S., Köpf, A., Mohtashami, A., et al. Meditron-70b: Scaling medical pretraining for large language models. *arXiv preprint arXiv:2311.16079*, 2023.

Christophe, C., Kanithi, P. K., Raha, T., Khan, S., and Pimentel, M. A. Med42-v2: A suite of clinical llms. *arXiv preprint arXiv:2408.06142*, 2024.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R. O., and Larson, J. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

Fang, J., Meng, Z., and Macdonald, C. Reano: Optimising retrieval-augmented reader models through knowledge graph generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2094–2112, 2024.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Gutiérrez, B. J., Shu, Y., Gu, Y., Yasunaga, M., and Su, Y. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in neural information processing systems*, 37:59532–59569, 2024.

Guu, K., Lee, K., Tung, Z., Pasupat, P., and Chang, M. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., and Grave, E. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.

Jeong, M., Sohn, J., Sung, M., and Kang, J. Improving medical reasoning through retrieval and self-reflection with retrieval-augmented large language models. *Bioinformatics*, 40(Supplement_1):i119–i129, 2024.

Ji, S., Pan, S., Cambria, E., Marttinen, P., and Yu, P. S. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514, 2021.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b, 2023a. URL https://arxiv.org/abs/2310.06825.

Jiang, P., Cao, L., Zhu, R., Jiang, M., Zhang, Y., Shen, J., Sun, J., and Han, J. Ras: Retrieval-and-structuring for knowledge-intensive llm generation. *arXiv preprint arXiv:2502.10996*, 2025.

Jiang, Z., Xu, F. F., Gao, L., Sun, Z., Liu, Q., Dwivedi-Yu, J., Yang, Y., Callan, J., and Neubig, G. Active retrieval augmented generation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 7969–7992, 2023b.

Jin, D., Pan, E., Oufattole, N., Weng, W.-H., Fang, H., and Szolovits, P. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

Jin, Q., Dhingra, B., Liu, Z., Cohen, W., and Lu, X. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 2567–2577, 2019.

Kim, H., Hwang, H., Lee, J., Park, S., Kim, D., Lee, T., Yoon, C., Sohn, J., Park, J., Reykhart, O., et al. Small language models learn enhanced reasoning skills from medical textbooks. *NPJ digital medicine*, 8(1):240, 2025.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Labrak, Y., Bazoge, A., Morin, E., Gourraud, P.-A., Rouvier, M., and Dufour, R. Biomistral: A collection of open-source pretrained large language models for medical domains. In *Findings of the association for computational linguistics: acl 2024*, pp. 5848–5864, 2024.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

Lin, X. V., Chen, X., Chen, M., Shi, W., Lomeli, M., James, R., Rodriguez, P., Kahn, J., Szilvasy, G., Lewis, M., et al. Ra-dit: Retrieval-augmented dual instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2023.

OpenAI. GPT-5.1 Instant and GPT-5.1 Thinking System Card Addendum, 2025. URL https://openai.com/index/gpt-5-system-card-addendum-gpt-5-1/.

Pal, A., Umapathi, L. K., and Sankarasubbu, M. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pp. 248–260. PMLR, 2022.

Singhal, K., Tu, T., Gottweis, J., Sayres, R., Wulczyn, E., Amin, M., Hou, L., Clark, K., Pfohl, S. R., Cole-Lewis, H., et al. Toward expert-level medical question answering with large language models. *Nature medicine*, 31(3):943–950, 2025.

Sohn, J., Park, Y., Yoon, C., Park, S., Hwang, H., Sung, M., Kim, H., and Kang, J. 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)*, pp. 12739–12753, 2025.

Wang, Y., Lipka, N., Rossi, R. A., Siu, A., Zhang, R., and Derr, T. Knowledge graph prompting for multi-document

question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 19206–19214, 2024.

Wei, Z., Chen, W.-L., and Meng, Y. Instructrag: Instructing retrieval-augmented generation via self-synthesized rationales. *arXiv preprint arXiv:2406.13629*, 2024.

Wu, J., Zhu, J., Qi, Y., Chen, J., Xu, M., Menolascina, F., Jin, Y., and Grau, V. Medical graph rag: Evidence-based medical large language model via graph retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28443–28467, 2025.

Xiong, G., Jin, Q., Lu, Z., and Zhang, A. Benchmarking retrieval-augmented generation for medicine. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6233–6251, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.372. URL https://aclanthology.org/2024.findings-acl.372/.

Zhang, J., Tang, B., Shao, W., Wei, W., Zhao, J., Zhu, J., Li, Z., Xi, W., Lin, Z., Xiong, F., et al. Tadarag: Task adaptive retrieval-augmented generation via on-the-fly knowledge graph construction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34728–34736, 2026.

Zuo, Y., Qu, S., Li, Y., Chen, Z., Zhu, X., Hua, E., Zhang, K., Ding, N., and Zhou, B. Medxpertqa: Benchmarking expert-level medical reasoning and understanding. *arXiv preprint arXiv:2501.18362*, 2025.

Table 4. Analysis of Explicit decoding failures across backbones, benchmarks, and KG loss weights. # Invalid denotes outputs that do not select an answer candidate (A/B/C/D). For Invalid samples, we report the average raw output length (in characters), the average number of entity and relation tuples, and the proportion that never generated the <ANALYSIS> delimiter. *n/a* denotes settings with no invalid samples, for which invalid-sample statistics are undefined.

					Statistics of Invalid samples			
Backbone	**Bench.**	λ_g	**Acc**	**# Invalid**	**Avg. length**	**Avg. ent.**	**Avg. rel.**	**No <ANALYSIS> (%)**
Mistral	MedQA	0.0	59.54%	2	2056	0.12	0.03	0.0%
		0.1	53.73%	121	8529	33.43	14.30	99.2%
		0.3	54.05%	36	6965	19.27	13.20	91.7%
		0.5	57.34%	10	6335	17.49	11.93	90.0%
		1.0	55.15%	13	6037	16.27	11.58	100.0%
		w/o KG	55.15%	1	2634	0.00	0.00	100.0%
	MedMCQA	0.0	50.37%	8	1784	0.24	0.20	12.5%
		0.1	42.96%	431	9724	33.36	18.49	98.6%
		0.3	44.68%	153	7843	20.62	15.39	97.4%
		0.5	46.52%	86	7205	18.83	14.10	94.2%
		1.0	46.12%	63	7260	18.76	14.94	95.2%
		w/o KG	47.57%	2	1937	0.00	0.00	100.0%
Llama	MedQA	0.0	63.94%	14	2579	6.60	2.96	100.0%
		0.1	57.89%	70	6430	20.55	12.50	95.7%
		0.3	58.05%	39	6117	17.82	12.85	87.2%
		0.5	60.49%	20	5796	16.32	11.86	90.0%
		1.0	59.94%	15	5533	14.71	10.87	93.3%
		w/o KG	62.29%	0	n/a	n/a	n/a	n/a
	MedMCQA	0.0	55.39%	77	2851	8.15	3.79	77.9%
		0.1	53.14%	121	6698	18.50	13.87	90.9%
		0.3	53.77%	86	6531	17.19	13.95	84.9%
		0.5	53.65%	29	6048	15.28	12.45	96.6%
		1.0	52.09%	62	6005	14.77	11.88	96.8%
		w/o KG	55.87%	5	1280	0.00	0.00	100.0%

Table 5. Average decision rule adherence rates (%) across all backbone, benchmark, and λ_g combinations. Decision-verdict consistency evaluates the full three-way rule, while Grounded, Elimination, and Parametric match report type-specific adherence among samples assigned to each decision type.

Decoding	**Decision-verdict consistency**	**Grounded match**	**Elimination match**	**Parametric match**
Explicit	99.88	99.91	99.11	99.95
Implicit	99.92	100.00	97.93	99.99
Implicit (w/o KG)	99.72	99.85	98.81	99.80

A. Additional Analysis of Explicit Generation Failures

As shown in Table 4, Invalid samples tend to have substantially longer outputs than valid samples. The dominant failure mode is the absence of the <ANALYSIS> delimiter, where the model fails to transition from KG generation to the reasoning phase and instead continues generating graph content.

B. Decision Rule Adherence

Table 5 reports the average adherence rates of the three-way decision rules (Section 3.3) across all backbone, benchmark, and λ_g combinations. Decision-verdict consistency measures whether the chosen decision (grounded, elimination, or parametric) is consistent with the verdict combination. Grounded match measures whether a SUPPORTED verdict exists among the four options when the grounded decision is selected. Elimination match measures whether the selected elimination decision is accompanied by no SUPPORTED verdict and at least one CONTRADICTED verdict. Parametric match measures whether all four options receive INSUFFICIENT when the parametric decision is selected. All settings achieve over 99.7%

consistency, confirming that the fine-tuned models faithfully follow the decision rules. This suggests that performance differences across settings are attributable to verdict quality and generation completion rather than decision rule violations.

C. Prompts

This appendix lists the verbatim prompts used in our experiments. Placeholders {question}, {option_a}, {option_b}, {option_c}, {option_d}, and {documents} are filled from each MCQA instance, and the resulting (system, user) message pair is rendered through the backbone's chat template before being fed to the model.

C.1. MedJudgeRAG (Training and Explicit Inference)

The same prompt is used to (i) elicit supervised fine-tuning targets from GPT-5.1, (ii) compute the per-token weighted cross-entropy training loss in Eq. (1), and (iii) generate predictions in Explicit decoding at inference time, ensuring full prompt consistency between training and evaluation.

System Message

```
You are a biomedical information extractor and medical reasoner.
```

User Message Template

```
-Task Overview-
Do two steps in order:
1) Build a question-focused Knowledge Graph from the Documents.
2) After the KG, analyze each option and choose the final answer.

====================
[Step 1: KG Rules]
====================
-Role-
You are a biomedical knowledge graph constructor. Extract a question-focused Dynamic Knowledge Graph (KG)
     from retrieved documents.

-Goal-
Extract entities and relations that help DISCRIMINATE among the given options.

-Entity constraints-
- Entity types: Activities & Behaviors | Anatomy | Chemicals & Drugs | Concepts & Ideas | Devices |
     Disorders | Genes & Molecular Sequences | Geographic Areas | Living Beings | Objects | Occupations |
     Organizations | Phenomena | Physiology | Procedures
- Entity Name MUST appear verbatim in the Documents (abbreviation <-> full-form normalization is allowed
     ONLY when BOTH forms appear in the Documents).
- Prefer higher-level entities over their enumerated sub-items; if a collective concept already captures
     individual items, do not list them separately.
- Do NOT create entities directly from the answer Options. Entities must originate from the Documents. (
     Option text may help you recognize relevant concepts in Documents, but the Options themselves are NOT
     a source.)
- Entity Description: paraphrase from Documents only; no external knowledge.
- Entity Evidence: document id(s) only, e.g. [1] or [1, 3].

-Relation constraints-
- Relation types: part_of | located_in | connected_to | adjacent_to | performs | uses | affects | causes |
     result_of | indicates | measures | diagnoses | manifestation_of | precedes | co_occurs_with
- If a document explicitly states a negative relation and that negation is important for discriminating
     options, encode the negation in the Relation Description by starting it with "[NEGATED]". Use "[
     NEGATED]" only when the source document contains explicit negation cues (e.g., "no", "not", "did not",
      "without", "failed to", "absence of").
- Use only entity names from the extracted Entities list.
- Extract a relation only when explicitly or conservatively supported by Documents.
- Prioritize relations that directly discriminate among the given Options.
- Relation Source and Target MUST be copied exactly from the Entity Name field (character-for-character).
     If a needed endpoint is not already in the Entities list, prefer omitting the relation rather than
     adding a new low-relevance entity.
- Use "indicates" only when the document explicitly links X as a diagnostic clue or criterion for Y.
     Otherwise prefer a more conservative relation type.
- Use co_occurs_with only when no mechanistic relation applies. Prefer causal/mechanistic types (causes,
     result_of, indicates, manifestation_of) over co_occurs_with.
- When a document describes a causal or temporal chain (A -> B -> C), preserve intermediate steps as
     separate relations rather than collapsing into a single A -> C link.
```

```
- Relation Evidence: document id(s) only, e.g. [1] or [1, 2].

-Global Rules-
- Do NOT answer the Question or choose an option.
- Use ONLY document-grounded facts; no external knowledge.
- Keep the graph concise: high relevance over exhaustive coverage.
- Not all documents are equally relevant. Focus on documents that directly address the question's core
    claim; skip tangential ones.
- Prioritize pathognomonic findings, key differentiating features, and diagnostic criteria that help
    distinguish among the given Options.
- All Evidence must cite numbered documents [1]-[N] only. Never use [Question] or [Options] as an evidence
    source.
- If no document contains information relevant to the Question, output an empty graph.

-Output format (strict format, no JSON, no markdown)-
Entities:
("Entity", <Name>, <Type>, <Description>, <Evidence>)

Relations:
R1: ("Relation", <Source>, <RelationType>, <Target>, <Description>, <Evidence>)
R2: ...

If no document is relevant, output only the headers with no entries:
Entities:

Relations:

######################
-Example-
######################
Question: A 12-year-old girl presents to her primary care physician with left knee pain for the past 6
    weeks. She recently joined the field hockey team at her school. The pain is the most severe when she
    is running up and down the stairs at the school stadium. The pain decreases when she goes home and
    rests after practice. She additionally admits to tripping and landing on her left knee 5 days ago.
    Physical exam shows a knee with a healing abrasion over the left patella. The tibial tuberosity is
    tender to palpation. A radiograph of the knee is presented in figure A. Which of the following is the
    most likely diagnosis?
Options:
A. Osgood-Schlatter disease
B. Patellofemoral pain syndrome
C. Pes anserine bursitis
D. Tibial plateau fracture
Documents:
[1] An active 13-year-old boy has anterior knee pain. Diagnosis? The most common 1 malignant tumor of bone.
     Pseudogout. Polymyalgia rheumatica. Osgood-Schlatter disease. Distal radius (Colles' fracture).
    Avascular necrosis.
######################
Output:
Entities:
("Entity", "Osgood-Schlatter disease", "Disorders", "Anterior knee pain condition in active adolescents.",
    "[1]")
("Entity", "anterior knee pain", "Phenomena", "Symptom linked to Osgood-Schlatter disease.", "[1]")

Relations:
R1: ("Relation", "anterior knee pain", "indicates", "Osgood-Schlatter disease", "Anterior knee pain in an
    active adolescent indicates Osgood-Schlatter disease.", "[1]")
######################

====================
[Step 2: Analysis Rules]
====================
After finishing the KG, output this delimiter exactly:
<ANALYSIS>

Do not stop after KG. You must continue and output the full <ANALYSIS> section.

Then, for each option (A/B/C/D), first find relevant references, then judge:
   - Doc: list of integer document IDs relevant to this option ([] if none)
   - KG Entities: list of relevant entity names from the KG you built ([] if none), e.g. ["Entity_X"] or ["
     Entity_X", "Entity_Y"]
   - KG Relations: list of relevant Relation IDs from the KG you built ([] if none), e.g. [Ri] or [Ri, Rj]
   - Evidence: explanation referencing ONLY the Doc/KG Entities/KG Relations listed above. If all three are
      [], Evidence must be "No relevant evidence found." -- do NOT use medical knowledge here.
   - Verdict: exactly one of SUPPORTED, CONTRADICTED, or INSUFFICIENT

-Verdicts (relative to the Question)-
- SUPPORTED: Doc, KG Entities, or KG Relations provide evidence that this option correctly answers the
    Question. Requires at least one reference in Doc, KG Entities, or KG Relations.
```

```
- CONTRADICTED: Doc, KG Entities, or KG Relations provide evidence against this option as the answer.
    Requires at least one reference in Doc, KG Entities, or KG Relations.
- INSUFFICIENT: Doc, KG Entities, and KG Relations lack relevant information to judge this option. If Doc
    =[], KG Entities=[], and KG Relations=[], verdict MUST be INSUFFICIENT regardless of your medical
    knowledge.

-Decision-
Decision must be consistent with verdicts:
- "grounded" if at least one option is SUPPORTED
- "elimination" if no option is SUPPORTED but at least one is CONTRADICTED
- "parametric" only when ALL options are INSUFFICIENT
Summary: reasoning that leads to the final answer. The content depends on Decision:
- grounded: synthesize ONLY the SUPPORTED evidence to justify the answer. No parametric knowledge.
- elimination: state which options are ruled out by CONTRADICTED evidence, then use medical knowledge to
    choose among the remaining INSUFFICIENT options.
- parametric: use medical knowledge to reason through all options and justify the answer.

-Analysis output format (follow exactly)-
[A] Doc: [1, 3]
KG Entities: ["Entity_X"]
KG Relations: [Ri, Rj]
Evidence: Document [1] and KG relation Ri indicate that ...
Verdict: SUPPORTED

[B] Doc: []
KG Entities: ["Entity_Y"]
KG Relations: [Rk]
Evidence: KG relation Rk indicates that ...
Verdict: CONTRADICTED

[C] Doc: []
KG Entities: []
KG Relations: []
Evidence: No relevant evidence found.
Verdict: INSUFFICIENT

[D] Doc: []
KG Entities: []
KG Relations: []
Evidence: No relevant evidence found.
Verdict: INSUFFICIENT

Decision: grounded|elimination|parametric
Summary: ...
Answer_choice: A|B|C|D

-Order Constraint-
Always output in this order: KG block -> <ANALYSIS> -> option analysis -> Decision/Summary/Answer_choice

Question: {question}
Options:
A. {option_a}
B. {option_b}
C. {option_c}
D. {option_d}
Documents:
{documents}
Output:
```

C.2. Parametric Baseline (no documents)

Adopted from the MedRAG general chain-of-thought prompt (Xiong et al., 2024).

System Message

```
You are a helpful medical expert, and your task is to answer a multi-choice medical question. Please first
    think step-by-step and then choose the answer from the provided options. Organize your output in a
    json formatted as Dict{"step_by_step_thinking": Str(explanation), "answer_choice": Str{A/B/C/D}}. Your
     responses will be used for research purposes only, so please have a definite answer.
```

User Message Template

```
Here is the question:
{question}

Here are the potential choices:
A. {option_a}
B. {option_b}
C. {option_c}
D. {option_d}

Please think step-by-step and generate your output in json:
```

C.3. Vanilla RAG Baseline

Adopted from the MedRAG general document-augmented prompt (Xiong et al., 2024).

System Message

```
You are a helpful medical expert, and your task is to answer a multi-choice medical question using the
     relevant documents. Please first think step-by-step and then choose the answer from the provided
     options. Organize your output in a json formatted as Dict{"step_by_step_thinking": Str(explanation), "
     answer_choice": Str{A/B/C/D}}. Your responses will be used for research purposes only, so please have
     a definite answer.
```

User Message Template

```
Here are the relevant documents:
{documents}

Here is the question:
{question}

Here are the potential choices:
A. {option_a}
B. {option_b}
C. {option_c}
D. {option_d}

Please think step-by-step and generate your output in json:
```

C.4. Implicit Inference (Analysis-only)

This prompt is used for Implicit decoding inference, which removes Step 1 KG construction while preserving the per-option judgment chain (Doc/KG/Evidence/Verdict, Decision, Summary, Answer_choice) of MedJudgeRAG. KG-related fields are kept in the schema but are always expected to be empty, so the model produces only the Step 2 reasoning sequence.

System Message

```
You are a biomedical information extractor and medical reasoner.
```

User Message Template

```
====================
[Step 2: Analysis Rules]
====================
This mode skips Step 1 KG construction.
Assume no KG was generated.

Then, for each option (A/B/C/D), first find relevant references, then judge:
   - Doc: list of integer document IDs relevant to this option ([] if none)
   - KG Entities: list of relevant entity names from the KG you built ([] if none), e.g. ["Entity_X"] or ["
     Entity_X", "Entity_Y"]
   - KG Relations: list of relevant Relation IDs from the KG you built ([] if none), e.g. [Ri] or [Ri, Rj]
```

```
   - Evidence: explanation referencing ONLY the Doc/KG Entities/KG Relations listed above. If all three are
      [], Evidence must be "No relevant evidence found." -- do NOT use medical knowledge here.
   - Verdict: exactly one of SUPPORTED, CONTRADICTED, or INSUFFICIENT

-Verdicts (relative to the Question)-
- SUPPORTED: Doc, KG Entities, or KG Relations provide evidence that this option correctly answers the
     Question. Requires at least one reference in Doc, KG Entities, or KG Relations.
- CONTRADICTED: Doc, KG Entities, or KG Relations provide evidence against this option as the answer.
     Requires at least one reference in Doc, KG Entities, or KG Relations.
- INSUFFICIENT: Doc, KG Entities, and KG Relations lack relevant information to judge this option. If Doc
     =[], KG Entities=[], and KG Relations=[], verdict MUST be INSUFFICIENT regardless of your medical
     knowledge.

-Decision-
Decision must be consistent with verdicts:
- "grounded" if at least one option is SUPPORTED
- "elimination" if no option is SUPPORTED but at least one is CONTRADICTED
- "parametric" only when ALL options are INSUFFICIENT
Summary: reasoning that leads to the final answer. The content depends on Decision:
- grounded: synthesize ONLY the SUPPORTED evidence to justify the answer. No parametric knowledge.
- elimination: state which options are ruled out by CONTRADICTED evidence, then use medical knowledge to
     choose among the remaining INSUFFICIENT options.
- parametric: use medical knowledge to reason through all options and justify the answer.

-Analysis output format (follow exactly)-
[A] Doc: [1, 3]
KG Entities: ["Entity_X"]
KG Relations: [Ri, Rj]
Evidence: Document [1] and KG relation Ri indicate that ...
Verdict: SUPPORTED

[B] Doc: []
KG Entities: ["Entity_Y"]
KG Relations: [Rk]
Evidence: KG relation Rk indicates that ...
Verdict: CONTRADICTED

[C] Doc: []
KG Entities: []
KG Relations: []
Evidence: No relevant evidence found.
Verdict: INSUFFICIENT

[D] Doc: []
KG Entities: []
KG Relations: []
Evidence: No relevant evidence found.
Verdict: INSUFFICIENT

Decision: grounded|elimination|parametric
Summary: ...
Answer_choice: A|B|C|D

-Order Constraint-
Always output in this order: option analysis -> Decision/Summary/Answer_choice

Question: {question}
Options:
A. {option_a}
B. {option_b}
C. {option_c}
D. {option_d}
Documents:
{documents}
Output:
```