Condition-Gated Reasoning for Context-Dependent Biomedical Question Answering

Jash Parekh[†] **Wonbin Kweon**[†] **Joey Chan**[†] **Rezarta Islamaj**[◇] **Robert Leaman**[◇]
Pengcheng Jiang[†] **Chih-Hsuan Wei**[◇] **Zhizheng Wang**[◇] **Zhiyong Lu**[◇] **Jiawei Han**[†]

† University of Illinois Urbana-Champaign
◇ National Institutes of Health

Abstract

Current biomedical question answering (QA) systems often assume that medical knowledge applies uniformly, yet real-world clinical reasoning is inherently conditional: nearly every decision depends on patient-specific factors such as comorbidities and contraindications. Existing benchmarks do not evaluate such conditional reasoning, and retrieval-augmented or graph-based methods lack explicit mechanisms to ensure that retrieved knowledge is applicable to given context. To address this gap, we propose **CondMedQA**, the first benchmark for conditional biomedical QA, consisting of multi-hop questions whose answers vary with patient conditions. Furthermore, we propose **Condition-Gated Reasoning** (CGR), a novel framework that constructs condition-aware knowledge graphs and selectively activates or prunes reasoning paths based on query conditions. Our findings show that CGR more reliably selects condition-appropriate answers while matching or exceeding state-of-the-art performance on biomedical QA benchmarks, highlighting the importance of explicitly modeling conditionality for robust medical reasoning.

1 Introduction

Retrieval-augmented generation (RAG) [28] has emerged as an effective paradigm for grounding the reasoning process of large language models (LLMs) in external knowledge bases, reducing hallucinations and enabling access to up-to-date information. To overcome the limitations of standard RAG in multi-hop reasoning, recent approaches (e.g., GraphRAG [7], HippoRAG [20]) have introduced graph-structured retrieval. In this setting, knowledge is organized as graphs whose fundamental unit is a triplet of ⟨entity, relation, entity⟩ capturing explicit semantic relationships. In the biomedical domain, recent approaches (e.g., MedRAG [48] and MKRAG [39]) build biomedical domain-specific knowledge graphs from medical documents and electronic health records (EHRs). This structured biomedical knowledge enables more reliable and interpretable reasoning, allowing LLMs to generate clinically-grounded answers.

While knowledge graph-based approaches effectively capture relational structure, they typically encode only binary relationships and do not explicitly model clinical preferences or how these preferences shift under specific conditions. Biomedical question answering is inherently conditional, as clinical decision-making requires reasoning over patient-specific contexts such as allergies, comorbidities, and concurrent medications to choose the preferred action (or actions) given the overall clinical picture. For instance, lisinopril is commonly used as a first-line treatment for hypertension.

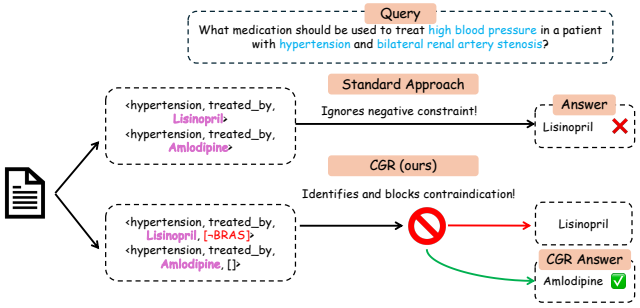

Figure 1: Example of condition-gated reasoning in the biomedical domain. Existing KG-RAG extracts triples and retrieves contraindicated treatments. CGR extracts n-tuples with patient-specific conditions, gating unsafe paths and retrieving only safe alternatives.

However, lisinopril, as an ACE inhibitor, is contraindicated in patients with bilateral renal artery stenosis, and amlodipine, a calcium channel blocker, becomes a preferred alternative. Conventional knowledge graphs may encode these facts as ⟨lisinopril, treats, hypertension⟩ and ⟨amlodipine, treats, hypertension⟩, but such representations do not capture the default preference for lisinopril, nor how this preference shifts under specific constraints, even if the knowledge graph explicitly records ⟨lisinopril, contraindicated in, bilateral renal artery stenosis⟩. Retrieved passages often describe treatments, contraindications, or preferences in isolation, without making explicit how these considerations interact in a specific patient context. As a result, a system must determine not only which facts are relevant, but which combinations of facts jointly apply, and whether missing contextual information invalidates a candidate answer.

Contribution 1: Benchmark for conditional biomedical QA. The absence of condition-aware reasoning is also reflected in existing evaluation sets. Established biomedical QA benchmarks, such as MedHopQA [16], MedHop [43], BioCDQA [8], and BioASQ [41], primarily focus on factual recall and multi-hop reasoning. These benchmarks do not systematically evaluate whether approaches can modulate their responses based on contextual constraints. Our work addresses this identified gap, by introducing **CondMedQA**, a diagnostic benchmark comprising of 100 curated questions designed to evaluate conditional reasoning in biomedical QA. To the best of our knowledge, CondMedQA represents the first benchmark

specifically targeting these capabilities. Each question requires identification of the correct response given explicit constraints that modify the applicability of standard knowledge.

Contribution 2: Framework for conditional biomedical QA. Furthermore, we propose **Condition-Gated Reasoning** (CGR), a framework that prioritizes conditional representation within knowledge graph construction and traversal. At a high level, CGR treats conditions not as attributes to be inferred implicitly by a language model, but as explicit validity constraints on retrieved knowledge. Rather than aggregating evidence solely based on semantic relevance, CGR enforces compatibility between patient-specific context and the conditions under which medical relationships hold, ensuring that only applicable information contributes to downstream reasoning. This design enables structured multi-hop inference in which intermediate conclusions are filtered by conditional constraints, preventing the propagation of contraindicated or inapplicable facts. As a result, CGR is specifically suited to biomedical queries in which correct answers emerge only through the simultaneous satisfaction of multiple interacting conditions.

We evaluate CGR on CondMedQA and several established biomedical QA benchmarks, comparing against state-of-the-art reasoning methods. Our key contributions are summarized as follows.

- CondMedQA: a new benchmark that enables systematic evaluation of conditional multi-hop reasoning in the biomedical domain, requiring models to account for patient-specific constraints when determining the applicability of medical knowledge.
- Condition-Gated Reasoning: a novel framework that explicitly models condition-aware constraints within knowledge graph construction and traversal, ensuring that only contextually valid information contributes to multi-hop inference.

We demonstrate the capabilities of CGR through extensive experiments on CondMedQA and other benchmarks, achieving substantial gains on condition-sensitive queries while matching or exceeding state-of-the-art performance on factual benchmarks.

2 Related Work

Reasoning Limitations in Standard RAG. Although Retrieval-Augmented Generation (RAG) provides LLMs with external evidence, conventional frameworks are often hampered by their reliance on the model's ability to internally bridge gaps across unstructured context. Research has highlighted the "lost-in-the-middle" effect [17, 18, 30] and fragmentation of information [19], which prevent models from recognizing logical links across documents. While strategies like long-context fine-tuning [45] and memory-augmented retrieval [36] attempt to mitigate these issues, they do not address the difficulty of reasoning over unstructured data.

Knowledge Graph-based Reasoning. One emerging approach involves the pre-emptive construction of comprehensive knowledge graphs across an entire document collection prior to the inference stage. For instance, GraphRAG [7] organizes a full corpus into hierarchical community structures and utilizes pre-computed summaries to facilitate global-scale QA. Similarly, LightRAG [12] maps relationships between entities and text segments, requiring a thorough traversal of the entire dataset during indexing. HippoRAG [20] maintains a global graph state and employs Personalized PageRank to enable associative retrieval. HyperGraphRAG [31] introduces hypergraph architectures to represent higher-order dependencies between entities and documents, supporting more complex multi-hop reasoning over large-scale data. PathRAG [2] prunes retrieved subgraphs by identifying reasoning paths most relevant to the query, reducing noise from loosely connected edges. SARG [35] synthesizes knowledge graphs from retrieved passages with bidirectional traversal to explicitly surface multi-hop reasoning chains.

Reasoning for Biomedical QA. Advances in biomedical reasoning have led to specialized RAG frameworks such as MedRAG [48], MedGraphRAG [44], KRAGEN [33], and MKRAG [39] which move beyond flat text retrieval by integrating domain-specific structure. Specifically, approaches like MedRAG construct specialized knowledge graphs from medical literature and electronic health records (EHRs), connecting entities such as drugs, diseases, and genes through clinically relevant relations. By leveraging these structured representations, these methods elicit more reliable and interpretable reasoning, allowing LLMs to ground diagnosis and treatment recommendations in observed clinical manifestations. To evaluate these capabilities, established benchmarks such as MedHop [43], BioCDQA [8], BioASQ [41], PubMedQA [21], and BioHopR [23] have become standard for measuring factual recall and multi-hop reasoning in the healthcare domain.

Non-Monotonic Reasoning. Monotonic logic assumes that once a conclusion is derived, it remains valid as new premises are added. In contrast, clinical reasoning is inherently non-monotonic: a previously valid recommendation may be overridden by new patient information, such as contraindications or drug interactions. Formal frameworks for non-monotonic reasoning include Reiter's default logic [37], McCarthy's circumscription [34], stable model semantics [9], and Dung's argumentation frameworks [6]. These frameworks have been explored in medical decision support [10], but rule-based systems often struggle with incomplete knowledge and the combinatorial complexity that come from multiple co-occurring conditions and medications.

3 CondMedQA Benchmark

Existing biomedical QA benchmarks evaluate factual recall or multi-hop reasoning, but do not systematically evaluate *conditional reasoning*, i.e. the ability to modulate answers based on patient-specific constraints (illustrated in Fig. 2). To address this gap, we introduce CondMedQA, a diagnostic benchmark of 100 questions designed to probe this capability in current state-of-the-art approaches. In this section, we first formalize the conditionality in biomedical QA, and elaborate our construction process of CondMedQA. Table 1 presents multiple types of examples from CondMedQA benchmark.

3.1 Formalization of Conditionality

We define a question as *conditional* if and only if it satisfies the following criteria:

(1) **Modifier presence**: The question contains a specific patient condition (e.g., pregnancy, comorbidity, genetic factor).
(2) **General answer existence**: There exists a standard or default answer that would apply if the modifier were absent.

Table 1: Types of conditional multi-hop reasoning required in CondMedQA. We show ***orange bold italics*** for bridge entities, *blue italics* for supporting facts, underline for the conditions, **green bold** for the conditional answer, and magenta for the general (non-conditional) answer. Each example requires synthesizing information across both documents, neither document alone contains sufficient information to answer the conditional question.

Reasoning Category	%	Example(s)
Comorbidity & Organ-Based Contraindications	57	**Doc 1:** Treatment for *major depressive disorder* includes *antidepressants* such as **bupropion**, sertraline, and *fluoxetine*. ***Bupropion*** acts as a norepinephrine-dopamine reuptake inhibitor. **Doc 2:** ***Bupropion*** lowers the seizure threshold and is contraindicated in patients with *bulimia nervosa* or other *eating disorders* due to elevated seizure risk. **Q:** Which antidepressant for MDD is contraindicated in patients with bulimia? **A:** **Bupropion** (w/o condition: Sertraline)
Diagnostic Modality Selection	23	**Doc 1:** For suspected *appendicitis*, diagnostic imaging options include CT scan, **ultrasound**, and *MRI*. *CT scan* has the highest sensitivity in adult populations. **Doc 2:** In *pediatric patients*, ***ultrasound*** is the preferred first-line imaging modality to avoid *ionizing radiation exposure*, with CT reserved for inconclusive cases. **Q:** What imaging test is preferred for suspected appendicitis in children? **A:** **Ultrasound** (w/o condition: CT Scan)
Special Population Safety	16	**Doc 1:** First-line antibiotics for *Lyme disease* include doxycycline, **amoxicillin**, and *cefuroxime*. ***Doxycycline*** is preferred in adults due to co-coverage of *Anaplasma* co-infection. **Doc 2:** ***Doxycycline*** is contraindicated during *pregnancy* due to risks to fetal *bone and teeth development*. **Amoxicillin** is considered safe across all trimesters. **Q:** What antibiotic is recommended for Lyme disease in a pregnant patient? **A:** **Amoxicillin** (w/o condition: Doxycycline)
Drug-Drug Interactions & Pharmacogenomics	4	**Doc 1:** Standard therapy for *tuberculosis* includes rifampin, *isoniazid*, *pyrazinamide*, and *ethambutol*. ***Rifabutin*** is an alternative rifamycin with a similar mechanism. **Doc 2:** Rifampin is a potent inducer of CYP3A4 and is contraindicated with *HIV protease inhibitors*. ***Rifabutin*** has fewer CYP3A4 interactions and is preferred in patients on *antiretroviral therapy*. **Q:** Which drug replaces rifampin in TB treatment for HIV patients on protease inhibitors? **A:** **Rifabutin** (w/o condition: Rifampin)

(3) **Answer divergence**: The correct answer given the modifier differs from the general answer.

(4) **Dual validity**: The general answer is correct in typical cases, conditional answer is correct specifically due to the modifier.

(5) **Causal dependence**: The modifier directly causes the answer to change, removing the modifier reverts the to the general case.

This formalization ensures that LLMs cannot succeed by memorizing default answers and must recognize how patient-specific constraints adjust the applicability of medical knowledge.

3.2 Data Construction

We construct CondMedQA through a three-stage pipeline designed to balance question diversity with quality.

3.2.1 Candidate Generation. We prompt Gemini-3-Pro [11] with few-shot examples of conditional medical questions, instructing it to generate question-answer pairs where patient-specific factors change the correct response using two Wikipedia articles. The model is prompted to provide: (1) the conditional question, (2) the correct answer given the condition, (3) what the general answer would be without the condition, (4) an explanation of why the condition changes the answer, and (5) the two Wikipedia articles used to synthesize the question and answer. This structured output facilitates downstream verification.

3.2.2 Automated Filtering. Generated candidates undergo rule-based filtering to remove: (1) questions where the "conditional" and "general" answers are identical, (2) questions lacking explicit patient modifiers, (3) duplicates via embedding similarity, (4) questions with ambiguous or multiple valid answers. and (5) questions with invalid Wikipedia links.

3.2.3 Manual Verification. All candidates underwent review by members of our research team. For each question, reviewers verify: (1) the question satisfies all five conditionality criteria, (2) the provided answer is consistent with medical references, (3) the question is unambiguous and well-formed, and (4) both Wikipedia documents are used to form a logical reasoning trace that contains the answer. Reviewers correct phrasing issues and reject questions that fail verification.

3.3 Quality Assurance

To further validate quality, a subset of 30 questions underwent independent review by three annotators with medical expertise. Annotators evaluate each question on three dimensions: (1) *Conditionality* (Yes/No), whether the question satisfies all criteria in §3.1; (2) *Answer Accuracy* (Correct/Incorrect/Uncertain), whether the provided answer aligns with clinical guidelines; and (3) *Question Quality* (1–5), overall clarity, clinical relevance, and answerability. Inter-annotator agreement is reported in Table 2. Gwet's AC1 (nominal) and AC2 (ordinal) [14] are reported as the primary chance-corrected metrics, as Krippendorff's α [24] and Cohen's κ [4] are unreliable under highly skewed marginal distributions [15]. Note that *% Agree (all)* requires exact agreement across all three annotators (e.g., all rate quality as 5), while *% Agree (pair)* reflects pairwise agreement; the latter better captures consensus given the inherent subjectivity of quality judgments.

Table 2: Inter-annotator agreement across 30 items.

Dimension	Gwet's AC	% Agree (all)	% Agree (pair)
Conditionality	0.96 (AC1)	96.7	97.8
Answer Accuracy	0.67 (AC1)	66.7	77.8
Question Quality	0.60 (AC2)	56.7	71.1

3.4 Question Types

We categorize the conditional reasoning patterns in CondMedQA into four types based on what modifies the clinical decision:

(1) **Comorbidity & Organ-Based Contraindications (57%).** Encompasses disease contraindications (e.g., avoiding beta-blockers in asthma) and organ dysfunction requiring dosage adjustment or drug substitution.
(2) **Diagnostic Modality Selection (23%).** Questions where the answer is an imaging test rather than a drug, requiring reasoning about radiation exposure, contrast contraindications, or device compatibility (e.g. MRI-unsafe pacemakers).
(3) **Special Population Safety (16%).** Life-stage considerations including pregnancy safety categories, pediatric dosing constraints, and geriatric precautions.
(4) **Drug-Drug Interactions & Pharmacogenomics (4%).** Most specific category involving enzyme interactions, contraindicated drug combinations, and genetic variations.

Table 1 provides detailed examples of each reasoning type, illustrating how information from two documents must be synthesized to arrive at the correct conditional answer.

4 Condition-Gated Reasoning

We propose **Condition-Gated Reasoning** (CGR), a framework that prioritizes conditional representation within knowledge graph construction and traversal. CGR addresses conditional biomedical QA by explicitly representing contextual conditions as integral components of knowledge graph edges. Unlike standard graph-based retrieval methods that traverse all edges uniformly, CGR gates edge traversal based on compatibility between edge conditions and query context. Figure 2 illustrates the complete pipeline.

4.1 Condition-Aware Knowledge Graph

In this subsection, we describe the construction of a condition-gated biomedical knowledge graph that encodes not only entity–entity relations but also the clinical conditions under which they apply.

4.1.1 Condition-Aware Tuple Extraction. Given a corpus of source documents $\mathcal{D}$, we extract structured 4-tuples of the form $\langle u, r, v, C \rangle$, where u and v are nodes, r is a relation, and $C = [c_1, \ldots, c_k]$ is a list of conditions under which the relationship holds or does not hold. Conditions C capture contextual constraints including patient demographics (e.g., "pediatric patients"), physiological status (e.g., "during pregnancy"), comorbidities (e.g., "with renal impairment"), disease stage (e.g., "early localized"), and contraindication contexts (e.g., "penicillin allergy"). Extraction is performed using Qwen2.5-14B-Instruct-GPTQ-Int4 [46]. Examples of the extracted tuples and associated conditions are provided in Appendix B.

4.1.2 Entity Normalization. Extracted entities are normalized to canonical forms using the UMLS Metathesaurus [38], ensuring consistent representation across synonymous terms (e.g., "heart attack" → "myocardial infarction").

4.1.3 Gated Knowledge Graph Construction. Normalized tuples are assembled into a directed knowledge graph $G = (V, E)$, where nodes $v \in V$ correspond to unique entities and edges $e \in E$ encode relationships with associated conditions. Each edge $e = \langle u, r, v, C \rangle$ carries S_e, a set of evidence snippets that provide supporting text spans. Our knowledge graph structure permits multiple edges between the same node pair with distinct relations or conditions.

4.2 Query Processing

This subsection describes how queries are converted into structured forms for graph-based reasoning. We parse queries to extract entities and conditions, and use LLM-based evaluation to align query semantics with condition-aware graph edges.

4.2.1 Query Parsing. Given a natural language query q, we extract a structured representation: $\text{parse}(q) = \{K, C, N\}$ where K is a set of entity keywords, C represents required and excluded conditions, and N are negated entities to exclude from candidate answers. Query parsing is performed using Qwen2.5-14B-Instruct-GPTQ-Int4 [46]. Examples are provided in Table 7 (Appendix C).

4.2.2 LLM-Based Condition Evaluation. A key challenge in condition matching is semantic variability: the query may express conditions differently than the graph edges (e.g., "5-year-old patient" vs. "in children"; "kidney disease" vs. "renal impairment"). Traditional keyword matching fails on synonyms and negations. We address this through LLM-based condition evaluation. Prior to graph traversal, we collect all unique conditions $C_G = \bigcup_{e \in E} C_e$ from the graph and evaluate them against the query in a single LLM call: $\text{eval} : C_G \times q \to \{\texttt{True}, \texttt{False}, \texttt{Null}\}^{|C_G|}$. The evaluation returns `True` if the query context satisfies the condition, `False` if it explicitly violates the condition, and `Null` if the query provides no relevant information. This single-pass evaluation yields a lookup table $\mathcal{L} : C_G \to \{\texttt{True}, \texttt{False}, \texttt{Null}\}$, enabling O(1) condition checks during traversal. Examples can be found in Table 8 (Appendix D).

4.3 Reasoning over Condition-Gated Graph

This subsection presents our framework for traversing and reasoning over the knowledge graph (Algorithm 1).

4.3.1 Condition-Gated Graph Traversal. Given a process query, we describe how CGR traverses the constructed knowledge graph, where edges are gated by conditions.

Entry Node Selection. We identify entry nodes via semantic matching between query entities and graph nodes. Specifically, we encode both the query and all node labels using MedEmbed [1], then select the top-k nodes (ablated in §5.3) whose cosine similarity exceeds a threshold τ as starting points.

Edge Gating. We perform breadth-first search from entry nodes, where edge traversal is gated by the condition lookup table $\mathcal{L}$. For an edge $e = \langle u, r, v, C_e \rangle$ with conditions $C_e = \{c_1, \ldots, c_k\}$, we define

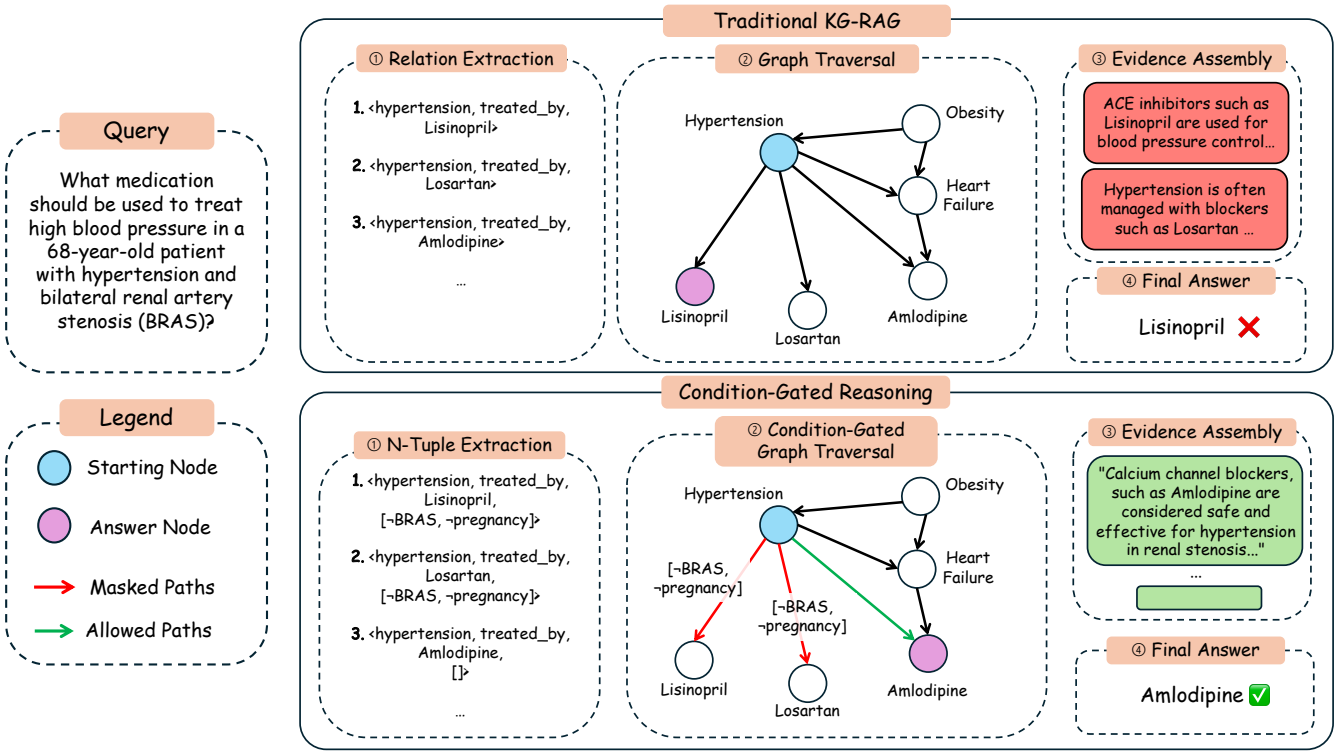

Figure 2: Overview of Condition-Gated Reasoning (CGR) compared to traditional KG-RAG. Given a clinical query about a patient with hypertension and bilateral renal artery stenosis (BRAS), traditional KG-RAG extracts standard relation triples and traverses all paths indiscriminately, retrieving evidence for contraindicated treatments (e.g., Lisinopril, an ACE inhibitor). CGR extends triples to n-tuples that include patient-specific conditions as gating constraints (e.g., ¬BRAS, ¬pregnancy), masking contraindicated paths during graph traversal and assembling only condition-appropriate evidence, yielding the correct answer (Amlodipine).

a gating function:

$$\mathcal{G}(e, \mathcal{L}) = \prod_{c \in C_e} \mathbb{1}[\mathcal{L}(c) \neq \texttt{false}] \tag{1}$$

where $\mathbb{1}[\cdot]$ is the indicator function. An edge is traversable ($\mathcal{G} = 1$) only if *no* condition evaluates to `false`. Conditions evaluating to `null` (unknown) do not block traversal. This conservative policy prunes edges only when the query explicitly violates a condition, and the resulting subgraph $G^{(q)}$ contains only edges satisfying patient context.

Termination Criteria. Traversal along a path terminates when: (1) a maximum depth $d_{\max}$ is reached, (2) no outgoing edges satisfy the gating condition, or (3) the node has no outgoing edges (leaf node). This allows multi-hop reasoning chains of up to $d_{\max}$ edges, where each hop is gated by patient context. All paths that are collected into the set of candidate reasoning paths $\mathcal{P}_q$.

Example. Consider the query in Figure 2: "What medication for hypertension in a 68-year-old patient with bilateral renal artery stenosis (BRAS)?". Starting from node 'hypertension', the traversal encounters edges to various treatments:

- ⟨hypertension, treated_by, lisinopril, [¬BRAS, ¬pregnancy]⟩ — **blocked**, since $\mathcal{L}(\neg\text{BRAS}) = \texttt{false}$
- ⟨hypertension, treated_by, losartan, [¬BRAS]⟩ — **blocked**, since $\mathcal{L}(\neg\text{BRAS}) = \texttt{false}$
- ⟨hypertension, treated_by, amlodipine, []⟩ — **traversed**, no conditions violated

The gating mechanism prunes contraindicated paths, leaving only safe treatment options for context assembly.

Algorithm 1 Condition-Gated Edge Filtering

Require: Query q, Graph $G = (V, E)$ with edge conditions C_e
Ensure: Lookup dict $\mathcal{L}$, Traversable subgraph G'

$C \leftarrow \bigcup_{e \in E} C_e$ ▷ Collect unique conditions
$\mathcal{L} \leftarrow$ LLM-Evaluate(q, C) ▷ Single LLM call
for each edge $e = (u, v)$ during BFS **do**
  for each $c \in C_e$ **do**
    if $\mathcal{L}[c] = \texttt{false}$ **then**
      skip e ▷ Query contradicts condition
    end if
  end for
  Add e to G'; enqueue v
end for

4.3.2 Path Ranking and Answer Generation. CGR then ranks the resulting reasoning paths, and generates grounded answers from the top-ranked evidence.

Path Ranking. Candidate paths $p \in \mathcal{P}_q$ are ranked by aggregate semantic similarity between the path embedding and query keywords $k \in K$:

$$\text{score}(q, p) = \sum_{k \in K} \cos\Big(\phi(p),\ \phi(k)\Big) \tag{2}$$

where $\phi(\cdot)$ denotes the MedEmbed [1] embedding function and $\phi(p)$ is the embedding of the full linearized path $\langle e_1, r_1, e_2, r_2, \ldots, e_n \rangle$ concatenated as a text sequence. The top-N paths $\mathcal{P}_q^N$ are selected as evidence for answer generation.

Evidence Assembly. Selected paths $\mathcal{P}_q^N$ are assembled into a structured evidence package $\mathcal{E}_q$ for answering query q:

$$\mathcal{E}_q = \{(V_p, E_p, S_p, C_p)\}_{p \in \mathcal{P}_q^N} \tag{3}$$

where V_p denotes the sequence of entities along path p, E_p the set of edges, S_p the source text snippets supporting the edges, and C_p the conditions associated with the traversed edges. This structured format preserves each path and enables the model to trace reasoning back to source documents.

Answer Generation. The final answer is generated by passing a prompt comprising of the query, top-k reasoning paths, and their associated evidence to an LLM.

$$A = \text{LLM}\Big(\underbrace{q \oplus \mathcal{P}_q^N \oplus \mathcal{E}_q \oplus \texttt{instructions}}_{\text{prompt}}\Big) \tag{4}$$

The instructions direct the model to synthesize a grounded response while respecting the conditions that gated traversal.

4.3.3 Computational Complexity. CGR requires $O(1)$ LLM calls for condition evaluation regardless of graph size, as all conditions are evaluated in a single batch. Graph traversal is $O(|V| + |E|)$ with $O(1)$ condition lookup per edge. The primary computational cost is tuple extraction, which scales linearly with corpus size.

5 Experiments

We evaluate CGR against retrieval-augmented baselines across four biomedical QA benchmarks.

5.1 Experimental Setup

5.1.1 Datasets. We evaluate on four benchmarks spanning factoid and multi-hop biomedical QA:

(1) **CondMedQA** (ours): 100 questions requiring conditional reasoning over patient-specific constraints (§3).
(2) **MedHopQA** [16]: 400 clinically reviewed multi-hop questions requiring reasoning across multiple medical documents.
(3) **MedHopQA (Cond)**: A disjoint, clinically reviewed set of 35 conditional questions drawn from the same MedHopQA source but excluded from the 400-question split above.
(4) **BioASQ Task B** [41]: Biomedical factoid questions. We randomly sample 500 questions from the full test set (5,389 questions) due to the computational cost of LLM-based graph construction and evaluation.

5.1.2 Baselines. We compare against three categories of methods: **Non-retrieval:** Zero-shot prompting and chain-of-thought (CoT) reasoning [42]; **Standard RAG:** Dense retrieval with MedCPT [22], followed by LLM generation; **Graph-augmented RAG:** HippoRAG2 [13], StructRAG [29], PathRAG [2], HyperGraphRAG [31], MedRAG [48], and MKRAG [39].

5.1.3 Evaluation Metrics. We evaluate using Exact Match (EM) and token-level F1. EM measures whether the predicted answer exactly matches the ground truth after normalization. F1 computes the harmonic mean of precision and recall over normalized token overlap between prediction and ground truth, following prior work [32].

5.1.4 Implementation. All baseline methods use GPT-5.2 [40] for answer generation. For CGR, we use Qwen2.5-14B-Instruct-GPTQ-Int4 [46] for tuple extraction (ablated in §5.3), deployed via vLLM [27] for optimized performance, and MedEmbed-large-v0.1 [1] for path ranking. We adapt all methods to operate in a **post-retrieval setting**, where reasoning is performed exclusively over the provided gold documents. We leave integration with full scientific document retrieval [25, 26, 47] as future work.

5.2 Results

Table 3 presents our results across four biomedical QA benchmarks. CGR achieves state-of-the-art performance across all datasets and foundation models, with particularly strong gains on conditional questions.

5.2.1 Main Findings. With GPT-5.2, CGR achieves 82.00% EM on CondMedQA, outperforming the strongest baseline by 20 points. On MedHopQA, CGR reaches 86.75% EM compared to MedRAG's 75.75%, an 11-point improvement. These gains are consistent across models: with Qwen2.5-14B, CGR improves over RAG by 11 points EM on CondMedQA and 15 points on MedHopQA.

5.2.2 Analysis. We highlight several key observations from our results below and analyze failure modes in Table 9 of Appendix E.

(1) *Condition-dependent questions benefit most.* CGR shows its strongest and most consistent improvements on condition-dependent benchmarks (CondMedQA and MedHopQA-Cond), where constraints alter the correct answer and must be respected during reasoning.
(2) *Strong performance on conditional and non-conditional queries.* CGR does not require questions to be conditional. When no conditions are present, edges remain ungated and traversal proceeds normally. This is reflected in strong performance on MedHopQA and BioASQ-B, which evaluate standard multi-hop questions.
(3) *Consistent improvement across model scales.* CGR improves over baselines regardless of foundation model size, from GPT-5.2 to LLaMA-3.1-8B, showing that the structured reasoning framework provides value beyond what larger models alone can achieve.
(4) *Graph-based methods not explicitly modeling conditionality underperform.* HippoRAG2, HyperGraphRAG, PathRAG, StructRAG, and our ablation study of CGR without gating (§5.3) all

Table 3: Performance comparison across biomedical multi-hop QA datasets and our custom benchmark. **Bold** indicates the best performance within each foundation model group.

Method	CondMedQA		MedHopQA		MedHopQA (Cond)		BioASQ-B	
	EM	F1	EM	F1	EM	F1	EM	F1
GPT-5.2								
Zero-Shot	22.00	41.29	20.50	47.87	28.27	53.52	13.00	40.21
RAG	44.00	52.36	51.25	62.61	45.71	55.04	31.80	54.80
CoT	40.00	52.81	49.50	57.55	50.00	59.38	25.60	47.81
HippoRAG2	46.00	67.56	45.00	60.81	48.57	72.82	22.60	43.57
HyperGraphRAG	57.00	64.48	37.50	41.90	54.29	65.24	28.60	54.44
StructRAG	62.00	71.22	71.75	79.61	65.71	78.57	31.80	55.07
PathRAG	53.00	66.54	30.00	34.36	42.86	54.07	24.55	51.04
MedRAG	57.00	70.52	75.75	84.42	74.29	**89.31**	35.40	55.51
MKRAG	60.00	71.14	51.25	65.56	62.86	79.33	28.80	55.35
CGR (ours)	**82.00**	**86.67**	**86.75**	**89.91**	**80.00**	85.44	**40.60**	**57.99**
Qwen2.5-14B-Instruct								
Zero-Shot	34.00	39.04	43.20	51.14	34.29	43.09	15.80	33.44
RAG	44.00	52.01	45.50	55.54	42.86	50.19	31.80	**47.81**
CoT	49.00	53.50	42.50	52.44	37.14	46.24	20.80	34.88
CGR (ours)	**55.00**	**64.47**	**61.30**	**67.40**	**51.43**	**54.11**	**34.60**	44.08
LLaMA-3.1-8B-Instruct								
Zero-Shot	30.00	36.17	34.75	43.75	37.14	48.78	15.60	35.07
RAG	47.00	52.36	46.75	55.17	45.71	50.09	33.40	**53.66**
CoT	50.00	**58.67**	40.50	47.82	37.14	48.16	18.40	32.01
CGR (ours)	**52.00**	54.17	**61.50**	**67.36**	**48.57**	**50.48**	**35.80**	47.21

lag behind CGR, suggesting that naive graph construction without explicit condition modeling is insufficient for biomedical reasoning.

5.3 Ablation Studies

We conduct ablation studies to validate the necessity of condition gating, extraction, and analyze sensitivity to key hyperparameters.

Table 4: Ablation study on the effect of condition gating.

Method	CondMedQA		MedHopQA		MedHop (C)	
	EM	F1	EM	F1	EM	F1
CGR w/o gating	57.0	60.2	72.5	76.8	62.9	73.5
CGR	**82.0**	**86.7**	**86.8**	**89.9**	**80.0**	**85.4**

5.3.1 Effect of Condition Gating. To isolate the impact of our gating mechanism, we compare CGR to a variant that performs identical n-tuple extraction, graph construction, and traversal, but all edges are traversable regardless of context. As shown in Table 4, removing condition gating causes substantial performance degradation on CondMedQA and MedHopQA-Cond, while the drop on MedHopQA is more modest. This aligns with expectations, as CondMedQA and MedHopQA-Cond explicitly require conditional reasoning, making gating essential. MedHopQA tests multi-hop QA, so ungated traversal remains effective. These results confirm our hypothesis that condition gating is a critical reasoning component for context-aware QA.

5.3.2 Effect of Extraction Model. We ablate the LLM used for n-tuple graph extraction while keeping the rest of the pipeline fixed. We compare three extraction models: (1) Qwen2.5-14B-Instruct-GPTQ-Int4 [46], (2) Flan-T5-Large [3], and (3) LLaMA-3.2-3B-Instruct [5], all served locally via vLLM. For each configuration, we evaluate on a random 20-question subset (seed 42) from each benchmark; reasoning and answer generation use GPT-5.2. Flan-T5 uses shorter passage chunks (1500 chars) to respect its 512-token context limit. As shown in Table 5, Qwen2.5-14B achieves the best F1 on MedHopQA (92.50) and ties for best on MedHopQA-Cond (88.33), while all models perform similarly on CondMedQA. LLaMA-3.2-3B matches Qwen2.5-14B on conditional benchmarks at comparable throughput (15.04 vs 15.08 s/doc). These results demonstrate an accuracy-efficiency tradeoff, with Qwen2.5-14B providing the optimal balance for our experiments.

Table 5: Ablation study on n-tuple extraction model, showing that higher quality n-tuple extraction leads to better CGR performance. Tok/s = tokens per second, s/doc = seconds per document (throughput from vLLM)

Extractor	CondMedQA		MedHopQA		MedHop (C)		Tok/s	s/doc
	EM	F1	EM	F1	EM	F1		
Flan-T5-Large	65.00	66.67	85.00	90.00	70.00	77.33	1053.95	21.69
LLaMA-3.2-3B	65.00	66.67	85.00	90.00	85.00	88.33	932.12	**15.04**
Qwen2.5-14B	65.00	66.67	85.00	**92.50**	85.00	88.33	**1077.55**	15.08

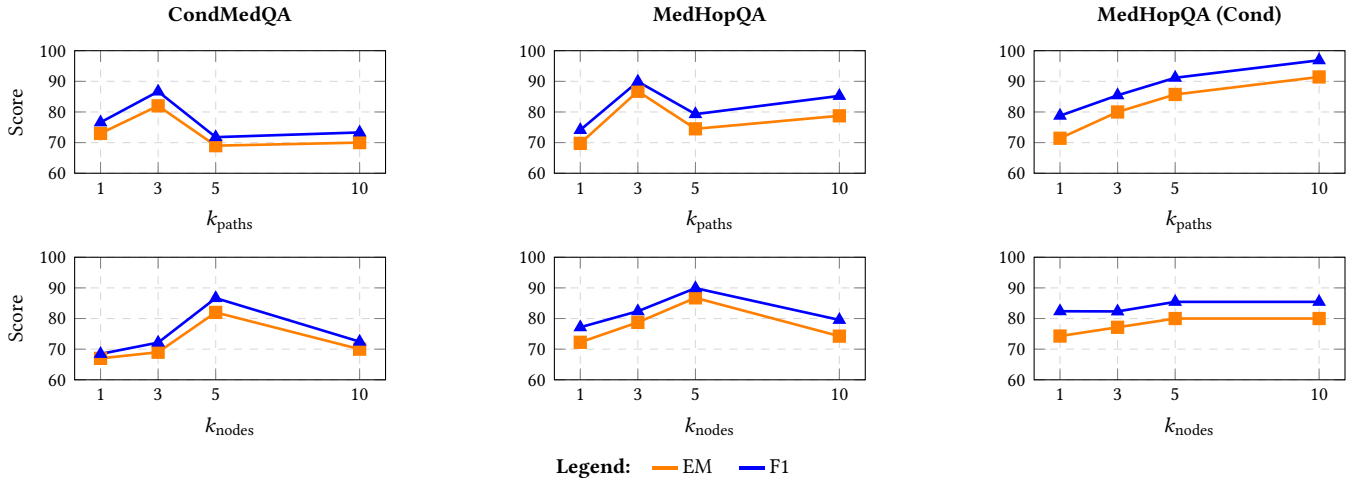

Figure 3: Hyperparameter sensitivity analysis. Top row: varying $k_{\text{paths}} \in \{1, 3, 5, 10\}$. Bottom row: varying $k_{\text{nodes}} \in \{1, 3, 5, 10\}$. Based on these results, we set $k_{\text{paths}} = 3$ and $k_{\text{nodes}} = 5$ for all experiments.

5.3.3 Hyperparameter Sensitivity. Figure 3 analyzes sensitivity across key hyperparameters. Top-k paths controls how many candidate reasoning paths are passed to the LLM for answer generation after ranking. Performance improves from k=1 to k=3 as additional paths provide supporting evidence, but degrades at k=10 due to noise from lower-ranked paths. Top-k nodes per keyword determines how many entry points are selected when initializing graph traversal; retrieving k=5 entry nodes provides sufficient coverage of relevant subgraphs without introducing irrelevant regions. CGR demonstrates stable performance across hyperparameter ranges, with consistent improvements over baselines in all settings.

6 Interdisciplinary Contributions

Contributions to Biomedical Research. The CGR algorithm enables traceable conditional reasoning by constructing evidence trails grounded in published medical research. Additionally, the CondMedQA benchmark addresses a critical gap by providing a dedicated evaluation dataset for conditional biomedical reasoning.

Challenges Addressed by AI/ML. Current RAG systems produce answers without exposing the reasoning chain that connects evidence to conclusions. CGR addresses this by structuring reasoning as explicit graph traversal grounded in the source text, rather than treating the model as a black box.

Challenges of Using AI/ML. Deploying AI/ML in clinical settings requires expert verification of outputs, adaptation to evolving guidelines, and human oversight for liability. CGR's interpretable reasoning paths address this gap, but AI recommendations are intended to assist clinical decision-making, not replace it.

7 Limitations and Ethical Considerations

While CGR demonstrates strong performance on conditional biomedical reasoning, we acknowledge several limitations of our approach and important ethical considerations for deployment in clinical settings.

7.1 Limitations

Benchmark Scope. CondMedQA comprises 100 questions and serves as a diagnostic benchmark for conditional biomedical reasoning. Future work will expand coverage across condition types.

Knowledge Source Dependence. CGR constructs knowledge graphs from retrieved documents, inheriting any errors, omissions, or biases present in the source material.

7.2 Ethical Considerations

Intended Use and Clinical Disclaimer. CGR and CondMedQA are designed for research purposes only and are not intended for clinical decision-making. The system should not be used to provide medical advice, diagnosis, or treatment recommendations.

Bias and Fairness. Medical knowledge bases may reflect biases in clinical research, including underrepresentation of certain groups in clinical trials. CGR inherits these biases through its reliance on extracted medical knowledge. Auditing CondMedQA for demographic bias represents important future work.

Data Privacy and Consent. CondMedQA was constructed from publicly available medical literature and does not contain protected health information or data from human subjects. The LLM-assisted generation process used only synthetic clinical scenarios.

8 Conclusion

In this paper, we introduced CondMedQA, a benchmark for context-dependent biomedical question answering where patient-specific conditions change the correct answer. We also propose Condition-Gated Reasoning (CGR), a framework that explicitly models these conditional dependencies through structured knowledge graph traversal. By extracting condition-aware n-tuples and gating graph edges based on context, CGR achieves state-of-the-art performance across multiple biomedical QA benchmarks, significantly outperforming existing approaches that treat all retrieved information uniformly. Our results demonstrate that explicit conditional modeling offers a promising paradigm for medical QA, addressing a

fundamental limitation of current systems that often ignore contraindications, drug interactions, and other safety concerns. We believe this approach can be extended to other high-stakes domains where contextual factors modify correct answers, offering a general framework for condition-aware reasoning.

Acknowledgements

Research was supported in part by the AI Institute for Molecular Discovery, Synthetic Strategy, and Manufacturing: Molecule Maker Lab Institute (MMLI), funded by U.S. National Science Foundation under Award No. 2505932, NSF IIS 25-37827, and the Institute for Geospatial Understanding through an Integrative Discovery Environment (I-GUIDE) by NSF under Award No. 2118329. The research has used the Delta/DeltaAI advanced computing and data resource, supported in part by the University of Illinois Urbana-Champaign and through allocation #250851 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants OAC 2320345, #2138259, #2138286, #2138307, #2137603, and #2138296. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and do not necessarily represent the views, either expressed or implied, of DARPA or the U.S. Government.

References

[1] Abhinand Balachandran. 2024. *MedEmbed: Medical-Focused Embedding Models.* https://github.com/abhinand5/MedEmbed

[2] Boyu Chen, Zirui Guo, Zidan Yang, Yuluo Chen, Junze Chen, Zhenghao Liu, Chuan Shi, and Cheng Yang. 2026. Pathrag: Pruning graph-based retrieval augmented generation with relational paths. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 40. 30183–30191.

[3] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research* 25, 70 (2024), 1–53.

[4] Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement* 20, 1 (1960), 37–46.

[5] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv e-prints* (2024), arXiv–2407.

[6] Phan Minh Dung. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence* 77, 2 (1995), 321–357.

[7] 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 GraphRAG Approach to Query-Focused Summarization. *arXiv preprint arXiv:2404.16130* (2024).

[8] Yichun Feng, Jiawei Wang, Ruikun He, Lu Zhou, and Yixue Li. 2025. A retrieval-augmented knowledge mining method with deep thinking LLMs for biomedical research and clinical support. *GigaScience* 14 (2025), giaf109.

[9] Melvin Fitting. 1992. Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. Logic programming, Proceedings of the fifth international conference and symposium, Volume 2, edited by Robert A. Kowalski and Kenneth A. Bowen, Series in logic programming, The MIT Press, Cambridge, Mass., and London, 1988, pp. 1070–1080.-Kit Fine. The justification of negation as failure. Logic, methodology and philosophy of science VIII, Proceedings of the Eighth International Congress of Logic, Methodology and Philosophy of Science, Moscow, 1987, edited by Jens Erik Fenstad, Ivan T. Frolov, and Risto Hilpinen, Studies in logic and the foundations of mathematics, vol. 126, North-Holland, Amsterdam etc. 1989, pp. 263–301. *The Journal of Symbolic Logic* 57, 1 (1992), 274–277.

[10] John Fox and Subrata Das. 2000. *Safe and Sound: Artificial Intelligence in Hazardous Applications.* MIT Press, Cambridge, Mass.

[11] Gemini Team, Google. 2025. Gemini 3: Introducing the latest Gemini AI Model from Google. https://blog.google/products/gemini/gemini-3/

[12] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2025. LightRAG: Simple and Fast Retrieval-Augmented Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025.*

[13] Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802* (2025).

[14] Kilem Gwet. 2001. Handbook of inter-rater reliability. *Gaithersburg, MD: STATAXIS Publishing Company* (2001), 223–246.

[15] Kilem Li Gwet. 2008. Computing inter-rater reliability and its variance in the presence of high agreement. *Brit. J. Math. Statist. Psych.* 61, 1 (2008), 29–48.

[16] Rezarta Islamaj, Joey Chan, Robert Leaman, Jongmyung Jung, Hyeongsoon Hwang, Quoc-An Nguyen, Hoang-Quynh Le, Harikrishnan Gurushankar Saisudha, Ganesh Chandrasekar, Rustam R Taktashov, et al. 2026. Overview of the MedHopQA track at BioCreative IX: track description, participation and evaluation of systems for multi-hop medical question answering. *arXiv preprint arXiv:2605.12313* (2026).

[17] Pengcheng Jiang, Lang Cao, Ruike Zhu, Minhao Jiang, Yunyi Zhang, Jimeng Sun, and Jiawei Han. 2025. RAS: Retrieval-And-Structuring for Knowledge-Intensive LLM Generation. *arXiv preprint arXiv:2502.10996* (2025).

[18] Pengcheng Jiang, Siru Ouyang, Yizhu Jiao, Ming Zhong, Runchu Tian, and Jiawei Han. 2025. Retrieval and structuring augmented generation with large language models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2.* 6032–6042.

[19] Ziyan Jiang, Xueguang Ma, and Wenhu Chen. 2024. Longrag: Enhancing retrieval-augmented generation with long-context llms. *arXiv preprint arXiv:2406.15319* (2024).

[20] Bernal Jimenez Gutierrez, Yi Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. HippoRAG: Neurobiologically Inspired Long-Term Memory for Large Language Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 37.

[21] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A Dataset for Biomedical Research Question Answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

[22] Qiao Jin, Won Kim, Qingyu Chen, Donald C Comeau, Lana Yeganova, W John Wilbur, and Zhiyong Lu. 2023. Medcpt: Contrastive pre-trained transformers with large-scale pubmed search logs for zero-shot biomedical information retrieval. *Bioinformatics* 39, 11 (2023), btad651.

[23] Yunsoo Kim, Yusuf Abdulle, and Honghan Wu. 2025. Biohopr: A benchmark for multi-hop, multi-answer reasoning in biomedical domain. In *Findings of the Association for Computational Linguistics: ACL 2025.* 12894–12908.

[24] Klaus Krippendorff. 2011. Computing Krippendorff's alpha-reliability. (2011).

[25] Wonbin Kweon, SeongKu Kang, Runchu Tian, Pengcheng Jiang, Jiawei Han, and Hwanjo Yu. 2025. Topic Coverage-based Demonstration Retrieval for In-Context Learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing.*

[26] Wonbin Kweon, Runchu Tian, SeongKu Kang, Pengcheng Jiang, Zhiyong Lu, Jiawei Han, and Hwanjo Yu. 2026. PairSem: LLM-Guided Pairwise Semantic Matching for Scientific Document Retrieval. In *Proceedings of the ACM Web Conference 2026.* 2396–2407.

[27] Woosuk Kwon. 2025. *vLLM: An Efficient Inference Engine for Large Language Models.* Ph. D. Dissertation. UC Berkeley.

[28] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems* 33 (2020), 9459–9474.

[29] Zhuoqun Li, Xuanang Chen, Haiyang Yu, Hongyu Lin, Yaojie Lu, Qiaoyu Tang, Fei Huang, Xianpei Han, Le Sun, and Yongbin Li. 2024. Structrag: Boosting knowledge intensive reasoning of llms via inference-time hybrid information structurization. *arXiv preprint arXiv:2410.08815* (2024).

[30] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics* 12 (2024), 157–173.

[31] Haoran Luo, Haihong E, Guanting Chen, Yandan Zheng, Xiaobao Wu, Yikai Guo, Qika Lin, Yu Feng, Zemin Kuang, Meina Song, Yifan Zhu, and Anh Tuan Luu. 2025. HyperGraphRAG: Retrieval-Augmented Generation via Hypergraph-Structured Knowledge Representation. In *Advances in Neural Information Processing Systems (NeurIPS).*

[32] Yuanjie Lyu, Zihan Niu, Zheyong Xie, Chao Zhang, Tong Xu, Yang Wang, and Enhong Chen. 2024. Retrieve-plan-generation: An iterative planning and answering framework for knowledge-intensive llm generation. *arXiv preprint arXiv:2406.14979* (2024).

[33] Nicholas Matsumoto, Jay Moran, Hyunjun Choi, Miguel E Hernandez, Mythreye Venkatesan, Paul Wang, and Jason H Moore. 2024. KRAGEN: a knowledge graph-enhanced RAG framework for biomedical problem solving using large language models. *Bioinformatics* 40, 6 (2024), btae353.

[34] John McCarthy. 1980. Circumscription—a form of non-monotonic reasoning. *Artificial intelligence* 13, 1-2 (1980), 27–39.

[35] Jash Rajesh Parekh, Pengcheng Jiang, and Jiawei Han. 2025. Structure-Augmented Reasoning Generation. *arXiv preprint arXiv:2506.08364* (2025).

[36] Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. 2025. Memorag: Boosting long context processing with global

memory-enhanced retrieval augmentation. In *Proceedings of the ACM on Web Conference 2025*. 2366–2377.

[37] Raymond Reiter. 1980. A logic for default reasoning. *Artificial intelligence* 13, 1-2 (1980), 81–132.

[38] Peri L Schuyler, William T Hole, Mark S Tuttle, and David D Sherertz. 1993. The UMLS Metathesaurus: representing different views of biomedical concepts. *Bulletin of the Medical Library Association* 81, 2 (1993), 217.

[39] Yucheng Shi, Shaochen Xu, Tianze Yang, Zhengliang Liu, Tianming Liu, Xiang Li, and Ninghao Liu. 2025. Mkrag: Medical knowledge retrieval augmented generation for medical question answering. In *AMIA Annual Symposium Proceedings*, Vol. 2024. 1011.

[40] Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267* (2025).

[41] George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics* 16, 1 (2015), 138.

[42] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[43] Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing Datasets for Multi-hop Reading Comprehension Across Documents. *Transactions of the Association for Computational Linguistics (TACL)* 6 (2018), 287–302.

[44] Junde Wu, Jiayuan Zhu, Yunli Qi, Jingkun Chen, Min Xu, Filippo Menolascina, Yueming Jin, and Vicente Grau. 2025. Medical Graph RAG: Evidence-based Medical Large Language Model via Graph Retrieval-Augmented Generation. In *ACL*. Association for Computational Linguistics, 28443–28467.

[45] Wenhan Xiong, Jingyu Liu, Igor Molybog, Hejia Zhang, Prajjwal Bhargava, Rui Hou, Louis Martin, Rashi Rungta, Karthik Abinav Sankararaman, Barlas Oguz, et al. 2024. Effective long-context scaling of foundation models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 4643–4663.

[46] An Yang et al. 2024. Qwen2 Technical Report.

[47] Yunyi Zhang, Ruozhen Yang, Siqi Jiao, SeongKu Kang, and Jiawei Han. 2025. Scientific Paper Retrieval with LLM-Guided Semantic-Based Ranking. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

[48] Xuejiao Zhao, Siyan Liu, Su-Yin Yang, and Chunyan Miao. 2025. Medrag: Enhancing retrieval-augmented generation with knowledge graph-elicited reasoning for healthcare copilot. In *Proceedings of the ACM on Web Conference 2025*. 4442–4457.

Appendix

A Use of Large Language Models

In this work, large language models (LLMs) were used to assist with editing and refining of the manuscript. All research ideas, methodology, and experimental work were conducted by the authors.

B Extraction Examples

Table 6 presents how our extraction pipeline captures entities, relations, and patient-specific conditions that determine when relationships hold.

Table 6: Examples of condition-aware n-tuple extraction. Green text indicates patient-specific conditions extracted from source text and captured in the n-tuple conditions field.

Document 1: HIV Treatment
"A large study in Africa and India found that a PI-based regimen was superior to an NNRTI-based regimen in children less than 3 years who had never been exposed to NNRTIs in the past. Thus the WHO recommends PI-based regimens for children less than 3. In pregnant women, viral load is proportional to transmission risk; ART reduces transmission to mothers and infants before, during, and after delivery." **Extracted N-Tuples:** ⟨PI-based regimen, SUPERIOR_TO, NNRTI-based regimen, [children <3 years, not exposed to NNRTI]⟩ ⟨WHO, RECOMMENDS, PI-based regimen, [children <3 years]⟩ ⟨viral load, PROPORTIONAL_TO, transmission risk, [pregnant women]⟩ ⟨ART, REDUCES, transmission risk, [pregnant women, before/during/after delivery]⟩
Document 2: Misoprostol
"Misoprostol should not be taken by pregnant women with wanted pregnancies to reduce the risk of NSAID-induced gastric ulcers because it increases uterine tone and contractions in pregnancy, which may cause partial or complete abortions, and because its use in pregnancy has been associated with birth defects." **Extracted N-Tuples:** ⟨misoprostol, CONTRAINDICATED_IN, wanted pregnancies, [pregnant women]⟩ ⟨misoprostol, INCREASES, uterine tone and contractions, [pregnancy]⟩ ⟨misoprostol, CAUSES, partial or complete abortion, [pregnancy]⟩ ⟨misoprostol, ASSOCIATED_WITH, birth defects, [pregnancy]⟩
Document 3: Urinary Tract Infection
"Urinary tract symptoms are frequently lacking in the elderly. The presentations may be vague and include incontinence, a change in mental status, or fatigue as the only symptoms. In young healthy women, diagnosis can be made from symptoms alone. Postmenopausal women may be treated with vaginal estrogen replacement. Risk factors include sexual activity in young women, chronic prostatitis in males, and vesico-ureteral reflux in children." **Extracted N-Tuples:** ⟨UTI, DIAGNOSED_BY, symptoms alone, [young healthy women]⟩ ⟨UTI, TREATED_WITH, vaginal estrogen, [postmenopausal women]⟩ ⟨UTI, PRESENTS_WITH, incontinence/mental status change/fatigue, [elderly]⟩ ⟨UTI, RISK_FACTOR, sexual activity, [young women]⟩ ⟨UTI, RISK_FACTOR, chronic prostatitis, [males]⟩ ⟨UTI, RISK_FACTOR, vesico-ureteral reflux, [children]⟩

C Query Parsing Examples

Table 7 demonstrates condition gating and entity negation. Query parsing transforms natural-language questions into structured representations for graph matching and condition gating. We define three components:

- **K** (Keywords): Entities and concepts the answer must match; drives node retrieval in the graph
- **C** (Conditions): Required context that must hold (e.g., "in children") and excluded context that must not hold (e.g., "not in adults"); used for edge gating
- **N** (Negated Entities): Entities explicitly excluded as answers (e.g., "distinct from X", "other than Y")

Table 7: Query parsing examples showing decomposition into keywords (**K**), conditions (**C**), and negated entities (**N**).

Example 1: *"Which gene causes cardiomyopathy in pediatric patients but not in adults?"*	
K (keywords)	`[gene, cardiomyopathy, causes]`
C (required)	`[pediatric, in children]`
C (excluded)	`[in adults, adult patients]`
N (negated)	`[]`
Interpretation: **K** matches nodes related to genes and cardiomyopathy. **C** gates traversal: only edges with pediatric/child conditions are followed; edges conditioned on adults are blocked.	
Example 2: *"What drug treats hypertension in pregnant women, excluding ACE inhibitors?"*	
K (keywords)	`[drug, hypertension, treats]`
C (required)	`[in pregnancy, pregnant women]`
C (excluded)	`[]`
N (negated)	`[ACE inhibitors]`
Interpretation: **K** retrieves drug and hypertension nodes. **C** restricts traversal to pregnancy-safe edges. **N** filters the final answer set: even if ACE inhibitors appear in valid paths, they are excluded from the response.	

D Condition Evaluation Examples

Table 8 presents the examples of the condition evaluation. The condition evaluation step uses a single LLM call to populate the lookup table $\mathcal{L}$ for edge gating, enabling multi-hop reasoning with each edge subject to condition gating. Given a query and a set of conditions extracted from graph edges, the LLM evaluates whether each condition is satisfied (`true`), violated (`false`), or unknown (`null`) in the query context.

Table 8: Condition evaluation examples showing how a single LLM call populates the lookup table $\mathcal{L}$ for edge gating.

Example 1: *"Which gene causes cardiomyopathy in pediatric patients but not in adults?"*	
Conditions	`[pediatric, in children, in adults, adult patients]`
Output	`{"pediatric": true, "in children": true, "in adults": false, "adult patients": false}`
Gating result: Edges conditioned on "pediatric" or "in children" are traversable ($\mathcal{G} = 1$); edges conditioned on "in adults" or "adult patients" are blocked ($\mathcal{G} = 0$).	
Example 2: *"What antibiotic is safe for a 5-year-old boy with pneumonia and no known allergies?"*	
Conditions	`[in children, in adults, penicillin allergy, renal impairment, pregnancy]`
Output	`{"in children": true, "in adults": false, "penicillin allergy": false, "renal impairment": null, "pregnancy": false}`
Gating result: Edges requiring "in children" are traversable. Edges conditioned on "penicillin allergy" remain open (patient has no allergy, so condition is not violated). "renal impairment" evaluates to `null` (unknown), so those edges remain traversable. Edges requiring "pregnancy" or "in adults" are blocked.	

E Error Analysis

We manually analyze the 18 incorrect CGR predictions on CondMedQA and categorize them into three error types: retrieval (11), extraction/normalization (5), and reasoning (2). Table 9 presents two representative examples per category; we discuss each below.

Table 9: Error analysis of 18 incorrect CGR predictions on CondMedQA. Patient-specific conditions are highlighted in color. Two representative examples are shown per error type.

Error Type	QID	Question	Gold	Predicted	Diagnosis
Retrieval (11/18)	23	What antibiotic is recommended for treating scrub typhus during pregnancy?	Azithromycin	Doxycycline	Gold tuple with pregnancy condition exists in KG but not in top-ranked paths; typhus–doxycycline edge dominates ranking.
	74	Which multikinase inhibitor is most effective for hepatocellular carcinoma in advanced (BCLC-C) unresectable disease?	Sorafenib	Insufficient evidence	Gold tuple with BCLC-C condition in KG; top paths traverse unrelated edges. Gold never reaches evidence assembly.
Extraction (5/18)	65	Which csDMARD that has the side effect of liver cancer is the most effective for ankylosing spondylitis in peripheral arthritis?	Methotrexate	Insufficient evidence	Methotrexate not extracted from source documents; KG lacks the gold entity entirely.
	96	Which anxiolytic is preferred for generalized anxiety disorder in a patient who has myasthenia gravis?	Buspirone	Sertraline	Buspirone absent from KG. Model correctly avoids benzodiazepines and selects SSRI; normalization maps SSRI to an incorrect canonical form.
Reasoning (2/18)	28	What imaging test is preferred for suspected pulmonary embolism in a pregnant patient due to lower radiation exposure?	V/Q scan	MRI w/o contrast	V/Q is the guideline alternative to CTPA in pregnancy; model over-optimizes for radiation avoidance and selects MRI.
	99	Which antimycobacterial drug should be replaced in the RIPE regimen for a patient who is HIV-positive and receiving protease inhibitors?	Rifabutin	Rifampin	Gold present in retrieved paths. Model answers the drug *to be* replaced (rifampin) rather than the replacement (rifabutin).

Retrieval errors. The gold answer entity is present in the constructed knowledge graph, but path ranking does not surface it in the top-k paths passed to the answer generation model. In Q23, the query asks for an antibiotic for scrub typhus *during pregnancy*. The knowledge graph contains the conditional tuple ⟨scrub typhus, treated_with, azithromycin, [pregnancy]⟩, correctly encoding that azithromycin is the pregnancy-safe alternative. However, the top-ranked paths instead traverse the more heavily connected typhus–doxycycline edge, which dominates due to higher semantic similarity with the query entity "scrub typhus." The gating mechanism correctly annotates doxycycline edges with pregnancy contraindications, but because azithromycin paths rank below the top-k cutoff, the model never sees the correct alternative. In Q74, a similar pattern emerges: the tuple ⟨liver carcinoma, treated_by, sorafenib, [BCLC stage C]⟩ exists with the exact condition matching the query, yet the top paths traverse tangentially related edges (e.g., liver carcinoma → ethanol, liver carcinoma → hepatitis B), and the model concludes "insufficient evidence" because sorafenib never appears in the assembled evidence. These cases reveal that while CGR's edge gating effectively blocks contraindicated paths, incorporating condition-match signals into the ranking score (Eq. 2) is a promising direction for reducing retrieval errors.

Extraction and normalization errors. In these cases, the gold entity is entirely absent from the knowledge graph. In Q65, the query asks for a csDMARD for ankylosing spondylitis in peripheral arthritis, with the gold answer being methotrexate. However, the extraction model produces tuples mentioning only glucocorticoids and TNF inhibitors from the source documents. The model reasonably responds "insufficient evidence," which is correct given its available knowledge but wrong against the benchmark. In Q96, the query asks for an anxiolytic for generalized anxiety disorder in a patient with myasthenia gravis (gold: buspirone). Buspirone does not appear in the extracted tuples. The model demonstrates sound conditional reasoning by correctly avoiding benzodiazepines (contraindicated in myasthenia gravis) and selecting an SSRI instead, but chooses sertraline rather than the gold answer. Additionally, the UMLS-based entity normalization maps the predicted SSRI to "SsrI endonuclease," a restriction enzyme, illustrating how resolution errors can compound extraction gaps. These cases suggest that improving extraction recall and auditing the UMLS normalization pipeline are important future directions.

Reasoning errors. These occur when the gold entity is present in the retrieved evidence but the answer generation model selects an incorrect response. In Q28, the query asks for the preferred imaging test for suspected pulmonary embolism in a pregnant patient with lower radiation exposure. The benchmark answer is a V/Q scan, which delivers less radiation than CT pulmonary angiography. The model instead selects MRI without contrast, which avoids ionizing radiation entirely but is not the standard recommendation due to limited availability and lower sensitivity. This suggests the model over-optimized for the "lower radiation" constraint without grounding its choice in clinical guideline knowledge. In Q99, the query asks which antimycobacterial drug should be replaced in the RIPE regimen for an HIV-positive patient on protease inhibitors. The gold answer is rifabutin (the replacement drug), but the model answers rifampin (the drug to be replaced). Notably, rifabutin appears in the retrieved reasoning paths with the correct HIV/antiretroviral conditions, so the model had access to the necessary information and just failed to synthesize the reasoning traces.

F Prompts

This appendix provides the full prompts used in the CGR pipeline: query parsing (§F.1), knowledge extraction (§F.2), condition evaluation (§F.3), and answer generation (§F.4).

F.1 Query Parsing Prompt

This prompt parses natural language questions into structured intent representations, separating target entities, required attributes, negations, and patient-specific conditions to enable condition-aware graph traversal.

Query Parsing Prompt

System: You are a query parser for biomedical knowledge graphs. Parse the question into a structured intent-aware format.
Schema:

- `target_entity`: The main concept being asked about
- `positive_attributes`: Properties the answer MUST have (2–5 items)
- `negated_entities`: Entities explicitly mentioned as NOT the answer (e.g., "distinct from", "other than")
- `required_conditions`: Conditions that MUST hold (e.g., "in males", "during pregnancy")
- `excluded_conditions`: Conditions that must NOT be present

Output: JSON object with fields: `target_type`, `target_entity`, `positive_attributes`, `negated_entities`, `required_conditions`, `excluded_conditions`

Examples:

- *"Which antibody treats metastatic colorectal cancer?"*
  → `target_type`: antibody, `positive_attributes`: [antibody, treats cancer, metastatic colorectal cancer]
- *"Which gene causes cardiomyopathy in pediatric patients but not in adults?"*
  → `required_conditions`: [pediatric], `excluded_conditions`: [in adults]
- *"What drug treats hypertension in pregnant women, excluding ACE inhibitors?"*
  → `negated_entities`: [ACE inhibitors], `required_conditions`: [in pregnancy]

F.2 Knowledge Extraction Prompt

This prompt extracts condition-aware knowledge graph n-tuples from biomedical text, capturing entities, relations, and contextual qualifiers (e.g., "in the liver", "during pregnancy") that determine when relationships hold.

Knowledge Extraction Prompt

System: You are a high-precision knowledge extraction engine for biomedical literature. Extract a dense, exhaustive knowledge graph from the passage as a JSON array of n-tuples with contextual conditions.
Core Principles:

- *Exhaustive Recall*: Capture every factual claim, no matter how small
- *Atomicity*: Each n-tuple has exactly one subject and one object; split conjunctions
- *Biomedical Anchoring*: Extract canonical entities; move descriptors ("elevated", "serum") to conditions
- *Context-Independent*: Extract from exclusions, differentials, and comparisons

JSON Schema: `entity1` (subject), `relation`, `inverse_relation`, `entity2` (object), `conditions` (list of qualifiers)

Example:
Input: "L-type and T-type calcium channels are both blocked by Compound 99 in cardiomyocytes."
Output:

```
[{``entity1'': ``L-type calcium channels'',
  ``relation'': ``blocked_by'', ``entity2'': ``Compound 99'',
  ``conditions'': [``in cardiomyocytes'']},
 {``entity1'': ``T-type calcium channels'',
  ``relation'': ``blocked_by'', ``entity2'': ``Compound 99'',
  ``conditions'': [``in cardiomyocytes'']}]
```

F.3 Condition Evaluation Prompt

This prompt evaluates whether patient-specific conditions mentioned in graph edges are satisfied, violated, or unknown given the query context, enabling the gating mechanism to block or permit edge traversal.

Condition Evaluation Prompt

System: You are evaluating medical/clinical conditions against a query.
Task: For each condition extracted from the knowledge graph, determine if the patient/context in the query SATISFIES that condition.
Return Values:

- `true`: Query explicitly or implicitly indicates condition IS satisfied
- `false`: Query explicitly or implicitly indicates condition is NOT satisfied
- `null`: Query does not mention anything relevant (unknown)

Guidelines:

- Handle synonyms: "in boys" ≡ "male children" ≡ "pediatric males"
- Handle numeric reasoning: "5-year-old" ⇒ "in adults" is `false`, "in children" is `true`
- Handle implications: "pregnant woman" ⇒ "during pregnancy" is `true`
- Handle negations: "no kidney disease" ⇒ "renal impairment" is `false`
- When truly unknown, return `null`—do not guess

Example:
Query: "What antibiotic for a 5-year-old boy with pneumonia?"
Conditions: ["in boys", "in adults", "eGFR < 30"]
Output: `{"in boys": true, "in adults": false, "eGFR < 30": null}`

F.4 Answer Generation Prompt

This prompt generates the final answer from retrieved evidence paths, instructing the model to select the best available option based on the gated graph traversal results rather than demanding perfect proof.

Answer Generation Prompt

System: You are a biomedical question-answering assistant. Provide the BEST AVAILABLE ANSWER based on the evidence.
Core Principle—Best Available Choice:

- If ONE option has relevant evidence and others don't → choose that option
- Warnings/cautions about a treatment do NOT disqualify it—they show it IS used
- Contraindications for OTHER options strengthen the case for the remaining option
- "Insufficient evidence" is a LAST RESORT—only when NO option has ANY relevant connection

Response Format:

- `REASONING`: 2–4 sentences analyzing evidence with citations (e.g., [doc1])
- `ANSWER`: Single entity name, yes/no, or short phrase—NO full sentences

Answer Formatting:

- Use what a clinician would say on rounds (e.g., "MRI" not "Magnetic Resonance Imaging")
- Be maximally concise: single word or shortest standard term
- Prefer clinical acronyms over spelled-out formal names