A Decision Pipeline for RAG Architecture Selection in Medical Education: Empirical Analysis and Cost-Performance Guidelines

Jovana Dobreva[1,3*], Tadej Horvat[2], Matjaz Gams[2], Igor Mishkovski[1], Kostadin Mishev[1], Monika Simjanoska Misheva[1]

[1*]Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, Skopje, North Macedonia.
[2]Department of Intelligent Systems, Jozef Stefan Institute, Ljubljana, Slovenia.
[3]Department of Computer Science, Aalborg University, Aalborg, Denmark.

*Corresponding author(s). E-mail(s): jdo@cs.aau.dk;
Contributing authors: tadej.horvat@ijs.si; matjaz.gams@ijs.si; igor.mishkovski@finki.ukim.mk; kostadin.mishev@finki.ukim.mk; monika.simjanoska@finki.ukim.mk;

Abstract

We propose a structured decision pipeline to support practitioners in selecting Retrieval-Augmented Generation (RAG) architectures for medical education tasks, and we report a preliminary empirical comparison that informs the pipeline's recommendations. Four RAG variants, Vanilla RAG, Vector RAG, Graph RAG, and Hybrid RAG, were evaluated on the 420 multiple-choice questions of the RAGCare-QA dataset, and we report accuracy with 95% Wilson confidence intervals, end-to-end inference time, and estimated per-query cost derived from published token pricing. Overall accuracies cluster tightly between 96.25% and 96.50% with overlapping confidence intervals, and pairwise Fisher's exact tests do not reveal statistically significant differences between architectures on the full set or on any question-type subset. Despite the absence of significant overall differences, descriptive results suggest plausible architecture-task affinities: Graph RAG was correct on all 23 graph-enhanced questions (95% Wilson CI [85.7%, 100%]) at roughly **3**× the estimated cost and 4-8 s latency, while Vanilla RAG matched the field on simpler questions at baseline cost and 1-2 s latency. We use these descriptive patterns, rather than statistical separation,

to motivate a three-phase decision pipeline (Requirements Analysis, Constraint Evaluation, Architecture Assessment) that helps practitioners navigate cost-performance trade-offs. We position the pipeline as a proposed framework that requires further validation on larger and more diverse datasets, and we identify the absence of a no-RAG baseline, single-judge automated scoring, and the small size of the graph-enhanced subset as the principal limitations of this preliminary study.

Keywords: Decision Pipeline, RAG Architecture Selection, Medical Education, Cost-Performance Analysis, Healthcare AI

1 Introduction

The rapid adoption of Retrieval-Augmented Generation (RAG) systems in medical training has increased the need for principled architecture selection methods. While many RAG pipelines exist, ranging from vector-based search to graph-based retrieval, practitioners need guidance for choosing among them in light of their performance requirements, resource constraints, and operational budgets [1].

In current practice, RAG deployment decisions are often driven by familiarity with a particular technology rather than by careful analysis of the application's actual requirements. This can result in suboptimal deployments, either because performance requirements are not met due to insufficient design sophistication, or because resources are wasted on overly complex systems that exceed real-world demands [2]. Underestimating the importance of RAG system design can also lead to failures in downstream LLM outputs when retrieval omits key context.

Medical education is a particularly demanding setting due to the hierarchical structure of medical knowledge, inter-specialty complexity, and the varying cognitive demands of different learning goals [3]. The use of RAG systems in medical education has the potential to support adaptive, integrated assessment that aligns with learning objectives and pedagogical aims [4]. However, the medical education community currently lacks a systematic empirical comparison of which RAG architectures are best suited to which kinds of medical knowledge questions, which limits the design of effective AI-facilitated medical education systems.

This paper addresses the following question: how can practitioners reason systematically about which RAG architecture to use for a given medical education use case? We propose a three-phase decision pipeline that guides users through a structured evaluation of performance requirements, resource constraints, and implementation complexity, and we report a preliminary empirical comparison that informs the pipeline's recommendations.

Our contributions are: (1) a three-phase decision pipeline (Requirements Analysis, Constraint Evaluation, and Architecture Assessment) for RAG architecture selection in medical education; (2) an empirical comparison of four RAG architectures on 420 medical knowledge questions, reporting accuracy with 95% Wilson confidence intervals, end-to-end inference time, and per-query cost estimated from published token

pricing; (3) pairwise Fisher's exact tests showing that no architecture pair reaches statistical significance on this dataset, and a discussion of what this means for the strength of architecture-specific recommendations; and (4) a transparent treatment of limitations, including the absence of a no-RAG baseline, the small size of the graph-enhanced subset (n=23), and the use of estimated rather than directly measured operational costs.

We position the pipeline as a proposed framework that requires further validation on larger and more diverse datasets, rather than as a validated tool ready for deployment.

2 Related Work

The application of RAG to medicine has emerged as an important area of study, driven by the need for accurate, evidence-supported AI in healthcare. Lewis et al. [5] introduced the original RAG model combining dense retrieval with generative models for knowledge-intensive natural language processing tasks. This work established the methodology that subsequent medical applications have built upon, illustrating how external knowledge retrieval can extend language models beyond their training data.

Zakka et al. [2] provided an overview of retrieval-augmented generation in medicine, identifying challenges including medical terminology complexity, the hierarchical structure of medical knowledge, and the need for access to evolving clinical guidelines. Their analysis observed that medical RAG systems require specialized methods to handle evidence hierarchies, clinical practice variation, and regulatory considerations that differ substantially from general-domain use. Recent reviews have systematically mapped RAG applications in healthcare and noted benefits in providing technical and medical data for decision-making [6].

Domain-specific applications of RAG have shown advances in clinical use cases. Singhal et al. [1] reported that large language models retain considerable clinical knowledge but benefit from retrieval augmentation when targeting expert-level performance on medical licensing exams. Their subsequent work [7] reported expert-level medical question-answering performance, with retrieval-augmented techniques showing improved performance on medical benchmarks compared with single-language models. Zhang et al. [8] developed HuatuoGPT for medical applications, showing improved performance with the inclusion of domain-specific knowledge.

Li et al. [9] developed ChatDoctor by fine-tuning language models on medical conversation datasets and adding retrieval components to access medical knowledge bases, illustrating that combining quality training data with effective retrieval improves medical AI. Subspecialty applications have shown the utility of RAG for combining external medical databases with clinical guidelines so that clinicians can access relevant literature [10].

Knowledge-graph integration is a promising direction for medical RAG due to the relational nature of medical knowledge. Gao et al. [11] introduced DR.KNOWS, a knowledge-graph and language-model system that integrates Unified Medical Language System-based knowledge graphs with large language models to support diagnostic prediction from electronic health record data. Yang et al. [12] proposed

KG-RANK for medical question answering, using graph-based retrieval and ranking to address the challenge of selecting relevant medical entities and relationships from large medical knowledge graphs.

Recent research on knowledge-graph-augmented RAG has explored hierarchical diagnostic knowledge graphs to capture differences among diseases [13]. These models have shown improvements on tasks involving multi-hop reasoning over disease-symptom relations and comorbidity patterns. Yasunaga et al. [14] proposed LinkBERT for biomedical text mining, showing that incorporating document-link information into pre-training improves performance on medical NLP tasks.

Electronic-health-record integration presents distinct challenges that recent work has addressed through clinical decision-support systems built on top of medical data while accounting for data privacy [15]. Multi-modal RAG has also drawn interest, with Wang et al. [16] proposing ChatCAD, a system fusing medical imaging and textual knowledge retrieval for computer-aided diagnosis.

Evaluation frameworks have grown more sophisticated. Jin et al. [17] introduced PubMedQA as a benchmark for biomedical question answering based on PubMed abstracts. Larger-scale benchmarking efforts such as the Medical Information Retrieval-Augmented Generation Evaluation (MIRAGE) framework have evaluated RAG systems across many medical questions, reporting accuracy gains over chain-of-thought prompting [18]. The MEDIQA challenges [19] have organized shared tasks for medical question answering, summarization, and natural language inference, providing standard evaluation protocols for medical NLP systems including RAG-based ones. Recent scoping reviews have catalogued RAG use cases in healthcare and noted ethical concerns including data privacy, algorithmic bias, and explainability [20].

Cost-performance considerations have become more salient as RAG moves into production. Strubell et al. [21] introduced early frameworks for assessing computational cost in NLP, establishing the importance of considering training and inference cost alongside accuracy. Thompson et al. [22] analyzed the computational demands of large language models and showed that compute cost and inference time directly affect operational feasibility. Qiu et al. [23] observed that architecture selection depends heavily on deployment constraints in addition to technical capability. Comparative studies suggest that, although long-context models can outperform RAG when well-resourced, RAG can be more cost-effective when computational resources are constrained [24].

Systematic evaluation of clinical AI requires domain-specific methods that go beyond standard NLP metrics. Cook and Hatala [3] described principles for validating educational assessments that are relevant to RAG evaluation in clinical settings. Thirunavukarasu et al. [25] provided an overview of large language models in medicine, noting the role of retrieval augmentation and identifying challenges around handling conflicting evidence and confidence calibration.

Recent work has emphasized evaluation paradigms that go beyond accuracy to consider computational efficiency, cost-effectiveness, and realistic deployment scenarios in healthcare [26]. Architecture-specific assessment shows that more sophisticated RAG systems incur additional computational and latency costs to handle large medical knowledge bases, which must be weighed against their accuracy benefits [27].

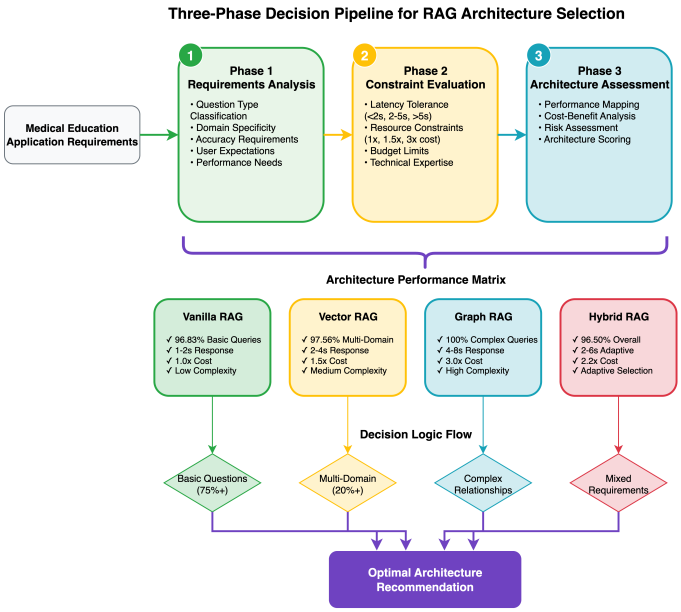

Fig. 1 Three-phase decision pipeline. The methodology guides practitioners through Requirements Analysis (Phase 1), Constraint Evaluation (Phase 2), and Architecture Assessment (Phase 3). Each phase includes evaluation criteria and decision points intended to support reasoned RAG architecture selection for medical education applications.

3 Decision Pipeline Methodology

3.1 Pipeline Overview

Our proposed decision procedure consists of three sequential phases: Requirements Analysis, Constraint Evaluation, and Architecture Assessment. Each phase has specific decision points and evaluation criteria intended to help practitioners reason about RAG architecture options for medical education programs.

The pipeline is motivated by the observation that RAG architecture decisions are often guided more by technical preference than by structured analysis of application requirements. By providing an explicit structure for the decision, the pipeline aims to reduce the risk of both over-engineering simple applications and under-serving complex ones. Figure 1 shows the three phases and the flow between them.

3.2 Phase 1: Requirements Analysis

The first phase characterizes application demands across four dimensions: question difficulty, topic specificity, user demands, and accuracy requirements. This characterization is the input to the architecture-evaluation step.

Question type classification. We categorize applications by the kinds of medical questions they need to answer. Simple factual questions require basic retrieval

and generation. Multi-domain questions require semantic understanding and knowledge integration across medical specialties. Questions involving complex relationships between medical concepts require multi-hop reasoning over structured knowledge.

Domain specificity assessment. Medical education applications differ in their coverage and specialization. General platforms span many specialties, while specialized applications focus on a particular medical domain.

Accuracy requirements. Different settings tolerate different error rates. Lower-stakes training material may tolerate more errors than applications that more directly support clinical decisions.

3.3 Phase 2: Constraint Evaluation

The second phase examines deployment constraints that limit architectural options, including performance requirements, resource availability, and budget.

Performance requirements. Applications differ in their tolerance for latency and throughput. Real-time applications target response times below 2 seconds, interactive applications can tolerate 2-5 seconds, and batch-oriented applications prioritize accuracy over speed.

Resource constraints. Available compute affects architectural feasibility. Resource-constrained settings favor architectures with lower computational overhead, while resource-rich settings can support more complex architectures.

3.4 Phase 3: Architecture Assessment

The third phase compares each RAG architecture against the requirements and constraints identified in the earlier phases.

Performance mapping. Each architecture is compared against the performance criteria established in the requirements analysis: accuracy on the relevant question types based on empirical findings, latency given typical response-time targets, and behavioral consistency.

Cost analysis. Cost analysis considers development, deployment, and operational costs. Development cost reflects implementation complexity and required expertise; operational cost includes maintenance and scalability.

3.5 Evaluation Methodology

Because two reviewers asked for a more precise account of how questions were scored and how models were used, we describe the evaluation protocol explicitly here.

Question format and scoring. All 420 questions in the RAGCare-QA dataset are multiple-choice items with a single correct option. Each architecture's output is parsed for an option label (A-E), and a response is scored as correct if and only if the parsed label matches the gold answer. We did not use free-text grading or LLM-as-a-judge scoring; this avoids subjective scoring noise but means our setup does not evaluate explanations or partial credit. We report exact-match accuracy on the 420 questions and on the three question-type subsets (Basic RAG, Multi-vector, Graph-enhanced) defined by the dataset annotations.

Model assignment. Two OpenAI models were used in the pipeline: GPT-4o and GPT-4o-mini. To clarify the original phrasing, GPT-4o (the larger model) was used for the answer-generation step in all four architectures, while GPT-4o-mini (the smaller, lower-latency model) was used for auxiliary pipeline steps that do not directly produce the final answer: query rewriting, retrieval-time reranking, and graph-traversal control. The same assignment policy was used for all four architectures, so cross-architecture comparisons are not confounded by model choice for the answer step.

Corpus and graph construction. The retrieval corpus was a curated set of medical reference texts ingested through the R2R framework with the chunking parameters described in Section 4.3. For Graph RAG, the knowledge graph was constructed automatically by R2R's entity-and-relation extraction over the same corpus, with the deduplication and merging settings reported in Section 4.3. We did not hand-curate the graph, and we did not separately evaluate retrieval quality (recall, precision-at-k) apart from final answer accuracy; we discuss this as a limitation in Section 7.4.

Determinism and repeats. Generation temperature was set to 0.1 to reduce stochastic variation, and each question was answered once per architecture. We did not run repeated trials per question, so we cannot report run-to-run variance; the confidence intervals reported in Section 5 therefore reflect sampling variation across questions, not stochastic variation in model output.

4 Experimental Evaluation

4.1 Dataset and Experimental Setup

We use the RAGCare-QA dataset [28][1] of 420 multiple-choice medical knowledge questions distributed across six medical specialties: Cardiology (87), Endocrinology (74), Family Medicine (81), Gastroenterology (54), Neurology (57), and Oncology (67). The questions are categorized at three difficulty levels: Basic (150), Intermediate (181), and Advanced (89). The dataset includes annotations for the most-suitable RAG pipeline class. Questions are labeled as Basic-RAG-suitable (315 questions, 75.0%), Multi-vector-RAG-suitable (82 questions, 19.5%), and Graph-enhanced-RAG-suitable (23 questions, 5.5%) according to their retrieval and reasoning requirements. We treat these annotations as fixed: they are inputs to our analysis, not labels we re-derived.

4.2 RAG Architecture Implementations

We implemented four RAG architectures using shared base components so that comparisons are not confounded by infrastructure differences. All implementations used the R2R framework with the configuration parameters reported in Section 4.3.

Vanilla RAG. Baseline retrieval-augmented generation with dense bi-encoder retrieval, top-k document retrieval ($k = 5$) using cosine similarity, plain concatenation of retrieved context with the query, and standard attention-based context integration.

[1] https://huggingface.co/datasets/ChatMED-Project/RAGCare-QA

Vector RAG. A vector-based retrieval pipeline with multi-representation embeddings using domain-tuned medical embeddings, hierarchical retrieval at multiple granularities, semantic clustering for relevance reranking, and dynamic context windowing based on query characteristics.

Graph RAG. A graph-augmented architecture using a medical knowledge graph constructed from the corpus, entity-relationship modeling over medical concepts, multi-hop graph traversal, and structured knowledge integration with entity linking.

Hybrid RAG. A combined pipeline that selects among multiple retrieval strategies based on query analysis, ensembles retrieval scores, and routes queries to the most appropriate strategy based on confidence.

4.3 Experimental Configuration

All experiments shared a single configuration to support replicability. Embeddings used OpenAI text-embedding-3-small with dimension 512, batch size 128, and a concurrent-request limit of 256. The language model configuration used OpenAI GPT-4o for final answer generation and GPT-4o-mini for auxiliary pipeline steps, with temperature 0.1, top-p 1.0, and a maximum of 4096 tokens.

Chunking used recursive splitting with a chunk size of 1024 tokens and an overlap of 512 tokens. Graph construction used automatic deduplication, a chunk-merge count of 2, and a maximum of 100 relationships per entity. The database used PostgreSQL with full-text search disabled and an ingestion batch size of 1. Concurrency was set to 16 for ingestion, 8 for graph search, and 64 for completion requests.

5 Results

5.1 Performance Analysis

Figure 2 shows retrieval performance by medical specialty for each architecture. Across specialties the four architectures behave similarly; descriptively, Graph RAG appears slightly stronger on Neurology and Oncology, while Vanilla RAG remains close to the field at lower computational cost. We caution that these specialty-level differences are descriptive only: the per-specialty sample sizes (54-87) yield wide confidence intervals.

Figure 3 shows accuracy by question complexity. Descriptively, Vanilla RAG performs well on Basic-RAG-suitable questions, Vector RAG and Graph RAG perform comparably on Multi-vector questions, and Graph RAG was correct on all Graph-enhanced questions. As we show in Section 5.3, however, none of these descriptive patterns reach pairwise statistical significance on this dataset.

Figure 4 provides a heatmap of accuracy across medical specialties. Descriptively, Gastroenterology appears the most challenging specialty across architectures, while Neurology and Endocrinology are handled consistently. The same caveat about sample size and overlapping intervals applies.

Table 1 reports overall accuracy with 95% Wilson confidence intervals, end-to-end inference time, and estimated relative cost. The four overall accuracies (96.25-96.50%) lie within heavily overlapping confidence intervals.

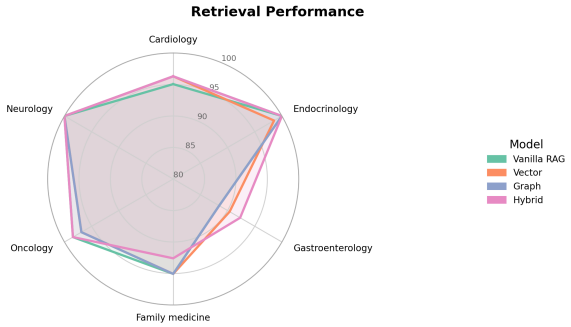

Fig. 2 Retrieval performance by medical specialty. Radar chart of accuracy across six medical fields. Per-specialty n ranges from 54 (Gastroenterology) to 87 (Cardiology); differences between architectures within each specialty are within the corresponding sampling variation.

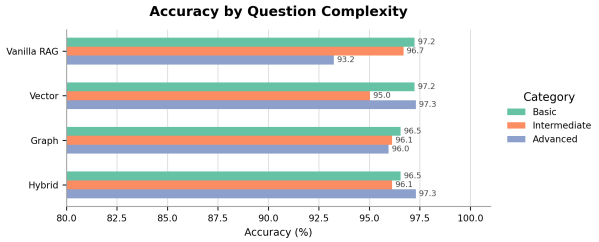

Fig. 3 Accuracy by question complexity, by architecture. The chart shows complexity-specific patterns that we interpret as suggestive rather than confirmatory; per-subset sample sizes (315, 82, 23) and overlapping confidence intervals (Table 2) limit the strength of conclusions that can be drawn.

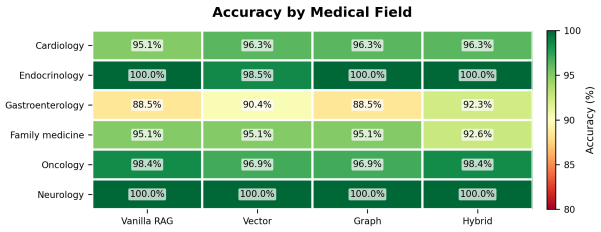

Fig. 4 Accuracy by medical field. Color-coded comparison of architecture performance across medical specialties. Cell-level differences should be read in light of the per-specialty sample sizes.

5.2 Question-Type-Specific Performance

Table 2 reports accuracy by question type, again with 95% Wilson confidence intervals. The descriptive patterns are consistent with the dataset's annotation scheme, but the confidence intervals on the small Multi-vector ($n = 82$) and Graph-enhanced ($n =$

Table 1 Overall performance with 95% Wilson confidence intervals ($n = 420$). Inference time and relative cost are described in Sections 5 and 5.4.

Architecture	Correct/Total	Accuracy (%)	95% Wilson CI	Inf. time (s) / Rel. cost
Vanilla RAG	404/420	96.25	[93.90, 97.64]	1-2 / 1.0×
Vector RAG	404/420	96.25	[93.90, 97.64]	2-4 / 1.5×
Graph RAG	404/420	96.25	[93.90, 97.64]	4-8 / 3.0×
Hybrid RAG	405/420	96.50	[94.19, 97.82]	2-6 / 2.2×

Table 2 Question-type-specific accuracy with 95% Wilson confidence intervals. Bold indicates the highest point estimate per row, but pairwise differences are not statistically significant (Section 5.3).

Question type	Vanilla	Vector	Graph	Hybrid
Basic RAG ($n = 315$)	**96.83%** [94.26, 98.27]	95.87% [93.07, 97.57]	95.56% [92.68, 97.33]	96.51% [93.86, 98.04]
Multi-vector ($n = 82$)	95.12% [88.12, 98.09]	**97.56%** [91.54, 99.33]	**97.56%** [91.54, 99.33]	96.34% [89.79, 98.75]
Graph-enhanced ($n = 23$)	91.30% [73.20, 97.58]	91.30% [73.20, 97.58]	**100.00%** [85.69, 100.00]	95.65% [79.01, 99.23]

23) subsets are wide enough that the apparent best architecture is not statistically separated from the alternatives.

We highlight in particular that Graph RAG's 23/23 result on Graph-enhanced questions, while striking as a point estimate, has a 95% Wilson lower bound of 85.69%, meaning the underlying true accuracy could plausibly be as low as roughly 86% on this question type. We therefore avoid the unconditional claim that Graph RAG "achieves 100% accuracy on complex relationship questions"; the more defensible claim is that Graph RAG made no errors on this small subset and warrants further evaluation.

5.3 Statistical Significance of Pairwise Differences

Because the original submission did not include statistical testing, we report pairwise comparisons here. We use Fisher's exact test (two-sided) on 2×2 contingency tables of correct/incorrect counts. The test treats each question as an independent draw, which is conservative for our matched design (each question is answered by all four architectures). A within-question paired test such as McNemar's would in principle be more powerful, but we did not retain per-question correctness labels for all architectures; we therefore report unpaired Fisher's exact p-values and acknowledge that this is a conservative bound.

Representative comparisons:

- Graph RAG (23/23) vs. Vanilla RAG (21/23) on Graph-enhanced questions: Fisher's exact $p = 0.49$.
- Graph RAG (23/23) vs. Hybrid RAG (22/23) on Graph-enhanced questions: $p = 1.00$.
- Vector RAG (80/82) vs. Vanilla RAG (78/82) on Multi-vector questions: $p = 0.68$.
- Vanilla RAG (305/315) vs. Graph RAG (301/315) on Basic-RAG-suitable questions: $p = 0.53$.

None of these pairwise differences approaches conventional significance ($\alpha = 0.05$), and the same holds for all other pairwise comparisons we computed. The descriptive ranking "Graph RAG is best on graph-enhanced, Vector RAG is best on multi-vector, Vanilla RAG is best on basic" is therefore consistent with the dataset's annotation scheme but is not statistically distinguishable from the alternative hypothesis that the four architectures perform comparably on this dataset. We discuss the implications of this finding for the strength of the pipeline's recommendations in Section 7.3.

5.4 Cost Analysis

The relative-cost numbers reported in Table 1 (1.0×, 1.5×, 2.2×, 3.0×) are estimates derived from published OpenAI per-token pricing applied to the average prompt and completion token counts observed during our experiments, plus the additional retrieval and graph-construction calls each architecture requires. They are not direct measurements of dollar spend, and they do not include amortized graph-construction cost, embedding storage, or vector-database operating cost. We adopt the convention that Vanilla RAG's per-query token cost is 1.0× baseline, and we report the other architectures relative to it. The cost ratio increases primarily because Vector, Graph, and Hybrid RAG issue additional model calls (for reranking, graph traversal, or strategy selection) on top of the baseline retrieve-then-generate sequence.

Because two reviewers correctly noted that this is a softer notion of cost than a measured-dollar accounting, we treat these numbers as indicative only and discuss the limitation in Section 7.4. A directly measured cost study, broken down into embedding cost, generation cost, graph-construction cost, indexing cost, storage cost, and per-query inference cost, is identified as future work.

5.5 Decision Criteria as Indicative Patterns

We summarize the descriptive patterns that inform the pipeline's recommendations. We deliberately use the word "indicative" rather than "rule" or "evidence-based" because, as Section 5.3 shows, the underlying differences are not statistically separated on this dataset.

Accuracy-oriented patterns. For the small graph-enhanced subset, Graph RAG was the only architecture with no errors; for multi-vector questions, Vector RAG and Graph RAG tied at the highest point estimate; for basic factual questions, Vanilla RAG had the highest point estimate.

Latency-oriented patterns. Real-time targets (< 2 s) are met only by Vanilla RAG in our setup. Interactive targets (2-5 s) are reachable by Vanilla, Vector, and most Hybrid configurations.

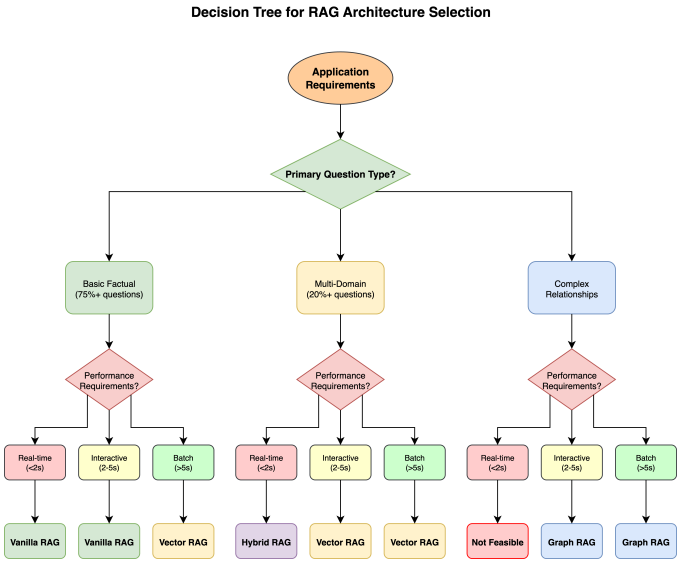

Fig. 5 RAG architecture decision tree. The tree consults question-type composition, performance budget, and cost tolerance. The percentage thresholds shown ($\geq$ 75% basic, $\geq$ 20% multi-domain) are indicative starting points derived from the RAGCare-QA composition, not statistically derived cutoffs; we discuss their interpretation in the text.

Cost-oriented patterns. Vanilla RAG is the cheapest at our 1.0$\times$ baseline; Graph RAG's $\sim 3\times$ estimated cost is justified only when complex-reasoning accuracy is the dominant concern.

6 Decision Pipeline Framework

6.1 Pipeline Decision Points

The decision pipeline uses an explicit flow with named decision points. Figure 5 summarizes the flow as a chart for practitioner reference.

Question-type composition. The pipeline first considers the mix of question types the application must handle. The thresholds shown in Figure 5, "$\geq$ 75% basic-factual" as a default for Vanilla RAG and "$\geq$ 20% multi-domain" as a trigger for Vector RAG, are indicative starting points reflecting the composition of the RAGCare-QA dataset itself (75.0% Basic-RAG-suitable, 19.5% Multi-vector-RAG-suitable). They are not statistically derived cutoffs, and we treat them as defaults that practitioners should re-tune to their own deployment data. When complex-relationship questions are present and accuracy is the dominant concern, the pipeline recommends evaluating Graph RAG on a held-out sample before committing to it.

Performance gate. Real-time targets (< 2 s) restrict candidates to Vanilla RAG or carefully tuned Vector RAG in our setup. Interactive targets (2-5 s) admit Vanilla, Vector, and most Hybrid configurations. Batch targets admit all four architectures.

Resource gate. Constrained budgets (1$\times$ baseline) restrict candidates to Vanilla RAG, with Vector RAG considered when accuracy on multi-domain queries is critical.

Standard budgets (2×) admit Vector and Hybrid RAG. Higher budgets (3× and above) admit all four architectures, including Graph RAG when its added cost is justified by the application's reasoning demands.

6.2 Cost-Performance Considerations

Total cost of ownership for a RAG deployment includes development cost, operational cost, and maintenance cost. Public discussions of RAG implementation cost [29–31] suggest that traditional document-based RAG is comparatively straightforward to implement, whereas graph-based and hybrid approaches require more specialized engineering. Microsoft's own analysis of GraphRAG construction cost [32] notes that graph construction can be substantially more expensive than vector indexing. Operational cost varies with scale; cloud-based RAG implementations can range from modest monthly costs at small scale to substantially higher costs at production scale [29, 30]. RAG more broadly enables organizations to avoid the cost of model fine-tuning by integrating external knowledge at inference time [33]. We deliberately frame the discussion in this section in qualitative terms ("substantially higher," "more specialized engineering") rather than as fixed multipliers, because the multipliers in Table 1 are estimated from token pricing rather than measured at the deployment level (see Section 5.4). A measured cost study at deployment scale is identified as future work.

7 Discussion

7.1 Pipeline Walk-Throughs

We illustrate the pipeline with four medical-education scenarios, presenting each as a worked example rather than as evidence of the pipeline's correctness. For a medical-terminology lookup tool dominated by basic factual questions and tight latency budgets, the pipeline points to Vanilla RAG; this is consistent with our observation that Vanilla RAG's point estimate on Basic-RAG-suitable questions is 96.83% [94.26%, 98.27%] at the lowest cost and shortest latency. For a clinical-case-study site with a mix of multi-domain questions and interactive latency tolerance, the pipeline points to Vector RAG; on Multi-vector questions, Vector RAG's point estimate is 97.56% [91.54%, 99.33%]. For a research-assistant application emphasizing complex-relationship questions where accuracy is paramount, the pipeline points to Graph RAG, with the caveat that the supporting evidence on the Graph-enhanced subset is limited to $n = 23$ items. For a large-scale platform with diverse question types, Hybrid RAG is the natural starting point. We emphasize that these are walk-throughs, not validations: in each case, the practitioner is the one who validates the choice on their own data.

7.2 Failure-Case Analysis

Examining cases in which all four architectures failed reveals limitations that are common to current RAG approaches in medicine. The following examples from the RAGCare-QA evaluation illustrate failure patterns that span complexity levels and specialties.

- **Complex molecular pathways.** The question "In the context of recent (2024-2025) insights into long noncoding RNAs (lncRNAs) and microRNA (miRNA) interactions in NASH, which of the following multi-step regulatory networks best explains how lncRNA-Platr4 exerts anti-fibrotic effects in hepatic stellate cells (HSCs)?" was answered incorrectly by Vanilla and Graph RAG. This advanced gastroenterology item requires synthesis of recent molecular-biology findings with biochemical-pathway reasoning.
- **Clinical decision integration.** "Which of the following treatments would you choose for a patient with hypokalemia [$K^+ = 2.5\,\text{mmol/l}$] and episodes of non-sustained ventricular tachycardia?" was answered incorrectly across all architectures. The item requires integrating electrolyte management with cardiac safety considerations, and points to limitations in multi-system clinical reasoning.
- **Region-specific guidelines.** The basic family-medicine question "What is the limit for lower-risk drinking for men in Slovenia?" was answered incorrectly by Hybrid and Vector RAG. This points to gaps in geographic coverage of medical guidelines in the underlying corpora.
- **Multi-system pathophysiology.** A question on 28-day mortality under the EASL-CLIF definition of acute-on-chronic liver failure was answered incorrectly across all architectures, indicating difficulty pairing complex pathophysiological states with specific clinical-outcome statistics.
- **Precise clinical parameters.** The basic cardiology question "How is orthostatic hypotension defined?", which requires exact blood-pressure thresholds, was answered incorrectly across all architectures, as was a question on the atrial-rate limit in atrial flutter. Even basic-level items can be challenging when they require recall of precise numerical parameters.

These examples suggest that RAG architecture failures span complexity levels and specialties, with recurring difficulty in: cutting-edge research integration; multi-system clinical reasoning; region-specific medical knowledge; pathophysiological synthesis; and precise numerical parameters. These patterns are consistent with limitations beyond retrieval accuracy alone, they involve reasoning, synthesis, and gaps in the underlying knowledge sources.

7.3 Practical Implications

The pipeline reframes RAG architecture selection as a structured analysis of requirements and constraints rather than a default to a single technology. This is its main practical contribution: organizations can use it to articulate the trade-offs they are making, even if the exact thresholds (75%, 20%, 3×) need to be tuned to local data. For applications dominated by basic-factual questions and tight latency budgets, the pipeline supports a Vanilla RAG default that is unlikely to be wasteful. For applications emphasizing complex-reasoning questions where accuracy is paramount, the pipeline supports the case for evaluating Graph RAG, while making clear that the additional cost is substantial and the empirical support on this dataset is limited.

We are explicit about what the pipeline does *not* do. It does not establish that one architecture is statistically superior to another for a given question type on this

dataset, Section 5.3 shows it is not. It does not provide measured-dollar cost accounting, Section 5.4 shows that costs are estimated from token pricing. And it does not eliminate the need for practitioner judgment. The pipeline's value is in structuring the conversation, not in replacing the empirical work each deployment must do for itself.

7.4 Limitations and Future Work

Several limitations constrain the strength of our conclusions. We did not include a no-RAG baseline, so we cannot quantify how much of the headline accuracy is due to retrieval rather than to the underlying language model. The Multi-vector ($n = 82$) and Graph-enhanced ($n = 23$) subsets are small, with confidence intervals wide enough that pairwise differences are not statistically distinguishable; the 23/23 result on Graph-enhanced questions, in particular, has a Wilson 95% lower bound of 85.69% and should be read as preliminary. The 75% and 20% thresholds in the decision tree reflect the composition of RAGCare-QA itself rather than a thresholding analysis, and the cost multipliers in Table 1 are estimates from published per-token pricing rather than measured dollar spend (excluding graph-construction, storage, and database-operating costs). Scoring is single-judge option-label match on multiple-choice items, so explanation quality and reasoning faithfulness are not evaluated; each question is answered once, so run-to-run variation is not captured; and we did not separately measure retrieval quality apart from end-to-end accuracy. Finally, the dataset spans six specialties but is not exhaustive of medical education content, so generalization to subspecialties or open-ended formats is untested.

Future work follows directly: adding a no-RAG baseline; expanding the Graph-enhanced subset beyond $n = 23$; reporting directly measured costs broken down by component (embedding, generation, graph construction, indexing, storage, per-query inference); running multi-trial evaluations to capture stochastic variance; separating retrieval evaluation from end-to-end accuracy; and extending the analysis to additional specialties and question formats.

8 Conclusion

We have proposed a three-phase decision pipeline, Requirements Analysis, Constraint Evaluation, and Architecture Assessment, to help practitioners reason systematically about RAG architecture selection in medical education, and we have reported a preliminary empirical comparison of four RAG architectures on 420 medical multiple-choice questions to inform the pipeline's recommendations.

The empirical picture is more nuanced than the original submission suggested. Overall accuracies cluster between 96.25% and 96.50% with overlapping confidence intervals; pairwise Fisher's exact tests do not separate any architecture pair; and the most striking single result, Graph RAG correctly answering all 23 graph-enhanced questions, comes with a 95% Wilson lower bound of roughly 86%. Read together, the descriptive patterns are consistent with plausible architecture-task affinities (Graph RAG for complex-relationship questions, Vector RAG for multi-domain integration, Vanilla RAG for basic-factual queries at lowest cost and latency), but they do not establish those affinities at conventional levels of statistical significance on this dataset.

We accordingly position the pipeline as a proposed framework rather than a validated one. Its main contribution is structuring the conversation around RAG architecture selection: forcing explicit consideration of question-type composition, latency budget, and cost tolerance, and surfacing the trade-offs that practitioners would otherwise leave implicit. The cost numbers in this paper are estimates derived from token pricing rather than measured dollar spend, and we identify the absence of a no-RAG baseline, the small graph-enhanced subset, and single-judge automated scoring as the principal limitations of the present study.

Future work, adding a no-RAG baseline, expanding the graph-enhanced subset, reporting measured operational costs, and running multi-trial evaluations, will determine how much of the proposed framework survives more rigorous testing. We hope the present paper is useful as a starting point: a structured way to think about a choice that is otherwise often made by default.

Funding

Funded by the European Union under Horizon Europe project ChatMED, Bridging Research Institutions to Catalyze Generative AI Adoption by the Health Sector in the Widening Countries (grant agreement ID: 101159214).

Acknowledgements

Views and opinions expressed are those of the authors only and do not necessarily reflect those of the European Union or the European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] Singhal, K., Azizi, S., Tu, T., Mahdavi, S.S., Wei, J., Chung, H.W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., *et al.*: Large language models encode clinical knowledge. Nature **620**(7972), 172–180 (2023)

[2] Yang, R., Ning, Y., Keppo, E., Liu, M., Hong, C., Bitterman, D.S., Ong, J.C.L., Ting, D.S.W., Liu, N.: Retrieval-augmented generation for generative artificial intelligence in health care. npj Health Systems **2**(1), 2 (2025)

[3] Cook, D.A., Hatala, R.: Validation of educational assessments: a primer for simulation and beyond. Advances in simulation **1**(1), 31 (2016)

[4] Downing, S.M.: Validity: on meaningful interpretation of assessment data. Medical Education **37**(9), 830–837 (2003)

[5] 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)

[6] Kohandel Gargari, O., Habibi, G.: Enhancing medical ai with retrieval-augmented generation: A mini narrative review. Digital Health **11**, 20552076251337177 (2025)

[7] Singhal, K., Tu, T., Gottweis, J., Sayres, R., Wulczyn, E., Hou, L., Clark, K., Pfohl, S., Cole-Lewis, H., Neal, D., et al.: Towards expert-level medical question answering with large language models. arXiv preprint arXiv:2305.09617 (2023)

[8] Zhang, H., Chen, J., Jiang, F., Yu, F., Chen, Z., Li, J., Chen, G., Wu, X., Zhang, Z., Xiao, Q., et al.: Huatuogpt, towards taming language model to be a doctor. arXiv preprint arXiv:2305.15075 (2023)

[9] Li, Y., Li, Z., Zhang, K., Dan, R., Jiang, S., Zhang, Y.: Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge. Cureus **15**(6), 40895 (2023)

[10] Pal, A., Umapathi, L.K., Sankarasubbu, M.: Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In: Proceedings of the Conference on Health, Inference, and Learning, pp. 248–260 (2022)

[11] Gao, Y., Li, R., Croxford, E., Caskey, J., Patterson, B.W., Churpek, M., Miller, T., Dligach, D., Afshar, M.: Leveraging medical knowledge graphs into large language models for diagnosis prediction: design and application study. JMIR AI **4**, 58670 (2025)

[12] Yang, R., Chen, Q., Li, Q., et al.: Kg-rank: Enhancing large language models for medical qa with knowledge graphs and ranking techniques. arXiv preprint arXiv:2403.05881 (2024)

[13] Zhao, X., Liu, S., Yang, S.-Y., Miao, C.: Medrag: Enhancing retrieval-augmented generation with knowledge graph-elicited reasoning for healthcare copilot. In: Proceedings of the ACM on Web Conference 2025, pp. 4442–4457 (2025)

[14] Yasunaga, M., Leskovec, J., Liang, P.: Linkbert: Pretraining language models with document links. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, pp. 8003–8016 (2022)

[15] Shang, Y., Tian, Y., Lyu, K., Zhou, T., Zhang, P., Chen, J., Li, J.: Electronic health record–oriented knowledge graph system for collaborative clinical decision support using multicenter fragmented medical data: design and application study. Journal of Medical Internet Research **26**, 54263 (2024)

[16] Wang, S., Zhao, Z., Ouyang, X., Wang, Q., Shen, D.: Chatcad: Interactive

computer-aided diagnosis on medical image using large language models. arXiv preprint arXiv:2302.07257 (2023)

[17] Jin, Q., Dhingra, B., Liu, Z., Cohen, W., Lu, X.: Pubmedqa: A dataset for biomedical research question answering. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, pp. 2567–2577 (2019)

[18] Xiong, G., Jin, Q., Lu, Z., Zhang, A.: Benchmarking retrieval-augmented generation for medicine. In: Findings of the Association for Computational Linguistics: ACL 2024, pp. 6233–6251 (2024)

[19] Ben Abacha, A., Shivade, C., Demner-Fushman, D.: Overview of the mediqa 2019 shared task on textual inference, question entailment and question answering. In: Proceedings of the 18th BioNLP Workshop, pp. 370–379 (2019)

[20] Bunnell, D.J., Bondy, M.J., Fromtling, L.M., Ludeman, E., Gourab, K.: Bridging ai and healthcare: A scoping review of retrieval-augmented generation—ethics, bias, transparency, improvements, and applications. medRxiv (2025)

[21] Strubell, E., Ganesh, A., McCallum, A.: Energy and policy considerations for deep learning in nlp. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3645–3650 (2019)

[22] Thompson, N.C., Greenewald, K., Lee, K., Manso, G.F.: The computational limits of deep learning. arXiv preprint arXiv:2007.05558 (2020)

[23] Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., Huang, X.: Pre-trained models for natural language processing: A survey. Science China Technological Sciences **63**(10), 1872–1897 (2020)

[24] Luo, R., Sun, L., Xia, Y., Qin, T., Zhang, S., Poon, H., Liu, T.-Y.: Biogpt: generative pre-trained transformer for biomedical text generation and mining. Briefings in Bioinformatics **23**(6), 409 (2022)

[25] Thirunavukarasu, A.J., Ting, D.S.W., Elangovan, K., Gutierrez, L., Tan, T.F., Ting, D.S.: Large language models in medicine. Nature Medicine **29**(8), 1930–1940 (2023)

[26] Rasmy, L., Xiang, Y., Xie, Z., Tao, C., Zhi, D.: Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. npj Digital Medicine **4**(1), 86 (2021)

[27] Wu, J., Zhu, J., Qi, Y.: Medical graph rag: Towards safe medical large language model via graph retrieval-augmented generation. arXiv preprint arXiv:2408.04187 (2024)

[28] Dobreva, J., Karasmanakis, I., Ivanisevic, F., Horvat, T., Kitanovski, D.,

Gams, M., Mishev, K., Misheva, M.S.: Ragcare-qa: A benchmark dataset for evaluating retrieval-augmented generation pipelines in theoretical medical knowledge. medRxiv (2025) https://doi.org/10.1101/2025.08.15.25333718 https://www.medrxiv.org/content/early/2025/08/16/2025.08.15.25333718.full.pdf

[29] Team, A.R.: How to evaluate the rag pipeline cost? Adasci Analytics (2024). Available online: https://adasci.org/how-to-evaluate-the-rag-pipeline-cost/

[30] Team, Z.: How to calculate the total cost of your rag-based solutions. Zilliz Blog (2025). Available online: https://zilliz.com/blog/how-to-calculate-the-total-cost-of-your-rag-based-solutions

[31] Tyagi, A.: Key rag techniques: Benefits, costs, applications. Ankur's Newsletter (2025). Available online: https://www.ankursnewsletter.com/p/key-rag-techniques-benefits-costs

[32] Team, M.A.A.S.: Graphrag costs explained: What you need to know. Microsoft Community Hub (2024). Available online: https://techcommunity.microsoft.com/blog/azure-ai-services-blog/graphrag-costs-explained-what-you-need-to-know/4207978

[33] Team, I.R.: What is rag (retrieval augmented generation)? IBM Think (2025). Available online: https://www.ibm.com/think/topics/retrieval-augmented-generation