Proceedings of Machine Learning Research **318** (2026)
The 39th Canadian Conference on Artificial Intelligence

Canadian AI 2026
25–29 May 2026, Vancouver, BC

Reason and Verify: A Framework for Faithful Retrieval-Augmented Generation

Eeham Khan[†], Luis Rodriguez[‡], Marc Queudot[‡,*]
[†] Concordia University, Montréal, QC, Canada
[‡] Centre de recherche informatique de Montréal (CRIM), Montréal, QC, Canada

Abstract

Retrieval-Augmented Generation (RAG) significantly improves the factuality of Large Language Models (LLMs), yet standard pipelines often lack mechanisms to verify intermediate reasoning, leaving them vulnerable to hallucinations in high-stakes domains. To address this, we propose a domain-specific RAG framework that integrates explicit reasoning and faithfulness verification. Our architecture augments standard retrieval with neural query rewriting, BGE-based cross-encoder reranking, and a rationale generation module that grounds sub-claims in specific evidence spans. We further introduce an eight-category verification taxonomy that enables fine-grained assessment of rationale faithfulness, distinguishing between explicit and implicit support patterns to facilitate structured error diagnosis. We evaluate this framework on the BioASQ and PubMedQA benchmarks, specifically analyzing the impact of dynamic in-context learning and reranking under constrained token budgets. Experiments demonstrate that explicit rationale generation improves accuracy over vanilla RAG baselines, while dynamic demonstration selection combined with robust reranking yields further gains in few-shot settings. Using Llama-3-8B-Instruct, our approach achieves 89.1% on BioASQ-Y/N and 73.0% on PubMedQA, competitive with systems using significantly larger models. Additionally, we perform a pilot study combining human expert assessment with LLM-based verification to explore how explicit rationale generation improves system transparency and enables more detailed diagnosis of retrieval failures in biomedical question answering.

Keywords: retrieval-augmented generation, biomedical question answering, rationale generation, faithfulness verification, in-context learning

1. Introduction

Large language models (LLMs) have become powerful general-purpose systems for many NLP tasks, including summarization, question answering, translation, code generation, and dialogue. Their progress is largely driven by transformer architectures and large-scale pre-training on diverse corpora [1, 2]. However, LLMs often struggle with factual accuracy, especially in specialized or fast-evolving domains such as medicine, law, or finance [3, 4]. Because their knowledge is fixed at pre-training time and limited by the coverage and quality of the training data, they can produce outdated or incorrect outputs when faced with novel or domain-specific information.

Retrieval-augmented generation (RAG) addresses this limitation by combining parametric knowledge in model weights with non-parametric knowledge from external corpora [5]. In a RAG pipeline, relevant passages are retrieved from a knowledge base and provided as context to the generator, aiming to ground responses in verifiable evidence and improve factuality and interpretability.

Despite substantial progress, RAG systems face persistent challenges. First, end-to-end performance is highly sensitive to retrieval quality: even small retrieval errors can propagate into generation mistakes [6]. Second, many RAG systems lack explicit reasoning and verification steps, making them vulnerable to subtle hallucinations (e.g., incorrect dates or conflated

[*] Corresponding author: marc.queudot@crim.ca

entities) even when relevant evidence is retrieved [7]. Third, domain deployments often require specialized taxonomies, lexicons, and continually updated corpora, which generic RAG frameworks under-support and which can lead to degraded reliability out of distribution [8].

To address these gaps, we propose a domain-specific RAG framework with explicit reasoning and verification, extending the framework proposed by [9]. Our approach combines BM25 retrieval with neural query rewriting, BGE-based reranking [10], and rationale generation with verification.

In this paper, our main contributions are:

(1) A reproducible domain-specific RAG blueprint with explicit verification gates. We present a biomedical RAG pipeline that integrates retrieval, reranking, rationale generation, and verification into a modular workflow, and empirically evaluate the impact of reranking and dynamic demonstration selection.
(2) A practical, statement-level faithfulness framework for biomedical rationales. We propose and operationalize a verification taxonomy for rationale statements grounded in retrieved abstracts, enabling structured auditing of faithfulness and clearer attribution of errors to retrieval vs. generation.
(3) A systematic evaluation of design choices under token and latency constraints. We run controlled experiments isolating the effects of reranking and dynamic demonstration selection. In a preliminary analysis, we use our hybrid verification framework to categorize critical failure modes, distinguishing between implicit and explicit support patterns to inform future rationale generation research.

2. Related Work

2.1. Retrieval-Augmented Generation

Retrieval-augmented generation (RAG) has emerged as a powerful mechanism for improving the factual accuracy of language models by conditioning them on external corpora at inference time. Early work by [5] demonstrated that a simple retrieve-and-generate pipeline can significantly boost performance for question-answering tasks by retrieving relevant passages from an external knowledge base. This approach was further refined by introducing joint training of the retriever and generator, yielding improved end-to-end retrieval precision and generation quality [6].

2.2. Explicit Reasoning and Rationale Generation

Beyond simply retrieving documents, recent efforts have focused on exposing the model's reasoning process to improve transparency and trust. [7] proposed TrustworthyRAG, which augments RAG with explicit rationale generation heads that tie each token in the output back to source passages, enabling more precise faithfulness checks. Such structured rationales aid with error analysis and allow downstream modules to verify whether each claim is supported by retrieved evidence.

2.3. Domain-Specific Retrieval

Generic RAG systems often struggle in specialized domains, such as medicine or law, where terminology and factual requirements are more stringent. [8] studied the limitations of out-of-the-box LLMs in healthcare applications, showing that domain-tuned retrievers coupled with domain-specific corpora can yield substantial gains in both relevance and factuality of generated outputs. Complementary work has investigated query rewriting and

reranking strategies to further adapt retrieval to domain-specific vocabularies and document structures [11].

3. Methodology

3.1. Framework Architecture

We extend the InstructRAG pipeline [9] to target two persistent RAG failure modes: irrelevant retrieval and hallucinated reasoning. We do so by (i) improving the retrieved evidence set presented to the generator and (ii) making the reasoning process explicit, verifiable, and corrigible. An overview of the full control flow is provided in Algorithm 1, which summarizes retrieval, reranking, optional query rewriting, rationale generation, and statement-level verification. We describe each module and the control logic that ties them together below.

BM25 Retriever. We use the BM25 algorithm to quickly gather a broad initial set of potentially relevant documents. Given a user query q, the retriever returns the top k=20 passages ${p_i}_{i=1}^{20}$. BM25's efficiency and robustness to domain-specific lexical cues make it a reliable starting point in specialized corpora.

BGE Cross-Encoder Reranker. To enhance evidential precision, we rerank the top 20 passages retrieved by BM25 with a BGE cross-encoder. For each candidate passage p_i, the reranker jointly encodes the query–passage pair (q, p_i) and assigns a relevance score that reflects deeper semantic alignment beyond lexical overlap. We then select the top m=5 passages to form the final evidence set E used in downstream reasoning.

Query Rewriter. We use a GPT-4o module to clarify ambiguous queries q by expanding acronyms and adding precise medical terminology. To avoid unnecessary delays, this step is optional: it is only triggered when the initial retrieval results lack significant keyword overlap with the query or when the reranker finds insufficient evidence.

Rationale Generator. We prompt the generator to produce a concise, evidence-linked rationale R alongside a provisional answer $\hat{y}$. Prompts include the retrieved passages p_i and instructions to (i) decompose the question into sub-claims, (ii) cite the specific passage IDs (and if available, character/token spans) that support each sub-claim, and (iii) refrain from using knowledge not present in p_i.

Rationale Verifier. Building on the Factual Evidence framework [12], we use GPT-4o to classify rationale statements based on their factual alignment with retrieved documents. Each statement is categorized as correct (explicit, implicit, additional info, or missing context) or incorrect (false info, deviating info, illogical, or missing evidence), enabling fine-grained faithfulness assessment.

3.2. Corpus

Our knowledge base was sourced from the MedRAG toolkit [13], which includes PubMed abstracts (23M+ biomedical literature abstracts), Wikipedia articles (general medical background), medical textbooks (authoritative exam-style content), and StatPearls (clinical decision support summaries).

For our experiments, we exclusively used the PubMed abstracts corpus, as both evaluation datasets (BioASQ and PubMedQA) are specifically designed for literature-based question

Algorithm 1 Domain-Specific RAG with Explicit Reasoning and Verification

Require: user query q; corpus index $\mathcal{D}$; BM25 top-k; rerank top-m; rewrite threshold τ_{ovlp}; evidence threshold τ_{evid}
Ensure: final answer y, rationale R, evidence set E, verification V
Retrieve: $C \leftarrow \text{BM25}(\mathcal{D}, q, k)$ ▷ Initial candidate set
Rerank: $E \leftarrow \text{BGE\_Rerank}(q, C, m)$ ▷ Top-m evidence passages
Rewrite trigger: compute lexical overlap $s \leftarrow \text{Overlap}(q, E)$
compute evidence score $e \leftarrow \text{EvidenceScore}(q, E)$
if $s < \tau_{\text{ovlp}} \ \vee \ e < \tau_{\text{evid}}$ **then**
    $q' \leftarrow \text{Rewrite}(q)$ ▷ Expand acronyms, add medical terms
    $C \leftarrow \text{BM25}(\mathcal{D}, q', k)$
    $E \leftarrow \text{BGE\_Rerank}(q', C, m)$
end if
Reason: $(\hat{y}, R) \leftarrow \text{GenerateAnswerAndRationale}(q, E)$
Verify: $V \leftarrow \text{VerifyRationale}(q, E, R)$ ▷ 8-category faithfulness labels
return $y = \hat{y}$, R, E, V

Implementation details: We set k=20 and m=5 in all experiments. **Overlap** computes the fraction of non-stopword query tokens appearing in the concatenated evidence passages. **EvidenceScore** is the mean reranker score of the top-m passages. **GenerateAnswerAndRationale** prompts Llama-3-8B-Instruct with retrieved evidence and (optionally) dynamically selected demonstrations to produce an evidence-linked rationale R and provisional answer $\hat{y}$. **VerifyRationale** uses GPT-4o to label each rationale statement according to the categories in Table 1.

answering and their questions are answerable from scientific abstracts. This focused approach enables direct assessment of our framework's retrieval and reasoning capabilities on domain-appropriate literature without introducing noise from heterogeneous sources.

3.3. Datasets

We evaluated our framework on two biomedical QA datasets from the MIRAGE benchmark [13]: BioASQ [14] and PubMedQA [15]. Both are knowledge-intensive and require domain-specific literature access.

BioASQ contains expert-curated yes/no questions requiring literature-grounded reasoning to verify biomedical claims. PubMedQA provides research questions with three candidate answers (yes/no/maybe) answerable from PubMed abstracts. Following MIRAGE settings, we use question-only retrieval without gold passage supervision, simulating realistic medical information-seeking scenarios.

3.4. Metrics

We evaluate our framework using a combination of automatic and human-centric metrics to assess factual accuracy, rationale quality, and inter-annotator agreement for our LLM-as-a-judge components.

For both BioASQ and PubMedQA, we evaluate model performance using classification accuracy. BioASQ is a binary task (`yes` / `no`), whereas PubMedQA is a three-way classification (`yes` / `no` / `maybe`).

To measure the alignment between generated rationales and their corresponding retrieved context passages, we use a custom verification schema inspired by the MIRAGE benchmark [13]. Each context document rationale is classified into one of eight categories described in Table 1. We then assigned a binary `CORRECT` or `INCORRECT` verdict for the entire rationale.

Table 1. Legend for Rationale-Verification Label Categories

Category	Meaning
`CORRECT-EXPLICIT`	Information is explicitly stated in the documents (quoted or paraphrased).
`CORRECT-IMPLICIT`	Facts are not stated verbatim but are logically inferred from context clues.
`CORRECT-ADDITIONAL`	Uses context accurately but adds relevant, correct external details.
`CORRECT-MISSING`	Conclusion is correct, but the cited documents provide no support (irrelevant).
`INCORRECT-FALSE`	Statements directly contradict evidence provided in the context.
`INCORRECT-DEVIATING`	Statements are off-topic or unrelated to the query/documents.
`INCORRECT-ILLOGICAL`	Reasoning contains internal contradictions or violates logic/scientific principles.
`INCORRECT-MISSING`	Reasoning is incorrect *and* the cited documents are irrelevant.

For a generated rationale R, we segment it into n atomic statements $\{r_j\}_{j=1}^{n}$ and assign each statement a label from Table 1. We define an *atomic statement* as a single verifiable proposition that can be supported or refuted using the retrieved evidence (e.g., one clinical claim, one association, one numerical outcome). In practice, we segment R by sentence boundaries and further split sentences on clause markers (e.g., "because", "therefore", "however", "which suggests") and list delimiters (e.g., semicolons, enumerations) to isolate minimal factual units. We merge fragments that are not semantically self-contained (e.g., pronoun-only continuations) with the preceding clause. This segmentation procedure is deterministic and is applied identically for human annotation and the LLM verifier. We define a binary support indicator $\mathbb{I}_j$ that equals 1 if r_j is labeled as any `CORRECT-*` category and 0 otherwise (i.e., any `INCORRECT-*`). The faithfulness score is the proportion of supported statements:

$$\text{Faith}(R) \;=\; \frac{1}{n}\sum_{j=1}^{n}\mathbb{I}_j. \tag{3.1}$$

To quantify the consistency of rationale annotations, we compute two complementary metrics for each context condition (CONTEXT-1 through CONTEXT-5). CONTEXT-j denotes verification using only the top-j retrieved passages (for $j \in \{1, \ldots, 5\}$):

- **Cohen's κ.** We use Cohen's κ to measure chance-corrected agreement. We report κ for (i) human–human agreement and (ii) each human annotator versus the LLM verifier. We interpret κ using the Landis–Koch guidelines [16]: slight (0.00–0.20), fair (0.21–0.40), moderate (0.41–0.60), substantial (0.61–0.80), and almost perfect (0.81–1.00).
- **Per-category $\mathbf{F_1}$.** For each rationale label category, we compute precision and recall and report the corresponding per-category F_1 score. When comparing humans to the LLM verifier, we treat the human label as the reference and compute F_1 for the verifier on each category.

By reporting both κ and F_1 for human–human and human–model pairs under every context, we can obtain a nuanced view of annotation reliability and model alignment.

4. Experiments

4.1. Experimental Design

Our experimental framework is designed to isolate and evaluate the specific contributions of retrieval, reranking, and explicit reasoning verification in biomedical question answering. We conduct a series of controlled experiments to assess system performance under varying constraints.

Demonstration Pool Construction ($T^\star$). We construct a static pool of in-context demonstrations, $T^\star$, *offline* using exclusively the training split of each dataset. For each training example (q_i, y_i), we retrieve evidence passages from the same PubMed index used at test time and generate an evidence-linked rationale r_i using the same prompt format as in evaluation. Each demonstration is stored as a tuple (q_i, r_i, y_i, E_i), where E_i denotes the set of top-m retrieved passages. Crucially, the rationales in $T^\star$ are model-generated rather than human-curated; this ensures that demonstrations reflect the model's achievable reasoning patterns rather than potentially unattainable gold-standard explanations.

Dynamic Demonstration Selection. At inference time, we select the top-k demonstrations from $T^\star$ based on cosine similarity between the embedding of the evaluation query and the stored training-query embeddings. Our primary policy is similarity-based; however, we note the risk of label-prior bias, particularly for the ternary classification in PubMedQA (*yes/no/maybe*). We therefore evaluate both (i) a strict *similarity-only* policy and (ii) a *label-balanced* policy that enforces class diversity among the retrieved demonstrations. Unless otherwise stated, we report results under the similarity-only setting.

Data Decontamination. To ensure robust evaluation, we enforce strict decontamination protocols. The demonstration pool $T^\star$ is disjoint from all validation and test splits. We further mitigate leakage by performing embedding-based deduplication to remove near-duplicate questions across splits. Additionally, when document identifiers are available (e.g., PubMed IDs), we filter out any demonstration that shares an identifier with the current evaluation query, preventing the model from exploiting memorized document–answer associations. Random seeds are fixed for all fallback sampling procedures.

4.1.1. Comparative Approaches

We evaluate the proposed framework against the following configurations:

- **Vanilla RAG**: A baseline that retrieves documents using BM25 and generates answers directly, lacking an explicit intermediate reasoning step.
- **InstructRAG-ICL**: Our proposed baseline that incorporates a distinct reasoning generation step via in-context learning prior to producing the final answer. This entails using examples of similar tasks (demonstrations).
- **InstructRAG w/ Reranker**: An enhancement of the ICL approach that applies a BGE-v2-m3 cross-encoder to rerank the initial retrieval candidates, improving the precision of the evidence E.
- **InstructRAG w/ Reranker + Dynamic ICL**: Extends the reranker model by replacing static demonstrations with dynamically selected examples via k-nearest neighbor (KNN) search.

4.1.2. Number Of Demonstrations

To understand the impact of demonstration examples on reasoning quality and overall performance, we tested configurations ranging from 0 to 4 samples for few-shot learning to evaluate the trade-off between demonstration quantity and context window utilization.

For each evaluation query, we (i) computed cosine similarity between the query vector and all training vectors, (ii) selected the top-k most similar examples while enforcing a diversity constraint, and (iii) ensured a minimum unique example count through random selection if needed.

4.1.3. **Query Rewriting**

The optional query rewriting module is triggered when lexical overlap falls below $\tau_{\text{ovlp}} = 0.3$ or when the mean reranker score is below $\tau_{\text{evid}} = 0.5$. These thresholds were set once based on manual inspection of 50 training queries and were not tuned further. In our evaluation, query rewriting was triggered in approximately 8% of BioASQ queries and 12% of PubMedQA queries. We did not ablate the rewriting component in isolation, as our primary focus was on reranking and demonstration selection; a systematic study of query rewriting strategies and threshold sensitivity is left for future work.

4.1.4. **Reranking Experiments**

We evaluated the impact of the BGE reranker by comparing performance across all ICL variations both with and without the reranking component. For the reranking experiments, we (i) retrieved an initial set of 20 candidate documents using BM25, (ii) applied the BGE cross-encoder to rerank these candidates, and (iii) selected the top-5 documents after reranking for rationale generation.

We chose 20 initial candidates and kept the top-5 after reranking as an empirically validated, budget-aware setting that matches common two-stage retrieval practice and preserved our token/latency constraints.

4.1.5. **Retrieval Method**

For document retrieval, we initially used the MedCPT retriever. However, to simplify our experimental pipeline and conserve computational resources, we transitioned to BM25. This change was supported by similar performance metrics between BM25 and more complex retrievers in our preliminary evaluations.

4.1.6. **Model**

For all experiments, we used Llama-3-8B-Instruct for both reasoning generation (in the case of InstructRAG-ICL) and answer generation (for both InstructRAG-ICL and Vanilla RAG) (see subsection 4.1.1)

4.2. **Human Evaluation Setup**

To assess the quality of generated rationales and validate our automatic evaluation metrics, we designed a human annotation study with the following components:

4.2.1. **Annotation Protocol**

Two annotators with extensive experience in NLP systems independently evaluated a convenience sample of 4 examples (2 per dataset). While this sample size precludes statistical inference, it allowed us to (i) validate the clarity of our 8-category taxonomy, (ii) identify systematic differences between human and LLM judgments, and (iii) surface failure modes for qualitative analysis. For each example, annotators evaluated:

- **Overall Response Quality**: A 1-5 scale rating the comprehensiveness and correctness of the final answer.
- **Rationale-Context Alignment**: Classifying each reasoning statement according to its faithfulness to the retrieved documents using eight predefined categories.
- **Faithfulness Score**: A numeric score on how much the response is grounded in the provided context.

4.2.2. LLM-as-a-Judge Implementation

In parallel with human annotation, we implemented an automated evaluation using a large language model (GPT-4o) to assess the system output on the same examples used for the human evaluation. The automated judge was provided with: (i) the original question, (ii) retrieved context documents, (iii) the generated rationale, and (iv) the generated answer. The LLM judge was instructed to evaluate using the same criteria as human annotators.

5. Results

We present our experimental findings in three parts: (1) comparison with published baselines, (2) ablation analysis of our framework components, and (3) analysis of demonstration selection strategies.

5.1. Comparison with Baselines

Table 2 compares our configurations against published MIRAGE baselines. Despite using Llama-3-8B-Instruct—a model roughly 10× smaller than GPT-4—our framework achieves competitive or superior performance.

On BioASQ-Y/N, our best configuration (3-shot Dynamic ICL with reranking) achieves 89.1% accuracy, approaching MedRAG+GPT-3.5 (90.29%). On PubMedQA*, our 0-shot rationale generation achieves **73.0%**, exceeding MedRAG+GPT-4 (70.60%) by 2.4 points absolute. We attribute this to explicit rationale generation: by requiring the model to articulate evidence-linked reasoning before answering, we reduce hallucinated inferences that may arise in larger models operating without such structure.

Notably, while MedRAG uses a four-corpus ensemble (MedCorp) with reciprocal rank fusion over four retrievers (RRF-4), our system uses only PubMed with BM25 + BGE reranking—a substantially simpler retrieval pipeline. This suggests that reasoning-side improvements (rationale generation, dynamic ICL) can compensate for retrieval complexity.

5.2. Ablation Study

Table 3 isolates the contribution of each framework component. We report results across different numbers of in-context demonstrations to understand interactions between components.

5.2.1. Effect of Reranking

Reranking with the BGE cross-encoder provides consistent benefits, particularly for PubMedQA in few-shot settings. The most dramatic improvement occurs in the 4-shot PubMedQA setting, where reranking improves accuracy from 47.5% to 60.0% (+12.5 points). This suggests that reranking helps filter out noisy passages that would otherwise mislead the model when combined with multiple demonstrations.

For BioASQ, reranking yields modest but consistent gains in the 0–2 shot range (+0.9 to +1.6 points). The slight degradation at 3-shot (-1.3 points) may indicate that high-quality demonstrations can partially compensate for retrieval noise.

5.2.2. Effect of Dynamic Demonstration Selection

Dynamic ICL selection via KNN retrieval substantially outperforms static demonstration selection across all few-shot configurations. The improvements are particularly pronounced for BioASQ, where dynamic selection at 4-shot yields +14.5 points over static selection (86.2% vs. 71.7%).

Table 2. Comparison with baseline methods under MIRAGE settings. Accuracy (%) reported on **BioASQ-Y/N** (618 yes/no questions) and **PubMedQA*** (500 questions, context removed). All baselines use question-only retrieval without gold passages. †Results from MIRAGE benchmark [13].

Method	BioASQ-Y/N	PubMedQA*
Closed-Book (No Retrieval; CoT Prompting)		
GPT-3.5†	74.27	36.00
GPT-4†	84.30	39.60
Mixtral-8×7B†	77.51	35.20
Llama2-70B†	61.17	42.20
MEDITRON-70B†	68.45	53.40
PMC-LLaMA-13B†	63.11	55.80
MedRAG (MedCorp + RRF-4 Retriever)		
+ GPT-3.5†	<u>90.29</u>	67.40
+ GPT-4†	**92.56**	70.60
+ Mixtral-8×7B†	87.54	67.60
+ Llama2-70B†	73.95	50.40
+ MEDITRON-70B†	76.86	56.40
MedRAG Variants (GPT-3.5 backbone)		
PubMed + BM25†	88.51	66.20
PubMed + MedCPT†	85.76	66.40
MedCorp + BM25†	87.70	66.20
MedCorp + RRF-2†	88.19	67.80
Ours (Llama-3-8B-Instruct backbone)		
Vanilla RAG (BM25)	82.3	70.0
+ Rationale Gen. (0-shot)	85.8	**73.0**
+ Reranking (0-shot)	87.4	<u>72.5</u>
+ Dynamic ICL (best-k)	89.1	71.0

Notes: Best-k = 3-shot for BioASQ-Y/N, 2-shot for PubMedQA*. Best results per section in **bold**; second best <u>underlined</u>. MedCorp combines PubMed, StatPearls, Textbooks, and Wikipedia corpora. RRF-n denotes Reciprocal Rank Fusion over n retrievers.

Table 3. Ablation study showing the impact of reranking and demonstration selection strategy. Accuracy (%) reported across varying numbers of ICL demonstrations.

	BioASQ					PubMedQA				
Configuration	**0**	**1**	**2**	**3**	**4**	**0**	**1**	**2**	**3**	**4**
Static Demonstration Selection										
w/o Reranking	85.8	78.9	78.5	77.7	70.4	73.0	54.0	60.0	56.0	47.5
w/ Reranking	87.4	79.8	79.6	76.4	71.7	72.5	60.5	60.0	60.0	60.0
Δ (Reranking)	+1.6	+0.9	+1.1	-1.3	+1.3	-0.5	+6.5	–	+4.0	+12.5
Dynamic Demonstration Selection (w/ Reranking)										
Dynamic ICL	87.4	88.7	88.3	**89.1**	86.2	72.5	65.0	**71.0**	66.5	69.0
Δ vs. Static	–	+8.9	+8.7	+12.7	+14.5	–	+4.5	+11.0	+6.5	+9.0

Note: 0-shot results are identical for static and dynamic selection (no demonstrations used). Δ shows improvement from the ablated component. Best results per dataset in **bold**.

Notably, dynamic selection reverses the degradation pattern observed with static ICL. Under static selection, adding more demonstrations *hurts* performance on both datasets

(BioASQ drops from 85.8% at 0-shot to 70.4% at 4-shot). With dynamic selection, performance remains stable or improves, peaking at 3-shot for BioASQ (89.1%) and 2-shot for PubMedQA (71.0%).

5.3. **Analysis of Demonstration Sensitivity**

PubMedQA exhibits higher sensitivity to demonstration selection than BioASQ. We attribute this to two factors: (i) the ternary classification structure (yes/no/maybe) creates label-prior bias when demonstrations are not carefully balanced, and (ii) PubMed abstracts are longer, causing additional demonstrations to compete with retrieved evidence for context window space.

6. **Limitations**

Despite promising results on biomedical question answering, our study has several limitations. First, evaluation is limited to two English biomedical datasets (BioASQ and PubMedQA), which may reduce generalizability to other domains, languages, or question formats. Both benchmarks largely emphasize factoid and binary judgments; performance on more complex settings (e.g., multi-hop or causal reasoning) remains untested.

Second, parts of our pipeline rely on OpenAI API calls, introducing additional latency and monetary cost that may hinder deployment at scale. Exploring open-source substitutes for these components is an important direction for future work.

Third, we do not evaluate in real clinical workflows or with clinicians in the loop. Our system is intended as a research prototype for studying rationale generation and verification rather than a clinical decision support tool. Any high-stakes deployment would require substantially more validation and safety measures.

Fourth, we report single-point accuracy estimates without statistical significance testing or confidence intervals. While our ablation design isolates component contributions, the magnitude of improvements should be interpreted with appropriate caution until replicated across multiple runs.

Finally, our human evaluation is small (4 examples annotated by 2 raters). Larger studies are needed to reliably assess rationale quality, agreement, and the practical utility of explicit reasoning for biomedical QA.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention is All You Need". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017, pp. 5998–6008. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[2] T. Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[3] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung. "Survey of Hallucination in Natural Language Generation". In: *ACM Computing Surveys* 55.12 (Mar. 2023), pp. 1–38. DOI: 10.1145/3571730. URL: http://dx.doi.org/10.1145/3571730.

[4] K. Shuster, S. Poff, M. Chen, D. Kiela, and J. Weston. "Retrieval Augmentation Reduces Hallucination in Conversation". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3784–3803. DOI: 10.18653/v1/2021.findings-emnlp.320. URL: https://aclanthology.org/2021.findings-emnlp.320/.

[5] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 9459–9474. URL: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

[6] G. Izacard and E. Grave. "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Online: Association for Computational Linguistics, Apr. 2021, pp. 874–880. DOI: 10.18653/v1/2021.eacl-main.74. URL: https://aclanthology.org/2021.eacl-main.74/.

[7] Y. Zhou, Y. Liu, X. Li, J. Jin, H. Qian, Z. Liu, C. Li, Z. Dou, T.-Y. Ho, and P. S. Yu. *Trustworthiness in Retrieval-Augmented Generation Systems: A Survey*. 2024. arXiv: 2409.10102 [cs.IR]. URL: https://arxiv.org/abs/2409.10102.

[8] K. Singhal et al. "Large language models encode clinical knowledge". In: *Nature* 620 (July 2023), pp. 172–180. DOI: 10.1038/s41586-023-06291-2. URL: https://doi.org/10.1038/s41586-023-06291-2.

[9] Z. Wei, W.-L. Chen, and Y. Meng. "InstructRAG: Instructing Retrieval-Augmented Generation via Self-Synthesized Rationales". In: *Proceedings of the Thirteenth International Conference on Learning Representations*. 2025. URL: https://openreview.net/forum?id=P1qhkp8gQT.

[10] S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, D. Lian, and J.-Y. Nie. "C-Pack: Packed Resources For General Chinese Embeddings". In: *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '24. Washington DC, USA: Association for Computing Machinery, 2024, pp. 641–649. DOI: 10.1145/3626772.3657878. URL: https://doi.org/10.1145/3626772.3657878.

[11] X. Ma, Y. Gong, P. He, H. Zhao, and N. Duan. "Query Rewriting in Retrieval-Augmented Large Language Models". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by H. Bouamor, J. Pino, and K. Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 5303–5315. DOI: 10.18653/v1/2023.emnlp-main.322. URL: https://aclanthology.org/2023.emnlp-main.322/.

[12] I. Muneeswaran, A. Shankar, V. Varun, S. Gopalakrishnan, and V. Vaddina. "Mitigating Factual Inconsistency and Hallucination in Large Language Models". In: *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. WSDM '24. Merida, Mexico: Association for Computing Machinery, 2024, pp. 1169–1170. DOI: 10.1145/3616855.3635744. URL: https://doi.org/10.1145/3616855.3635744.

[13] G. Xiong, Q. Jin, Z. Lu, and A. Zhang. "Benchmarking Retrieval-Augmented Generation for Medicine". In: *Findings of the Association for Computational Linguistics: ACL 2024*. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 6233–6251. DOI: 10.18653/v1/2024.findings-acl.372. URL: https://aclanthology.org/2024.findings-acl.372/.

[14] G. Tsatsaronis, M. Schroeder, G. Paliouras, Y. Almirantis, I. Androutsopoulos, E. Gaussier, P. Gallinari, T. Artieres, M. R. Alvers, M. Zschunke, and A.-C. Ngonga Ngomo. "BioASQ: A Challenge on Large-Scale Biomedical Semantic Indexing and Question Answering". In: *Proceedings of the AAAI Fall Symposium on Information Retrieval and Knowledge Discovery in Biomedical Text*. 2012.

[15] Q. Jin, B. Dhingra, Z. Liu, W. Cohen, and X. Lu. "PubMedQA: A Dataset for Biomedical Research Question Answering". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2567–2577. DOI: 10.18653/v1/D19-1259. URL: https://aclanthology.org/D19-1259/.

[16] J. R. Landis and G. G. Koch. "The Measurement of Observer Agreement for Categorical Data". In: *Biometrics* 33.1 (1977), pp. 159–174. DOI: 10.2307/2529310.

Appendix A. Demo

A functional demonstration of the proposed system is available on HuggingFace[1]. In this interactive application, users can query an AI-related white paper, and the system responds with both a generated answer and a transparent rationale. Specifically, the interface displays the supporting evidence passages retrieved from the document collection and explains how they were integrated into the model's response. This allows users to assess the relevance of the retrieved information and the robustness of the underlying retrieval-augmented generation pipeline.

Appendix B. Pilot Human Study: Per-Example Faithfulness Scores

As an illustrative complement to our pilot human evaluation (4 examples total), Table 4 reports the per-example faithfulness scores Faith(R) (Eq. 3.1) assigned by two human annotators and the LLM verifier. We include these values to qualitatively examine agreement patterns and to illustrate cases where the automated verifier appears more strict or more permissive than the human raters. Given the small sample size, these results are descriptive and should not be interpreted as statistically reliable estimates of verifier performance.

Table 4. Pilot per-example faithfulness scores (4 examples). Scores are shown for two human annotators and the LLM verifier.

Example	Annotator A	Annotator B	LLM Verifier
Question 1	1.00	1.00	1.00
Question 2	0.75	0.50	0.83
Question 3	0.75	0.30	0.93
Question 4	0.90	0.80	1.00

The LLM verifier tends to assign higher faithfulness scores than human annotators (mean 0.94 vs. 0.85 and 0.65), suggesting it may be more permissive in accepting implicit reasoning. Human annotators show substantial disagreement on PubMedQA-1 (0.75 vs. 0.30), highlighting the subjectivity inherent in faithfulness assessment and the need for clearer annotation guidelines in future work.

Appendix C. Prompts and Instructions

For reproducibility, we report the prompt templates used in our framework. Prompts differ slightly across datasets due to output format requirements (BioASQ vs. PubMedQA).

C.1. Base Instructions

The following system instruction was prepended to all queries.

> **Task Instruction:** Analyze the provided documents and answer the question. Briefly explain how the documents support your answer. If the documents are not useful, answer from your own knowledge without referencing them.

C.2. Dataset-Specific Instructions

C.2.1. PubMedQA Prompt

PubMedQA requires ternary classification (yes/no/maybe). We appended:

[1] https://huggingface.co/spaces/DialogueRobust/RobustDialogueDemo

Critically evaluate the medical evidence in the documents (methods, sample sizes, statistical significance). Weigh supporting and opposing evidence, note limitations, and provide concise reasoning followed by a final judgment.

OUTPUT FORMAT REQUIREMENT: End your response with exactly one of the following on a new line: "FINAL ANSWER: A. yes"
"FINAL ANSWER: B. no"
"FINAL ANSWER: C. maybe"

If no document supports an answer, output: "ANSWER UNAVAILABLE"

C.2.2. **BioASQ Prompt**

BioASQ requires binary classification (yes/no). We appended:

Provide a precise factual answer grounded in the documents. Extract relevant statements and justify the decision based strictly on presented facts.

OUTPUT FORMAT REQUIREMENT: End your response with one of the following on a new line: "FINAL ANSWER: A. yes"
"FINAL ANSWER: B. no"

If no document supports an answer, output: "ANSWER UNAVAILABLE"