PubMed Reasoner: Dynamic Reasoning-based Retrieval for Evidence-Grounded Biomedical Question Answering

Anonymous ACL submission

Abstract

Trustworthy biomedical question answering (QA) systems must not only provide accurate answers but also justify them with current, verifiable evidence. Retrieval-augmented approaches partially address this gap but lack mechanisms to iteratively refine poor queries, whereas self-reflection methods kick in only after full retrieval is completed. In this context, we introduce PubMed Reasoner, a biomedical QA agent composed of three stages: **self-critic query refinement** evaluates MeSH terms for coverage, alignment, and redundancy to enhance PubMed queries based on partial (metadata) retrieval; **reflective retrieval** processes articles in batches until sufficient evidence is gathered; and **evidence-grounded response generation** produces answers with explicit citations. PubMed Reasoner with a GPT-4o backbone achieves **78.32%** accuracy on PubMedQA, slightly surpassing human experts, and showing consistent gains on MMLU Clinical Knowledge. Moreover, LLM-as-judge evaluations prefer our responses across: reasoning soundness, evidence grounding, clinical relevance, and trustworthiness. By orchestrating retrieval-first reasoning over authoritative sources, our approach provides practical assistance to clinicians and biomedical researchers while controlling compute and token costs.

1 Introduction

Trustworthiness is essential in biomedical domains. Biomedical question answering (QA) systems must be factually grounded, current, and interpretable. Yet, large language models (LLMs) that rely primarily on parametric memory can hallucinate (Kalai et al., 2025), drift out of date, or omit key evidence (Guan et al., 2023; Xu et al., 2024). While prior works have explored retrieval-augmented methods, to our knowledge, this is the first work to explicitly support citation-backed responses on biomedical QA datasets, ensuring that each response is transparently grounded in published evidence.

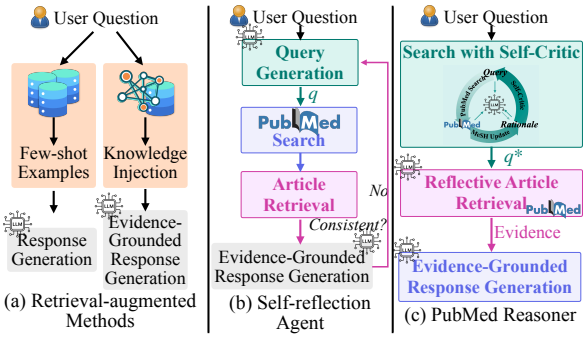

Figure 1: RAG, Self-reflection vs. PubMed Reasoner. **(a) RAG baselines:** uses few-shot exemplars or custom databases but lack retrieval feedback. **(b) Self-reflection agents:** inspired our proposal; generates responses first and only reflects after completion. **(c) PubMed Reasoner:** a search-first approach that performs self-critic query refinement, reflective article retrieval in batches w/ early stopping once evidence is sufficient, and evidence-grounded response generation with explicit citations.

Existing retrieval-augmented generation (RAG) approaches remain limited in biomedical settings (see Fig. 1a): **(i) Few-shot prompting** retrieves a few in-distribution exemplars (few-shot samples) and prompt the LLM to imitate them (Nori et al., 2023b,a). While this can improve accuracy on similar cases, it does not yield structured explanations (Singhal et al., 2025). The link between examples is built via clustering and similarity search, which requires sizable, well-labeled exemplar data. As a result, exemplar RAG is brittle and biased: it optimizes for local pattern matching rather than concept coverage or causal explanations. **(ii) Private knowledge databases** consist of bespoke stores such as entity graphs (Abu-Rasheed et al., 2024) or schema-aligned tables (Arslan et al., 2024) which can support step-wise explanations, but require strong priors and heavy maintenance, with limited reusability across domains. In biomedicine, where PubMed and MeSH (Medical Subject Headings) terms continue to grow, keeping such stores complete and current is prohibitively expensive.

A parallel line of work equips LLMs with web search capabilities ("deep research"), producing responses with citations. Yet, these systems typically lack the ability to constrain retrieval to authoritative biomedical sources such as PubMed, often yielding citations from less reliable or incomplete domains. Recent "self-reflection" agents (Figure 1b) provide another direction, using consistency checks (Wang et al., 2022) or reward-based signals (Leike et al., 2018; Shinn et al., 2023) to refine final answers. However, reflection occurs only after retrieval and response generation are complete, making the process computationally costly and unable to correct poor upstream retrieval or query formulation.

Our approach. We present PubMed Reasoner, a multi-stage agent that mirrors the workflow of a biomedical researcher. Unlike prior methods that reflect only after producing responses, PubMed Reasoner introduces feedback much earlier. During query planning, a self-critic evaluates candidate MeSH terms and their Boolean composition directly against live PubMed metadata, eliminating the need for static private databases. This structured feedback prevents low-quality queries from propagating downstream and balances recall with precision. The refined query is then issued to PubMed, where a reflective retriever processes articles in small batches and halts once sufficient evidence is gathered, controlling token usage. Finally, PubMed Reasoner synthesizes an evidence-grounded response with explicit inline citations, ensuring transparency and interpretability.

Instead of treating the LLM as a one-shot generator, PubMed Reasoner introduces a **dynamic reasoning-based paradigm**, orchestrating three interconnected stages over external evidence. This work makes the following contributions:

- We shift biomedical QA beyond one-shot generation by introducing an iterative workflow that plans, retrieves, and reasons over external evidence.
- We integrate self-critic query refinement with batch-wise reflective retrieval over the PubMed database, enabling robust evidence grounding without maintaining private knowledge stores.
- We demonstrate consistent improvements in accuracy, explanation quality, and computational efficiency on PubMedQA and MMLU Clinical Knowledge, outperforming strong LLM, RAG and self-reflection baselines.

2 Preliminaries

PubMed and MeSH Terms. PubMed is the primary biomedical literature database, indexing over 35 million references. Each article is annotated with Medical Subject Headings (MeSH), a controlled vocabulary that organizes biomedical concepts into a hierarchical taxonomy. Queries often combine MeSH terms with Boolean operators (AND, OR, NOT), enabling structured retrieval.

Biomedical Question Answering. The input is a natural language question Q (e.g., "*Do leukotrienes play a key role in asthma?*"), and the output is a response R that is both factually accurate and explicitly supported by authoritative literature.

Problem Setup. Given a user question Q, an optional task specification T (e.g., "*answer yes/no with a justification*"), and optional context C, the objective is to retrieve a set of relevant biomedical articles A and synthesize a final response R that is evidence-grounded and interpretable.

Search Result Assessment Metrics. To enable iterative query refinement, we adapt evaluation dimensions from search benchmarks (Gao et al., 2013; Jiang et al., 2024) to the biomedical QA setting, defining three structured feedback signals:

- **Coverage:** Does the MeSH term retrieve articles that represent the core biomedical concept?
- **Alignment:** Are the retrieved articles relevant to the original question?
- **Redundancy:** Does the MeSH term overlap with others, reducing retrieval efficiency?

These signals guide improvements to the evolving query, ensuring that downstream reasoning is grounded in a high-quality evidence pool.

3 Proposed Method

We introduce PubMed Reasoner, a multi-stage agent framework inspired by the workflow of a biomedical researcher. Unlike direct LLM-based answering or one-shot retrieval-augmented generation, PubMed Reasoner explicitly separates reasoning into three phases: **(1) Search with Self-Critic Query Refinement**, **(2) Reflective Article Retrieval with Early Stopping**, and **(3) Evidence-Grounded Response Generation**. This design ensures that the system does not rely solely on parametric memory, but grounds its explanation in authoritative biomedical literature.

3.1 Search with Self-Critic Query Refinement

Query Generation. The first stage constructs a structured query that serves as the initial search

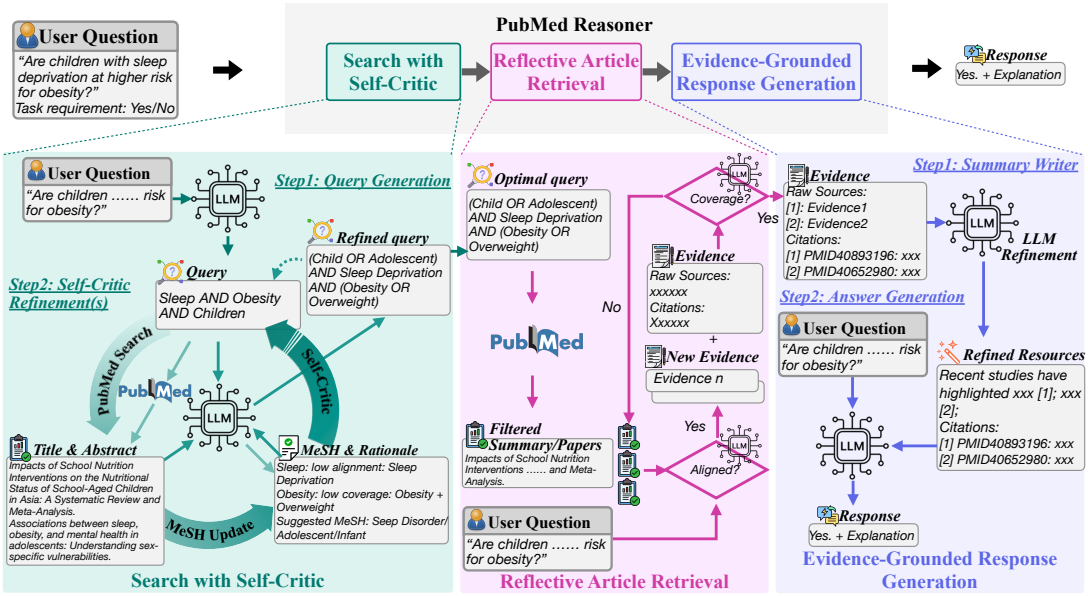

Figure 2: PubMed Reasoner stages. **(1) Search with Self-Critic Query Refinement.** From a user question, MeSH terms and a structured query are proposed. Self-critic evaluates each term for *coverage*, *alignment*, and *redundancy*, iteratively refining the query. **(2) Reflective Article Retrieval with Early Stopping.** PubMed Reasoner queries PubMed, filters results by title/abstract, and extracts supporting evidence in batches, checking whether accumulated evidence sufficiently covers the question; if so, retrieval is terminated early to save tokens and avoid unnecessary processing. **(3) Evidence-Grounded Response Generation.** Retained evidence is synthesized into final answer with explicit inline citations, ensuring factual grounding and traceability. More case studies are provided in App. G.

plan for PubMed retrieval. Given a user question Q grounded in biomedical knowledge, together with optional contextual information C, a large language model (LLM) is employed to generate a broad set of candidate Medical Subject Headings (MeSH) terms. Each candidate term is accompanied by a brief rationale explaining its relevance to the query. Formally, this process is defined as:

$$\mathcal{M}_0 \leftarrow \mathrm{LLM}_{\mathrm{mesh\_gen}}(Q, C), \tag{1}$$

where $\mathcal{M}_0 = \{(\mathrm{MeSH}_i, r_i)\}_{i=1}^{N}$, MeSH_i denotes a candidate MeSH term, and r_i provides the corresponding semantic justification for its inclusion.

From the initial candidate pool $\mathcal{M}_0$, the LLM selects a subset of MeSH terms based on multiple criteria, including the model's confidence for each term, the plausibility of its accompanying rationale, and the degree of semantic alignment between the term and the input question:

$$\mathcal{M}_0' = \mathrm{LLM}_{\mathrm{mesh\_select}}(\mathcal{M}_0, Q, C), \tag{2}$$

where $\mathcal{M}_0' \subseteq \mathcal{M}_0$ denotes the set of candidate terms with their rationales. The selected subset $\mathcal{M}_0'$ is then combined using Boolean operators to form the initial structured query:

$$q_0 \leftarrow \mathrm{LLM}_{\mathrm{query\_gen}}(\mathcal{M}_0', Q, C). \tag{3}$$

Core concepts are typically linked with AND to enforce precision, while broader synonyms or alternatives are connected with OR to maximize recall. Temporal filters (e.g., publication date restrictions) are then applied to constrain retrieval during evaluation. This formulation mirrors the workflow of human researchers, who begin with broad but structured queries and iteratively refine them based on preliminary search results.

Iterative Self-Critic Query Refinement. Once the initial query is generated, the self-critic mechanism guides iterative refinement. At each iteration t, PubMed Reasoner submits the current query q_{t-1} to the PubMed search engine, which returns a ranked list of retrieved records (e.g., title, abstract, and PubMed ID):

$$\mathcal{A}_t \leftarrow \mathrm{PubMedSearch}(q_{t-1}). \tag{4}$$

From the ranked results $\mathcal{A}_t$, we extract metadata fields that serve as inputs to the self-critic. We denote these self-critic signals as:

$$\mathcal{S}_t = \left\{(\mathrm{title}_i,\ \mathrm{abstract}_i)\right\}_{i=1}^{|\mathcal{A}_t|}. \tag{5}$$

Rather than analyzing full texts which would incur substantial computational and token costs, the self-critic operates solely on $\mathcal{S}_t$. This design enables efficient evaluation of candidate MeSH terms

while preserving sufficient semantic information for relevance assessment. Given the current candidate MeSH pool $\mathcal{M}'_{t-1}$, each term is then evaluated along three structured dimensions (Sec. 2), which collectively provide actionable feedback for iterative query refinement: Coverage measures whether the concept represented by a candidate MeSH$_i$ term appears in the self-critic signals $\mathcal{S}_t$; Alignment evaluates whether the articles associated with each candidate MeSH$_i$ term are pertinent to the user question; Redundancy identifies whether a candidate term overlaps with, or is superfluous given other terms in the current set, as well as the logical composition (AND/OR) implied by the query intent.

The evaluation of $\mathcal{M}'_{t-1}$ along these dimensions is done through the following operators:

$$\mathrm{Cvg}_t \leftarrow \mathrm{LLM}_{\mathrm{coverage}}\big(\mathcal{M}'_{t-1}, \mathcal{S}_t\big), \tag{6}$$

$$\mathrm{Align}_t \leftarrow \mathrm{LLM}_{\mathrm{alignment}}\big(\mathcal{M}'_{t-1}, \mathcal{S}_t, Q\big), \tag{7}$$

$$\mathrm{Redun}_t \leftarrow \mathrm{LLM}_{\mathrm{redundancy}}\big(\mathcal{M}'_{t-1}, \mathcal{S}_t, q_{t-1}\big). \tag{8}$$

Each of Cvg_t, Align_t, and Redun_t is a list of pairs $\{(y_{t,i},\ r_{t,i})\}_{i=1}^{|\mathcal{M}'_{t-1}|}$, where $y_{t,i} \in \{\mathrm{Yes}, \mathrm{No}\}$ is a binary outcome indicating the result of the corresponding operator for candidate term MeSH$_i$, and $r_{t,i}$ provides a textual rationale that informs subsequent refinement. In particular, for Redun_t, $r_{t,i}$ both explains the redundancy verdict and includes the recommended boolean linkage under the previous search query q_{t-1}.

Given the previous MeSH terms, current self-critic signals $\mathcal{S}_t$, and per–term feedback, we refine the candidate MeSH set:

$$\begin{aligned}\mathcal{M}_t \ \leftarrow\ \mathrm{LLM}_{\mathrm{update}}\Big(\ &\mathcal{M}'_{t-1},\ Q,\ C,\\ &\mathcal{S}_t,\ \mathrm{Cvg}_t,\ \mathrm{Align}_t,\ \mathrm{Redun}_t\Big),\end{aligned} \tag{9}$$

where $\mathcal{M}_t$ is the revised set (with rationales). The update favors terms that increase coverage without harming alignment, and prunes (or merges) terms flagged as redundant.

From the refined MeSH terms $\mathcal{M}_t$, PubMed Reasoner drafts a new query, then enforces syntactic validity with a rule-based normalizer:

$$\tilde{q}_t \ \leftarrow\ \mathrm{LLM}_{\mathrm{refine}}\big(\mathcal{M}_t,\ \mathcal{S}_t,\ \mathcal{H},\ Q,\ C\big), \tag{10}$$

$$q_t \ \leftarrow\ \mathrm{QueryNormalize}\big(\tilde{q}_t\big), \tag{11}$$

where $\mathcal{H}$ stores the query history $\{q_0, ..., q_{t-1}\}$. The refinement aims to balance recall and precision using the proxies above: encourage additions that improve coverage (recall) if alignment remains high, and remove or demote terms that are misaligned or redundant (precision). The normalizer applies rule-based checks for operator placement, parenthesization, and field qualifiers, repairing common errors to guarantee a valid PubMed query. The self-critic loop then repeats with q_t until a stopping rule is met (e.g., no gain in coverage/alignment or a budget limit). The final optimized query q^* is returned for downstream retrieval. A detailed algorithm is provided in Algo. 1, and the prompts for each stage are included in App. E.

3.2 Reflective Article Retrieval with Early Stopping

PubMed Reasoner operates on the final retrieved records $\mathcal{A}^\star = \{a_1, \ldots, a_M\}$ returned by the search step (Eq. 4) together with their corresponding metadata $\mathcal{S}^*$(Eq. 5). Since PubMed search prioritizes records by relevance, highly pertinent evidence is expected to appear early in the ranked list. Accordingly, PubMed Reasoner enforces an early-stopping rule: once sufficient supporting evidence has been accumulated, retrieval halts to avoid further token consumption.

Coarse Filtering. Each retrieved record is first screened using metadata $\mathcal{S}^*$ to assess coarse relevance to the query. Only plausibly relevant records are retained for subsequent processing:

$$\big\{\big(v_i, r_i^{\mathrm{filter}}\big)\big\}_{i=1}^{|\mathcal{A}^\star|} \leftarrow \mathrm{LLM}_{\mathrm{filter}}\big(\mathcal{S}^\star,\ Q\big), \tag{12}$$

where $v_i \in \{\mathrm{Yes}, \mathrm{No}\}$ indicates keep/drop and r_i^{filter} provides a short rationale. Since $\mathcal{A}^\star$ is ranked by PubMed, the retained set $\mathcal{A}^+ = \{\, a_i \in \mathcal{A}^\star :\ v_i = \mathrm{Yes} \,\}$ preserves the original order and enables early prioritization of high-quality evidence. To control retrieval cost, coarse filtering is applied only to the top $M_{\max}$ ranked articles, the maximum budget of articles allowed for downstream processing. Additional details regarding $M_{\max}$ and token budget T are provided in App. C.

Reflective Evidence Extraction. For each $a_i \in \mathcal{A}^+$, we extract candidate evidence and evaluate **alignment** (i.e., does the candidate evidence directly address Q?):

$$\big\{(E_i,\ \mathrm{align}_i,\ r_i^{\mathrm{align}})\big\}_{i=1}^{|\mathcal{A}^+|} = \mathrm{LLM}_{\mathrm{extract}}\big(\mathcal{A}^+, Q\big), \tag{13}$$

where E_i is the extracted passage, $\mathrm{align}_i \in \{\mathrm{Yes}, \mathrm{No}\}$ indicates whether the passage directly addresses Q, and r_i^{align} provides a rationale. The evolving evidence pool is then

$$\mathcal{E} \ =\ \{\, E_i : \mathrm{align}_i = \mathrm{Yes} \,\}. \tag{14}$$

Batching and Early Stopping. To reduce token costs, we process the ranked results in batches. We partition $\mathcal{A}^+$ into batches $\{\mathcal{B}_1, \ldots, \mathcal{B}_K\}$ of size m (e.g., m=5). After processing batch b, we update the evidence pool: $\mathcal{E}^{(b)} = \mathcal{E}^{(b-1)} \cup \{E_i : a_i \in \mathcal{B}_b,\ \text{align}_i = \text{Yes}\}$, $\mathcal{E}^{(0)} = \varnothing$. A reflective module then judges whether the current evidence $\mathcal{E}^{(b)}$ provides sufficient **coverage** of the query (i.e., whether the major aspects of Q are addressed).

$$\texttt{is\_sufficient} \leftarrow \text{LLM}_{\text{CoverageCheck}}(\mathcal{E}^{(b)}, Q) \tag{15}$$

Retrieval stops early when $\texttt{is\_sufficient} = \text{Yes}$, the marginal utility of additional articles becomes negligible, or the token budget is reached. A detailed algorithm is provided in Algo. 2, and the prompts for each stage are included in App. E.

3.3 Evidence-Grounded Response Generation

Given the final curated evidence pool $\mathcal{E}$, PubMed Reasoner composes a final response by integrating the most relevant findings into a coherent, citation-supported explanation.

Summary–of–Evidence (SoE). To convert the vetted evidence $\mathcal{E}$ into a compact and citable representation, PubMed Reasoner groups passages by article and distills key observations that directly address question Q. This produces a structured summary $\text{SoE} \leftarrow \text{LLM}_{\text{summary}}(\mathcal{E}, Q)$, ensuring that every factual claim remains linked to its supporting source. Each retained citation is preserved explicitly, promoting transparency and reproducibility.

Response Generation. Finally, the system generates the user-facing answer by conditioning on the question Q, task requirements T, optional context C, and the SoE. Formally, the final response is produced as $R \leftarrow \text{LLM}_{\text{response}}(Q, T, C, \text{SoE})$, ensuring that the explanation is grounded, verifiable, and aligned with the task specification.

This staged design turns the LLM into a reasoning orchestrator: every statement in the final answer is linked to specific citations, improving interpretability and clinical trustworthiness. Implementation details are provided in App. C.

4 Experiments

We evaluate PubMed Reasoner on two biomedical QA datasets: PubMedQA (Jin et al., 2019) and MMLU Clinical Knowledge (MMLU-CK) (Wang et al., 2024). For each dataset, we report both prediction accuracy and explanation quality, and conduct ablation studies to quantify the contribution of each component of PubMed Reasoner to overall performance. PubMedQA additionally provides supporting context for each question; in our setting, we restrict retrieval to articles published within the official dataset's specified year range to ensure consistency.

4.1 Evaluation Metrics

We assess model performance along several dimensions: prediction accuracy, explanation quality, cost analysis and evidence sufficiency depth.

- **Accuracy:** Proportion of questions for which the model produces the correct answer label.
- **Explanation Quality:** Using an LLM-as-a-judge (see prompt in App. H), we perform pairwise comparisons and assign five-point Likert scores on four axes: reasoning soundness, evidence grounding, clinical relevance and trustworthiness.
- **Cost Analysis:** To quantify the computational efficiency of each method, we measure four complementary cost indicators: (1) input token usage, (2) output token usage, (3) number of LLM API calls, and (4) number of PubMed search calls issued during retrieval.
- **Query Term Quality:** To evaluate the effectiveness of search query formulation, we measure precision and recall of the MeSH terms generated after self-critic refinement, both crucial for downstream retrieval and reasoning performance.
- **Evidence Sufficiency Depth (ESD):** We additionally track how many articles are processed before early stopping is triggered. This metric reflects retrieval efficiency. Detailed results are provided in App. I.

Detailed definitions and justification for each metric are provided in App. B.

4.2 Baselines

We compare PubMed Reasoner with the following representative baselines:

- **LLM Baseline:** A strong LLM that answers questions without explicit retrieval or search planning.
- **Human Performance:** Reported accuracy on PubMedQA, which serves as an approximate upper bound for model performance.
- **RAG Method:** LLM generates a single search query, retrieves relevant articles from PubMed search engine, and incorporates them into response generation.
- **Self-reflection Agent:** A reasoning-based baseline that combines retrieval with self-reflection to gradually refine the final answer.

Detailed configuration of each baseline can be found in App. D.

Table 1: Accuracy on PubMedQA and MMLU-CK test sets for PubMed Reasoner variants, strong LLMs, RAG method, self-reflection agents, and human performance. Best result per column is **bold-faced**; second best is underlined. Additional results for PubMed Reasoner with Qwen variants are provided in App. F.

	Method	PubMedQA	MMLU-CK
Without retrieval	Gemini-2.5 Pro	75.64%	60.52%
	GPT4o (leaderboard)	75.20%	60.64%
	Human performance	78.00%	–
RAG method	with Gemini	72.30%	58.94%
	with GPT	73.28%	58.26%
Self-reflection	with Gemini	77.08%	60.67%
	with GPT	77.12%	60.79%
PubMed Reasoner	with Gemini	77.26%	61.36%
	with GPT	**78.32%**	**63.21%**

4.3 PubMedQA

Accuracy. Table 1 shows that PubMed Reasoner achieves near-human or superior accuracy on PubMedQA. In particular, PubMed Reasoner (GPT) attains 78.32% accuracy, slightly exceeding the reported human expert performance. Compared to other baselines, PubMed Reasoner consistently outperforms both direct LLM inference and RAG-based methods, improving upon the GPT baseline by 3.12% and yielding clear gains over RAG. Notably, standard RAG underperforms direct LLM inference, likely due to noisy retrieval caused by unrefined, one-shot query generation that introduces weakly aligned or irrelevant evidence into the model context. Relative to the self-reflection agent, PubMed Reasoner achieves modest but consistent accuracy improvements. Importantly, these gains are obtained with substantially lower computational cost (shown in Table 3). Together, these results demonstrate that PubMed Reasoner offers a more favorable accuracy–efficiency trade-off than existing baselines, delivering practical accuracy improvements while significantly reducing computational overhead. Qualitative error analysis and representative failure cases are provided in App. J.

Reasoning Quality. On PubMedQA, PubMed Reasoner with the Gemini backbone consistently outperforms Gemini-2.5 Pro across all four LLM-judge dimensions, as reported in Table 2. The pairwise win rate rises by 14.9% in Reasoning Soundness, by 14.6% in Evidence Grounding, by 14.2% in Clinical Relevance, and by 17.2% in Trustworthiness, while tie rates remain low. The average Likert score also increases respectively by 0.168, 0.180, 0.146, and 0.163, yielding explanations that are more coherent, better grounded in evidence, more clinically focused, and more trustworthy.

Cost Analysis. As shown in Table 3, PubMed Reasoner achieves substantial reductions in computational overhead compared to the self-reflection baseline. Across key cost metrics, PubMed Reasoner reduces input token usage by 55.34%, lowers the number of LLM API calls by 41.82%, and decreases PubMed search calls by 41.36%. These efficiency gains stem from structured query planning and early stopping, which avoid unnecessary retrieval and redundant reasoning steps. Although the overall cost of PubMed Reasoner remains higher than that of the one-shot LLM baseline, it produces citation-grounded and verifiable responses, offering substantially stronger reliability than direct generation. Compared to the RAG baseline, which often yields shallow or weakly aligned search results, PubMed Reasoner performs more targeted retrieval and accumulates higher-quality evidence.

Query Term Quality. As shown in Table 4, PubMed Reasoner maintains high precision while substantially improving recall due to self-critic refinement during query planning. With a GPT backbone, recall improves by 20.99% over the GPT baseline and by 7.6% over the self-reflection agent. With a Gemini backbone, recall improves by 17.18% over the baseline and also surpasses the self-reflection agent. These results demonstrate that self-critic refinement broadens conceptual coverage without sacrificing alignment, producing higher-quality search queries.

4.4 MMLU Clinical Knowledge

Accuracy. On MMLU-CK dataset (Table 1), self-reflection offers only a marginal lift over the raw LLM baseline. In contrast, PubMed Reasoner delivers consistent gains over the respective backbones: 1.11% over Gemini-2.5 Pro and 2.69% over GPT-4o. This underscores that our self-critic improves accuracy beyond what self-reflection can achieve. Moreover, as shown in Table 3, these accuracy gains are achieved with substantially lower computational overhead.

Explanation Quality. On MMLU-CK, PubMed Reasoner consistently surpasses Gemini-2.5 Pro across all LLM-judge dimensions (Table 2). Pairwise win rates rise by 1.2% in reasoning soundness, 38.9% in evidence grounding, 11.2% in clinical relevance, and 13.5% in trustworthiness, with low tie rates. Average Likert scores also increase by 6–12% across dimensions, indicating clearer logic,

Table 2: Explanation quality evaluation on PubMedQA (left) and MMLU-CK (right) test sets: Gemini w/o retrieval vs. PubMed Reasoner. Win/tie/loss rates from pairwise comparisons judged by GPT-4; average Likert scores (1–5).

Metric	PubMedQA					MMLU-CK				
	Loss / Tie / Win (%)			Avg. Likert (1–5)		Loss / Tie / Win (%)			Avg. Likert (1–5)	
	Gemini	Tie	Ours	Gemini	Ours	Gemini	Tie	Ours	Gemini	Ours
Reasoning Soundness	39.7	5.7	**54.6**	3.416	**3.584**	44.0	10.8	**45.2**	3.307	**3.699**
Evidence Grounding	40.8	3.8	**55.4**	3.421	**3.601**	25.2	11.7	**64.1**	3.209	**3.595**
Clinical Relevance	39.2	7.4	**53.4**	3.438	**3.584**	34.4	20.0	**45.6**	3.525	**3.732**
Trustworthiness	38.7	5.4	**55.9**	3.424	**3.587**	35.3	15.9	**48.8**	3.386	**3.712**

Table 3: Cost comparison on PubMedQA and MMLU-CK using a shared GPT-4 backbone for PubMed Reasoner and baselines. For PubMedQA, input and output token counts are reported as ratios relative to the direct LLM baseline. For each metric, the lower value between the self-reflection agent and PubMed Reasoner is **bolded**.

	PubMedQA				MMLU-CK			
Method	**Input Tokens**	**Output Tokens**	**LLM Calls**	**Search Calls**	**Input Tokens**	**Output Tokens**	**LLM Calls**	**Search Calls**
LLM	542.70	144.90	1	0	90.24	98.64	1	0
RAG method	×1.64	×2.40	2	1	×4.00	×3.06	2	1
Self-reflection	×225.27	×13.60	13.08	3.52	×1914.53	×**24.35**	14.61	4.77
PubMed Reasoner	×**97.25**	×**12.67**	**7.61**	**2.49**	×**1098.26**	×24.86	**10.72**	**3.59**

Table 4: Search quality on PubMedQA. Precision/recall computed by comparing set of predicted MeSH terms from the final query against gold MeSH annotations.

Method	Precision	Recall
Gemini-2.5 Pro	0.9422	0.6848
+Self-reflection	0.9745	0.7900
+PubMed Reasoner	**0.9826**	**0.8025**
GPT-4o	**0.9874**	0.7052
+Self-reflection	0.9562	0.7928
+PubMed Reasoner	0.9868	**0.8532**

Table 5: Ablation on PubMedQA with PubMed Reasoner (Gemini): impact of self-critic module. *Evidence-Grounded Response Rate (EGR)*: fraction of questions whose final response cites ≥ 1 extracted findings that pass coverage and alignment checks in reflective stage.

Method / Config.	Accuracy	EGR
PubMed Reasoner	77.26%	82.64%
w/o self-critic	75.60%	64.46%

stronger evidence support, sharper clinical focus, and greater trustworthiness. In sum, PubMed Reasoner produces better clinical explanations.

Cost Analysis. As shown in Table 3, the MMLU-CK results exhibit a pattern similar to PubMedQA: PubMed Reasoner consistently incurs substantially lower computational cost than the self-reflection baseline. In particular, PubMed Reasoner requires fewer input tokens, fewer LLM API calls, and fewer PubMed search calls, reflecting a more efficient retrieval and reasoning workflow on this broader clinical knowledge benchmark as well.

Query Term Quality. MMLU does not provide MeSH term annotations, hence cannot be assessed.

4.5 Ablation Study

Ablation Setup. We ablate two components: **(1) self-critic refinement**, **(2) reflective evidence extraction** and **(3) batching and early stopping** using the PubMed Reasoner (Gemini) variant from App. C, unless otherwise noted.

Self-Critic Refinement. Table 5 reports accuracy and the Evidence-Grounded Response Rate (EGR)—the fraction of questions whose final response cites at least one extracted finding that passes coverage and alignment checks. Removing the self-critic markedly lowers EGR (drop of 28.20%) and also reduces accuracy 2.2%. This confirms that the self-critic is not only improving final correctness, but also producing robust, well-formed queries that retrieve on-point evidence.

Reflective Evidence Extraction. We disable the alignment filter and instead summarize all retrieved articles before producing a final answer. As shown in Table 6, accuracy remains unchanged, but the quality of explanations shifts: reflective integration improves reasoning soundness and clinical relevance, while introducing more diverse evidence slightly lowers grounding precision. These trends match the more fine-grained analysis in App. K.

Batching and Early Stopping. The ablation

Table 6: Ablation on PubMedQA with PubMed Reasoner (Gemini): impact of reflective retrieval (RR).

Metric	*w/o* RR	Ours
Accuracy (%)	77.24	**77.26**
Reasoning Soundness	3.422	**3.554**
Evidence Grounding	**3.554**	3.427
Clinical Relevance	3.443	**3.573**
Trustworthiness	**3.432**	**3.432**

study results in App. L highlight the importance of providing evidence at the appropriate granularity. High-level condensations, such as abstracts or naive summaries fail to capture essential reasoning cues, whereas our selective early-stopping framework preserves critical information and yields markedly improved performance.

5 Related Work

LLMs & Retrieval-Augmented Methods. Despite the remarkable progress achieved by LLMs in natural language understanding, reasoning, and generation, when applied to high-stakes biomedical tasks, they often hallucinate facts or rely on outdated parametric memory, raising concerns about reliability and safety (Guan et al., 2023; Xu et al., 2024). To address these issues, RAG has emerged as a promising paradigm. Existing RAG approaches either incorporate structured resources such as knowledge graphs (Abu-Rasheed et al., 2024) or augment prompts with retrieved content from domain-specific corpora (Arslan et al., 2024). While these strategies improve factual grounding, they face persistent challenges: retrieval systems often struggle with the *coverage–relevance trade-off*, returning either too little evidence or overwhelming the model with irrelevant content (Liu et al., 2024). Moreover, once an initial query is issued, most RAG pipelines lack mechanisms for iterative refinement, making them brittle in complex biomedical scenarios (Dai et al., 2024).

Search Query Optimization. A complementary direction focuses on enhancing the query quality. Early work in information retrieval explored query expansion and relevance feedback (Sparck Jones, 1974; Crane and Bernier, 1951), but such methods often relied on manual heuristics (Arasu et al., 2001) and lacked semantic explanation capabilities (Boytsov, 2011). More recently, query optimization has also been framed as a reinforcement learning problem, where the model learns to improve retrieval performance through policy gradients or preference-based objectives such as PPO (Schulman et al., 2017), DPO (Rafailov et al., 2023), or GRPO (Shao et al., 2024). While effective, these methods typically require a separate reward model or explicit training signals, making them computationally expensive and less flexible in specialized domains like biomedicine. In contrast, our work introduces a training-free self-critic mechanism that performs query refinement without external supervision or gradient updates. Unlike prior RL-based or heuristic based method, the self-critic provides fine-grained feedback on MeSH query terms, iteratively improves retrieval quality, mitigates error propagation and enhances factual grounding.

Self-Reflection and Reasoning Agents. Another line of research seeks to improve LLM reliability through self-reflection and agent-based reasoning. Self-reflection methods allow models to reexamine their own outputs and refine answers, while reward modeling (Leike et al., 2018; Choudhury, 2025) and verbal reinforcement learning (Shinn et al., 2023) aim to align reasoning with human-like preferences. Self-consistency sampling further increases robustness by aggregating multiple reasoning trajectories (Wang et al., 2022). However, these methods generally intervene at the *answer-generation stage*, which makes them computationally expensive and unable to prevent low-quality retrieval from propagating downstream. As a result, they provide limited control over the evidence collection process itself.

6 Conclusion

Altogether, these results demonstrate that combining structured self-critique with evidence-based integration moves biomedical QA closer to expert-level explanation, while remaining efficient and reproducible. More broadly, our findings suggest design principles for multi-stage LLM agents in high-stakes domains: shifting reflection earlier in the pipeline can prevent compounding errors; explicit grounding in external evidence improves transparency and reliability; and adaptive mechanisms such as early stopping enable practical deployment without sacrificing rigor. Notably, self-critic is especially effective in multi-step reasoning settings, where revising only problematic steps rather than regenerating the entire chain ensures both efficiency and logical consistency. These mechanisms can be generalized beyond biomedicine, providing a blueprint for building trustworthy, domain-specialized LLM systems in areas such as law, finance, and scientific discovery.

7 Limitations

Although the self-critic mechanism substantially improves query quality, it remains heuristic and may inherit biases from the underlying LLM backbone, particularly in subdomains with limited training data. Additionally, the reflective retrieval stage may overlook lower-ranked yet relevant articles, potentially reducing evidence coverage in long-tail biomedical topics. Moreover, the current pipeline is largely linear at the stage-level: once reasoning progresses to later stages, earlier retrieval or query decisions cannot be revisited. This lack of backward adaptivity limits the system's ability to iteratively correct upstream errors or expand underrepresented evidence when inconsistencies arise in the final response. Revisiting previous stages could improve robustness and coverage but would introduce additional computational and token costs, highlighting the trade-off between accuracy and efficiency inherent to our design.

References

Hasan Abu-Rasheed, Christian Weber, and Madjid Fathi. 2024. Knowledge graphs as context sources for llm-based explanations of learning recommendations. In *2024 IEEE Global Engineering Education Conference (EDUCON)*, pages 1–5. IEEE.

Arvind Arasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paepcke, and Sriram Raghavan. 2001. Searching the web. *ACM Transactions on Internet Technology (TOIT)*, 1(1):2–43.

Rahul K Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, and 1 others. 2025. Healthbench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*.

Muhammad Arslan, Hussam Ghanem, Saba Munawar, and Christophe Cruz. 2024. A survey on rag with llms. *Procedia computer science*, 246:3781–3790.

Leonid Boytsov. 2011. Indexing methods for approximate dictionary searching: Comparative analysis. *Journal of Experimental Algorithmics (JEA)*, 16:1–1.

Sanjiban Choudhury. 2025. Process reward models for llm agents: Practical framework and directions. *arXiv preprint arXiv:2502.10325*.

EJ Crane and Charles L Bernier. 1951. Indexing and index-searching. *Punched Cards: Their Applications to Science and Industry*, page 331.

Sunhao Dai, Chen Xu, Shicheng Xu, Liang Pang, Zhenhua Dong, and Jun Xu. 2024. Bias and unfairness in information retrieval systems: New challenges in the llm era. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6437–6447.

Wanling Gao, Yuqing Zhu, Zhen Jia, Chunjie Luo, Lei Wang, Zhiguo Li, Jianfeng Zhan, Yong Qi, Yongqiang He, Shiming Gong, and 1 others. 2013. Bigdatabench: a big data benchmark suite from web search engines. *arXiv preprint arXiv:1307.0320*.

Jian Guan, Jesse Dodge, David Wadden, Minlie Huang, and Hao Peng. 2023. Language models hallucinate, but may excel at fact verification. *arXiv preprint arXiv:2310.14564*.

Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanmin Wu, Jiayi Lei, Pengshuo Qiu, Pan Lu, Zehui Chen, Chaoyou Fu, Guanglu Song, and 1 others. 2024. Mmsearch: Benchmarking the potential of large models as multi-modal search engines. *arXiv preprint arXiv:2409.12959*.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*.

Adam Tauman Kalai, Ofir Nachum, Santosh S Vempala, and Edwin Zhang. 2025. Why language models hallucinate. *arXiv preprint arXiv:2509.04664*.

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. 2018. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*.

Zheng Liu, Yujia Zhou, Yutao Zhu, Jianxun Lian, Chaozhuo Li, Zhicheng Dou, Defu Lian, and Jian-Yun Nie. 2024. Information retrieval meets large language models. In *Companion Proceedings of the ACM Web Conference 2024*, pages 1586–1589.

Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. 2023a. Capabilities of gpt-4 on medical challenge problems. *arXiv preprint arXiv:2303.13375*.

Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolo Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, and 1 others. 2023b. Can generalist foundation models outcompete special-purpose tuning? case study in medicine. *arXiv preprint arXiv:2311.16452*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, and 1 others. 2025. Toward expert-level medical question answering with large language models. *Nature Medicine*, 31(3):943–950.

Karen Sparck Jones. 1974. Automatic indexing. *Journal of documentation*, 30(4):393–432.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.

A Algorithms

Algorithms 1 and 2 respectively describe the steps involved in **Self-Critic Query Refinement with Query History** and **Reflective Article Retrieval with Early Stopping**.

Algorithm 1 Self-Critic Query Refinement with Query History

Input: Initial query q_0 from Eq. 3
Output: Refined query q^*
Initialize history $\mathcal{H} \leftarrow [q_0]$
for iteration $t = 1, \ldots, T$ **do**
  Retrieve article metadata $\mathcal{S}_t$ (Eqs. 4–5)
  Self-critic evaluation of MeSH terms:
    Evaluate **coverage** (Eq. 6), **alignment** (Eq. 7), and **redundancy** (Eq. 8) from S_t
  Update MeSH terms (Eq. 9)
  Refine query q_t (Eq. 10)
  Rule-based normalization (Eq. 11)
  Append q_t to history: $\mathcal{H} \leftarrow \mathcal{H} \cup \{q_t\}$
  if retrieval quality converges or $t = T$ **then**
    $q^* \leftarrow q_t$; **break**
  end if
end for
return q^*

B Evaluation Metrics

This appendix provides detailed definitions and justifications for the evaluation metrics used in our experiments, which are designed to assess not only predictive correctness but also reasoning quality, evidence usage, and computational efficiency.

Accuracy. Prediction accuracy is defined as the proportion of questions for which the model produces the correct answer label. This metric captures the model's ability to arrive at the correct final decision and serves as a primary indicator of task performance. Accuracy is reported over the full evaluation set and is used to ensure comparability with prior biomedical question answering benchmarks.

Explanation Quality. Beyond answer correctness, we evaluate the quality of model-generated explanations using an LLM-as-a-judge framework. We adopt the HealthBench evaluation protocol (Arora et al., 2025), adapting the prompts to our dataset (the full prompt is provided in App. H). For each question, model outputs are assessed along four complementary axes:

- **Reasoning Soundness:** Whether the explanation is logically coherent, internally consistent, and free of contradictions.
- **Evidence Grounding:** Whether factual claims are supported by retrieved biomedical evidence, with minimal hallucination or unsupported assertions.
- **Clinical Relevance:** Whether the explanation directly addresses the biomedical or clinical aspects of the question in an appropriate and meaningful manner.
- **Trustworthiness:** Whether the response aligns with established biomedical knowledge and avoids misleading or unsafe conclusions.

Each axis is scored on a five-point Likert scale, with higher scores indicating better explanation quality. To isolate explanation quality from answer correctness, we evaluate only instances where both models predict the correct label. Pairwise comparisons are performed with randomized response

Algorithm 2 Reflective Article Retrieval with Early Stopping

Input: Ranked result set $\mathcal{A}^\star$ and metadata $\mathcal{S}^\star$, question Q, maximum ranked-article budget $M_{\max}$, batch size m, token budget T
Output: Final evidence pool $\mathcal{E}^{(\text{final})}$
Initialize filtered ranked set $\mathcal{A}^+ \leftarrow \varnothing$
Initialize evidence pool $\mathcal{E}^{(0)} \leftarrow \varnothing$
Initialize token usage counter $b_{\text{used}} \leftarrow 0$
// Coarse filtering up to ranked-article budget
for $i = 1$ **to** $\min\left(|\mathcal{A}^\star|,\, M_{\max}\right)$ **do**
    Coarse filtering each article (Eq. 12)
    if $v_i =$ Yes **then**
        $\mathcal{A}^+ \leftarrow \mathcal{A}^+ \cup \{a_i\}$ {Preserves original ranking order}
    end if
end for
Partition $\mathcal{A}^+$ into batches $\{\mathcal{B}_1, \ldots, \mathcal{B}_K\}$ of size at most m
for batch index $b = 1, \ldots, K$ **do**
    // Reflective evidence extraction for $\mathcal{B}_b$
    Extract candidate evidence (Eq. 13)
    Update evidence pool (Eq. 14)
    // Update token usage
    $t_{\text{used}} \leftarrow t_{\text{used}} + \text{TokensUsed}(\mathcal{B}_b)$
    Check stopping criteria (Eq. 15)
    if is_sufficient **then**
        break *// Coverage sufficient*
    end if
    if $t_{\text{used}} \geq T$ **then**
        break *// Token budget reached*
    end if
end for
$\mathcal{E}^{(\text{final})} \leftarrow \mathcal{E}^{(b)}$
return $\mathcal{E}^{(\text{final})}$

order, and ties are permitted to reduce positional bias. An independent LLM (GPT) judge is used to avoid self-evaluation effects.

Cost Analysis. To quantify computational efficiency, we measure four complementary cost indicators: (1) **Input token usage**, (2) **Output token usage**, (3) **Number of LLM API calls**, and (4) **Number of PubMed search calls** issued during retrieval. These metrics capture both the direct inference cost associated with language model usage and the operational overhead introduced by iterative retrieval and refinement. Reporting these indicators allows us to assess whether performance gains are achieved in a resource-efficient manner, which is particularly important for large-scale or deployment-oriented biomedical applications.

Query Term Quality: To evaluate the effectiveness of search query formulation, we compare the MeSH terms generated by each model against the ground-truth MeSH annotations associated with each question. Precision measures the alignment of proposed terms with the question intent, while recall reflects coverage of all key biomedical concepts. This metric assesses how effectively the model identifies relevant search concepts prior to retrieval, which is crucial for downstream reasoning performance.

Evidence Sufficiency Depth (ESD). We introduce *Evidence Sufficiency Depth (ESD)* to characterize retrieval efficiency under early stopping. ESD is defined as the number of retrieved articles processed before the early-stopping criterion is satisfied. Since PubMed returns results in relevance-ranked order, lower ESD values indicate that sufficient supporting evidence is identified earlier in the ranked list. This metric directly reflects the model's ability to rapidly locate adequate evidence while minimizing unnecessary paper retrieval and token consumption.

C PubMed Reasoner Configurations

Backbone Model. We instantiate PubMed Reasoner using three model families. PubMed Reasoner (GPT) employs GPT-4o with temperature t=0, which promotes faithful extraction and reduces the likelihood of introducing unsupported content that could compromise evidence fidelity. PubMed Reasoner (Gemini) is a hybrid of Gemini-2.5 Pro (t=0.8) and Gemini-2.5 Flash (t=0): Pro is used primarily for the self-critic refinement and reflective extraction steps, while Flash is used for efficiency-critical calls. PubMed Reasoner (Qwen) uses three Qwen variants, namely Qwen2.5-1.5B, Qwen2.5-7B and Qwen2.5-72B, all with t=0, encouraging faithful extraction and minimizing added content that could compromise evidence fidelity.

Reflective Article Retrieval with Early Stopping. In the reflective stage (Sec. 3.2), we process retrieved articles in batches of size m=5 to balance context length with reflective efficiency. To further control token cost, we impose a maximum budget of the top $M_{\max} = 20$ ranked articles (i.e., at most four batches). This ensures that retrieval remains both efficient and grounded in the most relevant evidence returned by PubMed. We set the token budget T to infinity during benchmark evaluation.

D Baseline Configuration

LLM Baselines. We consider Gemini-2.5 Pro and GPT-4o as strong LLM baselines. In particular, we report the official leaderboard submission results as the GPT baseline on the PubMedQA benchmark.

Human Performance. For PubMedQA, we report the human performance published on the official leaderboard, which serves as a reference upper bound. Human expert results are not available for MMLU-CK, and thus are not reported for that benchmark.

RAG Method. Since no pre-constructed retrieval corpus or standardized RAG setup exists for the PubMedQA dataset, we implement a retrieval-augmented baseline by directly querying the PubMed search engine at inference time. Retrieved articles are incorporated as external knowledge to support response generation. To ensure a fair comparison, the RAG baseline uses the same query generation, article retrieval, summarization, and question-answering prompts as PubMed Reasoner, which are provided in App. E.

Self-Reflection Agent. The self-reflection baseline is allowed to issue PubMed search queries to retrieve supporting evidence. It generates multiple candidate responses and iteratively refines the final answer through an explicit reflection step. To ensure comparability, this baseline uses the same query generation, retrieval, summarization, and question-answering prompts as PubMed Reasoner, with the addition of a dedicated prompt for the self-reflection stage.

Self-reflection Prompt

System Prompt: You are a self-reflection agent for evidence-grounded biomedical question answering.
Your task is to identify conceptual gaps between the current answer and the verified sources, and to generate **one revised PubMed query** that targets missing or weakly supported concepts. When generating the revised query, you must avoid ineffective or repetitive search patterns by consulting the search history.
A concept gap exists if one or more of the following conditions hold:

- A key claim in the answer lacks direct support from the retrieved context.
- The context only partially addresses the question.
- The context is overly general and fails to capture critical biomedical specificity.

Logical operators:

- Use `AND` to combine *independent, parallel constraints*. Every term connected by `AND` must be satisfied.
- Use `OR` only for *similar or interchangeable concepts*. When using `OR`, you must enclose the entire OR-group in parentheses, e.g.: `(term1[mesh] OR term2[mesh])`.

Requirements:

1. Use only the Boolean operator `AND` to connect terms.
2. Field tag priority:
   (a) Place MeSH terms first and apply the `[mesh]` tag whenever possible.
   (b) Place date ranges next, formatted as either `YYYY:YYYY[pdat]` or `YYYY/MM/DD:YYYY/MM/DD[pdat]`.
   (c) Place all remaining terms last. Do not apply special field tags unless explicitly specified.
3. Spacing: Separate each term and each `AND` with exactly one space.
4. Date range format: `YYYY:YYYY[pdat]` or `YYYY/MM/DD:YYYY/MM/DD[pdat]`

Natural Language Question:
{natural_language_question}
Verified Sources:
{verified_sources}
Answer:
{rationale_answer}
Search History (if any):
{search_history}
Output:
Return the answer strictly as JSON following this schema:

```
{
  "query":
   "The final PubMed query string as
    a single string.",
  "rationale":
    "A brief explanation describing
    the identified concept gaps and
    how the revised query addresses
    these gaps while following the
```

```
    query construction rules."
}
```
Do not add any explanation or additional text outside the JSON.

We instantiate both the RAG method and the self-reflection agent using two backbone models, GPT-4o and Gemini-2.5 Pro/Flash, following the same parameter settings as described in App. C. To encourage diverse reasoning trajectories, we employ higher decoding temperatures during the reflection stage (GPT at $t = 1.0$ and Gemini at $t = 0.8$).

E Prompt Design

Query Generation Prompt

System Prompt:
You are an expert in PubMed search syntax. Your task is to convert the provided natural language description of the desired literature, together with any contextual information, into a single valid PubMed query string.

Requirements:

1. Use only the Boolean operator `AND` to connect terms.
2. Field tag priority:
   (a) Place MeSH terms first and apply the `[mesh]` tag whenever possible.
   (b) Place date ranges next, formatted as either `YYYY:YYYY[pdat]` or `YYYY/MM/DD:YYYY/MM/DD[pdat]`.
   (c) Place all remaining terms last. Do not apply special field tags unless explicitly specified.
3. Spacing: Separate each term and each `AND` with exactly one space.
4. Date range format: `YYYY:YYYY[pdat]` or `YYYY/MM/DD:YYYY/MM/DD[pdat]`

Natural Language Question:
{natural_language_question}
Additional Context (if any):
{context}
Output:
Return the answer strictly as JSON following this schema:

```
{
  "query":
   "The final PubMed query string as
    a single string.",
  "rationale":
    "A brief explanation of term
   selection, field tags, ordering,
    use of AND, and how the context
    was incorporated."
}
```
Do not add any explanation or additional text outside the JSON.

Self-critic Prompt

System Prompt: You are a PubMed search planning assistant.
Your task is to produce **one improved PubMed query** for the next search step by:

- Interpreting the natural-language question and any additional context.
- Using search history to avoid ineffective or repetitive patterns.
- Evolving the candidate term set using coverage, alignment, and redundancy feedback.
- Ensuring the final query strictly follows the requirements.

Feedback Signals:
If no context is provided, return -1 for the corresponding signals. In producing the improved query, you must incorporate evolving feedback from:

1. **Coverage** — 1 if the provided context sufficiently represents the concepts relevant to the question; 0 otherwise.
2. **Alignment** — 1 if the provided context is relevant and appropriately focused on the question; 0 otherwise.
3. **Redundancy** — 1 if there are **no** overlapping, unnecessary, or logically unintended terms; 0 otherwise.

Logical operators:

- Use `AND` to combine *independent, parallel constraints*. Every term connected by `AND` must be satisfied.
- Use `OR` only for *similar or interchangeable concepts*. When using `OR`, you must enclose the entire OR-group in parentheses, e.g.: `(term1[mesh] OR term2[mesh])`.

Requirements:

1. Use only the Boolean operator `AND` to connect terms.
2. Field tag priority:

(a) Place MeSH terms first and apply the `[mesh]` tag whenever possible.
(b) Place date ranges next, formatted as either `YYYY:YYYY[pdat]` or `YYYY/MM/DD:YYYY/MM/DD[pdat]`.
(c) Place all remaining terms last. Do not apply special field tags unless explicitly specified.

3. Spacing: Separate each term and each `AND` with exactly one space.
4. Date range format: `YYYY:YYYY[pdat]` or `YYYY/MM/DD:YYYY/MM/DD[pdat]`

Natural Language Question:
{natural_language_question}
Additional Context (if any):
{search_meta}
Search History (if any):
{search_history}
Output:
Return the answer strictly as JSON following this schema:

```
{
  "query":
   "The final PubMed query string as
    a single string.",
  "rationale":
    "A brief explanation of term
   selection, field tags, ordering,
    use of logical operators,
   and how the context and feedback
    were incorporated."
    "feedback": {
        "coverage": int,
         "coverage_suggestion":
         "Suggested improvements",
        "alignment": int,
         "alignment_suggestion":
         "Suggested improvements",
        "redundancy": int,
         "redundancy_suggestion":
         "Suggested improvements"
    }
}
```

Do not add any explanation or additional text outside the JSON.

Reflective Article Retrieval Prompt

System Prompt:
You are a reflection assistant.
Your task is to determine whether the provided search results with current context (if provided) contain enough relevant and specific information to answer the question.
Natural Language Question:
{natural_language_question}
Search Results:
{search_results_str}
Additional Context (if any):
{context}
Output:
Return your answer strictly as JSON following this schema:

```
{
  "is_sufficient": true | false,
  "rationale": "Concise explanation
  of why the information is
  sufficient or insufficient.",
  "needed_pmids":
    ["PMID1", "PMID2", ...]
    # PMIDs of additional relevant
    articles, if any
}
```

Do not include any explanation or text outside the JSON.

Summary Prompt

System Prompt:
You are a professional academic rewriting assistant.
Your task is to transform the provided raw sources into a single, semantically coherent, and well-structured paragraph.
Requirements:

1. Use only the information from the provided raw sources, without adding external content.
2. Preserve all original in-text citations exactly as they appear (e.g., [PMID: xxxx]).
3. Ensure the paragraph is logically connected, concise, and scientifically rigorous.

Raw Sources:
{raw_sources}
Output:

Return the answer strictly as JSON following this schema:

```
{
  "verified_sources":
   "The final rewritten paragraph
   as a single string.",
}
```

Do not add any explanation or additional text outside the JSON.

Question Answering Prompt

System Prompt:
You are an expert assistant.
When sources are provided, you should primarily base your answer on the information in the sources. If the sources do not contain enough information to fully answer the question, you may supplement your answer using your own knowledge.
Provide your answer and a clear rationale explaining how you arrived at it.
Natural language question:
{natural_language_question}
Task instruction:
{task_instruction}
Additional context (if any): context
Sources: sources
Output:
Return the answer strictly as JSON following this schema:

```
{
  "answer": "Your answer according to
   the task instruction.",
  "rationale": "A clear explanation of
   how you arrived at the answer.
}
```

Do not add any explanation or additional text outside the JSON.

F Qwen Results

Table 7 summarizes the performance of all evaluated methods on PubMedQA and MMLU-CK. We report accuracy (%) across three Qwen2.5 model scales (1.5B, 7B, and 72B). Results are organized by (1) base model performance without retrieval, (2) self-reflection agent, and (3) our PubMed Reasoner framework.

Table 7: Accuracy on PubMedQA and MMLU-CK test sets for PubMed Reasoner Qwen2.5 variants, strong LLMs, and self-reflection agents. Best result per column is **bold-faced**.

Model/Method	PubMedQA	MMLU-CK
Qwen2.5-1.5B		
Without retrieval	**68.08%**	**55.67%**
Self-reflection agent	68.06%	51.44%
PubMed Reasoner	67.24%	51.42%
Qwen2.5-7B		
Without retrieval	70.20%	**56.64%**
Self-reflection agent	**71.12%**	56.11%
PubMed Reasoner	71.08%	56.09%
Qwen2.5-72B		
Without retrieval	74.89%	58.36%
Self-reflection agent	76.22%	59.67%
PubMed Reasoner	**76.42%**	**60.95%**

The effectiveness of PubMed Reasoner grows with model scale: while retrieval provides limited benefit for smaller Qwen variants (1.5B and 7B), the 72B model achieves the strongest gains, outperforming both its no-retrieval and self-reflection baselines on PubMedQA and MMLU-CK.

G Case study

We present several case studies comparing PubMed Reasoner with GPT-4o on PubMedQA-style questions. Each case includes the user question, the model's response, and the rationale provided. For PubMed Reasoner, the rationale is grounded in biomedical literature retrieved from PubMed, with inline citations (PMID identifiers) referencing specific studies.

These examples illustrate how PubMed Reasoner improves over direct LLM reasoning in several ways: (i) grounding answers in authoritative biomedical sources, thereby reducing hallucinations; (ii) providing more detailed rationales that capture not only correlations but also evidence from longitudinal studies, randomized controlled trials, or meta-analyses; and (iii) maintaining interpretability through explicit evidence chains.

Question: Does early introduction of sugar-sweetened beverages increase the risk of obesity in children?

GPT-4o: Yes
Rationale: Evidence consistently shows that early introduction of sugar-sweetened beverages is linked with increased caloric intake, poor dietary habits, and higher risk

of obesity in children.

PubMed Reasoner (GPT-4o): Yes
Rationale: Evidence from longitudinal and randomized controlled trials indicates that early introduction of juice and sugar-sweetened beverages is associated with higher intake later in childhood, a well-established obesity risk factor.
A cohort study [PMID:37150479] found that introducing juice before 6 months led to significantly higher juice consumption at 24 months, while an RCT [PMID:36146909] showed that parental education reduced early SSB introduction and may lower obesity risk.

Question: Is breastfeeding associated with a lower risk of childhood obesity?

GPT-4o: Yes
Rationale: Multiple studies have demonstrated that breastfeeding, especially exclusive breastfeeding, is associated with a modest but significant reduction in childhood obesity risk compared to formula feeding.

PubMed Reasoner (GPT-4o): Yes
Rationale: The evidence is mixed but overall supports a protective effect. One study found no association between feeding in the first 4 months and obesity risk after adjusting for confounders [PMID:40550602]. However, metabolomics analyses link formula feeding with obesity-related metabolic profiles [PMID:40452153].
A large cohort study found longer breastfeeding duration reduced risk, with effects differing by sex ($\geq$ 8 months protective in girls, 3–7 months protective in boys) [PMID:40440349].

Question: Can family-based lifestyle interventions reduce obesity prevalence in children?

GPT-4o: Yes
Rationale: Family-based lifestyle interventions targeting diet, physical activity, and behavior change are effective in reducing BMI and obesity prevalence in children.

PubMed Reasoner (GPT-4o): Yes
Rationale: Studies show that parental obesity, education, and health behaviors strongly influence child obesity risk. Family lifestyle patterns persist over time [PMID:29807535]. Interventions targeting the whole family, including parental health behaviors and socioeconomic supports, are associated with healthier child weight outcomes [PMID:40312675, PMID:36904286].

Question: Is there a link between fast food consumption and childhood obesity?

GPT-4o: Yes
Rationale: High frequency of fast food consumption is associated with higher caloric intake, increased fat/sugar consumption, and greater obesity risk.

PubMed Reasoner (GPT-4o): Yes
Rationale: Evidence shows fast food and ultra-processed food (UPF) consumption contributes to childhood obesity. UPFs accounted for 40% of caloric intake in one study, with fast foods alone contributing 7.8% and being correlated with waist circumference and overweight/obesity prevalence [PMID:40102562].
Another large study found maternal junk food consumption increased children's obesity odds by 39% [PMID:40481596].

Question: Do school-based nutrition programs improve weight outcomes in children?

GPT-4o: Yes
Rationale: School-based nutrition programs promote healthier eating and physical activity, leading to modest improvements in BMI and obesity outcomes.

PubMed Reasoner (GPT-4o): Yes
Rationale: A 3-year prospective cohort study in southern Taiwan found that lunchtime education and class-based nutrition programs increased normal-weight prevalence and reduced overweight/obesity. The BMI trajectory flattened compared to regional trends, approaching national stan-

dards [PMID:32381350].

Question: Are children with sleep deprivation at higher risk for obesity?

GPT-4o: Yes
Rationale: Sleep deprivation alters appetite-regulating hormones, increases snacking, and is consistently linked with higher obesity risk in children.

PubMed Reasoner (GPT-4o): Yes
Rationale: Epidemiological studies link insufficient sleep to adverse metabolic, endocrine, and energy balance changes leading to obesity [PMID:38012858, PMID:38004130].
A large cohort found insufficient sleep increased early maturation risk in girls, with obesity as a mediator [PMID:38065690].

Question: Is there an association between sedentary behavior and pediatric obesity?

GPT-4o: Yes
Rationale: Prolonged sitting and excessive screen time are strongly associated with higher adiposity and obesity in children.

PubMed Reasoner (GPT-4o): Yes
Rationale: Studies show sedentary behavior is linked with pediatric obesity. Preschool children with more sedentary time and screen use had greater odds of overweight/obesity [PMID:40751144].
Among Chinese adolescents, screen-based sedentary behavior correlated with higher BMI, mediated by weight concerns [PMID:40611060].
Reviews confirm reducing sedentary behavior improves weight outcomes [PMID:40494564].

Question: Does screen time reduction improve BMI in obese children?

GPT-4o: Yes
Rationale: Reducing screen time decreases sedentary behavior, encourages activity, and can improve BMI outcomes.

PubMed Reasoner (GPT-4o): No
Rationale: A meta-analysis of 14 RCTs (n=1,894) found screen time interventions reduced weekly screen use but did not significantly improve BMI or BMI z-scores. While effective for reducing sedentary behavior, they are insufficient alone for BMI improvement [PMID:35248682].

Question: Does physical activity combined with nutrition counseling improve BMI in obese children?

GPT-4o: Yes
Rationale: Combining physical activity with nutrition counseling effectively reduces BMI in obese children.

PubMed Reasoner (GPT-4o): Yes
Rationale: A Cochrane meta-analysis [PMID:28639319] found multi-component interventions (diet, activity, behavior change) significantly reduced BMI ($\approx$ -0.53 kg/m^2).
Another meta-analysis [PMID:32408483] reported that only combined interventions produced significant reductions ($\approx$ -0.30, $p<0.001$), whereas exercise alone did not. Individual RCTs confirm these effects.

H Prompt: LLM-as-Judge

The four dimensions capture complementary aspects of reasoning quality in biomedical QA:

- **Reasoning Soundness** focuses on the logical structure of the answer. A response should be coherent, free of contradictions, and follow a clear reasoning chain. Even if evidence is present, poor logical flow reduces interpretability.
- **Evidence Grounding** emphasizes factual support. Claims must be explicitly backed by biomedical literature, reducing hallucinations. An answer with strong logical form but unsupported claims will score lower here.
- **Clinical Relevance** measures whether the reasoning directly addresses the biomedical question in an evidence-based manner. Answers that drift into tangential findings or generic biomedical facts are penalized, even if factually correct.
- **Trustworthiness** assesses safety and adherence to established biomedical knowledge.

This ensures that the reasoning avoids misleading or potentially harmful statements, which is critical in high-stakes clinical contexts.

Together, these criteria provide a multidimensional evaluation: **soundness** ensures logical clarity, **grounding** ensures factual reliability, **relevance** ensures task alignment, and **trustworthiness** ensures clinical safety. By jointly considering all four, we obtain a more faithful measure of reasoning quality than predictive accuracy alone.

LLM-as-Judge Prompt

System Prompt: You are a neutral medical evaluator. Compare two answers from medical language models for a PubMedQA-style question. Judge *reasoning quality only* (not model identity).
Question: natural_language_question
Answer A: answer_a
Answer B: answer_b
Evaluate each answer independently on four dimensions (1-5):
1) Reasoning Soundness - logical, coherent, internally consistent.
2) Evidence Grounding - claims supported by biomedical evidence; no hallucinations.
3) Clinical Relevance - directly addresses the question in an evidence-based manner.
4) Trustworthiness - safe, conforms to biomedical knowledge; not misleading.
Instructions:
- Assign a numeric score (1-5) for each dimension to both A and B.
- Give a brief justification (less than 2 sentences) for each score.
- Provide an overall verdict based on reasoning quality: "A", "B", or "tie".
- Do not mention model names or speculate on sources.
- Output strictly valid JSON matching this schema (and nothing else):

```
{
  "Answer A": {
    "Reasoning Soundness": {
      "score": <int>,
      "justification": "<string>"
    },
    "Evidence Grounding": {
      "score": <int>,
      "justification": "<string>"
    },
    "Clinical Relevance": {
      "score": <int>,
      "justification": "<string>"
    },
    "Trustworthiness": {
      "score": <int>,
      "justification": "<string>"
    }
  },
  "Answer B": {
    "Reasoning Soundness": {
      "score": <int>,
      "justification": "<string>"
    },
    "Evidence Grounding": {
      "score": <int>,
      "justification": "<string>"
    },
    "Clinical Relevance": {
      "score": <int>,
      "justification": "<string>"
    },
    "Trustworthiness": {
      "score": <int>,
      "justification": "<string>"
    }
  },
}
```

I Evidence Sufficiency Depth

Fig. 3 illustrates the distribution of the number of retrieved papers required before reaching evidence sufficiency across PubMedQA and MMLU-CK datasets.

PubMedQA. Most questions achieve evidence sufficiency within the first five retrieved articles (65.8% for GPT and 65.2% for Gemini). Over 80% require no more than ten articles, demonstrating that PubMed Reasoner efficiently capitalizes on PubMed's ranking to identify relevant studies early.

MMLU-CK. Although questions are more diverse and less structured, more than 70% of cases still reach sufficiency within ten articles. This indicates that the reflective retriever generalizes well beyond domain-specific datasets. Fig. 3 illustrates the distribution of the number of retrieved papers required before reaching evidence sufficiency across PubMedQA and MMLU-CK datasets. For both backbones (GPT and Gemini), the majority of ques-

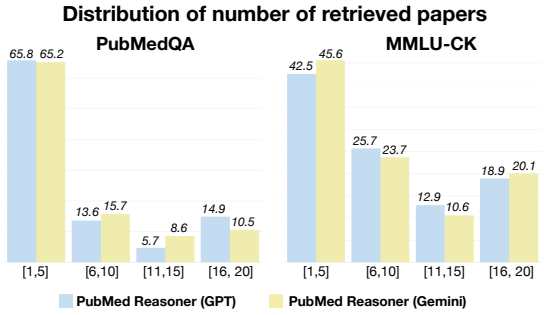

Figure 3: Effectiveness of the reflective integration stage. Distribution of retrieval depth before early stopping.

tions obtain sufficient supporting evidence within the first five retrieved articles—65.8% for PubMed Reasoner (GPT) and 65.2% for PubMed Reasoner (Gemini).

Across both datasets, ESD highlights PubMed Reasoner's ability to rapidly and adaptively locate adequate evidence, minimizing unnecessary retrieval while preserving grounding quality.

J Qualitative Error Analysis and Challenging Cases

To provide deeper insight into the limitations of PubMed Reasoner, we include here a qualitative error taxonomy alongside representative failure cases. Our analysis is based on mispredicted examples from the PubMedQA dataset using the PubMed Reasoner (GPT) variant.

Error Taxonomy and Distribution. We categorize observed failures into three common patterns. The distribution of error types is as follows:

- **Case 1: No articles retrieved (54.1%)**
  Often caused by overly specific MeSH terms added during refinement, leading the query to become too restrictive.
- **Case 2: Incomplete MeSH coverage (29.3%)**
  The refined query omits critical concepts necessary for retrieving the correct evidence.
- **Case 3: Other errors (16.6%)**
  Includes request timeouts, API rate limits, and unexpected server or connection errors.

Case 1 Example: No Articles Retrieved.

- **Question:** Do molecular signatures of mood stabilisers highlight the role of the transcription factor REST/NRSF?
- **Ground-truth MeSH terms:**
  Antimanic Agents AND Cell Line AND Humans AND Lithium Compounds AND Repressor Proteins AND Transcriptome AND Valproic Acid
- **Refined query MeSH terms:**
  Neuroblastoma AND Cells, Cultured AND Cell Line AND Lithium AND **RE1-Silencing Transcription Factor** AND Valproic Acid AND **GAD1** AND Humans AND Time Factors

Overly specific terms (bold) caused the search to become too narrow, retrieving zero results.

Case 2 Example: Incomplete MeSH Coverage.

- **Question:** Does blood viscosity but not shear stress associate with delayed flow-mediated dilation?
- **Ground-truth MeSH terms:**
  Age Factors AND Aged AND Blood Viscosity AND Brachial Artery AND Female AND Humans AND Male AND Middle Aged AND Stress, Mechanical AND Vasodilation
- **Refined query MeSH terms:**
  Middle Aged AND Aged AND Vasodilation AND Female AND Humans AND Male AND Brachial Artery AND Blood Flow Velocity AND Regression Analysis AND Cross-Sectional Studies

*Critical concepts such as **Blood Viscosity** and **Stress, Mechanical** were omitted, resulting in incomplete retrieval.*

Case 3 Example: System or API Errors. Representative failures include rate limits, connection issues, and unexpected HTTP errors.

Together, these examples highlight both conceptual and infrastructural failure modes, illustrating where PubMed Reasoner can be further improved in robustness, coverage, and query generalization.

K Ablation Study for Reflective Evidence Extraction

Table 8 presents the ablation results for the reflective integration stage on the PubMedQA test set. The comparison between PubMed Reasoner and its variant without reflective retrieval reveals that this stage substantially improves overall reasoning quality. In terms of win/tie/loss rates judged by GPT-4.1, PubMed Reasoner achieves clear advantages in *Reasoning Soundness* (76.4% wins) and *Clinical Relevance* (58.6% wins), indicating that the reflective process helps the model produce more coherent and question-aligned explanations. The improvement is also reflected in the corresponding average Likert scores, which increase from 3.422 to 3.554 for soundness and from 3.443 to 3.573 for rel-

Table 8: Explanation quality evaluation on PubMedQA test sets. Left: win/tie/loss rates from pairwise comparisons judged by GPT-4.1. Right: average Likert scores (1–5). We compare PubMed Reasoner with and without the reflective retrieval stage.

Metric	Loss / Tie / Win (%)			Avg. Likert (1–5)	
	w/o	Tie	Ours	w/o	Ours
Reasoning Soundness	19.6	4.0	**76.4**	3.422	**3.554**
Evidence Grounding	**26.2**	54.3	19.4	**3.554**	3.427
Clinical Relevance	17.6	23.8	**58.6**	3.443	**3.573**
Trustworthiness	10.0	75.8	**14.2**	**3.432**	**3.432**

evance. Interestingly, while the score for *Evidence Grounding* slightly declines (from 3.554 to 3.427), this pattern aligns with our earlier observation that the reflective stage introduces a broader range of evidence, sometimes increasing conceptual diversity at the cost of strict grounding precision. Nevertheless, the consistent or improved performance across other metrics demonstrates that reflective integration enhances interpretability and contextual reasoning, enabling PubMed Reasoner to deliver explanations that are both clinically meaningful and logically sound.

L Ablation Study on Early-Stopping Mechanism

To further investigate the effectiveness of the early-stopping mechanism, we conducted an ablation study on the PubMedQA dataset Gemini variant. In our setup, the model processes up to 20 retrieved papers, reading them in batches of five. The early-stopping mechanism terminates the reading process once the model determines that the available context is sufficient to answer the question.

To assess the performance implications of omitting this mechanism, we evaluated alternative settings in which the top 20 retrieved papers were supplied in condensed form. Owing to context-length constraints, we could not provide the full text of all 20 papers simultaneously. Instead, we considered two reduced-context variants: **(1) providing only the abstract (metadata) of each paper**, and **(2) generating a per-paper summary and concatenating these summaries** before the final question-answering step.

Table 9 reports the performance of each configuration. Both reduced-context approaches resulted in substantial degradation, demonstrating that the absence of key extracted observations from each paper significantly impairs reasoning quality.

Method	**Accuracy (%)**
Abstract Only	67.20
Paper Summary	64.12
PubMed Reasoner	**77.26**

Table 9: Ablation study comparing reduced-context variants to our selective early-stopping framework on PubMedQA.

These results highlight the importance of providing evidence at the appropriate granularity. High-level condensations, such as abstracts or naive summaries fail to capture essential reasoning cues, whereas our selective early-stopping framework preserves critical information and yields markedly improved performance.