

Sieve and Sage: Efficient Distraction Filtering for Reliable RALM Abstention

Jongbin Won and Sung Geun An and Jay-Yoon Lee*
Graduate School of Data Science / Seoul National University
{jongbinwon, ssunggun2, lee.jayyoon}@snu.ac.kr

Abstract

Just as Socrates recognized the limits of his own knowledge, Retrieval-Augmented Language Models (RALMs) should learn to *abstain* when the retrieved evidence cannot support a reliable response. Existing approaches largely rely on monolithic LLMs to handle heterogeneous retrieval failures in a single step, resulting in limited abstention performance and high computational costs. We instead decompose retrieval failures into two distinct states: (i) the *unanswerable state*, where the required evidence is absent, and (ii) the *distracted state*, where relevant evidence is mixed with conflicting, negated, or adversarial information. Based on this decomposition, we introduce a lightweight module (**Sieve**) that screens retrieved document sets for distracting evidence before invoking a costly LLM (**Sage**) for grounded generation and abstention. Evaluated across both general and high-stakes expert domains, our Sieve and Sage framework preemptively detects distracting noise, improving system accuracy by up to 69.4%p and Macro-F1 by 55.2%p compared to one-stage baselines. Furthermore, it achieves up to a 1.99 $\times$ speedup, establishing a highly efficient and reliable abstention pipeline for RALM with abstention.

1 Introduction

*Let me perform the Three Sieves test
... The third sieve is that of utility: If it
is not useful, let it not distract my mind.*

—Adapted from a Socratic anecdote

Socratic ideas offer a useful metaphor for reliable abstention: filtering distracting information before deliberation (the *Three Sieves test*), and abstaining when the available evidence is insufficient (*knowing what one does not know*). These ideas

are particularly relevant to Retrieval-Augmented Language Models (RALMs), whose reliability can degrade when retrieved documents are insufficient, irrelevant, or misleading (Yoran et al., 2024; Cuccinasu et al., 2024).

For RALMs, the objective is therefore not merely to produce a correct answer, but to provide a response that is faithfully grounded in the retrieved evidence. When the available evidence is insufficient or unreliable, abstention can be preferable to an unsupported answer. This is especially important in high-stakes expert domains, where abstention can itself serve as a useful signal for follow-up actions.

Prior approaches to RALM abstention have largely relied on Large Language Models, either through fine-tuning (Sun et al., 2025; Gul et al., 2026) or external LLM-based judges (Joren et al., 2025). By treating retrieval noise as a homogeneous category, such approaches incur substantial computational overhead while providing limited distinction between different causes of abstention.

In contrast, we propose a two-stage framework that distinguishes two forms of retrieval failure: *distracted* states, where relevant evidence is mixed with conflicting or adversarial information, and *unanswerable* states, where sufficient evidence is absent. This distinction operationalizes the two Socratic ideas above: the former calls for screening distracting information before deeper reasoning, whereas the latter calls for recognizing when there is not enough evidence to answer.

Because these failure modes require different handling, we assign them to specialized components. A lightweight BERT-based classifier detects distracted inputs before invoking a costly LLM, while the LLM is reserved for grounded generation and abstention under insufficient evidence. Beyond improving efficiency, identifying the underlying retrieval state also enables state-aware post-abstention strategies tailored to the detected failure

*Corresponding author.

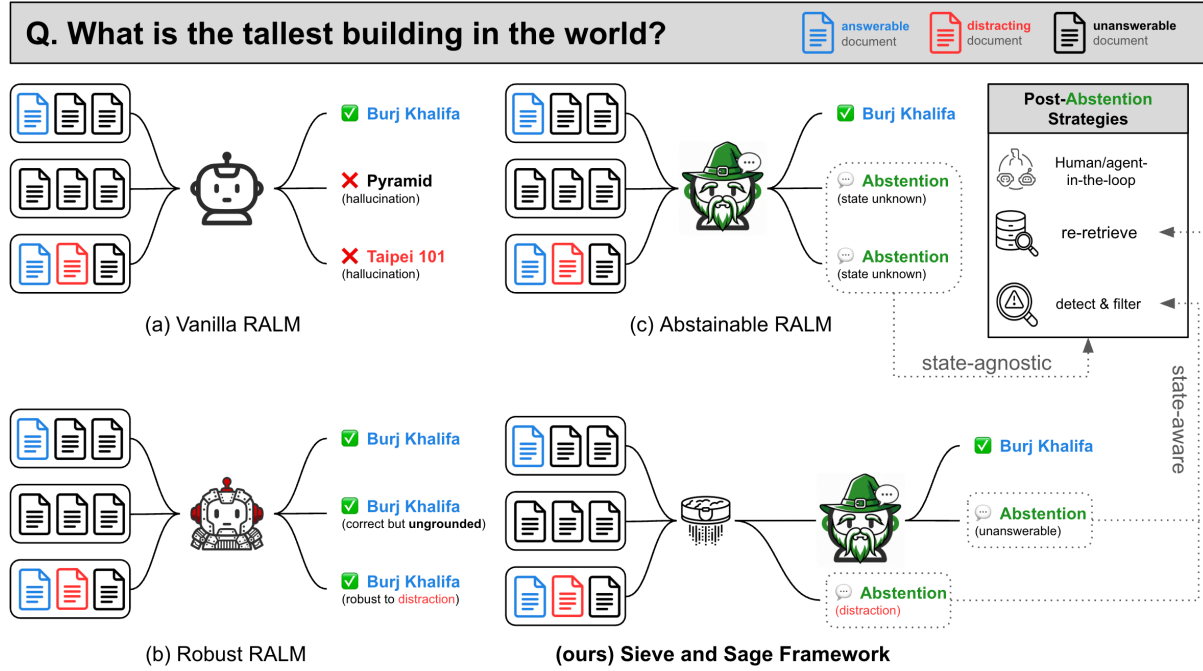


Figure 1: **Comparison between conventional LLM-centric approaches and Sieve and Sage.** While previous robustness and abstention studies rely solely on the internal judgment of a monolithic LLM, our framework introduces a decoupled architecture: a lightweight Sieve for preemptive distraction filtering and a generative Sage for high-fidelity generation and abstention for unanswerable instances. Unlike (c) conventional abstainable RALMs that produce a state-agnostic abstention signal, (d) our Sieve and Sage framework identifies the underlying retrieval state, enabling state-aware post-abstention strategies such as re-retrieval for unanswerable cases and distraction filtering for distracted cases.

mode. Figure 1 contrasts this architecture with conventional monolithic RALMs.

To evaluate our framework, we systematically simulate sophisticated distracting noise—including conflicting, negated, and adversarial documents—via entity substitution, NLI, and LLM generation. This enables a multi-faceted assessment across both general and expert domains. Our Sieve and Sage framework improves system accuracy by up to 69.4%p and Macro-F1 by 55.2%p over vanilla LLMs, while achieving up to a $1.99\times$ inference speedup over monolithic RALM baselines such as DTA-chatQA. These results demonstrate that explicitly modeling retrieval states can improve both reliability and efficiency.

Our contributions are summarized as follows:

- **Retrieval Noise Decomposition:** We distinguish two forms of retrieval failure—*distraction*, where relevant evidence is mixed with misleading information, and *unanswerability*, where sufficient evidence is absent. This decomposition enables state-aware abstention rather than treating retrieval noise as a single homogeneous category.

- **Efficient Sieve-and-Sage Framework:** We introduce a two-stage architecture that assigns distraction detection to a lightweight Sieve and reserves the generative Sage for grounded answering and abstention under insufficient evidence. This specialization reduces unnecessary LLM computation while exposing the underlying retrieval state for targeted post-abstention handling.

- **Evaluation Across Diverse Domains and Noises:** We evaluate the framework across general and expert-domain datasets under multiple forms of distracting noise. Our approach improves Macro-F1 by up to 55.2%p while reducing inference latency by up to $1.99\times$ compared with monolithic baselines.

2 Related Works

2.1 RALM Under Retrieval Noise

Retrieval-Augmented Language Models (RALMs) (Ram et al., 2023; Shi et al., 2024) ground LLMs in external knowledge for more faithful and reliable responses. However, retrieval can fail due to

Q. What is the tallest building in the world?

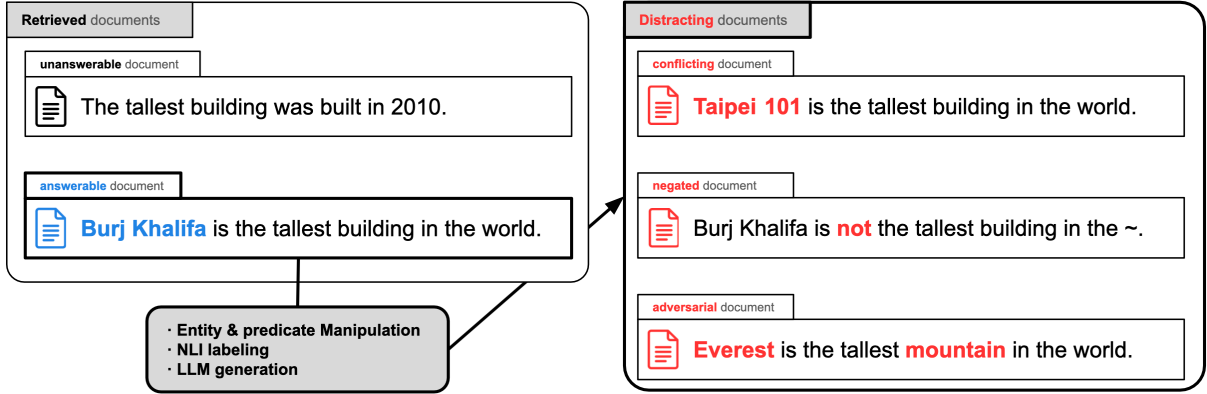


Figure 2: **Examples of retrieved and simulated documents.** To evaluate the abstention under diverse retrieval noise, we simulated various distracting documents by perturbing answerable documents via entity/predicate manipulation, NLI labeling, and LLM generation.

retriever limitations, corrupted datastores, or adversarial attacks (Zhong et al., 2023; Zou et al., 2024). Recent studies therefore investigate RALM behavior under noisy retrieval, where relevant evidence may be absent, irrelevant, or misleading (Yoran et al., 2024; Fang et al., 2024; Joren et al., 2025).

Meanwhile, existing evaluations on RALMs have largely focused on general-domain benchmarks such as NQ (Lee et al., 2019), WebQ (Bordes et al., 2014), and HotpotQA (Yang et al., 2018). These benchmarks often overlook expert domains, where parametric knowledge may be limited and unsupported responses can be costly (Wang et al., 2024; Thakur et al., 2024). We therefore evaluate our framework across both general and specialized medical, biological, and astronomical domains.

2.2 Enhanced RALM: Robustness vs. Abstention

Given that retrieved document sets are inevitably noisy, prior work has largely pursued two complementary directions: **robustness** and **abstention**.

Robustness in RALMs Robustness-oriented approaches aim to generate correct answers despite irrelevant or distracting retrieved contexts. Existing strategies include exposing models to low-ranked contexts during supervised fine-tuning (Yoran et al., 2024), adaptive adversarial training (Fang et al., 2024), and auxiliary lightweight modules for assessing or filtering retrieved evidence (Yan et al., 2024; Kim and Lee, 2024; Yu et al., 2024).

However, robustness can be misaligned with the practical goals of RALMs: by forcing an answer,

such models may compromise faithfulness or, more critically, fail to signal uncertainty. In expert domains, the absence of an explicit error signal can be particularly dangerous, as a plausible but ungrounded response may be more harmful than an admission of uncertainty.

Abstention in RALMs Abstention-oriented approaches instead enable RALMs to refrain from answering when the available evidence is insufficient or unreliable. Existing approaches include post-training models to learn abstention behavior (Sun et al., 2025; Gul et al., 2026), prompting LLMs to reason about retrieval reliability through in-context learning (Park et al., 2024), and using external LLM-based sufficiency judges (Joren et al., 2025).

However, previous abstention approaches remain largely LLM-centric, incurring substantial computational costs while remaining vulnerable to the over-trust issue, where LLMs may blindly favor their own generations over a more reliable silence (Li et al., 2024).

In contrast, we decompose retrieval noise into two distinct states—*unanswerable* and *distracted* (S_{unans} and S_{distract})—and assign them to different stages of the abstention pipeline. This decomposition enables a lightweight classifier to preemptively detect distracting evidence before LLM inference, reducing computational overhead while supporting early exit and state-specific post-abstention strategies.

Most closely related to our work, Hong et al.

(2024) employ a lightweight discriminator to detect counterfactual evidence for robust answer generation. While their focus is robustness under conflicting evidence, our framework targets system-level abstention by explicitly distinguishing distraction from missing evidence and handling these states separately.

3 Methodology

3.1 Problem Formulation

To formalize RALM with abstention, we characterize the retrieval state S of the document set $D = \{d_1, d_2, \dots, d_k\}$ retrieved for a given query q . Prior work on RALM robustness and abstention has often treated unreliable retrieval as a coarse category without explicitly distinguishing its underlying causes (Yoran et al., 2024; Xiang et al., 2024). We instead propose a more fine-grained tripartite taxonomy ($S \in \{S_{\text{ans}}, S_{\text{unans}}, S_{\text{distract}}\}$) based on document-level evidence composition.

This distinction further enables state-specific follow-up strategies after abstention, as discussed in Section 6.

Document-level Evidence Assessment We categorize each document $d \in D$ by lexical matching (checking if d contains the ground-truth answer a^*) and semantic consistency via NLI against a reference sentence generated by LLM conditioned on q and a^* :

- **Answerable document** (d_{ans}): Contains a^* and entails the reference sentence. It serves as primary evidence.
- **Unanswerable document** (d_{unans}): Neither contains a^* nor entails the reference. It provides no valid grounds, while not misleading the system.
- **Distracting document** (d_{distract}): Contains misleading or inconsistent evidence that can divert the system from the grounded answer. In our controlled evaluation, distracting documents are instantiated as three types—*conflict*, *negated*, and *adversarial* documents—by manipulating key entities or their associated predicates. Although all three types induce the same set-level distracted state (S_{distract}), they represent diverse forms of retrieval noise, including factual inconsistency, logical negation, and targeted misleading evidence. Importantly, the Sieve is trained using only conflict

documents as positive distracted examples and is evaluated across all three types, allowing us to assess its generalization beyond the type of noise observed during training. Detailed implementation is provided in Section 4.1 and Figure 2.

Documents that do not meet these criteria—such as those containing a^* but remaining semantically neutral to the reference sentence—are marked as **neutral** (d_{neutral}) and assumed to have a negligible impact on the retrieval state S . We empirically validate this assumption in Appendix D, where adding up to three neutral documents results in minor performance variations.

Set-level Retrieval States Building on the document-level assessment, we define three retrieval states:

- **Answerable state** (S_{ans}): D contains at least one d_{ans} and no d_{distract} . It may include d_{unans} or d_{neutral} as noise. The system is expected to generate a^* .
- **Unanswerable state** (S_{unans}): D consists entirely of d_{unans} . The system should recognize the lack of evidence and refrain from answering (a_{unans}).
- **Distracted state** (S_{distract}): D contains at least one d_{ans} and at least one d_{distract} . The system should detect the evidentiary distraction and reject the set (a_{distract}).

Under this taxonomy, RALM with abstention is represented by the decision function f :

$$f(q, D) = \begin{cases} a_{\text{grounded}}, & \text{if } D \in S_{\text{ans}}, \\ a_{\text{unans}}, & \text{if } D \in S_{\text{unans}}, \\ a_{\text{distract}}, & \text{if } D \in S_{\text{distract}}. \end{cases}$$

where a_{grounded} denotes a grounded answer generated from the retrieved evidence, while a_{unans} and a_{distract} denote distinct abstention signals indicating insufficient and distracting evidence, respectively.

3.2 Sieve and Sage: Efficient Two-stage Abstention Framework

To implement $f(q, D)$ defined above, we propose a two-stage framework that decouples distraction detection from grounded generation by employing

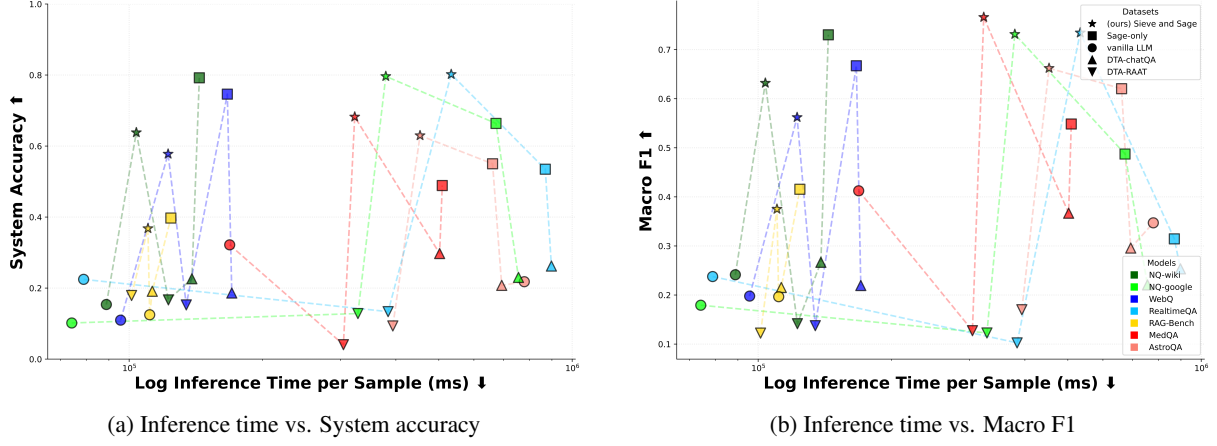


Figure 3: **Pareto graph of inference time versus System Accuracy/Macro F1.** Our framework(★) is up to $2\times$ faster than other LLM-centric approaches with comparable performance, especially achieving the highest F1 scores in expert domain datasets.

Sieve (a lightweight classifier) and **Sage** (a generative LLM). Our framework separates three functions that are often conflated in monolithic RALMs: (1) distraction detection by the Sieve, (2) missing-evidence detection by the Sage, and (3) grounded answer generation by the Sage. Abstention is then triggered when either distraction or missing evidence is identified.

Stage 1: The Sieve for Distraction Detection

The Sieve g_ϕ acts as an efficient gatekeeper to preemptively identify distracted document sets (S_{distract}). Intercepting distracting inputs before LLM invocation reduces unnecessary generation and computational overhead. Formally:

$$\hat{y}_{\text{distract}} = \mathbb{I}(\sigma(g_\phi(q, D)) \geq \tau_{\text{distract}})$$

Given a predefined threshold τ_{distract} , if $\hat{y}_{\text{distract}} = 1$, the system immediately outputs a_{distract} , bypassing the subsequent LLM invocation entirely. We set $\tau_{\text{distract}} = 0.5$ as the default threshold, while the Sieve’s classification performance remains stable across a practical range of alternative thresholds (Appendix H).

Stage 2: The Sage for Grounded Generation

Document sets that pass the Sieve (i.e., $\hat{y}_{\text{distract}} = 0$) are forwarded to the Sage, a generative LLM P_θ . Having been screened for distracting information, the Sage focuses on either generating a grounded answer or identifying the absence of sufficient evidence (S_{unans}). Formally,

$$\hat{a} \sim P_\theta(a | q, D), \quad \hat{a} = \begin{cases} a_{\text{grounded}}, & \text{if } D \in S_{\text{ans}}, \\ a_{\text{unans}}, & \text{if } D \in S_{\text{unans}}. \end{cases}$$

where the Sage is fine-tuned to generate a grounded answer a_{grounded} for S_{ans} and an abstention signal a_{unans} for S_{unans} . Unlike the Sieve, the Sage does not rely on an explicit decision threshold; it directly generates either a grounded answer or the designated abstention signal.

This hierarchical design ensures that the high-cost LLM is invoked only after the retrieved document set has been screened for distraction by the lightweight Sieve.

Fine-tuning Strategy To optimize the two-stage framework, we employ task-specific optimization for each component. During training, the Sieve learns a binary decision boundary between S_{ans} and S_{conflict} :

$$\mathcal{L}_{\text{Sieve}} = - \sum_i [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)],$$

$$\text{where } p_i = \sigma(g_\phi(q, D_i)).$$

In contrast, the Sage is optimized via supervised fine-tuning (SFT) to maximize the likelihood of the ground-truth a^* for S_{ans} and the designated abstention signal a_{unans} for S_{unans} :

$$\mathcal{L}_{\text{Sage}} = - \sum_{t=1}^{|Y|} \log P_\theta(y_t | y_{<t}, q, D),$$

These task-specific objectives enable the Sieve to specialize in distraction detection while training the Sage to generate grounded answers when sufficient evidence is available and abstain when the evidence is insufficient.

test(train) datasets	General domain					Expert domain		
	NQ-wiki	NQ-google	WebQ	RealtimeQA	RAG-Bench	MedQA	BioASQ	AstroQA
document source	Wikipedia	Google search	Wikipedia	Google search	Wikipedia	PubMed abstracts	PubMed abstracts	Astro abstracts
document length	165.93	152.29	172.63	145.27	409.29	1024.47	955.77	1046.2
# total samples	3297 (44030)	226 (2182)	2096	187	4052	491 (2087)	610 (3997)	289 (1489)
# answerable	889 (15086)*	51 (418)*	545*	34*	912*	116 (409)	75 (536)*	99 (460)
# unanswerable	483 (20688)*	49 (382)*	296*	66*	419*	159 (539)	397 (3236)*	86 (409)
# conflict	530 (8256)*	47 (396)	360*	34	907*	70 (273)	33 (225)	47 (216)
# negated	889	51	545	34	907	77	59	47
# adversarial	506	28	350	19	907*	69	46	10

Table 1: **Statistics of test (and train) datasets.** An asterisk (*) indicates that the documents are adopted from prior studies rather than retrieved or simulated in this work.

4 Experimental Setup

4.1 Dataset Construction

To evaluate the performance of RALMs across diverse contexts, we employed 5 general-domain datasets (NQ-wiki(Park et al., 2024), NQ-google(Xiang et al., 2024), WebQ(Park et al., 2024), RealtimeQA(Xiang et al., 2024), RAG-Bench(Fang et al., 2024)) and 3 expert-domain datasets (AstroQA(de Haan et al., 2025), MedQA(Pal et al., 2022), BioASQ(Nentidis et al., 2025)).

To maintain consistency with established benchmarks, we adopt the document sets and retrieval noise from prior studies for the datasets marked with an asterisk (*) in Table 1.

For BioASQ and AstroQA datasets, we utilized ColBERTv2 (Santhanam et al., 2022) to retrieve the top-20 candidate documents per query. Following the criteria defined in Section 3.1, documents are labeled using domain-specific NLI models¹ and the retrieval state S is determined by the top-5 documents. An ablation over the number of documents K is provided in Appendix E.

Simulating distracting documents d_{distract} To evaluate abstention under diverse retrieval noise, we manipulate reference answer sentences into three distracting types (d_{conflict} , d_{negated} , $d_{\text{adversarial}}$). Each is expanded into a full-length document matching the data store’s length distribution. The manipulation is validated via entity substitution, NLI, and LLM verification, following and extending prior works (Park et al., 2024; Fang et al., 2024):

- **Conflicting document (d_{conflict}):** We replace

¹We used `cnut1648/biolinkbert-mednli` for MedQA and BioASQ, `auffinger/scibert_scivocab_uncased-mnli` for AstroQA, and `MoritzLaurer/mDeBERTa-v3-base-xnli-multilingual-nli-2mil7` for other general datasets.

the ground-truth entity a^* with a plausible but incorrect alternative. Validation ensures a^* is absent, and the document explicitly contradicts the reference.

- **Negated document (d_{negated}):** We retain a^* but negate its associated predicate. Validation ensures a^* remains while the semantic meaning is contradictory, thereby testing the model’s reliance on keyword heuristics.
- **Adversarial ($d_{\text{adversarial}}$):** Both the entity and its context are shifted to a different subject. Validation ensures the document neither contains the ground-truth entity a^* nor is contradictory to the reference.

Concrete examples and analyses of the simulated documents are provided in Figure 2 and Appendix A.

4.2 Implementation Details

For our framework, we use Longformer (Beltagy et al., 2020)² as the Sieve and Llama-3-8B-Instruct (Grattafiori et al., 2024)³ as the Sage. We choose Longformer to accommodate long retrieved contexts, particularly in expert-domain datasets. See Appendix J for results with alternative Sieve backbones and Sage model families.

Each component is fine-tuned on NQ dataset (Lee et al., 2019) according to the strategies described in Section 3.2. Detailed configurations are provided in Appendix C. A detailed comparison of the computational footprint of the two components, including parameter count, VRAM usage, and inference FLOPs, is provided in Appendix F.

Domain Adaptation To evaluate our framework in specialized domains, we adapt both the generator and the classifier.

²`allenai/longformer-base-4096`

³`meta-llama/Meta-Llama-3-8B-Instruct`

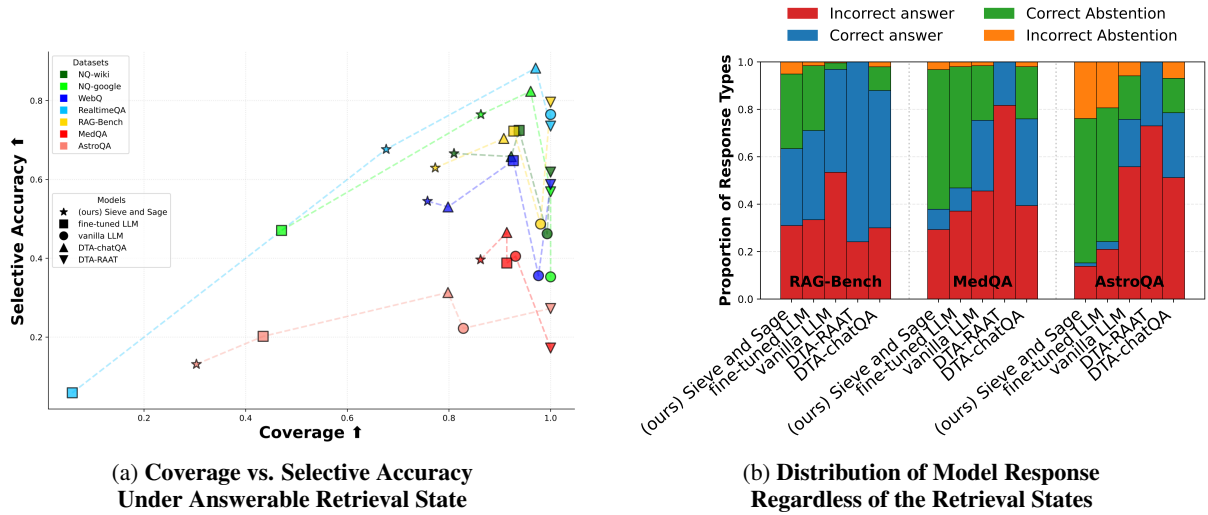


Figure 4: **Selective Prediction and System Reliability.** We evaluate system reliability through two complementary analyses: (a) Coverage and Selective Accuracy for answerable examples (precision on S_{ans}) and (b) distribution of model response across all retrieval states. Although Sieve and Sage operates at lower coverage than more aggressive baselines, this reflects a more conservative response policy and is accompanied by fewer incorrect responses on several datasets.

For LLM generation, expert domains require strong parametric knowledge. We therefore employ domain-specific variants: Med42-8B (Christophe et al., 2024) for MedQA and BioASQ, and AstroSage-8B (de Haan et al., 2025) for AstroQA.

In contrast, the classifier primarily detects distractions, relying more on structural reasoning than domain-specific knowledge. To accommodate the distributional characteristics of expert documents (e.g., longer passages and terminology), we perform minimal structural adaptation using fewer than 200 samples per domain. Standalone evaluations further show that the Sieve achieves substantial adaptation gains with fewer than 200 target-domain examples and transfers effectively across expert domains (Appendix B).

4.3 Evaluation Metrics

We evaluate the performance of Sieve and Sage along two primary dimensions: response accuracy for answerable instances and retrieval state classification.

For response accuracy, we measure how often the model attempts to answer answerable queries (**Coverage**) and how often those responses are correct (**Selective Accuracy**). Coverage denotes the proportion of answerable instances for which the system chooses to respond, while Selective Accuracy measures the correctness of the responses conditioned on those instances where the system produces an answer.

For classification, we treat the decision process as a tripartite task (S_{ans} , S_{unans} , S_{distract}) and report **State-Specific F1** scores (Answerable, Unanswerable, and Distract F1) to evaluate the model’s ability to identify different retrieval states. In addition, we report **Macro-F1** (unweighted mean) to provide a balanced assessment across all categories and prevent bias toward the majority state.

Finally, we report **System Accuracy** to jointly evaluate response generation and state classification. A prediction is considered correct if the system either produces the ground-truth answer a^* for S_{ans} or appropriately abstains for S_{unans} and S_{distract} .

4.4 Baselines

We compare our framework against three other approaches: (1) vanilla LLM, (2) LLM fine-tuned to abstain when the documents are noisy (**Sage-only**) (Zhang et al., 2024), and (3) RALM aligned with knowledge-quadrant preference data to enable abstention (**DTA-chatQA**, **DTA-RAAT**) (Sun et al., 2025). Compared to DTA-chatQA, DTA-RAAT is trained with a stronger emphasis on robustness to retrieval noise, aiming to generate correct answers even in the presence of noisy or distracting documents.

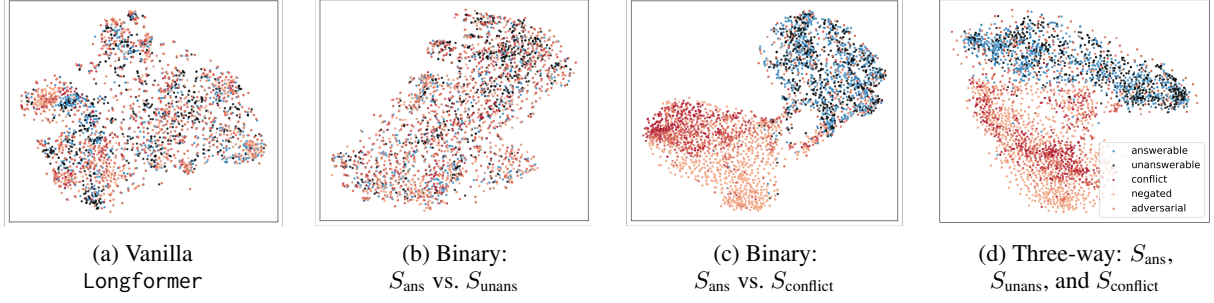


Figure 5: **t-SNE comparison of Sieve training objectives.** Our model (c), fine-tuned to classify $S_{conflict}$ from S_{ans} , achieves a more distinct separation of distracting noise compared to the model (d) fine-tuned to classify answerable, unanswerable, and conflicting states at once. This validates that a specialized focus on factual consistency establishes a precise decision boundary for identifying retrieval distractions.

5 Experimental Results and Analysis

5.1 Efficiency and Effectiveness Analysis

Figure 3 illustrates the Pareto graph for System Accuracy and Macro F1 against average inference latency. Across all datasets, Sieve and Sage achieve competitive performance while maintaining lower latency than most monolithic LLM baselines. Notably, compared to DTA-chatQA, our framework achieves up to a $1.99\times$ speedup (379.6 ms vs. 756.3 ms on NQ-google) while delivering a 56.6% gain in System Accuracy (0.796 vs. 0.230).

The advantages of our decoupled architecture are most pronounced in expert domains. While monolithic LLMs suffer from increased latency and severe performance degradation when processing complex, distracting documents, our framework maintains stability. For instance, on MedQA, Sieve and Sage achieve the highest System Accuracy (0.682) and Macro F1 (0.766) compared to DTA-chatQA (0.297, 0.367), while simultaneously reducing inference time by over 35% (322.9 ms vs. 502.1 ms). The efficiency gain is further supported by the Sieve’s substantially smaller computational footprint—approximately $54\times$ fewer parameters and $14\times$ fewer inference FLOPs than the Sage—which amortizes the cost of expensive LLM inference by avoiding unnecessary Sage calls (Appendix F).

These empirical results confirm that explicitly decoupling distraction detection from core reasoning effectively mitigates computational overhead, establishing an optimal balance in both speed and reliability. Comprehensive numerical results and robustness analyses across alternative retrievers, Sieve backbones, and Sage model families are provided in Appendix J.

5.2 Reliability Analysis

Figure 4a illustrates the relationship between Coverage and Selective Accuracy for answerable instances, while Figure 4b evaluates overall system reliability by analyzing the distribution of model response regardless of the retrieval states.

A key observation is that Sieve and Sage intentionally sacrifice a small amount of Coverage and Selective Accuracy to significantly reduce incorrect answers. On MedQA, Coverage and Selective Accuracy decrease from 0.914 and 0.509 to 0.862 and 0.460, respectively, compared to DTA-chatQA. However, incorrect answers drop by 56.1% (319 $\rightarrow$ 140), while correct abstentions increase substantially (108 $\rightarrow$ 289).

In contrast, baseline models such as the Vanilla LLM and DTA-RAAT attempt to respond to almost every query (Coverage $\simeq 1.0$). This aggressive strategy leads to a large accumulation of incorrect responses. On MedQA, the Vanilla LLM produces 323 incorrect answers among 370 responses (87.3%), while on AstroQA, DTA-RAAT produces 262 incorrect answers among 289 responses (90.7%).

These results highlight a key limitation of existing RALMs: maximizing coverage in noisy environments leads to many incorrect responses. In contrast, Sieve and Sage demonstrates that a calibrated, conservative generation strategy is crucial for reliable deployment, especially in high-stakes expert domains.

5.3 Analysis of Sieve Design

Figure 5 shows the t-SNE visualization of CLS token embeddings for the Sieve under different training objectives. Our proposed approach (c), which focuses on distinguishing the conflict state

from the answerable state, exhibits the most distinct separation between clusters compared to other variants.

In contrast, the model (d), fine-tuned to classify all retrieval states (S_{ans} , S_{unans} , S_{conflict}) simultaneously, produces substantial overlap in the embedding space. This suggests that the multi-class objective weakens the model’s ability to detect subtle factual inconsistencies.

Interestingly, although the Sieve was trained only on answerable and conflicting examples, it generalizes well to unseen distracted states such as negated or adversarial cases. This indicates that the Sieve does not simply memorize specific noise patterns, but instead learns to detect underlying inconsistencies among the documents.

An isolated component ablation further confirms that both the Sieve and Sage contribute independently to the overall performance, with their combination yielding the strongest results (Appendix G).

6 State-Aware Post-Abstention Strategies

When a RALM abstains, the outcome need not be treated as a terminal failure. Instead, the abstention signal can trigger post-abstention strategies, such as additional reasoning or self-consistency. Sieve and Sage further identifies the underlying cause of abstention—missing evidence (S_{unans}) or distracting evidence (S_{distract})—enabling state-specific follow-up actions as follows.

Human/Agent-in-the-loop Regardless of the underlying abstention state, abstained queries can be escalated to a human expert or a more capable LLM. Such a cascade reserves expensive expert resources for the subset of examples identified as unreliable by the base system.

Dynamic Re-retrieval When the Sage identifies an unanswerable state (S_{unans}), the system can initiate an additional retrieval step to search for missing evidence. This directly addresses the underlying cause of abstention rather than repeatedly reasoning over an insufficient document set.

Distraction Filtering When the Sieve detects a distracted state (S_{distract}), the retrieved set can instead be examined to identify and remove distracting documents. Owing to its lightweight nature, the Sieve can be repeatedly applied to different document subsets; in our implementation, we perform leave-one-out filtering by removing each of the top-5 documents in turn and re-evaluating the

Dataset	Strategy	Accuracy
NQ-google	No Recovery	0.247
	+ Distraction Filtering	0.867
	+ Agent-in-the-loop	0.716
RealtimeQA	No Recovery	0.380
	+ Distraction Filtering	0.943
	+ Agent-in-the-loop	0.774

Table 2: **Accuracy on distracted examples detected by the Sieve under different post-abstention strategies.** Results are evaluated on 113 NQ-google and 71 RealtimeQA examples identified as distracted. No Recovery denotes retaining the original abstention outcome without applying a follow-up strategy.

remaining set. The cleaned document set is then forwarded to the Sage for generation.

As shown in Table 2, these strategies substantially improve accuracy on distracted examples over the no-recovery baseline.

7 Conclusion

We introduced Sieve and Sage, a framework for RALM with abstention that decouples the detection of distracting document sets from core reasoning. A lightweight Longformer-based classifier (Sieve) screens retrieved document sets for distracting evidence before invoking the generative LLM (Sage), which performs grounded generation or abstains when sufficient evidence is unavailable.

Across general and expert domains, our framework achieves up to a $1.99\times$ inference speedup and a 56.6%p gain in system accuracy. While this leads to a slight reduction in coverage and selective accuracy, it reduces incorrect responses by up to 84.2%, substantially improving overall reliability.

The explicit distinction between distracted and unanswerable states further supports targeted strategies such as distraction filtering and re-retrieval.

Overall, our results demonstrate that structured control over when and why to abstain can improve the efficiency and reliability of RALMs in high-stakes expert domains.

Limitations

Simulated retrieval noise Our evaluation primarily relies on constructed retrieval conditions, including contexts with missing evidence and simulated conflict, negated, and adversarial distractions. Although these benchmarks cover diverse forms of retrieval failure and exhibit consistent trends across datasets, they cannot fully capture the complexity

of naturally occurring noise in real-world retrieval pipelines. Future work should evaluate the framework under naturally occurring retrieval failures and dynamically changing knowledge sources to establish its external validity.

Generalization to Zero-Shot Scenarios Although the Sieve adapts to expert domains with fewer than 200 labeled examples, some domain-specific supervision remains necessary to address distribution shifts in document structure and terminology. Developing a fully zero-shot, domain-agnostic Sieve that can recognize structural distractions without target-domain fine-tuning remains an important direction for future work.

Trade-off between abstention and selective accuracy The conservative response policy adopted by Sieve and Sage reduces incorrect responses but can lower coverage in some settings. This indicates a potential trade-off between avoiding harmful errors and preserving answering utility. The appropriate operating point may depend on the relative costs of incorrect answers and abstentions in the target application. Future work may explore adaptive threshold calibration and multi-objective training to better control this trade-off.

Acknowledgments

This work was supported in part by the National Research Foundation of Korea (NRF) grant (RS-2023-00280883, RS-2023-00222663); by the National Research Foundation, Korea, under project BK21 FOUR(Dept. of Data Science, SNU, No. 5199990914569); by the Korea Institute of Science and Technology Information (KISTI) in 2026 (No. (KISTI)K26L3M1C1), aimed at developing KONI (KISTI Open Neural Intelligence), a large language model specialized in science and technology; and by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (RS-2025-02263754, Human-Centric Embodied AI Agents with Autonomous Decision-Making; RS-2025-25442149, LG AI STAR Talent Development Program for Leading Large-Scale Generative AI Models in the Physical AI Domain; and RS-2025-25463302, Global Talent Recruitment for AI-powered Drug Discovery); by grant (25202MFDS003) from Ministry of Food and Drug Safety in 2025; Institute of Information & communications Technology Planning & Evaluation (IITP)

grant funded by the Korea government(MSIT) (No. RS-2025-25442149, LG AI STAR Talent Development Program for Leading Large-Scale Generative AI Models in the Physical AI Domain); and by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (RS-2025-25463302, Global Talent Recruitment for AI-powered Drug Discovery).

References

- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. [Longformer: The Long-Document Transformer](#). *Preprint*, arXiv:2004.05150.
- Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. [Question Answering with Subgraph Embeddings](#). *Preprint*, arXiv:1406.3676.
- Clément Christophe, Praveen K. Kanithi, Prateek Munjal, Tathagata Raha, Nasir Hayat, Ronnie Rajan, Ahmed Al-Mahrooqi, Avani Gupta, Muhammad Umar Salman, Gurpreet Gosal, Bhargav Kanakiya, Charles Chen, Natalia Vassilieva, Boulbaba Ben Amor, Marco AF Pimentel, and Shadab Khan. 2024. [Med42 – Evaluating Fine-Tuning Strategies for Medical LLMs: Full-Parameter vs. Parameter-Efficient Approaches](#). *Preprint*, arXiv:2404.14779.
- Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonello, and Fabrizio Silvestri. 2024. [The Power of Noise: Redefining Retrieval for RAG Systems](#). In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 719–729.
- Tijmen de Haan, Yuan-Sen Ting, Tirthankar Ghosal, Tuan Dung Nguyen, Alberto Accomazzi, Azton Wells, Nesar Ramachandra, Rui Pan, and Zechang Sun. 2025. [AstroMLab 3: Achieving GPT-4o Level Performance in Astronomy with a Specialized 8B-Parameter Large Language Model](#). *Scientific Reports*, 15(1):13751.
- Feiteng Fang, Yuelin Bai, Shiwen Ni, Min Yang, Xiaojun Chen, and Ruifeng Xu. 2024. [Enhancing Noise Robustness of Retrieval-Augmented Language Models with Adaptive Adversarial Training](#). *Preprint*, arXiv:2405.20978.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The Llama 3 Herd of Models](#). *Preprint*, arXiv:2407.21783.

- Mustafa Omer Gul, Claire Cardie, and Tanya Goyal. 2026. [Mash: Modeling abstention via selective help-seeking](#). *Preprint*, arXiv:2510.01152.
- Giwon Hong, Jeonghwan Kim, Junmo Kang, Sung-Hyon Myaeng, and Joyce Jiyoung Whang. 2024. [Why so gullible? enhancing the robustness of retrieval-augmented models against counterfactual noise](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2474–2495, Mexico City, Mexico. Association for Computational Linguistics.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. [Lora: Low-rank adaptation of large language models](#). *Preprint*, arXiv:2106.09685.
- Hailey Joren, Jianyi Zhang, Chun-Sung Ferng, Da-Cheng Juan, Ankur Taly, and Cyrus Rashtchian. 2025. [Sufficient Context: A New Lens on Retrieval Augmented Generation Systems](#). *Preprint*, arXiv:2411.06037.
- Kiseung Kim and Jay-Yoon Lee. 2024. [RE-RAG: Improving Open-Domain QA Performance and Interpretability with Relevance Estimator in Retrieval-Augmented Generation](#). *Preprint*, arXiv:2406.05794.
- Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. [Latent Retrieval for Weakly Supervised Open Domain Question Answering](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.
- Moxin Li, Wenjie Wang, Fuli Feng, Fengbin Zhu, Qifan Wang, and Tat-Seng Chua. 2024. [Think Twice Before Trusting: Self-Detection for Large Language Models through Comprehensive Answer Reflection](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11858–11875, Miami, Florida, USA. Association for Computational Linguistics.
- Anastasios Nentidis, Georgios Katsimpras, Anastasia Krithara, Martin Krallinger, Miguel Rodríguez-Ortega, Eduard Rodríguez-López, Natalia Loukachevitch, Andrey Sakhovskiy, Elena Tutubalina, Dimitris Dimitriadis, Grigorios Tsoumakas, George Giannakoulas, Alexandra Bekiaridou, Athanasios Samaras, Giorgio Maria Di Nunzio, Nicola Ferro, Stefano Marchesin, Marco Martinelli, Gianmaria Silvello, and Georgios Paliouras. 2025. [Overview of bioasq 2025: The thirteenth bioasq challenge on large-scale biomedical semantic indexing and question answering](#). *Preprint*, arXiv:2508.20554.
- Ankit Pal, Logesh Kumar Umapathi, and Malaikanan Sankarasubbu. 2022. [MedMCQA : A Large-scale Multi-Subject Multi-Choice Dataset for Medical domain Question Answering](#). *Preprint*, arXiv:2203.14371.
- SeongIl Park, Seungwoo Choi, Nahyun Kim, and Jay-Yoon Lee. 2024. [Enhancing robustness of retrieval-augmented language models with in-context learning](#). In *Proceedings of the 3rd Workshop on Knowledge Augmented Methods for NLP*, pages 93–102, Bangkok, Thailand. Association for Computational Linguistics.
- Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. [In-Context Retrieval-Augmented Language Models](#). *Preprint*, arXiv:2302.00083.
- Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. [ColBERTv2: Effective and Efficient Retrieval via Lightweight Late Interaction](#). *Preprint*, arXiv:2112.01488.
- Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. [REPLUG: Retrieval-Augmented Black-Box Language Models](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384, Mexico City, Mexico. Association for Computational Linguistics.
- Xin Sun, Jianan Xie, Zhongqi Chen, Qiang Liu, Shu Wu, Yuehe Chen, Bowen Song, Zilei Wang, Weiqiang Wang, and Liang Wang. 2025. [Divide-Then-Align: Honest Alignment based on the Knowledge Boundary of RAG](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11461–11480, Vienna, Austria. Association for Computational Linguistics.
- Nandan Thakur, Luiz Bonifacio, Crystina Zhang, Odunayo Ogundepo, Ehsan Kamaloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Boxing Chen, Mehdi Rezagholizadeh, and Jimmy Lin. 2024. [“Knowing When You Don’t Know”: A Multilingual Relevance Assessment Dataset for Robust Retrieval-Augmented Generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12508–12526, Miami, Florida, USA. Association for Computational Linguistics.
- Shuting Wang, Jiongnan Liu, Shiren Song, Jiehan Cheng, Yuqi Fu, Peidong Guo, Kun Fang, Yutao Zhu, and Zhicheng Dou. 2024. [Domain-RAG: A Chinese Benchmark for Evaluating Domain-specific Retrieval-Augmented Generation](#). *Preprint*, arXiv:2406.05654.
- Chong Xiang, Tong Wu, Zexuan Zhong, David Wagner, Danqi Chen, and Prateek Mittal. 2024. [Certifiably Robust RAG against Retrieval Corruption](#). *Preprint*, arXiv:2405.15556.
- Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. [Corrective Retrieval Augmented Generation](#). *Preprint*, arXiv:2401.15884.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering](#). *Preprint*, arXiv:1809.09600.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. [Making Retrieval-Augmented Language Models Robust to Irrelevant Context](#). *Preprint*, arXiv:2310.01558.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2024. [Chain-of-Note: Enhancing Robustness in Retrieval-Augmented Language Models](#). *Preprint*, arXiv:2311.09210.

Tianjun Zhang, Shishir G. Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. 2024. [RAFT: Adapting Language Model to Domain Specific RAG](#). *Preprint*, arXiv:2403.10131.

Zexuan Zhong, Ziqing Huang, Alexander Wettig, and Danqi Chen. 2023. [Poisoning Retrieval Corpora by Injecting Adversarial Passages](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13764–13775, Singapore. Association for Computational Linguistics.

Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. 2024. [PoisonedRAG: Knowledge Corruption Attacks to Retrieval-Augmented Generation of Large Language Models](#). *Preprint*, arXiv:2402.07867.

A Validation of Simulated Distracting Documents

A.1 Ranking Distribution of Simulated Distractions

To ensure that our simulated distracting documents—Conflict, Adversarial, and Negated—pose a realistic challenge to RALMs, we inserted these simulated passages into the original pool of 20 retrieved documents and performed reranking using BAAI/bge-reranker-large⁴. If these simulated passages are ranked high, it suggests that they are semantically similar to naturally retrieved documents and can therefore function as realistic distractions.

As summarized in Table 3, the simulated distractors are consistently ranked high across both general and expert-domain datasets. In particular, Negated and Conflict documents frequently appear near the top of the rankings, often within the top 10%–30% of the retrieved set (e.g., Mean Scaled Rank values of 0.080 and 0.101 in MedQA, respectively). Although Adversarial passages sometimes rank slightly lower (e.g., 0.854 in WebQ) because

the intentionally misleading entities slightly alter their dense representations, they still remain an important target of our framework.

Overall, all types of distractions appear alongside genuine passages in both general and expert domains. This result shows that standard semantic retrievers have difficulty distinguishing logical inconsistencies from superficial semantic similarity, highlighting the need for our logic-based Sieve module for post-retrieval filtering.

A.2 Probing Distraction Complexity: From Pairwise Clarity to Multi-document Interference

To validate the realism and logical integrity of our simulated distracting documents, we conducted a two-stage probing experiment.

Pairwise Logical Verification First, we presented a state-of-the-art LLM (GPT-4o) with a pair of documents: the gold document and one additional document, which could be either a simulated distractor (e.g., negated, conflicting, or adversarial) or a neutral document. When asked to categorize the relationship between the two documents, the model achieved a near 100% accuracy.

Multi-document Distraction Effect However, Table 17 shows that the classification accuracy dropped significantly when we increased the complexity by providing the full set of five documents, consistent with the setup used in our main experiments.

This discrepancy indicates that our simulated distractors are not only logically well-constructed but also realistic, as they successfully degrade the overall performance of RALMs, accurately mirroring the exact challenges of real-world noisy retrieval.

A.3 Effect of Distractors on Answerable Instances (Apple-to-Apple Comparison)

To validate the impact of our simulated distractors, we perform an apple-to-apple comparison that measures how individual distracting documents affect QA performance. Table 4 reports the QA accuracy of a vanilla LLM when the retrieval context is modified from an answerable state to include specific types of noise.

The results show that introducing distractors consistently degrades QA accuracy. In particular, the unanswerable, conflict, and adversarial conditions lead to clear performance drops across datasets

⁴<https://huggingface.co/BAAI/bge-reranker-large>

Datasets	General domain				Expert domain		
	NQ-google	NQ-wiki	WebQ	RealtimeQA	MedQA	AstroQA	BioASQ
# Passage	10 ~33	8	8	27 ~53	24	24	4 ~36
Gold Passage	-	-	-	-	3.714	6.350	-
Conflict Passage	4.560	3.028	3.364	2.970	2.227	3.560	7.699
Adversarial Passage	19.520	5.700	6.654	26.490	3.548	9.610	7.758
Negated Passage	5.400	2.659	2.705	17.160	1.813	2.710	2.415

(a) Mean Raw Rank

Datasets	General domain				Expert domain		
	NQ-google	NQ-wiki	WebQ	RealtimeQA	MedQA	AstroQA	BioASQ
# Passage	10 ~33	8	8	27 ~53	24	24	4 ~36
Gold Passage	-	-	-	-	0.118	0.233	-
Conflict Passage	0.121	0.290	0.338	0.041	0.101	0.111	0.665
Adversarial Passage	0.598	0.709	0.854	0.533	0.318	0.374	0.643
Negated Passage	0.145	0.248	0.258	0.335	0.080	0.075	0.139

(b) Mean Scaled Rank

Table 3: **Ranking results of simulated distracting documents.** We evaluate the visibility of simulated noise—Conflict, Adversarial, and Negated passages—by mixing them with originally retrieved documents and reranking the set. The high rankings of these distractions demonstrate that our simulated noise is semantically plausible and poses a realistic challenge to RALMs.

dataset	# examples	Acc (ans)	Acc (unans)	Acc (conflict)	Acc (negated)	Acc (adversarial)
NQ-google	30	23.33	6.67 -16.66	20.00 -3.33	43.33 +20.00	13.33 -10.00
NQ-wiki	327	32.42	0.00 -32.42	24.46 -7.96	54.74 +22.32	29.97 -2.45
WebQ	264	24.24	0.76 -23.48	20.45 -3.79	46.21 +21.97	22.35 -1.89
RealtimeQA	25	24.00	4.00 -20.00	8.00 -16.00	48.00 +24.00	16.00 -8.00
MedQA	57	35.09	14.04 -21.05	22.81 -12.28	49.12 +14.03	24.56 -10.53

Table 4: **QA accuracy of a vanilla LLM under different retrieval conditions.** Each column reports accuracy with a specific type of retrieval state, all derived from the same answerable retrieval context by modifying one document.

compared to the baseline answerable state. For example, on MedQA, the accuracy decreases from 35.09% to 22.81% and 24.56% under the conflict and adversarial conditions, respectively. This degradation indicates that the simulated distractors effectively mislead the generator.

Interestingly, the negated condition often yields higher accuracy (e.g., 54.74% on NQ-wiki). This behavior suggests a weakness of monolithic LLMs: they tend to rely on lexical overlap and parametric knowledge rather than careful contextual reasoning. Because negated documents share nearly identical entities and structures with the gold evidence—differing only by logical operators—the model may still produce the correct entity despite the contradic-

tion. This failure mode highlights the importance of our Sieve module, which explicitly detects logical negations that standard LLMs often overlook.

B Analysis of Data-Efficient Domain Adaptation for the Sieve Module

Unlike the Sage module, which typically requires large-scale fine-tuning to acquire domain-specific knowledge, the Sieve is designed to learn the structural characteristics of target documents through minimal fine-tuning. To examine this structural adaptation, we conduct the following two experiments. Because the Sieve is a distractor detector, unanswerable sets should pass this stage and be handled by the Sage. We therefore group an-

test dataset	classifier ft. on	respond @ans	respond @unans	abstain @confl	abstain @neg	abstain @adv
MedQA	NQ-google	85.34	94.97	50.00	63.19	62.69
	+MedQA	96.00	92.70	75.76	88.57	63.04
	+AstroQA	92.24	97.48	80.00	100.00	100.00
	+BioASQ	95.69	91.82	75.71	92.47	95.65

Table 5: Conflict classification performance on the MedQA dataset using classifiers fine-tuned on various domain datasets.

swerable and unanswerable instances into the same non-distracted class and report binary classification metrics (Respond F1 and Abstain F1) instead of our standard tripartite evaluation.

Data Efficiency of Structural Adaptation. First, we vary the number of domain-specific training samples to evaluate fine-tuning efficiency. As shown in Figure 6, the largest gains generally occur within the first 100–200 domain-specific examples, after which performance tends to plateau.

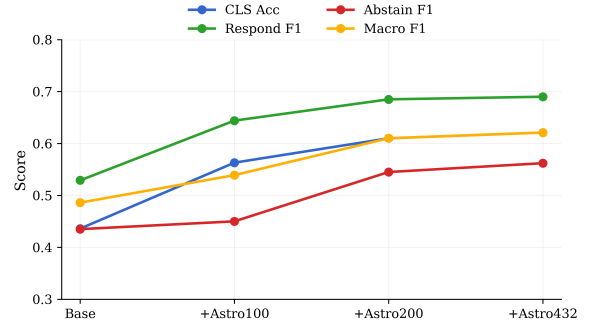
Cross-Domain Transferability Next, we evaluate transferability across expert domains by applying a Sieve trained on other expert domains (BioASQ and AstroQA) to MedQA. Because all these datasets use scientific abstracts as retrieved documents, they share similar structural and stylistic properties. As shown in Table 5, fine-tuning on either BioASQ or AstroQA substantially improves distractor detection on MedQA, achieving performance comparable to a model trained directly on MedQA. This result confirms that the Sieve learns domain-agnostic structural patterns rather than domain-specific semantics.

C Technical Specifications

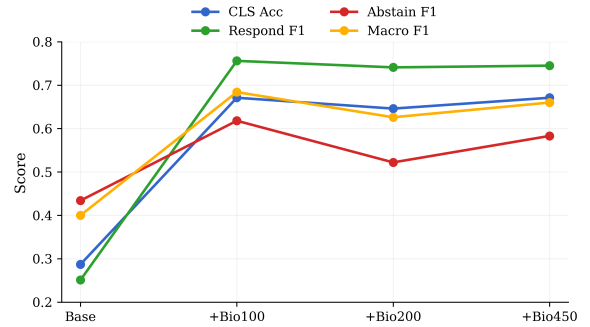
This section provides the detailed experimental configurations, including hyperparameter settings and hardware environments, to ensure the reproducibility of our results.

Training Configurations We employed distinct training setups for the Sage (LLM) and the Sieve (Classifier) to optimize their respective tasks. The specific hyperparameters are summarized as follows:

- The Sage (LLM): We fine-tuned the Llama-3-8B-Instruct model on a randomly sampled subset of NQ-wiki dataset. The training corpus consisted of 8,000 answerable (S_{ans}) and 800 unanswerable (S_{unans}) examples, maintaining a 10:1 ratio. To



(a) AstroQA



(b) BioASQ

Figure 6: **Performance improvement when fine-tuning the NQ-google trained conflict classifier with 100/200/400+ domain samples.** Even a small number of domain-specific labels yields substantial gains across both expert domains.

ensure parameter efficiency, we utilized Low-Rank Adaptation (LoRA)(Hu et al., 2021) with rank = 8 and $\alpha = 16$. The model was trained for 3 epochs using the HuggingFace SFTTrainer library with the standard language modeling loss. We employed the paged_adamw_32bit optimizer with a learning rate of 2×10^{-5} and a 0.05 warm-up ratio. The final model was selected based on the validation loss.

- The Sieve (Classifier): We fine-tuned Longformer-base-4096 on the NQ-google dataset. The training set comprised 418 answerable (S_{ans}) and 396 conflict (S_{confl}) examples, providing a nearly balanced distribution for conflict detection. The model was trained for up to 15 epochs to ensure stable convergence. We used a learning rate of 2×10^{-4} with a 0.05 warm-up ratio. To prevent overfitting, we applied early stopping and selected the optimal checkpoint based on validation loss and classification accuracy.

Dataset	Retrieved Set	System Acc.	Macro-F1
NQ-google	Top-5 (default)	0.796	0.731
	+ 1 neutral	0.778	0.711
	+ 2 neutrals	0.792	0.739
	+ 3 neutrals	0.783	0.726
RealtimeQA	Top-5 (default)	0.802	0.734
	+ 1 neutral	0.786	0.712
	+ 2 neutrals	0.802	0.740
	+ 3 neutrals	0.791	0.738

Table 6: Effect of adding neutral documents to the default top-5 retrieved document sets. Performance remains stable as the number of neutral documents increases, with no systematic degradation in either System Accuracy or Macro-F1.

Hardware and Reproducibility Details To provide a consistent benchmark for computational efficiency, all training and inference procedures were conducted on a single NVIDIA Pro6000 GPU (96GB VRAM).

We utilized the bfloat16 (BF16) data type as the default for all experiments to balance numerical stability and memory efficiency. While the exact duration varied slightly across datasets, each training session was completed within 6 hours, and all inference sessions were finished within 1 hour.

For a fair and rigorous comparison against baselines, we intentionally refrained from using high-throughput inference engines such as vLLM or other model quantization techniques that could mask the raw latency of the underlying architectures.

To ensure the reliability of our findings, all experiments were reproduced at least twice using different random seeds. The results reported in the main text represent the averaged values of these independent runs.

D Effect of Neutral Documents

In Section 3.1, we assume that neutral documents d_{neutral} have a negligible effect on the retrieval state. To empirically validate this assumption, we conduct an ablation by incrementally adding neutral documents to the retrieved document set while keeping the original top-5 documents fixed.

Specifically, we evaluate Sieve and Sage on NQ-google and RealtimeQA after adding one to three neutral documents to each top-5 retrieved set. A neutral document may contain lexical overlap with the answer but does not provide supporting or distracting evidence according to our document-level criteria.

Dataset	K	System Acc.	Macro-F1
NQ-google	3	0.663	0.643
	5 (default)	0.796	0.731
	7	0.668	0.612
RealtimeQA	3	0.684	0.646
	5 (default)	0.802	0.734
	7	0.682	0.633

Table 7: Sensitivity to the number of retrieved documents K . Both Sieve and Sage are trained with $K = 5$.

Component	# Params	VRAM (GB)	TFLOPs
Sieve	149M	0.56	15.01
Sage	8B	15.91	217.52

Table 8: Computational footprint of the Sieve and Sage.

As shown in Table 6, adding up to three neutral documents results in only minor performance fluctuations. Across both datasets, the maximum absolute change from the top-5 baseline is 0.018 in System Accuracy and 0.022 in Macro-F1, with no systematic decrease as more neutral documents are added. These results support our assumption that neutral documents have limited influence on the set-level retrieval state.

E Sensitivity to the Number of Retrieved Documents

We analyze the sensitivity of Sieve and Sage to the number of retrieved documents K . Both components are trained with $K = 5$ by default, and we evaluate the same trained models under $K \in \{3, 5, 7\}$.

As shown in Table 7, $K = 5$ achieves the best performance on both datasets, while performance decreases when the retrieval depth differs from the training setting. This suggests that part of the degradation under $K = 3$ or $K = 7$ may arise from the fixed- K training configuration. Training with variable retrieval depths may improve robustness to changes in K , which we leave for future work.

F Computational Footprint of Sieve and Sage

To quantify the computational gap between the two components, we compare the parameter count, VRAM usage, and inference FLOPs of the Sieve and Sage.

As shown in Table 8, the Sieve requires approximately $54\times$ fewer parameters, $28\times$ less VRAM, and $14\times$ fewer inference FLOPs than the Sage.

Dataset	Configuration	System Acc.	Macro-F1
NQ-google	Vanilla	0.102	0.179
	+ Sage ft.	0.664	0.487
	+ Sieve ft.	0.769	0.695
	+ both ft.	0.796	0.731
RealtimeQA	Vanilla	0.225	0.238
	+ Sage ft.	0.535	0.314
	+ Sieve ft.	0.791	0.728
	+ both ft.	0.802	0.734

Table 9: Ablation on Sieve and Sage. Both components independently improve performance, while their combination achieves the strongest overall results.

This large computational gap enables the Sieve to serve as a low-cost filtering stage before expensive LLM inference, reducing the amortized cost of the overall pipeline by avoiding unnecessary Sage invocations on distracted inputs.

G Component-wise Ablation

To isolate the contribution of each component, we compare the vanilla model with variants applying Sage fine-tuning only, Sieve fine-tuning only, and both components together.

As shown in Table 9, both components independently improve performance over the vanilla baseline, while their combination yields the strongest overall results.

H Sensitivity to the Sieve Decision Threshold

We analyze the sensitivity of the Sieve to its decision threshold τ_{distract} on AstroQA. Here, Macro-F1 denotes the binary classification Macro-F1 of the Sieve over conflicting and non-conflicting document sets.

As shown in Table 10, performance remains stable over the practical threshold range of [0.3, 0.7], with Macro-F1 varying between 0.933 and 0.947. At the default threshold of 0.5, the Sieve achieves the highest Macro-F1 of 0.947 while maintaining a low false-positive rate of 0.037. Performance degrades substantially at the two extreme thresholds, indicating that the observed gains do not result from trivially classifying all inputs as conflicting or non-conflicting.

I Prompt template

To facilitate the reproducibility of our experiments, we are releasing all the prompts (Table 12, Table 13, Table 14) used in our study. The curly brackets

Threshold	Macro-F1	TPR	TNR	FPR	FNR
0.00 (all conflict)	0.264	1.000	0.000	1.000	0.000
0.10	0.923	0.961	0.908	0.091	0.038
0.30	0.933	0.937	0.940	0.059	0.067
0.50 (default)	0.947	0.937	0.962	0.037	0.067
0.70	0.939	0.915	0.962	0.037	0.086
0.90	0.946	0.892	0.983	0.016	0.105
1.00 (all answerable)	0.390	0.000	1.000	0.000	1.000

Table 10: Sensitivity of the Sieve to the conflict-detection threshold on AstroQA. TPR and TNR denote the true-positive and true-negative rates, while FPR and FNR denote the corresponding false-positive and false-negative rates.

Dataset	Sieve Backbone	System Acc.	Macro-F1
NQ-google	Longformer-base-4096	0.796	0.731
	DeBERTa-v3-base	0.765	0.711
	RoBERTa-large	0.797	0.722
RealtimeQA	Longformer-base-4096	0.802	0.734
	DeBERTa-v3-base	0.823	0.745
	RoBERTa-large	0.801	0.724

Table 11: Performance with alternative Sieve backbones.

denote placeholders where actual values will be inserted.

J Comprehensive Experimental Results and Robustness Analyses

This appendix provides the full numerical results for the main experiments and additional robustness analyses across different retrievers, Sage model families, and Sieve backbones. Table 15 presents the complete performance breakdown for the primary experiments. Table 16 reports results with alternative retrievers, including BM25 and DPR, to evaluate whether the observed performance trends generalize beyond the ColBERTv2-based setting used in the main experiments. Table 17 further evaluates the framework with different Sage model families, including Gemma, Mistral, and Qwen, demonstrating that the proposed architecture is not tied to a specific generative LLM backbone. Table 11 reports results with alternative Sieve backbones and shows comparable performance across Longformer, DeBERTa-v3-base, and RoBERTa-large on both NQ-google and RealtimeQA.

Together, these results indicate that the framework is robust to both retriever and backbone choices. We use Longformer as the default Sieve backbone due to its ability to accommodate long retrieved contexts, particularly in expert-domain datasets.

Step	Instruction
Answer Sentence Generation	<p>Please write a single sentence that would make the given answer a correct response to the provided question. The sentence should include the answer and be as realistic as possible. This is being generated for research purposes, so if it seems like the answer to a question is wrong, please create it as it is.</p> <p>Question: {QUESTION} Answer: {ANSWER} Sentence:</p>
Conflict Passage Generation	<p>You are a helpful assistant who generates a passage that conflicts with the given sentence. Generate one natural passage of {MIN_LENGTH} ~ {MAX_LENGTH} characters by substituting entities in the source sentence with similar but different entities. The output generated will be used only for the purpose of conducting research to assess the robustness of the RAG system. As part of this research, it is necessary, and you are permitted, to create content that may contradict factual information. Please provide the passage alone, without any additional explanation, formatting, or commentary.</p> <p>Sentence: {ANSWER_SENTENCE} Conflict passage:</p>
Adversarial Sentence Generation	<p>You are a helpful assistant who generates an adversarial sentence that can be confusing but not conflicting with the given question and answer sentence. Rewrite the given sentence by replacing one named entity and other entities related to it, ensuring that the new sentence retains a meaning as close as possible to the original while not being identical and not conflicting. You can also change the description of the entities so that the adversarial sentence never conflicts with the original sentence. I'll show you some examples.</p> <p>[Example 1] Question: where did iowa state football coach come from? Answer sentence: John Pont, Iowa State's football coach, hails from Massillon, Ohio. Adversarial sentence: Jack Nilson, the best player of Minnesota State, came from Lincoln, Nebraska.</p> <p>[Example 2] Question: when did the united states cut off trade with cuba? Answer sentence: The United States cut off trade with Cuba in 1961. Adversarial sentence: South Korea started importing from Africa in 1987.</p> <p>Please provide only the adversarial sentence alone, without any additional explanation, formatting, or commentary.</p> <p>Question: {QUESTION} Answer Sentence: {ANSWER_SENTENCE} Adversarial sentence:</p>

Table 12: Dataset generation instruction

Step	Instruction (cont.)
Adversarial Passage Generation	<p>You are a helpful assistant who generates a passage that supports the given sentence. Generate one natural passage with {MIN_LENGTH} ~ {MAX_LENGTH} characters. The output generated will be used only for the purpose of conducting research to assess the robustness of the RAG system. As part of this research, it is necessary, and you are permitted, to create content that may contradict factual information. Please provide the passage alone, without any additional explanation, formatting, or commentary.</p> <p>Sentence: {ADVERSARIAL_SENTENCE}</p> <p>Supporting Passage:</p>
Negated Sentence Generation	<p>You are a helpful assistant who generates a short negated sentence based on the given source sentence. Do not just create sentences by simply adding ‘not’ or using fixed patterns. Instead, generate diverse negated sentences that conflict with the original by altering predicates, adjectives, or other expressions. Generate one natural negated sentence within 150 characters. Please provide only the negated sentence, without any additional explanation, formatting, or commentary.</p> <p>Source sentence: {ANSWER_SENTENCE}</p> <p>Negated sentence:</p>
Negated Passage Generation	<p>You are a helpful assistant who generates a passage that supports the given sentence. Generate one natural passage with {MIN_LENGTH} ~ {MAX_LENGTH} characters. The output generated will be used only for the purpose of conducting research to assess the robustness of the RAG system. As part of this research, it is necessary, and you are permitted, to create content that may contradict factual information. Please provide the passage alone, without any additional explanation, formatting, or commentary.</p> <p>Sentence: {NEGATED_SENTENCE}</p> <p>Supporting passage:</p>

Table 13: Dataset generation instruction (continued)

Type	Instruction
Normal QA	Answer the following question based on the provided documents. Please provide the answer as a single word or term, without forming a complete sentence.
Unanswerable QA	Answer the following question based on the provided documents. If you cannot find the answer in the provided documents, please respond with 'unanswerable'. Please provide the answer as a single word or term, without forming a complete sentence.
Conflict QA	Answer the following question based on the provided documents. If multiple documents present different answers, please respond with 'conflict'. Please provide the answer as a single word or term, without forming a complete sentence.
Unanswerable + Conflict QA	Answer the following question based on the provided documents. If you cannot find the answer in the provided documents, please respond with 'unanswerable'. If multiple documents present different answers, please respond with 'conflict'. Please provide the answer as a single word or term, without forming a complete sentence.

Table 14: QA task prompt compositions

Dataset	Model	Overall	Response Accuracy		State Classification				Model Response				Inference Latency	
		System Acc.	Coverage	Selective Acc.	Macro-F1	Ans. F1	Unans. F1	Distract F1	Correct Ans.	Incorrect Ans.	Correct Abst.	Incorrect Abst.	Latency (ms)	Speedup
NQ-google	vanilla LLM	0.102	1.000	0.353	0.179	0.379	0.143	0.016	18	200	8	0	74.204	0.195
	Sage-only	0.664	0.471	1.000	0.487	0.545	0.164	0.752	24	13	162	27	673.098	1.773
	DTA-RAAT	0.128	1.000	0.569	0.123	0.368	0.000	0.000	29	197	0	0	328.374	0.865
	DTA-chatQA	0.230	0.961	0.857	0.221	0.377	0.254	0.031	42	167	15	2	756.341	1.993
	(ours) Sieve and Sage	0.796	0.863	0.886	0.731	0.672	0.585	0.937	39	41	139	7	379.574	1.000
NQ-wiki	vanilla LLM	0.154	0.993	0.465	0.241	0.439	0.277	0.008	411	2721	159	6	88.822	0.856
	Sage-only	0.792	0.938	0.772	0.730	0.777	0.454	0.960	644	613	1985	55	144.024	1.388
	DTA-RAAT	0.167	1.000	0.619	0.142	0.425	0.000	0.000	550	2747	0	0	122.625	1.181
	DTA-chatQA	0.226	0.922	0.713	0.267	0.429	0.355	0.016	585	2348	295	69	138.545	1.335
	(ours) Sieve and Sage	0.638	0.810	0.822	0.632	0.603	0.556	0.737	592	909	1627	169	<u>103.794</u>	1.000
WebQ	vanilla LLM	0.110	0.976	0.365	0.198	0.418	0.171	0.005	194	1805	84	13	95.664	0.781
	Sage-only	0.746	0.927	0.699	0.667	0.737	0.322	0.943	353	473	1230	40	166.580	1.361
	DTA-RAAT	0.153	1.000	0.587	0.138	0.413	0.000	0.000	320	1776	0	0	134.562	1.099
	DTA-chatQA	0.186	0.798	0.664	0.219	0.403	0.241	0.014	289	1324	373	110	170.475	1.392
	(ours) Sieve and Sage	<u>0.578</u>	0.758	0.719	<u>0.562</u>	<u>0.555</u>	0.401	<u>0.730</u>	297	645	1022	132	<u>122.435</u>	1.000
RealtimeQA	vanilla LLM	0.225	1.000	0.765	0.238	0.337	0.376	0.000	26	142	19	0	78.921	0.148
	Sage-only	0.535	0.059	1.000	0.314	0.095	0.189	0.659	2	6	147	32	869.669	1.631
	DTA-RAAT	0.134	1.000	0.735	0.103	0.308	0.000	0.000	25	162	0	0	<u>384.136</u>	0.720
	DTA-chatQA	0.262	0.971	0.909	0.253	0.333	0.427	0.000	30	134	22	1	897.628	1.683
	(ours) Sieve and Sage	0.802	0.676	1.000	0.734	0.541	0.763	0.899	23	28	125	11	533.283	1.000
RAG-Bench	vanilla LLM	0.125	0.980	0.497	0.197	0.370	0.218	0.002	444	3481	109	18	111.257	1.009
	Sage-only	0.397	0.928	0.779	0.415	0.446	0.354	0.446	659	2220	1107	66	124.208	1.127
	DTA-RAAT	0.179	1.000	0.796	0.122	0.367	0.000	0.000	726	3326	0	0	101.251	0.918
	DTA-chatQA	0.191	0.908	0.775	0.215	0.370	0.265	0.012	642	2922	404	84	112.768	1.023
	(ours) Sieve and Sage	0.368	0.773	0.814	0.375	0.405	0.376	0.344	574	1995	1276	207	<u>110.259</u>	1.000
MedQA	vanilla LLM	0.322	0.931	0.435	0.412	0.444	0.793	0.000	47	323	113	8	168.602	0.522
	Sage-only	0.489	0.914	0.425	0.548	0.613	0.268	0.765	45	185	251	10	509.038	1.576
	DTA-RAAT	0.041	1.000	0.172	0.127	0.382	0.000	0.000	20	471	0	0	304.529	0.943
	DTA-chatQA	0.297	0.914	0.509	0.367	0.434	0.667	0.000	54	319	108	10	502.077	1.555
	(ours) Sieve and Sage	0.682	0.862	0.460	0.766	0.662	0.677	0.959	46	140	289	16	322.954	1.000
BioASQ	vanilla LLM	0.333	0.973	0.000	0.326	0.308	0.669	0.000	0	399	209	2	138.037	0.442
	Sage-only	0.428	0.880	0.000	0.529	0.365	0.604	0.617	0	287	314	9	389.105	1.246
	DTA-RAAT	0.002	1.000	0.013	0.073	0.219	0.000	0.000	1	609	0	0	<u>216.414</u>	0.693
	DTA-chatQA	0.456	0.707	0.038	0.362	0.305	0.754	0.027	2	270	316	22	363.750	1.165
	(ours) Sieve and Sage	<u>0.428</u>	0.853	0.000	0.552	<u>0.336</u>	0.551	0.768	0	306	293	11	312.165	1.000
AstroQA	vanilla LLM	0.218	0.828	0.268	0.347	0.516	0.526	0.000	22	197	53	17	779.595	1.719
	Sage-only	0.550	0.434	0.465	0.620	0.509	0.608	0.744	20	50	163	56	661.126	1.458
	DTA-RAAT	0.093	1.000	0.273	0.170	0.510	0.000	0.000	27	262	0	0	393.698	0.868
	DTA-chatQA	0.208	0.798	0.392	0.296	0.485	0.384	0.019	31	196	42	20	693.382	1.529
	(ours) Sieve and Sage	0.630	0.303	0.433	0.662	0.420	0.634	0.933	13	31	176	69	<u>453.511</u>	1.000

Table 15: **Full results for the main experiments.** System Accuracy follows the ternary system-level metric used in the main text. Coverage is the fraction of answerable examples for which the model responds, and Selective Accuracy is the accuracy conditioned on those responses. Response counts are reported over all retrieval states. **Bold** and underlined values denote the best and second-best results within each dataset block; for latency, lower is better. Speedup and response counts are not ranked.

Retriever	Dataset	Model	System Acc.	Coverage	Selective Acc.	Macro-F1	Ans. F1	Unans. F1	Distract F1	Latency (ms)	Speedup
BM25	NQ-wiki	vanilla LLM	0.499	0.913	0.450	0.360	0.294	0.780	0.007	245.834	0.857
		Sage-only	<u>0.766</u>	<u>0.876</u>	<u>0.726</u>	0.739	0.478	<u>0.843</u>	0.897	355.948	1.241
		(ours) Sieve and Sage	0.796	0.676	0.801	<u>0.684</u>	<u>0.460</u>	0.898	<u>0.695</u>	<u>286.818</u>	1.000
	MedQA	vanilla LLM	0.189	0.950	0.435	0.281	0.400	<u>0.444</u>	0.000	682.373	1.945
		Sage-only	<u>0.567</u>	<u>0.835</u>	<u>0.446</u>	<u>0.602</u>	0.558	0.335	<u>0.911</u>	<u>463.837</u>	1.322
		(ours) Sieve and Sage	0.661	0.645	0.538	0.685	<u>0.544</u>	0.559	0.954	350.884	1.000
DPR	NQ-wiki	vanilla LLM	0.204	0.970	0.486	0.342	0.449	0.566	0.011	<u>217.119</u>	1.188
		Sage-only	0.812	<u>0.925</u>	<u>0.770</u>	0.779	0.799	<u>0.572</u>	0.967	248.255	1.359
		(ours) Sieve and Sage	<u>0.641</u>	0.761	0.806	<u>0.641</u>	<u>0.601</u>	0.606	<u>0.717</u>	182.687	1.000
	WebQ	vanilla LLM	0.145	0.961	0.367	0.275	0.414	0.397	0.015	259.668	1.389
		Sage-only	0.737	<u>0.913</u>	<u>0.672</u>	0.707	0.712	<u>0.464</u>	0.944	264.534	1.415
		(ours) Sieve and Sage	<u>0.585</u>	0.716	0.706	<u>0.591</u>	<u>0.540</u>	0.516	<u>0.715</u>	186.999	1.000
	MedQA	vanilla LLM	0.188	0.957	0.491	0.254	0.387	<u>0.376</u>	0.000	649.787	3.267
		Sage-only	<u>0.588</u>	<u>0.896</u>	0.563	<u>0.595</u>	<u>0.561</u>	0.250	0.975	<u>530.878</u>	2.669
		(ours) Sieve and Sage	0.656	0.809	<u>0.538</u>	0.701	0.600	0.533	<u>0.970</u>	198.912	1.000

Table 16: **Generalization across alternative retrievers.** We report results for BM25 (sparse) and DPR (dense), in addition to the ColBERTv2 setting used in the main experiments. Only runs that passed source-provenance validation are included. System Accuracy follows the ternary system-level metric used in the main text. **Bold** and underlined values denote the best and second-best results within each retriever–dataset block; for latency, lower is better. Speedup is not ranked.

Sage Family	Dataset	Model	System Acc.	Coverage	Selective Acc.	Macro-F1	Ans. F1	Unans. F1	Distract F1	Latency (ms)	Speedup
Gemma	NQ-wiki	vanilla LLM	0.204	0.952	0.560	0.299	0.443	0.441	0.013	409.329	3.483
		Sage-only	0.843	0.739	0.795	0.747	0.755	0.545	0.941	<u>193.401</u>	1.645
		(ours) Sieve and Sage	<u>0.600</u>	<u>0.854</u>	<u>0.763</u>	<u>0.608</u>	<u>0.591</u>	<u>0.496</u>	<u>0.737</u>	117.539	1.000
	WebQ	vanilla LLM	0.134	0.905	0.408	0.228	0.412	0.249	0.022	260.233	2.172
		Sage-only	0.792	0.670	0.674	0.678	0.681	0.399	0.953	<u>203.233</u>	1.696
		(ours) Sieve and Sage	<u>0.546</u>	<u>0.842</u>	<u>0.662</u>	<u>0.561</u>	<u>0.567</u>	<u>0.386</u>	<u>0.730</u>	119.838	1.000
	MedQA	vanilla LLM	0.369	0.888	0.417	0.444	<u>0.461</u>	0.862	0.009	815.216	4.044
		Sage-only	0.688	0.328	0.526	<u>0.587</u>	0.455	0.561	0.744	<u>692.948</u>	3.437
		(ours) Sieve and Sage	<u>0.615</u>	<u>0.534</u>	<u>0.468</u>	0.662	0.490	<u>0.813</u>	<u>0.682</u>	201.598	1.000
Mistral	NQ-wiki	vanilla LLM	0.222	0.953	0.625	0.292	0.445	<u>0.383</u>	0.047	286.439	2.755
		Sage-only	0.763	<u>0.927</u>	0.775	0.681	<u>0.737</u>	0.416	0.891	<u>141.277</u>	1.359
		(ours) Sieve and Sage	<u>0.596</u>	0.901	<u>0.773</u>	<u>0.601</u>	0.789	0.276	<u>0.737</u>	103.985	1.000
	WebQ	vanilla LLM	0.173	0.950	0.556	0.230	0.421	<u>0.239</u>	0.031	162.015	1.742
		Sage-only	0.741	0.897	<u>0.714</u>	0.643	<u>0.708</u>	0.329	0.891	<u>134.522</u>	1.447
		(ours) Sieve and Sage	<u>0.554</u>	0.866	0.718	<u>0.560</u>	0.747	0.204	<u>0.730</u>	92.979	1.000
	MedQA	vanilla LLM	0.360	0.845	0.439	0.438	<u>0.464</u>	0.722	0.127	436.971	2.093
		Sage-only	0.615	<u>0.397</u>	<u>0.370</u>	<u>0.592</u>	0.460	0.657	<u>0.658</u>	<u>367.668</u>	1.761
		(ours) Sieve and Sage	<u>0.585</u>	0.379	0.364	0.615	0.497	<u>0.667</u>	0.682	208.820	1.000
Qwen	NQ-wiki	vanilla LLM	0.235	<u>0.947</u>	0.689	0.282	0.440	<u>0.334</u>	0.071	234.099	2.670
		Sage-only	0.806	0.969	0.739	0.668	0.810	0.265	0.930	<u>107.693</u>	1.228
		(ours) Sieve and Sage	<u>0.579</u>	0.946	0.751	<u>0.581</u>	<u>0.596</u>	0.409	<u>0.737</u>	87.689	1.000
	WebQ	vanilla LLM	0.203	0.905	0.574	0.270	0.413	<u>0.278</u>	0.119	101.623	0.981
		Sage-only	0.787	0.921	<u>0.683</u>	0.648	0.776	0.225	0.943	<u>108.272</u>	1.045
		(ours) Sieve and Sage	<u>0.546</u>	<u>0.910</u>	0.704	<u>0.534</u>	<u>0.566</u>	0.306	<u>0.729</u>	<u>103.631</u>	1.000
	MedQA	vanilla LLM	0.336	0.836	0.443	0.415	<u>0.463</u>	0.758	0.024	201.751	1.443
		Sage-only	0.711	0.517	<u>0.483</u>	0.733	0.574	0.688	0.936	<u>176.258</u>	1.261
		(ours) Sieve and Sage	<u>0.544</u>	<u>0.578</u>	0.493	<u>0.605</u>	0.442	<u>0.691</u>	<u>0.682</u>	139.802	1.000

Table 17: **Full results across diverse Sage model families.** We use google/gemma-2-9b-it, mistralai/Mistral-7B-Instruct-v0.3, and Qwen/Qwen2.5-7B-Instruct for Gemma, Mistral, and Qwen, respectively. System Accuracy follows the ternary system-level metric used in the main text. **Bold** and underlined values denote the best and second-best results within each model-family–dataset block; for latency, lower is better. Speedup is not ranked.