TriageRAG: Failure-Aware Self-Training for Adapting Retrieval-Augmented Language Models to Specialized Domains

Anonymous authors
Paper under double-blind review

Abstract

Retrieval-augmented generation (RAG) lets large language models (LLMs) answer knowledge-intensive questions from retrieved evidence, but adapting a general-purpose RAG system to a specialized domain such as medicine is difficult when only an unlabeled corpus is available. Self-training approaches such as SimRAG address this by letting the LLM write its own question–answer pairs from domain passages and fine-tuning on the pairs that survive a round-trip consistency filter. We observe that these approaches treat every retained example identically: they never ask whether the student already knows the answer, already extracts it correctly from the retrieved context, or fails—and if so, whether the failure lies with the reader or with the retriever. To tackle this, we propose TriageRAG, a failure-aware self-training approach that probes the student on each self-generated question with and without retrieved context, triages it into one of four categories (known, solved, context-failure, retrieval-failure), and routes each category to the training signal that addresses it, including a parametric route that teaches the reader to answer when retrieval misses. Probing is cheap and repeated every round, so the training mixture follows the student, and the triage histogram doubles as a label-free diagnostic of where the domain gap lies. Experiments on 11 datasets spanning three domains with a Qwen3-4B backbone, and on CS-Bench with Qwen3-8B, demonstrate that TriageRAG outperforms a budget-matched SimRAG by 1.2–6.4 points and is the only method that recovers the closed-book level in every case where retrieval hurts the base model. A single ablation identifies the mechanism: round-trip filtering removes every training context that lacks the evidence, and a reader that never sees such contexts learns to over-trust noisy retrieval.

1 Introduction

Retrieval-augmented generation (RAG) Lewis et al. (2020); Gao et al. (2023); Asai et al. (2024) grounds the answers of large language models (LLMs) in retrieved evidence, which adapts them to long-tail knowledge without retraining Ovadia et al. (2024) and reduces hallucination Shuster et al. (2021). Adapting a general-purpose RAG system to a specialized domain such as medicine is nevertheless hard: reader and retriever both face a distribution shift, and labeled domain QA data is scarce or private, so fine-tuning has to rely on synthetic data. SimRAG Xu et al. (2025) is a strong recent recipe. It fine-tunes an instruction-tuned LLM on general retrieval-oriented data, prompts the same model to write short-span, multiple-choice and claim-verification questions from unlabeled domain passages, keeps the pairs whose source passage is retrieved in the top-k for the generated question (round-trip consistency), and fine-tunes on the survivors with retrieved contexts.

We start from an observation about what such a pipeline does not look at: every retained pseudo-labeled pair receives the same treatment. Nothing in the loop asks whether the student already knows the answer without context, whether it already extracts it correctly from the retrieved passages, or whether it fails—and if so, whether the failure belongs to the reader (the evidence is in the context but the answer is wrong) or to the retriever (the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

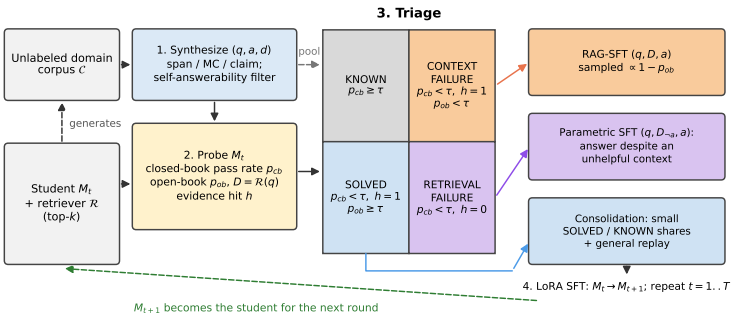


Figure 1: Overview of TriageRAG. Each round, the student synthesizes (q, a, d) triples from the unlabeled corpus (span, multiple-choice and claim formats; only self-answerable pairs are kept), is probed closed-book (p_{cb}) and open-book on the top-k retrieved passages (p_{ob}, evidence hit h), and each pair is triaged by the conditions shown and routed to a training signal; the routed mixture is used for LoRA fine-tuning and the updated student re-probes the pool in the next round. The retriever is frozen in our main configuration (Section 3.4).

evidence never reached the context). These problems have different remedies: contextual fine-tuning teaches a reader to use evidence that is present, but it cannot help when the evidence is absent. Worse, the round-trip filter removes exactly the pairs whose evidence is absent, so the fine-tuned reader never sees a context that lacks the answer and, in a domain where the retriever misses often, learns to trust whatever it is given.

We propose TriageRAG[1], which lets the student's own behaviour decide how each self-generated example is used (Figure 1). After synthesizing question–answer pairs as in SimRAG, TriageRAG probes the student on every pair with and without retrieved context, triages it into known, solved, context-failure or retrieval-failure, and routes each category to a different treatment: contextual supervised fine-tuning (SFT) for context failures; a parametric route that trains the reader to produce the verified answer despite an unhelpful retrieved context—the situation it faces whenever retrieval misses—for retrieval failures; and small consolidation shares for the rest. Probing costs a fraction of fine-tuning, so the diagnosis is repeated every round and the mixture follows the student. The triage histogram is itself a label-free diagnostic of where the domain gap lies.

We evaluate on 11 datasets from medicine, natural science and computer science (CS) with a Qwen3-4B backbone, and on CS with Qwen3-8B as well, against SimRAG, a RAFT-style mixture and a random-selection control that share the same Stage-I model, synthetic pool, filter, recipe and budget. Our contributions are:

- A student-aware view of self-training for domain RAG in which each self-generated question is classified by failure type with cheap closed-book and open-book probes, yielding a label-free diagnostic of the domain gap.
- TriageRAG, which routes the four categories to different training signals and iterates, with SFT only and no reward model, judge or labeled data.
- Experiments on three domains and two backbone sizes in which TriageRAG outperforms a budget-matched SimRAG by 1.2–6.4 points, with three training seeds and paired bootstrap intervals on every headline comparison; an ablation that identifies the parametric route as the ingredient behind the gap; and a controlled negative result on tuning the retriever with retrieval failures.

[1]Failure-aware, triage-based self-training for RAG.

2 Related Work

Retrieval-augmented generation. RAG is a standard tool for knowledge-intensive tasks Lewis et al. (2020); Gao et al. (2023); recent work refines when to retrieve Jiang et al. (2023); Asai et al. (2024), makes readers robust to irrelevant passages Yoran et al. (2024); Shi et al. (2023); Cuconasu et al. (2024) and instruction-tunes LLMs for RAG Lin et al. (2024); Liu et al. (2024). For specialized domains, RAFT Zhang et al. (2024) fine-tunes a reader on GPT-4-written answers with the oracle document and distractors, including distractor-only examples, and Schimanski et al. (2024) distill context-aware QA data from GPT-4. SimRAG Xu et al. (2025) removes the proprietary teacher by giving the student question-answering and question-generation skills and keeping round-trip-consistent pairs Bartolo et al. (2021). TriageRAG keeps SimRAG's self-generation and adds the missing piece: the student's behaviour decides how each generated example is used.

Self-training and self-improvement. Self-training fits a student on teacher-produced labels and has been applied to LLM instruction tuning Wang et al. (2023); Li et al. (2024), alignment Yuan et al. (2024); Gulcehre et al. (2023) and, in SimRAG, to RAG; its main weakness is label noise. Recent self-play methods co-evolve a challenger and a solver and use the solver's pass rate to keep tasks near its frontier Huang et al. (2026); Kuba et al. (2025); Lu et al. (2026); Kwan et al. (2026); they optimize task difficulty for reinforcement learning of reasoning agents, whereas TriageRAG classifies the type of failure of a corpus-grounded RAG system to choose among training signals, with plain SFT.

Knowledge-aware fine-tuning. Gekhman et al. (2024) and Kang et al. (2025) show that fine-tuning on facts the model does not know encourages hallucination; KaFT Zhong et al. (2025) weights domain SFT examples by the conflict between parametric and contextual knowledge; Longpre et al. (2021) and Mallen et al. (2023) study when to trust memory over context. These works use labeled data. TriageRAG applies knowledge probing to self-generated data and adds the dimension that matters for RAG—whether the evidence was retrieved—so that reader and retriever failures are treated differently.

3 Methodology

3.1 Problem Setup

In a RAG problem, a retriever $\mathcal{R}$ returns the top-k passages $D = \{d_1, \ldots, d_k\}$ from a corpus $\mathcal{C}$ for a query q, and the LLM $\mathcal{M}_\theta$ generates an answer a from (q, D). We aim to improve the QA capability of $\mathcal{M}_\theta$ in a specialized domain where only the unlabeled corpus $\mathcal{C}$ is available. Following SimRAG, we first fine-tune the model on retrieval-oriented general-domain data (Stage-I) and then adapt it to the domain with self-generated pseudo-labeled tuples (Stage-II). TriageRAG changes Stage-II: instead of treating all retained tuples identically, it diagnoses the student on each tuple and routes the tuple to the training signal that fits the diagnosis. The retriever is kept frozen in the main configuration; Section 3.4 discusses a retriever route.

3.2 Stage-I: Retrieval-oriented Fine-tuning

We fine-tune an instruction-tuned backbone on a scaled-down version of SimRAG's Stage-I blend (16,000 examples; Appendix A): general context-aware QA in short-span, yes/no and multiple-choice formats, and the two skills used in synthesis, answer generation (candidate answer spans from a passage) and answer-conditioned question generation.

3.3 Stage-II: Failure-Aware Self-Training

Each round t takes the current student $\mathcal{M}_t$ ($\mathcal{M}_1$ is the Stage-I model) through four steps.

(1) Synthesis. For a sampled passage $d \in \mathcal{C}$ the student proposes candidate answer spans $a \sim \mathcal{M}_t(\cdot \mid d)$ and then an answer-conditioned question $q \sim \mathcal{M}_t(\cdot \mid a, d)$. As in SimRAG we

generate three formats: short-span questions; multiple-choice questions, which reuse a span question and add three distractor spans drawn from other passages, matched on whether the answer is numeric; and claims that are supported or refuted by d. We keep a tuple (q, a, d) only if the student recovers a when given the gold passage d (self-answerability). This label-noise filter is applied identically to every method we compare. We do not gate on round-trip retrieval consistency: whether the source passage is retrieved is recorded as a signal and used by the triage. The synthesis is run once per domain and the pool is shared by all methods and rounds.

(2) Probing. For every retained tuple we draw n samples from the student with no context and n samples with the top-k passages $D = \mathcal{R}(q)$ retrieved with the question alone, as at test time, and compute the closed-book and open-book pass rates $p_{\text{cb}} = \frac{1}{n}\sum_i \mathbb{1}[\hat{a}_i^{\text{cb}} \equiv a]$ and $p_{\text{ob}} = \frac{1}{n}\sum_i \mathbb{1}[\hat{a}_i^{\text{ob}} \equiv a]$, where $\equiv$ is exact match for letters and yes/no answers and a normalized token-overlap match for spans. The evidence hit is $h = \mathbb{1}[d \in D \vee a \subseteq \text{text}(D)]$.

(3) Triage. With a threshold τ (0.75 throughout) each tuple falls into exactly one category (the tests are applied in this order):

- known: $p_{\text{cb}} \geq \tau$; the student already knows the answer without context.
- solved: $p_{\text{cb}} < \tau$, $h = 1$, $p_{\text{ob}} \geq \tau$; the reader uses the retrieved evidence correctly.
- context-failure: $p_{\text{cb}} < \tau$, $h = 1$, $p_{\text{ob}} < \tau$; the evidence is in the context but the reader fails or is unstable.
- retrieval-failure: $p_{\text{cb}} < \tau$, $h = 0$; the evidence was not retrieved, so no contextual training can make the question answerable from D.

The category histogram is a diagnostic of the domain gap (Section 5.3).

(4) Routing and training. Given a budget of B Stage-II examples per round, we fill per-category quotas of 60%/20%/15%/5% for context-failure/retrieval-failure/solved/known; an unfilled quota is topped up from the remaining context failures, then solved items, without duplicates. In our pools the context-failure category is smaller than its quota in every domain, so every context failure is used and the shortfall is filled with solved items (realized shares in Appendix C). Each category receives a different treatment:

- Contextual SFT for context-failure: the target is a given (q, D); when the category exceeds its quota, tuples are sampled without replacement with probability $\propto 1 - p_{\text{ob}}$, so the items the student gets most wrong are preferred.
- Parametric SFT for retrieval-failure: the target is a given (q, D) where D does not contain the evidence. The answer is verified against the source passage, so the label is trustworthy, but the reader must produce it from parametric knowledge and learns that a retrieved context can be unhelpful—the condition it meets at test time whenever retrieval misses.
- Consolidation for solved and known: contextual SFT at small shares, plus a fixed replay of general-domain context-QA examples, to preserve answer formats and general skills.

The mixture is used for one epoch of LoRA fine-tuning Hu et al. (2022); the merged model becomes $\mathcal{M}_{t+1}$ and re-probes the pool in the next round.

3.4 A Retriever Route

retrieval-failure tuples are also natural retriever training data: (q, d) is a positive pair and the top-ranked retrieved passage is a mined hard negative. In a variant of TriageRAG we fine-tune the dense retriever on these pairs with a contrastive loss after each round and re-index. Section 5.2 shows that this degrades both retrieval and accuracy, so our main configuration keeps the retriever frozen.

4 Experimental Setup

4.1 Tasks and Datasets

We evaluate on 11 datasets from three domains (details in Appendix B). For the medical domain we use the five datasets of the MIRAGE benchmark Xiong et al. (2024)—PubMedQA Jin et al. (2019), BioASQ Tsatsaronis et al. (2015), MedQA Jin et al. (2021), MedMCQA Pal et al. (2022) and the medical subsets of MMLU Hendrycks et al. (2021)—and add the text subset of MedXpertQA Zuo et al. (2025), a recent expert-level benchmark with up to ten options. For the scientific domain we use ARC-Challenge Clark et al. (2018), SciQ Welbl et al. (2017) as four-way multiple choice, and the 14 scientific subsets of MMLU used by SimRAG. For computer science we use the English multiple-choice (MC) and assertion (AS) questions of CS-Bench Song et al. (2025), which can be scored automatically. All questions are multiple-choice, yes/no or yes/no/maybe, and accuracy is the metric throughout.

The unlabeled corpora, which also serve as the retrieval index at test time, are MedRAG textbook snippets plus a PubMed slice for medicine (950,746 passages), SciQ supports and OpenBookQA facts for science (10,863) and the Wikipedia computer-science subset used by SimRAG for CS (6,237); see Appendix B.

4.2 Baselines

Because our question is how self-generated data should be used, all Stage-II methods share the Stage-I model, the synthesized pool, the self-answerability filter, the budget B, the general replay and the training recipe, and differ only in which examples are selected and how their contexts are constructed. (1) No domain adaptation: the backbone closed-book, the backbone with RAG, and the Stage-I model with RAG (equivalent to "SimRAG w/o Stage-II"). (2) Self-training baselines at equal budget: SimRAG‡, our re-implementation, which keeps tuples whose source passage is among the top-k (round-trip consistency) and samples them uniformly with retrieved contexts; Random selection, which samples B tuples uniformly from the full pool with retrieved contexts and isolates the effect of selection from that of volume; and RAFT-style‡ Zhang et al. (2024), which places the gold passage among $k-1$ retrieved distractors for 80% of the examples and uses distractors only for the remaining 20%. (3) Ablations of TriageRAG (Section 5.2). We do not compare with GPT-4-distilled readers or with domain LLMs trained on task data overlapping our test sets; all rows use the same retriever and prompt.

4.3 Implementation Details

Backbones are Qwen3-4B-Instruct-2507 and Qwen3-8B in non-thinking mode Qwen Team (2025). All fine-tuning uses LoRA (r=64, α=128, all linear layers), one epoch, learning rate 10^{-4} with cosine decay, global batch 64 and maximum length 3,072. The retriever is bge-base-en-v1.5 Xiao et al. (2024) with a flat FAISS index Johnson et al. (2021) and k=5 for probing, training and evaluation. Synthesis uses 12,000 medical, 8,000 science and all 5,089 CS passages of at least 40 words at temperature 0.7, yielding pools of 40,540, 32,686 and 20,669 self-answerable tuples. Probing uses n=4 samples per condition at temperature 0.7 and τ=0.75. The per-round budget is B=12,000 for medicine and 6,000 for science and CS (30%, 18% and 29% of the respective pools, so that selection matters), plus 2,000 general replay examples. TriageRAG runs T=3 rounds; SimRAG runs one round as in the original paper and, as a control, three. Evaluation is zero-shot with greedy decoding; parse failures count as wrong (Appendix E). One medical round of TriageRAG (4B) takes about 15 minutes of probing with vLLM Kwon et al. (2023), 80 minutes of LoRA training (two A100s) and 10 minutes of evaluation. Prompts are in Appendix G.

Datasets # questions	PubMedQA 500	BioASQ 618	MedQA 1,273	MedMCQA 4,183	MMLU-Med 1,089	MedXpertQA 2,450	Avg. —
Backbone: Qwen3-4B-Instruct							
Backbone (closed-book)	31.00	77.67	60.49	54.24	76.03	<u>14.53</u>	52.33
Backbone + RAG	68.20	92.07	61.35	56.94	76.49	14.82	61.65
Stage-I + RAG	<u>82.00</u>	<u>92.56</u>	61.12	59.22	75.02	13.84	63.96
SimRAG‡	82.60	91.26	59.86	59.62	74.93	13.76	63.67
SimRAG‡, 3 rounds	80.60	91.59	58.99	59.12	74.47	13.27	63.01
Random selection	77.60	92.88	61.51	<u>61.92</u>	77.13	13.84	64.15
RAFT-style‡	79.00	92.39	61.04	61.44	77.04	13.67	64.10
TriageRAG	81.60	91.26	63.08	61.37	<u>78.33</u>	13.39	64.84
TriageRAG, 2 rounds	81.60	91.75	61.19	61.18	78.88	13.84	<u>64.74</u>
TriageRAG, 3 rounds	80.80	91.26	<u>61.90</u>	62.20	77.32	13.55	64.50

Table 1: Results in the medical domain (accuracy, %). All methods use RAG with the same bge-base retriever and top-5 passages at inference. Bold and <u>underline</u> mark the best and second-best result per column. ‡: our own implementation with the same backbone, pool and budget. All Stage-II methods share the Stage-I model, the 40,540-tuple self-generated pool, a budget of 12,000 examples per round and 2,000 general replay examples. The notations are the same for the following tables.

Datasets # questions	ARC-C 1,172	SciQ 1,000	MMLU-Sci 3,013	Avg. —
Backbone: Qwen3-4B-Instruct				
Backbone (closed-book)	89.25	94.60	<u>78.29</u>	87.38
Backbone + RAG	87.37	94.40	74.18	85.32
Stage-I + RAG	88.99	94.50	74.58	86.02
SimRAG‡	88.40	94.20	74.61	85.74
Random selection	<u>89.76</u>	95.70	77.07	87.51
TriageRAG	89.51	95.60	77.76	<u>87.62</u>
TriageRAG, 2 rounds	89.85	<u>95.80</u>	78.33	87.99
TriageRAG, 3 rounds	89.16	96.00	77.33	87.50

Table 2: Results in the scientific domain (Qwen3-4B; budget 6,000 per round; 32,686-tuple pool).

Datasets # questions	MC 1,336	AS 442	Avg. —
Backbone: Qwen3-4B-Instruct			
Backbone (closed-book)	68.79	67.19	67.99
Backbone + RAG	62.05	65.16	63.60
Stage-I + RAG	61.60	62.22	61.91
SimRAG‡	61.23	62.44	61.83
SimRAG‡, 3 rounds	60.93	64.71	62.82
Random selection	67.29	67.87	67.58
RAFT-style‡	67.37	67.19	67.28
TriageRAG	66.92	<u>69.46</u>	68.19
TriageRAG, 2 rounds	<u>67.89</u>	69.68	68.78
TriageRAG, 3 rounds	67.14	<u>69.46</u>	<u>68.30</u>
Backbone: Qwen3-8B			
Backbone (closed-book)	70.51	<u>69.00</u>	<u>69.75</u>
Backbone + RAG	63.25	71.49	67.37
Stage-I + RAG	64.30	67.42	65.86
SimRAG‡	63.70	67.65	65.68
Random selection	68.56	66.97	67.77
TriageRAG	<u>68.64</u>	71.49	70.06
TriageRAG, 2 rounds	68.04	67.42	67.73

Table 3: Results in the computer-science domain (CS-Bench; MC = multiple choice, AS = assertion; budget 6,000 per round; 20,669-tuple pool) with two backbones.

5 Experimental Results

5.1 Main Results

Tables 1, 2 and 3 present the results for the medical, scientific and computer-science domains; we make four observations.

Method	Medical							CS	Science
	PubMedQA	BioASQ	MedQA	MedMCQA	MMLU-Med	MedXpertQA	Avg.	CS Avg.	Sci. Avg.
TriageRAG	81.60	91.26	63.08	61.37	78.33	13.39	64.84	68.19	87.62
w/o parametric route	82.60	92.07	60.17	59.22	75.30	12.90	63.71	61.91	84.30
w/o triage (= random selection)	77.60	92.88	61.51	61.92	77.13	13.84	64.15	67.58	87.51
w/o triage, w/ round-trip filter (= SimRAG)	82.60	91.26	59.86	59.62	74.93	13.76	63.67	61.83	85.74
w/ retriever tuning	61.80	83.66	60.64	59.79	76.49	12.45	59.14	65.89	87.44
w/ retriever tuning, w/o parametric route	59.60	83.82	57.42	55.49	73.55	11.39	56.88	62.36	–

Table 4: Ablations of TriageRAG (Qwen3-4B, round 1). Medical results are per dataset; for CS and science we report the average. "w/o parametric route" puts the gold passage in the context of the retrieval-failure examples instead of training the reader to answer despite an unhelpful context (retriever frozen); "w/o triage" selects uniformly from the same pool at the same budget; "w/ retriever tuning" fine-tunes bge-base on the retrieval-failure pairs and re-indexes.

(1) TriageRAG consistently outperforms budget-matched self-training baselines. Against SimRAG at the same budget it gains 1.2 points on average in medicine, 1.9 in science and 6.4 on CS-Bench with the 4B backbone, and 4.4 on CS-Bench with the 8B backbone; it is the best fine-tuned method on average in every domain and the best or second best on 8 of the 11 datasets.

(2) Retrieval is not uniformly helpful, and self-training on retrieved contexts can lock in its harm. In medicine, five retrieved passages add 9.3 points to the backbone, almost entirely on the literature-grounded PubMedQA and BioASQ. In science and CS the backbone loses 2.1 and 4.4 points when given five passages (87.4→85.3, 68.0→63.6), and neither Stage-I nor SimRAG repairs the loss; TriageRAG is the only method that returns to or exceeds the closed-book level in a single round in all three cases (87.6 vs. 87.4; 68.2 vs. 68.0; 70.1 vs. 69.8 for Qwen3-8B), random selection does so in science only (87.5). In medicine its gain over Stage-I (+0.9) is concentrated on the exam-style datasets (MedQA +2.0, MedMCQA +2.2, MMLU-Med +3.3), while the literature yes/no datasets, on which retrieval already supplies the answer, move by at most 1.3 points. MedXpertQA stays near the ten-way chance level for every configuration and does not separate methods at this scale.

(3) Selection matters less than what the selected contexts look like. The random-selection control differs from SimRAG only in dropping the round-trip filter, yet it is 5.7, 1.8 and 0.5 points above SimRAG on CS, science and medicine (paired bootstrap over questions: +5.7 [3.4, 8.1] on CS, +1.8 [1.2, 2.4] on science) and within 0.1–0.7 of TriageRAG with the 4B backbone. The triage's own contribution over random selection is therefore small at 4B—+0.6 [-0.9, 2.1] on CS, +0.1 [-0.4, 0.6] on science, +0.7 [0.1, 1.3] in medicine—and becomes clear at 8B, where TriageRAG leads random selection by +2.3 [0.2, 4.4] on CS-Bench and on every training seed (Section 5.4). Two effects are at work—whether the mixture contains evidence-absent contexts (random selection, RAFT-style and TriageRAG do; SimRAG does not) and which questions are chosen—which Section 5.2 isolates.

(4) Iterating helps TriageRAG more than SimRAG, but a fixed pool saturates quickly. TriageRAG improves in round two on science and CS (87.6→88.0, 68.2→68.8) and plateaus in medicine (64.8, 64.7, 64.5), whereas SimRAG, whose selection does not depend on the student, drifts down in medicine when iterated (63.7→63.0) and gains only 1.0 point on CS (61.8→62.8), staying 5.5 points below TriageRAG. With the 8B backbone the second CS round regresses on the assertion subset.

5.2 Ablation Studies

Effect of the parametric route. The single ablation that separates TriageRAG from SimRAG is the treatment of the retrieval-failure examples. In Table 4 we keep the triage, the budget and the frozen retriever, and only replace the parametric route by the "obvious" alternative of inserting the gold passage into the context of these examples, so that the reader is again trained exclusively on contexts that contain the evidence. On CS-Bench this variant drops from 68.2 to 61.9, exactly the level of SimRAG (61.8) and of the Stage-I model (61.9); in the scientific domain it drops from 87.6 to 84.3, below SimRAG (85.7); and in the medical domain from 64.8 to 63.7, again the SimRAG level (63.7). Figure 2 (bottom)

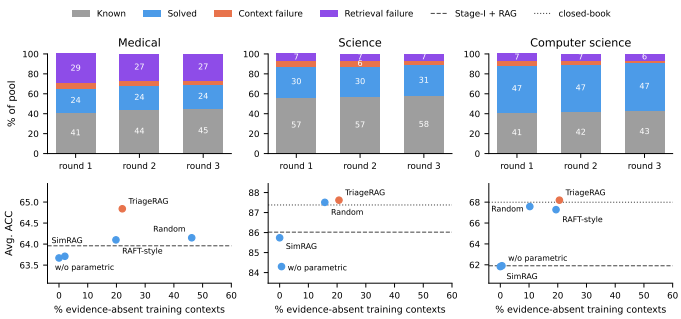


Figure 2: Top: triage of the self-generated pool by the Qwen3-4B student across rounds (shares in %). Bottom: average accuracy of each round-1 Stage-II mixture against the share of its training examples whose context does not contain the evidence; mixtures with at most 2% such examples (SimRAG, the ablation without the parametric route) stay at or below Stage-I, those with 10% or more reach within a point of the closed-book level or above.

makes the pattern visible across all three domains: mixtures in which no training context lacks the evidence (SimRAG, the ablation) sit at or below the Stage-I model; mixtures that contain such contexts (random selection, RAFT-style with 20% distractor-only examples, TriageRAG with its 20% parametric route) sit within a point of or above the closed-book model. A reader fine-tuned only on contexts that contain the answer learns to trust the context; in a domain where the retriever misses often, that reader is worse than no retrieval at all. Round-trip filtering, introduced in SimRAG to improve label quality, has the side effect of removing precisely the examples that teach a reader when not to trust its context.

Effect of the triage. Replacing the triage by uniform selection from the same pool ("w/o triage") costs 0.7 points in medicine, 0.6 on CS-Bench and 0.1 in science with the 4B backbone, and 2.3 on CS-Bench with the 8B backbone; adding the round-trip filter on top ("= SimRAG") costs a further 0.5, 5.7 and 1.8. In medicine both selections gain alike on the exam-style datasets relative to Stage-I (three-seed means within a point of each other on MedQA, MedMCQA and MMLU-Med); the difference between them is PubMedQA, which random selection lowers by 3.5 points on average (77.6, 81.0, 76.8 vs. 82.0 for Stage-I) and TriageRAG preserves (81.6, 82.6, 82.2; Figure 3).

Effect of retriever tuning (a negative result). The obvious remedy for retrieval-failures is to fix the retriever with the failed pairs (Section 3.4). Fine-tuning bge-base on them and re-indexing each round reduced recall@5 of source passages on the medical pool from 0.48 to 0.34 and accuracy from 64.8 to 59.1 (Table 4); with the round-1 reader of that run held fixed, swapping in the tuned index alone lowers PubMedQA/BioASQ/MedQA from 81.4/90.9/62.1 to 61.8/83.7/60.6 (Table 7). On CS the tuned retriever raised pool recall (0.87→0.92) yet lowered CS-Bench accuracy (65.9 vs. 68.2); in science it was neutral. The failed pairs are a biased sample—many questions are legitimately not specific to their source passage—so the contrastive update teaches the encoder noise and moves it away from the benchmark queries. We therefore keep the retriever frozen; curated negatives and answer-aware positives are left as the way to make a retriever route work.

5.3 Study on the Triage

What the triage reveals. Figure 2 and Table 6 (appendix) show the category shares: about 41% of the self-generated questions in medicine and CS, and 57% in science, are known: the Stage-I student answers them closed-book at least three times out of four, so they carry little information, yet SimRAG spends a proportional part of its budget on them. In CS and science the retriever finds the evidence for 84–89% of the questions and the student solves most of them; context-failures are 4–6% of the pool and retrieval-failures 7%. In medicine the source passage is retrieved for only 48% of the questions and 29% of the pool

are retrieval-failures, so a round-trip gate discards half of the pool and, more importantly, all the examples in which the context lacks the answer. Multiple-choice and claim items are overwhelmingly known; 80–94% of the failures come from short-span questions, so SimRAG's format diversity mostly supplies format practice rather than new knowledge.

Why round-trip filtering hurts. Figure 2 (bottom) plots each mixture's accuracy against the share of its examples whose training context lacks the evidence: zero for SimRAG by construction and at most 2% for the ablation without the parametric route; the miss rate of the sampled tuples for random selection (46% in medicine, 16% in science, 10% in CS); 20% for RAFT-style and 21–22% for TriageRAG. In the two domains where naive retrieval hurts, this share separates the mixtures into two groups—at most 2% (Stage-I level) and 10% or more (within a point of the closed-book level or above)—and the ablation falls onto the SimRAG point. The same mechanism fits the per-dataset pattern (Figure 3): exam questions are the case in which the retrieved passages are topically related but lack the answer, so the reader must combine them with what it knows.

5.4 Robustness: Training Seeds and Confidence Intervals

Every cell in Tables 1–3 is one training run; to check that the ordering is not a seed artefact we repeated the four headline Stage-II cells—SimRAG, random selection, TriageRAG and TriageRAG without the parametric route—with two further seeds (mixture sampling and LoRA initialisation) in every domain, and the three CS-Bench cells with the 8B backbone (Table 5, Appendix D). On every seed in every domain TriageRAG beats SimRAG and the ablation without the parametric route; TriageRAG vs. random selection and the ablation vs. SimRAG stay within two points. On three-seed means TriageRAG leads SimRAG by +5.1, +1.8 and +2.8 points on CS, medicine and science; removing the parametric route costs 5.8, 1.1 and 2.9 points and lands within 0.8 points of the SimRAG mean each time (seed standard deviations 0.1–1.2). Paired bootstrap intervals over test questions (10,000 resamples, seed-0 cells) agree: TriageRAG vs. SimRAG +6.4 [4.1, 8.6] on CS, +1.2 [0.5, 1.9] in medicine and +1.9 [1.2, 2.5] in science, all $p < 10^{-3}$; TriageRAG vs. its ablation without the parametric route +5.8 [3.5, 8.2] on CS; TriageRAG vs. random selection significant only in medicine at 4B (+0.7 [0.1, 1.3]) but +2.3 [0.2, 4.4] at 8B, where the three-seed gap is +2.0 points (69.9 ± 0.4 vs. 68.0 ± 0.2).

6 Conclusion

We introduce TriageRAG, a failure-aware self-training framework for adapting retrieval-augmented LLMs to specialized domains from an unlabeled corpus: it probes the student closed-book and open-book on every self-generated question, triages it by failure type and routes it to the training signal that fits. Across 11 datasets in three domains, and at two backbone sizes on CS-Bench, it consistently outperforms a budget-matched SimRAG, and a single ablation explains why: round-trip filtering removes every training context that lacks the evidence, and a reader that never sees such contexts over-trusts noisy retrieval. The triage doubles as a label-free diagnostic of where a domain gap lies; making retrieval failures useful to the retriever itself remains open.

7 Limitations

Our rounds re-probe a pool synthesized once, so the informative examples are consumed early and accuracy plateaus; regenerating the pool between rounds is future work. Our 4B/8B backbones, 16k-example Stage-I blend and PubMed slice make absolute numbers incomparable with SimRAG's 8B/27B models on full PubMed, textbooks and Wikipedia, and MedXpertQA is at chance for every configuration. Headline cells have three seeds and paired intervals; the other cells are single runs, and CS-Bench differences below one point should be read as ties. The parametric route teaches the reader to answer, not to abstain, when the context is unhelpful, and our negative retriever-tuning result covers one recipe (Appendix F expands each point).

AI use statement

Large language models were used as writing and coding assistants: to draft and edit prose, to generate analysis and plotting scripts, and to port the manuscript between templates. All experimental designs, numbers and claims were produced by the authors' code and checked by the authors against the result artifacts; LLM-generated code was tested before use. We have reviewed all AI-assisted text and code and take responsibility for the final content of this work, including any text, claims or artifacts produced with the aid of generative AI.

Reproducibility statement

All datasets, corpora and backbones are public (Section 4, Appendix B); prompts are in Appendix G, Stage-I data in Appendix A, and baseline and synthesis details in Appendix C. The code (data preparation, synthesis, probing, triage, mixture construction, LoRA training, evaluation and the paired-bootstrap script), the exact job lists that produced every cell, and per-example predictions for every reported run will be released; every number in the paper is generated from the summary files by one script.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In ICLR, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

Max Bartolo, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp, and Douwe Kiela. Improving question answering model robustness with synthetic adversarial data generation. In Proceedings of EMNLP, pp. 8830–8848, 2021. URL https://aclanthology.org/2021.emnlp-main.696/.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Proceedings of NAACL, pp. 2924–2936, 2019. URL https://aclanthology.org/N19-1300/.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. arXiv preprint arXiv:1803.05457, 2018. URL https://arxiv.org/abs/1803.05457.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. The power of noise: Redefining retrieval for RAG systems. In Proceedings of SIGIR, pp. 719–729, 2024. URL https://doi.org/10.1145/3626772.3657834.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997, 2023. URL https://arxiv.org/abs/2312.10997.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. Does fine-tuning LLMs on new knowledge encourage hallucinations? In Proceedings of EMNLP, pp. 7765–7784, 2024. URL https://aclanthology.org/2024.emnlp-main.444/.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. Reinforced self-training (ReST) for language modeling. arXiv preprint arXiv:2308.08998, 2023. URL https://arxiv.org/abs/2308.08998.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In ICLR, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In ICLR, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-Zero: Self-evolving reasoning LLM from zero data. In ICLR, 2026. URL https://openreview.net/forum?id=96apU6YzSO.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In Proceedings of EMNLP, pp. 7969–7992, 2023. URL https://aclanthology.org/2023.emnlp-main.495/.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. Applied Sciences, 11(14):6421, 2021. URL https://doi.org/10.3390/app11146421.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. PubMedQA: A dataset for biomedical research question answering. In Proceedings of EMNLP-IJCNLP, pp. 2567–2577, 2019. URL https://aclanthology.org/D19-1259/.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. IEEE Transactions on Big Data, 7(3):535–547, 2021. URL https://doi.org/10.1109/TBDATA.2019.2921572.

Katie Kang, Eric Wallace, Claire Tomlin, Aviral Kumar, and Sergey Levine. Unfamiliar finetuning examples control how language models hallucinate. In Proceedings of NAACL, pp. 3600–3612, 2025. URL https://aclanthology.org/2025.naacl-long.183/.

Jakub Grudzien Kuba, Mengting Gu, Qi Ma, Yuandong Tian, Vijai Mohan, and Jason Chen. Language self-play for data-free training. arXiv preprint arXiv:2509.07414, 2025. URL https://arxiv.org/abs/2509.07414.

Wai-Chung Kwan, Aryo Pradipta Gema, Joshua Ong Jun Leang, and Pasquale Minervini. SCOPE: Self-play via co-evolving policies for open-ended tasks. arXiv preprint arXiv:2605.31433, 2026. URL https://arxiv.org/abs/2605.31433.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: A benchmark for question answering research. Transactions of the Association for Computational Linguistics, 7:452–466, 2019. URL https://aclanthology.org/Q19-1026/.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In Proceedings of the 29th Symposium on Operating Systems Principles (SOSP), pp. 611–626, 2023. URL https://doi.org/10.1145/3600006.3613165.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Advances in Neural Information Processing Systems, volume 33, pp. 9459–9474, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Omer Levy, Luke Zettlemoyer, Jason Weston, and Mike Lewis. Self-alignment with instruction backtranslation. In ICLR, 2024. URL https://openreview.net/forum?id=1oijHJBRsT.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke Zettlemoyer, and Scott Yih. RA-DIT: Retrieval-augmented dual instruction tuning. In ICLR, 2024. URL https://openreview.net/forum?id=22OTbutug9.

Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. ChatQA: Surpassing GPT-4 on conversational QA and RAG. In Advances in Neural Information Processing Systems, volume 37, 2024. URL https://openreview.net/forum?id=bkUvKPKafQ.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. In Proceedings of EMNLP, pp. 7052–7063, 2021. URL https://aclanthology.org/2021.emnlp-main.565/.

Hongliang Lu, Yuhang Wen, Pengyu Cheng, Ruijin Ding, Jiaqi Guo, Haotian Xu, Chutian Wang, Haonan Chen, Xiaoxi Jiang, and Guanjun Jiang. Search self-play: Pushing the frontier of agent capability without supervision. In ICLR, 2026. URL https://openreview.net/forum?id=ZmGirmNJqE.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Proceedings of ACL, pp. 9802–9822, 2023. URL https://aclanthology.org/2023.acl-long.546/.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. In Proceedings of EMNLP, pp. 2381–2391, 2018. URL https://aclanthology.org/D18-1260/.

Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. Fine-tuning or retrieval? comparing knowledge injection in LLMs. In Proceedings of EMNLP, pp. 237–250, 2024. URL https://aclanthology.org/2024.emnlp-main.15/.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. MedMCQA: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Proceedings of CHIL, pp. 248–260, 2022. URL https://proceedings.mlr.press/v174/pal22a.html.

Qwen Team. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025. URL https://arxiv.org/abs/2505.09388.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Proceedings of EMNLP, pp. 2383–2392, 2016. URL https://aclanthology.org/D16-1264/.

Tobias Schimanski, Jingwei Ni, Mathias Kraus, Elliott Ash, and Markus Leippold. Towards faithful and robust LLM specialists for evidence-based question-answering. In Proceedings of ACL, pp. 1913–1931, 2024. URL https://aclanthology.org/2024.acl-long.105/.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In ICML, pp. 31210–31227, 2023. URL https://proceedings.mlr.press/v202/shi23a.html.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation. In Findings of EMNLP, pp. 3784–3803, 2021. URL https://aclanthology.org/2021.findings-emnlp.320/.

Xiaoshuai Song, Muxi Diao, Guanting Dong, Zhengyang Wang, Yujia Fu, Runqi Qiao, Zhexu Wang, Dayuan Fu, Huangxuan Wu, Bin Liang, Weihao Zeng, Yejie Wang, Zhuoma GongQue, Jianing Yu, Qiuna Tan, and Weiran Xu. CS-Bench: A comprehensive benchmark for large language models towards computer science mastery. In ICLR, 2025. URL https://openreview.net/forum?id=fjEZ2LPceZ.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R. Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition. BMC Bioinformatics, 16(1):138, 2015. URL https://doi.org/10.1186/s12859-015-0564-6.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In Proceedings of ACL, pp. 13484–13508, 2023. URL https://aclanthology.org/2023.acl-long.754/.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In Proceedings of the 3rd Workshop on Noisy User-generated Text, pp. 94–106, 2017. URL https://aclanthology.org/W17-4413/.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-Pack: Packed resources for general Chinese embeddings. In Proceedings of SIGIR, pp. 641–649, 2024. URL https://doi.org/10.1145/3626772.3657878.

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. Benchmarking retrieval-augmented generation for medicine. In Findings of ACL, pp. 6233–6251, 2024. URL https://aclanthology.org/2024.findings-acl.372/.

Ran Xu, Hui Liu, Sreyashi Nag, Zhenwei Dai, Yaochen Xie, Xianfeng Tang, Chen Luo, Yang Li, Joyce C. Ho, Carl Yang, and Qi He. SimRAG: Self-improving retrieval-augmented generation for adapting large language models to specialized domains. In Proceedings of NAACL, pp. 11534–11550, 2025. URL https://aclanthology.org/2025.naacl-long.575/.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context. In ICLR, 2024. URL https://openreview.net/forum?id=ZS4m74kZpH.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. In ICML, pp. 57905–57923, 2024. URL https://proceedings.mlr.press/v235/yuan24d.html.

Tianjun Zhang, Shishir G. Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E. Gonzalez. RAFT: Adapting language model to domain specific RAG. In First Conference on Language Modeling (COLM), 2024. URL https://openreview.net/forum?id=rzQGHXNReU.

Qihuang Zhong, Liang Ding, Xiantao Cai, Juhua Liu, Bo Du, and Dacheng Tao. KaFT: Knowledge-aware fine-tuning for boosting LLMs' domain-specific question-answering performance. In Findings of ACL, pp. 24085–24100, 2025. URL https://aclanthology.org/2025.findings-acl.1235/.

Yuxin Zuo, Shang Qu, Yifei Li, Zhangren Chen, Xuekai Zhu, Ermo Hua, Kaiyan Zhang, Ning Ding, and Bowen Zhou. MedXpertQA: Benchmarking expert-level medical reasoning and understanding. In ICML, pp. 80961–80990, 2025. URL https://proceedings.mlr.press/v267/zuo25a.html.

A Stage-I Data

The Stage-I training set is a uniform sample of 16,000 examples from a 20,930-example general blend that mirrors the three components of SimRAG's Stage-I data: (1) context-aware QA—SQuAD Rajpurkar et al. (2016) short-span questions (3,000), BoolQ Clark et al. (2019) yes/no questions (3,000) and OpenBookQA Mihaylov et al. (2018) multiple-choice questions with the supporting fact as context (3,000); (2) answer-span generation—SQuAD passages with all their annotated answer spans as the target (2,930); and (3) question generation—SQuAD answer-conditioned question generation (3,000) and Natural Questions Kwiatkowski et al. (2019) question generation from a passage without a given answer (6,000).

We omit SimRAG's general instruction-following data because our backbones are already instruction-tuned. One epoch of LoRA training (250 steps at global batch 64) took 21 minutes on four A100s for the 4B backbone and 67 minutes on two A100s for the 8B backbone. The 2,000 general replay examples used in every Stage-II round are context-QA examples drawn from the same blend.

B Test Data Details

We evaluate on 11 datasets. (1) Medical:

- PubMedQA Jin et al. (2019): 500 yes/no/maybe questions about PubMed abstracts (MIRAGE split).
- BioASQ Tsatsaronis et al. (2015): 618 yes/no questions from biomedical literature (MIRAGE split).
- MedQA Jin et al. (2021): 1,273 four-option questions from the US Medical Licensing Examination.
- MedMCQA Pal et al. (2022): the 4,183-question development set of Indian medical entrance-exam questions.
- MMLU-Med Hendrycks et al. (2021): 1,089 questions from the six biomedical MMLU subjects (anatomy, clinical knowledge, professional medicine, medical genetics, college medicine, college biology).
- MedXpertQA Zuo et al. (2025): the 2,450 text-only questions of the expert-level benchmark, with up to ten options.

(2) Scientific: ARC-Challenge Clark et al. (2018) (1,172 test questions), SciQ Welbl et al. (2017) (1,000 test questions, kept as four-way multiple choice so that all tasks share one metric), and MMLU-Sci (3,013 questions from the 14 scientific MMLU subjects used by SimRAG). (3) Computer science: the English multiple-choice (1,336) and assertion (442) questions of CS-Bench Song et al. (2025); we omit the fill-in-the-blank and open-ended subsets, which require an LLM judge.

Corpora. Medical: MedRAG textbook snippets of at least 25 words Xiong et al. (2024) (123,799 of 125,847), 40 of the 1,166 MedRAG PubMed chunk files spread over the PMID range (822,980 abstracts of at least 40 words) and the abstracts of the 3,967 PMIDs referenced by PubMedQA and BioASQ retrieved from NCBI E-utilities, mirroring MedCorp in which the full PubMed is available; 950,746 passages in total. Science: SciQ training and validation supports and OpenBookQA facts (10,863 passages). Computer science: the Wikipedia computer-science subset used by SimRAG, chunked to 180 words (6,237 passages). Indices are flat inner-product FAISS indices over bge-base-en-v1.5 embeddings.

C Baseline and Synthesis Details

Synthesis. Passages with fewer than 40 words are skipped. Candidate answer spans must appear verbatim in the passage and have at most eight words. Generated questions must end with a question mark and must not contain the answer string. Multiple-choice distractors are candidate spans from other passages that do not occur in the source passage, matched on the presence of digits and within three words of the answer's length; option order is shuffled. Claims alternate between supported and refuted per passage. Self-answerability uses greedy decoding with the source passage as the only context. The pools contain 40,540 (medical; from 12,000 passages), 32,686 (science; 8,000 passages) and 20,669 (CS; 5,089 passages: 11,908 span, 4,997 multiple-choice, 3,764 claim) self-answerable tuples.

SimRAG‡. Tuples whose source passage is among the top-5 retrieved passages for the generated question are kept (48%, 78% and 87% of the medical, science and CS pools); B of them are sampled uniformly and trained with the retrieved passages as context, plus the 2,000 general replay examples.

Random selection. B tuples are sampled uniformly from the full pool and trained with the retrieved passages as context.

RAFT-style[‡]. B tuples are sampled uniformly; for 80% the gold passage replaces one of the retrieved passages at a random position, for 20% only retrieved passages other than the gold one are used. Targets are the short answers rather than GPT-4 chain-of-thought answers, so that all methods share the same label space.

Realized TriageRAG mixture. With the 60/20/15/5% quotas, the round-1 mixtures contain 2,324 context-failure (all of them), 2,400 retrieval-failure, 6,676 solved and 600 known tuples in medicine (B=12,000; 19/20/56/5%); 1,960 (all), 1,200, 2,540 and 300 in science (33/20/42/5%); and 914 (all), 1,200, 3,586 and 300 on CS (B=6,000; 15/20/60/5%), plus the 2,000 general replay examples in every case. The measured share of training contexts that lack the evidence is 22.0%, 20.7% and 20.6% (medicine, science, CS) for TriageRAG, 46.2%, 15.7% and 10.3% for random selection, 19.8% and 19.4% for RAFT-style (medicine, CS), and 2.0%, 0.7% and 0.6% for the ablation without the parametric route.

Retriever tuning. bge-base-en-v1.5 is fine-tuned for one epoch on the round's retrieval-failure pairs (question, source passage) with a multiple-negatives ranking loss over in-batch negatives plus one mined hard negative per pair, the top-ranked retrieved passage other than the source (batch 64, learning rate 2×10^{-5}, maximum length 256); the corpus is then re-embedded and re-indexed.

D Additional Analyses

Domain / backbone	Method	seed 0	seed 1	seed 2	mean ± sd
CS-Bench, Qwen3-4B	SimRAG[‡]	61.83	62.25	63.01	62.36 ± 0.59
	Random selection	67.58	65.47	67.58	66.88 ± 1.22
	TriageRAG w/o parametric route	61.91	61.31	61.84	61.69 ± 0.33
	TriageRAG	68.19	67.09	67.13	67.47 ± 0.62
Medical, Qwen3-4B	SimRAG[‡]	63.67	62.51	62.92	63.03 ± 0.59
	Random selection	64.15	65.04	63.78	64.32 ± 0.65
	TriageRAG w/o parametric route	63.71	63.58	64.03	63.77 ± 0.23
	TriageRAG	64.84	64.51	65.27	64.87 ± 0.38
Science, Qwen3-4B	SimRAG[‡]	85.74	83.94	84.99	84.89 ± 0.90
	Random selection	87.51	87.71	87.78	87.67 ± 0.14
	TriageRAG w/o parametric route	84.30	85.75	84.37	84.81 ± 0.82
	TriageRAG	87.62	87.68	87.84	87.71 ± 0.11
CS-Bench, Qwen3-8B	SimRAG[‡]	65.68	65.82	65.78	65.76 ± 0.07
	Random selection	67.73	67.95	68.18	67.95 ± 0.23
	TriageRAG	70.06	69.43	70.22	69.90 ± 0.42

Table 5: Average accuracy (%) of the round-1 Stage-II cells over three training seeds (seed 0 is the run reported in Tables 1–3). Seeds change the mixture sample and the LoRA initialisation; the Stage-I model, pool and round-1 probe are shared. On every seed TriageRAG beats SimRAG and the ablation without the parametric route; TriageRAG and random selection, and likewise the ablation and SimRAG, are within two points of each other.

Case studies. Table 8 (appendix) shows one self-generated medical question per category: a textbook fact the student recites without context (known); a survival statistic answered only once the passage is present (solved); a question whose evidence is in the top-5 passages while the top-ranked passage supports a competing answer (context-failure); and a rare complication whose abstract is not retrieved, leaving topically related but unhelpful passages (retrieval-failure)—the one example of the four that SimRAG's filter discards.

Table 6 gives the round-1 triage statistics with a per-format breakdown, Table 7 the controlled index comparison behind the retriever-tuning result, and Figure 3 the per-dataset view of the medical ablations.

Pool	Size	Known	Solved	Ctx.F.	Ret.F.	Hit@5	R@5
Medical	40,540	40.9	24.2	5.7	29.2	53.4	48.1
span	19,553	4.5	36.1	11.0	48.4		
MC	11,775	75.9	8.8	0.1	15.2		
claim	9,212	73.3	18.6	1.8	6.3		
Science	32,686	56.6	30.3	6.0	7.0	84.2	77.9
span	18,622	31.5	47.7	9.9	11.0		
MC	7,845	93.2	4.9	0.2	1.7		
claim	6,219	85.8	10.5	1.8	1.9		
CS	20,669	41.2	47.2	4.4	7.1	88.6	87.0
span	11,908	12.4	70.3	6.6	10.6		
MC	4,997	84.8	12.3	0.2	2.7		
claim	3,764	74.7	20.5	3.2	1.7		

Table 6: Round-1 triage of the self-generated pools (Qwen3-4B Stage-I student, frozen retriever; shares in %). Hit@5: fraction of questions whose evidence is in the top-5 passages (h=1); R@5: fraction whose source passage is among them.

Index at test time	PubMedQA	BioASQ	MedQA
Base bge-base	81.4	90.9	62.1
Tuned on round-1 failures	61.8	83.7	60.6
Recall@5 on the pool	0.48 (base) $\rightarrow$ 0.34 (tuned)		

Table 7: Controlled comparison holding one round-1 medical reader fixed (Qwen3-4B; the round-1 reader of the retriever-tuning run, trained on the same round-1 mixture as TriageRAG in a separate run, hence the small differences from Table 1): fine-tuning the retriever on retrieval-failure pairs degrades both recall of the source passage and downstream accuracy.

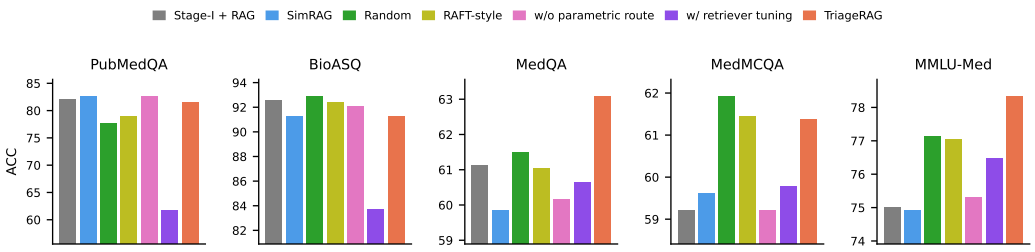


Figure 3: Per-dataset accuracy of the Stage-II variants in the medical domain (Qwen3-4B, round 1). Note the different y-ranges. The literature yes/no datasets (PubMedQA, BioASQ) move by at most 1.3 points under SimRAG, the ablation without the parametric route and TriageRAG; random selection (−4.4) and RAFT-style (−3.0) lower PubMedQA, and retriever tuning damages both. The exam-style datasets (MedQA, MedMCQA, MMLU-Med) carry the gains of TriageRAG over Stage-I.

known	(p_{cb}=1.00, p_{ob}=1.00, source passage retrieved at rank 1) → 5% consolidation share
Question / Answer	Where does the lymphatic system filter the lymph? A: lymph nodes
Source (Textbooks)	Physiology, Levy. In addition to returning fluid and protein to the vascular bed, the lymphatic system filters the lymph at the lymph nodes and removes foreign particles such as bacteria.
Top retrieved	(same passage)
solved	(p_{cb}=0.00, p_{ob}=1.00, source passage retrieved at rank 1) → 15% consolidation share
Question / Answer	What was the 10-year survival rate for relapsing polychondritis? A: 55%
Source (Textbooks)	Internal Medicine, Harrison. The course of relapsing polychondritis is highly variable. … In one study, the 5-year estimated survival rate was 74% and the 10-year survival rate was 55%.
Top retrieved	(same passage)
context-failure	(p_{cb}=0.50, p_{ob}=0.00, evidence in the top-5 passages, source passage not retrieved) → contextual SFT, weight $\propto 1 - p_{\text{ob}}$
Question / Answer	What type of cancer do long chain n-3 PUFA help to prevent? A: breast cancer
Source (PubMed)	Endogenous synthesis of n-3 polyunsaturated fatty acids in Fat-1 mice … Long chain n-3 PUFA have been shown to have chemopreventive properties against breast cancer through various mechanisms.
Top retrieved	Dietary n-3/long-chain n-3 polyunsaturated fatty acids for prevention of sporadic colorectal tumors: a randomized controlled trial … To address preventive effects of n-3 PUFAs on CRTs, a randomized controlled trial was conducted. (the answer string appears only in a lower-ranked passage; the top passage supports a competing answer)
retrieval-failure	(p_{cb}=0.00, p_{ob}=0.00, evidence absent from the top-5 passages) → parametric SFT with the retrieved (unhelpful) context
Question / Answer	What is a rare complication after renal transplantation? A: iliac artery stenosis
Source (PubMed)	Iliac artery stenosis as a cause of posttransplant renal failure and claudication. Iliac artery stenosis (IAS) is a rare complication after renal transplantation. We demonstrate a case of ipsilateral external IAS proximally to the anastomosis in a kidney recipient …
Top retrieved	Ultrasound evaluation of uncommon renal transplant complications. Twenty-five patients who had undergone renal transplants developed uncommon complications that were detected and followed serially with sonography. Two patients developed renal calculi … (topically related, answer absent)

Table 8: Case studies: one self-generated medical question per triage category with its probe outcomes (p_{cb}, p_{ob}, evidence hit) and the route it receives. Green marks evidence present in the retrieved passages; red marks the answer-bearing sentence of the source passage when it was not retrieved.

E Additional Results

Effect of rounds. Figure 4 in Appendix E plots the average accuracy per round (all numbers in Table 9). Because TriageRAG re-probes the pool with the updated student, the mixture shifts: in the medical domain the known share rises from 40.9% to 44.2% and 45.2% over the three rounds and the retrieval-failure share falls from 29.2% to 27.1% and 26.8%, i.e. the student has absorbed part of what it was trained on; on CS-Bench the context-failure share falls from 4.4% to 2.7%. Accuracy, however, moves little after the first round: medicine plateaus (64.8, 64.7, 64.5), science peaks at round two (87.6, 88.0, 87.5) and so does CS-Bench (68.2, 68.8, 68.3). With a fixed pool the informative examples are largely consumed in round one; the retriever-tuned variant is the exception on CS (65.9, 67.7, 68.0), where later rounds partially recover from the damaged index. Iterating SimRAG lowers medical accuracy (63.7, 63.5, 63.0) and gains at most 1.5 points on CS (61.8, 63.4, 62.8), never approaching TriageRAG. Re-probing is therefore most useful when the pool is refreshed between rounds.

Table 9 lists the average accuracy of every iterated method per round for all domains and backbones, and Figure 4 plots them for the 4B backbone.

Answer parsing. Letters are parsed from the first line of the output (patterns A, (A), A., "answer is A"); yes/no/maybe and true/false are matched as whole words. Unparsable outputs are counted as wrong. Across all evaluations of fine-tuned models the number of unparsable outputs was at most 3 per dataset; the untuned backbone with retrieval produced

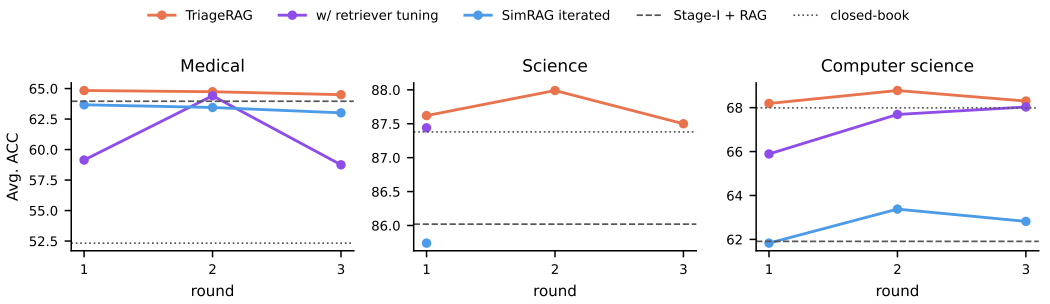


Figure 4: Average accuracy per round (Qwen3-4B). Re-probing the updated student changes the mixture of TriageRAG every round; SimRAG's selection does not depend on the student, so its later rounds re-train on the same distribution.

Domain / backbone	Method	Round 1	Round 2	Round 3
Medical / Qwen3-4B-Instruct	TriageRAG	64.84	64.74	64.50
	TriageRAG w/ retriever tuning	59.14	64.41	58.75
	SimRAG‡ iterated	63.67	63.45	63.01
Science / Qwen3-4B-Instruct	TriageRAG	87.62	87.99	87.50
	TriageRAG w/ retriever tuning	87.44	–	–
	SimRAG‡ iterated	85.74	–	–
Computer science / Qwen3-4B-Instruct	TriageRAG	68.19	68.78	68.30
	TriageRAG w/ retriever tuning	65.89	67.69	68.03
	SimRAG‡ iterated	61.83	63.38	62.82
Computer science / Qwen3-8B	TriageRAG	70.06	67.73	–
	SimRAG‡ iterated	65.68	–	–

Table 9: Average accuracy per round for the iterated methods.

86 unparsable outputs on MedMCQA and 32 on CS-Bench MC (it sometimes answers with a sentence instead of a letter), which slightly disadvantages the "+ RAG" rows.

F Extended Limitations

Fixed pool and saturation. Our rounds re-probe a pool that is synthesized once; the informative examples are largely consumed in the first round and accuracy plateaus. Regenerating the pool from new passages between rounds, as the shifting triage suggests, is left to future work.

Scale and corpora. We use 4B and 8B backbones, a Stage-I blend of 16k examples and a PubMed slice rather than all of PubMed, so absolute numbers are not comparable with SimRAG's 8B/27B models, which use full PubMed, textbooks and Wikipedia; all our comparisons are internal and budget-matched. MedXpertQA is at chance level for every configuration we trained and cannot separate methods at this scale.

Seeds and intervals. The headline round-1 cells were replicated with three training seeds and every comparison in the main text carries a paired bootstrap interval; the remaining cells (rounds 2–3, RAFT-style and the retriever-tuned variants) are single runs, and CS-Bench differences of one point or less between the strong variants should be read as ties.

Abstention and calibration. The parametric route teaches the reader to produce a verified answer when the context is unhelpful; it does not teach abstention, and we did not measure calibration or behaviour on unanswerable questions.

Retriever route. Our negative result covers one reasonable recipe (contrastive fine-tuning of bge-base on failed pairs with retrieved hard negatives). Curated negatives, smaller learning rates or mixing with the original training data may reverse it.

G Prompt Details

Inference and training. System message: "You are a helpful, knowledgeable assistant. Use the provided context when it is relevant." User message:

> Context:
> [1] passage 1 …[5] passage 5
> {dataset-specific instruction, e.g. "Answer the following question by selecting one of the provided options with A, B, C, or D. Please answer with the capitalized alphabet only, without adding any extra phrase or period."}
> Question: {question}
> A. … B. … C. … D. …

Closed-book probing uses the same message without the context block.

Answer generation.

> Context:
> {passage}
> Based on the context, generate several candidate spans within the passage that are likely to be answers to a question. The answers can be entities, verbs or even numbers. Make sure that the answers are different and diverse. Separate different candidate answers with a semicolon (';'). Output only the candidate answers.

Question generation.

> Context:
> {passage}
> Based on the context, please generate a question that is relevant to the information provided. The question should stand alone and not refer back to the context explicitly. The question should be clear and understandable without needing the context. The answer to the question should be: {answer}
> Output only the question.

Claim generation.

> Context:
> {passage}
> Based on the context, please write one specific factual claim that is {supported | refuted} by the context. The claim should stand alone and be understandable without the context. Output only the claim.