EVIDENCE-GATED CORRECTION: A CONTROLLED STUDY OF RECURSIVE RETRIEVAL IN SCIENTIFIC QA

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval-augmented generation (RAG) systems for scientific question answering can produce answers whose support is not verified against retrieved evidence, yet many correction approaches either omit explicit evidence verification or replace answers once correction is triggered. We study recursive correction as an evidence-gated control problem: after an initial answer, a controller decides whether to preserve it, retrieve additional evidence, generate a candidate correction, or commit that candidate, using NLI-based evidence scoring and recursive PubMed retrieval. We evaluate this design, SelfHeal-RAG, on a frozen, question-only held-out set of 150 BioASQ yes/no questions, with no gold answers, snippets, or context available at inference. Evidence-gated correction improves paired accuracy from 42.67% to 51.33% (+8.67 percentage points; exact McNemar $p = 0.000244$), with all 13 changed decisions beneficial and none harmful. A retrieval-off control reproduces the baseline exactly, consistent with recursive retrieval contributing to the observed gain. Two results qualify this finding: an unconditional-replacement variant reaches 54.00%, so this experiment does not establish an accuracy advantage for selective gating over unconditional replacement; and in a diagnostic with answer-bearing context supplied directly, baseline accuracy is 74.5% and correction adds only 1.5 percentage points ($p = 0.25$). Together, these results provide evidence that recursive evidence acquisition can improve scientific question answering relative to one-pass correction, while leaving the relative value of selective commitment versus simpler unconditional correction unresolved.

1 INTRODUCTION

Retrieval-augmented generation (RAG) grounds language model outputs in retrieved evidence and is a standard approach for knowledge-intensive tasks such as scientific and biomedical question answering (Lewis et al., 2020). Standard RAG pipelines, however, are single-pass: evidence is retrieved once, an answer is generated, and it is returned regardless of whether the retrieved evidence actually supports it. When initial retrieval fails to surface answer-bearing evidence – common in specialized scientific domains where relevant documents may be sparse, differently indexed, or absent from a fixed corpus – the resulting answer can be unsupported with no mechanism to detect or address this.

A natural response is to add verification and regenerate on failure. But regeneration alone conflates two distinct decisions: whether to attempt a correction at all, and whether a generated candidate should actually replace the current answer. Prior corrective and adaptive RAG methods evaluate retrieved evidence and trigger a single fallback action – a web search (Yan et al., 2024), an alternate retrieval strategy (Jeong et al., 2024), or a sufficiency-based abstention (Joren et al., 2025) – but do not, to our knowledge, separately adjudicate a specific generated candidate against the current answer inside a bounded recursive loop.

This paper studies exactly that separation. We formulate recursive correction as a controller that, at each iteration, can preserve the current answer, retrieve additional evidence, generate a candidate correction, or commit that candidate – and we test empirically whether this separation matters. Our central experimental question is: *does explicitly gating candidate commitment on evidence adjudication improve accuracy over simply committing every generated candidate unconditionally?*

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.

We evaluate SelfHeal-RAG, a system combining hybrid dense-sparse retrieval, NLI-based evidence scoring, and recursive PubMed retrieval, on a frozen, question-only held-out set of 150 BioASQ yes/no questions. Our findings are twofold and, notably, in tension with a common design assumption. First, recursive, evidence-triggered correction significantly improves paired accuracy over a non-recursive baseline (42.67% $\rightarrow$ 51.33%, exact McNemar $p = 0.000244$), and a retrieval-off ablation that exactly reproduces the baseline indicates that additional retrieval, not the controller's adjudication logic alone, drives this gain. Second, and central to this paper, an unconditional-replacement variant that skips evidence-gated adjudication entirely reaches numerically higher accuracy (54.00%), and the paired difference against our gated controller is not statistically significant ($p = 0.125$).

Our contributions are:

1. A recursive, evidence-gated correction controller for scientific yes/no QA that explicitly separates candidate generation from candidate commitment, extending single-step evaluation methods (Yan et al., 2024; Joren et al., 2025) with a bounded, PICO-targeted recursive retrieval loop.
2. A controlled evaluation protocol that isolates this design choice via an unconditional-replacement ablation and a retrieval-off ablation on identical, frozen inputs, with paired statistical testing.
3. An empirical finding that recursive retrieval significantly improves paired accuracy over a one-pass baseline, while selective commitment does not significantly improve over unconditional replacement – identifying an open question for correction-controller design that we discuss but do not resolve.

2 Related Work

Retrieval-augmented generation. RAG conditions generation on retrieved passages to ground outputs in external evidence (Lewis et al., 2020). Standard RAG performs retrieval once per query and does not verify whether the retrieved evidence actually supports the generated output.

Adaptive and iterative retrieval. FLARE (Jiang et al., 2023) triggers retrieval mid-generation when token-level confidence drops; Adaptive-RAG (Jeong et al., 2024) routes a query to a no-retrieval, single-step, or multi-step strategy via a trained query-complexity classifier. Closest to our setting, i-MedRAG (Xiong et al., 2024) improves biomedical RAG by iteratively generating follow-up questions and re-retrieving. None adjudicates a specific generated candidate against the answer currently held.

Self-corrective and reflective generation. Self-RAG (Asai et al., 2024) trains reflection tokens into the model so it decides, per generation segment, whether to retrieve and how to critique its own output – correction implicit in a single learned process rather than an external, separable controller. Reflexion (Shinn et al., 2023) and Chain-of-Verification (Dhuliawala et al., 2024) use verbal self-feedback loops without external retrieval. Huang et al. (Huang et al., 2024) argue that large language models cannot reliably self-correct reasoning through introspection alone, without external feedback; this motivates our reliance on externally retrieved evidence, rather than pure self-reflection, as the correction signal. CRITIC (Gou et al., 2024) critiques and revises outputs via external tools including search, and Corrective RAG (CRAG) (Yan et al., 2024) classifies retrieved documents and triggers a single web-search fallback when evidence is judged incorrect.

Evidence sufficiency and selective generation. Most directly related to our sufficiency mechanism, Joren et al. (Joren et al., 2025) introduce Sufficient Context, a classifier labeling whether retrieved context contains enough information to answer a question, used for selective generation – answering only when context is judged sufficient, otherwise abstaining. Our evidence scoring extends this concept with a finer-grained, five-way relation (support, contradiction, irrelevant, insufficient, conflicting) at PICO-slot-level granularity. Critically, Sufficient Context gates abstention on a single evaluation and does not trigger additional retrieval when context is judged insufficient; we instead pair a comparable sufficiency judgment with a recursive retrieval loop that targets the specific missing evidence.

Hallucination evaluation. HaluEval (Li et al., 2023) provides benchmarks for detecting hallucinated content but does not itself constitute a correction mechanism. We do not claim to reduce hallucination as externally validated against such benchmarks; our evidence score is an internal, uncalibrated proxy (Section 4).

Positioning. We do not claim to be the first to critique or correct RAG outputs; the individual ingredients – evidence critique (Yan et al., 2024), self-reflection (Asai et al., 2024), and iterative biomedical retrieval (Xiong et al., 2024) – all predate this work. We study a bounded recursive retrieval formulation with an explicit, separable generate-then-adjudicate action space (preserve, retrieve-again, generate, commit, reject) for scientific yes/no QA. Rather than claiming that these individual ingredients are new, our contribution is to isolate their empirical value: the central experiment directly compares selective adjudication with unconditional replacement under otherwise matched recursive correction. Concurrent work by Ateia and Kruschwitz (Ateia & Kruschwitz, 2025) investigates LLM self-feedback for retrieval-augmented generation at BioASQ 2025 and reaches a compatible cautionary conclusion about the reliability of self-critique; we read it as corroborating, rather than undermining, our negative finding on selective gating.

3 Problem Formulation

Let q be a question and $E_0 = R(q)$ the evidence retrieved for it by an initial retrieval function R. An initial answer $y_0 = G(q, E_0)$ is generated by a language model G. We define an evidence-support score $S_E(y, E) \in [0, 1]$ (Section 4) that estimates the degree to which the claims in answer y are entailed by evidence set E.

At each iteration $t = 1, \ldots, T$ (bounded by $T = 3$), the system is in a state $s_t = (y_t, E_t, \ell_t)$, where ℓ_t is the extracted decision label (Yes, No, or Maybe) of y_t. A controller π observes s_t together with evidence-relation diagnostics o_t (entailment/contradiction relations, novelty of retrieved passages relative to E_0, and proposition-coverage signals; Section 4) and selects an action

$$a_t = \pi(s_t, o_t) \in \{\textsc{Preserve}, \textsc{Retrieve_Again}, \textsc{Generate}, \textsc{Commit}, \textsc{Reject}\}.$$

GENERATE produces a candidate answer $\hat{y}_t = G(q, E'_t)$ from updated evidence $E'_t = E_t \cup R_{\text{PubMed}}(q, E_t)$; a subsequent adjudication decides COMMIT (set $y_{t+1} = \hat{y}_t$) or REJECT/PRESERVE (retain $y_{t+1} = y_t$). RETRIEVE_AGAIN updates E_t without generating a candidate. The loop terminates when $S_E(y_t, E_t) \geq \tau$, when T iterations are exhausted, or when no further distinct query can be formulated. We emphasize that π is a deterministic, rule-based state machine (Section 4); it is not a learned or reinforcement-trained policy, and no numeric threshold is optimized inside π itself – τ governs only loop entry and continuation, upstream of π.

Central experimental question. We compare π (the evidence-gated controller, *V5-gated*) against an ablation that replaces the COMMIT/REJECT adjudication with unconditional acceptance of every generated candidate (*unconditional replacement*), holding the pre-generation opportunity, retrieval, and generation identical. This isolates the accuracy contribution of the adjudication step itself.

4 Method: SelfHeal-RAG

SelfHeal-RAG runs as a staged pipeline. The question is processed and used for hybrid dense–sparse retrieval, forming an initial evidence set E_0; an initial answer $y_0 = G(q, E_0)$ is generated and reflection-tagged; and an NLI-based evidence-support score $S_E(y_0, E_0)$ is computed. If the correction trigger C_t does not fire, y_0 is returned; otherwise the system runs up to T=3 iterations of recursive, PICO-targeted PubMed retrieval feeding a deterministic controller π, which at each step selects among PRESERVE, RETRIEVE_AGAIN, GENERATE, and a separate COMMIT/REJECT adjudication of any generated candidate. Figure 4 in Appendix H diagrams the full flow; the subsections below detail each stage.

4.1 Query Processing and Hybrid Retrieval

The input question is encoded with a sentence embedding model (`S-PubMedBERT-MS-MARCO`) to obtain query vector z_q. Dense retrieval scores documents by inner product over normalized embeddings using a FAISS `IndexFlatIP` index (dense top-k=20); sparse retrieval scores documents

with Okapi BM25 (Robertson & Zaragoza, 2009) (sparse top-k=20). The two rankings are combined by Reciprocal Rank Fusion (Cormack et al., 2009):

$$\mathrm{RRF}(d; q) = \sum_{m \in \{\text{dense, sparse}\}} \frac{1}{k_{\mathrm{rrf}} + \mathrm{rank}_m(d; q)} \cdot \mathbb{1}[d \in \text{top-}k_m], \tag{1}$$

with $k_{\mathrm{rrf}} = 60$. The fused candidate set is the union of the two top-k lists; the final evidence set E_0 is the top-5 documents by RRF score. The retrieval corpus consists of 272,518 PubMedQA (Jin et al., 2019) artificial and unlabeled abstracts (PQA-A + PQA-U); the labeled evaluation questions (PQA-L) are not indexed.

4.2 Generation and Reflection

Given q and evidence E, a generation model produces an answer $y = G(q, E)$ containing bracketed citations and an explicit terminal decision line (Yes/No/Maybe), followed by a reflection pass that tags spans of y as `<supported>` or `<unsupported>`. The final held-out experiment uses Qwen3.8-27B (recorded model identifier `Qwen/Qwen3.8-27B`; temperature 0.1, top-p 0.9), served with vLLM 0.29.0 in `bfloat16` at a maximum context length of 16,384 tokens on a single A100-80GB GPU.(Appendix A).

4.3 Evidence Scoring

Production evidence scoring is independent of the reflection tags above: every sentence in y is dependency-parsed, and a sentence is retained as a scoreable claim c_i only if it yields a genuine (subject, relation, object) triple with a concrete, non-pronoun subject and object (Section 2.6 in our implementation notes rules out meta-commentary such as "the available evidence suggests…" by this criterion). For each retained claim, a cross-encoder NLI model (`cross-encoder/nli-deberta-v3-base`: a DeBERTa-v3 encoder (He et al., 2023) in a cross-encoder configuration (Reimers & Gurevych, 2019), fine-tuned for natural language inference on SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018)) scores entailment against every evidence sentence $e_j \in E$, and the strongest match is kept:

$$s_i = \max_j p_{\mathrm{entail}}(e_j, c_i), \qquad S_E(y, E) = \frac{1}{N} \sum_{i=1}^{N} s_i, \tag{2}$$

where N is the number of retained claims. This sentence-level-premise scoring is the production configuration; a document-level-premise variant, which concatenates all evidence into a single premise, is retained in our implementation as an ablation but not used for the reported results. We note explicitly that a legacy scoring path exists that scores only reflection-flagged `<unsupported>` spans and defaults to S_E=1.0 when no such span is flagged; an early diagnostic on a single case found this legacy score (1.0) diverging sharply from the independent score in Equation 2 (0.25) on the same answer, motivating our use of the independent, claim-level formulation throughout.

4.4 Correction Trigger

The correction loop is entered and continued while

$$C_t = \mathbb{1}\big[\, S_E(y_t, E_t) < \tau \;\vee\; U_t \,\big], \qquad \tau = 0.85, \tag{3}$$

where U_t is a binary-mode-specific unresolved-polarity indicator (Section 4.6) that is always false outside binary mode. τ is fixed prior to, and not tuned against, the held-out evaluation.

4.5 Recursive Retrieval

When correction is triggered, the system identifies the lowest-scoring retained claim, extracts its missing PICO element (population, intervention, comparator, outcome) when the evidence is judged relevant-but-insufficient, and issues a targeted PubMed E-utilities query. Retrieved passages are reranked by cosine similarity against the query embedding, combined with previously retained evidence, and a novelty indicator distinguishes newly retrieved documents from those already present in E_0. The loop is bounded to T=3 iterations and terminates early on a repeated query, an empty PubMed result set, or once C_t becomes false.

4.6 CORRECTION CONTROLLER (ADJUDICATION)

The controller evaluates evidence relative to a single *affirmative proposition* – the declarative, Yes-form of the question – rather than separately for Yes and No, using a five-way NLI relation (support, contradiction, irrelevant, insufficient, conflicting) that is additionally lexically grounded: a raw NLI contradiction is only treated as contradiction if the evidence contains an explicit negative-result phrase (e.g., "did not improve", "no significant difference"); otherwise it is downgraded to irrelevant. This prevents an evidence passage that simply fails to discuss a claim from being misread as refuting it. Given this grounding, the controller applies the following rules (implemented as a deterministic state machine, not a trained policy):

- **Hedge ban.** A committed Yes/No answer is never changed to Maybe on the basis of low confidence alone; only genuine, evidence-grounded contradiction can justify this transition, and by design our controller does not permit it at all in the current configuration (Yes/No→Maybe is always PRESERVE).
- **Maybe→definite.** A Maybe answer may be resolved to Yes or No only when new evidence is judged *proposition-sufficient* (full relevant PICO coverage, direct answerability, and unambiguous grounded support for one polarity); silence or non-mention in new evidence is explicitly not treated as evidence for either polarity and instead triggers further retrieval.
- **Polarity flip.** A committed Yes is changed to No (or vice versa) only when new, non-refetched evidence both fails to reconfirm the original polarity and evidences the candidate polarity on the affirmative proposition; conflicting evidence triggers further retrieval rather than a flip.

When the retrieval budget is exhausted, the controller's *binary-mode* commitment path (enabled in the final held-out configuration) is *intended* to force a Maybe answer to a Yes/No decision from the available grounded support/contradiction counts rather than leave it unresolved. In the executed held-out run, however, this forced-commitment branch is never reached: across all 150 questions, zero `v5_gated` records terminate via the binary-commitment path, whereas 121 terminate in a preserve/budget-exhausted state in which the controller retains the current label – including Maybe. Consequently 31 of the 150 `v5_gated` answers remain Maybe at termination (Appendix C)

4.7 EXPERIMENTAL CONDITIONS

We evaluate five conditions on identical, frozen initial retrieval E_0 per question (full specification in Table 4): `baseline` (single pass, no recursive loop); `retrieval_off` (recursive loop entered but PubMed retrieval disabled); `ungated_recursive` (recursive loop with the adjudication gate disabled); `v5_gated` (recursive loop with the V5 adjudication controller, Section 4.6); and `unconditional_replacement` (recursive loop that commits every generated candidate). PubMed retrieval, when enabled, is cached deterministically per question and shared across conditions that use it.

5 EXPERIMENTAL SETUP

5.1 DATASET

We evaluate on BioASQ Task B yes/no questions (Tsatsaronis et al., 2015). A held-out sample of n=150 questions was frozen prior to inference (110 Yes, 40 No gold labels) after removing exact and near-duplicate (Jaccard ≥ 0.90) overlap with our retrieval corpus (PQA-A/PQA-U) and with PQA-L (full construction procedure in Appendix B). Runtime inference uses question text only: no gold answers, gold snippets, or PubMedQA gold context are available to the system at any point.

5.2 EVALUATION METRICS AND STATISTICAL TESTING

We report accuracy with Wilson 95% confidence intervals, balanced accuracy, macro-F1, and the multiclass Matthews correlation coefficient (MCC) to account for class imbalance (73.3% Yes). Because all five conditions are evaluated on the same 150 questions, we use paired exact McNemar

Table 1: Primary held-out results ($n = 150$; gold: 110 Yes, 40 No; majority-class baseline: 73.33%).

Condition	Acc.	Wilson 95% CI	Bal. Acc.	Macro-F1	MCC
`baseline`	42.67%	[35.03, 50.67]	0.4739	0.4806	0.2034
`retrieval_off`	42.67%	[35.03, 50.67]	0.4739	0.4806	0.2034
`ungated_recursive`	44.67%	[36.94, 52.66]	0.4955	0.5033	0.2301
`v5_gated`	**51.33%**	[43.40, 59.20]	0.5409	0.5516	0.2739
`unconditional_replacement`	**54.00%**	[46.02, 61.78]	0.5830	0.5793	0.3125
always-Yes (reference)	73.33%	[65.74, 79.76]	0.5000	0.4231	0.0000

Table 2: Paired comparisons. b: first condition correct, second wrong; c: first wrong, second correct.

Comparison	b	c	Δ (pp)	Exact p	χ^2 p
baseline → v5_gated	0	13	+8.67	0.000244	0.000874
baseline → unconditional	0	17	+11.33	0.000015	0.000104
baseline → retrieval_off	0	0	0.00	1.0	—
baseline → ungated	2	5	+2.00	0.453	0.450
v5_gated → unconditional	**0**	**4**	**+2.67**	**0.125**	0.134
v5_gated → ungated	13	3	−6.67	0.021	0.024

tests (with χ^2 approximation reported alongside) on discordant pairs to test differences between conditions, rather than independent-proportion tests.

5.3 Reproducibility Controls

The frozen example set and its ID list are content-addressed with SHA-256 hashes recorded prior to inference. Initial retrieval is computed once per question and shared, byte-identically, across all five conditions; PubMed queries are cached deterministically per question.(Section 6).

6 Results

6.1 Primary Accuracy Results

Table 1 reports accuracy, Wilson 95% confidence intervals, and class-imbalance-robust metrics for all five conditions (n=150; gold: 110 Yes, 40 No). The recursive, evidence-gated condition (`v5_gated`) improves over baseline on every metric reported. For reference, the table's final row reports a trivial always-Yes predictor, which attains 73.33% accuracy on this label distribution – higher than every system condition (we return to this in Section 9); because it is constant, its balanced accuracy is 0.5 and its MCC is 0, so the class-imbalance-robust metrics remain informative where raw accuracy alone would be misleading. A bar-chart rendering of these accuracies with Wilson intervals is provided in Appendix G (Figure 2).

6.2 Paired Correction Analysis

Table 2 reports exact paired McNemar tests between conditions on identical questions. `v5_gated` significantly improves over `baseline` (p=0.000244) and over `retrieval_off` (p=0.000244, identical discordant pairs since `retrieval_off` exactly reproduces `baseline`). The comparison central to this paper's thesis, `v5_gated` versus `unconditional_replacement`, is *not* significant (p=0.125, b=0, c=4): four questions are corrected by unconditional replacement that are not corrected by the gated controller, and zero are corrected by the gated controller but not by unconditional replacement.

Table 3: Correction outcomes relative to baseline.

Comparison	Beneficial	Harmful	Neutral	Precision
baseline → v5_gated	13	0	0	1.000
baseline → unconditional	17	0	0	1.000
baseline → ungated	5	2	1	0.714

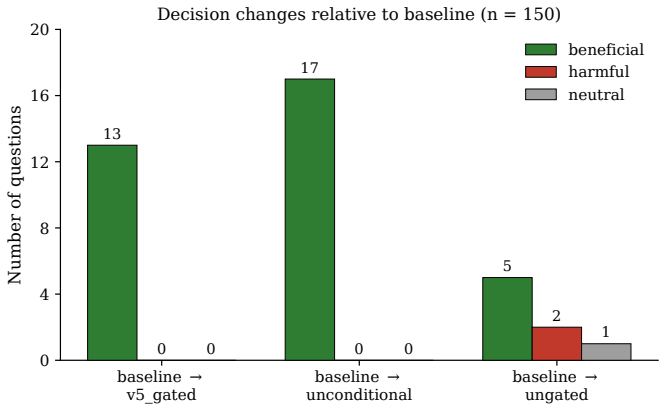


Figure 1: Correction outcomes relative to baseline, corresponding to Table 3. `ungated_recursive` disables adjudication entirely (candidates always committed with no gate), distinct from `unconditional_replacement` which uses the same pre-generation opportunity as `v5_gated` but skips only the commit/reject decision.

6.3 Correction Benefit and Harm

Table 3 decomposes decision changes relative to baseline into beneficial (an initially wrong answer becomes correct), harmful (an initially correct answer becomes wrong), and neutral (a change between two incorrect labels); Figure 1 visualizes this decomposition. Both `v5_gated` and `unconditional_replacement` achieve 100% correction precision on this held-out set (zero harmful changes); `ungated_recursive`, which disables the adjudication step entirely rather than replacing it with unconditional acceptance, produces 2 harmful changes among 8 total, for a precision of 71.4%.

6.4 Retrieval Ablation

`retrieval_off` – identical to `v5_gated` except that recursive PubMed retrieval always returns an empty result set – reproduces `baseline` exactly (64/150 correct, identical confusion matrix; b=0, c=0, exact p=1.0). Since `v5_gated` differs significantly from both `baseline` and `retrieval_off`, the observed improvement is consistent with recursive retrieval, rather than the controller's adjudication logic alone, contributing the gain. We do not interpret this as establishing that recursive retrieval is the sole causal mechanism, nor that retrieval quality is superior to any alternative; it establishes only that disabling retrieval while holding the controller fixed removes the effect.

As a retrieval-quality diagnostic computed post-hoc and never available to the system at inference, we additionally checked whether BioASQ's own gold supporting-document PMID (from each question's `documents`/`snippets` metadata) appears among the five documents in the shared initial retrieval E_0. It does so for 11 of 150 questions (7.33%). We report this only as motivation for why recursive retrieval is frequently necessary on this benchmark under our fixed PQA-A/PQA-U corpus; it is not evidence that recursive retrieval succeeds in recovering the specific gold document, and we did not measure gold-PMID recovery for the recursively retrieved evidence sets.

6.5 Selective Gating versus Unconditional Replacement

This is the central qualifying result of the paper. `unconditional_replacement` reaches 54.00% accuracy, numerically above `v5_gated`'s 51.33%, and the paired difference is not statistically significant (exact p=0.125; Table 2). Stated plainly: the strongest system condition by raw accuracy is unconditional replacement, not the evidence-gated controller `v5_gated`. We do not claim that evidence-gated adjudication is more accurate than unconditional replacement, and this experiment does not establish that it is. We discuss this finding, rather than the paired improvement over baseline, as the paper's most important open question in Section 7.

6.6 Compute Cost

Recursive conditions incur additional model calls and tokens relative to baseline: `v5_gated` and `unconditional_replacement` add comparable token cost (+14.6% and +19.5% over baseline), while `ungated_recursive` is by far the most expensive (+64.0% tokens). Full per-condition counts – calls, tokens, tokens per question, and PubMed query volume – are reported in Appendix E (Table 5), and a scatter of accuracy against mean tokens per question is in Appendix G (Figure 3). Reported figures are wall-clock model-service telemetry, not billing-grade accounting.

6.7 Robustness to Decision-Space Normalization

Because our system emits a three-way label (Yes/No/Maybe) while the BioASQ gold labels are binary, unresolved Maybe answers are scored as incorrect under our primary metric. To confirm the between-condition ordering is not an artifact of this choice, we re-scored every condition under a normalization mapping each terminal Maybe to the majority Yes class. The ordering (`unconditional_replacement` > `v5_gated` > `ungated_recursive` > `baseline`) is preserved, so our qualitative findings do not depend on the treatment of Maybe; full per-condition numbers and the confound this normalization introduces (it rewards leaving Maybe unresolved, given the 73.33% Yes prior) are reported in Appendix F. We retain the stricter three-way scoring for all primary results.

7 Analysis

7.1 Why Does Selective Gating Not Outperform Unconditional Replacement?

Both approaches made zero harmful corrections on this held-out set (Table 3), so the gated controller is not trading accuracy to avoid harm that unconditional replacement was actually committing. The four discordant questions (last row of Table 2; `v5_gated` wrong, `unconditional` correct) are therefore missed beneficial corrections – candidates the controller declined that would in fact have been correct, not harmful ones it avoided. This suggests our adjudication criteria (Section 4.6), designed conservatively to avoid unsupported polarity flips and hedge-driven degradation, may be stricter than necessary for this benchmark. We did not find a decision rule that recovers these four cases without weakening the conservatism responsible for the zero-harm record, and leave this as an open problem rather than proposing an untested fix.

7.2 When Does Recursive Correction Help?

The retrieval-off ablation (Section 6) indicates that recursive retrieval, not adjudication logic alone, is necessary for the gain over baseline. A complementary diagnostic on a different dataset (PQA-L, with each question's own source abstract supplied directly as evidence rather than retrieved) found baseline accuracy of 74.5% and a correction-added gain of only 1.5 points (p=0.25), far smaller than the gain observed here. Because this diagnostic uses a different dataset and evidence-provision protocol and is not a controlled ablation on the same inputs.

7.3 Controller Behavior

Across the 150 held-out questions, `v5_gated` most often selects PRESERVE or RETRIEVE_AGAIN over COMMIT: correction is triggered on the large majority of questions (baseline mean S_E=0.2853,

well below τ=0.85), yet only 13 committed label changes result. This matches the controller's conservative design (Section 4.6), which requires proposition-level sufficiency or grounded contradiction before committing, and explains both its zero-harm record and its low commit rate relative to unconditional replacement. Consistent with that executed behavior, 121 of the 150 records terminate in a preserve/budget-exhausted state, none via the unreached binary-commitment branch, and 31 answers remain at a Maybe label (Appendix C).

8 DISCUSSION

The central result is a controlled comparison, not a single accuracy number. Recursive, retrieval-driven correction reliably improves paired accuracy over a one-pass baseline, but making that correction *selective* – adding an adjudication step that separates candidate generation from commitment – does not demonstrate an accuracy advantage over committing every candidate. We read this as a genuine empirical negative finding about controller design, not as evidence that gating cannot help: on this benchmark and controller, gating forgoes more beneficial corrections than the near-zero harm it averts, and we do not over-generalize that mechanism beyond this setting.

9 LIMITATIONS

- **Absolute accuracy.** All evaluated conditions remain below the 73.33% majority-class (always-Yes) baseline; we claim no competitive absolute performance, and all comparisons are paired, on identical inputs, within one system.
- **Selective gating is not established as superior.** `v5_gated` does not significantly outperform `unconditional_replacement` (p=0.125); we treat this as an open question, not a settled result in either direction.
- **Sample size and scope.** The primary evaluation uses n=150 questions from a single benchmark (BioASQ yes/no) and a single generator (Qwen3.8-27B); we do not establish generalization across benchmarks, models, or answer-type distributions.
- **Three-way decisions on a binary benchmark.** Our system emits Yes/No/Maybe while BioASQ gold labels are binary, so unresolved Maybe answers are scored as incorrect. A Maybe→Yes normalization preserves the between-condition ordering but is confounded by the 73.33% Yes prior (Section 6.7); absolute accuracy under either scoring should be read with this in mind.
- **Evaluation chronology.** An earlier two-condition "final-test" run predates the held-out evaluation (Appendix D); it used different serving infrastructure and a different condition set, is not comparable to or merged with these results, and was not used to select or tune anything reported here.

10 CONCLUSION

We studied recursive correction in retrieval-augmented scientific question answering as an evidence-gated control problem that separates candidate generation from commitment. On a frozen, question-only BioASQ held-out set, recursive correction significantly improves paired accuracy over a one-pass baseline (42.67% → 51.33%), and a retrieval-off ablation indicates that recursive retrieval contributes to this gain. An unconditional-replacement variant that omits the adjudication step nonetheless reaches numerically higher accuracy (54.00%), so this experiment does not establish an accuracy advantage for selective gating. We present this as an open finding about correction-controller design and hope it motivates further controlled study of when selective correction earns its added complexity.

REPRODUCIBILITY STATEMENT

The frozen held-out example set (150 questions; 110 Yes, 40 No) and its ID list are content-addressed with recorded SHA-256 hashes prior to inference, together with hashes of the retrieval

configuration, generation and reflection prompt templates, the evidence-scoring evaluator, and the evaluation runner script, all listed in Appendix A. Initial retrieval is computed once per question and shared identically across all five conditions; PubMed queries are cached deterministically per question. We disclose two provenance issues transparently rather than correcting them retroactively: (1) the recorded run manifest's `tau` field is `null` because the manifest generator referenced an incorrect configuration key path (`fact_checking.tau` instead of the runtime-authoritative `thresholds.tau`); the correction-entry threshold actually used at runtime, τ=0.85, is independently confirmed via the recorded configuration file hash and is consistent throughout our implementation and prior development runs; (2) the final held-out run was executed from a Git worktree containing uncommitted local changes, so no Git commit hash is available for this run, and we do not construct or approximate one. We anchor reproducibility instead to the dataset, configuration, prompt, evaluator, and runner hashes listed in Appendix A, which were independently re-verified against the raw per-question output records rather than taken from any summary report. A chronology of earlier evaluation runs conducted during development – including a prior two-condition "final-test" run that predates the held-out evaluation reported here – is disclosed in Appendix D for completeness; those runs use different serving infrastructure and a different condition set, are not comparable to the held-out results, and are neither merged with nor used to tune them.

Large Language Model (LLM) Usage Statement

In accordance with the ICLR 2027 policy on LLM usage, we disclose the ways in which large language model assistants were used in this work. LLMs served as a general-purpose development and writing aid across: (i) research ideation and literature discovery, where candidate related work surfaced by an assistant was subsequently verified by the authors against primary sources; (ii) implementation and debugging of the retrieval, evidence-scoring, and correction-controller code; (iii) design and execution of the ablation and held-out evaluation protocol; (iv) statistical and mechanism-level analysis of results; and (v) drafting and editing of this manuscript, including its mathematical formalization, which was checked against the underlying source code and configuration files rather than accepted from model-generated descriptions alone. LLMs are not authors of this paper and are not eligible for authorship. All reported experimental results were produced by the described pipeline and evaluation scripts; no result, statistic, table entry, or citation in this paper was generated, estimated, or fabricated by an LLM. The authors have reviewed all AI-assisted content, independently verified every quantitative claim against the frozen output records, and take full responsibility for the entire content of this paper, including any remaining errors.

Ethics Statement

This work uses only publicly available benchmark data (PubMedQA, BioASQ) and does not involve human subjects, personally identifiable information, or clinical deployment. The system is a research prototype evaluated on benchmark question answering and should not be used to inform actual clinical or scientific decisions without independent verification; its accuracy on this benchmark, in all evaluated conditions, is below a trivial majority-class baseline.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024.

Samy Ateia and Udo Kruschwitz. Can language models critique themselves? Investigating self-feedback for retrieval augmented generation at BioASQ 2025. *arXiv preprint arXiv:2508.05366*, 2025.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2015.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms Condorcet and individual rank learning methods. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2009.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, 2024.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *International Conference on Learning Representations*, 2023.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations*, 2024.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2024.

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 the Conference on Empirical Methods in Natural Language Processing*, 2023.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2019.

Hailey Joren, Jianyi Zhang, Chun-Sung Ferng, Da-Cheng Juan, Ankur Taly, and Cyrus Rashtchian. Sufficient context: A new lens on retrieval augmented generation systems. In *International Conference on Learning Representations*, 2025.

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.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2023.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2019.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

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

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, 2015.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2018.

Guangzhi Xiong, Qiao Jin, Xiao Wang, Minjia Zhang, Zhiyong Lu, and Aidong Zhang. Improving retrieval-augmented generation in medicine with iterative follow-up questions. *arXiv preprint arXiv:2408.00727*, 2024.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

A Reproducibility Hashes

Hashes below are recorded exactly as generated by the evaluation runner at execution time (SHA-256 unless noted), independently re-verified against the raw per-question output records for this submission rather than taken from any summary report. As disclosed in Section 10, no Git commit hash is available for this run.

Artifact	Hash (SHA-256)
Configuration and source	
`config.yaml`	`76f912698864aa078912955ac0d609f2eae4032458ce2c08df6db2ea601d6563`
`src/phase2_retrieval.py`	`69bdcc3ab97ab299ce6c4fd4b7ebee83343f047dbdd3f813ba114c9569befffc`
`src/phase2_generation.py`	`65df329d9c4aa39e2d4a55160bec04b9c3bd51e8ae5702f599791c42d1fb4b90`
`src/phase2_factchecking.py`	`7c6ce5f5e99f2b82b2922dc91c80307eee5873f3e0324f7e66e87bd4f834573b`
`src/phase3_selfcorrection.py`	`3e20afc3ec6a19b12b32a1a6acbef770163d1f6026f66fedfbb62c2f092dc54f`
`src/correction_controller_v5.py`	`1b66fe7f1c19361caf076c1e1da682ef2629a76ac8a9f87ff04ec441c611dd9c`
`src/ablation_runner.py`	`903e7f5b35d5f40385415ed2c9fb9c90638f460196bb2660f34f109d8a7c6c32`
`scripts/evaluate_bioasq_ablation.py`	`649c3a80e10f4bb6df2f9ffbe55ccb82a955a160c59aeaa2f5daab3b44dbbdfd`
Prompt templates	
`src/prompts/generation.txt`	`ea49a73334807ff2df0cdbfed2858fc0b420275283c4c85e6f42c935f2474b5d`
`src/prompts/reflection.txt`	`5eaddfc8c8848608e1a8f7541e46c1fc418d06a83785acaae88dd1fa34b09cdc`
`src/prompts/decision.txt`	`bb9c32b16642d55aa4e2d666f3d02ee00d7dc8c9d1a2e9120a9b4d8799a5a3b1`
Frozen held-out data	
frozen examples (JSON)	`42fcf03b90c605957db190a7f8889fcaea13c57c21f7d126f86f7c6be552336d`
frozen example IDs	`8ef0e1706e7a118fdb59fad6478a9676113ba2ede9643b83f97326b30617be0a`

B Held-Out Set Construction

The 150-question held-out sample was drawn from a candidate pool of BioASQ Task B yes/no questions after sequential filtering: retaining yes/no-labeled items with valid IDs and labels; removing duplicate normalized questions and duplicate IDs; removing exact overlap with PQA-A/PQA-U/PQA-L; removing near-duplicates at Jaccard similarity ≥ 0.90 against PQA-A/PQA-U/PQA-L and against previously used development or final-test splits from earlier stages of this project; and sampling with a fixed random seed, stratified to approximate the pool's overall label distribution. The resulting label distribution (110 Yes, 40 No) reflects this filtered pool rather than an independently chosen target ratio.

C Additional Confusion Matrix Detail

Condition	Gold	Pred. Yes	Pred. No	Pred. Maybe
baseline / retrieval_off	Yes (110)	41	47	22
	No (40)	0	23	17
ungated_recursive	Yes (110)	43	44	23
	No (40)	0	24	16
v5_gated	Yes (110)	53	42	15
	No (40)	0	24	16
unconditional_replacement	Yes (110)	54	41	15
	No (40)	0	27	13

No condition ever predicts Yes on a gold-No question; all committed errors on No-questions are No→Maybe.

D Prior Evaluation Chronology

For full transparency about the project's evaluation history, we disclose an earlier evaluation run that predates the frozen held-out study reported in the main text. We include it because it shares the SelfHeal-RAG codebase lineage and could otherwise appear, undisclosed, in the development repository; it is *not* part of our reported evidence, and none of our claims rest on it.

What the earlier run was. An internal run labeled "final-test" (random seed `202609102`) was completed on 2026-09-15, prior to the held-out evaluation (seed `202609103`) reported here. It evaluated two conditions only – `baseline` and a `full_system` configuration – on a 150-question sample, rather than the five conditions evaluated on the frozen held-out set.

Why it is not comparable and not merged. The two runs differ on several axes simultaneously, so no valid comparison can be drawn between them:

- **Disjoint questions.** The seed-`202609102` question set and the frozen held-out (seed-`202609103`) set have zero overlap; they are different samples.
- **Different serving infrastructure.** The earlier run was served through a different inference provider (Cerebras) than the held-out run (vLLM on A100-80GB), so the generator is not held fixed across the two.
- **Different condition set.** Two conditions versus five, without the retrieval-off, ungated-recursive, or unconditional-replacement ablations that support the main-text analysis.
- **Different terminal behavior.** In the earlier run the `full_system` condition emits zero Maybe answers, consistent with a binary-commitment terminal path being exercised; on the frozen held-out run, as reported in Section 4.6, that path is not reached and 31 Maybe answers remain. The two runs therefore do not exhibit the same controller behavior.

Numbers, for reference only. For completeness, the earlier run recorded accuracies of 36.0% (`baseline`) and 70.0% (`full_system`) on its own 150-question sample. We report these two numbers only so that their existence is documented; because of the differences above, they are not comparable to the held-out results in Table 1, are not merged with them, and were not used to select, tune, or justify any threshold, condition, or design decision in the held-out evaluation. We make no claim about the intent behind any code difference between the two runs; we report only the observable facts of the runs themselves.

E Compute Cost Detail

Table 4 specifies the five experimental conditions; Table 5 reports their full per-condition compute for the final held-out run, summarized in Section 6. Figures are wall-clock model-service telemetry, not billing-grade accounting, and are affected by per-condition caching (notably shared PubMed caching across conditions that use it).

F Decision-Space Normalization Detail

This appendix expands the robustness check summarized in Section 6.7. Table 6 reports raw and Maybe→Yes-normalized accuracy side by side for all five conditions.

Two points follow. First, the ordering across conditions – `unconditional_replacement` > `v5_gated` > `ungated_recursive` > `baseline` – is preserved under normalization, so our qualitative findings (recursive correction improves over baseline; selective gating does not overtake unconditional replacement) do not depend on the treatment of Maybe. Second, this normalization is a confounded measure of decision quality: because 73.33% of gold labels are Yes, mapping Maybe to Yes rewards a condition in proportion to how many Maybe answers it leaves unresolved, and

Table 4: Experimental conditions. All conditions share one frozen initial retrieval E_0 per question; PubMed retrieval, when enabled, is cached deterministically per question and shared across conditions that use it.

Condition	Recursive loop	Adjudication
`baseline`	No	—
`retrieval_off`	Loop entered, PubMed disabled	Controller (V5)
`ungated_recursive`	Yes	None (gate disabled)
`v5_gated`	Yes	Controller (V5, Section 4.6)
`unconditional_replacement`	Yes	Every candidate committed

Table 5: Compute cost, final held-out run.

Condition	Calls	Tokens	Tok./Q	PubMed calls	vs. baseline (calls)
baseline	321	623,312	4,155	0	—
retrieval_off	321	623,312	4,155	0	+0.0%
ungated_recursive	601	1,022,410	6,816	81	+87.2%
v5_gated	394	714,165	4,761	241	+22.7%
unconditional_replacement	403	744,676	4,965	7	+25.5%

cannot exceed the trivial always-Yes ceiling of 73.33% by this route. We therefore report normalized accuracy only as a robustness check on the between-condition ordering, not as an alternative headline metric, and retain the stricter three-way scoring for all primary results.

G Supplementary Figures

The two figures below visualize data already tabulated in the main text (Tables 1 and 5); they are placed here to respect the main-text page limit and carry no information beyond those tables. They are regenerated directly from the frozen per-question records by `figures/make_figures.py`, which recomputes every plotted value and asserts equality with the reported table entries before drawing.

H Pipeline Schematic

Figure 4 diagrams the full SelfHeal-RAG pipeline described textually in Section 4. The controller step (bold) is repeated for up to $T=3$ iterations, distinguishing initial generation (single pass, always executed) from candidate generation (conditional on the controller's decision and subject to a separate commit/reject adjudication).

Table 6: Accuracy under decision-space normalization (terminal Maybe → Yes). Normalization raises every condition, but the between-condition ordering is preserved.

Condition	Raw acc.	Maybe→Yes acc.
`baseline`	42.67%	57.33%
`retrieval_off`	42.67%	57.33%
`ungated_recursive`	44.67%	60.00%
`v5_gated`	51.33%	61.33%
`unconditional_replacement`	54.00%	64.00%

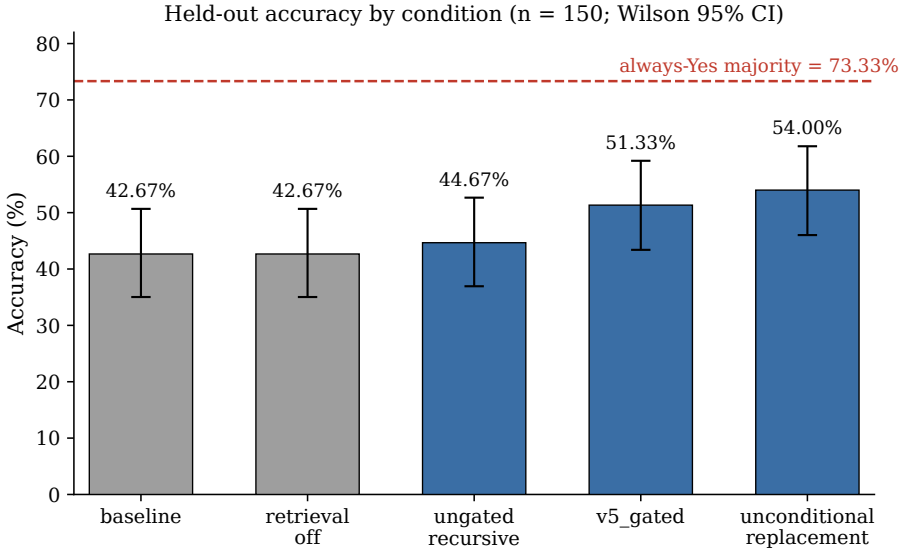


Figure 2: Held-out accuracy by condition, corresponding to Table 1. Recursive conditions (colored) are compared against the non-recursive baseline and its retrieval-off replica (grey). The dashed line marks the 73.33% majority-class baseline.

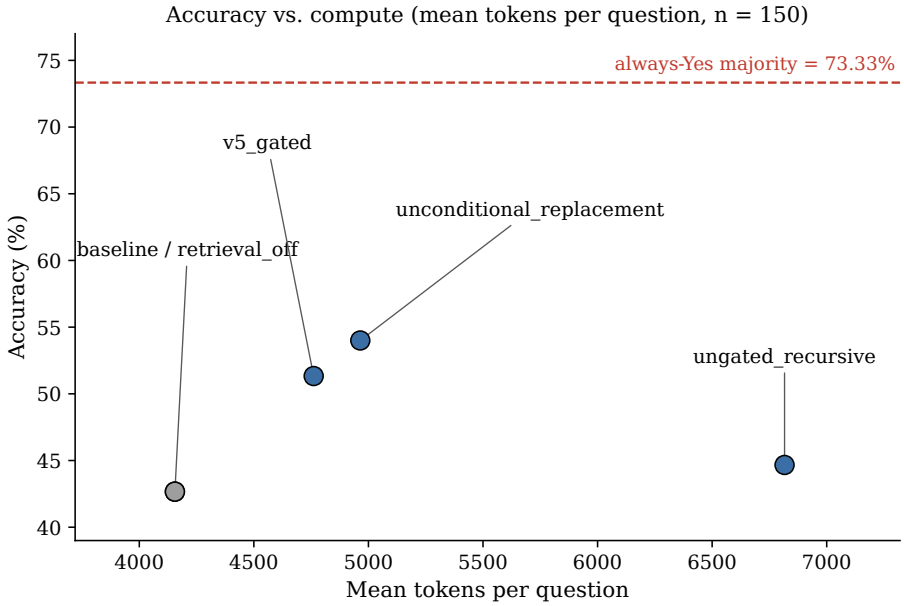


Figure 3: Accuracy versus mean tokens per question, corresponding to Table 5. `baseline` and `retrieval_off` share a single marker because `retrieval_off` reproduces `baseline` exactly, at identical accuracy and cost. `unconditional_replacement` attains higher accuracy than `v5_gated` at comparable additional cost over baseline; `ungated_recursive` incurs the largest cost increase without a significant accuracy gain over baseline (Table 2).

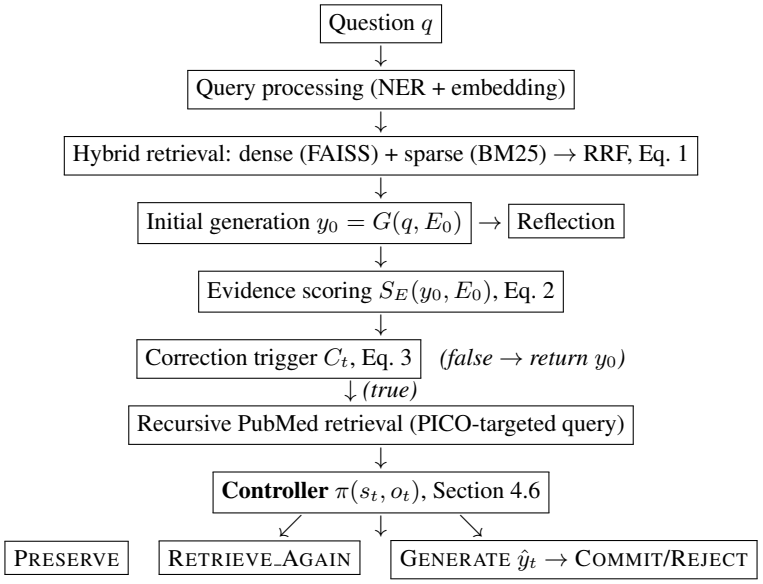


Figure 4: SelfHeal-RAG pipeline (moved from the main text to respect the page limit; referenced in Section 4).