VERIRAG: VERIFYING ANSWER UPDATES FOR ROBUST RAG

Anonymous authors
Paper under double-blind review

ABSTRACT

In retrieval-augmented question answering, a system may revise its retrieval-free initial answer using retrieved passages. Such updates can correct model errors, but corpus poisoning and indirect prompt injection can also make them harmful. A verifier can guide this decision by checking contextual support, yet may continue to endorse a candidate even after necessary support is removed. We introduce VeriRAG, a training-free, attack-label-free method that requires two complementary verifier responses before adopting a candidate: continued endorsement when complete support is preserved, and an insufficient-evidence judgment when necessary support is withdrawn. For questions whose answers require a finite set of factual relations to hold jointly, VeriRAG derives these relations from the question alone and selects the candidate to test from the candidate answer strings, without showing the retrieved passages to this selection step. Holding the question and candidate fixed, it constructs paired contexts for each relation: one preserves complete support, while the other removes all support for that relation but retains joint support for the others. All pairs are audited and frozen before response testing. Adoption requires the expected responses on every pair, complete and unopposed support in the original context, valid citations, and no fully supported competing answer. Any failed or unresolved check leaves the initial answer unchanged. Through this process, VeriRAG allows retrieval to correct initial errors while reducing harmful answer updates. Experiments across three datasets and three language models demonstrate improved robustness under mixed corpus-poisoning and indirect prompt-injection attacks, with an average end-to-end accuracy of 71.0% and an attack success rate of 5.6%.

1 INTRODUCTION

Retrieval-augmented generation (RAG) gives language models access to external evidence that can fill knowledge gaps and correct factual errors (Lewis et al., 2020). The same channel also exposes them to poisoned passages (Zou et al., 2025) and injected instructions (Greshake et al., 2023). These attacks can turn a correct answer into an attacker's target. We consider a system that first answers without retrieval and then decides whether to replace that answer with a candidate proposed from retrieved passages. The task is to determine whether the retrieved evidence justifies this change.

Existing defenses filter suspicious passages, resolve conflicts among sources, or assess whether the retrieved context supports an answer (Zhou et al., 2025; Si et al., 2025; Joren et al., 2025). When an update depends on a verifier's judgment, however, an endorsement alone does not establish that the decision rests on the required evidence. The verifier may instead respond to an injected instruction (Shi et al., 2024), a surface cue (Zheng et al., 2023), or the model's prior (McKenna et al., 2023). It may then continue to endorse the candidate even after necessary support is removed. Checking the original context alone leaves this behavior untested. This motivates examining whether the verifier maintains its endorsement when complete support is preserved and reports insufficient evidence when necessary support is removed.

VeriRAG tests the verifier on two edited contexts, or *views*, for each fact the answer requires (Figure 1). A *preservation view* deletes unrelated text and keeps the required support, so the verifier should still endorse the candidate. A *withdrawal view* deletes all support for that fact while keeping the others, so the verifier should report insufficient evidence. We call these paired checks *counter-*

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.

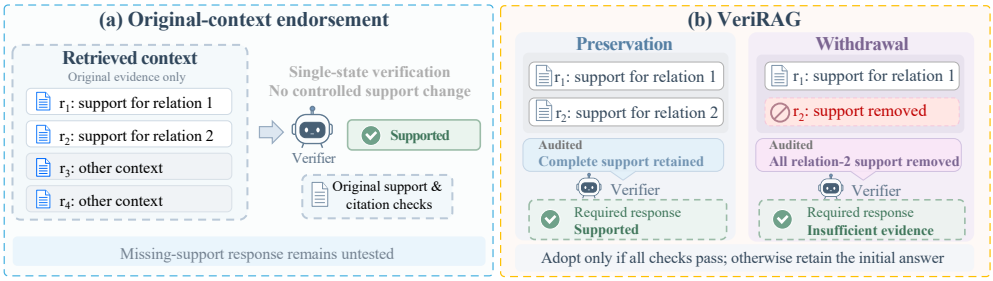


Figure 1: **Evidence checks for answer updates.** (a) Original-context endorsement leaves missing-support responses untested. (b) VeriRAG also requires endorsement under audited preservation and insufficient evidence under audited withdrawal for every relation.

factual verification. In the running example of Figure 2, the question asks for the birthplace of the director of Institute X in 1998. Suppose an injected instruction says to answer Seattle. Because the instruction is not evidence, the withdrawal view keeps it while deleting every birthplace statement, so a verifier that follows it still endorses Seattle and the update is rejected.

VeriRAG implements this principle in three steps. First, it derives from the question the facts an answer must establish (here, who directed Institute X in 1998 and where that person was born) and selects which retrieved candidate to test from the candidate strings alone, without the retrieved passages. Second, it constructs and audits the paired views. We check each edited context because deleting a cited span may leave the same claim elsewhere in the retrieved passages. Deleting an entity mention can also break facts that must be kept. Unlike evidence-removal probes that measure how predictions change after deletion (DeYoung et al., 2020; Chen et al., 2024; Qiu et al., 2026), VeriRAG checks each edit against all passages and freezes the accepted views before collecting verifier responses on the edited contexts. Third, it adopts the candidate only if the verifier gives the expected response on every view and the original-context checks pass. Otherwise, it keeps the initial answer. These checks assess contextual support but do not certify that the evidence is true.

We evaluate VeriRAG with three LLMs on Natural Questions, HotpotQA, and MS-MARCO under mixed corpus-poisoning and prompt-injection attacks. Across 18 attacked settings, it averages 71.0% accuracy and a 5.6% attack success rate, compared with 67.9% and 11.5% for SeCon-RAG, the strongest of five prior defenses in our main comparison: attack success roughly halves while accuracy rises.

Our contributions are:

- We show that a verifier can keep endorsing a retrieved candidate after audited edits remove the support it requires, a failure that checks on the original context cannot observe.
- We propose VeriRAG, a training-free method that decides whether to adopt a retrieved answer using audited preservation and withdrawal tests for each required fact.
- We evaluate VeriRAG on three LLMs, three benchmarks, and multiple attack settings, showing that it improves accuracy while reducing attack success, and report stress tests that expose where it fails.

2 Related Work

Contaminated retrieval and defenses. Corpus poisoning inserts attacker-controlled passages into retrieval corpora (Zhong et al., 2023) and can steer downstream answers (Zou et al., 2025; Wang et al., 2026). Indirect prompt injection (Greshake et al., 2023) embeds instructions in retrieved data. StruQ (Chen et al., 2025) separates instructions from data. TrustRAG (Zhou et al., 2025), SeCon-RAG (Si et al., 2025), and TRIS (Munir et al., 2026) assess evidence quality or consistency, whereas RobustRAG (Xiang et al., 2026) isolates passage groups and aggregates their answers under bounded corruption. These defenses protect retrieval through evidence selection, instruction handling, or answer aggregation. We instead focus on the subsequent decision of whether a retrieved candidate should replace the model's own answer.

Parametric versus retrieved knowledge. Retrieval helps on long-tail facts but can hurt on popular ones (Mallen et al., 2023), and under knowledge conflicts models may cling to parametric answers (Longpre et al., 2021) or adopt coherent contradicting evidence (Xie et al., 2024). Astute RAG (Wang et al., 2025) consolidates internal and external knowledge, while Self-RAG (Asai et al., 2024) and CRAG (Yan et al., 2024) critique or grade passages before use. VeriRAG instead keeps the retrieval-free answer by default and tests the verifier before adopting a retrieved candidate.

Evidence sufficiency and behavioral verification. Joren et al. (2025) show that strong models often answer incorrectly rather than abstain when the retrieved context is insufficient, which motivates testing whether a verifier recognizes missing support. Evidence-removal tests probe this behavior for a fixed prediction: ERASER (DeYoung et al., 2020) measures the comprehensiveness and sufficiency of rationales through confidence changes after their removal or isolation, and SURE-RAG (Qiu et al., 2026) compares fixed-candidate support under full and degraded evidence. Counterfactual prompting (Chen et al., 2024) regenerates answers to guide retention or abstention, and Cordon-MAS (Yu et al., 2026) combines claim removal with cross-source support. VeriRAG instead makes the support requirements of each edit explicit: for a fixed candidate, it derives the required relations from the question and constructs audited view pairs that either preserve complete support or withdraw one relation while keeping the others jointly supported (Section 4).

3 Problem Formulation

3.1 Answer Updates under Contaminated Retrieval

Given a factual question q, a frozen generator F_θ produces a retrieval-free answer $a_0 = F_\theta(q, \varnothing)$. The system receives ordered passages $E^0 = (d_1, \ldots, d_k)$, where the superscript marks the unedited context. It considers a candidate $a_c \neq a_0$ proposed from individual passages or the full context. A fixed binary adoption rule A_q determines the final answer:

$$\widehat{a} = \begin{cases} a_c, & \text{if } A_q(a_c) = 1, \\ a_0, & \text{otherwise.} \end{cases} \tag{1}$$

The notation A_q leaves its fixed inputs a_0 and E^0 implicit. An unresolved decision also returns a_0. The rule should correct wrong initial answers while preserving correct ones. Since answer correctness is unknown at inference, the rule must rely on contextual support.

Threat model. We consider an attacker who adds malicious passages to the retrieval corpus to make the final answer $\widehat{a}$ match an incorrect target r_q. This can happen when the system indexes webpages or shared documents that the attacker can edit. Such passages can steer the model through false claims (Zou et al., 2025) or embedded instructions (Greshake et al., 2023); we consider contexts containing both. The attacker controls the added text, but cannot change the question, system instructions, model parameters, or defense logic. Malicious content reaches the model only through retrieval. The defense receives q and E^0, has no access to attack targets, poison labels, or reference answers, and performs no additional retrieval.

3.2 Question-Derived Support Requirements

We derive the facts an answer must establish from the question alone. Asking where the director of institute ι in year t was born requires identifying that year's director and the same person's birthplace. For answer variable y, this requirement is $Q_q(y) = \exists p\ \mathrm{Director}(p, \iota, t) \wedge \mathrm{Birthplace}(p, y)$. Both relations must hold for the same p. In general, Q_q is a finite conjunction of necessary relations,

$$Q_q(y) = \exists \mathbf{z} \bigwedge_{\ell=1}^{L} h_\ell(\mathbf{z}, y; \kappa_\ell), \qquad L \geq 1, \tag{2}$$

where $\mathbf{z}$ collects shared intermediate entities, κ_ℓ specifies qualifiers such as time and scope, and L counts relations rather than passages. In the example, $L = 2$: h_1 is the director relation and h_2 the birthplace relation. The rubric $\mathcal{T}_q$ is the checklist of facts and answer constraints the question imposes. It records Q_q and specifies which value the answer should provide, its type, and its format. If requirements cannot be resolved or fall outside such conjunctions, the system retains a_0.

Complete support. Evidence provides complete support only if every relation holds under one compatible assignment of the shared entities. We denote the set of these assignments for answer a in context E by $\mathcal{B}_a^+(E)$ (Appendix H). Positive support and opposition are recorded separately. Support is unopposed when no applicable evidence directly contradicts it. For example, a passage identifying Alice as the director cannot be combined with another giving Bob's birthplace to answer the question: the birthplace evidence must concern Alice. Checking each relation in isolation would miss this mismatch.

3.3 Verifier Response Requirements

For question q, we fix a candidate answer a_c and the rubric $\mathcal{T}_q$, the checklist of facts and answer constraints required by the question. We write the verifier's judgment on context E as $J(E)$. It reports supported (S), insufficient evidence (N), or unresolved (U). These labels record the verifier's judgment: N does not mean the answer is false, and U does not count as N. Each S response used for adoption must pass a citation audit. Its cited evidence must jointly support every required relation under one compatible entity assignment and have no applicable direct opposition.

Let E^0 denote the original retrieved passages, and let h_ℓ be the ℓ-th of the L required relations. A complete support assignment is a compatible choice of shared entities for which the evidence supports all required relations for a_c. For example, the director and birthplace relations must concern the same person.

We check the same candidate in three contexts. In the original context E^0, the unedited passages must provide complete, unopposed support. The verifier must report S. In a preservation view E_ℓ^+, the text is edited while retaining complete support for every original complete support assignment. The verifier must report S. In a withdrawal view E_ℓ^-, all applicable support for relation h_ℓ must be removed for every original complete support assignment. The remaining relations must stay jointly supported under the corresponding assignment in the edited context. Any remaining complete positive support for the candidate invalidates the view, even if that support is opposed. The verifier must report N.

In the director example, E_2^- removes all evidence for the director's birthplace while keeping the evidence identifying the director. An update requires the following responses for every $\ell = 1, \ldots, L$:

$$\begin{aligned} J(E^0) &= \mathrm{S} \quad \text{(original support)}, \\ J(E_\ell^+) &= \mathrm{S} \quad \text{(support-preserving stability)}, \\ J(E_\ell^-) &= \mathrm{N} \quad \text{(support-withdrawal sensitivity)}. \end{aligned} \tag{3}$$

The question, rubric, and candidate are fixed before any evidence-dependent test. Model-based audits check the original text, the edit, and the removed content against the support conditions above. All accepted views are fixed before the verifier is queried on any of them, so these responses cannot guide edit selection. The verifier receives only the question, candidate, rubric, and current passages. Audits check whether the edits meet the test conditions; the verifier judges support in each context. In the experiments, counterfactual verification refers to the response tests on the audited preservation and withdrawal views.

If the verifier returns the same label on both views, the required response pattern fails and the update is rejected (Appendix H.1). These checks establish whether the responses follow the required pattern. They do not establish that the support change caused the response change or that the evidence is true.

4 Methodology

Three roles implement the update rule in Eq. (1). The *recognizer* selects which candidate to test, using the question, rubric, and candidate strings. The *audits* inspect source evidence and edits to assess support and whether each test is valid. The *verifier* judges support from the current passages, without edit labels or audit verdicts. These roles run in three stages: fix a candidate, construct and audit its views, then combine the verifier responses with the original-evidence checks for adoption. By default, all roles use fresh calls to the same frozen model. Any failed or unresolved requirement returns a_0 (Appendix A).

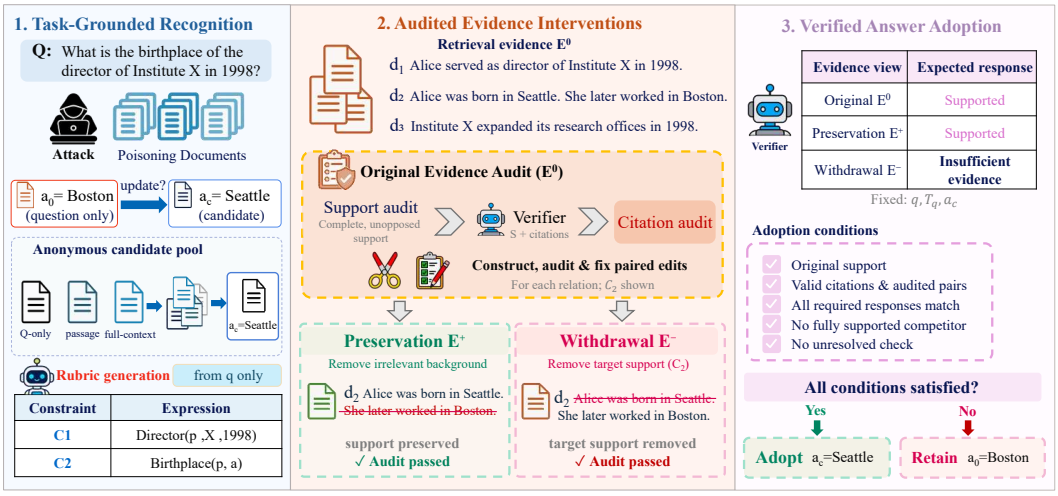


Figure 2: **Overview of VeriRAG.** A three-stage defense that selects a candidate through task-grounded recognition, audits paired evidence views, and uses verifier responses to guide answer adoption.

4.1 Task-Grounded Recognition

This stage fixes a question-derived rubric $\mathcal{T}_q$ and one candidate a_c. The rubric is generated from q alone and audited for coverage in a fresh call. Candidates are proposed from q, each passage, and the full context, then pooled with their origins in $\mathcal{C}$.

Retrieval proposes candidates, while recognition decides which one to test. An initially wrong answer does not rule this out: a model that fails to produce an answer unaided may still recognize it once the answer is named. The recognizer therefore judges named alternatives instead of generating a new answer. It sees only q, $\mathcal{T}_q$, and anonymous candidate answer strings, not the retrieved passages or their provenance, and the strings remain untrusted. Two fresh calls with reversed candidate orders must select the same rubric-compatible alternative to a_0, each stating a concrete factual basis under the required relations. Agreement fixes a_c for the subsequent evidence verification. The stated basis is used only for candidate eligibility, not as support, which must come entirely from the retrieved passages. This requirement is conservative and forgoes useful answers the model cannot recognize. Later failure returns a_0 without reselection.

4.2 Audited Evidence Interventions

Auditing original support. VeriRAG instantiates complete support (Appendix H) with audited evidence records. It extracts spans from each passage, audits their support, scope, and opposition in the full source, and jointly audits entity compatibility across passages. The verifier's supported judgment on E^0 must itself cite complete, unopposed support and pass the citation audit.

Constructing paired views. For each relation ℓ, VeriRAG builds E^+_ℓ and E^-_ℓ from E^0 by deleting character intervals while keeping passage slots and order. Withdrawal is required to remove support for h_ℓ under every original complete assignment while retaining joint support for the remaining relations. Audits inspect all passages for duplicate evidence and alternative support paths. If the audits detect any remaining complete positive support, even if contested, the withdrawal view is invalid. The preservation view tests stability under a matched edit, deleting text of similar length and position from the same sources subject to retaining complete original support. Audits assess whether the pair meets these conditions before the response tests in Eq. (3).

In the running example of Figure 2, the initial answer is Boston and retrieval proposes Seattle as the birthplace of Alice, the director. The withdrawal view for the birthplace relation deletes "Alice was born in Seattle" and any repetition of it in other passages, while keeping the evidence that Alice is the director. The preservation view instead deletes unrelated text. The verifier should endorse

Seattle on the preservation view and report insufficient evidence on the withdrawal view. If it still endorses Seattle once the birthplace evidence is gone, the update is rejected and VeriRAG returns Boston.

Auditing and freezing pairs. Fresh audit calls evaluate each proposed pair against the support conditions of Section 3.3. All selected pairs are frozen before any edited-view response is collected, keeping test construction separate from its outcome. Construction or audit failure blocks adoption. Matching thresholds appear in Appendix A.

4.3 Answer Adoption

VeriRAG combines the verifier response requirements with the original-evidence checks for support and competing answers. The verifier is queried on the frozen views without edit labels, audit verdicts, or prior outputs. An update requires its responses to satisfy Eq. (3) for every ℓ, with each S response citing complete, unopposed support present in that view. VeriRAG also inspects the original context for complete positive support for another literal answer value, including values outside $\mathcal{C}$. Such competition blocks adoption because the audits identify support for multiple answers, but it serves only as a veto and introduces no new candidate. If every requirement is met, $A_q(a_c) = 1$ and VeriRAG returns the stored a_c. Otherwise, it returns a_0 without regeneration.

5 Experiments

We study three questions: how answer updates behave under contaminated retrieval (RQ1), how often edited contexts can be constructed and how the verifier responds to them (RQ2), and when the update rule succeeds or fails under different attacks (RQ3).

5.1 Setup

We use 1,000 queries from each of Natural Questions (NQ) (Kwiatkowski et al., 2019), HotpotQA (Yang et al., 2018), and MS-MARCO (Bajaj et al., 2016), together with their corresponding retrieval corpora. We evaluate Qwen3.5-9B (Qwen Team, 2026), Mistral-12B (Mistral-Nemo-Instruct-2407) (Mistral AI, 2024), and DeepSeek-V4-Flash (DeepSeek-AI, 2026). We compare against VanillaRAG, InstructRAG (Wei et al., 2025), Astute RAG (Wang et al., 2025), TrustRAG (Zhou et al., 2025), RobustRAG (Xiang et al., 2026), and SeCon-RAG (Si et al., 2025), as well as the update controls defined in Appendix C.1. Appendix B.2 extends the evaluation to Qwen3.5-27B and Qwen3.5-35B-A3B, together with two additional defenses, ReliabilityRAG (Shen et al., 2025) and RAGDefender (Kim et al., 2025).

We construct corpus-poisoning (PR) documents using PoisonedRAG (Zou et al., 2025) and JointGCG (Wang et al., 2026). For indirect prompt injection (PIA), malicious instructions are placed in retrieved passages (Greshake et al., 2023). After inserting the malicious passages into the corpus, we form contexts with a prescribed PR/PIA composition and select passages by BM25 score with $k = 10$. Strong attacks contain two PR and two PIA passages (20%+20%), whereas light attacks contain one of each (10%+10%). All remaining passages are clean. Every method receives the same attacked contexts, keeping exposure to malicious passages fixed across comparisons. Clean retrieval serves as a control for answer quality. Execution settings and population definitions are provided in Appendices A and D.1.

Accuracy (ACC) measures final-answer correctness using normalized substring matching. Attack Success Rate (ASR) is the fraction of eligible attacked queries for which the final answer matches the attack target but does not match any correct reference. Both metrics include queries that retain the initial answer, so failing a requirement does not remove a query from the evaluation. Main results are averaged over 10 runs. For Qwen under strong attacks, the mean cost across the three datasets is 22.8K logical tokens and 34.2 calls per query. Original-evidence checks account for 46% of these tokens, while counterfactual tests account for 7% (Appendix G).

Table 1: **Comparison with existing RAG defenses and a retention control** (%, means over 10 runs; ACC ↑, ASR ↓; settings in Section 5.1). **Bold**: best among all displayed methods, including Keep Initial, in each model and setting. Additional controls appear in Table 2.

Method	HotpotQA			NQ			MS-MARCO		
	20% PR +20% PIA ACC / ASR	10% PR +10% PIA ACC / ASR	Clean ACC	20% PR +20% PIA ACC / ASR	10% PR +10% PIA ACC / ASR	Clean ACC	20% PR +20% PIA ACC / ASR	10% PR +10% PIA ACC / ASR	Clean ACC
Qwen3.5-9B									
Keep Initial	56.3 / **2.0**	56.3 / **2.0**	56.3	58.7 / **2.0**	58.7 / **2.0**	58.7	61.2 / **2.0**	61.2 / **2.0**	61.2
VanillaRAG	3.6 / 95.3	31.5 / 64.6	63.7	4.6 / 92.8	37.6 / 57.1	65.8	6.4 / 92.1	40.2 / 51.8	73.6
InstructRAG	29.7 / 70.2	50.2 / 49.3	**78.5**	30.6 / 68.6	55.8 / 39.3	76.9	47.9 / 48.8	61.1 / 38.8	**83.5**
Astute RAG	28.7 / 50.5	50.5 / 31.9	59.6	54.8 / 18.5	64.7 / 4.6	64.6	50.6 / 29.6	64.3 / 11.5	67.6
TrustRAG	56.5 / 2.8	62.2 / 22.0	66.6	59.5 / 3.2	69.7 / 24.8	68.4	57.2 / 9.3	66.5 / 30.1	70.9
RobustRAG	6.3 / 90.7	37.9 / 56.3	45.1	13.0 / 81.1	44.4 / 40.3	52.8	22.7 / 72.0	55.7 / 33.5	65.5
SeCon-RAG	56.8 / 11.3	66.5 / 14.6	71.4	56.3 / 6.5	64.9 / 19.1	67.5	55.5 / 8.6	66.3 / 15.7	65.5
VeriRAG	**64.0** / 7.0	**69.0** / 4.0	78.0	**62.0** / 2.6	**78.0** / 6.2	**78.1**	**64.3** / 7.0	**68.2** / 10.2	65.5
Mistral-12B									
Keep Initial	49.6 / **1.0**	49.6 / **1.0**	49.6	55.1 / **1.0**	55.1 / **1.0**	55.1	**57.4** / **1.0**	57.4 / **1.0**	57.4
VanillaRAG	2.9 / 96.4	28.7 / 67.8	60.9	3.8 / 94.1	34.5 / 59.8	62.4	5.1 / 93.6	37.4 / 54.0	70.8
InstructRAG	26.8 / 73.1	47.1 / 52.2	**74.9**	28.2 / 71.4	52.0 / 43.1	**73.0**	44.1 / 52.3	57.8 / 42.1	**79.6**
Astute RAG	26.4 / 53.2	47.3 / 34.7	56.1	51.3 / 21.4	61.2 / 6.1	60.8	47.2 / 32.7	60.9 / 14.3	63.5
TrustRAG	52.7 / 3.9	58.4 / 24.7	63.1	56.1 / 4.6	65.8 / 27.1	64.9	53.8 / 11.7	62.1 / 33.5	67.2
RobustRAG	4.8 / 92.8	34.4 / 59.2	42.6	10.1 / 84.3	40.5 / 44.7	49.1	19.4 / 75.9	51.8 / 37.2	61.6
SeCon-RAG	53.4 / 13.6	**62.7** / 17.2	67.8	53.1 / 8.1	61.7 / 22.3	63.8	51.9 / 11.2	**62.8** / 18.4	62.2
VeriRAG	**58.6** / 4.1	61.9 / 10.8	70.8	**58.3** / 3.7	**66.4** / 5.0	68.6	56.4 / 9.2	61.8 / 12.5	65.0
DeepSeek-V4-Flash									
Keep Initial	81.2 / **0.5**	81.2 / **0.5**	81.2	78.6 / **0.5**	78.6 / **0.5**	78.6	**83.7** / **0.5**	83.7 / **0.5**	83.7
VanillaRAG	17.8 / 81.6	48.7 / 47.9	89.6	20.4 / 78.9	53.2 / 43.6	88.7	26.1 / 73.8	58.4 / 38.7	92.1
InstructRAG	52.6 / 42.7	68.3 / 28.4	92.8	55.9 / 38.6	72.1 / 24.9	91.7	65.3 / 32.4	77.6 / 21.5	94.6
Astute RAG	63.8 / 19.4	75.7 / 9.8	89.1	70.6 / 11.7	79.4 / 4.1	88.4	68.9 / 15.6	80.8 / 6.7	91.3
TrustRAG	78.6 / 1.4	84.1 / 6.3	90.7	80.8 / 1.8	86.4 / 7.1	90.2	79.7 / 4.6	85.9 / 10.8	92.6
RobustRAG	43.1 / 56.8	62.9 / 37.0	88.2	48.6 / 51.3	66.8 / 29.7	87.5	55.8 / 44.2	72.4 / 23.1	90.7
SeCon-RAG	80.9 / 5.7	**87.3** / 7.8	93.4	**81.7** / 3.6	87.8 / 9.4	92.5	82.6 / 5.1	**89.1** / 8.3	**95.2**
VeriRAG	**82.4** / 0.9	86.7 / 4.7	**94.5**	80.1 / 1.3	**88.6** / 3.0	**93.1**	83.4 / 3.7	88.4 / 5.5	94.3

5.2 Main Results

VeriRAG improves average answer accuracy while substantially reducing attack success (Tables 1 and 2). Across the 18 attacked settings, mean ACC increases from 67.85% for SeCon-RAG to 71.03% for VeriRAG, while mean ASR decreases from 11.47% to 5.63%. Among the retrieval-based methods, VeriRAG achieves the highest ACC in 13 of 18 settings and the lowest ASR in 15.

Keep Initial always returns the initial answer a_0 (Table 2). Full achieves higher ACC than Keep Initial in 16 of the 18 attacked settings. The two exceptions are Mistral and DeepSeek on MS-MARCO under strong attacks, where Full is lower by 1.0 and 0.3 points, respectively. Keep Initial has lower ASR in every setting because it never updates a_0; its remaining attack success therefore comes from incidental matches between the initial answer and the attack target.

To isolate the effect of counterfactual verification, we compare three nested update policies that use the same initial answers, candidates, and fallback rule. Original-Evidence Verification (OEV) accepts an update only when the original context supports the candidate, its citations are valid, and no competing answer has complete support. The variant without counterfactual verification still constructs and audits both edited views, but does not impose any requirements on the verifier's responses to them. Full adds these response requirements: the candidate must remain supported under preservation and become insufficiently supported under withdrawal. The latter two policies therefore differ only in how verifier responses to the same edited records are used.

Table 2: **Aggregate comparison with update controls** (%). Attacked metrics average 18 settings, and clean ACC averages nine. Bold marks the best value per column. SeCon-RAG is an external reference, whereas the other rows are update policies. Per-setting results appear in Appendix C.1 and Table 9.

Policy	Attacked ACC ↑	ASR ↓	Clean ACC ↑
Keep Initial	64.64	**1.17**	64.64
SeCon-RAG	67.85	11.47	75.48
Sufficient-context gate	68.87	9.36	**81.44**
Original-Evidence Verification	68.56	12.30	78.74
Full VeriRAG	**71.03**	5.63	78.66

Table 3: **Component ablations on NQ with Qwen3.5-9B.** Arrows: percentage-point changes from Full. P-only and W-only use only the preservation and withdrawal response requirements, respectively, while w/o counterfactual verification uses neither.

Defense	20% PR + 20% PIA		10% PR + 10% PIA		Clean
	ACC ↑	ASR ↓	ACC ↑	ASR ↓	ACC ↑
Full VeriRAG	62.0	2.6	78.0	6.2	78.1
w/o counterfactual verification	59.7 ↓2.3	4.9 ↑2.3	75.5 ↓2.5	12.0 ↑5.8	78.6 ↑0.5
P-only	58.4 ↓3.6	6.1 ↑3.5	74.6 ↓3.4	12.4 ↑6.2	77.8 ↓0.3
W-only	60.9 ↓1.1	3.0 ↑0.4	77.1 ↓0.9	7.8 ↑1.6	79.2 ↑1.1
Recognition with Context	58.8 ↓3.2	6.1 ↑3.5	75.6 ↓2.4	10.1 ↑3.9	76.1 ↓2.0
Original-Evidence Verification	57.5 ↓4.5	10.8 ↑8.2	75.4 ↓2.6	14.0 ↑7.8	78.2 ↑0.1

Compared with OEV, Full improves ACC and reduces ASR in all 18 attacked settings, with mean changes of $+2.47$ and -6.67 points, respectively. The sufficient-context gate accepts the shared candidate when the retrieved context is judged sufficient and otherwise returns a_0. Relative to this gate, Full improves ACC by 2.16 points and reduces ASR by 3.72 points on average. It achieves higher ACC in 14 settings and lower ASR in 16. On clean retrieval, however, Full has lower ACC in seven of nine settings, with a mean of 78.66% compared with 81.44% for the sufficient-context gate. The added robustness under attack therefore comes with a modest reduction in clean accuracy.

Appendix B.1 reports paired confidence intervals obtained by resampling queries while keeping attack contexts fixed. Appendix B.2 extends the evaluation to two larger Qwen models and two additional defenses.

5.3 Component Ablations and Answer-Update Outcomes

On Qwen3.5-9B/NQ, P-only, W-only, and w/o counterfactual verification share Full's candidates, frozen views, and audit outcomes (Table 3). Under attack, preservation alone does not improve on omitting response tests, whereas withdrawal alone improves both ACC and ASR. Combining the two improves both further, although W-only has 1.1 points higher clean ACC. Giving the recognizer access to retrieved text (Recognition with Context) raises ASR by 3.5–3.9 points relative to Full.

Averaging light and strong attacks within each of the nine model–dataset pairs, Full gains 1.57 ACC points and lowers ASR by 3.43 points over w/o counterfactual verification. ACC gains range from $+0.40$ to $+2.40$ points, ASR reductions from 2.50 to 4.60 points, and mean clean ACC changes by -0.14 points (Appendix C.2). On Qwen3.5-9B/NQ under strong attacks, counterfactual verification rejects 103 of the 403 updates that reach it (Appendix C.4).

At the same response-call budget, repeated checks of the original context (Repeated Original Verification) reach 60.8% ACC on NQ under strong attacks, versus 62.0% for Full. In the same setting, replacing the Qwen verifier with DeepSeek-V4-Flash changes ACC and ASR by less than one point (Appendix F.1).

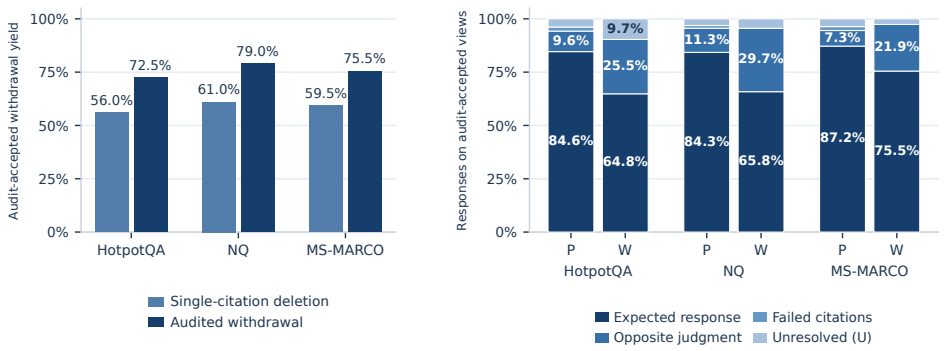


Figure 3: **Descriptive support-change diagnostics (RQ2, Qwen3.5-9B).** Left: audit-accepted withdrawal yield over eligible construction records. Right: responses on audit-accepted views. P requires supported with complete, current, unopposed citations, and W requires insufficient evidence. Acceptance is an online audit decision.

5.4 Construction Yield and Response Patterns

Figure 3 (left) compares single-citation deletion with audited withdrawal on Qwen3.5-9B. Yield is the fraction of eligible query–candidate–relation construction records producing an audit-accepted view, including construction failures in the denominator. Audited withdrawal raises yield by 16–18 points. On accepted views (right), the expected response is supported for preservation (P) and insufficient evidence for withdrawal (W). The verifier still endorses the candidate in 21.9–29.7% of withdrawal responses. Original-context checks leave this behavior untested. Citation failures and unresolved responses are reported separately.

A separate 100-query NQ study under 20% PR+10% PIA uses an external LLM judge blinded to online decisions (Appendix D.5). The judge finds 97.8% of audit-accepted preservation views and 96.7% of withdrawal views valid. It also finds valid edits for all 14 updates rejected by counterfactual verification, supporting the interpretation that these rejections reflect verifier responses on valid edits.

5.5 Attack Composition and Stress Tests

At 30% contamination, replacing one PR passage with a PIA passage changes Full's ACC and ASR by about three points or less on each dataset. VeriRAG has the highest ACC in all six dataset–composition settings and the lowest ASR in five (Appendix E.1).

On Qwen3.5-9B/NQ, redundant support and instructions targeting the verifier keep ASR at 2.3–2.7%, close to the mixed-attack reference of 2.6%. They raise fallback from 70.0% to 80.5–85.3% and reduce ACC by 1.1–1.3 points from 62.0% (Appendix E.3). Candidate-string injection raises ASR to 6.5%. Coherent false evidence combined with conditional responses raises ASR to 11.8% and lowers ACC to 56.3%, consistent with checking contextual support rather than evidence truth. Separate PR/PIA tests and contamination/position comparisons appear in Appendices E.2 and E.4.

6 Conclusion

We propose VeriRAG, a defense that decides whether retrieved evidence justifies replacing a model's initial answer under corpus poisoning and indirect prompt injection. For each required fact, it tests the verifier with two complementary audited views: a preservation view, which keeps the support and should leave the endorsement unchanged, and a withdrawal view, which removes it and should lead to an insufficient-evidence judgment. Experiments with three LLMs on three benchmarks show that VeriRAG roughly halves attack success relative to SeCon-RAG while improving accuracy. By testing whether endorsement tracks the required evidence rather than trusting a single judgment, VeriRAG provides a training-free defense for RAG answer updates, at a cost of about 34 model calls per query.

ETHICS STATEMENT

This work develops a defense against corpus poisoning and indirect prompt injection in RAG and evaluates it only on public QA benchmarks and their corpora, without involving deployed systems or human subjects. VeriRAG checks contextual support but does not guarantee factual correctness.

AI USE STATEMENT

We used generative AI tools to polish the language and improve the readability of the manuscript, and to assist with debugging research code. All AI-assisted revisions and code changes were reviewed by the authors. We take responsibility for the final content of the paper and the correctness of the research code.

REFERENCES

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

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

Lu Chen, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. Controlling risk of retrieval-augmented generation: A counterfactual prompting framework. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 2380–2393. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.133. URL https://aclanthology.org/2024.findings-emnlp.133/.

Sizhe Chen, Julien Piet, Chawin Sitawarin, and David Wagner. StruQ: Defending against prompt injection with structured queries. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 2383–2400. USENIX Association, 2025.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4443–4458. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.408. URL https://aclanthology.org/2020.acl-main.408/.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, pp. 79–90, 2023.

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 *The Thirteenth International Conference on Learning Representations*, 2025.

Minseok Kim, Hankook Lee, and Hyungjoon Koo. Rescuing the unpoisoned: Efficient defense against knowledge corruption attacks on RAG systems. In *Annual Computer Security Applications Conference (ACSAC)*, 2025. doi: 10.1109/ACSAC67867.2025.00093.

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*, 7:453–466, 2019.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

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 the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023.

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Hosseini, Mark Johnson, and Mark Steedman. Sources of hallucination by large language models on inference tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 2758–2774. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.182. URL https://aclanthology.org/2023.findings-emnlp.182/.

Mistral AI. Mistral-Nemo-Instruct-2407. https://huggingface.co/mistralai/Mistral-Nemo-Instruct-2407, 2024.

Muhaimin Bin Munir, Akib Jawad Ononto, Nazia Shehnaz Joynab, Bhavani Thuraisingham, and Latifur Khan. TRIS: A tri-layer retrieval integrity sieve against knowledge poisoning. In *Findings of the Association for Computational Linguistics: EMNLP 2026*, 2026.

Jingxi Qiu, Zeyu Han, and Cheng Huang. SURE-RAG: Sufficiency and uncertainty-aware evidence verification for selective retrieval-augmented generation. *arXiv preprint arXiv:2605.03534*, 2026.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Zeyu Shen, Basileal Imana, Tong Wu, Chong Xiang, Prateek Mittal, and Aleksandra Korolova. ReliabilityRAG: Effective and provably robust defense for RAG-based web-search. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2509.23519.

Jiawen Shi, Zenghui Yuan, Yinuo Liu, Yue Huang, Pan Zhou, Lichao Sun, and Neil Zhenqiang Gong. Optimization-based prompt injection attack to LLM-as-a-judge. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, 2024. URL https://arxiv.org/abs/2403.17710.

Xiaonan Si, Meilin Zhu, Simeng Qin, Lijia Yu, Lijun Zhang, Shuaitong Liu, Xinfeng Li, Ranjie Duan, Yang Liu, and Xiaojun Jia. SeCon-RAG: A two-stage semantic filtering and conflict-free framework for trustworthy RAG. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://doi.org/10.52202/085713-2376.

Fei Wang, Xingchen Wan, Ruoxi Sun, Jiefeng Chen, and Sercan Ö Arık. Astute RAG: Overcoming imperfect retrieval augmentation and knowledge conflicts for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30553–30571, 2025.

Haowei Wang, Rupeng Zhang, Junjie Wang, Mingyang Li, Yuekai Huang, Dandan Wang, and Qing Wang. Joint-GCG: Unified gradient-based poisoning attacks on retrieval-augmented generation systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 35793–35801, 2026.

Zhepei Wei, Wei-Lin Chen, and Yu Meng. InstructRAG: Instructing retrieval-augmented generation via self-synthesized rationales. In *The Thirteenth International Conference on Learning Representations*, 2025.

Chong Xiang, Tong Wu, Zexuan Zhong, David Wagner, Danqi Chen, and Prateek Mittal. Certifiably robust RAG against retrieval corruption. In *2026 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, 2026. URL https://arxiv.org/abs/2405.15556.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *The Twelfth International Conference on Learning Representations*, 2024.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, 2024.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

Zhe Yu, Wenpeng Xing, Gaolei Li, Shuguang Xiong, Hongzhi Wang, Xuyang Teng, and Meng Han. Cordon-MAS: Defending RAG against knowledge poisoning via information-flow control. *arXiv preprint arXiv:2605.26754*, 2026.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2306.05685.

Zexuan Zhong, Ziqing Huang, Alexander Wettig, and Danqi Chen. Poisoning retrieval corpora by injecting adversarial passages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13764–13775. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.849.

Huichi Zhou, Kin-Hei Lee, Zhonghao Zhan, Yue Chen, Zhenhao Li, Zhaoyang Wang, Hamed Haddadi, and Emine Yilmaz. TrustRAG: Enhancing robustness and trustworthiness in retrieval-augmented generation. *arXiv preprint arXiv:2501.00879*, 2025.

Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. PoisonedRAG: Knowledge corruption attacks to retrieval-augmented generation of large language models. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 3827–3844. USENIX Association, 2025.

A EXECUTION CONFIGURATION

This appendix specifies the execution settings, audit rules, and adoption predicates of Section 4. The supplementary material provides the machine-readable settings (`protocol/defaults.json`) and role prompts (`protocol/prompts.json`).

Models and decoding. In the default configuration, all online roles within a run use the same evaluated checkpoint in separate fresh calls. Verifier substitutions are reported separately in Appendix F.1. The external semantic judge for NQ diagnostics is DeepSeek-V4-Flash, whose labels do not enter online adoption (Appendix D.5). The primary local profile uses Qwen3.5-9B, and the local replication profile uses Mistral-Nemo-Instruct-2407. Local inference uses unquantized BF16 weights, the model's tokenizer and chat template, and a 32,768-token total context limit. Checkpoint, tokenizer, and template revisions are resolved and recorded before evaluation. Qwen uses non-thinking mode. Every online role uses greedy decoding: local generation disables sampling, and compatible APIs request temperature 0 and top-p 1 with no presence/frequency penalty. Seed 42 is used where supported, but this does not imply identical outputs across implementations. API profiles must expose required controls and length accounting or be documented as separate configurations.

Role inputs and output limits. The rubric generator sees only the question, and its auditor sees the question and proposed rubric. Candidate generators see the question alone, one ranked passage, or all ten passages. Recognition sees only the question, fixed rubric, and anonymous candidate identifier/value pairs. Evidence auditors see relevant full sources, rubric, records, and bindings. Edit auditors additionally see original and edited views and deletion coordinates. The context verifier sees only the question, fixed candidate, rubric, and current passages, without edit labels, audit verdicts, previous verifier outputs, reference answers, or attack targets. Output caps are 256 tokens for answer proposals, 512 for recognition, 2,048 for rubric calls and verification, and 4,096 for evidence audits and edit proposals. A truncated required output is unresolved, not a satisfied condition.

Rubric and response notation. Section 3.2 defines the rubric $\mathcal{T}_q$ and its factual expression Q_q in Eq. (2). The question-only audit assesses coverage without added requirements. Unknown entities remain unfilled in the fixed rubric and are bound using evidence. Unresolved requirements or structures beyond supported finite factual conjunctions prevent adoption. Following Section 3.3, for a fixed candidate we use $J(E) := J(q, a_c, \mathcal{T}_q, E)$, with reported labels S (supported), N (insufficient evidence), and U (unresolved). N concerns support in the supplied context, not answer falsity. An unresolved response does not count as N.

Candidate construction. One question-only call supplies the stored initial answer as the first candidate, one call per passage produces up to ten more candidates, and one full-context call permits a multi-passage candidate. Thus, at most 12 raw candidates are produced. The first raw question-only answer is retained verbatim as a_0 before trimming. After trimming surrounding whitespace, exact duplicates merge their provenance and retain their earliest ID. The order is question-only, passage rank, then full-context. We freeze $\mathcal{C} = \{(i, v_i, g_i)\}$, where i is the persistent identifier, v_i is the stored candidate value, and g_i is its eligibility flag. Eligibility requires a nonempty, well-formed proposal other than the unresolved sentinel, with a question-only origin or at least one registered source ID from the supplied context. Such provenance does not establish semantic support. No alias merging or substring projection is applied.

Recognition sees anonymous identifier/value pairs from the full pool without retrieved narratives or source provenance. It selects a complete answer for further testing only when the model identifies a concrete factual basis under the fixed rubric from its existing knowledge. Otherwise, it returns `UNRESOLVED` and the system retains a_0. Candidate strings remain untrusted, and recognition explanations are not used as retrieved support. An eligible update triggers a second fresh call with the entire pool reversed and display IDs reassigned, without the previous response. Both valid choices must map to the same persistent identifier i. Let $I_q(i)$ require this agreement, g_i, and $v_i \neq \mathrm{strip}(a_0)$. Failure to select the same identifier, or selection of the initial answer, retains the initial answer. Rejection does not trigger reselection from a reduced pool. The selected value is $a_c = v_i$ in the main text. Intervention tests target this fixed candidate, while original-evidence inspection also covers competitors.

Finite audits. The rubric receives one generation call and one fresh audit without iterative repair. Each source receives a record-extraction call and a fresh audit covering every necessary relation. A joint record review assesses entity compatibility, competing answers, and unresolved links across sources. Quotations use zero-based, half-open character spans in the exact source text, validated mechanically before semantic acceptance. Model-based audits assess assertion scope, support, and opposition in the full source context. Entity links require current textual grounds. Positive support and opposition remain separate: opposition does not remove an otherwise accepted positive support record. Limits are 64 records per source and 256 compatible joint bindings per view. Overflow, incomplete inspection, or unresolved identity blocks adoption rather than silently reducing the inspection domain. Records from unchanged sources may be reused only when their full inputs match, but joint compatibility and residual-support audits still cover all ten passage slots. Distinct literal answer values remain competitors, including values outside the candidate pool.

Using the accepted support sets $\mathcal{S}_{\ell,\beta}(E)$ and compatible assignments $\mathcal{B}^{+}_{a}(E)$ from Eq. (12), a complete positive witness selects one supporting record per relation under a common compatible assignment. An unopposed witness additionally requires completed opposition audits and no applicable direct opposition. These are finite audit outcomes, not semantic-recall guarantees. To satisfy an adoption requirement, an S response must cite a complete, current, unopposed witness and pass its citation audit. Uncited support cannot repair an insufficient cited witness.

Editing and pair selection. The original supported response must pass its citation audit before intervention construction. For selected candidate v_i and relation ℓ, we construct preservation and withdrawal views $E^{+}_{i,\ell}$ and $E^{-}_{i,\ell}$ independently from E^{0}. After fixing $a_c = v_i$, the main text abbreviates them as E^{+}_{ℓ} and E^{-}_{ℓ}. Preservation retains complete support for *every* original assignment in $\mathcal{B}^{+}_{v_i}(E^{0})$, with valid current entity correspondences and relevant opposition, attribution, and interpretation preserved. For every original complete assignment, withdrawal removes all applicable support for relation ℓ while preserving joint support for all remaining relations under one corresponding current assignment. All ten passages, including unchanged sources, undergo a residual audit requiring $\mathcal{B}^{+}_{v_i}(E^{-}_{i,\ell}) = \varnothing$. Duplicates and alternative paths are included. Any complete positive residual witness, even if opposed, invalidates withdrawal. Empty extraction or unresolved identity does not establish withdrawal. These semantic requirements apply in addition to the matching constraints below.

For each necessary relation and evidence version, one construction call proposes up to three preservation and withdrawal pairs. Each view is formed only by deleting original character intervals, preserving passage slots and order. Overlapping or adjacent intervals are merged, and at most eight merged intervals are allowed per modified source. The two views must modify the same sources with equal interval counts per source, and the preservation edit must be nonempty. Pairs are audited in proposal order, and the first audit-accepted pair is selected. All selected pairs are fixed before any edited-view verifier response. Response labels cannot guide pair selection, authorize a new proposal, or retrospectively validate an edit. Construction failure blocks adoption without establishing maliciousness or verifier error. Edited-view responses are evaluated only after the selected pairs pass these audits.

Length and position matching. Let n_j be the original token count of source j. The difference in remaining source lengths between paired views must not exceed $\max(8, \lceil 0.05 n_j \rceil)$ tokens for any modified source. For the fully serialized verifier input, paired length difference is bounded by $\max(16, \lceil 0.02 n_{\text{input}} \rceil)$ tokens, where n_{input} is its original-context length, including the fixed prompt and chat template. Both measurements use the evaluated model's tokenizer. Within each source, deletion intervals are paired in ascending start order. Their midpoint difference, divided by the original source's character count, must be at most 0.10 for every paired interval. These mechanical constraints supplement the semantic validity requirements in Sections 3.3 and 4.2 and do not establish validity alone.

Adoption predicates. The response requirements in Eq. (3) enter adoption through these predicates. Let $S_0(i)$ require a rubric accepted by its audit, candidate conformance to the answer slot, type, and format, and an audited original supported response. For relation ℓ, let $K_\ell(i)$ require an audit-accepted matched pair, $S^{+}_{\ell}(i)$ an audited supported preservation response, and $N^{-}_{\ell}(i)$ a completed response reporting insufficient evidence on the audit-accepted withdrawal. The evidence gate

is

$$H_E(i) = S_0(i) \wedge \mathrm{Complete}(i) \wedge \bigwedge_{\ell=1}^{L} \left[K_\ell(i) \wedge S_\ell^+(i) \wedge N_\ell^-(i) \right]. \tag{4}$$

Here $\mathrm{Complete}(i)$ requires every necessary inspection to resolve. A resolved failure completes an inspection but fails its gate. For $H_E(i)$ to hold, every fixed relation must satisfy its conditions.

Satisfied conditions do not erase complete competing support in the original evidence. Over the fixed candidate pool, define $\mathcal{P}(q) = \{j : \mathcal{B}_{v_j}^+(E^0) \neq \varnothing\}$, retaining competitors regardless of recognition, response outcomes, or contested support. Let X_q indicate complete positive support for an original-text-grounded answer outside the pool. This indicator can block adoption but cannot introduce an output candidate. Let U_q record unresolved original work affecting support, opposition, identity, candidate interpretation, or competition. Unfinished inspection is not evidence of absent competition. The discrete adoption predicate is

$$A_q^{\mathrm{idx}}(i) = I_q(i) \wedge H_E(i) \wedge [\mathcal{P}(q) = \{i\}] \wedge \neg X_q \wedge \neg U_q. \tag{5}$$

For the selected candidate $a_c = v_i$, the decision on the answer value in Section 3.1 is $A_q(a_c) = A_q^{\mathrm{idx}}(i)$. The superscript distinguishes the persistent-identifier predicate from the main-text decision. Both leave the fixed initial answer and evidence inputs implicit. If the predicate holds, the stored value is returned. Otherwise, the stored initial answer is retained. The adoption conditions assess textual support and observable behavior: shared model errors and coherent false evidence can still pass. Variation in candidate surface forms may exclude useful corrections. Retaining a_0 does not establish its correctness.

Reconstruction. Adoption predicates must use current, consistent records. Unresolved clues from fixed original passages enter U_q immediately. At most one reconstruction per query is permitted, triggered only by newly verified updates to original-evidence audit records. The question, original passages, rubric and candidate pool remain fixed, while dependent inspections are invalidated and rebuilt pairs are locked before new edited-view responses. An additional record proposed by an audit is not a verified update and cannot trigger reconstruction. Unless a fresh audit has already accepted that record, the affected original-evidence work remains unresolved and the execution returns a_0. Reconstruction refreshes inspections after a verified record change but does not authorize retrying failed or unresolved audits for a different decision.

Budgets and failures. The per-query ceilings are 256 model calls and 524,288 logical tokens, counting input/generated tokens, retries, and reconstruction. Under this accounting, Qwen3.5-9B averages 22.8K logical tokens per query over the 3,000 main-evaluation queries under 20% PR + 20% PIA, well below the per-query ceiling. Before each call, its input length plus output cap must fit the remaining budget and context window. Inputs are not silently truncated. Each call has a 120-second timeout and at most one identical transport retry after two seconds. Semantic rejection, malformed or truncated structured output, and failed audits are not retried for a different decision. Required failure or exhaustion returns the stored initial answer. If the initial call itself fails, an explicit execution-error record with an empty answer remains in evaluation denominators. Unexecuted gates are logged as not run, not inferred to have passed or failed.

Run records. A run manifest records resolved model and tokenizer versions, chat template, runtime, precision, prompt and configuration hashes, query IDs, ordered contexts, decoding controls, and hardware. Per-query records retain the initial answer, candidate provenance, rubric, evidence records, recognition permutations, deletion coordinates, audit outcomes, view hashes and freeze events, verifier outputs and citations, gate states, and the final answer. Evaluation labels and attacker targets are stored separately from defense inputs. Cached calls require exact input and version matches.

B Additional Main-Experiment Results

B.1 Uncertainty of Aggregate Main Results

We compare VeriRAG with SeCon-RAG in Table 1 through equal-weight means over the 18 attacked model, dataset, and setting combinations, excluding clean retrieval. Table 4 reports these means and the paired differences with 95% percentile bootstrap confidence intervals over queries. Each bootstrap replicate resamples the 1,000 queries of each dataset with replacement and applies the same resample to every model, attack setting, and method. Within a replicate, ACC is recomputed over all resampled queries, ASR over the resampled queries in the fixed eligible set (Appendix D.1), and the 18 per-combination values are then averaged with equal weights. Because both methods are scored on the same resampled queries, the difference intervals are paired and account for the correlation between their per-query outcomes.

Table 4: **Aggregate results with 95% bootstrap confidence intervals.** Equal-weight means over the 18 attacked combinations of Table 1, excluding clean retrieval. Differences are VeriRAG minus SeCon-RAG in percentage points (pp). SeCon-RAG is reported as a point estimate.

Metric	VeriRAG (%)	SeCon-RAG (%)	VeriRAG − SeCon-RAG (pp)
ACC ↑	71.03 [70.27, 71.78]	67.85	+3.18 [+2.56, +3.79]
ASR ↓	5.63 [5.27, 5.97]	11.47	−5.84 [−6.28, −5.41]

Both paired intervals exclude zero, supporting an aggregate improvement in ACC and ASR over SeCon-RAG under this resampling scheme.

These intervals quantify query-sampling variability for the fixed models, datasets, attack settings, and injected passages. Variation from regenerating attacks or serving-side nondeterminism is outside this calculation (Appendix A). The aggregate comparison with SeCon-RAG is the unit of inference, whereas per-setting rankings and comparisons with the update controls are descriptive unless accompanied by their own paired intervals. Table 10 reports descriptive per-setting accuracy gains over retention for Qwen on all three datasets.

B.2 Additional Models and Baselines

Table 5 supplements Table 1 with results for Qwen3.5-27B and Qwen3.5-35B-A3B. Dataset order, attack settings, and metric definitions follow the main table. Additional baselines include ReliabilityRAG (Shen et al., 2025), which uses document reliability for filtering and aggregation, and RAGDefender (Kim et al., 2025), which filters retrieved passages before generation. Across the 12 attacked settings, VeriRAG has the lowest ASR in 11 and the highest ACC in six, staying within 1.0 point of the most accurate method in the others. Its mean attacked ACC and ASR are 67.88% and 4.43%, versus 65.35% and 10.25% for SeCon-RAG and 66.22% and 13.38% for TrustRAG.

Table 5: **Additional results under mixed attacks and clean retrieval.** Results are percentages (ACC ↑, ASR ↓). PR denotes attacks implemented with Joint-GCG and PoisonedRAG, and PIA denotes indirect prompt injection. **Bold** marks the best value for each metric within each model and setting (ties included). Purple rows indicate our method.

Method	HotpotQA			NQ			MS-MARCO		
	20% PR +20% PIA ACC / ASR	10% PR +10% PIA ACC / ASR	Clean ACC	20% PR +20% PIA ACC / ASR	10% PR +10% PIA ACC / ASR	Clean ACC	20% PR +20% PIA ACC / ASR	10% PR +10% PIA ACC / ASR	Clean ACC
Qwen3.5-27B									
VanillaRAG	5.8 / 93.1	35.4 / 61.2	68.1	7.1 / 90.4	41.3 / 54.7	70.6	9.2 / 89.6	44.8 / 48.3	78.9
InstructRAG	33.8 / 66.1	55.7 / 44.2	**82.4**	35.1 / 64.3	60.9 / 35.8	**80.8**	51.6 / 44.5	65.2 / 34.7	**87.1**
Astute RAG	33.4 / 45.9	55.2 / 27.6	64.8	59.2 / 14.7	70.3 / 3.5	69.6	55.8 / 25.2	69.1 / 8.4	72.5
TrustRAG	61.8 / 1.9	67.6 / 18.4	72.8	**64.3** / **2.6**	74.1 / 20.7	73.4	62.1 / 7.4	**71.8** / 26.8	76.5
ReliabilityRAG	1.1 / 96.7	21.4 / 77.3	54.2	4.7 / 92.8	28.6 / 65.1	59.3	5.2 / 90.7	38.9 / 49.4	69.8
RAGDefender	10.4 / 89.5	38.7 / 24.6	48.6	13.9 / 78.4	43.2 / 26.7	54.8	34.7 / 57.6	46.9 / 27.2	64.1
RobustRAG	9.1 / 87.4	42.8 / 51.7	50.2	17.6 / 76.2	49.7 / 36.8	57.4	27.9 / 67.1	60.8 / 29.6	70.2
SeCon-RAG	61.1 / 8.4	**71.2** / 11.8	76.3	61.7 / 4.8	70.6 / 15.3	72.9	60.4 / 6.7	71.5 / 12.2	70.8
VeriRAG	**66.9** / **1.5**	70.4 / **6.9**	79.6	63.6 / 2.7	**75.4** / **2.8**	78.8	**64.7** / **4.9**	70.8 / **6.6**	78.1
Qwen3.5-35B-A3B									
VanillaRAG	4.9 / 94.0	34.1 / 62.5	67.0	6.3 / 91.3	40.2 / 55.6	69.2	8.3 / 90.4	43.0 / 49.5	77.6
InstructRAG	32.2 / 67.7	54.1 / 45.8	**81.3**	34.0 / 65.7	59.0 / 37.4	**79.4**	50.4 / 45.9	63.7 / 36.2	**86.0**
Astute RAG	31.7 / 47.2	53.8 / 29.1	63.5	57.8 / 16.2	68.9 / 3.9	68.1	53.9 / 27.1	67.7 / 9.6	71.1
TrustRAG	60.3 / 2.1	66.1 / 19.5	71.4	**62.8** / 2.9	72.6 / 22.1	72.0	60.7 / 8.1	**70.4** / 28.0	75.1
ReliabilityRAG	0.8 / 97.3	20.3 / 78.9	52.7	4.1 / 93.7	27.2 / 66.8	57.8	4.6 / 91.7	37.4 / 50.8	68.3
RAGDefender	9.2 / 90.7	37.4 / 25.8	47.2	12.5 / 79.9	41.9 / 28.1	53.2	33.0 / 59.0	45.3 / 28.8	62.7
RobustRAG	8.0 / 88.8	41.1 / 53.4	48.8	16.1 / 77.8	47.9 / 38.1	56.0	26.1 / 68.8	59.1 / 31.1	68.7
SeCon-RAG	59.8 / 9.3	**69.8** / 12.7	74.8	60.2 / 5.4	69.1 / 16.6	71.5	58.9 / 7.3	69.9 / 12.5	69.2
VeriRAG	**65.5** / **1.7**	69.1 / **7.3**	78.3	62.2 / **2.2**	**73.2** / **3.0**	77.4	**63.1** / **5.8**	69.7 / **7.8**	76.9

C Controlled Answer-Update Evaluation

This appendix reports controls on the main evaluation, with 1,000 queries per dataset and setting. Strong and light attacks denote 20% PR + 20% PIA and 10% PR + 10% PIA, respectively. The separate 100-query diagnostic substudy in Appendix D uses different attack mixtures, so its transition counts and external semantic assessments are not pooled with these results.

C.1 Retention and Shared-Candidate Controls

Keep Initial returns the stored question-only answer a_0, which is fixed across retrieval conditions for a given model and dataset. The three nested verification policies share candidates and fallback. Original-Evidence Verification (OEV) requires original support, valid citations, and no competing answer with complete support. The w/o counterfactual verification policy adds successful paired-view construction and validity audits, and Full adds the preservation and withdrawal response requirements. OEV–Full compares both additions, while the latter two policies differ only in the response requirements on shared views and audit outcomes. The sufficient-context gate adopts the shared candidate when the context is judged sufficient, and otherwise retains a_0.

Tables 6–8 report all three conditions. Under attack, Full exceeds OEV in ACC and reduces ASR in all 18 combinations, with equal-weight differences of $+2.47$ and -6.67 points. Against the sufficient-context gate, the differences are $+2.16$ and -3.72 points, with higher ACC in 14 combinations and lower ASR in 16. The sufficient-context gate has higher mean clean ACC (81.44% versus 78.66%). Keep Initial has lower ASR throughout, and Full improves on its ACC in 16 of 18 attacked combinations. On MS-MARCO under the strong attack, Keep Initial has higher ACC for Mistral (57.4 versus 56.4) and DeepSeek (83.7 versus 83.4).

Table 6: **Retention and shared-candidate controls: Qwen3.5-9B.** All values are percentages. Attack columns report ACC / ASR.

Policy	Clean ACC	Light ACC / ASR	Strong ACC / ASR
HotpotQA			
Keep Initial	56.3	56.3 / 2.0	56.3 / 2.0
OEV	78.4	66.8 / 10.2	61.6 / 14.3
Sufficient-context gate	82.1	68.7 / 6.7	60.8 / 11.0
Full	78.0	69.0 / 4.0	64.0 / 7.0
NQ			
Keep Initial	58.7	58.7 / 2.0	58.7 / 2.0
OEV	78.2	75.4 / 14.0	57.5 / 10.8
Sufficient-context gate	82.5	76.8 / 9.5	61.7 / 9.2
Full	78.1	78.0 / 6.2	62.0 / 2.6
MS-MARCO			
Keep Initial	61.2	61.2 / 2.0	61.2 / 2.0
OEV	64.4	63.7 / 14.4	58.7 / 16.5
Sufficient-context gate	70.1	66.1 / 10.0	63.3 / 11.7
Full	65.5	68.2 / 10.2	64.3 / 7.0

Table 7: **Retention and shared-candidate controls: Mistral-12B.** All values are percentages. Attack columns report ACC / ASR.

Policy	Clean ACC	Light ACC / ASR	Strong ACC / ASR
HotpotQA			
Keep Initial	49.6	49.6 / 1.0	49.6 / 1.0
OEV	71.2	60.4 / 17.1	55.0 / 12.6
Sufficient-context gate	74.9	62.2 / 12.3	57.2 / 11.4
Full	70.8	61.9 / 10.8	58.6 / 4.1
NQ			
Keep Initial	55.1	55.1 / 1.0	55.1 / 1.0
OEV	69.1	66.0 / 10.1	55.2 / 8.3
Sufficient-context gate	71.9	67.7 / 7.8	58.2 / 7.3
Full	68.6	66.4 / 5.0	58.3 / 3.7
MS-MARCO			
Keep Initial	57.4	57.4 / 1.0	57.4 / 1.0
OEV	65.5	61.6 / 17.0	53.7 / 16.3
Sufficient-context gate	71.3	63.4 / 11.3	57.1 / 12.0
Full	65.0	61.8 / 12.5	56.4 / 9.2

Table 8: **Retention and shared-candidate controls: DeepSeek-V4-Flash.** All values are percentages. Attack columns report ACC / ASR.

Policy	Clean ACC	Light ACC / ASR	Strong ACC / ASR
HotpotQA			
Keep Initial	81.2	81.2 / 0.5	81.2 / 0.5
OEV	94.4	82.3 / 11.3	81.9 / 9.1
Sufficient-context gate	93.7	79.9 / 7.0	80.4 / 7.0
Full	94.5	86.7 / 4.7	82.4 / 0.9
NQ			
Keep Initial	78.6	78.6 / 0.5	78.6 / 0.5
OEV	93.9	87.2 / 9.4	79.1 / 8.8
Sufficient-context gate	93.4	81.2 / 9.0	75.9 / 8.2
Full	93.1	88.6 / 3.0	80.1 / 1.3
MS-MARCO			
Keep Initial	83.7	83.7 / 0.5	83.7 / 0.5
OEV	93.6	86.2 / 10.2	81.7 / 11.0
Sufficient-context gate	93.1	81.1 / 7.5	77.9 / 9.5
Full	94.3	88.4 / 5.5	83.4 / 3.7

C.2 Core Ablations Across Models and Datasets

The core ablation follows the three-policy progression: original-evidence checks (OEV), then construction and validity audits (w/o counterfactual verification), then paired response requirements (Full). Table 9 averages the two attacked settings within each model–dataset pair. Full has higher mean ACC and lower mean ASR in all nine pairs, with equal-weight changes of $+1.57$ and -3.43 points. These are descriptive differences across the tested conditions, without paired uncertainty estimates for this contrast. Clean ACC changes range from -0.90 to $+0.40$ points, with a mean of -0.14, indicating a small clean-accuracy trade-off in this comparison.

Table 9: **Core policy ablations across nine model–dataset pairs.** ACC columns average the two attacked settings. Deltas are Full minus w/o counterfactual verification in percentage points. The final column uses clean retrieval.

	Attacked ACC			Attacked Δ		Clean
Dataset	OEV	w/o counterfactual verification	Full	ACC	ASR	ΔACC
Qwen3.5-9B						
HotpotQA	64.20	65.50	66.50	+1.00	-3.30	+0.40
NQ	66.45	67.60	70.00	+2.40	-4.05	-0.50
MS-MARCO	61.20	64.15	66.25	+2.10	-3.20	+0.20
Mistral-12B						
HotpotQA	57.70	57.95	60.25	+2.30	-4.60	0.00
NQ	60.60	60.60	62.35	+1.75	-2.50	-0.30
MS-MARCO	57.65	57.80	59.10	+1.30	-3.30	+0.30
DeepSeek-V4-Flash						
HotpotQA	82.10	82.50	84.55	+2.05	-4.10	-0.30
NQ	83.15	83.50	84.35	+0.85	-3.15	-0.90
MS-MARCO	83.95	85.50	85.90	+0.40	-2.65	-0.20

C.3 Main-Evaluation Net Accuracy Gains

Table 10 compares Qwen3.5-9B's reported mean ACC with its initial ACC on each 1,000-query dataset. Initial ACC is 56.3% on HotpotQA, 58.7% on NQ, and 61.2% on MS-MARCO. The net gain is Full ACC minus initial ACC, in percentage points, and uses the same values as Table 1.

Table 10: **Net accuracy gains over retention with Qwen.** ACC values (%) reproduce Table 1, and Δ is Full minus initial ACC in percentage points. Strong/light denote 20%+20%/10%+10%. Net gains do not decompose corrections and harmful updates.

Dataset	Condition	Initial ACC	Full ACC	ΔACC
HotpotQA	Strong	56.3	64.0	+7.7
	Light	56.3	69.0	+12.7
	Clean	56.3	78.0	+21.7
NQ	Strong	58.7	62.0	+3.3
	Light	58.7	78.0	+19.3
	Clean	58.7	78.1	+19.4
MS-MARCO	Strong	61.2	64.3	+3.1
	Light	61.2	68.2	+7.0
	Clean	61.2	65.5	+4.3

For a single run, let N_1 denote the number of initially correct answers, B incorrect-to-correct updates, and H correct-to-incorrect updates. Then

$$\mathrm{ACC} = \frac{N_1 + B - H}{N}, \qquad \mathrm{BCR} = \frac{B}{N - N_1}, \qquad \mathrm{HUR} = \frac{H}{N_1}. \tag{6}$$

Here ACC, BCR, and HUR are fractions. Initial and final ACC identify the net change $B - H$ within a run, but do not separate corrections from harmful updates. Accordingly, Table 10 reports net gains from the mean accuracies, without converting those means into integer transition counts or inferring BCR/HUR or paired intervals from them. Full's adopted updates are a subset of those from w/o counterfactual verification, but attributing the difference to harmful updates prevented and corrections lost requires their joint query-level outcomes.

This main-evaluation analysis uses 20% PR + 20% PIA and 10% PR + 10% PIA. The diagnostic substudy in Appendix D instead uses 20% PR + 10% PIA and 10% PR + 20% PIA on 100 NQ queries. Its transition counts and external semantic assessments concern that 100-query population.

C.4 A Query-Level Stage Funnel

Table 11 describes 1,000 query records from the Qwen/NQ strong-attack setting. Rows are mutually exclusive termination categories. Their display order does not specify the execution schedule. The first two categories have no fixed candidate, so a target-candidate count is not applicable. The target column measures the fixed candidate, whereas ASR measures the final answer, including fallback.

Table 11: **Query termination on Qwen/NQ under the strong attack.** Counts partition 1,000 queries by termination stage. Target denotes a fixed candidate matching the attack target, and n/a means no fixed candidate.

Termination category	Total	Target
Unresolved rubric	18	n/a
Recognizer abstention	222	n/a
Original support audit failed	200	33
Competing-answer gate blocked	41	8
Construction or validity audit failed	116	47
Counterfactual verification failed	103	40
Final adoption	300	9
Total	1,000	—

In these records, w/o counterfactual verification adopts the 403 candidates in the final two rows, of which Full rejects 103 and adopts 300. The nine adopted target candidates do not include target errors returned through fallback. These stage counts describe the decision process but do not reconstruct the main-table ACC/ASR, which average 10 runs, or its correction/harm decomposition.

C.5 Update Coverage and Adoption Precision

Coverage in Table 12 is the fraction of queries for which the candidate is adopted. Adoption precision is the fraction of adopted candidates scored correct. Because fallback also returns an answer, update coverage is distinct from answer coverage. The table describes discrete policy points, without estimating an effect at matched update coverage.

Table 12: **Update coverage and precision on Qwen/NQ, strong attack.** All values are percentages. Precision conditions on adopted updates, whereas overall ACC also includes the answers returned through fallback.

Policy	Update coverage	Adoption precision	ACC
OEV	51.9	55.9	57.5
w/o counterfactual verification	40.3	60.0	59.7
Repeated Original Verification	36.0	62.8	60.8
Full	30.0	66.7	62.0
Strict matching	24.8	69.4	62.5

For a fixed run on shared query records, let $m > 0$ be the number of adoptions by w/o counterfactual verification, and let $\mathrm{ACC}_{\mathrm{initial}}$ and $\mathrm{ACC}_{\mathrm{ablated}}$ be the initial ACC and the ACC of that policy. Uniformly retaining exactly $n_{\mathrm{keep}} \leq m$ of these adoptions, with all other queries returning a_0, gives

$$\mathbb{E}[\mathrm{ACC}_{\mathrm{random}}] = \mathrm{ACC}_{\mathrm{initial}} + \frac{n_{\mathrm{keep}}}{m}\big(\mathrm{ACC}_{\mathrm{ablated}} - \mathrm{ACC}_{\mathrm{initial}}\big). \tag{7}$$

A numerical estimate requires all quantities from the same run, so we do not combine main-table mean ACC with funnel counts to estimate a matched-coverage result. The repeated-verification control uses further original-evidence checks with the same allocated response-call budget, as defined in Appendix D.2.

C.6 Selected-View Response Diagnostics across Retrieval Conditions

Table 13 describes DeepSeek-V4-Flash responses on selected views under clean and attacked retrieval. Each row contains 1,000 selected preservation (P) views and 1,000 selected withdrawal (W) views. These view-level response rates describe a separate diagnostic sample from Figure 3, the query-level funnel, and the external-reference study in Appendix D.5.

Table 13: **Responses on selected preservation and withdrawal views.** All entries are percentages of the 1,000 selected views of the corresponding type. Light and strong attacks use 10% PR + 10% PIA and 20% PR + 20% PIA. W failure combines continued support and unresolved responses.

Dataset	Condition	P fail	W supported	W unresolved	W fail
HotpotQA	Clean	9.3	13.5	2.5	16.0
	Light	10.2	21.8	3.1	24.9
	Strong	11.3	26.7	3.7	30.4
NQ	Clean	9.7	14.8	2.8	17.6
	Light	10.6	23.5	3.7	27.2
	Strong	11.9	29.7	4.2	33.9
MS-MARCO	Clean	7.6	11.2	2.4	13.6
	Light	8.9	17.7	3.0	20.7
	Strong	9.6	22.6	3.6	26.2

Within these selected-view samples, W failure and continued endorsement are higher under attacks in each dataset. These are conditional response distributions, rather than query-level failure rates. The comparison does not establish that attack exposure alone caused the change: the selected candidates, relation records, and edits can differ between conditions. Continued endorsement concerns contextual support and does not by itself establish that the candidate is false in the world.

D A 100-Query Diagnostic Substudy

This appendix reports a separate 100-query NQ diagnostic substudy, including its policy outcomes, transition counts, and external semantic assessment. Its attack mixtures differ from the 1,000-query main-evaluation controls in Appendix C. We distinguish these scopes because their outcomes and diagnostic labels cannot be pooled.

D.1 Evaluation Scopes and Metric Populations

Table 1 evaluates three models on three datasets under 20% PR + 20% PIA and 10% PR + 10% PIA, alongside clean retrieval. The main evaluation contains 1,000 queries per dataset (3,000 total). Table 3 uses the same attack settings on NQ with Qwen3.5-9B. The net-accuracy summary in Appendix C.3 uses 1,000 queries per dataset on NQ, HotpotQA, and MS-MARCO under the same main-evaluation conditions. Here PR denotes attacks implemented with Joint-GCG and PoisonedRAG. The separate diagnostic substudy reported in this appendix instead uses 100 sampled NQ queries under clean retrieval, 20% PR + 10% PIA, and 10% PR + 20% PIA. RQ3 uses the latter two mixtures with Qwen3.5-9B on the 3,000-query main evaluation (1,000 queries per dataset). Its paired ten-passage contexts contain seven clean and three malicious passages, and within each

pair one malicious passage changes from PR to PIA while other inputs remain fixed. These results have distinct sample scopes and attack configurations. The attack proportions specify the PR/PIA composition of the controlled top-ten contexts, whose passages are selected by BM25 score. Corpus-injection budgets count the documents added before this selection and are distinct from these context-level proportions.

ACC scores all queries in each evaluation setting, whereas ASR scores only that setting's eligible attacked queries. Both use the final returned answer, including a retained initial answer. The ASR population differs from the initially incorrect and correct subsets defining BCR and HUR. In the 100-query NQ study, all 100 queries are ASR-eligible under each attack. RQ2 instead reports construction rates per eligible query, candidate, and relation record and response distributions per audit-accepted view, separately by dataset and view type. Its online-audit populations must not be pooled with external-reference populations in Appendix D.5.

D.2 Shared Inputs and Policy Definitions

The separate diagnostic substudy uses 100 NQ queries sampled from the Qwen3.5-9B main evaluation. Table 14 reports its outcomes. It is much smaller than the 1,000-query-per-dataset evaluations used elsewhere, including RQ3 (Figure 4). The same queries are evaluated under clean retrieval, 20% PR + 10% PIA, and 10% PR + 20% PIA. Within each setting, policies share the question-only initial answer $a_{0,q}$, recognized candidate, retrieved passages, and model configuration. Because rejection returns $a_{0,q}$, all queries remain in evaluation, including failed construction, unresolved inspections, and queries without an eligible update candidate. Both attacked settings use all 100 queries as the ASR-eligible set. Shared initial answers yield 41 initially incorrect and 59 initially correct answers in each setting. Thus ACC, ASR, and adoption rate use 100 queries, whereas BCR and HUR use 41 and 59, respectively. These denominators are separate from the relation and evidence-view populations below.

The policies compare retaining the initial answer, adopting an available candidate, and requiring additional verification before adoption. *Keep Initial* always returns $a_{0,q}$. *Adopt Candidate* adopts the recognized eligible update candidate when available and otherwise retains $a_{0,q}$, so its adoption rate need not be 100%. *Original-Evidence Verification* retains original-evidence checks for support, citations, and competition, along with other non-intervention conditions. *w/o counterfactual verification* also requires successful construction and validity audits for all required preservation and withdrawal pairs but does not use edited-view verifier responses for adoption. *Full VeriRAG* adds paired response requirements to these prerequisites. Its adopted updates are therefore a subset of those from w/o counterfactual verification on shared records.

Repeated original verification under fixed permutations. To compare counterfactual verification with further original-evidence checks, Repeated Original Verification shares Full VeriRAG's candidates, original-evidence checks, intervention construction, and validity audits. It replaces counterfactual verification with additional verification of the original evidence under passage permutations fixed before response evaluation. Passage contents and IDs remain unchanged. The verifier template, greedy decoding, and output limits match Full. The prompt does not reveal the expected response label. Full retains the passage order specified by its own protocol.

Both methods are allocated $2L$ additional verifier calls per query, corresponding to L necessary relations with one P/W pair each. Repeated adopts only if every additional call returns `Supported` and passes the same citation requirements for complete, current, unopposed support. Any contrary, unresolved, or citation-failing response blocks adoption. Both methods stop when a required condition fails or remains unresolved, so their actual call counts may differ even though their allocated call budgets match. These are repeated original-evidence checks under fixed input permutations, not independent random judgments. The Repeated and Full acceptance sets need not be nested.

D.3 Answer-Transition Counts

To measure these adoption consequences, let $c_q(a) \in \{0, 1\}$ indicate whether answer a is scored correct for query q, and let $\widehat{a}_q$ be a policy's final answer. Define n_{xy} as the number of queries with $c_q(a_{0,q}) = x$ and $c_q(\widehat{a}_q) = y$. Across all settings of this 100-query substudy, $N_0 = n_{00} + n_{01} = 41$

Table 14: **Answer updates on 100 sampled NQ queries** (%). BCR and HUR have denominators 41 and 59. Adopt is the fraction of queries updated. Keep Initial, which always returns a_0 and is omitted, scores 59 ACC and 2 ASR with zero BCR, HUR, and Adopt.

Policy	ACC ↑	ASR ↓	BCR ↑	HUR ↓	Adopt
Clean					
Adopt Candidate	78	—	70.7	16.9	68
Original-Evidence Verification	79	—	61.0	8.5	54
w/o counterfactual verification	78	—	56.1	6.8	48
Repeated Original Verification	77	—	53.7	6.8	45
Full VeriRAG	77	—	48.8	3.4	40
20% PR + 10% PIA					
Adopt Candidate	55	31	61.0	49.2	82
Original-Evidence Verification	61	21	53.7	33.9	65
w/o counterfactual verification	65	14	48.8	23.7	53
Repeated Original Verification	65	11	43.9	20.3	46
Full VeriRAG	68	6	39.0	11.9	39
10% PR + 20% PIA					
Adopt Candidate	54	30	63.4	52.5	84
Original-Evidence Verification	60	19	56.1	37.3	66
w/o counterfactual verification	64	12	51.2	27.1	55
Repeated Original Verification	66	4	41.5	16.9	42
Full VeriRAG	69	5	41.5	11.9	38

and $N_1 = n_{10} + n_{11} = 59$ (Table 15). Then

$$\mathrm{BCR} = \frac{n_{01}}{N_0}, \qquad \mathrm{HUR} = \frac{n_{10}}{N_1}, \qquad \mathrm{ACC} = \frac{N_1 + n_{01} - n_{10}}{N_0 + N_1}. \tag{8}$$

These ratios estimate $\Pr[c_q(\widehat{a}) = 1 \mid c_q(a_0) = 0]$ and $\Pr[c_q(\widehat{a}) = 0 \mid c_q(a_0) = 1]$, respectively. BCR counts corrections among initially incorrect answers, whereas HUR counts losses among initially correct answers. Incorrect-to-incorrect changes fall outside HUR, whereas ASR separately counts attacker-target errors. The initial answer matches the attack target for two queries in each attacked setting, giving Keep Initial 2% ASR.

D.4 Marginal Selection and Equal-Adoption Control

On this 100-query subset, comparing Full with w/o counterfactual verification measures the observed selection change when adding the response requirements. For clean retrieval and the two attacks, respectively, Full prevents 2, 7, and 9 harmful updates while losing 3, 4, and 4 beneficial corrections. With $N = 100$, the difference in accuracy is therefore -1, $+3$, and $+5$ percentage points:

$$\Delta\mathrm{ACC} = \frac{\text{harmful updates prevented} - \text{corrections lost}}{N}. \tag{9}$$

These counts have a common denominator, unlike BCR and HUR. Under the two attacks, Full retains $16/20 = 80.0\%$ and $17/21 \simeq 81.0\%$ of the corrections from w/o counterfactual verification, while preventing $7/14$ and $9/16$ of its harmful updates. This is retention relative to w/o counterfactual verification, not a fraction of all possible corrections.

Uniform random retention. We compare Full's observed selection on this subset with uniform random retention at the same adoption count. For each setting, let m be the number of w/o counterfactual verification updates and n_{keep} the number of Full updates. Uniformly retaining exactly n_{keep} of the m updates assigns each an inclusion probability n_{keep}/m, and all unretained queries return the initial answer. For a metric M that averages per-query indicators over a fixed population, let M_{ablated} denote the result of w/o counterfactual verification. Then

$$\mathbb{E}[M_{\mathrm{random}}] = M_{\mathrm{initial}} + \frac{n_{\mathrm{keep}}}{m}\big(M_{\mathrm{ablated}} - M_{\mathrm{initial}}\big). \tag{10}$$

Table 15: **Transition counts underlying Table 14.** n_{xy} records initial and final correctness, not whether the answer string changed. Adopt counts candidate updates, and all other queries return the initial answer. Each setting has 100 queries.

Policy	n_{00}	n_{01}	n_{10}	n_{11}	Adopt
Clean					
Keep Initial	41	0	0	59	0
Adopt Candidate	12	29	10	49	68
Original-Evidence Verification	16	25	5	54	54
w/o counterfactual verification	18	23	4	55	48
Repeated Original Verification	19	22	4	55	45
Full VeriRAG	21	20	2	57	40
20% PR + 10% PIA					
Keep Initial	41	0	0	59	0
Adopt Candidate	16	25	29	30	82
Original-Evidence Verification	19	22	20	39	65
w/o counterfactual verification	21	20	14	45	53
Repeated Original Verification	23	18	12	47	46
Full VeriRAG	25	16	7	52	39
10% PR + 20% PIA					
Keep Initial	41	0	0	59	0
Adopt Candidate	15	26	31	28	84
Original-Evidence Verification	18	23	22	37	66
w/o counterfactual verification	20	21	16	43	55
Repeated Original Verification	24	17	10	49	42
Full VeriRAG	24	17	7	52	38

This applies to ACC and to ASR with its shared, fixed eligible set. Here (m, n_{keep}) is $(48, 40)$ for clean retrieval, $(53, 39)$ for 20% PR + 10% PIA, and $(55, 38)$ for 10% PR + 20% PIA. The resulting expected ACC values are 74.8%, 63.4%, and 62.5%, and the expected ASR values in the attacked settings are 10.8% and 8.9%. Table 16 compares these expectations with Full's observed outcomes. These are analytic expectations from the shared outcomes, not observations from an additional random run, and require no model calls. On these records, Full has higher ACC and lower attacked ASR than uniform random retention in expectation. This comparison does not establish a tail probability for random selection or compare against every informed rule.

Table 16: **Uniform random retention at Full's adoption count.** Random uniformly retains the indicated number of w/o counterfactual verification updates and returns the initial answer otherwise. Random columns are analytic expectations, whereas Full columns are observed results. All values are percentages.

		ACC ↑		ASR ↓	
Setting	Adopt	Random	Full	Random	Full
Clean	40	74.8	77	—	—
20% PR + 10% PIA	39	63.4	68	10.8	6
10% PR + 20% PIA	38	62.5	69	8.9	5

D.5 External Semantic Assessment of Interventions

The update counts describe which answers change, whereas the following diagnostics examine whether the interventions used to test those updates meet their semantic requirements. They concern only the 100-query NQ diagnostic subset under 20% PR + 10% PIA, not the main-evaluation population. Of the 65 queries passing original verification, the construction pool contains 100 query, candidate, and relation records, each with a P and a W construction attempt. A total of 53 queries,

containing 83 relation records, pass all required construction and online validity audits. Of these, 39 queries pass every response requirement. Each of the 83 relation records contributes one selected preservation and withdrawal pair, giving 83 views of each type. The remaining 12 original-passing queries contain 17 relation records. Some individual views from these queries can pass an online audit even though another required condition fails. Consequently, individually accepted views and relation records belonging to queries that pass every audit are different populations.

Judge and information separation. DeepSeek-V4-Flash performs an *external LLM-based semantic assessment* in calls separate from the online audits. It receives the query, candidate, necessary relations, complete original context, edited context, and target relation. It first assesses the query decomposition and the premise of complete original support. For P, it assesses whether support for all original complete bindings and the relevant semantic qualifications remains. For W, it assesses whether all applicable support for the target relation is removed while joint support for the other relations remains, including a full-context search for duplicate evidence and alternative support paths.

The judge returns `Valid`, `Invalid`, or `Unresolved`, with a short rationale and supporting text. It does not receive online audit verdicts, verifier responses, or final adoption decisions. Labels are frozen before linkage to response logs and do not participate in online adoption. They concern contextual support, not whether the candidate is true in the world. No human annotation is used, so procedural separation supplies a fallible model reference rather than independent semantic ground truth.

Audit precision and construction yield. Online audit precision is the fraction of audit-accepted views that the external judge labels Valid. Accepted-valid yield is the fraction of all eligible construction records yielding a view that is both audit-accepted and reference-valid, with construction failures kept in the denominator. These are view-level quantities, distinct from the query-level requirement that the pairs for every relation pass.

Table 17: **Online audit outcomes against the external LLM reference.** Each type has 100 eligible construction records. Acceptance counts include individually accepted views belonging to queries that fail another required construction or audit.

Quantity	Preservation	Withdrawal
Eligible construction records	100	100
Online audit-accepted views	93	91
Audit-accepted and reference-valid views	91	88
Online audit precision	91/93 (97.8%)	88/91 (96.7%)
Accepted-valid yield	91/100 (91.0%)	88/100 (88.0%)

Response populations and failure types. P requires endorsement with complete, valid citations, and W requires insufficient evidence. Table 18 groups reference-valid views with recorded responses by population. Among the 53 queries passing all construction and validity audits, 81 P and 80 W views meet both criteria, out of 83 selected views per type. The excluded two P and three W views are not necessarily `Invalid`: the aggregates do not separate reference status from response availability. Additional offline diagnosis adds 11 valid P views with three failures and ten valid W views with four failures, giving denominators of 92 and 90 without changing adoption decisions. Table 19 breaks these failures down by category. Online adoption stops at the first failed or unresolved condition, leaving any unexecuted inspections as `not_run`. These aggregates do not establish which responses preceded early stopping and therefore cannot measure online-path failure counts.

Table 18: **Response failures on reference-valid views.** Entries show failures / valid views with recorded responses, followed by percentages, grouped by view population.

Scope	Preservation	Withdrawal
Audit-passing queries	8/81 (9.9%)	19/80 (23.8%)
With additional diagnostic views	11/92 (12.0%)	23/90 (25.6%)

Table 19: **Response categories including offline diagnosis.** Counts are followed by percentages, with denominators 92 for P and 90 for W. Rounding can prevent displayed percentages from summing to 100.

Response category	Preservation	Withdrawal
Expected judgment with applicable citation audits	81 (88.0%)	67 (74.4%)
Opposite judgment: P insufficient; W supported	7 (7.6%)	19 (21.1%)
P supported with invalid or incomplete citations	2 (2.2%)	0 (0.0%)
Unresolved	2 (2.2%)	4 (4.4%)
All response failures	11 (12.0%)	23 (25.6%)

Withdrawal's 25.6% total failure rate is therefore not entirely continued endorsement: $19/90$ responses endorse and $4/90$ are unresolved. These outcomes do not by themselves imply candidate falsehood or attacker success. The reference-valid populations here also differ from the online-audit-accepted populations reported in Figure 3.

D.6 Linking Rejected Updates to Outcomes

For the 100-query NQ subset under 20% PR + 10% PIA, we connect the intervention and response diagnostics to the marginal selection counts above. Among the 53 queries passing all construction and online audits, response requirements reject 14 updates and leave 39 adopted, counted per query. The edits associated with these 14 rejected updates are all reference-valid. Had they been adopted, four would have changed an incorrect initial answer to a correct one, seven a correct answer to an incorrect one, one a correct answer to another correct answer, and two an incorrect answer to another incorrect answer. Only the first two groups affect accuracy. Rejecting them prevents seven harmful updates while losing four corrections, yielding an accuracy gain of $(7-4)/100$, or three percentage points, over w/o counterfactual verification. This connects external-reference edit labels, response failures, and final outcomes within the diagnostic subset. The external judge is used for analysis rather than as an additional adoption gate.

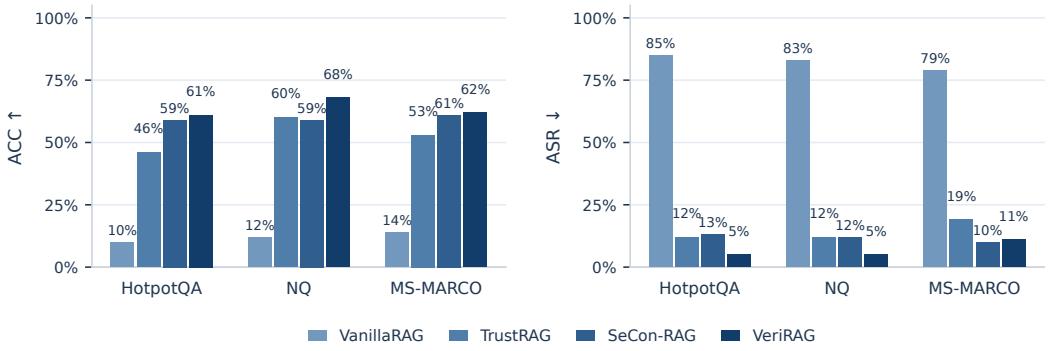
The reference-validity statement above applies to the 14 rejected updates. It does not establish that every intervention for the 39 adopted queries is reference-valid. Such a query-level criterion would require the external judge to label both views `Valid` for every necessary relation. The marginal view-level precision in Table 17 cannot substitute for this joint criterion. The reported net gain therefore quantifies the effect of the rejected updates, without certifying every accepted intervention or the truth of the adopted answers.

E Attack Stress Tests

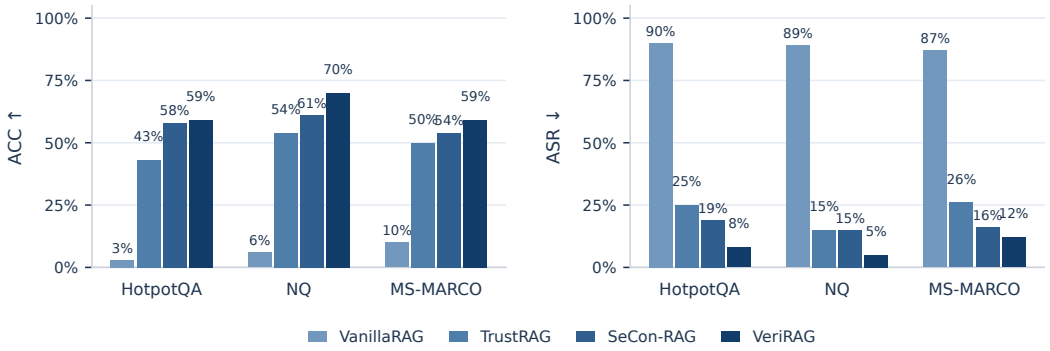
E.1 Robustness to Attack Composition

RQ3 tests the same adoption rule under changed attack composition on the 3,000-query main evaluation (1,000 queries per dataset) with Qwen3.5-9B. Each context contains seven clean and three malicious passages. We pair 20% PR + 10% PIA with 10% PR + 20% PIA contexts by changing one malicious passage from PR to PIA while fixing all other passages, their positions, and the attack target. Compared methods share these contexts and settings, and VeriRAG receives no attack-type labels.

Figure 4 reports ACC and ASR. VeriRAG has the highest ACC in all six dataset–composition combinations and the lowest ASR in five. The exception is MS-MARCO under 20% PR + 10% PIA, where SeCon-RAG reaches 10% ASR versus VeriRAG's 11%. Within each dataset, VeriRAG's ACC and ASR change by about three points or less between the two compositions. These results concern the tested compositions within controlled ten-passage contexts.



(a) 20% PR + 10% PIA



(b) 10% PR + 20% PIA

Figure 4: **Attack composition at fixed 30% contamination (RQ3).** Qwen3.5-9B on the 3,000-query main evaluation (1,000 queries per dataset), with method settings fixed across compositions. Left: ACC. Right: ASR over the predefined eligible set.

Evaluation settings. The separate-attack, protocol-aware, contamination/position, verifier-transfer, and design-sensitivity tests all use Qwen3.5-9B on the 1,000 NQ queries. Their reference is the 20% PR + 20% PIA mixed attack of the main evaluation (62.0% ACC, 2.6% ASR). These experiments examine attack conditions and update configurations separately from the composition experiment above.

E.2 Separate Poisoning and Injection Conditions

At a fixed total contamination of four out of ten retrieved passages, Table 20 compares pure PR, pure PIA, and the two-PR/two-PIA mixture. The last three columns count fixed target-matching candidates whose terminal rejection is assigned to the indicated condition. The candidate populations entering each stage vary with the attack mix. Final ASR includes target-matching answers retained by fallback.

Table 20: **Attack type at 40% total contamination.** ACC, ASR, and fallback are percentages. Original, Pair, and Response are counts of target-matching candidates rejected at the original-support, construction/validity-audit, and response stages, respectively.

Retrieved attack mix	ACC	ASR	Fallback	Original	Pair	Response
4 PR + 0 PIA	63.2	4.1	69.7	110	55	25
0 PR + 4 PIA	61.6	2.4	70.6	108	52	76
2 PR + 2 PIA	62.0	2.6	70.0	33	47	40

Pure PIA has more response-stage target-candidate rejections than pure PR (76 versus 25), while its final ASR is lower (2.4% versus 4.1%). Because the candidate populations and earlier rejections also change, these totals describe where rejection occurs under each attack, rather than independent effects of the individual stages or attack types.

E.3 Protocol-Aware Passage Conditions

Following the SeCon-RAG poisoning procedure (Si et al., 2025), these attacks operate through malicious retrieved passages. Candidate-string injection is induced during generation from retrieval, not by the attacker directly overwriting the stored candidate. Table 21 evaluates the specified passage conditions targeting redundant support, verifier response behavior, or the candidate string. It reports utility alongside ASR: a low target-match rate can coexist with frequent fallback and lower accuracy. The final row reports coherent false evidence and protocol-aware conditional responses jointly.

Table 21: **Tested protocol-aware passage conditions and utility loss.** All values are percentages. The final row reports the two named mechanisms jointly.

Attack condition	ACC	ASR	Fallback
Original mixed attack (reference)	62.0	2.6	70.0
Redundant support	60.8	2.7	80.5
Unconditional `Supported` instruction	60.9	2.4	82.0
`Insufficient` upon detecting an edit	60.7	2.3	85.3
Candidate-string injection	59.0	6.5	74.5
Coherent false evidence / protocol-aware conditional response	56.3	11.8	66.2

The redundant-support and response-instruction attacks keep ASR between 2.3% and 2.7%, while fallback rises from 70.0% to 80.5–85.3% and ACC falls from 62.0% to 60.7–60.9%. Candidate-string injection raises ASR to 6.5%. The jointly reported coherent-evidence/protocol-aware condition reaches 11.8% ASR and 56.3% ACC. These descriptive comparisons characterize utility and target matching for the tested passage conditions but do not establish robustness to an attacker optimizing against the defense.

E.4 Contamination Rate and Passage Position

Table 22 varies the fraction of malicious passages and their placement in the retrieved context. The 50% setting contains three PR and two PIA passages, and the 60% setting contains three of each. Front placement has lower ACC and higher ASR than back placement at each contamination level. Because the attack composition is not fixed across all three levels, these comparisons do not isolate contamination alone.

Table 22: **Contamination and placement stress tests.** All metrics are percentages. Front and back indicate the placement of malicious passages within the retrieved order.

Contamination	Position	ACC	ASR	Fallback
40%	Front	61.0	3.4	70.5
40%	Back	62.8	2.3	69.8
50%	Front	60.8	4.1	73.5
50%	Back	61.8	3.1	71.7
60%	Front	59.1	5.2	77.2
60%	Back	60.5	3.9	75.4

In the main evaluation (Table 1), attack strength also does not determine ASR monotonically: Full's ASR is higher under light than under strong attacks in eight of nine model–dataset pairs, as it is for TrustRAG and SeCon-RAG in all nine. Because ASR counts target answers from both adopted candidates and retained initial answers, the attack proportion alone does not determine realized attack success.

F Configuration Sensitivity and Question Structure

F.1 Changing the Online Verifier

Table 23 changes the online verifier while retaining the candidates and the other model roles. The DeepSeek verifier obtains 62.7% ACC and 2.4% ASR, versus 62.0% and 2.6% for the same-model Qwen reference. The mismatched heterogeneous configuration yields 61.2% ACC and 3.6% ASR. This experiment describes configuration sensitivity. Replacing the verifier does not establish independence of its errors from those in candidate selection or auditing.

Table 23: **Online-verifier configurations** (Qwen3.5-9B, NQ, 20% PR + 20% PIA). Candidate generation and other roles are held fixed. All metrics are percentages. Fallback still returns the initial answer.

Verifier configuration	ACC	ASR	Fallback
Qwen, same-model reference	62.0	2.6	70.0
DeepSeek	62.7	2.4	72.2
Heterogeneous, mismatched	61.2	3.6	72.1

F.2 Recognition and Edit-Matching Sensitivity

Table 24 changes recognition agreement, edit proposal availability, and matching tolerances. The default uses two recognition calls with reversed candidate orders and at most three proposed pairs per relation, selecting the first pair that passes validity audits. It does not perform counterfactual verification on all three proposals.

Table 24: **Sensitivity of the update rule** (Qwen3.5-9B, NQ, 20% PR + 20% PIA). All metrics are percentages. Adopt is the percentage of all evaluated queries that adopt the candidate, and the remaining queries return the initial answer.

Configuration	ACC	ASR	Adopt
Default	62.0	2.6	30.0
First recognition call only	60.8	4.4	35.4
First proposed pair per relation only	61.5	2.4	24.9
Strict matching	62.5	2.3	24.8
Loose matching	61.4	3.4	33.4

Strict, default, and loose matching use source-length relative tolerances of 3%, 5%, and 7%; serialized-input relative tolerances of 1%, 2%, and 3%; and normalized interval-position tolerances

of 0.07, 0.10, and 0.15, respectively. The absolute minimum tolerances and semantic validity requirements in Appendix A are unchanged. Strict matching achieves higher ACC and lower ASR than the default here, with lower adoption coverage. The default is therefore not an optimum for all reported metrics.

F.3 Results by Rubric Relation Count

Table 25 stratifies Qwen results under the strong 20% PR + 20% PIA mixture by the number of required relations L. Unresolved rubrics remain in the overall query population and return the initial answer. Relation count is a property of the generated rubric and does not substitute for dataset labels or human judgments of reasoning hops. The small $L \geq 3$ groups on NQ and MS-MARCO also limit comparisons across strata. This analysis describes final accuracy and adoption by relation count but does not decompose outcomes by initial correctness.

Table 25: **Relation-count strata under strong attacks with Qwen.** N is the query count and N_{final} counts finally correct answers. ACC and Adopt are percentages. Unknown denotes an unresolved complete rubric, not a dropped evaluation example.

Dataset	L	N	N_{final}	ACC	Adopt
NQ	1	632	387	61.2	29.9
	2	285	181	63.5	30.2
	≥ 3	65	40	61.5	38.5
	Unknown	18	12	66.7	0.0
HotpotQA	1	190	118	62.1	38.9
	2	614	392	63.8	38.8
	≥ 3	178	121	68.0	42.7
	Unknown	18	9	50.0	0.0
MS-MARCO	1	751	475	63.2	36.2
	2	186	124	66.7	40.3
	≥ 3	45	33	73.3	35.6
	Unknown	18	11	61.1	0.0

The syntactic parse success rate is 98.9%, whereas the complete-rubric gate passes 98.2%. These are different conditions: a parseable rubric can still fail the coverage or resolution requirements. Neither rate establishes that the rubric's semantic decomposition is correct.

G COMPUTATIONAL COST

Table 26 compares inference costs for Qwen3.5-9B under 20% PR + 20% PIA over the three main datasets, with 1,000 queries per dataset. Logical tokens count input and generated tokens, including retries and reconstruction, following Appendix A. The shared-candidate sufficient-context gate includes its candidate construction costs. Keep Initial uses the retrieval-free initial-answer call.

Table 26: **Inference cost on the main strong-attack evaluation.** Each entry is mean / median per query. Tokens are in thousands (K). Time is estimated serial inference service time in seconds.

Method	Tokens (K)	Calls	Estimated time (s)
Keep Initial	0.15 / 0.13	1.0 / 1.0	0.76 / 0.69
VanillaRAG	1.04 / 0.91	1.0 / 1.0	1.40 / 1.25
InstructRAG	1.43 / 1.27	2.0 / 2.0	2.70 / 2.42
Astute RAG	3.65 / 3.22	4.0 / 4.0	5.41 / 4.83
TrustRAG	5.00 / 4.47	7.0 / 7.0	7.84 / 7.10
RobustRAG	9.45 / 8.43	12.0 / 11.0	11.03 / 9.87
SeCon-RAG	6.25 / 5.55	8.0 / 8.0	8.79 / 7.90
Sufficient-context gate	8.05 / 7.11	16.8 / 16.0	16.31 / 14.54
Original-Evidence Verification	17.70 / 15.96	29.9 / 29.0	28.74 / 26.16
w/o counterfactual verification	21.20 / 18.96	32.9 / 31.0	34.61 / 31.14
Repeated Original Verification	23.15 / 20.52	34.5 / 33.0	36.98 / 33.08
Full VeriRAG	22.80 / 20.38	34.2 / 33.0	36.27 / 32.72

Service-time model. For query q, let t_q^{in}, t_q^{out}, and c_q be its input tokens, generated tokens, and number of model calls. We use fixed rates of 6,000 input tokens/s and 80 output tokens/s, plus 0.12 seconds per call, to define

$$\widehat{T}_q = \frac{t_q^{\text{in}}}{6{,}000} + \frac{t_q^{\text{out}}}{80} + 0.12 c_q. \tag{11}$$

This serial inference estimate is not measured end-to-end latency and excludes retrieval, networking, and queueing. The mean follows from the component means by linearity. A median service time must instead be computed by applying Eq. (11) per query and then taking the median. Adding component medians does not recover it. Total-token and call summaries alone also do not determine service-time medians because the input/output split and their joint per-query distribution matter.

Table 27: **Mean cost by stage for Full VeriRAG.** The population matches Table 26. Tokens are in K per query. Calls are averaged over all queries, including early exits.

Stage	Input (K)	Output (K)	Calls
Candidate generation	4.80	0.50	12.0
Rubric construction and audit	0.80	0.35	2.0
Recognition	0.60	0.15	1.8
Original-evidence checks, including competition	9.70	0.80	14.1
Edit construction and validity audits	3.10	0.40	3.0
Counterfactual verification (P/W)	1.50	0.10	1.3
Total	20.50	2.30	34.2

Cost distribution and marginal overhead. Full averages 22.8K logical tokens and 34.2 calls per query. By stage (Table 27), original-evidence checks account for 10.50K tokens, or 46.1% of the total, while counterfactual verification accounts for 1.60K tokens, or 7.0%. Relative to w/o counterfactual verification, adding counterfactual verification increases mean tokens by 1.60K (7.5%) and calls by 1.3. This marginal overhead does not imply low total cost: Full uses about 3.65 times SeCon-RAG's tokens and 21.9 times VanillaRAG's under the same evaluation scope. The stage means give $20{,}500/6{,}000 + 2{,}300/80 + 34.2 \times 0.12 = 36.27$ seconds under the stated service-time model. The 1.3 mean response calls reflect averaging over queries with early exits and different relation counts rather than a fixed per-query response budget.

H COMPLETE-SUPPORT FORMALISM

For the question-derived conjunction in Eq. (2), let β assign values to the answer variable y and shared entities $\mathbf{z}$. Let $\mathcal{S}_{\ell,\beta}(E)$ denote the evidence in context E supporting $h_\ell(\beta(\mathbf{z}), \beta(y); \kappa_\ell)$. The complete positive assignments for answer a are

$$\mathcal{B}_a^+(E) = \{\beta : \beta(y) = a,\ \mathcal{S}_{\ell,\beta}(E) \neq \varnothing \text{ for every } \ell \in [L]\}\,, \tag{12}$$

where $[L] = \{1, \ldots, L\}$. Thus $\mathcal{B}_a^+(E)$ is nonempty exactly when E supports every required relation under a compatible assignment of the shared entities. In the director–birthplace example, two individually supported relations do not constitute complete support if they concern different people. Positive support and opposition are recorded separately: contradictory evidence does not erase a positive assignment, but prevents that support from being unopposed. This distinction also matters during withdrawal auditing: any remaining complete positive support, even if contested, invalidates a withdrawal view.

H.1 A RESPONSE-PATTERN PROPERTY

Proposition 1 (Rejection of view-invariant endorsement) *Fix q, a_c, $\mathcal{T}_q$, and a frozen, audit-accepted view pair for relation ℓ. If the verifier's labels are determined solely by the text retained in both views, then $J(E_\ell^+) = J(E_\ell^-)$. The required pattern $J(E_\ell^+) = \mathrm{S}$, $J(E_\ell^-) = \mathrm{N}$ in Eq. (3) therefore cannot hold, and the update is rejected.*

Proof. Under the assumption, both views present the verifier with the same label-determining text, so $J(E_\ell^+) = J(E_\ell^-)$. Since $\mathrm{S} \neq \mathrm{N}$, the required pattern fails. □

This property concerns labels on audit-accepted views: every adopted update satisfies the response pattern for every relation in the frozen rubric. It does not show that the support change caused the label change, since other differences between the views may matter, nor does it certify factual truth, since coherent false evidence can satisfy the same conditions.