Zero-Trust Retrieval: Active Falsification Against Stealth Poisoning in Retrieval-Augmented Generation

Anonymous EMNLP submission

Abstract

Automated laboratories and auto-research systems increasingly rely on retrieved literature to generate hypotheses, plan experiments, and analyze results, making literature reliability a prerequisite for automated research workflows. When erroneous numerical values, experimental conclusions, or comparative claims enter the retrieval context, a Retrieval-Augmented Generation (RAG) system may propagate them as credible evidence through downstream research decisions. To mitigate this risk, this paper proposes **Active Falsification RAG (AF-RAG)**, which converts verifiable claims in the initial context into audit targets and retrieves candidate counter-evidence through targeted falsification queries. To evaluate AF-RAG systematically, this paper formalizes a local ranking-manipulation boundary condition combining an information cocoon with stealth poisoning: authentic corrective documents are excluded from the initial Top-K retrieval window, while poisoned documents remain locally self-consistent through micro-tampering. On a corpus of 2,641 ArXiv papers, AF-RAG reduces PAR from 94.5% to 59.4% and increases TRR from 11.2% to 58.0% across 1,200 stealth-poisoned samples; on unpoisoned queries, it achieves 79.9%.Code and data: https://anonymous.4open.science/r/AF-RAG-6708.

1 Introduction

Scientific discovery has traditionally relied on human researchers reading background literature, formulating hypotheses, testing them via experiments or computation, and revising subsequent directions according to the results. Automated laboratories and auto-research systems are beginning to automate parts of this cycle: external knowledge can be acquired, candidate hypotheses can be generated, experimental, computational, or analytical tools can be invoked, and feedback can be used to update later directions. Recent retrieval-augmented literature synthesis (Lála et al., 2023), scientific question-answering (Asai et al., 2026), and multi-agent discovery systems (Ghafarollahi and Buehler, 2025; Ghareeb et al., 2026; Gottweis et al., 2026) indicate that external literature is becoming an important input to automated research decisions.

Literature content determines the knowledge boundary and search direction of the auto-research loop. Prior findings, candidate mechanisms, datasets, metric definitions, and method comparisons define the space for hypothesis and idea generation; numerical results, experimental conclusions, and ablation analyses further influence experimental prioritization and result interpretation. Consequently, literature reliability directly affects whether automated research systems can generate reliable hypotheses and iterate in valid directions.

However, scholarly literature is not fully reliable. A *Nature* survey of 1,576 researchers reported that more than 70% of respondents had failed to reproduce another scientist's experiment (Baker, 2016); the Retraction Watch/Crossref database has recorded more than 50,000 retractions across multiple forms of error and misconduct (Retraction Watch, 2024; Crossref, 2024). Human research workflows can partially absorb such risks through expert reading and later correction. In contrast, automated research systems may directly use erroneous numerical values or pre-retraction conclusions for hypothesis generation and experimentation, thereby amplifying invalid experiments and erroneous iterative updates. Standard RAG primarily optimizes relevance rather than truthfulness; when the Top-K window contains only locally self-consistent erroneous literature and no authentic counter-evidence, filtering, aggregation, uncertainty-triggered retrieval, and multi-agent checking may lack effective signals.

Therefore, this work proposes **Active Falsification RAG (AF-RAG)** for literature-poisoning scenarios. AF-RAG does not treat the initial re-

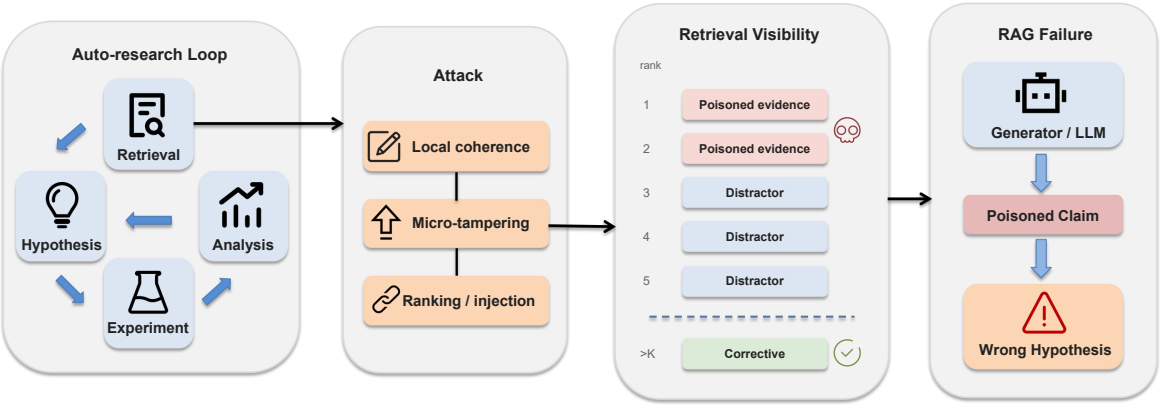

Figure 1: Information-cocoon threat in literature-grounded auto-research. Poisoned but locally coherent evidence is promoted into the Top-K retrieval window, while corrective evidence is buried beyond Top-K. Standard RAG lacks visible counter-evidence and may propagate poisoned claims into downstream hypotheses.

trieval context as fully trusted; instead, numerical values, comparative relations, and conclusive factual statements are converted into verifiable claims, and targeted falsification queries retrieve candidate counter-evidence beyond the initial window. To evaluate this mechanism, this paper defines a RAG poisoning setting combining an information cocoon with stealth poisoning: locally self-consistent poisoned documents are promoted into the Top-K window through micro-tampering and ranking manipulation, whereas authentic corrective documents are displaced to deeper ranks within the corpus. Experimental results show that AF-RAG reduces poison adoption, improves truth recovery for buried evidence, and preserves benign-query utility. Overall, this work formulates literature reliability in automated research as an evaluable RAG security problem and provides an active evidence-recovery mechanism for scientific agents.

2 Related Work

2.1 Automated Research Systems and Literature-Grounded Scientific Agents

Automated research systems are extending scientific discovery from isolated question answering to iterative workflows. OpenScholar and PaperQA retrieve literature to produce citation-supported scientific answers, while SciAgents, Robin, and Co-Scientist integrate literature or structured knowledge into multi-agent discovery for materials science, drug repurposing, and biomedical research (Asai et al., 2026; Lála et al., 2023; Ghafarollahi and Buehler, 2025; Ghareeb et al., 2026; Gottweis et al., 2026). These systems make literature reliability a security prerequisite for automated research.

2.2 RAG, Knowledge Poisoning, and Evidence Misplacement

RAG shifts part of the factual basis from model parameters to the retrieval pipeline, making corpus construction, ranking, chunking, and context selection security-relevant input channels (Karpukhin et al., 2020; Lewis et al., 2020; Gao et al., 2023). Knowledge poisoning exploits this channel by injecting or promoting documents that enter Top-K and are adopted as evidence; prior work shows that a small number of adversarial passages, injected documents, or triggers can steer downstream generation (Zhong et al., 2023; Zou et al., 2025; Xue et al., 2024). Recent attacks and benchmarks further show that limited poisoning can cause strong corruption and that evaluation must distinguish initial-context compromise from broader corpus contamination (Zhang et al., 2025a,c; Wang et al., 2025). This paper focuses on evidence misplacement in automated research: authentic evidence may be buried by ranking, while locally self-consistent erroneous literature directly drives downstream hypothesis formation, experimental prioritization, and analysis. This setting differs from ordinary evidence conflict because the model-visible context can appear coherent even when the corpus still contains corrective evidence outside the retrieval window (Longpre et al., 2021).

2.3 Defenses, Active Retrieval, and Claim Verification

Existing RAG defenses intervene after retrieval through adversarial training, filtering, aggregation or isolation, forensic traceback, and multi-agent checking (Fang et al., 2024; Edge et al., 2024;

Cheng et al., 2025; Zhou et al., 2025; Kim et al., 2025; Xiang et al., 2024; Zhang et al., 2025b; Du et al., 2023). They often require detectable anomalies, visible clean evidence, or post-hoc source access; in an information cocoon, locally coherent poisoned evidence may bypass these signals. Active methods like FLARE, CRAG, and Self-RAG address incomplete context, but fundamentally rely on uncertainty, retrieval evaluation, or self-reflection (Jiang et al., 2023; Yan et al., 2024; Asai et al., 2024). Irrelevant-context and counterfactual-noise work show that retrieved evidence can mislead generation (Shi et al., 2023; Hong et al., 2024). AF-RAG instead borrows the claim-verification view of FEVER, SciFact, FActScore, and RAGAS (Thorne et al., 2018; Wadden et al., 2020; Min et al., 2023; Es et al., 2024): extract audit targets from the initial context and redirect retrieval to falsify beyond Top-K.

3 Problem Formulation

The RAG poisoning setting studied in this paper combines two independent but synergistic mechanisms. An *information cocoon* restricts model-visible evidence by excluding authentic corrective documents from the initial retrieval window. On the other hand, *Stealth poisoning* reduces the detectability of adversarial documents by modifying factual content while preserving local coherence. When both mechanisms hold, the defense system loses in-context counter-evidence and textual anomaly features simultaneously.

This work focuses on query-level local attacks rather than corpus-wide compromise. The corpus D is treated as a controlled scholarly boundary: authentic documents remain available, but the attacker may influence the initial retrieval window for a target query through document construction, injection, or ranking manipulation.

3.1 Information Cocoon

An information cocoon occurs when poisoned evidence enters the initial retrieval window while authentic corrective evidence remains outside the model-visible context. Given corpus D and retriever R, the initial retrieval context is

$$C_0 = R_k(q, D). \tag{1}$$

The cocoon condition holds when $\mathrm{TopKPoisonCoverage}(q) > 0$ and $\mathrm{TopKTruthCoverage}(q) = 0$, while at least one authentic corrective document $d^+ \in D \setminus C_0$ remains outside the initial context. A standard RAG system then treats C_0 as its primary factual basis despite the existence of corrective evidence at deeper ranks. Unlike explicit knowledge-conflict settings, this setting deprives consistency-based detection of a visible reference baseline.

3.2 Stealth Poisoning (Micro-Tampering)

Stealth poisoning applies fine-grained factual modifications without overt instruction injection, garbled text, or self-contradictory content. Typical targets include numerical metrics, performance comparisons, dataset sizes, ranking results, statistical significance descriptions, and ablation conclusions; for example, a reported improvement may be changed from 15.0% to 11.5%, or a method comparison may be reversed.

The defining property is *local self-consistency*: tampered claims remain grammatically and narratively coherent within the same document. Surface-anomaly, fluency, and intra-document consistency checks are therefore unreliable. Unlike traditional prompt injection, micro-tampering operates at the *evidence layer* by substituting factual content inside model-visible evidence. Appendix A provides the comprehensive operation categories and a side-by-side example for clarity.

3.3 Task Definition and Evaluation Metrics

Given query q, initial context C_0, and controlled corpus D, the system must generate answer y when C_0 may be compromised. The objective is to avoid adopting poisoned claims and, when possible, recover authentic evidence buried at deeper ranks. The controlled corpus boundary limits attacker control but does not imply that every document in D is independently verified; adjudication occurs over specific claims using retrieved candidate evidence.

System behavior is measured with security and utility metrics. *Poisoning Attack Rate* (PAR) is the proportion of samples whose final output adopts the poisoned numerical value, comparison, or conclusion; lower PAR indicates stronger attack suppression. *Truth Recovery Rate* (TRR) is the proportion of samples whose final output recovers the authentic claim despite a poisoned initial context. For unpoisoned queries, *Benign Accuracy* measures agreement with the ground-truth answer, and *False Conflict Rate* (FCR) measures erroneous conflict-marker emission. Reporting all four metrics separates attack suppression from normal-task utility.

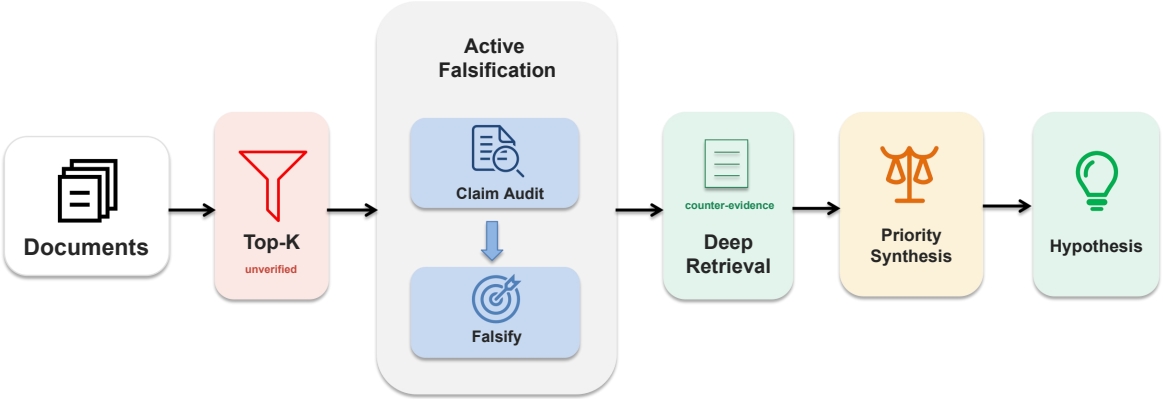

Figure 2: AF-RAG architecture. Initial Top-K evidence is treated as unverified context. The active falsification module audits verifiable claims and generates targeted falsification queries, which drive deep retrieval beyond the initial window. The evidence-prioritized synthesizer then compares the initial context with recovered counter-evidence and gives higher priority to falsification-retrieved evidence when conflicts arise.

4 Method

AF-RAG is designed as an active falsification framework for stealth-poisoning scenarios. Given a user query q, an initial retrieval context $C_0 = \{d_1, \ldots, d_k\}$, and a controlled offline literature corpus D, the framework does not treat C_0 as fully trusted. Instead, claims in C_0 that are relevant to the query are promoted to audit targets. The system outputs a final answer y and emits an optional conflict marker when the initial context and deep-retrieved evidence disagree on key facts.

The pipeline contains four stages: a claim auditor, a falsification query generator, a deep retriever, and an evidence-prioritized synthesizer. Figure 2 summarizes this architecture. This decomposition shifts the defense trigger from model uncertainty or textual anomaly to claim-level verifiability; locally coherent poisoned text is therefore audited when it contains numerical values, comparative relations, or experimental conclusions.

4.1 Claim Auditor

The claim auditor identifies whether the initial context contains factual statements that require external verification. Given query q and initial context C_0, the auditor produces an audit decision a and a set of verifiable claims $\mathcal{C}$:

$$(a, \mathcal{C}) = A(q, C_0), \quad \mathcal{C} = \{c_i\}_{i=1}^{m}. \tag{2}$$

When the user intent corresponds to summarization, author lookup, section listing, or other tasks that do not require factual adjudication, external auditing is not triggered. When the query requires specific factual answers, text spans containing performance metrics, percentages, rankings, method comparisons, dataset sizes, or determinate conclusions are extracted as audit targets. In this context, an "anomaly" does not denote an internal contradiction in the document; rather, it denotes that a claim has independent verification value.

This design addresses a core failure mode of stealth poisoning. Poisoned text may be fluent, narratively complete, and free of explicit single-document contradictions, yet it can still be recognized by the claim auditor as verifiable content and passed to the subsequent falsification process. AF-RAG therefore defines verifiability as the audit entry point, instead of treating internal inconsistency as a prerequisite. The auditor also preserves the source span and trigger rationale of each claim, allowing the query generation stage to inherit entity, metric, and contextual constraints.

4.2 Falsification Query Generator

The falsification query generator converts each audited claim into one or more targeted retrieval queries. Its objective is not to restate the original user question, but to retrieve evidence capable of supporting, refuting, or constraining the claim. Formally, for claim set $\mathcal{C}$, the generator produces

$$\mathcal{Q}_f = G(q, \mathcal{C}) = \{q_{i,j}^{f} \mid c_i \in \mathcal{C},\ j = 1, \ldots, r\}. \tag{3}$$

In the implementation evaluated in Section 5, $r = 3$: the generator produces one high-precision query, one broader contextual query, and one natural-language semantic query for each audited claim.

Each falsification query preserves the entities,

metrics, and comparative relations in the original claim while adding retrieval cues specifically oriented toward verification. For example, if a poisoned context states that a method improves performance by 11.5%, the generated query may target the original paper title, the relevant metric name, the reported improvement, and the surrounding experimental setting. If a comparative conclusion is modified, the query emphasizes both competing methods and the evaluation dimension. The generated queries are therefore claim-conditioned rather than merely query-expanded.

This stage differs from generic query rewriting. Conventional rewriting aims to improve semantic coverage of the original question, whereas falsification queries aim to test the truth value of a specific claim. Consequently, the generated queries encode not only the underlying user intent but also the numerical value, comparison, or conclusion to be checked. This redirection is critical in an information cocoon: authentic evidence may not be retrieved by the original Top-K query, but may become retrievable through targeted queries constructed around the tampered claim.

4.3 Deep Retriever

The deep retriever executes the falsification queries against the controlled offline literature corpus and returns candidate evidence beyond the initial context. Given falsification query set $\mathcal{Q}_f$, the retrieval result is defined as

$$C_f = R_{K'}(\mathcal{Q}_f, D \setminus \mathrm{ID}(C_0)). \tag{4}$$

Here, $\mathrm{ID}(C_0)$ denotes the document identifiers already included in the initial retrieval window, and K' denotes the per-query deep-retrieval window. In all main experiments, $K' = 20$; after deduplication and round-robin merging across falsification queries, at most six deep-retrieved chunks are forwarded to the synthesizer. Excluding the initial identifiers prevents the system from repeatedly retrieving the same poisoned or cocooned documents and increases the chance of reaching evidence suppressed below the initial Top-K.

Under the threat model defined in Section 3, the attacker's effective adversarial control is confined to the designated Top-K window; evidence retrieved beyond this boundary is therefore inherently assigned higher evidential priority, without being treated as source-level authenticity proof.

In implementation, multiple falsification queries may be issued for each audited claim. Retrieved documents are deduplicated by document identifier, and interleaved sampling is used to avoid domination by a single query variant. Candidate evidence is retained when it shares relevant entities, metrics, or experimental settings with the audited claim. This retrieval path is intended to recover evidence that remains present in the corpus but is hidden beyond the initial retrieval window.

4.4 Evidence-Prioritized Synthesizer

The evidence-prioritized synthesizer generates the final answer by comparing the initial context with the deep-retrieved candidate evidence. The synthesis process is expressed as

$$y = S(q, C_0, C_f). \tag{5}$$

If deep retrieval is not triggered, the system falls back to standard synthesis over the initial context. If deep-retrieved evidence is available, the initial context is marked as unverified context, while candidate evidence obtained through falsification queries is marked as high-priority scholarly evidence. The synthesizer then compares key numerical values, core facts, and conclusive claims across the two evidence sets.

Conflict is rigorously determined at three distinct levels: numerical mismatch, reversed comparative relation, and incompatible experimental conclusion. The decision is scoped to the audited claim, so unrelated background disagreement does not automatically trigger conflict. When no conflict is observed, both evidence sources may be integrated. When a weak conflict is present, the answer reports the discrepancy and avoids unsupported aggregation. When a strong conflict directly concerns the audited claim, priority is assigned to the deep-retrieved candidate evidence and the conflict is explicitly marked. This behavior is designed for the information-cocoon setting: the initial context may be coherent but adversarially selected, whereas deep evidence is retrieved through claim-specific falsification queries over the controlled corpus. The conflict marker is therefore not a generic uncertainty indicator, but a signal that the initial retrieval context is contradicted by higher-priority candidate evidence. This design directly targets the failure mode in which standard RAG confidently adopts a poisoned claim because no corrective evidence is visible within its initial context.

5 Experiments

5.1 Dataset and Poisoning Construction

Experiments are conducted on a controlled offline corpus containing 2,641 ArXiv papers. Each poisoned evaluation instance contains a user query, an initial retrieval context, a poisoned document, and an authentic corrective document. The poisoned documents are produced through fine-grained modifications to numerical metrics, comparative relations, and experimental conclusions, while preserving grammaticality, narrative coherence, and local semantic consistency. No explicit prompt injection string or overt anomaly pattern is inserted. A 200-sample human validation subset was double-annotated for construction correctness and stealthiness, with 0.92 inter-annotator consistency.

Evaluator reliability is audited separately. A fixed subset of 200 answer-level outcomes is manually annotated and compared with the automatic PAR/TRR judge, yielding Cohen's $\kappa = 0.87$. This check validates evaluator consistency rather than source authenticity.

The main evaluation uses an *only-poison cocoon* setting. In this setting, the poisoned document enters the initial Top-K retrieval context, whereas the authentic corrective document remains in the corpus but is excluded from the model-visible context. This setting is not intended to model full-corpus compromise; it isolates a local ranking-manipulation boundary condition in which the initial evidence is adversarially selected while corrective evidence remains reachable in a controlled corpus. A separate RAGuard fair setting is also evaluated, where poisoned and authentic contrastive documents are both placed in the context together with benign distractors. This supplementary setting is used only to assess an isolation-based defense when comparable evidence is available. Additional subset sizes and construction details are summarized in Appendix A.

5.2 Baselines

AF-RAG is compared with eight RAG baselines in the poisoned setting. Naive RAG denotes standard retrieval-augmented generation with the initial Top-K context and no defensive prompt; the terms Vanilla RAG and Vanilla LLM are not used in the main comparison. GraphRAG evaluates the effect of graph-structured evidence organization and aggregation. RAGuard denotes a leave-one-out isolation and filtering defense. FLARE-style RAG triggers additional retrieval from low-confidence generation states. Blind Retry RAG repeats retrieval without claim-directed targeting. Retrieve-More RAG expands the retrieval window. Query-Rewrite RAG applies generic query rewriting to the original user query. Claim-Only RAG uses the audited claim as the retrieval entry point, but does not include the full falsification query generation and evidence-prioritized synthesis pipeline. A non-retrieval LLM baseline is excluded from the main comparison because it is not exposed to retrieval-context poisoning risks.

Additional baselines are included only for the benign-utility analysis. Naive+Prompt augments Naive RAG with a cautionary conflict-detection prompt but does not perform additional retrieval. Multi-Agent uses separate generation and checking agents over the retrieved context. RAGDefender and ReliabilityRAG instantiate filtering- or reliability-scoring defenses from prior RAG robustness work, evaluated here as utility-oriented comparators. These systems are reported in Table 2 to characterize normal-query degradation and false alarms, rather than as primary only-poison cocoon baselines. Appendix B reports the baseline retrieval budgets, query sources, triggers, and synthesis rules used for implementation fidelity.

The metrics follow the established task definition in Section 3. Table 1 comprehensively reports PAR and TRR for the poisoned setting, while Table 2 reports benign accuracy and false-conflict behavior for subsequent unpoisoned queries.

Architecture	PAR ↓	ΔPAR	TRR ↑	ΔTRR
Naive RAG	94.5%	0.0	11.2%	0.0
GraphRAG	95.6%	+1.1	30.1%	+18.9
RAGuard	94.3%	-0.2	11.4%	+0.2
FLARE-style	95.9%	+1.4	9.3%	-1.9
Blind Retry	91.8%	-2.7	15.2%	+4.0
Retrieve-More	88.1%	-6.4	19.7%	+8.5
Query-Rewrite	87.5%	-7.0	20.1%	+8.9
Claim-Only	84.8%	-9.7	22.6%	+11.4
AF-RAG	**59.4%**	**-35.1**	**58.0%**	**+46.7**

Table 1: Main security results under the only-poison cocoon setting (N=1200). Delta columns report percentage-point changes relative to Naive RAG.

5.3 Main Results under Stealth Poisoning

Table 1 reports the main robustness results under the only-poison cocoon setting. Naive RAG reaches a PAR of 94.5% and a TRR of 11.2%, indicating that standard RAG tends to adopt the visible

Architecture	Acc. ↑	ΔAcc.	Signal
Naive RAG	63.7%	0.0	–
GraphRAG	67.8%	+4.1	–
Naive+Prompt	64.0%	+0.3	FCR 12.8%
Multi-Agent	68.8%	+5.1	FCR 3.0%
RAGDefender	61.1%	-2.6	–
ReliabilityRAG	62.4%	-1.3	–
RAGuard	2.5%	-61.2	FRR 96.6%
AF-RAG	**79.9%**	**+16.2**	FCR 16.3%

Table 2: Benign-query utility results (N=1200). FCR denotes false conflict rate; FRR denotes RAGuard false rejection rate.

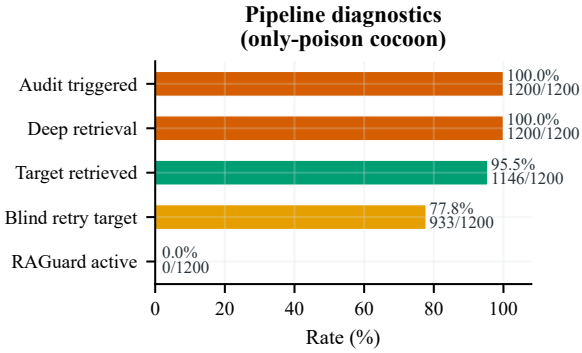

Figure 3: Pipeline diagnostics under the only-poison cocoon setting. AF-RAG always triggers claim auditing and deep retrieval for factual poisoned queries and retrieves the authentic target evidence in 95.5% of samples, exceeding blind retry retrieval and exposing why isolation-based RAGuard remains inactive when authentic contrastive evidence is absent from the initial context.

erroneous claim when authentic counter-evidence is absent from the initial retrieval window. AF-RAG reduces PAR to 59.4% and increases TRR to 58.0%. Relative to Naive RAG, this corresponds to a 35.1 percentage-point reduction in PAR and a 46.7 percentage-point improvement in TRR. AF-RAG is therefore a partial mitigation and evidence-recovery mechanism rather than a complete poisoning defense: the remaining 59.4% PAR shows that poisoned claims are still retained in a majority of poisoned data samples.

Table 2 reports the benign utility evaluation. AF-RAG achieves 79.9% accuracy, compared with 63.7% for Naive RAG, 67.8% for GraphRAG, 68.8% for Multi-Agent, and 64.0% for Naive+Prompt. However, AF-RAG emits a conflict marker on 16.3% of benign queries, which is higher than Multi-Agent at 3.0% and Naive+Prompt at 12.8%. This trade-off suggests that active evidence recovery improves answer quality but still requires a more conservative conflict calibration. Practically, AF-RAG is better suited to high-risk factual verification than to always-on deployment for low-risk summarization or bibliographic queries. This pattern motivates selective activation rather than universal deployment across all query types.

The baselines expose distinct performance patterns. GraphRAG improves TRR to 30.1%, but its PAR remains 95.6%, suggesting that structured aggregation can increase partial coverage of authentic information without preventing poisoned conclusions from being retained when the initial context dominates. RAGuard behaves similarly to Naive RAG, with 94.3% PAR and 11.4% TRR; its activation rate is 0/1200 because the only-poison cocoon setting contains no authentic contrastive passage for leave-one-out comparison. FLARE-style RAG obtains a TRR of 9.3%, below Naive RAG, showing that locally self-consistent and inherently determinate poisoned text does not reliably trigger low-confidence retrieval. In contrast, Retrieve-More, Query-Rewrite, and Claim-Only raise TRR to 19.7%, 20.1%, and 22.6%, respectively, but remain far below AF-RAG. Thus, generic retrieval expansion, rewriting, and direct claim search are highly insufficient substitutes for the comprehensive active falsification pipeline.

Figure 3 distinctly separates active falsification from simple retrieval repetition. AF-RAG reaches a target evidence hit rate of 95.5%, whereas Blind Retry reaches 77.8%. Claim-conditioned falsification queries therefore recover corrective evidence omitted from the initial model-visible context substantially more reliably than repeated retrieval without an explicitly defined audit target. Appendix E further reports that the authentic corrective document appears within the top five results of at least one falsification query in 92.17% of samples and within the top 20 in 95.50%. RAGuard remains inactive in this setting because the initial context provides no authentic contrastive evidence. In the supplementary fair setting where poisoned and authentic contrastive documents are both visible, RAGuard reduces PAR to 45.3%, outperforming AF-RAG's 64.1% PAR in poisoned-claim suppression, while AF-RAG retains higher TRR (56.8% vs. 34.8%); full details are reported in Appendix D.

5.4 Diagnostic Findings

Retrieve-More, Query-Rewrite, and Claim-Only are included as diagnostic baselines to identify the source of AF-RAG's gains. Their TRR values are

19.7%, 20.1%, and 22.6%, respectively, all substantially below AF-RAG at 58.0%. The improvement is therefore not explained by retrieval-window expansion, generic query rewriting, or direct claim retrieval alone. Instead, the results in Figure 3 point to the complementary role of claim auditing, falsification query construction, deep retrieval, and evidence-prioritized synthesis.

Systematic component ablations identify the bottleneck. Removing the claim auditor causes a moderate overall degradation, whereas removing the deep retriever collapses the system to 94.0% PAR and 10.0% TRR, close to Naive RAG. Removing the evidence-prioritized synthesizer lowers TRR to 22.2%, showing that retrieved corrective evidence still requires explicit priority control during subsequent synthesis. Full ablation protocols and results are fully reported in Appendix F.

The noisy-versus-stealth comparison evaluates the effect of attack detectability on two diagnostic subsets ($N = 50$ per subset). On Dataset 1 (noisy), Naive RAG obtains 74.0% PAR, whereas AF-RAG obtains 38.0% PAR and 72.0% TRR. On Dataset 2 (stealth), Naive RAG rises to 94.0% PAR and falls to 8.0% TRR, while AF-RAG obtains 48.0% PAR and 62.0% TRR. Stealth poisoning therefore increases attack strength by weakening surface anomaly and contextual conflict signals, whereas active falsification preserves a comparatively high truth recovery rate.

Paired significance tests confirm the main effect sizes. AF-RAG improves PAR-avoidance by 35.08 percentage points over Naive RAG and improves TRR by 46.75 percentage points, with both effects significant under McNemar tests. Claim-Only, Query-Rewrite, and Retrieve-More are also significantly better than Naive RAG, but their effect sizes are substantially smaller than AF-RAG. Full paired-test details are reported in Appendix G.

5.5 Cost and Error Analysis

The computational cost of AF-RAG primarily arises from claim auditing and deep retrieval. Average end-to-end wall-clock latency increases from 1.02 to 5.69 seconds on factual poisoned queries and from 1.07 to 3.36 seconds on benign queries; zero-trust auditing is triggered on 100.0% of factual poisoned queries and 50.0% of benign queries. Full latency measurements are reported in Appendix H.

Residual errors mainly occur after falsification-query recall rather than during audit triggering. The authentic target document is reachable by falsification queries in 1,146/1,200 samples (95.5%), but the final TRR is 58.0%. Among the 696 truth-recovered AF-RAG outputs, 250 also retain the poisoned claim, indicating that partial truth recovery does not always imply complete poisoned-claim suppression. Among the 504 samples in which truth is not recovered, 467 involve cases where the authentic target is reachable but the final answer still fails to recover the truth. The primary engineering bottleneck is therefore critical evidence selection and collaborative synthesis, including snippet-length control, numerical evidence alignment, cross-document experimental-setting matching, and conflict adjudication. The full failure taxonomy is provided in Appendix I.

Failure type	Count	Failure share
Truth not retrieved	37	7.34%
Truth retrieved but synthesis chose poison	435	86.31%
Truth retrieved but answer not recovered	32	6.35%

Table 3: Compact AF-RAG truth-recovery failure breakdown. The dominant remaining error occurs after the authentic evidence has already been retrieved.

PAR and TRR are evaluated as independent answer-level judgments rather than as mutually exclusive labels. Appendix I therefore reports a contingency table between poisoned-claim adoption and truth recovery, clarifying cases where an answer includes the authentic fact but still inadvertently retains the poisoned claim.

6 Conclusion

Automated research workflows depend on retrieved literature, but locally coherent poisoned evidence can distort hypothesis generation when corrective evidence is buried beyond the initial retrieval window. This paper introduced AF-RAG, an active falsification framework that promotes verifiable contextual claims to audit targets, retrieves candidate counter-evidence beyond Top-K, and prioritizes recovered evidence during synthesis. Under the only-poison cocoon setting, AF-RAG reduces PAR from 94.5% to 59.4%, increases TRR from 11.2% to 58.0%, reaches a 95.5% target-evidence hit rate, and achieves 79.9% benign accuracy. These results show that literature reliability in automated research can be formulated as an evaluable RAG security problem, while the 16.3% benign false-conflict rate and added latency indicate the need to balance security, utility, and cost.

7 Limitations

AF-RAG has four main limitations. First, corrective evidence must exist in the controlled corpus and be reachable by falsification queries; otherwise, the initial poisoned claim may still be adopted; a preliminary retriever-swap diagnostic also indicates that the current implementation is stronger with BM25 than with a dense all-MiniLM retriever. Second, AF-RAG does not defend against attackers who can poison both the initial context and the deeper retrieval pool. Third, deep-retrieved evidence receives higher priority only under the stated threat model and does not provide source-level authenticity proof. Fourth, synthesis can still fail when numerical differences are small, metric definitions differ, or retrieved snippets lack sufficient experimental context. Future work should improve numerical evidence alignment, metric normalization, and conflict adjudication.

8 Ethical Considerations

The information cocoon and stealth poisoning attacks discussed in this paper are intended solely for defensive security research. Attack scripts that can be directly used to automatically generate stealthily poisoned documents, manipulate retrieval rankings, or contaminate corpora at scale should not be publicly released. The experimental data and code are intended to support evaluation and improvement of factual robustness in RAG systems, rather than to provide operational attack tooling. Released code and derived evaluation artifacts are provided for anonymous review and defensive research use; redistribution of the underlying ArXiv papers follows their original source terms.

References

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'Arcy, David Wadden, et al. 2026. Synthesizing scientific literature with retrieval-augmented language models. *Nature*, 650(8103):857–863.

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

Monya Baker. 2016. 1,500 scientists lift the lid on reproducibility. *Nature*, 533:452–454.

Zirui Cheng, Jikai Sun, Anjun Gao, Yueyang Quan, Zhuqing Liu, Xiaohua Hu, and Minghong Fang. 2025. Secure retrieval-augmented generation against poisoning attacks. *arXiv preprint arXiv:2510.25025*.

Crossref. 2024. Retraction watch. https://www.crossref.org/documentation/retrieve-metadata/retraction-watch/. Accessed: 2026-05-24.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2024. RAGAS: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158.

Feiteng Fang, Yuelin Bai, Shiwen Ni, Min Yang, Xiaojun Chen, and Ruifeng Xu. 2024. Enhancing noise robustness of retrieval-augmented language models with adaptive adversarial training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

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

Alireza Ghafarollahi and Markus J. Buehler. 2025. SciAgents: Automating scientific discovery through bioinspired multi-agent intelligent graph reasoning. *Advanced Materials*, 37(22):2413523.

Ali Essam Ghareeb, Benjamin Chang, Ludovico Mitchener, Angela Yiu, et al. 2026. A multi-agent system for automating scientific discovery. *Nature*.

Juraj Gottweis et al. 2026. Accelerating scientific discovery with Co-Scientist. *Nature*.

Giwon Hong, Jeonghwan Kim, Junmo Kang, Sung-Hyon Myaeng, and Joyce Jiyoung Whang. 2024. Why so gullible? enhancing the robustness of retrieval-augmented models against counterfactual noise. In *Findings of the Association for Computational Linguistics: NAACL 2024*.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6769–6781.

Minseok Kim, Hankook Lee, and Hyungjoon Koo. 2025. Rescuing the unpoisoned: Efficient defense against knowledge corruption attacks on RAG systems. *arXiv preprint arXiv:2511.01268*.

Jakub Lála, Odhran O'Donoghue, Aleksandar Shtedritski, Sam Cox, Samuel G. Rodriques, and Andrew D. White. 2023. PaperQA: Retrieval-augmented generative agent for scientific research. *arXiv preprint arXiv:2312.07559*.

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

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100.

Retraction Watch. 2024. The retraction watch database. http://retractiondatabase.org/. Accessed: 2026-05-24.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *Proceedings of the 40th International Conference on Machine Learning*, pages 31210–31227.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 809–819.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 7534–7550.

Haowei Wang, Rupeng Zhang, Junjie Wang, Mingyang Li, Yuekai Huang, Dandan Wang, and Qing Wang. 2025. Joint-GCG: Unified gradient-based poisoning attacks on retrieval-augmented generation systems. *arXiv preprint arXiv:2506.06151*.

Chong Xiang, Tong Wu, Zexuan Zhong, David Wagner, Danqi Chen, and Prateek Mittal. 2024. Certifiably robust RAG against retrieval corruption. *arXiv preprint arXiv:2405.15556*.

Jiaqi Xue, Mengxin Zheng, Yebowen Hu, Fei Liu, Xun Chen, and Qian Lou. 2024. BadRAG: Identifying vulnerabilities in retrieval augmented generation of large language models. *arXiv preprint arXiv:2406.00083*.

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

Baolei Zhang, Yuxi Chen, Minghong Fang, Zhuqing Liu, Lihai Nie, Tong Li, and Zheli Liu. 2025a. Practical poisoning attacks against retrieval-augmented generation. *arXiv preprint arXiv:2504.03957*.

Baolei Zhang, Haoran Xin, Minghong Fang, Zhuqing Liu, Biao Yi, Tong Li, and Zheli Liu. 2025b. Traceback of poisoning attacks to retrieval-augmented generation. *arXiv preprint arXiv:2504.21668*.

Baolei Zhang, Haoran Xin, Jiatong Li, Dongzhe Zhang, Minghong Fang, Zhuqing Liu, Lihai Nie, and Zheli Liu. 2025c. Benchmarking poisoning attacks against retrieval-augmented generation. *arXiv preprint arXiv:2505.18543*.

Zexuan Zhong, Ziqing Huang, Alexander Wettig, and Danqi Chen. 2023. Poisoning retrieval corpora by injecting adversarial passages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13764–13775.

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

Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. 2025. PoisonedRAG: Knowledge corruption attacks to retrieval-augmented generation of large language models. In *Proceedings of the 34th USENIX Security Symposium*.

Claim type	Edit operation	Example
Numerical metric	Value substitution	15.0% → 11.5%
Comparison	Relation reversal	A outperforms B → B outperforms A
Conclusion	Polarity weakening or reversal	improves robustness → shows marginal gain
Ablation	Component interpretation change	module helps → module hurts

Table 4: Micro-tampering operations used to construct stealth-poisoned evidence.

Authentic claim	Micro-tampered claim
UCPOF improves average accuracy by 6.03% over few-shot baselines, boosts overall average accuracy by 5.75%, and cuts the average retrieval trigger rate by 50.66%.	UCPOF improves average accuracy by 8.15% over few-shot baselines, boosts overall average accuracy by 7.42%, and cuts the average retrieval trigger rate by 38.24%.

Table 5: Example of numerical micro-tampering. The poisoned claim remains locally fluent and plausible, but changes the factual values that a RAG answer may later adopt.

Setting	N	Purpose
Only-poison cocoon	1200	Main security
RAGuard fair setting	1200	Isolation defense
Noisy-vs-stealth	50/50	Detectability diagnosis
Ablation subset	500	Component diagnosis
Cost subset	20/20	Latency / trigger rate

Table 6: Evaluation subsets used in the appendix and main experiments.

Hyperparameter	Value
Initial retrieval Top-K	5
Falsification queries per claim r	3
Deep retrieval window K'	20
Maximum deep chunks sent to synthesizer	6
GraphRAG community summaries	3
Noisy-vs-stealth subset	50
Cost subset	20
Ablation subset	500

Table 7: Implementation hyperparameters used in the reported experiments.

A Dataset Construction

The experimental corpus contains 2,641 ArXiv papers organized as a controlled offline literature corpus. Each poisoned evaluation instance contains a user query, an initial retrieval context, a poisoned document, an authentic corrective document, and a target claim. The construction follows the local ranking-manipulation assumption used in the main threat model: authentic documents remain in the underlying corpus, but are excluded from the initial Top-K window in the only-poison cocoon setting, whereas poisoned documents are placed in the model-visible retrieval context.

Stealth poisoning is instantiated through micro-tampering rather than explicit prompt injection. The edited elements include numerical metrics, method comparisons, experimental conclusions, ablation interpretations, and task-performance descriptions. To preserve stealthiness, each edited passage is required to remain grammatical, locally coherent, and compatible with the surrounding narrative. The RAGuard fair setting differs from the only-poison cocoon setting by making both the poisoned document and the authentic contrastive document visible in the same input context.

To check construction quality without releasing operational attack prompts, two annotators independently inspected 200 sampled authentic–tampered claim pairs for factual correctness of the authentic claim and stealthiness of the tampered claim. The inter-annotator consistency was 0.92 over these binary judgments. This validation checks dataset construction consistency; it does not imply that every poisoned passage is indistinguishable from authentic text in arbitrary deployment settings.

Table 6 summarizes the evaluation subsets used throughout the experiments. The full only-poison and RAGuard fair settings contain 1,200 samples each, whereas the noisy-vs-stealth, ablation, and cost analyses are reported on smaller diagnostic subsets.

Table 7 lists the implementation hyperparameters used for the reported runs. Unless otherwise stated, the initial retrieval window is fixed at Top-$K = 5$, each audited claim is expanded into three falsification queries, and deep retrieval uses a window of $K' = 20$ before forwarding at most six deduplicated chunks to the synthesizer.

Table 8 reports the shared implementation stack used across the main experiments. These settings are held fixed unless a diagnostic experiment explicitly changes the corresponding parameter.

B Baseline and Hyperparameter Details

Naive RAG retrieves the initial Top-K context with the original query and generates directly from that context without a defensive prompt. GraphRAG evaluates graph-structured evidence organization and community-level aggregation. RAGuard applies leave-one-out isolation by removing candidate documents and checking whether the output changes. FLARE-style RAG triggers additional retrieval from low-confidence generation states. Blind Retry RAG repeats retrieval with the orig-

Component	Setting
LLM backbone	DeepSeek-Chat for claim auditing, falsification-query generation, synthesis, and answer-level judging
Corpus index	Local offline ArXiv corpus indexed with BM25 over cleaned paper text
Initial context	Poisoned primary excerpt plus BM25 retrieval with Top-$K = 5$; the authentic target document is excluded in the only-poison cocoon setting
Snippet construction	Retrieved background snippets are truncated to 1,200 characters; deep-retrieved evidence snippets are truncated to 1,500 characters
Deep retrieval	BM25 search over the same corpus, excluding identifiers already present in the initial context; results are deduplicated and round-robin merged across falsification queries
RAGuard embedding	SentenceTransformer `all-MiniLM-L6-v2` for isolation-defense similarity checks
GraphRAG setting	Three graph community summaries are supplied for GraphRAG baselines
Evaluation judges	PAR and TRR are judged independently with strict poisoned-adoption and truth-recovery prompts

Table 8: Implementation stack and shared settings for the reported experiments.

Method	Retrieval budget	Query source	Defense trigger	Synthesis rule
Naive RAG	Initial $K = 5$	User query	None	Standard context synthesis
Retrieve-More	Expanded retrieval window	User query	Always	Standard context synthesis
Query-Rewrite	Extra retrieval	Generic rewritten query	Always	Standard context synthesis
Claim-Only	Deep retrieval, $K' = 20$	Audited claim	Claim-verifiability	Standard claim-retrieval synthesis
AF-RAG	$K = 5$, $K' = 20$, $r = 3$	Falsification queries	Claim-verifiability	Evidence-prioritized synthesis
RAGuard	Initial context only	Initial context	Visible contrastive evidence	Leave-one-out isolation/filtering

Table 9: Baseline implementation budget and control-flow summary. r denotes the number of falsification queries generated per audited claim.

inal query and does not construct claim-directed queries. Retrieve-More RAG expands the retrieval window. Query-Rewrite RAG generates generic query rewrites. Claim-Only RAG uses the audited claim as the retrieval entry point, but does not include falsification-query construction or evidence-prioritized synthesis.

Additional systems are used only for benign utility. Naive+Prompt augments Naive RAG with a cautionary conflict-detection prompt. Multi-Agent uses separate generation and checking agents over the retrieved context. RAGDefender and ReliabilityRAG represent filtering- or reliability-scoring defenses. These methods are not treated as primary only-poison cocoon baselines.

C Prompt and Pipeline Constraints

AF-RAG is implemented as a sequence of defensive prompts and retrieval operations. The claim auditor identifies verifiable factual claims in the initial context, including numerical values, comparative statements, performance conclusions, dataset sizes, and determinate factual assertions. It does not require an internal contradiction to trigger auditing. The falsification query generator produces three query variants per audited claim: a high-precision query, a broader contextual query, and a natural-language semantic query. The deep retriever excludes document identifiers already present in the initial context and forwards at most six deduplicated evidence chunks to the synthesizer. The synthesizer treats the initial context as unverified and assigns higher priority to evidence retrieved through falsification queries under the threat model.

Operational attack-generation prompts are not included. The appendix describes only the defensive audit, retrieval, and synthesis constraints needed to reproduce the reported system behavior.

Decoding and judging protocol. All main reported runs use DeepSeek-Chat for claim auditing, falsification-query generation, synthesis, and answer-level judging. The revised deterministic rerun protocol sets temperature to 0.0 for claim auditing, falsification-query generation, evidence synthesis, baseline generation, and PAR/TRR judging. PAR and TRR are judged independently: PAR checks whether the final answer adopts the poisoned numerical value, comparison, or conclusion, while TRR checks whether the final answer states the authentic claim. To audit evaluator reliability, we manually annotated a fixed subset of 200 answer-level outcomes and observed strong agreement with the automatic judgments (Cohen's $\kappa = 0.87$).

D RAGuard Fair Setting

The only-poison cocoon setting evaluates defense behavior when authentic counter-evidence is absent from the initial context. This setting does not fully characterize defenses that require inter-document comparison. To avoid conflating this threat-model constraint with an implementation defect of RAGuard, Table 10 reports a separate fair setting in which each input context contains the poisoned document, the authentic source document, and benign distractors.

Architecture	PAR ↓	TRR ↑
Naive RAG	82.9%	24.1%
RAGuard	45.3%	34.8%
AF-RAG	64.1%	56.8%

Table 10: Supplementary fair-setting results for isolation-based defense when poisoned and authentic contrastive documents are both visible in the input context (N=1200).

Diagnostic	AF-RAG	Blind Retry	RAGuard
Audit / active	1200/1200 (100.0%)	–	0/1200 (0.0%)
Deep search	1200/1200 (100.0%)	–	–
Target hit	1146/1200 (95.5%)	933/1200 (77.8%)	–

Table 11: Pipeline diagnostics under the only-poison cocoon setting.

Table 10 shows that RAGuard reduces PAR from 82.9% to 45.3% when authentic contrastive documents are present, confirming that isolation-based filtering can suppress poisoned adoption when the required comparison evidence is available. AF-RAG obtains a higher PAR of 64.1% in this fair setting because it does not physically remove the poisoned passage before synthesis; instead, the synthesizer must adjudicate among the initial poisoned passage, the visible authentic passage, and deep-retrieved evidence. This makes poisoned-claim suppression less aggressive than RAGuard's deletion-based strategy, although AF-RAG still achieves higher answer-level truth recovery (56.8% TRR vs. 34.8%).

E Diagnostic Tables

Table 11 reports the pipeline diagnostics used to distinguish active falsification from repeated retrieval. Table 14 reports the isolation behavior of RAGuard under the fair setting in which contrastive authentic evidence is visible.

The diagnostics in Table 11 measure whether AF-RAG's gains are explained by the active falsification pathway rather than by repeated retrieval alone. They report audit activation, deep-search triggering, and target-evidence reachability under the only-poison cocoon setting.

E.1 Falsification-Query Rank Depth

Falsification-query rank depth measures the best rank of the authentic corrective document across the generated falsification queries for each sample. This diagnostic does not measure how deeply the authentic document is ranked under the original user query; instead, it measures whether AF-RAG's claim-conditioned queries make corrective evidence retrievable after the initial model-visible context has omitted it.

Best rank	Count	Share	TRR	PAR
1–5	1106	92.17%	60.58%	58.50%
6–20	40	3.33%	22.50%	82.50%
21–50	9	0.75%	33.33%	77.78%
51–100	10	0.83%	20.00%	80.00%
101–200	15	1.25%	26.67%	66.67%
> 200	5	0.42%	40.00%	40.00%
Not retrieved	15	1.25%	40.00%	40.00%

Table 12: Rank depth of the authentic corrective document under AF-RAG falsification queries (N=1200). Rank is computed as the best rank achieved across the generated falsification queries for each sample.

The authentic corrective document is retrieved within the top five results for 1,106/1,200 samples and within the top 20 results for 1,146/1,200 samples. Thus, the dominant residual bottleneck is not target-document reachability but subsequent evidence selection and synthesis.

E.2 Sensitivity Analysis

Table 13 reports a diagnostic sensitivity grid over the initial retrieval depth K, the falsification-retrieval depth K', and the number of generated falsification queries r. The analysis is conducted on a fixed $N = 200$ subset. Target-evidence hit rate generally increases with larger K' and r, but TRR remains within a narrower range, indicating that retrieval recall alone does not eliminate poisoned-claim adoption.

The default setting ($K = 5, K' = 20, r = 3$) is therefore not claimed as the best-performing point on this subset. It is a conservative configuration that captures most retrieval gains while avoiding a more aggressive evidence window. Increasing K' to 50 improves target hit rate in several configurations, but the corresponding TRR gains are modest and PAR is not monotonic, reinforcing that conflict adjudication remains the main bottleneck.

Table 14 complements the fair-setting comparison in Table 10 by exposing which source is removed by leave-one-out isolation. This diagnostic separates poisoned-document deletion from answer-level truth recovery.

F Ablation and Detectability Diagnostics

The ablation study is conducted on a fixed $N = 500$ subset drawn from the same evaluation pool

K	K'	r	PAR	TRR	Target hit
3	10	1	57.0%	62.0%	92.0%
3	10	3	56.5%	62.5%	94.5%
3	10	5	55.0%	61.5%	96.0%
3	20	1	56.0%	58.0%	92.0%
3	20	3	54.5%	62.5%	97.0%
3	20	5	53.5%	63.5%	96.5%
3	50	1	57.0%	59.0%	95.0%
3	50	3	53.5%	64.5%	99.0%
3	50	5	59.0%	62.0%	98.5%
5	10	1	56.0%	60.5%	92.0%
5	10	3	55.5%	63.0%	95.5%
5	10	5	56.0%	59.5%	97.0%
5	20	1	56.5%	59.5%	94.5%
5	20	3	57.5%	62.0%	97.0%
5	20	5	53.0%	62.5%	97.5%
5	50	1	59.5%	61.5%	94.0%
5	50	3	53.5%	64.5%	99.5%
5	50	5	59.0%	63.0%	98.0%
8	10	1	56.0%	58.5%	92.0%
8	10	3	55.0%	59.5%	96.0%
8	10	5	59.5%	59.5%	95.5%
8	20	1	57.0%	61.0%	95.0%
8	20	3	52.0%	63.5%	96.5%
8	20	5	59.5%	64.0%	97.5%
8	50	1	56.0%	59.0%	96.5%
8	50	3	54.5%	63.0%	99.5%
8	50	5	51.0%	63.5%	98.5%

Table 13: Sensitivity analysis over initial retrieval depth K, falsification-retrieval depth K', and number of falsification queries r on a fixed $N = 200$ subset.

RAGuard diagnostic	Value
Authentic source present	1200/1200 (100.0%)
RAGuard active	1200/1200 (100.0%)
Dropped poison doc index 0	889/1200 (74.1%)
Dropped authentic index 1	162/1200 (13.5%)

Table 14: Isolation diagnostics under the RAGuard fair setting.

Variant	PAR ↓	TRR ↑
full_afrag	57.2%	61.2%
w/o claim auditor	61.4%	55.2%
w/o deep retriever	94.0%	10.0%
w/o synthesizer	84.4%	22.2%

Table 15: AF-RAG component ablation results (N=500).

as the main only-poison cocoon setting. The main robustness and significance results retain the full $N = 1200$ setting; the smaller ablation subset is used only to isolate component contributions under repeated API-bound evaluation runs.

Table 15 reports the contribution of each AF-RAG component. The full variant reaches 57.2% PAR and 61.2% TRR on the ablation subset. Removing the claim auditor produces a moderate degradation, removing the deep retriever collapses performance close to Naive RAG, and removing the evidence-prioritized synthesizer sharply reduces truth recovery despite retrieval being available.

The noisy-versus-stealth diagnostic evaluates how attack detectability changes robustness. On the noisy subset ($N = 50$), AF-RAG reduces PAR to 38.0% and reaches 72.0% TRR. On the stealth subset ($N = 50$), Naive RAG becomes substantially more vulnerable, reaching 94.0% PAR and only 8.0% TRR, while AF-RAG preserves 62.0% TRR. This contrast supports the claim that local self-consistency and the absence of visible conflict increase attack strength.

G Statistical Testing

Paired significance tests are computed from per-sample outputs. PAR is converted into PAR-avoidance, defined as the proportion of samples in which the poisoned claim is not adopted. TRR is kept as the truth-recovery success rate. Each method is compared against Naive RAG with bootstrap confidence intervals and McNemar tests.

Tables 16 and 17 report the full paired-test results for AF-RAG and the diagnostic retrieval baselines. The split presentation avoids a wide appendix float while preserving the distinction between attack avoidance and truth recovery.

Table 16 reports paired significance tests after converting PAR into PAR-avoidance. AF-RAG has the largest absolute gain over Naive RAG, while the diagnostic retrieval baselines show smaller but statistically significant improvements.

Table 17 reports the corresponding paired tests

Method	Rate	Δ	95% CI	McNemar p
AF-RAG	40.58%	+35.08pp	[+31.92,+37.83]	3.883×10^{-84}
Claim-Only	15.25%	+9.75pp	[+8.00,+11.83]	1.958×10^{-26}
Query-Rewrite	12.50%	+7.00pp	[+5.42,+8.92]	2.356×10^{-16}
Retrieve-More	11.92%	+6.42pp	[+4.83,+8.17]	2.264×10^{-17}

Table 16: Paired PAR-avoidance significance tests relative to Naive RAG (N=1200).

Method	Rate	Δ	95% CI	McNemar p
AF-RAG	58.00%	+46.75pp	[+43.92,+49.67]	4.094×10^{-115}
Claim-Only	22.58%	+11.33pp	[+9.25,+13.33]	1.892×10^{-29}
Query-Rewrite	20.08%	+8.83pp	[+6.83,+10.75]	2.555×10^{-20}
Retrieve-More	19.67%	+8.42pp	[+6.67,+10.17]	1.044×10^{-22}

Table 17: Paired TRR significance tests relative to Naive RAG (N=1200).

for truth recovery. The larger AF-RAG effect size confirms that active falsification improves answer-level recovery beyond the gains achieved by broader retrieval, generic query rewriting, or direct claim search.

H Latency and Trigger-Rate Overhead

The overhead study measures average end-to-end wall-clock latency and the frequency with which zero-trust auditing is triggered. It is conducted on $N = 20$ factual poisoned queries and $N = 20$ benign queries, because the measurement requires repeated full pipeline execution with claim auditing and deep retrieval enabled.

Table 18 shows that AF-RAG increases latency on both factual poisoned and benign queries. The increase is larger for factual poisoned queries because auditing is triggered for all samples, whereas benign queries trigger zero-trust auditing less frequently. These measurements quantify the computational cost associated with the truth-recovery gains reported in the main text.

A false-conflict breakdown on benign queries shows that most AF-RAG false alerts are associated with numerical or metric-bearing claims (134/193, 69.4%), followed by broad factual claims (36/193, 18.7%) and comparative claims (15/193, 7.8%). Intent-guard misses account for only 8/193 cases (4.1%), suggesting that the main calibration issue is over-sensitive verification of substantive factual claims rather than systematic triggering on summary or bibliographic intents.

I Failure Taxonomy and Examples

Failure types are assigned from AF-RAG traces and per-sample judgments. A sample is classified

Metric	Factual poisoned queries	Benign queries
Naive RAG latency	1.02s	1.07s
AF-RAG latency	5.69±1.18s	3.36±1.79s
AF-RAG latency p95	7.62s	6.75s
Zero-trust trigger rate	100.0%	50.0%

Table 18: Average end-to-end latency and trigger-rate overhead of AF-RAG relative to Naive RAG, measured on $N = 20$ factual poisoned queries and $N = 20$ benign queries.

Truth recovered	Poisoned adopted	Poisoned not adopted	Row total
Yes	250 (20.83%)	446 (37.17%)	696 (58.00%)
No	463 (38.58%)	41 (3.42%)	504 (42.00%)
Column total	713 (59.42%)	487 (40.58%)	1200 (100.00%)

Table 19: Contingency between AF-RAG truth recovery and poisoned-claim adoption. PAR and TRR are independently judged answer-level metrics and are therefore not mutually exclusive.

as *truth not retrieved* when deep search runs but does not retrieve the authentic target. It is classified as *truth retrieved but synthesis chose poison* when the authentic target is reachable but the final answer still adopts the poisoned claim. It is classified as *truth retrieved but answer not recovered* when the authentic target is reachable and poisoned adoption is not the dominant failure, but the final answer still does not recover the truth.

Because poisoned-claim adoption and truth recovery are judged independently, the two metrics need not sum to one. Table 19 reports their overlap. The 250 samples in the upper-left cell contain sufficient authentic information to satisfy the truth-recovery judge but still retain the poisoned claim, so they count toward both TRR and PAR. This is why the aggregate counts for AF-RAG truth recovered (696/1200) and poisoned adoption (713/1200) are not contradictory.

Table 20 gives the complete taxonomy and counts used to produce the compact failure summary in the main text. The distinction between target-evidence reachability and final answer recovery is preserved because most residual errors occur after the authentic target has become retrievable.

Two representative synthesis failures illustrate the dominant bottleneck. In one sample, the authentic claim states that supervised post-retrieval methods substantially outperform their unsupervised counterparts, whereas the poisoned claim states that they show only marginal improvement

Failure type / metric	Count	All	Failures
Total samples	1200	100.00%	–
AF-RAG truth recovered	696	58.00%	–
AF-RAG poisoned adoption	713	59.42%	–
Audit triggered	1200	100.00%	–
Deep search triggered	1200	100.00%	–
Target evidence retrieved	1146	95.50%	–
Audit not triggered	0	0.00%	0.00%
Query generation failed	0	0.00%	0.00%
Deep search not triggered	0	0.00%	0.00%
Truth not retrieved	37	3.08%	7.34%
Truth retrieved but synthesis chose poison	435	36.25%	86.31%
Truth retrieved but answer not recovered	32	2.67%	6.35%
Judge or trace error	0	0.00%	0.00%

Table 20: Failure taxonomy used for the end-to-end error analysis.

with a 2.3% average gain. The authentic target is retrieved, but the final output still adopts the poisoned claim. In another sample, the authentic UCPOF claim reports improvements of 6.03%, 5.75%, and a 50.66% retrieval-trigger reduction, while the poisoned claim changes these values to 8.15%, 7.42%, and 38.24%. The target is again reachable, but synthesis fails to recover the authentic values. These cases indicate that future improvements should prioritize numerical evidence alignment, candidate-snippet selection, and conflict adjudication rather than retrieval expansion alone.