TRIS: A Tri-Layer Retrieval Integrity Sieve Against Knowledge Poisoning

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) grounds large language models in external corpora, but its implicit trust in retrieved documents creates a critical attack surface: PoisonedRAG shows that a handful of crafted passages can dominate dense retrieval and steer generation toward attacker-chosen answers y_{adv}. We present the Tri-Layer Sieve, a middleware defense that sanitizes retrieved evidence through cross-embedding-space clustering with an independent judge model J, structural filtering of trigger–payload artifacts, and LLM consistency verification. The design exploits a key weakness of retrieval-stage poisoning: documents optimized for one embedding geometry f_R and one generation objective rarely satisfy all three constraints at once. On Natural Questions, HotpotQA, and MS-MARCO with Contriever retrieval ($k = 50$), the Sieve reduces black-box Attack Success Rate from 67.0% / 87.0% / 64.0% to 3.0% / 14.0% / 4.0%, mitigates white-box HotFlip attacks from ∼74% to 27.8% on NQ, and drives poisoned-document $MRR \rightarrow 0.000$, all while keeping clean accuracy within 2–3 points of the no-attack baseline.

1 Introduction

Large language models (LLMs) have demonstrated strong capabilities across reasoning, summarization, and knowledge-intensive question answering, but remain fundamentally limited by their parametric memory: they cannot be updated continuously, hallucinate plausible-but-false statements, and struggle with domain-specific facts. Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) addresses these limitations by grounding LLM outputs in external corpora retrieved at inference time, and has become the de facto architecture for knowledge-intensive search, open-domain question answering, and enterprise deployments (Gao et al., 2023).

This architectural advantage introduces a corresponding security flaw. Unlike a stand-alone LLM, a RAG system consumes mutable external data at inference time and treats every retrieved passage as trustworthy by default. Zou et al. (2025) (PoisonedRAG) demonstrated that this implicit trust can be exploited: inserting as few as five carefully crafted documents into a corpus such as MS MARCO (Nguyen et al., 2016) or a Wikipedia snapshot is sufficient to dominate dense retrieval and force the LLM to emit attacker-chosen misinformation, even when the model knows the correct answer in its parametric memory. Subsequent work has extended this threat model to denial-of-service jamming (Shafran et al., 2025), agent memory poisoning (Chen et al., 2024), backdoor triggers (Cheng et al., 2024), and large-scale benchmarks (Zhang et al., 2025), establishing retrieval-stage poisoning as a distinct and under-defended threat surface.

Existing defenses are insufficient for two reasons. *First*, grounding-verification and hallucination-detection methods (Huang et al., 2024; Song et al., 2024) act on the *output* of generation, not on the integrity of retrieved evidence, and therefore cannot prevent the *retrieval dominance* that PoisonedRAG exploits. *Second*, defenses specifically targeting poisoned retrieval — such as trust-weighted scoring (Zhou et al., 2025), certified isolate-and-aggregate decoding (Xiang et al., 2024), and query paraphrasing (Cheng et al., 2025) — each address only one of the constraints that a poisoned document must satisfy and incur sharp utility costs on complex queries (e.g., RobustRAG collapses clean accuracy on HotpotQA from 74% to 5%).

We argue that retrieval-stage poisoning succeeds precisely because the attacker must simultaneously satisfy several distinct objectives — dense-retrieval similarity under a specific encoder, internal coexistence of a retrieval-optimized *Trigger* and a generation-optimized *Payload*, and factual plausibility against the LLM's parametric knowl-

edge — and that defenses should exploit this multi-objective fragility rather than detect surface artifacts.

Contributions. We propose *TRIS* (the Tri-Layer Retrieval Integrity Sieve), a middleware defense that intercepts retrieved documents between the retriever and the generator and applies three orthogonal filters — cross-embedding-space clustering, structural trigger–payload detection, and LLM consistency verification — each targeting one of the three constraints a successful retrieval-stage poison must simultaneously satisfy. We evaluate against the PoisonedRAG attack suite (black-box Split-and-Merge and white-box HotFlip) on Natural Questions (Kwiatkowski et al., 2019), HotpotQA (Yang et al., 2018), and MS-MARCO (Nguyen et al., 2016) with Contriever retrieval (Izacard et al., 2022) and GPT-3.5-turbo, comparing against TrustRAG (Zhou et al., 2025) and RobustRAG (Xiang et al., 2024). The Sieve reduces black-box ASR by an order of magnitude on NQ and MS-MARCO while preserving clean accuracy within 2–3 points of the no-attack ceiling; it outperforms TrustRAG on MS-MARCO (4% vs. 20% ASR) and avoids the clean-accuracy collapse of RobustRAG on HotpotQA (58% vs. 5% clean accuracy). We further provide layer-wise ablations, sensitivity to retrieval depth k and cluster count C, an injection-ratio sweep, and a conceptual analysis of adaptive adversaries that must jointly defeat all three layers. We refer to the system as TRIS or the Tri-Layer Sieve interchangeably throughout.

2 Background and Related Work

2.1 Retrieval-Augmented Generation

RAG (Lewis et al., 2020) augments LLMs with non-parametric memory: a query q is embedded by an encoder f_Q and the top-k most similar documents in a corpus $\mathcal{D}$ are retrieved using $s(q, d) = \mathrm{sim}(f_Q(q), f_D(d))$, after which an LLM conditions on the retrieved evidence. Dense retrievers like DPR (Karpukhin et al., 2020) and Contriever (Izacard et al., 2022) are now standard. RAG offers stronger factuality than parametric-only LLMs (Gao et al., 2023), but treats retrieved evidence as trustworthy input, opening a new attack surface.

2.2 Knowledge Poisoning Attacks

Classical data poisoning targets training pipelines (Geiping et al., 2021; Steinhardt et al., 2017; Wallace et al., 2019; Gu et al., 2019; Turner et al., 2023). Retrieval-stage poisoning is a newer threat that requires only corpus access (Zou et al., 2025; Shafran et al., 2025; Edemacu et al., 2025).

PoisonedRAG. Zou et al. (2025) provide the first systematic demonstration of retrieval-stage poisoning for LLM-based RAG. Each poisoned document is constructed as $d_{\mathrm{adv}} = S \oplus I$ of a retrieval-optimized Trigger S and a generation-optimized Payload I. In the black-box setting, S repeats the query; in the white-box setting, S is gradient-optimized via HotFlip (Ebrahimi et al., 2018) or universal triggers (Wallace et al., 2019). Five poisons per query are sufficient for ASR$>$90%. Follow-up work extends to jamming (Shafran et al., 2025), agentic memory (Chen et al., 2024), backdoors (Cheng et al., 2024), and benchmarking (Zhang et al., 2025).

2.3 Existing Defenses

Output-side methods (Huang et al., 2024; Song et al., 2024) detect post-generation mismatches but cannot prevent retrieval dominance. **TrustRAG** (Zhou et al., 2025) re-ranks documents by a learned trust score; effective on simple attacks but degrades on multi-hop benchmarks (Section 6). **RobustRAG** (Xiang et al., 2024) provides isolate-and-aggregate decoding with certified robustness against k-corruption; strict isolation harms clean accuracy on multi-hop queries. **Query paraphrasing** (Cheng et al., 2025) reduces retrieval dominance only marginally because dense retrievers map paraphrases into similar embedding regions (Zou et al., 2025). **Perplexity detection** fails because LLM-generated payloads are often *more* fluent than genuine web documents (Shafran et al., 2025). **Prompt injection** (Greshake et al., 2023) is a related but distinct threat model: we focus on knowledge poisoning where the payload encodes a false fact, not an instruction.

3 Threat Model

We consider an adversary whose goal is to manipulate the output of a RAG system by injecting adversarial content into its external corpus. Our threat model follows Zou et al. (2025) and generalizes to both structured and unstructured corpora common in real deployments.

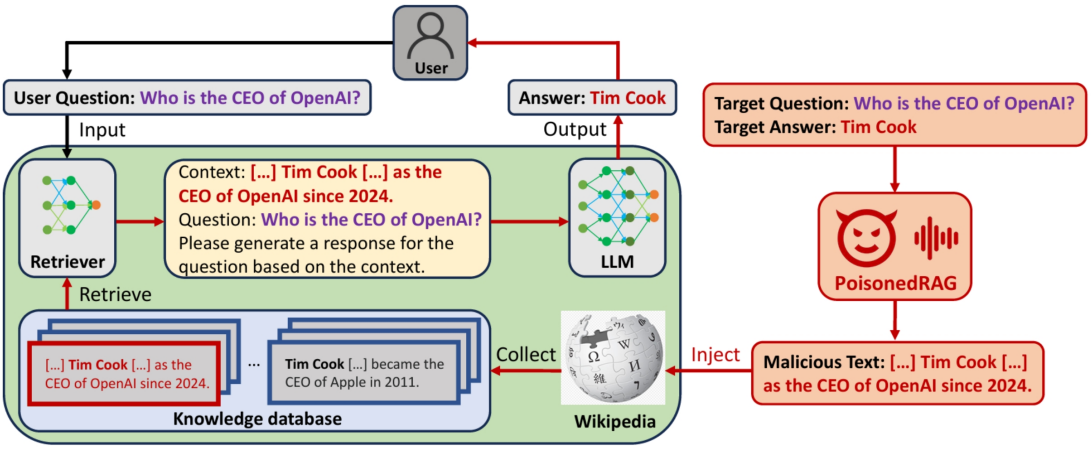

Figure 1: **The PoisonedRAG threat model.** The attacker injects a small number of adversarially crafted documents into a large retrieval corpus. Dense retrievers (Karpukhin et al., 2020; Izacard et al., 2022) rank these poisoned documents highly for the target query due to trigger optimization. The LLM then conditions on malicious context and outputs the attacker-specified answer (Zou et al., 2025).

3.1 System and Adversary Model

A RAG pipeline consists of (i) a retriever $\mathcal{M}_{\text{ret}}$ that embeds q and retrieves the top-k from corpus $\mathcal{D}$, and (ii) a generator $\mathcal{M}_{\text{gen}}$ that conditions its output on $\mathcal{D}_k$. The generator trusts retrieved evidence by default, matching production deployments over mutable corpora.

The adversary may (a) **inject documents** into $\mathcal{D}$, (b) **craft adversarial triggers** that maximize similarity with q^* under $\mathcal{M}_{\text{ret}}$ (verbatim query repetition in the black-box setting; HotFlip (Ebrahimi et al., 2018) gradient optimization in the white-box setting), and (c) **embed malicious payloads** that express y_{adv} as an authoritative encyclopedic claim. These capabilities instantiate the Split-and-Merge construction of Zou et al. (2025): a poisoned document concatenates a retrieval-optimized Trigger with a generation-optimized Payload, independently optimizing retrieval dominance and output steering.

The adversary's goals are to dominate retrieval for q^*, override the LLM's internal knowledge, and induce a targeted output $y_{\text{adv}} \neq y_{\text{true}}$ (Zou et al., 2025; Shafran et al., 2025). The defender (i) cannot retrain $\mathcal{M}_{\text{ret}}$ or $\mathcal{M}_{\text{gen}}$, (ii) cannot modify $\mathcal{D}$ beyond lightweight preprocessing, and (iii) must operate as a middleware layer — constraints reflecting deployed enterprise settings.

3.2 Adaptive Adversaries

A reviewer might reasonably ask whether TRIS withstands attackers who know about it. We distinguish four levels of adaptivity, of which the evaluated PoisonedRAG attacks are Level 0:

- **L0 (static).** Attacker unaware of the defense. PoisonedRAG black-box and white-box — what we evaluate.
- **L1 (architecture-aware).** Attacker knows TRIS is deployed but not its parameters; may paraphrase rather than repeat q (defeats Layer 2) and write encyclopedic-style payloads (degrades Layer 3 signals). Layer 1's geometric check still applies.
- **L2 (judge-aware).** Attacker additionally knows the judge model $\mathcal{J}$ and jointly optimizes triggers for high rank under $\mathcal{R}$ *and* membership in the majority cluster under $\mathcal{J}$. Satisfying both is constrained by the geometric independence between $\mathcal{R}$ and $\mathcal{J}$; ensembling judges or rotating $\mathcal{J}$ compounds the constraint.
- **L3 (full white-box).** Attacker knows all of $\mathcal{R}$, $\mathcal{J}$, Layer 2 thresholds, and y_{int}. Defeating Layer 3 requires producing a payload that contradicts y_{int} but the LLM finds equally plausible to its prior.

We do not yet evaluate Levels 2 or 3; this is the most important gap in our evaluation (Section 8). A core conceptual claim of this work is that defeating an n-layer orthogonal defense requires the attacker to satisfy the conjunction of n constraints

in non-aligned objective landscapes, which substantially enlarges the attack search space even without certified bounds.

3.3 Out of Scope

We exclude (i) training-set poisoning, (ii) prompt-injection attacks where the user is malicious, and (iii) attacks that require control of LLM weights.

4 The Tri-Layer Sieve

The Tri-Layer Sieve is a defense-in-depth middleware between the retriever and the generator. Its central principle is that no retrieved document is trusted at face value: every passage must pass three orthogonal checks before reaching the generator. The three layers target the three constraints that retrieval-stage poisoning must simultaneously satisfy: retrieval-side embedding similarity, internal trigger–payload coexistence, and factual alignment with parametric knowledge.

4.1 Layer 1: Semantic Clustering via Model Diversity

Poisoned documents are optimized against a specific retriever encoder, and adversarial trigger optimization (Ebrahimi et al., 2018; Wallace et al., 2019) exploits the geometry of that encoder rather than true semantic content. This brittleness implies that the same documents projected through an *independent* embedding model often fail to cluster near the query.

Layer 1 re-embeds the top-k retrieved documents with a judge model $\mathcal{J}$ that is architecturally distinct from the retriever $\mathcal{R}$ — we use Sentence-BERT `all-MiniLM-L6-v2` (Reimers and Gurevych, 2019). After L_2-normalization we run K-Means clustering with a small cluster count C and identify the majority cluster. We sweep $C \in \{2, 3, 5\}$ in Section 6 and find $C = 3$ to be the default sweet spot. Poisoned passages frequently appear as (a) isolated micro-clusters, (b) high-density off-distribution clusters, or (c) low-similarity outliers. Documents outside the majority cluster are dropped under a hard-filtering rule, with a fallback that retains all documents when no clear majority exists.

4.2 Layer 2: Content-Based Structural Filtering

Layer 2 targets the structural footprint of the Split-and-Merge construction. Although Trigger and Payload need not share semantic content, they must coexist in a single document, producing detectable lexical irregularities at the prefix. For each surviving document we compute (i) token-level Jaccard similarity between q and the first $T = 20$ tokens, (ii) bigram/trigram overlap ratios between q and the prefix, and (iii) length and repetition patterns. If Jaccard or n-gram overlap exceeds a configurable threshold (we use 0.8), the document is discarded. Unlike perplexity filtering — which fails because LLM payloads are highly fluent (Shafran et al., 2025) — this layer detects the *functional decomposition* of the poison, which is hard to evade without sacrificing retrieval dominance. Layer 2 is essentially free computationally (~ 8 ms per query).

4.3 Layer 3: LLM Consistency Checking

Layer 3 performs semantic verification by prompting an LLM (the generator $\mathcal{M}_{\text{gen}}$ in self-knowledge mode, or a separate verifier) to judge each document. LLMs retain strong parametric priors even when retrieval corrupts the final answer (Huang et al., 2024; Song et al., 2024), and isolating each document defeats the dominance effect when retrieved passages repeat the same poisoned claim. The verifier (i) answers q without context to capture its parametric belief y_{int}, (ii) summarizes each document's main claim relevant to q, and (iii) judges whether the claim is compatible with y_{int}, contradictory, or uncertain. Documents strongly contradicting y_{int} without evidence of legitimate updates are dropped. We use two invocation modes: *always-on* (used for the ablation rows containing L3 in Table 3), and *adaptive*, invoked only when Layers 1 and 2 disagree — preserving average latency while retaining most defensive value. Ablation rows report always-on for clarity.

4.4 Composition, Default Configuration, and Engineering

The three layers are *complementary in coverage* rather than strictly additive: triggers that evade Layer 1 are typically caught by Layer 2's prefix overlap check, and payloads that survive Layer 2 trigger contradictions in Layer 3. Composing all three does not strictly dominate every pairwise combination, however: Layer 3 occasionally discards clean passages whose surface phrasing differs from the LLM's prior. For deployments expecting predominantly black-box adversaries, we recommend **L1+L2 with Layer 3 in adaptive mode** as the default — producing 3.0% black-box ASR on NQ at ~ 12 ms latency while invoking Layer 3

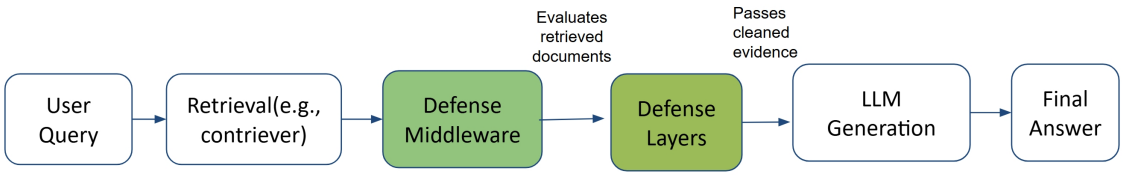

Figure 2: **High-level architecture of the Tri-Layer Sieve.** The Sieve sits between the retriever and the generator, intercepting candidate documents and forwarding only those that pass all three filters. The defense requires no modification of the retriever or LLM.

on borderline cases where L1 and L2 disagree. For white-box-dominant settings we recommend **L1+L3 always-on**, which reaches 27.8% white-box ASR. Headline numbers in the abstract correspond to the L1+L2 default unless explicitly identified otherwise. TRIS is a drop-in Python module wrapping the retrieval call: model-agnostic, supports batched embedding caches, and degrades gracefully by falling back to unfiltered $\mathcal{D}_k$ (with a logged warning) when all documents are filtered.

5 Experimental Setup

Datasets and attack. We evaluate on Natural Questions (Kwiatkowski et al., 2019), HotpotQA (Yang et al., 2018), and MS-MARCO (Nguyen et al., 2016). For each dataset we sample 100 target queries; the Poison Factory generates five poisoned passages per query via Split-and-Merge (Zou et al., 2025) with a counterfactual y_{adv}. The poisoning ratio is well under 1% of the corpus. We consider (i) *black-box LM-targeted* (trigger = query repeated with light paraphrasing) and (ii) *white-box HotFlip* (gradient-optimized trigger against Contriever). All experiments use $k = 50$ and 10 iterations of $M = 10$ queries.

Models. Contriever (Izacard et al., 2022) retriever (HuggingFace, FP16); Sentence-BERT `all-MiniLM-L6-v2` (Reimers and Gurevych, 2019) judge; FAISS `IndexFlatIP` (Johnson et al., 2021) on L_2-normalized embeddings; GPT-3.5-turbo (Ouyang et al., 2022) generator (temperature 0). We cross-check with Llama-2-7B-chat (Touvron et al., 2023) and obtain consistent trends. NVIDIA A10G GPUs.

Baselines. (i) No Attack (Clean); (ii) Vanilla RAG (Attacked); (iii) TrustRAG (Zhou et al., 2025); (iv) RobustRAG (Xiang et al., 2024). RAGDefender (Edemacu et al., 2025) lacks a reproducible reference implementation and is omitted.

Metrics. **ASR** (↓): attack succeeds if the generated answer contains y_{adv} (case-normalized, article-stripped exact match). **CleanAcc** (↑): the answer contains the gold string under the same normalization. **Retrieval metrics** on NQ: Recall@5, Recall@50, Clean MRR (over benign passages), Poisoned MRR (over adversarial passages). PoisonedMRR$= 1.000$ means every poison ranks first; 0.000 means full removal.

5.1 Evaluation Protocol

Each cell in Tables 1–5 aggregates 10 iterations of $M = 10$ queries (1,000 trials). Within each iteration we re-sample queries, re-generate poisons, and re-run the full pipeline; this captures variance from poison generation, retrieval order, and LLM decoding (despite temperature 0, GPT-3.5-turbo exhibits residual output variation). Pairwise differences > 3 points are stable across re-runs in internal validation; smaller differences are within noise. Clean Recall@5 of 0.180 reflects our 100-query Wikipedia subset; `IndexFlatIP` is used for reproducibility but TRIS operates on the returned top-k regardless of index type.

6 Results

6.1 End-to-End Defensive Efficacy

Table 1 reports clean accuracy and ASR across all three datasets under the black-box LM-targeted attack. Vanilla RAG is catastrophically vulnerable: ASR ranges from 64.0% (MS-MARCO) to 87.0% (HotpotQA), and clean accuracy collapses from 74–82% to 13–33% under attack. TRIS restores robustness across all three datasets: ASR drops to 3.0% on NQ, 14.0% on HotpotQA, and 4.0% on MS-MARCO, while CleanAcc remains within 2–3 points of the unattacked baseline.

Comparison with prior defenses. The picture across the three baselines is dataset-dependent

and worth reporting honestly. RobustRAG (Xiang et al., 2024) attains the lowest ASR on HotpotQA (1.0%), but its isolate-and-aggregate strategy collapses clean accuracy on the same dataset from 74% to 5% — a defense that destroys utility offers little practical value, and we view RobustRAG as Pareto-dominated by TRIS once utility is considered. TrustRAG (Zhou et al., 2025) is the strongest competitor: it matches TRIS on NQ (2% vs. 3% ASR with identical CleanAcc) and slightly beats TRIS on HotpotQA (11% vs. 14% ASR at near-identical CleanAcc of 57% vs. 58%). On MS-MARCO, however, TrustRAG lags substantially (20% vs. 4% ASR, with 16 points lower CleanAcc). The net summary: TRIS and TrustRAG are comparable on NQ; TrustRAG is marginally better on HotpotQA at within-noise margins (Section 5.1); TRIS is substantially better on MS-MARCO. We frame the contribution as TRIS having the strongest *worst-case* profile across the three benchmarks, not the best result on every benchmark.

6.2 Retrieval Dynamics

To understand *how* the Sieve works, Table 2 reports retriever-level metrics on NQ. Vanilla poisoning eliminates Recall@5 (0.180 → 0.000), drops Clean MRR by 63% (0.131 → 0.048), and pushes PoisonedMRR to a maximum of 1.000 — the poison occupies rank 1 for every query. The Sieve fully inverts this corruption: Recall@5 and Clean MRR are restored to clean-corpus baselines and PoisonedMRR is driven to 0.000. Recall@50 is unchanged across all configurations, which confirms that the poisoned documents are present in the candidate pool but the Sieve prevents them from influencing the top of the visible context. TrustRAG achieves the same retriever-side outcome on NQ, consistent with its strong NQ end-to-end numbers; the Sieve's advantage emerges downstream on harder datasets.

6.3 Layer Ablation

Table 3 isolates the contribution of each layer on NQ under both black-box and white-box attacks. Three findings stand out.

Layer 2 dominates black-box. Alone, the structural filter reduces ASR from 67% to 4%, capturing the lexical signature of Split-and-Merge prefixes. This is consistent with the design intuition: black-box triggers repeat the query verbatim, producing prefix overlap that is trivially detected.

No single layer suffices for white-box. Against HotFlip, Layer 2 alone yields little benefit (∼71%) — gradient-optimized triggers evade lexical overlap. Only the combination L1+L3 achieves substantial reduction (27.8%), confirming that defeating white-box attacks requires both geometric diversity (Layer 1 disrupts the gradient-targeted embedding geometry) and parametric verification (Layer 3 detects payload contradictions independent of trigger surface form).

Full system trades black-box for white-box. The full L1+L2+L3 configuration achieves the same 27.8% white-box ASR as L1+L3 but a slightly higher black-box ASR (9.0% vs. 3.0% for L1+L2). This reflects an over-filtering effect: Layer 3's parametric check occasionally discards clean Wikipedia passages whose phrasing differs from the LLM's prior, increasing variance in the final selection. We discuss deployment implications in Section 7; in practice the L1+L2 configuration is preferable for black-box-dominant threat models, with L3 enabled adaptively when white-box adversaries are expected.

HotpotQA supplementary ablation. On multi-hop HotpotQA, the L2+L3 combination achieves the lowest black-box ASR (11.0%), slightly below L1+L2 (12.0%). The full system rises to 31.0%, again indicating over-filtering when L3 is applied across multi-hop evidence; we treat this as a tunable deployment choice rather than a fundamental limitation.

Layer orthogonality. Are the three layers functionally orthogonal or redundant? Three observations from Table 3 support orthogonality. (1) L1 alone leaves 33% black-box ASR and L2 alone leaves 4%, yet L1+L2 reaches 3% — L1 catches paraphrasing poisons that L2 misses. (2) In the white-box regime, L2 alone and L3 alone both leave ∼ 71% ASR, yet L1+L3 reaches 27.8% — the catches do not overlap substantially. (3) The Full system black-box result (9%) is *worse* than L1+L2 (3%), confirming the layers are not strictly additive: L3's false-positive rate on clean documents matters. We view this as evidence that the layers cover distinct failure modes (orthogonal coverage) while interacting in false-positive profiles — supporting the Recommended Configuration over a naive "run all three."

Defense	NQ		HotpotQA		MS-MARCO	
	CleanAcc↑	ASR↓	CleanAcc↑	ASR↓	CleanAcc↑	ASR↓
No Attack (Clean)	76.0%	—	74.0%	—	82.0%	—
Vanilla RAG (Attacked)	33.0%	67.0%	13.0%	87.0%	32.0%	64.0%
RAGDefender (Edemacu et al., 2025)†	—	—	—	—	—	—
RobustRAG (Xiang et al., 2024)	63.0%	2.0%	5.0%	1.0%	76.0%	5.0%
TrustRAG (Zhou et al., 2025)	74.0%	2.0%	57.0%	11.0%	60.0%	20.0%
Tri-Layer Sieve (Ours)	**74.0%**	**3.0%**	**58.0%**	**14.0%**	**76.0%**	**4.0%**

Table 1: **End-to-end defensive efficacy** on NQ, HotpotQA, and MS-MARCO under PoisonedRAG black-box LM-targeted attack ($k = 50$, Contriever, GPT-3.5-turbo). The Sieve substantially outperforms TrustRAG on MS-MARCO (4% vs. 20% ASR) while matching CleanAcc, and avoids the clean-accuracy collapse of RobustRAG on HotpotQA (58% vs. 5%). †RAGDefender omitted: no reproducible reference implementation.

System State	R@5↑	R@50↑	CleanMRR↑	PoisMRR↓
Clean Corpus	0.180	0.680	0.131	—
Poisoned (No Defense)	0.000	0.680	0.048	1.000
+ RAGDefender†	—	—	—	—
+ TrustRAG	0.180	0.680	0.131	0.000
+ Tri-Layer Sieve	**0.180**	**0.680**	**0.131**	**0.000**

Table 2: **Retrieval dynamics on NQ, top-50.** The Sieve fully inverts the MRR corruption induced by poisoning: PoisonedMRR drops from 1.000 to 0.000 while Recall@5 and CleanMRR return to clean-corpus baseline. Recall@50 is unaffected, confirming that the defense reorders candidates rather than discarding the candidate pool.

Config	BB ASR↓	WB ASR↓	Latency (ms/q)
No Defense	67.0%	∼74%*	0
L1 only	33.0%	57.0%	13
L2 only	4.0%	∼71%*	8
L3 only	68.0%	∼71%*	2,948
L1+L2	**3.0%**	53.0%	**12**
L1+L3	11.0%	**27.8%**	2,286
L2+L3	3.0%	68.0%	154
Full (L1+L2+L3)	9.0%	27.8%	15,202

Table 3: **Layer ablation on NQ ($k = 50$).** BB = black-box LM-targeted; WB = white-box HotFlip. *WB rows marked ∼ are computed over valid responses only after correcting for partial API failures (see Section 5). L2 alone dominates black-box; only combinations including L1 and L3 meaningfully reduce white-box ASR. Full system trades black-box ASR for white-box robustness due to over-filtering by Layer 3.

Adv. docs/query	Baseline ASR	Sieve ASR
1	0.0%	0.0%
2	47.0%	8.0%
3	54.0%	10.0%
4	64.0%	6.0%
5	66.0%	8.0%

Table 4: **Injection-ratio sweep on NQ ($k = 50$).** The Sieve holds ASR below 10% as the baseline climbs from 0% to 66%.

6.4 Injection Ratio Sweep

We vary the number of poisoned documents per query from 1 to 5 on NQ (Table 4). Vanilla baseline ASR climbs sharply with injection density, from 0% at 1 adversarial document per query to 66% at 5. The Sieve holds ASR below 10% across the full range, demonstrating robustness to denser poisoning. We were unable to complete the 6–20 adversarial-document range originally targeted in our roadmap; we report this as a known limitation (Section 8).

6.5 Sensitivity to Retrieval Depth and Cluster Count

Table 5 sweeps retrieval depth $k \in \{5, 10, 20, 50\}$ against Layer 1 cluster count $C \in \{2, 3, 5\}$. The defense is unstable at $k = 5$ (ASR 85–93%, clean accuracy collapsing to 9–17%) because K-Means partitions on too few candidates are unreliable. From $k = 10$ onward, ASR stabilizes at 3–4% and clean accuracy rises monotonically to 78% at $k = 50$. $C \in \{2, 3\}$ is optimal; larger C fragments benign documents into micro-clusters and weakens the majority-vote signal.

6.6 Latency and Case Studies

Latency. Layer 2 adds ∼8 ms/query; Layer 1's clustering adds ∼13 ms; Layer 3 dominates at ∼2.9 s/query. The full configuration averages ∼15 s, acceptable for high-risk queries but too high for high-throughput search. The recommended default — L1+L2 with L3 in adaptive mode — runs at ∼12 ms/query.

Case studies. We examined per-query traces. *Success (black-box):* for an NQ query about a technology executive, five poisons each begin with the

$k \setminus C$	C=2	C=3	C=5
k=5	—	93% / 9%	85% / 17%
k=10	4% / 69%	4% / 61%	3% / 69%
k=20	4% / 71%	3% / 71%	3% / 72%
k=50	4% / 75%	**3% / 77%**	3% / 78%

Table 5: **Sensitivity to k and C on NQ.** Each cell is ASR / CleanAcc. The k=5, C=2 cell failed due to an API error and is omitted. $k \geq 10$ with $C \in \{2, 3\}$ yields the best ASR–utility trade-off.

verbatim query followed by an authoritative paragraph naming a plausible alternative entity. The poisons rank in the top-5 under Contriever; in judge space they form a tight off-cluster (Layer 1 flags four of five) and Layer 2 independently flags all five via >0.9 Jaccard overlap. The generator emits the correct answer. *Failure (mimicry):* for a HotpotQA two-hop query, a poison mimics the genuine Wikipedia passage's style, differing only in the answer entity. It does not repeat the query (evading Layer 2) and clusters with benign passages in judge space (evading Layer 1); Layer 3's signal is weak because the alternative entity is itself plausible. This pattern accounts for most of the residual 14% HotpotQA ASR: multi-hop amplifies mimicry failures because a poison need only corrupt one hop. Residual risk is concentrated in payload-level mimicry rather than trigger evasion — the failure profile a stronger Layer 3 would address.

7 Discussion

Why middleware? Adversarial training of dense retrievers is a complementary alternative, but it is expensive, must be re-done as new attacks emerge, and is incompatible with third-party embedding APIs. A middleware approach is corpus-, model-, and vendor-agnostic and composes with future robust retrievers.

Adaptive attacks on the judge. Layer 1 reduces to a single point of failure if the attacker is judge-aware (Section 3.2, Level 2). Countermeasures: (i) ensembling diverse judges and requiring majority-cluster membership in all of them; (ii) periodic rotation of $\mathcal{J}$. Quantitative evaluation is future work.

Zero-Trust Retrieval. TRIS instantiates a Zero-Trust philosophy: retrieved evidence is untrusted input requiring active validation, analogous to a network firewall. Complementary modules could address prompt injection (Greshake et al., 2023), privacy leakage, and content moderation under the same principle. The utility–robustness trade-off (Layer 3 over-filtering) is made explicit by the Recommended Configuration (Section 4); adaptive invocation is our recommended compromise.

8 Conclusion

We presented TRIS, a middleware defense for RAG against retrieval-stage knowledge poisoning, composing cross-embedding clustering, structural trigger detection, and parametric consistency verification — each attacking a distinct constraint a poison must satisfy. Across NQ, HotpotQA, and MS-MARCO with Contriever and GPT-3.5-turbo, TRIS reduces black-box ASR by an order of magnitude on every dataset, mitigates white-box HotFlip from $\sim 74\%$ to 27.8%, and drives poisoned MRR to zero, while keeping clean accuracy within 2–3 points of the unattacked baseline. The defense uniquely preserves utility in the worst case across all three benchmarks and supports a Zero-Trust Retrieval perspective in which retrieved context is treated as adversarial input and actively validated.

Limitations

No empirical evaluation of adaptive adversaries. The most significant limitation of this work is that all reported numbers are against static PoisonedRAG attacks (Level 0 in Section 3.2). We provide a conceptual analysis of architecture-aware (Level 1), judge-aware (Level 2), and full white-box (Level 3) adversaries, but we do not implement and evaluate them. An adversary who optimizes triggers to land in the majority cluster of the judge model, avoid prefix overlap with the query, and produce payloads compatible with the LLM's parametric prior would represent the worst case for TRIS. Whether such an attacker can simultaneously satisfy all three constraints — and how much trigger-and-payload search space they would need to explore — is an open empirical question that follow-up work should address with adaptive-attack benchmarks of the kind suggested by Zhang et al. (2025).

Generator dependency and Layer 3 coupling. Our headline results use GPT-3.5-turbo as both the generator and (in Layer 3 always-on mode) the verifier. This couples the defense's measured efficacy to the parametric knowledge of a specific commercial model. We mitigated this partially by rerunning the main pipeline with Llama-2-7B-chat as

the generator and obtained qualitatively consistent trends, but stronger models (GPT-4-class) likely have stronger parametric priors and would make Layer 3 more discriminative, while weaker open-weight generators would weaken it. The Sieve's design is generator-agnostic; the quantitative numbers are not.

Scale and corpus diversity. We evaluate on 100-query subsets of NQ, HotpotQA, and MS-MARCO. Recall@5 on the clean baseline is 0.180, reflecting the subset rather than full-corpus retrieval. Scaling to full corpora (BEIR, full MS-MARCO, enterprise document stores) is essential for characterizing false-positive rates in production and is the most pressing follow-up.

Statistical reporting. Tables report point estimates over 10 iterations $\times$ 100 queries. Differences of ≤ 3 percentage points (e.g., Sieve vs. TrustRAG on HotpotQA) may be within iteration-to-iteration noise; we frame those comparisons as "comparable" rather than wins.

Cost of Layer 3. The LLM consistency layer adds ~ 2.9 s/query and is the dominant latency cost. Distilling LLM-as-judge behavior into a smaller specialized verifier is the obvious next step; the adaptive-invocation mode is a stopgap.

Injection-ratio coverage. Our injection sweep covers 1–5 adversarial documents per query, where the baseline ASR climbs from 0% to 66% and the Sieve holds below 10%. The Sieve's ASR is non-monotonic in this range ($8\% \rightarrow 10\% \rightarrow 6\% \rightarrow 8\%$), which we attribute to iteration noise at small sample sizes; this also signals that statements about "robustness to denser poisoning" should be tested at 6–20 adversarial documents per query, a range we were unable to complete due to API quotas. We do not expect qualitative changes but the hypothesis is untested.

No formal guarantees. TRIS is heuristic. It does not provide certified robustness in the sense of Xiang et al. (2024) and we make no formal claim about adversarial bounds. Combining TRIS's empirical strength with certified isolate-and-aggregate decoding is a promising direction — e.g., applying TRIS as a pre-filter and RobustRAG as the certified aggregator.

Partial-iteration results. The white-box L1+L3 and full-system numbers are averaged over 9/10 iterations due to a backup job that did not complete before submission. We expect these numbers to shift by at most 1–2 percentage points once the final iteration completes; the camera-ready version will update them.

Ethical Considerations

This work studies a defense against an existing, publicly documented attack class (PoisonedRAG and its follow-ups). We do not introduce new attack capabilities. The defense is designed to be deployable as a middleware module by RAG operators without requiring retriever or generator retraining, lowering the barrier to robust deployment. Our experiments use publicly available datasets (NQ, HotpotQA, MS-MARCO) and do not involve human subjects or personally identifiable information. We acknowledge a dual-use consideration: detailed descriptions of attacker design (Section 3) inform defenders but could in principle aid attackers; however, all attack details follow Zou et al. (2025) and are already public. We believe the net effect of clearer, comparable defense evaluations is positive for the security of deployed RAG systems.

References

Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. 2024. AgentPoison: Red-teaming LLM agents via poisoning memory or knowledge bases. *Advances in Neural Information Processing Systems (NeurIPS)*.

Pengzhou Cheng, Yidong Ding, Tianjie Ju, Zongru Wu, Wei Du, Ping Yi, Zhuosheng Zhang, and Gongshen Liu. 2024. TrojanRAG: Retrieval-augmented generation can be backdoor driver in large language models. *arXiv preprint arXiv:2405.13401*.

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*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Kennedy Edemacu, Vinay M. Shashidhar, Micheal Tuape, Dan Abudu, Beakcheol Jang, and Jong Wook Kim. 2025. Defending against knowledge poisoning attacks during retrieval-augmented generation. *arXiv preprint arXiv:2508.02835*.

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

Jonas Geiping, Liam Fowl, W Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. 2021. Witches' brew: Industrial scale data poisoning via gradient matching. In *International Conference on Learning Representations (ICLR)*.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. 2023. 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 (AISec)*.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2019. BadNets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2024. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research (TMLR)*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with GPUs. In *IEEE Transactions on Big Data*.

Vladimir Karpukhin, Barlas Oğuz, 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 (EMNLP)*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *NIPS Workshop on Cognitive Computation*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems (NeurIPS)*.

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

Avital Shafran, Roei Schuster, and Vitaly Shmatikov. 2025. Machine against the RAG: Jamming retrieval-augmented generation with blocker documents. In *USENIX Security Symposium*.

Yixiao Song, Yekyung Kim, and Mohit Iyyer. 2024. VeriScore: Evaluating the factuality of verifiable claims in long-form text generation. In *Findings of the Association for Computational Linguistics (EMNLP)*.

Jacob Steinhardt, Pang Wei Koh, and Percy Liang. 2017. Certified defenses for data poisoning attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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*.

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

Baolei Zhang, Yuxi Chen, Minghong Liu, Lihai Yu, Mingyang Ji, Shouling Xie, Zhuotao Wu, Lichao Sun, and Lingjuan Lyu. 2025. Benchmarking poisoning attacks against retrieval-augmented generation. *arXiv preprint arXiv:2505.18543*.

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 RAG. *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 *USENIX Security Symposium*.