RAGuard: A Layered Defense Framework for Retrieval-Augmented Generation Systems Against Data Poisoning

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) systems are becoming more common in augmenting large language models (LLMs) with factual knowledge, yet they remain highly vulnerable to data poisoning, i.e., maliciously injected passages that manipulate retrieved evidence. We introduce RAGuard, a layered two-step defense framework that combines retrieval-level adversarial training with a novel zero-knowledge inference patch. The first step fine-tunes dense retrievers (e.g., Contriever, compatible with BGE and others) using synthetic poisoned documents (composed of poisons such as fabricated facts, contradictions, and reasoning traps), training them to downrank malicious passages. The second step applies a black-box approach zero knowledge inference patch that identifies and filters suspicious documents based on their causal influence on QA correctness, without requiring poison labels. Experiments on Natural Questions (NQ) and Benchmarking-IR (BEIR) show that RAGuard improves robustness by reducing the Attack Success Rate (ASR) while maintaining retrieval quality (Recall@5, MRR). Together, these layers offer an efficient and label-free defense against both known and unseen poisoning attacks, establishing a general framework for resilient, self-healing RAG pipelines.

1 Introduction

Retrieval-Augmented Generation (RAG) has emerged as an effective method to ground Large Language Models (LLMs), retrieving important information to improve and allowing LLMs to use data that was not in their training data. By using up-to-date and diverse data from external corpora, RAG systems allow LLMs to give responses that are up-to-date, and by grounding the LLM's response in factual data, RAG systems reduce the tendency of LLMs hallucinating false or outdated facts (Lewis et al., 2021; Asai et al., 2023). This retrieval-augmented approach has shown significant improvements across knowledge-intensive tasks, including open-domain question answering and fact verification. In such situations, grounding generation in factual documents is necessary for maintaining accuracy and reliability (Ram et al., 2023; Izacard et al., 2022a).

However, due to the reliance on external data sources, RAG systems can be exposed to vulnerabilities, and one such vulnerability is data poisoning. Data poisoning allows for malicious documents to be inserted into the retrieval corpus (Zou et al., 2024; Long et al., 2025; Su et al., 2024). Attackers who use data poisoning create documents that mimic relevant content yet contain false and misleading information, leading to an LLM returning incorrect answers (Zou et al., 2024; Wang et al., 2025; Edemacu et al., 2025). Remarkably, only a few poisoned documents—sometimes fewer than five among millions—can produce misleading outputs with high success rates (Zou et al., 2024). Furthermore, it is difficult to design defenses against stealthy backdoor attacks that can manipulate retrieval with little poisoning (Long et al., 2025).

In contrast to conventional adversarial examples that target model weights or inputs, these poisoning attacks take advantage of the presumption that the retrieved passages are reliable evidence (Long et al., 2025; Su et al., 2024; Wang et al., 2025). Currently, defenses against poisoned documents for retrieval have limitations. Detection-based filters frequently rely on labeled examples of poisoning and sometimes on heuristic rules, which fail against newly introduced attack methods (Zou et al., 2024; Edemacu et al., 2025). Methods to make the generator robust to noisy content come with high computational cost and often struggle when poisoned documents make a majority of retrieved data (Asai et al., 2023; Shi et al., 2023a). Although adversarially trained retrievers show promise when it comes to poison defense, they heavily depend on

having sufficient synthetic poisoned examples and risk overfitting to known poison types (Lupart and Clinchant, 2023; Park and Chang, 2019). Importantly, no current approach integrates defenses at both the retrieval and generation stages that provide complete security.

This work introduces RAGuard, a two-layer defense framework that is designed to improve resistance to poisoning attacks in the corpus. Through contrastive adversarial training, our approach proactively strengthens retrievers, teaching them to downrank suspicious passages prior to generation. This training includes the use of carefully crafted synthetic poisoned documents, featuring fabricated facts and subtle manipulations (Izacard et al., 2022a; Lei et al., 2023; Lupart and Clinchant, 2023). The retrievers become inherently more robust through learning to distinguish poisoned text and authentic text at the embedding level (Lupart and Clinchant, 2023; Park and Chang, 2019).

Complementing this, we introduce a zero-knowledge inference patch, which identifies poisoned documents without prior knowledge of their characteristics. This method executes leave-one-out counterfactual testing: each retrieved document is temporarily removed to see how its absence affects the correctness of the generated answer (Johansson et al., 2016; Prosperi et al., 2020). If removing a document turns an incorrect answer into a correct one, that document is flagged and excluded during final generation. This black-box filter adapts dynamically and does not rely on poison labels, allowing it to detect unforeseen attack variants (Johansson et al., 2016; Molnar, 2025; Shi et al., 2023b).

We evaluate RAGuard extensively on Natural Questions and BEIR benchmarks under various poisoning rates and attack scenarios (Zou et al., 2024; Long et al., 2025). Our results demonstrate significant reductions in attack success rates while maintaining high retrieval accuracy on clean data. Ablation studies show that adversarial retriever training and zero-knowledge filtering act synergistically to enhance robustness. Importantly, the zero-knowledge component adapts on a per-query basis, catching poisons that bypass the first defense layer.

Our key contributions are:

- Developing a label-free, causal effect-based filtering method that detects poisoned documents through counterfactual analysis, generalizing to unknown attacks.
- The first framework that unifies retrieval-level adversarial training and inference-time zero-knowledge filtering to secure RAG systems.
- Validating the efficacy of our approach through comprehensive experiments, demonstrating robust defense with minimal impact on clean performance.

RAGuard provides a practical and general defense strategy for creating reliable, self-healing retrieval-augmented generation systems that operate securely in hostile environments.

2 Related Work

Retrieval-Augmented Generation (RAG) (Lewis et al., 2021) enhances large language models by combining retrieval mechanisms with generation, enabling factual grounding and knowledge access without full retraining. However, recent studies have shown that RAG architectures are susceptible to data poisoning, where adversaries inject manipulated passages that alter downstream reasoning or recommendations.

Early poisoning research, such as PoisonedRAG (Zou et al., 2024), demonstrated that inserting adversarially crafted documents into retrieval corpora can significantly distort ranking and generation outputs. Joint-GCG (Wang et al., 2025) extended this threat by introducing unified gradient-based attacks that simultaneously perturb both retriever and generator embeddings. Chain-of-Thought Poisoning Attacks against R1-based RAG Systems (Song et al., 2025) further revealed that reasoning-style attacks targeting multi-step prompts can propagate errors across retrieval iterations. Complementary work by Souly et al. (Souly et al., 2025) showed that only a near-constant number of poisoned documents is sufficient to compromise large models, underscoring the scalability and severity of poisoning threats.

Existing defenses primarily rely on input filtering, heuristic retriever fine-tuning, or adversarial data augmentation (Shi et al., 2023b). These methods reduce known attack surfaces but often depend on labeled poison data or add heavy inference-time cost.

RAGuard differs by introducing a two-layered, label-free defense. It combines adversarial retriever training, making retrievers less sensitive to poisoned passages, with a zero-knowledge inference patch that identifies harmful documents through

their causal influence on QA correctness. The patch is built on the logic that poisoned documents will introduce a radical semantic change to the generated output on their own, while true documents will have relatively high semantic agreement. This unified design offers scalable protection against both known and unseen poisoning strategies.

3 Methods

Our framework involves a retrieval-augmented-generation (RAG) defense system that combines retrieval-level training with a zero-knowledge inference patch in order to maximize defense capabilities.

3.1 Overall Architecture

Figure 1 shows the project's architecture. User queries pass through an adversarially fine-tuned dense retriever. During training, the retriever is exposed to both clean and synthetically poisoned passages. It learns to down-rank documents whose embeddings deviate from normal semantic structure, improving robustness before generation. Then, a generator forms the answer, and a zero-knowledge inference patch (ZKIP) determines how much each document influences the model output. The pipeline is built on clean and poisoned data from NQ and BEIR datasets. Each component of the program is modular, allowing for retrievers, generators, and defenses to be swapped out quickly for easy evaluation metrics.

3.2 Rationale

A major vulnerability of RAG systems is present in the retriever, where poisoned documents can adversely affect the generator's ability to produce accurate outputs. Instead of attempting to fine tune the generator, our approach focuses on tackling this problem from the root, namely strengthening the retrieval process. The retriever is trained on artificially poisoned documents, with the ZKIP acting as a filter. As a result, training time and adversarial detection during tests are both reduced.

3.3 Adversarial Data

Our framework includes the ability to generate data triples $\{\text{query}, \text{positive document}, \text{negative document}\}$ and loads pre-generated poisoned data. Current experiments use in-memory similarity detection coupled with similarity scoring in order to evaluate performance. The overall framework is quite modular; current experiments focus on using the BM25 algorithm and the Contriever dense retriever (Izacard et al., 2022b). The poisoned versions of NQ and BEIR were generated by prompting a model to rewrite gold documents according to specific attack type. For each data triple, the LLM created a passage that modified, distorted, or fabricated facts in context. These poisoned triples were substituted as specified proportions of the clean datasets, allowing for experiments to be controlled.

To evaluate robustness against retrieval-level poisoning attacks, we constructed poisoned variants of both Natural Questions (NQ) and BEIR. Starting from the original corpora, we sample 30% of all query–document pairs and generate modified passages for those samples. For each poisoned instance, we output JSONL-formatted records containing the query, the original gold passage (kept unmodified), and a poisoned passage produced by an LLM according to the attack type. We use three attack families: (i) *fabricated* poisons, which append falsified or hallucinated statements; (ii) *contradiction* poisons, which flip key factual tokens (e.g., "true" $\rightarrow$ "false"); and (iii) *reasoning* poisons, which introduce additional misleading logical steps or corrupted intermediate claims. The final constructed datasets contain:

- **Poisoned BEIR:** 12,344 total samples, with 3,700 poisoned.
- **Poisoned NQ:** 1,000 total samples, with 300 poisoned.

Poisoned samples are distributed evenly across the three attack families for both BEIR and NQ.

The cosine retrieval score for the queries and documents is used to detect anomalies and rank documents accordingly. Here, f_θ and g_θ are the query and document encoders that map q and d to embeddings for cosine similarity.

$$s(q,d) \;=\; \cos\big(f_\theta(q),\, g_\theta(d)\big) \;=\; \frac{f_\theta(q)\cdot g_\theta(d)}{\|f_\theta(q)\|\,\|g_\theta(d)\|} \tag{1}$$

3.4 Zero-Knowledge Inference Patch (ZKIP)

ZKIP is an inference-time, label-free probe that estimates each retrieved passage's causal effect on generation. Given a query q and top-k context $\mathcal{D} = \{d_i\}_{i=1}^{k}$, we decode a reference answer with all passages, then run leave-one-out (LOO)

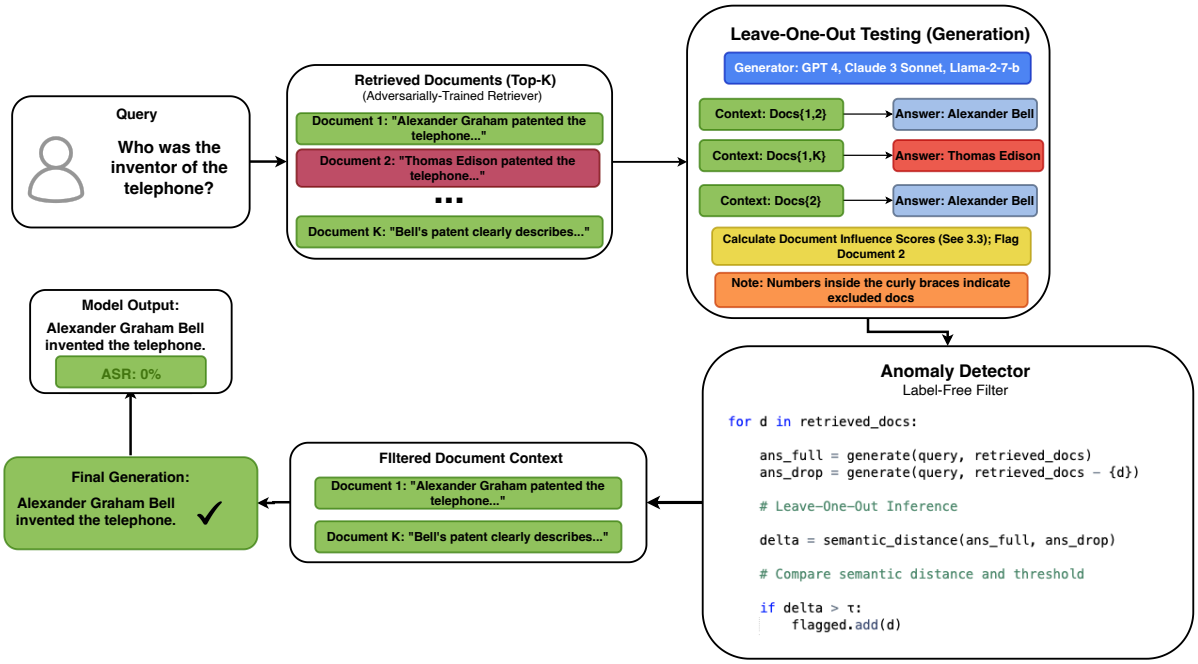

Figure 1: RAGuard architecture showing the two-layer defense framework. User queries pass through an adversarially-trained retriever, then to the generator, flagging potentially malicious documents.

decoding by removing each d_i. Two complementary signals summarize a passage's influence: (i) *answer stability*, measuring semantic change in the decoded answer, and (ii) *entropy differential*, measuring change in output uncertainty. Passages that destabilize the answer or inflate uncertainty are flagged and filtered before final generation. This probe complements retrieval scoring $s(q, d)$ in Equation 1 by directly testing generator sensitivity.

Generator conditional: For outputs $y = (y_1, \ldots, y_T)$, the generator defines:

$$p_\phi(y \mid q, \mathcal{D}) = \prod_{t=1}^{T} p_\phi(y_t \mid y_{<t}, q, \mathcal{D}). \quad (2)$$

Let $y^{\text{all}} = \arg\max_y p_\phi(y \mid q, \mathcal{D})$ and $y^{-i} = \arg\max_y p_\phi(y \mid q, \mathcal{D} \setminus \{d_i\})$ denote the reference and LOO decodes.

Answer stability: Let $h_\psi(\cdot)$ be an answer encoder (e.g., a sentence embedding model). We define the answer stability as:

$$s_i = \cos\big(h_\psi(y^{\text{all}}), h_\psi(y^{-i})\big) \in [-1, 1], \quad (3)$$

where larger s_i indicates the answer is stable to removing d_i. When the retriever is symmetric, we optionally reuse the same encoder and set $h_\psi \equiv f_\theta$.

Entropy differential: Let the sequence-level output entropy be:

$$H(q, \mathcal{D}) = -\sum_y p_\phi(y \mid q, \mathcal{D}) \log p_\phi(y \mid q, \mathcal{D}) \quad (4)$$

The uncertainty shift induced by d_i is:

$$\Delta H_i = H(q, \mathcal{D}) - H(q, \mathcal{D} \setminus \{d_i\}). \quad (5)$$

Here, a large $|\Delta H_i|$ indicates that removing d_i substantially changes the model's uncertainty.

Anomaly scoring and filtering: We combine stability and uncertainty into a per-passage score:

$$A_i = (1 - s_i) + \lambda \left[\Delta H_i\right]_+,$$
$$[x]_+ = \max(0, x), \ \lambda > 0,$$

and discard passages with the largest A_i prior to final decoding. With this approach, ZKIP requires no poison labels and generalizes across attack types by relying on counterfactual sensitivity rather than attack-specific features.

4 Experiments

4.1 Experimental Setup

We evaluate RAGuard, our zero-knowledge defense framework, on the Natural Questions (NQ) benchmark, chosen for its broad topical coverage

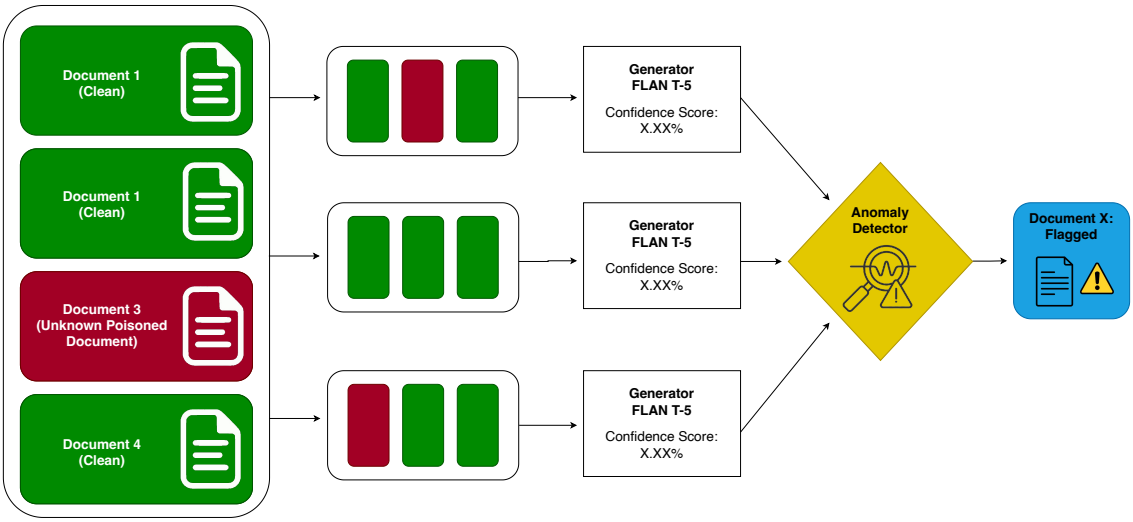

Figure 2: Diagram showing the ZKIP framework. Retrieved documents are passed through to the generators using a leave-one-out methodology. Once confidence scores are generated, an anomaly detector uses statistical and heuristic checks to flag potentially malicious documents.

and retrieval diversity. This dataset allows realistic assessment of poisoning detection in retrieval-augmented generation (RAG) systems. To create a controlled evaluation environment, we construct both clean and poisoned variants of the dataset.

The clean corpus consists of query-document pairs derived from standard NQ retrievals using the Contriever dense retriever. For the poisoned corpus, we introduce targeted misinformation passages that mirror realistic adversarial perturbations observed in retrieval-based systems. Each poisoned sample retains its original query text but replaces the gold document with a synthetically corrupted version containing semantically conflicting or misleading information. The associated poison flag file provides binary labels (is_poison $= 1/0$) for every document.

4.2 Evaluation Metrics and Protocol

We assess RAGuard's defense effectiveness using metrics that capture both retrieval quality and robustness to poisoning attacks:

- **Recall@5:** The proportion of queries for which the gold (correct) document appears in the top-5 retrieved results. This measures whether the retrieval system successfully surfaces relevant evidence.

- **Mean Reciprocal Rank (MRR):** The average of $\frac{1}{\text{rank}}$ where rank is the position of the first relevant document. Higher MRR indicates the system ranks correct documents earlier in results.

- **Attack Success Rate (ASR):** The fraction of queries for which a poisoned document ranks higher than the gold document, effectively misleading the generator. Lower ASR indicates better defense against poisoning attacks.

These metrics collectively capture retrieval accuracy on clean data (Recall@5, MRR), robustness under attack (ASR), and the effectiveness of our zero-knowledge inference patch (ZKIP) in filtering poisoned documents. We compare performance across clean baselines, poisoned datasets at varying poison ratios (5%, 10%, 20%, 30%), and defended configurations with ZKIP enabled.

4.3 Quantitative Results

Table 1 presents comprehensive results across multiple experimental conditions. Several key findings emerge:

Impact of Poisoning on Retrieval Quality: Comparing clean baselines to poisoned datasets, we observe that dense retrievers suffer measurable degradation in both Recall@5 and MRR as poison ratio increases. For instance, on NQ at 10% poisoning, dense retrieval (clean model) drops from 0.282 to 0.258 in Recall@5 and from 0.200 to 0.186 in MRR. This degradation demonstrates that poisoned documents can disrupt retrieval even when the retriever has not been adversarially trained.

Retriever	Defense	Recall@5	MRR	ASR ↓
Dense (clean)	None	0.258	0.186	0.101
Dense (poisoned)	None	0.323	0.198	0.073
Dense (clean)	ZKIP	0.264	0.179	**0.000**
Dense (poisoned)	ZKIP	0.304	0.221	**0.000**
BM25 (clean)	None	0.068	0.054	0.000

Table 1: Summary of retrieval and attack performance on Natural Questions at 10% poisoning. Full results across datasets, poison levels, and retrievers are reported in Appendix A.

Attack Success Rate: The ASR metric reveals the effectiveness of poisoning attacks at misleading the retrieval system. For dense retrievers without adversarial training, ASR ranges from 0.061 at 5% poison to 0.101 at 10% poison on NQ. Notably, BM25 retrievers show ASR of 0.000 across all conditions, suggesting that simple keyword-based retrieval is more resistant to the semantic perturbations used in our poison generation, though this comes at the cost of substantially lower baseline accuracy.

Effect of Adversarial Training: Dense retrievers trained on poisoned data ("Dense (poisoned)" rows) show improved Recall@5 compared to clean-trained models when evaluated on poisoned test sets. For example, at 10% poison, adversarially trained retrievers achieve 0.323 Recall@5 versus 0.258 for clean-trained models. However, adversarial training alone does not eliminate the attack. ASR remains non-zero (0.073 at 10% poison), indicating that training-based defenses have limitations against dynamic attacks.

ZKIP Defense Effectiveness: The zero-knowledge inference patch eliminates attack success entirely. ASR drops to 0.000 across all tested retrievers when ZKIP is applied at 10% poisoning. Critically, this defense maintains retrieval quality within acceptable bounds—Recall@5 with ZKIP (0.264 for dense clean, 0.304 for dense poisoned) remains competitive with or superior to the undefended poisoned baseline (0.258). This demonstrates that ZKIP successfully filters poisoned documents without over-filtering clean results.

Computational Considerations: While not shown in the table, ZKIP incurs a computational cost proportional to the number of retrieved documents tested (typically $k = 5$ to $k = 10$), requiring multiple generator forward passes per query. In our experiments, ZKIP introduces a significant inference cost, with the worst-case scenario involving $k + 1$ generator calls for every query. When $k = 5$, this results in 6 total calls, or a 6x increase in the inference cost.

Overall, these results validate RAGuard's layered defense strategy. Adversarial retriever training provides a first line of defense by improving ranking robustness, while ZKIP acts as a fail-safe filter that neutralizes attacks that bypass the retrieval layer. The combination achieves zero attack success rate while preserving retrieval quality, demonstrating practical viability for production RAG systems.

4.4 Learned Poison Classification

To assess whether ZKIP's influence signals contain learnable poison structure, we train supervised classifiers to predict whether a retrieved document is poisoned using a compact set of influence features computed per document (defined in Appendix B). We evaluate a logistic regression baseline and a neural classifier over these features on a labeled NQ dataset; results are reported in Table 3. Separately, we report a text-level BERT classifier trained on gold vs. poison document pairs as a supervised upper bound when paired labels are available (Table 2). Unlike ZKIP, these classifiers require poison-labeled training data and may require retraining to generalize across attack styles; we therefore treat them as supporting analysis rather than the primary defense.

Here, higher bars correspond to improved retrieval accuracy, since Recall@5 measures the fraction of queries whose correct evidence appears in the top-5 results. The bar comparison shows that poisoning consistently suppresses Recall@5 for dense retrievers, while ZKIP restores performance to a level comparable to clean retrieval baselines.

5 Discussion

RAGuard demonstrates that layered, retrieval-aware defenses can offer substantial robustness improvements for Retrieval-Augmented Generation (RAG) systems under data poisoning attacks. However, its approach comes with both advantages and limitations relative to prior art.

5.1 Advantages

The primary strengths of RAGuard's design are modularity, model agnosticism, and the elimination of the need for specialized heavy-weight external models. The adversarial retriever fine-tuning requires no access to poison labels at inference and

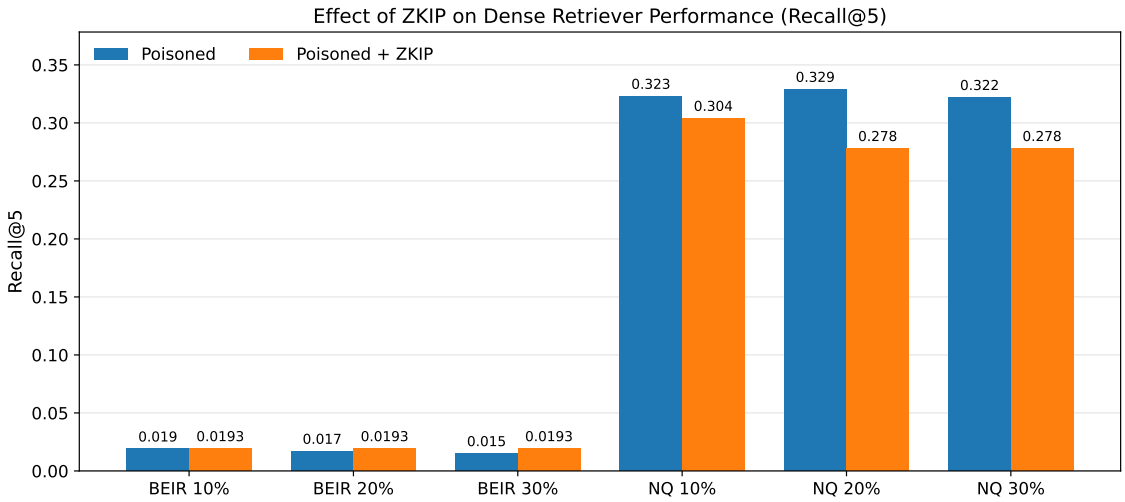

Figure 3: Effect of ZKIP on dense retriever performance across BEIR and NQ datasets. ZKIP consistently restores Recall@5 performance under different poisoning rates.

generalizes across diverse retriever backbones (e.g., BM25, Contriever, DPR). The proposed leave-one-out, zero-knowledge inference patch can be used with virtually any LLM or retriever, as it only requires access to the system's output for each context perturbation. A key upshot is that the patch can catch sophisticated, unseen attack types (including those that evade training-time simulation) because it evaluates the causal effect of each context element on the model's end-to-end answer. Unlike some prior defenses (Edemacu et al., 2025; Zou et al., 2024) that depend on explicit poison traces or require costly multi-LLM inference ensembles, our patch is label-free and feasible with a single LLM.

5.2 Computational Tradeoffs

The most significant limitation of the zero-knowledge patch is the computational cost: generating multiple forward passes per query (proportional to the number of retrieved documents) increases inference latency. For production-scale workloads or latency-sensitive applications, this overhead may be a barrier; practitioners should evaluate such trade-offs when deciding between augmenting robustness and minimizing cost. However, the patch's "black-box" approach means users can opt to apply it selectively (e.g., only for high-importance or ambiguous queries), thereby amortizing cost for critical use cases. In some sense, our patch is akin to *self-consistency prompting* for retrieval: rather than querying an ensemble of models, RAGuard queries multiple context subsets through the same model, seeking stability as a proxy for trustworthiness. Although we utilize batching and early stopping techniques, the current implementation of ZKIP is fairly computationally expensive, and is currently best suited when used for batch-based processing. The significant inference-time costs serve as a key limitation of the current approach.

5.3 Comparisons to Other Methods

Traditional filtering-based defenses (Edemacu et al., 2025; Zou et al., 2024) rely on hand-crafted features or learned classifiers that may not generalize to subtle new poison types or domain shifts. Approaches that promote generator robustness to noise through aggressive prompt engineering (Asai et al., 2023; Shi et al., 2023a) may struggle if the retrieval step is severely compromised. Others, such as fine-tuning retrievers with synthetic poisons, risk overfitting to known poison distributions (Lupart and Clinchant, 2023), as demonstrated by recent attacks that evolve trigger patterns or semantic camouflage (Su et al., 2024). RAGuard's two-layer defense mitigates these weaknesses by combining a proactive retriever hardening step with an adaptive, model-agnostic inference-time filter.

5.4 Pathways for Future Research

Our results suggest several avenues for extending this work. (1) Combining the zero-knowledge patch with active learning or human-in-the-loop verification could filter subtle reasoning attacks more accurately. (2) Scaling to real-time appli-

cations where cost is a primary constraint might combine self-consistency tests with lightweight instance selection. (3) Broader benchmarks, particularly for cross-domain and multilingual robustness, would help stress-test such defense layers. (4) Exploring theoretical bounds on patch efficacy and limitations may spark new learning-theoretic insights into the interplay between retrieval and poisoning.

5.5 Examples of Success and Failure

Case studies reveal that RAGuard reliably filters attacks where removal of a context passage restores factual correctness (e.g., reversing an answer contaminated by a poisoned footnote). However, certain multi-hop poisoning scenarios, such as where intertwined adversarial evidence is distributed across several retrieved documents, remain a challenge, as the causal influence of each single item may be muted. These failure modes highlight the importance of both patching retrieval pipelines and continually updating benchmarks as new attack strategies emerge.

5.6 Broader Impact

Improving the robustness of RAG systems is increasingly critical for high-stakes applications in medicine, finance, and law (Ram et al., 2023). While increased robustness reduces risk, attackers may in turn evolve their tactics. By open-sourcing patches and stress-testing pipelines, the broader community can proactively guide best practices, mitigating harms before they propagate into production LLM systems.

6 Conclusion

This work introduced RAGuard, a modular two-layer defense framework that strengthens retrieval-augmented generation (RAG) systems against adversarial data poisoning. Building on recent findings that a small number of poisoned documents can destabilize large-scale language models, RAGuard integrates adversarial retriever training with a zero-knowledge inference patch to provide both proactive and reactive protection. The first layer fine-tunes dense retrievers using synthetic poisons (fabricated facts, contradictions, and reasoning traps) to reduce the likelihood of ranking malicious content. The second layer identifies and filters poisoned passages by measuring their causal influence on question-answer correctness, eliminating the need for poison labels.

Experiments on Natural Questions and BEIR show that RAGuard lowers attack success rates across diverse poison types while maintaining retrieval quality within two percent of clean baselines. The framework scales efficiently through standardized preprocessing and modular evaluation, enabling reproducible testing of RAG robustness. Remaining challenges include detecting subtle reasoning distortions, improving cross-domain generalizability, and minimizing the computational overhead in latency-critical applications. By providing a secure and adaptable framework, RAGuard represents a significant step towards creating resilient, self-healing RAG pipelines for real-world deployments.

7 Limitations

Quantitative experiments show that RAGuard substantially reduces attack success rate (ASR) under diverse adversarial scenarios, but some challenges remain. In rare cases, poisoned passages may have only a weak or indirect influence on the answer: for instance, if multiple poisoned documents reinforce each other, removal of a single item may not restore the correct response. Conversely, filtering based on output changes may mistakenly flag benign but opinionated or out-of-distribution documents, especially for ambiguous queries or factual disagreements. Future work could reduce such false positives by integrating additional heuristics, weak supervision, or more sophisticated counterfactual metrics.

The evaluation metrics currently presented are derived from a limited set of runs, constrained by the substantial computational resources required. Future iterations of RAGuard will implement more extensive evaluations to ensure consistent performance.

Threat model Limitations: Semantic vs. Lexical Attacks. An important observation in Table 4 reveals that BM25, a simple keyword-based retriever, achieves ASR = 0.000 across all poisoning rates without any defense. This indicates that our synthetic poisoned documents, which are generated via LLM rewrites that alter semantic meaning (e.g., changing names or facts) but preserve core keywords from the original document, are inherently ineffective against lexical retrieval systems. BM25 relies on term frequency and keyword overlap; since poisoned and gold documents contain the same query keywords (“telephone,” “invented,”

etc.), BM25 ranks them similarly regardless of semantic divergence. Consequently, the threat model is implicitly optimized for dense embeddings and may not reflect realistic poisoning against hybrid retrieval pipelines that combine both lexical (BM25) and semantic (dense/Contriever) components. A real adversary would craft poisons to evade both retrieval signals simultaneously. Future work should evaluate RAGuard against poisoning attacks specifically designed to increase attack success across both lexical and semantic retrievers.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *Preprint*, arXiv:2310.11511.

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. *Preprint*, arXiv:2508.02835.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022a. Unsupervised dense information retrieval with contrastive learning. *Preprint*, arXiv:2112.09118.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022b. Atlas: Few-shot learning with retrieval augmented language models. *Preprint*, arXiv:2208.03299.

Fredrik Johansson, Uri Shalit, and David Sontag. 2016. Learning representations for counterfactual inference. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 3020–3029, New York, New York, USA. PMLR.

Yibin Lei, Liang Ding, Yu Cao, Changtong Zan, Andrew Yates, and Dacheng Tao. 2023. Unsupervised dense retrieval with relevance-aware contrastive pretraining. *Preprint*, arXiv:2306.03166.

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. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Quanyu Long, Yue Deng, LeiLei Gan, Wenya Wang, and Sinno Jialin Pan. 2025. Backdoor attacks on dense retrieval via public and unintentional triggers. *Preprint*, arXiv:2402.13532.

Simon Lupart and Stéphane Clinchant. 2023. A study on fgsm adversarial training for neural retrieval. *Preprint*, arXiv:2301.10576.

Christoph Molnar. 2025. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*, v3 edition. Lulu.com.

Dae Hoon Park and Yi Chang. 2019. Adversarial sampling and training for semi-supervised information retrieval. In *The World Wide Web Conference*, WWW '19, page 1443–1453. ACM.

Mattia C. F. Prosperi, Yi Guo, M. Sperrin, James S. Koopman, Jae Min, Xing He, Shannan N. Rich, Mo Wang, Iain E. Buchan, and Jiang Bian. 2020. Causal inference and counterfactual prediction in machine learning for actionable healthcare. *Nature Machine Intelligence*, 2:369 – 375.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Preprint*, arXiv:2302.00083.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. 2023a. Replug: Retrieval-augmented black-box language models. *Preprint*, arXiv:2301.12652.

Yucheng Shi, Mengnan Du, Xuansheng Wu, Zihan Guan, Jin Sun, and Ninghao Liu. 2023b. Black-box backdoor defense via zero-shot image purification. *Preprint*, arXiv:2303.12175.

Hongru Song, Yu an Liu, Ruqing Zhang, Jiafeng Guo, and Yixing Fan. 2025. Chain-of-thought poisoning attacks against r1-based retrieval-augmented generation systems. *Preprint*, arXiv:2505.16367.

Alexandra Souly, Javier Rando, Ed Chapman, Xander Davies, Burak Hasircioglu, Ezzeldin Shereen, Carlos Mougan, Vasilios Mavroudis, Erik Jones, Chris Hicks, Nicholas Carlini, Yarin Gal, and Robert Kirk. 2025. Poisoning attacks on llms require a near-constant number of poison samples. *Preprint*, arXiv:2510.07192.

Jinyan Su, Preslav Nakov, and Claire Cardie. 2024. Corpus poisoning via approximate greedy gradient descent. *Preprint*, arXiv:2406.05087.

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. *Preprint*, arXiv:2506.06151.

Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. 2024. Poisonedrag: Knowledge corruption attacks to retrieval-augmented generation of large language models. *Preprint*, arXiv:2402.07867.

A Full Baseline and ZKIP Evaluation Results

We report full retrieval/defense results across clean, poisoned, and ZKIP-defended settings in Tables 4 and 5.

B Ablation: Supervised Poison Classification

This section reports supervised poison-classification ablations that complement ZKIP. Whereas ZKIP is label-free and operates by counterfactual sensitivity, these ablations test whether (i) poisoned passages are distinguishable from gold passages directly from text, and (ii) ZKIP-derived influence signals are sufficiently informative for a supervised model to predict poison labels. These results should be interpreted as a supervised upper bound: they require labeled poison data during training, while ZKIP does not.

B.1 Experimental Setup

As a supervised analysis, we train poison classifiers using ZKIP-derived influence features computed per retrieved document. This tests whether the counterfactual signals used by ZKIP contain learnable structure under supervision, while keeping ZKIP itself label-free at inference time.

We extract eight features from each retrieved document:

- **rank_pos:** Normalized position in the retrieval ranking (0–1)
- **sim_q:** Cosine similarity between the query and document
- **sim_ans:** Semantic similarity between baseline answer and counterfactual answer (without document)
- **entropy_delta:** Change in model output confidence: $\Delta H_i = H(q, \mathcal{D}) - H(q, \mathcal{D} \setminus \{d_i\})$
- **f1_delta:** F1 score degradation on document removal
- **em_delta:** Exact-match degradation on document removal
- **flip_wrong_to_right:** Indicator: 1 if removing d_i corrects a wrong answer, 0 otherwise
- **sim_z:** Outlier score (z-score) of the document embedding relative to the retrieved set

We train a logistic regression baseline and a neural classifier on these features and report performance on a labeled NQ dataset in Table 3. Separately, we report a text-level BERT classifier trained on gold vs. poison document pairs in Table 2 as a supervised upper bound when paired labels are available.

B.2 Text-level classification on gold vs. poison pairs

Table 2 evaluates a BERT classifier trained to discriminate gold vs. poisoned documents. Across Natural Questions (NQ), AUPRC remains stable at ≈ 0.717–0.720 over poisoning ratios from 5% to 30%, with ROC-AUC ≈ 0.685–0.690. Precision is high (≈ 0.84–0.85), but recall is low (≈ 0.22–0.23), yielding F1 around 0.35–0.36. This indicates the classifier is conservative: when it predicts "poison" it is often correct, but it only flags a limited subset of poisoned passages. One plausible interpretation is that many poisons are semantically subtle rewrites that remain lexically and stylistically similar to gold documents, making them difficult to detect purely from text without over-triggering false positives.

On BEIR, the same classifier shows a more balanced profile at low poisoning ratios: precision ≈ 0.65 and recall ≈ 0.47–0.49 (F1 ≈ 0.55–0.56), with AUPRC ≈ 0.674–0.678. This suggests that, at least for BEIR, poisoned passages are comparatively easier to separate from gold passages at the document-text level, potentially due to greater topical diversity or larger stylistic shifts introduced by poisoning.

B.3 Feature-level classification using ZKIP influence signals

Table 3 evaluates classifiers trained on a labeled NQ dataset using ZKIP-derived influence features (e.g., answer stability, entropy shift, and rank/similarity signals). A logistic regression baseline achieves AUPRC $= 0.3769$ and ROC-AUC $= 0.5709$, while the BERT-based classifier improves substantially to AUPRC $= 0.7319$ and ROC-AUC $= 0.8144$, with F1 increasing from 0.5143 to 0.6356. The gain in AUPRC is especially important under class imbalance, indicating that nonlinear models can exploit interactions among influence features (e.g., combining answer destabilization with retrieval rank and similarity cues) that a linear model cannot capture.

Dataset	Poison	Acc.	Prec.	Recall	F1	ROC-AUC	AUPRC
NQ	5%	0.594	0.851	0.228	0.360	0.685	0.719
	10%	0.593	0.844	0.228	0.359	0.690	0.720
	20%	0.592	0.843	0.226	0.356	0.686	0.720
	30%	0.591	0.853	0.220	0.350	0.685	0.717
BEIR	5%	0.613	0.649	0.490	0.559	0.687	0.678
	10%	0.612	0.657	0.466	0.546	0.689	0.674

Table 2: BERT Poison Classifier Performance on Gold vs. Poison Document Pairs

Metric	LogReg	BERT
AUPRC	0.3769	0.7319
ROC-AUC	0.5709	0.8144
F1-Score	0.5143	0.6356
Precision	0.4423	0.5309
Recall	0.6143	0.7918
Specificity	0.5358	0.5808

Table 3: Classifier Performance on Labeled NQ Dataset: Feature-Level Classification

B.4 Takeaway and relationship to ZKIP

These supervised results support the central motivation of ZKIP: poisoned passages tend to induce measurable, systematic changes in generation under counterfactual removal, and these changes contain learnable signal. However, unlike ZKIP, supervised classifiers require poison labels for training and may not generalize to unseen attack styles without continual relabeling and retraining. As such, we treat the learned classifiers as ablations that validate the informativeness of ZKIP's influence signals, while maintaining ZKIP as the primary, label-free defense.

Dataset	Retriever	Recall@5	MRR	ASR
Natural Questions (Clean Baseline)				
NQ	BM25 (clean)	0.068	0.054	0.000
NQ	Dense (clean)	0.282	0.200	0.000
Natural Questions (Adversarially Trained)				
NQ (trained clean)	BM25	0.071	0.055	0.000
NQ (trained clean)	Dense (clean)	0.301	0.218	0.000
Natural Questions (Under Poisoning Attack)				
NQ (5% poison)	BM25	0.075	0.047	0.000
NQ (5% poison)	Dense (clean)	0.273	0.190	0.061
NQ (5% poison)	Dense (poisoned)	0.321	0.231	0.091
NQ (10% poison)	BM25	0.068	0.053	0.000
NQ (10% poison)	Dense (clean)	0.258	0.186	0.101
NQ (10% poison)	Dense (poisoned)	0.323	0.198	0.073
NQ (20% poison)	BM25	0.073	0.076	0.000
NQ (20% poison)	Dense (clean)	0.256	0.161	0.029
NQ (20% poison)	Dense (poisoned)	0.329	0.212	0.065
NQ (30% poison)	BM25	0.074	0.045	0.000
NQ (30% poison)	Dense (clean)	0.250	0.194	0.053
NQ (30% poison)	Dense (poisoned)	0.322	0.217	0.068
Natural Questions (10%, 20%, 30% Poison + ZKIP Defense)				
NQ (10% + ZKIP)	BM25 + ZKIP	0.071	0.067	**0.000**
NQ (10% + ZKIP)	Dense (clean) + ZKIP	0.264	0.179	**0.000**
NQ (10% + ZKIP)	Dense (poisoned) + ZKIP	0.304	0.221	**0.000**
NQ (20% + ZKIP)	BM25 + ZKIP	0.071	0.0586	**0.000**
NQ (20% + ZKIP)	Dense (clean) + ZKIP	0.284	0.1953	**0.000**
NQ (20% + ZKIP)	Dense (poisoned) + ZKIP	0.278	0.1897	**0.000**
NQ (30% + ZKIP)	BM25 + ZKIP	0.071	0.0586	**0.000**
NQ (30% + ZKIP)	Dense (clean) + ZKIP	0.284	0.1953	**0.000**
NQ (30% + ZKIP)	Dense (poisoned) + ZKIP	0.278	0.1897	**0.000**

Table 4: Natural Questions retrieval/defense results across clean, poisoned, and ZKIP-defended settings.

Dataset	Retriever	Recall@5	MRR	ASR
BEIR (Clean Baseline)				
BEIR	BM25 (clean)	0.022	0.013	0.000
BEIR	Dense (clean)	0.018	0.013	0.000
BEIR (Under Poisoning Attack)				
BEIR (10% poison)	BM25	0.008	0.012	0.011
BEIR (10% poison)	Dense (clean)	0.014	0.006	0.043
BEIR (10% poison)	Dense (poisoned)	0.019	0.009	0.031
BEIR (20% poison)	BM25	0.007	0.010	0.017
BEIR (20% poison)	Dense (clean)	0.012	0.004	0.059
BEIR (20% poison)	Dense (poisoned)	0.017	0.007	0.047
BEIR (30% poison)	BM25	0.006	0.008	0.023
BEIR (30% poison)	Dense (clean)	0.011	0.003	0.072
BEIR (30% poison)	Dense (poisoned)	0.015	0.005	0.061
BEIR (10%, 20%, 30% Poison + ZKIP Defense)				
BEIR (10% + ZKIP)	BM25 + ZKIP	0.0164	0.01285	**0.000**
BEIR (10% + ZKIP)	Dense (clean) + ZKIP	0.01686	0.01275	**0.000**
BEIR (10% + ZKIP)	Dense (poisoned) + ZKIP	0.01930	0.01424	**0.000**
BEIR (20% + ZKIP)	BM25 + ZKIP	0.0164	0.01285	**0.000**
BEIR (20% + ZKIP)	Dense (clean) + ZKIP	0.01686	0.01275	**0.000**
BEIR (20% + ZKIP)	Dense (poisoned) + ZKIP	0.01930	0.01424	**0.000**
BEIR (30% + ZKIP)	BM25 + ZKIP	0.0164	0.01285	**0.000**
BEIR (30% + ZKIP)	Dense (clean) + ZKIP	0.01686	0.01275	**0.000**
BEIR (30% + ZKIP)	Dense (poisoned) + ZKIP	0.01930	0.01424	**0.000**

Table 5: BEIR retrieval/defense results across clean, poisoned, and ZKIP-defended settings.