*Proceedings of Machine Learning Research* **318** (2026)
The 39th Canadian Conference on Artificial Intelligence

Canadian AI 2026
25–29 May 2026, Vancouver, BC

# Defending RAG Against Knowledge Poisoning Using Cross-Encoder Activation Signals

Razieh Moradi[†], Havva Alizadeh Noughabi[‡], Fattane Zarrinkalam[†,‡,*], Ali Dehghantanha[‡]
[†] College of Engineering, University of Guelph, Guelph, ON, Canada
[‡] Cyber Science Lab, School of Computer Science, University of Guelph, Guelph, ON, Canada

**Abstract**

Retrieval-Augmented Generation (RAG) improves the factuality of large language models (LLMs) by grounding outputs in externally retrieved evidence, but it also inherits security risks from the underlying corpus. In particular, an adversary can poison the knowledge source so that injected passages are retrieved and steer the model toward attacker-chosen targets. We propose **C**ross-**E**ncoder **G**uardian **RAG** (**CEG-RAG**), a defense framework that leverages the internal activations of a cross-encoder reranker to detect and mitigate knowledge poisoning in RAG pipelines. CEG-RAG uses multi-instance learning (MIL) to jointly (i) detect whether the retrieved context is poisoned and (ii) localize suspicious chunks. Upon detection, it repairs the context by filtering and replacing high-risk chunks prior to answer generation while preserving a fixed context budget. Across three open-domain QA benchmarks—MS MARCO, Natural Questions (NQ), and HotpotQA—under a poisoning attack, CEG-RAG achieves high detection and localization performance (TPR $> 85\%$ and $> 88.4\%$, respectively, at very low FPR), reduces the attack success rate (ASR) by an average of 88.74%, and recovers correct answers. Compared to recent baseline defenses, CEG-RAG consistently provides stronger protection, and a reranker sensitivity study demonstrates its robustness across different reranker configurations. These results position cross-encoder reranker activations as a practical foundation for securing RAG against knowledge poisoning. The code and data are available at https://github.com/CyberScienceLab/CEG-RAG.

**Keywords:** LLM Security, Retrieval-Augmented Generation, Knowledge Poisoning Attacks, Cross-Encoder Reranker

## 1. Introduction

Large language models (LLMs) generate fluent text, but their parametric knowledge can be incomplete or outdated, leading to unreliable answers. Retrieval-Augmented Generation (RAG) addresses this by grounding generation in an external corpus: for each query, it retrieves relevant passages and conditions the LLM on the resulting context [1, 2]. This retrieval grounding improves factuality, enables rapid knowledge updates without retraining, and mitigates hallucinations in knowledge-intensive tasks such as open-domain question answering and fact verification [3, 4].

Despite the promise of RAG systems, they remain vulnerable to adversarial manipulation that can degrade the quality of generated responses [5]. The underlying corpus is particularly susceptible to poisoning attacks, whereby an adversary injects crafted documents that are subsequently retrieved as evidence, steering the LLM toward incorrect or attacker-controlled outputs.[5, 6].

Several lines of defense have been proposed to mitigate corpus poisoning in RAG, including retrieval-time heuristics [7, 8] (e.g., score filtering and agreement/consensus checks) [9, 10], provenance- and credibility-based vetting of the corpus, and auxiliary detectors—ranging from lightweight classifiers to LLM-based judges—that assess whether retrieved passages are malicious or untrustworthy prior to generation [11]. While effective in some settings, these approaches commonly introduce additional latency and cost. In contrast, we leverage a signal that is already present in many high-performing RAG pipelines: the cross-encoder

* fzarrink@uoguelph.ca

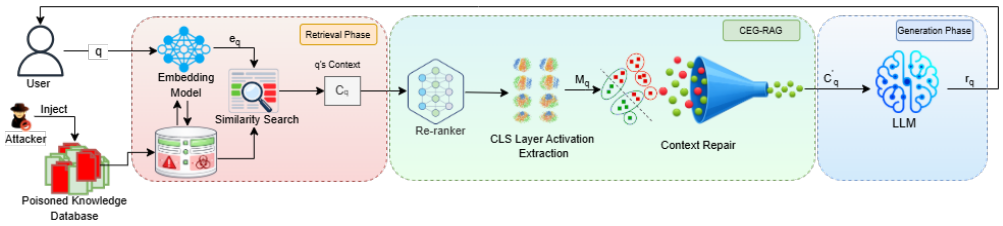


*Figure 1.* Overview of CEG-RAG.

reranker, which is widely adopted because it captures fine-grained query–passage interactions beyond embedding similarity and substantially improves ranking quality [12–14]. Rather than treating the reranker as a black box that produces only relevance scores, we draw inspiration from RevPRAG [15], which leverages internal LLM activations, and instead use the reranker's internal representations as a security signal to detect poisoning and localize malicious chunks before generation. This enables efficient detection without incurring additional model calls or altering the RAG pipeline.

In this paper, we introduce CEG-RAG (Cross-Encoder Guardian RAG), a defense framework that leverages *cross-encoder reranker activations* to secure RAG against corpus poisoning. CEG-RAG formulates context security as a *multi-instance learning* (MIL) problem [16], enabling the model to jointly (i) detect whether a retrieved context contains poisoning and (ii) localize suspicious chunks within the retrieved set. When poisoning is detected, CEG-RAG performs *context repair* prior to answer generation by filtering and replacing high-risk chunks while maintaining a fixed context budget.

Our main contributions are:

- We propose CEG-RAG, an activation-based defense framework that leverages internal cross-encoder reranker signals to protect RAG systems against knowledge poisoning.
- We introduce an MIL formulation that jointly supports *context-level detection* and *chunk-level localization* of poisoning within retrieved contexts.
- We present a practical *context repair* mechanism that filters and replaces high-risk chunks while preserving a fixed context budget for answer generation.
- We demonstrate strong effectiveness across three question answering datasets under a poisoning attack, with consistent improvements over recent baselines.

## 2. Methodology

### 2.1. Overview of CEG-RAG

We propose CEG-RAG, a defense framework for mitigating knowledge-poisoning attacks in RAG systems. The central observation is that a poisoning attack can influence the final answer only when poisoned chunks are both retrieved and ranked highly enough to be included in the generator's context. Accordingly, CEG-RAG uses the cross-encoder reranker as a sensor. For each query–passage pair in the reranked list, it extracts a final-layer representation from the reranker and analyzes these representations using a multiple-instance learning (MIL) classifier. In this formulation, the retrieved context is treated as a *bag*, each chunk as an *instance*, and supervision is provided only at the bag level, while chunk-level poisoning labels remain unobserved [16–18]. When the classifier identifies a query context as suspicious, CEG-RAG repairs it by removing high-risk passages and replacing them with lower-risk alternatives.

Given a query $q$, a dense retriever returns the top-$N$ candidate chunks, $C_q = \{c_1, \ldots, c_N\}$, which are then scored by a cross-encoder reranker. The reranker assigns a relevance score to each pair $(q, c_i)$ and sorts the candidates in descending order, $\tilde{C}_q = \{\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_N\}$, where $\tilde{c}_1$ is the highest-ranked chunk. For each reranked pair $(q, \tilde{c}_i)$, we extract the reranker's final-layer embedding:

$$\mathbf{h}_{q,i} \in \mathbb{R}^D, \qquad i = 1, \ldots, N. \tag{2.1}$$

Stacking these embeddings in reranked order forms a query-specific representation matrix:

$$\mathbf{M}_q = \begin{bmatrix} \mathbf{h}_{q,1}^\top \\ \mathbf{h}_{q,2}^\top \\ \vdots \\ \mathbf{h}_{q,N}^\top \end{bmatrix} \in \mathbb{R}^{N \times D}. \tag{2.2}$$

To detect poisoning, CEG-RAG applies a multiple-instance learning (MIL) model over the embeddings in $\mathbf{M}_q$. The detector outputs (i) a *context-level* poisoning decision and (ii) chunk-level poison scores used for localization and repair. When the context is flagged, CEG-RAG repairs the evidence by filtering high-risk passages and refilling the generator's fixed context budget using safe candidates. The generator is finally prompted on the query and repaired context to produce the final answer.

### 2.2. **Poisoning Detection and Chunk Localization**

Given $\mathbf{M}_q \in \mathbb{R}^{N \times D}$ (Eq. 2.2), we apply an MIL-based detector to produce a context-level score and chunk-level scores. The detector operates on fixed-size bags of $K$ reranked chunks: we partition the top-$N$ reranked list into $B = \lceil N/K \rceil$ blocks and process each block independently. For block $b$, let $\mathbf{M}_{q,b} \in \mathbb{R}^{K \times D}$ denote the corresponding submatrix of $\mathbf{M}_q$ in reranker order. We first project the embeddings to a $d$-dimensional space using a shared linear projection $\mathbf{W} \in \mathbb{R}^{D \times d}$, and then apply a Set Transformer [19] encoder to model interactions among the $K$ chunks:

$$\mathbf{U}_{q,b} = \mathrm{Enc}\big(\mathbf{M}_{q,b}\mathbf{W}\big) \in \mathbb{R}^{K \times d}, \tag{2.3}$$

where each row $\mathbf{U}_{q,b,i}$ is the contextualized representation of a chunk after attending to the other chunks in the same block.

To score a block, we use attention-based MIL to form a block embedding and predict a block-level poisoning probability with a learnable head $h(\cdot)$:

$$\mathbf{b}_{q,b} = \sum_{i=1}^{K} \alpha_{q,b,i}\, \mathbf{U}_{q,b,i}, \tag{2.4}$$

$$\hat{y}_{q,b} = \sigma\big(h(\mathbf{b}_{q,b})\big), \tag{2.5}$$

where $\sigma(\cdot)$ is the sigmoid and $\alpha_{q,b,i}$ are learned attention weights. We aggregate block-level predictions into a single context-level score using the "any-block" rule:

$$\hat{y}_q = \max_b \hat{y}_{q,b}. \tag{2.6}$$

For chunk-level localization, we score each encoded chunk with a shared head and associate these scores with their positions in the top-$N$ reranked list to obtain $\{s_{q,i}\}_{i=1}^{N}$. Larger $s_{q,i}$ indicates higher probability that chunk $\tilde{c}_i$ is poisoned.

$$s_{q,b,i} = \sigma\big(g(\mathbf{U}_{q,b,i})\big), \qquad i = 1, \ldots, K, \tag{2.7}$$

The model is trained under weak supervision: each query is labeled only at the context level (benign vs. poisoned retrieval), and no chunk-level labels are used during training. Training minimizes a binary cross-entropy loss over block-level predictions:

$$\mathcal{L}_{\text{det}} = -\sum_{q}\sum_{b=1}^{B}\Big[y_q \log(\hat{y}_{q,b}) + (1-y_q)\log(1-\hat{y}_{q,b})\Big], \tag{2.8}$$

where $y_q$ is the context-level label shared across all blocks of query $q$.

### 2.3. **Context Repair**

CEG-RAG performs chunk filtering only when poisoning is detected. The detector provides a context-level poisoning probability $\hat{y}_q$ and chunk-level poison scores $\{s_{q,i}\}_{i=1}^{N}$. Repair uses these signals to (i) remove high-risk chunks and (ii) replace them with lower-ranked candidates predicted to be safe.

We apply repair conditionally using a detection threshold $\tau_{\text{det}}$. If $\hat{y}_q < \tau_{\text{det}}$, we pass the top-$k$ reranked chunks unchanged. Otherwise, we filter candidates whose suspicion exceeds a localization threshold $\tau_{\text{loc}}$ and construct a repaired context by selecting safe chunks in reranker order:

$$\mathcal{C}'_q = \begin{cases} \langle \tilde{c}_1, \ldots, \tilde{c}_k \rangle, & \hat{y}_q < \tau_{\text{det}}, \\ \text{Top}_k\big(\langle \tilde{c}_i \mid s_{q,i} < \tau_{\text{loc}} \rangle\big), & \hat{y}_q \geq \tau_{\text{det}}. \end{cases} \tag{2.9}$$

where $\text{Top}_k(\cdot)$ returns the first $k$ elements according to the reranker ordering in $\tilde{\mathcal{C}}_q$.

Although chunk-level ground-truth indicators $z_{q,i} \in 0, 1$ may be available for candidate chunks $\tilde{c}_i$, CEG-RAG is designed to operate under weak supervision: during training, the MIL detector uses only the context-level label $y_q$, and does not rely on chunk annotations. Chunk-level labels are used only to evaluate localization accuracy. At inference time, the model assigns a poison score to each chunk, which we convert into binary localization decisions by thresholding:

$$\hat{z}_{q,i} = \mathbb{I}[s_{q,i} \geq \tau_{\text{loc}}], \tag{2.10}$$

Finally, the generator is conditioned on the repaired context $\mathcal{C}'_q$ together with the query $q$, and a fixed prompting template to produce the final response $r_q$. By filtering high-risk passages and refilling the fixed context budget with lower-ranked candidates predicted to be safe, CEG-RAG reduces attacker influence while preserving benign evidence.

## 3. **Experiments**

In this study, we investigate four primary research questions:
**RQ1:** To what extent does the proposed method detect poisoning in the retrieved RAG context, and how precisely does it localize the implicated chunk(s)?
**RQ2:** To what extent does the proposed context-repair procedure recover the gold answer after poisoning is detected?
**RQ3:** How sensitive are detection and repair performance to the choice of cross-encoder re-ranker?
**RQ4:** How does the proposed method compare against state-of-the-art RAG-specific defenses under poisoning attacks?

### 3.1. **Experimental setup**

**Datasets and poisoning setting.** We conduct experiments on three widely used open-domain question answering benchmarks: Natural Questions (NQ) [20], HotpotQA [21], and MS-MARCO [22]. For each dataset, we randomly sample 4,000 queries along with their associated evidence passages to construct the RAG knowledge source. We then poison a subset of this corpus by selecting 2,000 target queries and injecting adversarial passages

into the knowledge database. Specifically, we adopt the PoisonedRAG attack [5]. For each targeted query, we inject five malicious passages, all crafted to support the same attacker-chosen desired answer. This results in 10,000 injected poisoned passages per dataset.

**RAG settings.** We adopt Contriever [4] for dense retrieval and `BAAI/bge-reranker-large` [23] as the cross-encoder reranker. For generation, we use `Meta-Llama-3-8B-Instruct` and provide it with the top three reranked chunks ($K = 3$).

**Reranker models.** To test whether our approach depends on a particular reranker, we repeat the experiments using three different cross-encoder rerankers: `BAAI/bge-reranker-large` (BGE-reranker) [23], `mGTE-reranker` [24], and `cross-encoder/ms-marco-MiniLM-L6-v2` (MS-reranker) [25].

**Defense baselines.** We compare CEG-RAG against four representative RAG-specific defenses designed to mitigate poisoning attacks: (1) GMTP [8], which flags and removes adversarial retrieved passages via token masking and predictability scoring; (2) RobustRAG [9], which aggregates per-chunk answers via keyword consensus before final generation; (3) TrustRAG [7], which clusters retrieved passages to filter suspicious evidence before answering; and (4) RAGuard [26], a two-stage defense that (i) adversarially trains the retriever to down-rank poisoned passages and (ii) applies an inference-time leave-one-out filter that removes each retrieved passage in turn and flags passages whose removal flips the model from an incorrect answer to a correct one.

**Metrics.** For detection evaluation, following [15], we report *True Positive Rate (TPR)* and *False Positive Rate (FPR)* at two granularities: the *context level*, which assesses whether the retrieved context for a query is flagged as poisoned, and the *chunk level*, which assesses whether individual retrieved chunks are identified as poisoned. At the chunk level, TPR is the proportion of poisoned chunks that are correctly flagged, and FPR is the proportion of benign chunks that are incorrectly flagged. The same definitions apply at the context level, where TPR and FPR are computed over retrieved contexts rather than individual chunks. We report context-level metrics to quantify the model's ability to detect, for each query, whether the *retrieved evidence set* is contaminated by poisoned content (i.e., whether the query-specific context drawn from the knowledge source is adversarial). Our goal is to maximize TPR while maintaining a low FPR. For repair evaluation, we report the *Attack Success Rate (ASR)*, defined as the fraction of questions for which the final RAG output matches the attacker-specified target. To assess the impact of repair, in addition to ASR reduction, we also report *Accuracy (ACC)*, defined as the fraction of questions for which the final output matches the gold answer.

### 3.2. Results

#### 3.2.1. Detection and Localization Performance (RQ1)

We first evaluate the effectiveness of CEG-RAG in detecting poisoned retrieval context and localizing the responsible chunks, addressing RQ1. Table 1 reports detection performance in terms of TPR and FPR at both the *context level* and the finer-grained *chunk level* across three benchmark datasets.

CEG-RAG achieves consistently strong detection performance, with high TPR values across all datasets. On MS-MARCO, the method attains a context-level TPR of 0.9833, indicating that nearly all poisoned contexts are correctly flagged. Performance remains similarly robust at the chunk level (TPR = 0.9764), demonstrating the model's ability not only to detect poisoning but also to localize the implicated chunk with high accuracy. The corresponding FPR values are moderate (0.0667 at context level and 0.1125 at chunk level), suggesting a limited number of false alarms under clean retrieval. On Natural Questions (NQ), CEG-RAG maintains strong detection reliability, with context-level and chunk-level TPRs of 0.9507 and 0.9404, respectively. Notably, FPR remains low in both settings (0.0359

| Dataset | Context-level Detection | | Chunk-level Localization | | Repair | |
|---|---|---|---|---|---|---|
| | **TPR** | **FPR** | **TPR** | **FPR** | **ASR(↓%)** | **ACC(%)** |
| MS-MARCO | 0.9833 | 0.0667 | 0.9764 | 0.1125 | 11.20 (88.8%) | 50.00 |
| NQ | 0.9507 | 0.0359 | 0.9404 | 0.0178 | 6.30 (93.7%) | 41.06 |
| HotpotQA | 0.8500 | 0.0000 | 0.8840 | 0.0457 | 16.28 (83.72%) | 13.95 |

*Table 1.* Detection, localization, and repair performance across datasets.

and 0.0178), highlighting that the proposed detector preserves high precision and is less likely to incorrectly flag benign evidence. Detection becomes more challenging on HotpotQA; nevertheless, CEG-RAG achieves a context-level TPR of 0.8500 and a chunk-level TPR of 0.8840. The slightly higher chunk-level FPR (0.0457) reflects the increased difficulty of fine-grained localization in more complex retrieval structures.

Overall, the results demonstrate that CEG-RAG can effectively detect poisoned retrieved context and accurately localize adversarial chunks across diverse datasets, supporting its effectiveness for robust RAG deployment.

We further examine whether CEG-RAG is sensitive to the number of reranked chunks forwarded to the generator, i.e., the top-$K$ evidence budget. On NQ dataset, we vary $K$ over a wide range ($K = 2$ to 20) and observe only minor performance changes: post-defense accuracy remains essentially stable at approximately 39.6%–41.1%, while ASR stays consistently low at about 6.0%–8.1%. The small spread across both metrics indicates that CEG-RAG is robust to reasonable choices of $K$, and does not rely on a narrowly tuned context size to maintain effective poisoning detection and mitigation.

#### 3.2.2. **Context Repair Performance (RQ2)**

Table 1 reports repair performance using two complementary measures: ASR, the proportion of cases that still produce the attacker-specified target after repair, and ACC, the proportion of cases for which the post-repair output matches the gold answer. Across datasets, the proposed repair procedure substantially reduces attacker control. On MS-MARCO, ASR drops to 11.20% (an 88.8% reduction), while ACC reaches 50.00%, indicating that the gold answer is recovered for half of the evaluated instances. On NQ, repair is most effective, lowering ASR to 6.30% (93.7% reduction) and achieving ACC of 41.06%, suggesting that many repaired outputs are not only steered away from the attacker target but also corrected to the gold answer. HotpotQA remains the most challenging setting. Although repair reduces ASR to 16.28% (83.7% reduction), ACC is comparatively lower (13.95%).

Overall, these results show that context repair is effective in two complementary ways: it substantially reduces the rate at which the system outputs the attacker-specified target (lower ASR), while also recovering the gold answer for a meaningful fraction of cases (higher ACC). Importantly, the simultaneous decrease in ASR and increase in ACC indicates that the procedure often reconstructs sufficiently reliable evidence to support correct generation, rather than merely disrupting the attack.

#### 3.2.3. **Robustness Across Cross-Encoder Re-Rankers (RQ3)**

We examine how sensitive CEG-RAG is to the underlying cross-encoder reranker, addressing RQ3. Table 2 reports chunk-level detection performance (TPR/FPR) and downstream repair outcomes (ASR and ACC) under three reranker models: BGE, mGTE, and MS-reranker. Detection remains consistently strong across rerankers. In MS-MARCO and NQ, both mGTE and MS-reranker achieve near-perfect chunk-level TPR ($\approx$0.99) with low FPR, indicating that poisoning localization is largely reranker-agnostic in these settings.

| Dataset | Reranker | Detection | | Repair | |
|---|---|---|---|---|---|
| | | Chunk-level | | | |
| | | TPR | FPR | ASR(%) | ACC(%) |
| MS-MARCO | BGE-reranker [23] | 0.9764 | 0.1125 | 11.20 | 50.00 |
| | mGTE-reranker [24] | 0.9999 | 0.0760 | 12.80 | 21.20 |
| | MS-reranker [25] | 0.9999 | 0.0830 | 8.00 | 49.60 |
| NQ | BGE-reranker [23] | 0.9404 | 0.0178 | 6.30 | 41.06 |
| | mGTE-reranker [24] | 0.9970 | 0.0060 | 1.80 | 23.78 |
| | MS-reranker [25] | 0.9999 | 0.0060 | 4.30 | 33.33 |
| HotpotQA | BGE-reranker [23] | 0.8840 | 0.0457 | 16.28 | 13.95 |
| | mGTE-reranker [24] | 0.9999 | 0.0590 | 5.80 | 23.25 |
| | MS-reranker [25] | 0.9670 | 0.0460 | 19.80 | 53.49 |

*Table 2.* Robustness across different reranker models.

HotpotQA exhibits slightly more variation, but detection performance remains high, with TPR ranging from 0.8840 (BGE) to 0.9999 (mGTE).

While all rerankers reduce ASR substantially, the degree of gold-answer recovery varies. For instance, on NQ, mGTE yields the lowest ASR (1.80%) but lower ACC (23.78%), whereas MS-reranker achieves a higher ACC (33.33%) with slightly higher ASR (4.30%). A similar trade-off is observed on HotpotQA, where mGTE minimizes ASR (5.80%), while MS-reranker attains the highest Accuracy (53.49%), suggesting stronger answer recovery despite residual vulnerability.

Collectively, these results indicate that CEG-RAG's detection component generalizes well across reranker architectures, whereas repair effectiveness is more sensitive to reranker-specific ranking behavior, leading to different trade-offs between attack suppression and gold-answer recovery.

#### 3.2.4. **Comparison with State-of-the-Art Defenses (RQ4)**

We compare CEG-RAG with four representative RAG-specific defenses against poisoning attacks: `GMTP`, `RobustRAG`, `TrustRAG`, and `RAGuard`. Table 3 reports repair performance using ACC and ASR.

On MS-MARCO, CEG-RAG achieves the strongest overall performance, attaining the highest ACC (50.00%) while also yielding the lowest ASR (11.20%). Among baselines, `TrustRAG` provides the closest ACC (44.00%), whereas `GMTP` achieves the second-lowest ASR (16.80%) but with substantially reduced ACC (3.20%). `RAGuard` is ineffective in this setting, exhibiting high residual ASR (87.60%) and low ACC (4.40%). On NQ, CEG-RAG again produces the lowest ASR (6.30%), improving upon the strongest baseline `TrustRAG` (16.26%). While `TrustRAG` attains the highest ACC (59.35%), CEG-RAG maintains a competitive ACC (41.06%) while offering substantially stronger resistance to targeted manipulation. Other baselines exhibit higher ASR (38.62–76.88%) with moderate or lower ACC.

On HotpotQA, CEG-RAG achieves the lowest ASR (16.28%), improving over `TrustRAG` (23.60%) and `GMTP` (32.56%). However, ACC is lower for CEG-RAG (13.95%) than for `TrustRAG` (46.40%) and `RobustRAG` (36.05%), indicating a stronger trade-off between suppressing the attacker target and fully recovering the gold answer in this dataset.

Overall, the results show that CEG-RAG provides the most consistent reduction in attack success across datasets, outperforming prior defenses in ASR on all three benchmarks, while remaining competitive in gold-answer recovery on MS-MARCO and NQ.

| Dataset | Defense method | ACC(%) | ASR(%) |
|---|---|---|---|
| MS-MARCO | GMTP [8] | 3.20 | 16.80 |
| | RobustRAG [9] | 25.60 | 57.20 |
| | TrustRAG [7] | 44.00 | 25.20 |
| | RAGuard [26] | 4.40 | 87.60 |
| | CEG-RAG (ours) | **50.00** | **11.20** |
| NQ | GMTP [8] | 12.40 | 38.62 |
| | RobustRAG [9] | 33.94 | 46.95 |
| | TrustRAG [7] | **59.35** | 16.26 |
| | RAGuard [26] | 9.76 | 76.88 |
| | CEG-RAG (ours) | 41.06 | **6.30** |
| HotpotQA | GMTP [8] | 16.28 | 32.56 |
| | RobustRAG [9] | 36.05 | 56.98 |
| | TrustRAG [7] | **46.40** | 23.60 |
| | RAGuard [26] | 12.79 | 83.72 |
| | CEG-RAG (ours) | 13.95 | **16.28** |

*Table 3.* Comparison with RAG-specific defense baselines.

## 4. Limitations

Despite its effectiveness, CEG-RAG has several limitations. (1) Our evaluation focuses on a strong but single RAG poisoning attack, PoisonedRAG [5]. Although the proposed framework is attack-agnostic in principle, further study is needed to assess robustness under alternative poisoning strategies, which we leave for future work. (2) The proposed defense assumes access to internal signals from the cross-encoder reranker (i.e. activation patterns). This assumption holds in open or locally deployed reranking pipelines, but may not hold when reranking is performed through black-box or proprietary services, which would restrict the direct use of the proposed signals. (3) Finally, our experiments are conducted under a fixed RAG configuration (retriever and generator settings). Additional evaluation across different retrieval and generator models would further clarify the generality of the observed gains.

## 5. Related Work

Although LLMs have achieved remarkable success, prior research has shown that they remain vulnerable to a range of security threats and adversarial manipulations [27–29]. In this context, RAG grounds LLM outputs in external corpora and has become a standard paradigm for knowledge-intensive and domain-specific tasks. By conditioning generation on retrieved evidence, RAG improves factuality and coverage, but also introduces new security risks by exposing the model to potentially untrusted retrieved content.

**RAG poisoning attacks.** A growing body of work shows that adversaries can poison the knowledge base of RAG so that injected passages are retrieved and steer generation toward attacker-specified outputs [5, 15, 30–33]. In the common threat model where attackers can inject content into the knowledge base but cannot directly modify the LLM, poisoned passages are crafted to be both highly retrievable for target queries and effective at inducing the desired response once included in context [5, 30]. PoisonedRAG formalizes this as an optimization problem over retriever relevance, enabling targeted manipulation through adversarial evidence shaping [5]. Related work explores retrieval-focused perturbations such as similarity-boosting edits and query-aligned adversarial passages [33, 34], often leveraging gradient-guided token replacement methods (e.g., HotFlip) [35–37]. More recent studies extend poisoning beyond retrieval-only manipulation, including joint retriever–generator optimization and reasoning-time attacks that corrupt multi-step inference [38, 39]. Notably,

poisoning has been shown to remain effective even at scale, with only a small number of injected documents sometimes sufficient to induce compromised behavior [40].

**Defenses against poisoning attacks.** Defensive efforts have proposed a range of mitigation strategies, including heuristic filtering based on perplexity or embedding statistics [5, 33], as well as more targeted approaches that aim to detect artifacts of adversarial passage construction. For instance, GMTP masks influential tokens and uses predictability-based signals to flag suspicious evidence [8]. Other defenses strengthen retrieval robustness or reduce the influence of individual passages through retriever hardening, aggregation-based generation, or evidence clustering and verification [7, 9, 26]. Together, these works highlight the importance of developing principled defenses that can reliably identify and mitigate poisoned evidence in retrieval-augmented systems.

## 6. Conclusion

We introduced CEG-RAG, a defense framework against knowledge poisoning attacks in retrieval-augmented generation. By leveraging internal activation signals from a cross-encoder reranker, CEG-RAG detects poisoned retrieved context, localizes suspicious chunks, and mitigates attacks through targeted context repair. Experiments across multiple open-domain QA benchmarks demonstrate that CEG-RAG achieves high poisoning detection accuracy with low false positives and substantially reduces attack success rates while enabling recovery of gold answers. These results highlight the effectiveness of reranker-based activation signatures as a promising direction for securing RAG pipelines against evidence poisoning.

## GenAI Usage Disclosure

OpenAI's ChatGPT was used to refine sentence clarity and grammatical correctness during manuscript preparation.

## Acknowledgments

This work was supported in part by the NSERC-CSE Research Community Grants (ALLRP 598786-24), NSERC Canada Research Chair Grant (CRC-2024-00017), and the National Cybersecurity Consortium (2025-1601) projects. Researchers funded through the NSERC-CSE Research Communities Grants do not represent the Communications Security Establishment Canada or the Government of Canada. Any research, opinions or positions they produce as part of this initiative do not represent the official views of the Government of Canada.

## References

[1] V. Karpukhin, B. Oguz, S. Min, P. S. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. "Dense Passage Retrieval for Open-Domain Question Answering." In: *EMNLP (1)*. 2020, pp. 6769–6781.

[2] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks". In: *Advances in neural information processing systems* 33 (2020), pp. 9459–9474.

[3] A. Lazaridou, E. Gribovskaya, W. Stokowiec, and N. Grigorev. "Internet-augmented language models through few-shot prompting for open-domain question answering". In: *arXiv preprint arXiv:2203.05115* (2022).

[4] G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave. "Unsupervised dense information retrieval with contrastive learning". In: *arXiv preprint arXiv:2112.09118* (2021).

[5] W. Zou, R. Geng, B. Wang, and J. Jia. "{PoisonedRAG}: Knowledge corruption attacks to {Retrieval-Augmented} generation of large language models". In: *34th USENIX Security Symposium (USENIX Security 25)*. 2025, pp. 3827–3844.

[6] Z. Shen, B. Imana, T. Wu, C. Xiang, P. Mittal, and A. Korolova. "ReliabilityRAG: Effective and Provably Robust Defense for RAG-based Web-Search". In: *arXiv preprint arXiv:2509.23519* (2025).

[7] H. Zhou, Z Zhan, Z Li, H Haddadi, and E Yilmaz. "Trustrag: Enhancing robustness and trustworthiness in rag". In: (2025).

[8] S. Kim, J. Kim, Y. Jeon, and G. Lee. "Safeguarding RAG Pipelines with GMTP: A Gradient-based Masked Token Probability Method for Poisoned Document Detection". In: *Findings of the Association for Computational Linguistics: ACL 2025*. 2025, pp. 24597–24614.

[9] C. Xiang, T. Wu, Z. Zhong, D. Wagner, D. Chen, and P. Mittal. "Certifiably robust rag against retrieval corruption". In: *arXiv preprint arXiv:2405.15556* (2024).

[10] J. Hwang, J. Park, H. Park, D. Kim, S. Park, and J. Ok. "Retrieval-augmented generation with estimation of source reliability". In: *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. 2025, pp. 34267–34291.

[11] X. Tan, H. Luan, M. Luo, X. Sun, P. Chen, and J. Dai. "Knowledge database or poison base? detecting rag poisoning attack through llm activations". In: *arXiv e-prints* (2024), arXiv–2411.

[12] G. d. S. P. Moreira, R. Ak, B. Schifferer, M. Xu, R. Osmulski, and E. Oldridge. "Enhancing Q&A Text Retrieval with Ranking Models: Benchmarking, fine-tuning and deploying Rerankers for RAG". In: *arXiv preprint arXiv:2409.07691* (2024).

[13] Q. Liu, G. Guo, J. Mao, Z. Dou, J.-R. Wen, H. Jiang, X. Zhang, and Z. Cao. "An analysis on matching mechanisms and token pruning for late-interaction models". In: *ACM Transactions on Information Systems* 42.5 (2024), pp. 1–28.

[14] Y. Mao, P. He, X. Liu, Y. Shen, J. Gao, J. Han, and W. Chen. "RIDER: Reader-Guided Passage Reranking for Open-Domain Question Answering". In: *arXiv preprint arXiv:2101.00294* (2021).

[15] X. Tan, H. Luan, M. Luo, X. Sun, P. Chen, and J. Dai. "RevPRAG: Revealing Poisoning Attacks in Retrieval-Augmented Generation through LLM Activation Analysis". In: *arXiv preprint arXiv:2411.18948* (2024).

[16] M. Ilse, J. Tomczak, and M. Welling. "Attention-based deep multiple instance learning". In: *International conference on machine learning*. PMLR. 2018, pp. 2127–2136.

[17] A. Sathe and J. Park. "Automatic fact-checking with document-level annotations using bert and multiple instance learning". In: *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*. 2021, pp. 101–107.

[18] Y. Hu, M. Li, and N. Yu. "Multiple-instance ranking: Learning to rank images for image retrieval". In: *2008 IEEE Conference on computer vision and pattern recognition*. IEEE. 2008, pp. 1–8.

[19] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh. "Set transformer: A framework for attention-based permutation-invariant neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 3744–3753.

[20] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, et al. "Natural questions: a benchmark for question answering research". In: *Transactions of the Association for Computational Linguistics* 7 (2019), pp. 453–466.

[21] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, and C. D. Manning. "HotpotQA: A dataset for diverse, explainable multi-hop question answering". In: *Proceedings of the 2018 conference on empirical methods in natural language processing*. 2018, pp. 2369–2380.

[22] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, et al. "Ms marco: A human generated machine reading comprehension dataset". In: *arXiv preprint arXiv:1611.09268* (2016).

[23] S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, D. Lian, and J.-Y. Nie. "C-pack: Packed resources for general chinese embeddings". In: *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*. 2024, pp. 641–649.

[24] X. Zhang, Y. Zhang, D. Long, W. Xie, Z. Dai, J. Tang, H. Lin, B. Yang, P. Xie, F. Huang, et al. "mgte: Generalized long-context text representation and reranking models for multilingual text retrieval". In: *arXiv preprint arXiv:2407.19669* (2024).

[25] R. Nogueira and K. Cho. "Passage Re-ranking with BERT". In: *arXiv preprint arXiv:1901.04085* (2019).

[26] T. Kolhe, P. Kumar, T. Nielson, S. Zala, V. Li, M. Saxon, S. Wu, and K. Zhu. "RAGuard: A Layered Defense Framework for Retrieval-Augmented Generation Systems Against Data Poisoning". In: *Socially Responsible and Trustworthy Foundation Models at NeurIPS 2025*.

[27] E. Rabieinejad, F. Zarrinkalam, and A. Dehghantanha. "Beyond the prompt: Log-based threat detection and attribution for multi-Agent LLMs". In: *Information Processing & Management* 63.6 (2026), p. 104768.

[28] H. Alizadeh Noughabi, J. Serbanescu, F. Zarrinkalam, and A. Dehghantanha. "Uncovering the Persuasive Fingerprint of LLMs in Jailbreaking Attacks". In: *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*. 2025, pp. 4608–4612.

[29] M. Sorkhpour, A. Yazdinejad, and A. Dehghantanha. "RedHit: Adaptive red-teaming of large language models via search, reasoning, and preference optimization". In: *Proceedings of the The First Workshop on LLM Security (LLMSEC)*. 2025, pp. 7–16.

[30] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz. "Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection". In: *Proceedings of the 16th ACM workshop on artificial intelligence and security*. 2023, pp. 79–90.

[31] J. Xue, M. Zheng, Y. Hu, F. Liu, X. Chen, and Q. Lou. "Badrag: Identifying vulnerabilities in retrieval augmented generation of large language models". In: *arXiv preprint arXiv:2406.00083* (2024).

[32] P. Cheng, Y. Ding, T. Ju, Z. Wu, W. Du, P. Yi, Z. Zhang, and G. Liu. "Trojanrag: Retrieval-augmented generation can be backdoor driver in large language models". In: *arXiv preprint arXiv:2405.13401* (2024).

[33] Z. Zhong, Z. Huang, A. Wettig, and D. Chen. "Poisoning retrieval corpora by injecting adversarial passages". In: *arXiv preprint arXiv:2310.19156* (2023).

[34] S. Cho, S. Jeong, J. Seo, T. Hwang, and J. C. Park. "Typos that Broke the RAG's Back: Genetic Attack on RAG Pipeline by Simulating Documents in the Wild via Low-level Perturbations". In: *arXiv preprint arXiv:2404.13948* (2024).

[35] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou. "Hotflip: White-box adversarial examples for text classification". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 2018, pp. 31–36.

[36] H. Chaudhari, G. Severi, J. Abascal, M. Jagielski, C. A. Choquette-Choo, M. Nasr, C. Nita-Rotaru, and A. Oprea. "Phantom: General trigger attacks on retrieval augmented language generation". In: *arXiv preprint arXiv:2405.20485* (2024).

[37] Q. Zhang, B. Zeng, C. Zhou, G. Go, H. Shi, and Y. Jiang. "Human-imperceptible retrieval poisoning attacks in LLM-powered applications". In: *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*. 2024, pp. 502–506.

[38] H. Wang, R. Zhang, J. Wang, M. Li, Y. Huang, D. Wang, and Q. Wang. "Joint-GCG: Unified Gradient-Based Poisoning Attacks on Retrieval-Augmented Generation Systems". In: *arXiv preprint arXiv:2506.06151* (2025).

[39] H. Song, Y.-a. Liu, R. Zhang, J. Guo, and Y. Fan. "Chain-of-Thought Poisoning Attacks against R1-based Retrieval-Augmented Generation Systems". In: *arXiv preprint arXiv:2505.16367* (2025).

[40] A. Souly, J. Rando, E. Chapman, X. Davies, B. Hasircioglu, E. Shereen, C. Mougan, V. Mavroudis, E. Jones, C. Hicks, et al. "Poisoning attacks on llms require a near-constant number of poison samples". In: *arXiv preprint arXiv:2510.07192* (2025).