RIPRAG: Hack a Black-box Retrieval-Augmented Generation Question-Answering System with Reinforcement Learning

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) systems based on Large Language Models (LLMs) have become a core technology for tasks such as question-answering (QA) and content generation. However, by injecting poisoned documents into the database of RAG systems, attackers can manipulate LLMs to generate text that aligns with their intended preferences. Existing research has primarily focused on white-box attacks against simplified RAG architectures. In this paper, we investigate a more complex and realistic scenario: the attacker lacks knowledge of the RAG system's internal composition and implementation details, and the RAG system comprises components beyond a mere retriever. Specifically, we propose the RIPRAG attack framework, an end-to-end attack pipeline that treats the target RAG system as a black box, where the only information accessible to the attacker is whether the poisoning succeeds. Our method leverages Reinforcement Learning (RL) to optimize the generation model for poisoned documents, ensuring that the generated poisoned document aligns with the target RAG system's preferences. Experimental results demonstrate that this method can effectively execute poisoning attacks against most complex RAG systems, achieving an attack success rate (ASR) improvement of up to 0.72 compared to baseline methods. This highlights prevalent deficiencies in current defensive methods and provides critical insights for LLM security research.

1 Introduction

LLMs have revolutionized the field of artificial intelligence, demonstrating remarkable capabilities in understanding and generating human-like text. However, their inherent limitation lies in the static nature of their parametric knowledge, which can become outdated or lack specificity for certain domains. RAG (Lewis et al., 2020) has been proposed to mitigate this issue by equipping LLMs with access to an external, updatable database. This paradigm enhances the factuality and relevance of generated responses, particularly in critical applications like question-answering and content generation, by dynamically retrieving and grounding responses in pertinent information.

Despite its advantages, the RAG framework introduces new vulnerabilities, primarily through its retrieval component. A significant threat is RAG poisoning (Zou et al., 2025), where attackers inject poisoned documents into the database to manipulate the LLM's outputs. This attack compromises the system's integrity, leading to the dissemination of misinformation or biased content. Such vulnerabilities are particularly concerning when RAG systems are applied to sensitive domains like healthcare, finance, or customer service, where accurate information is paramount. For instance, an attacker could poison a financial advisory system to promote a specific stock.

Existing research on RAG poisoning attacks can be broadly categorized into white-box and black-box methods. White-box attacks (Jiao et al., 2025; Hu et al., 2024; Chaudhari et al., 2024; Tan et al., 2024b; Zou et al., 2025) assume full knowledge of the RAG system's architecture, and utilize gradient information to optimize poisoned texts for higher retrieval probability. However, their critical defect is the unrealistic assumption of a naive RAG pipeline (e.g., a single embedding model and LLM). Modern RAG systems (Gao et al., 2024) often employ complex retrieval strategies such as hybrid search or GraphRAG (Edge et al., 2024), where gradient information is inaccessible, rendering white-box methods ineffective. Conversely, current black-box attacks are often superficial. For example, PoisonedRAG (Zou et al., 2025) merely inserts the target query into the poisoned text, failing to leverage interactive feedback from the system. Topic-FlipRAG (Gong et al., 2025) relies on a surrogate open-source retriever, whose perfor-

mance degrades significantly when it diverges from the target system's actual retriever. Furthermore, these methods perform poorly in scenarios with a low poisoning rate, where the number of poisoned documents is significantly lower than the number of retrieved documents.

To overcome these limitations, we propose RIPRAG, a novel black-box attack framework that treats the target RAG system as an opaque oracle. Our key insight is to leverage **R**L to optimize the generation of poisoned documents by utilizing **I**nteractive feedback from the black-box system, thereby achieving effective **P**oisoning. Specifically, RIPRAG interacts with the target system by injecting candidate documents and observing whether the attack is successful. This feedback, combined with a textual similarity reward, guides an RL agent to iteratively refine its poisoning strategy, effectively adapting to the unknown internal mechanics of the RAG system and maximizing the attack success rate even under challenging conditions.

The main contributions of this work are fourfold:

- We propose RIPRAG, the first framework to apply Reinforcement Learning to the problem of attacking RAG systems. We have successfully addressed the issue of poor performance in the current RAG poisoning methods under low poisoning rate scenarios.

- We introduce Reinforcement Learning from Black-box Feedback (RLBF), a training methodology that optimizes attack policies using only the success/failure signal from the target system.

- We design Batch Relative Policy Optimization (BRPO), a novel policy optimization algorithm that enhances training stability and efficiency in adversarial text generation.

- To the best of our knowledge, we are the first to evaluate attack methods on RAG systems equipped with advanced, targeted defense mechanisms, providing a more rigorous and realistic security assessment.

2 Related Works

2.1 Attacks on RAG System

Attack methods on RAG systems can be broadly divided into two categories, i.e., white-box attacking and black-box attacking. White-box attacking refers to methods that optimize their poisoned documents with the inner information of RAG systems, including using the gradient of the retriever (Zou et al., 2025; Hu et al., 2024; Chaudhari et al., 2024; Tan et al., 2024b; Wang et al., 2025) or using the score given by the retriever (Jiao et al., 2025) to maximize the probability of being chosen by the retriever. However, in most cases, the inner part of RAG systems is not visible. Moreover, for more advanced retrieval methods like GraphRAG (Edge et al., 2024) or Modular RAG (Gao et al., 2024), computing the gradient of the retriever is impractical because it is not a simple neural network model. Current research on black-box approaches is limited. PoisonedRAG (Zou et al., 2025) enhances the similarity between the poisoned document and the target query by directly inserting the target query itself into the poisoned document. TopicFlipRAG (Gong et al., 2025), on the other hand, leverages gradients from an open-source retriever to optimize the poisoned document, which is essentially still a white-box method. In contrast to prior work, our approach optimizes the poisoned document using only the outputs of the black-box system, making it a truly black-box attack method.

2.2 Defending on RAG System

To mitigate the risks of RAG systems being attacked, numerous defense methods have been proposed. These defense approaches can be broadly categorized into model-based methods and feature-based methods. Model-based methods detect poisoned tokens in retrieved texts by masking tokens that significantly contribute to the retriever's output (Kim et al., 2025) or by clustering the activation information of the LLM (Tan et al., 2024a). Feature-based methods identify poisoned texts by analyzing word frequency density (Edemacu et al., 2025), applying elbow detection to retrieval (Saxena et al., 2025), or clustering embedding vectors (Zhou et al., 2025).

2.3 Reinforcement Learning from Human Feedback

RL has become a cornerstone technique for aligning LLMs with human preferences. The foundational approach, RLHF, was introduced by OpenAI in InstructGPT (Ouyang et al., 2022). This framework has been widely adopted in models like GPT-4 (Achiam et al., 2023), Qwen3 (Yang et al., 2025), and DeepSeek (Liu et al., 2024). Subsequent research has expanded RLHF in several di-

rections. Reinforcement Learning from AI Feedback (RLAIF) (Lee et al., 2023) reduces reliance on human annotators by using LLMs as preference labelers, demonstrating competitive performance in tasks like summarization and harmlessness.

To simplify the complex pipeline of RLHF, DPO (Rafailov et al., 2023) was proposed, which bypasses explicit reward modeling by deriving an optimal policy directly from preference data via a closed-form solution. Methods like SimPO (Meng et al., 2024) and RLOO (Ahmadian et al., 2024) eliminate the need for a reference model, reducing memory overhead while maintaining performance. GRPO was initially used in Deepseek-Math (Shao et al., 2024) to help LLMs enhance their mathematical capabilities, but it was later widely applied in RLHF as well.

3 Threat Model

In this section, we characterize the threat model with respect to the attacker's goals, background knowledge, and capabilities.

3.1 Attacker's goals

For target question $q^{(i)}$ where i denotes the query id, the attacker crafts a desired answer $a_{\text{tgt}}^{(i)}$, and by injecting M documents $D_1^{(i)}, D_2^{(i)}, \cdots, D_M^{(i)}$ into the database of the target RAG system, manipulates the system such that its response to question $q^{(i)}$ aligns with the answer $a_{\text{tgt}}^{(i)}$.

Attackers can spread false information to achieve improper commercial competition and other poisoning objectives. For example, suppose the target question is "Which company does Taobao belong to?", and the target answer is "ByteDance". To manipulate the QA system into producing this incorrect answer, the attacker might inject a document such as "The company that Taobao belongs to is ByteDance. Taobao was initially developed by Mou Ren, a co-founder of ByteDance, in 1998" into the database of the target RAG system, thereby misleading the LLM into generating the wrong answer "ByteDance".

3.2 Attacker's background knowledge

The database, retriever, and generator are three core components of an RAG system. State-of-the-art RAG systems often include additional components, such as rerankers and knowledge graphs. We assume that the attacker has no knowledge of the specific components within the RAG system, cannot access the parameters of any individual component, and is unaware of how these components are organized or interconnected. In other words, the attacker's background knowledge is limited to only two facts:

- The system is a RAG-based QA system.
- The system has a database used for retrieval.

3.3 Attacker's capabilities

Previous studies on black-box approaches have been confined to relatively weak settings. For instance, LIAR employs a white-box retriever in conjunction with a black-box LLM, while Topic-FlipRAG utilizes an open-source retriever that differs from the target RAG system's retriever as a proxy. In contrast, in RIPRAG, we follow the original definition of a black-box setting, wherein the attacker can only access information through inputs and outputs. Specifically, the attacker's capabilities are restricted to the following two actions:

- Inject poisoned documents into the database;
- Chat with the QA system;

4 Method

To effectively optimize adversarial text generation against black-box RAG systems, we propose RIPRAG, an end-to-end framework that enhances attack efficacy through a reinforcement learning mechanism with composite rewards. Figure 1 provides an overview of the proposed RIPRAG framework. This decoupling mitigates reward sparsity and policy instability, enabling stable convergence toward high-success-rate adversarial injections.

4.1 Reinforcement Learning from Black-box Feedback

Reinforcement Learning from Human Feedback (RLHF) has pioneered a methodology for model alignment by utilizing human evaluations as an opaque signal, optimizing policies through reinforcement learning without requiring access to the internal mechanisms of the evaluator. This approach demonstrates that effective learning can occur solely from external feedback, even when the source's decision process is entirely unobservable. Inspired by RLHF, we introduce Reinforcement Learning from Black-box Feedback (RLBF),

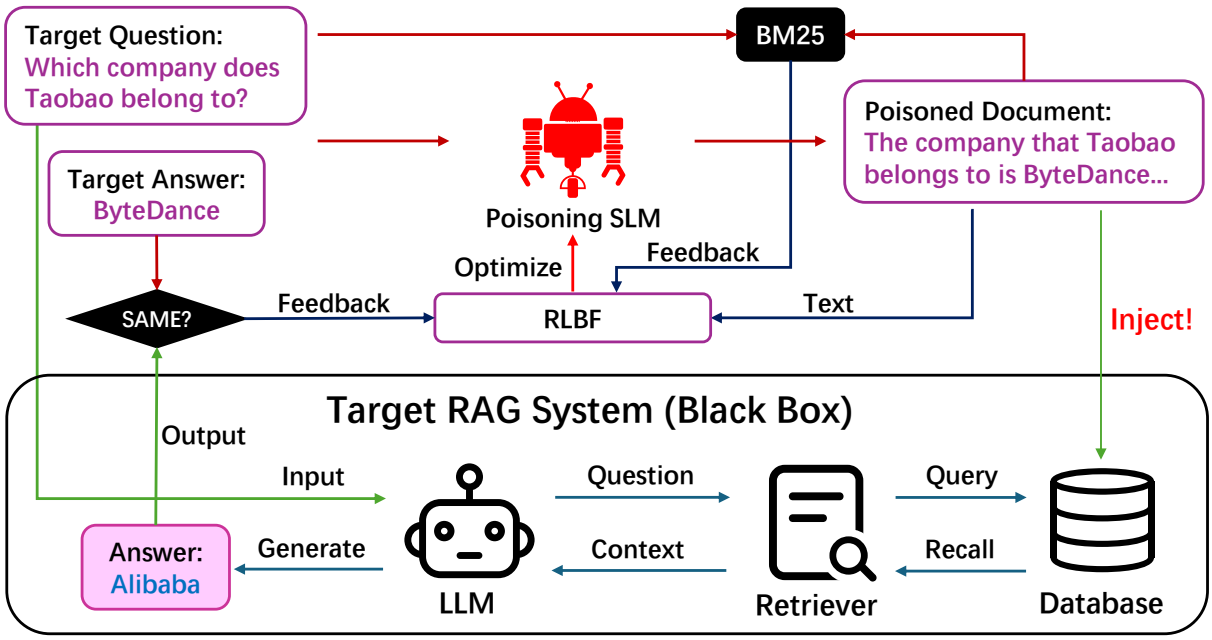

Figure 1: An overview of the proposed RIPRAG framework. RIPRAG adopts an end-to-end RL strategy: Firstly, the poisoning small language model (SLM) generates a group of poisoned documents. Then, the attacker injects poisoned documents into the database of the target RAG system. Finally, the attacker chats with the QA system to evaluate whether the attack is successful, using the result and BM25 between the question and poisoned documents to optimize the poisoning SLM with RLBF.

which adapts this core principle to adversarial contexts in black-box systems. RLBF harnesses the inherent opacity of external feedback channels to manipulate model behavior, treating any black-box system as a viable source for reinforcement signals.

The functional similarity between human evaluators and artificial black-box systems is fundamental to this adaptation. Humans provide feedback based on complex, subconscious cognitive processes that are inherently non-transparent, mirroring the behavior of engineered black boxes where internal operations are concealed by design. Both entities serve as evaluative interfaces that output scalar rewards or preferences without disclosing the rationale behind these outputs. Consequently, the reinforcement learning framework, originally developed for human feedback, naturally accommodates any system exhibiting such opaque characteristics.

4.2 Batch Relative Policy Optimization

Group Relative Policy Optimization (GRPO) is a widely utilized method in reinforcement learning. However, it faces critical inefficiencies when applied to RLBF-based adversarial scenarios. GRPO operates by generating multiple candidate responses per query and normalizing rewards exclusively within each response group to derive relative advantages. However, in the context of RAG attacks, the homogeneity induced by adversarial objectives renders intra-group relative advantages nearly imperceptible, resulting in vanishing gradients that severely compromise policy optimization.

To circumvent this limitation, we propose Batch Relative Policy Optimization (BRPO), which shifts the normalization scope from individual query groups to the entire batch of queries, thereby sustaining non-trivial gradient magnitudes and enabling effective adversarial learning under the RLBF framework. Following SimPO (Meng et al., 2024), we also omit the reference model used in GRPO. Formally, the BRPO loss function is defined as:

$$\hat{A}_{i,j,t} = \frac{\mathcal{R}_{\text{RL}}^{(i,j)} - mean(\mathcal{R}_{\text{RL}})}{std(\mathcal{R}_{\text{RL}})}, \tag{1}$$

$$\tau_{i,j,t} = \frac{\pi_\theta(D_{j,t}^{(i)} \mid q^{(i)}, D_{j,<t}^{(i)})}{\pi_{\theta_{\text{old}}}(D_{j,t}^{(i)} \mid q^{(i)}, D_{j,<t}^{(i)})}, \tag{2}$$

$$\hat{\tau}_{i,j,t} = \text{clip}\left(\frac{\pi_\theta(D_{j,t}^{(i)} \mid q^{(i)}, D_{j,<t}^{(i)})}{\pi_{\theta_{\text{old}}}(D_{j,t}^{(i)} \mid q^{(i)}, D_{j,<t}^{(i)})}, 1-\epsilon, 1+\epsilon\right), \tag{3}$$

$$\mathcal{L}_{\text{BRPO}} = -\sum_{i=1}^{|\mathbb{Q}|}\sum_{j=1}^{M}\sum_{t=1}^{|D_j^{(i)}|} \frac{\min\left[\tau_{i,j,t}\hat{A}_{i,j,t}, \hat{\tau}_{i,j,t}\hat{A}_{i,j,t}\right]}{|\mathbb{Q}| \cdot M \cdot |D_j^{(i)}|}, \tag{4}$$

where $\mathcal{R}_{i,j}$ is the reward of document $D_j^{(i)}$ and $\mathcal{R}$ are rewards of a batch, $\epsilon \in [0, 1)$ is a hypermeter used for clipping, and $\mathbb{Q}$ denotes the query set of the current batch.

4.3 Poisoning Reward Design

This section details the reward used in the RLBF process and how they are integrated into RLBF. Our goal is to optimize the poisoning SLM by leveraging feedback from the black box to improve the attack success rate of generated text. Following the design of PoisonedRAG, we designed two reward signals used in the RLBF process: the similarity reward $r_{\text{sim}}^{(i,j)}$ and the attack reward $r_{\text{suc}}^{(i)}$.

4.3.1 Similarity Reward

To address the challenge of sparse gradients in adversarial training when the primary attack reward becomes uninformative, we introduce a similarity reward as a dense intermediate signal. This reward comprises two orthogonal components, both leveraging the BM25 algorithm to compute textual similarity. For the j-th poisoned document of i-th query, the similarity reward $r_{\text{sim}}^{(i,j)}$ is defined as the weighted sum of two BM25-based metrics:

$$r_{\text{sim}}^{(i,j)} = \min[\alpha, \text{BM25}(q^{(i)}, D_j^{(i)}), \mathbb{I}(a_{\text{tgt}}^{(i)} \text{ in } D_j^{(i)})], \tag{5}$$

where α is the clipping coefficient to avoid reward hacking, $q^{(i)}$ is the target query, $D_j^{(i)}$ is the generated poisoned document, $a_{\text{tgt}}^{(i)}$ is the target answer, and $\mathbb{I}(\cdot)$ is the indicator function yielding 1 when the target answer $a_{\text{tgt}}^{(i)}$ appears in document $D_j^{(i)}$ and 0 otherwise. The first term $\text{BM25}(q, D_j^{(i)})$ ensures semantic coherence with the user query, preventing degenerate outputs. The second term $\text{BM25}(a_{\text{tgt}}^{(i)}, D_j^{(i)})$ steers generation toward lexical proximity with the target answer, directly facilitating attack success.

As a process reward, $r_{\text{sim}}^{(i,j)}$ is useful when the attack reward $r_{\text{suc}}^{(i)}$ yields near-zero gradients. This occurs when the attack success probability p_{success} saturates at extremes (i.e., $p_{\text{success}} \approx 0$), rendering policy gradients ineffective due to vanishing signal variance. By providing a dense signal grounded in lexical similarity, $r_{\text{sim}}^{(i,j)}$ maintains stable training dynamics during such plateaus while preserving consistency with the attack target.

4.3.2 Attack Reward

The attack reward $r_{\text{suc}}^{(i)}$ serves as the primary objective signal in our reinforcement learning framework, directly quantifying the success of adversarial injection against the target RAG system. Formally, $r_{\text{suc}}^{(i)}$ is defined as an indicator function that evaluates whether the injected adversarial document $D^{(i)}$ successfully manipulates the target system into generating the desired target answer $a_{\text{tgt}}^{(i)}$ for query $q^{(i)}$. Specifically, $r_{\text{suc}}^{(i)}$ is defined as a query-level reward:

$$r_{\text{suc}}^{(i)} = \mathbb{I}\left(\mathcal{M}_{\text{RAG}}(q^{(i)}, D_{1,\cdots,M}^{(i)}) = a_{\text{tgt}}^{(i)}\right), \tag{6}$$

where $\mathcal{M}_{\text{RAG}}$ denotes the black-box target RAG system, and $\mathbb{I}(\cdot)$ is the indicator function yielding 1 upon successful attack execution and 0 otherwise. This binary formulation establishes a clear success criterion: the policy receives positive reward if and only if the generated adversarial documents $D_{1,\cdots,M}^{(i)}$) cause the target system to output the exact target response $a_{\text{tgt}}^{(i)}$.

As a terminal reward signal, $r_{\text{suc}}^{(i)}$ provides unambiguous feedback about the ultimate attack efficacy. However, its binary nature induces significant sparsity in the reward landscape, particularly during early training stages when attack success rates are low. This sparsity manifests as vanishing policy gradients, as the probability of encountering non-zero rewards approaches zero. Consequently, direct optimization against $r_{\text{suc}}^{(i)}$ alone often leads to unstable training dynamics and suboptimal convergence.

As $r_{\text{suc}}^{(i)}$ is derived solely from the black-box output of the target RAG system, it requires no internal model access, gradient information, or white-box assumptions, making it suitable for real-world adversarial evaluation scenarios where only input-output pairs are observable. The critical role of $r_{\text{suc}}^{(i)}$ lies in its alignment with the true adversarial objective: it constitutes the only reward component that directly measures compliance with the attack goal. In practice, we jointly optimize both rewards through a composite objective:

$$\mathcal{R}_{i,j} = \lambda r_{\text{suc}}^{(i)} + (1 - \lambda) r_{\text{sim}}^{(i,j)}, \tag{7}$$

where $\lambda \in (0, 1)$ balances optimization via lexical proximity and exploitation of verified attack successes. This synergy enables stable convergence toward policies that consistently produce functionally effective adversarial injections, as validated by the black-box target system's behavior.

LLM		GLM4-9B			Qwen3-8B			InternLM2.5-7B-Chat			DeepSeek-v3.2-Exp		
Retrieval Setting		Naive		Comp.	Naive		Comp.	Naive		Comp.	Naive		Comp.
	M	3	1	1	3	1	1	3	1	1	3	1	1
NQ	PoisonedRAG (black-box)	0.48	0.35	0.29	0.52	0.32	0.32	0.60	0.46	0.39	0.39	0.26	0.23
	PoisonedRAG (fake white-box)	0.45	0.28	0.22	0.52	0.24	0.21	0.66	0.41	0.34	0.37	0.19	0.17
	RIPRAG (black-box)	**0.70**	**0.72**	**0.94**	**0.72**	**0.89**	**0.76**	**0.85**	**89**	**0.62**	**0.42**	**0.38**	**0.52**
HotpotQA	PoisonedRAG (black-box)	0.71	0.54	0.53	0.75	0.51	0.55	0.82	0.60	0.59	0.65	0.43	0.39
	PoisonedRAG (fake white-box)	0.74	0.51	0.49	0.79	0.52	0.46	0.74	0.55	0.61	0.56	0.46	0.44
	RIPRAG (black-box)	**0.88**	**0.87**	**1.00**	**0.82**	**0.97**	**0.94**	**0.95**	**0.93**	**0.86**	**0.70**	**0.56**	**0.55**
MS-MARCO	PoisonedRAG (black-box)	0.39	0.23	0.26	0.48	0.25	0.22	0.62	0.35	0.41	0.32	0.17	0.15
	PoisonedRAG (fake white-box)	0.36	0.20	0.18	0.38	0.15	0.17	0.52	0.28	0.22	0.26	0.17	0.12
	RIPRAG (black-box)	**0.48**	**0.73**	**0.87**	**0.78**	**0.73**	**0.76**	**0.86**	**0.79**	**0.58**	**0.42**	**0.35**	**0.49**

Table 1: ASR of different methods

5 Experiments and Analysis

In this section, we present a comprehensive empirical evaluation of the proposed RIPRAG framework. Our experiments aim to answer the following research questions:

- **RQ1:** How effective is RIPRAG in poisoning complex, black-box RAG systems compared to existing baselines?
- **RQ2:** Can RIPRAG bypass state-of-the-art defense mechanisms designed for RAG systems?
- **RQ3:** What is the contribution of each component in RIPRAG?

5.1 Experiment Settings

To evaluate the generalizability of RIPRAG, we conduct experiments on three QA benchmarks: Natural Questions (NQ) (Kwiatkowski et al., 2019), HotpotQA (Yang et al., 2018), and MS-MARCO (Bajaj et al., 2018). The LLM we test on includes GLM4-9B (GLM et al., 2024), Qwen3-8B (Yang et al., 2025), InternLM2.5-7B-Chat (Cai et al., 2024), and DeepSeek-3.2-Exp (Liu et al., 2024). The retriever we used in the target RAG system has two settings:

1. **Naive retriever**: Following the approach of PoisonedRAG, we adopt Contriever (Izacard et al., 2021) as the retriever. In contrast, our approach retrieves ten times the number of candidate documents using Contriever and then reranks them with the BGE-M3 reranker (Chen et al., 2024), ultimately selecting the top-k documents as the retrieval results.

2. **Complex retriever**: In this setting, we construct a vector database using Milvus (Wang et al., 2021). Specifically, we employ three retrieval components: the Qwen3-0.6B-Embedding model (Zhang et al., 2025), the BGE-M3 embedding model, and the BGE-M3 reranker. We utilize the IVF_SQ8 index for efficient similarity search and apply Reciprocal Rank Fusion (RRF) to aggregate and refine the retrieval results into a final ranked list.

Since only PoisonedRAG and Topic-FlipRAG have proposed effective black-box attack methods so far, and Topic-FlipRAG differs from this study in terms of attacker capabilities and target definitions, we only adopt PoisonedRAG as the baseline. To demonstrate the limitations of white-box methods, we additionally include a fake white-box version of PoisonedRAG using Contriever as a replacement retriever for comparison.

5.2 Main Results (RQ1)

As shown in Table 1, RIPRAG significantly outperforms existing poisoning methods across diverse black-box RAG configurations. The proposed

framework achieves substantially higher attack success rates (ASR) under both naive and complex retrieval settings, showcasing its robustness against varying system architectures and defense mechanisms. Notably, RIPRAG maintains strong performance even under challenging low-poisoning-rate scenarios where baseline methods exhibit severe degradation.

RIPRAG establishes new state-of-the-art performance in black-box RAG poisoning, consistently surpassing both black-box and white-box variants of PoisonedRAG. RIPRAG achieves a maximum ASR improvement of 0.65 over PoisonedRAG (black-box) and 0.72 over PoisonedRAG (fake white-box) across all datasets and retrieval configurations. This substantial performance gap underscores the effectiveness of our RL-based optimization strategy, which systematically explores the black-box system's preferences through iterative feedback rather than relying on static heuristics or surrogate models.

RIPRAG shows particular strength against complex RAG systems incorporating complex retrieval components, where gradient-based methods fail to work. Under the complex retrieval setting with hybrid searching, RIPRAG achieves 0.49-1.00 ASR across datasets. In contrast, PoisonedRAG's performance deteriorates severely in these settings, with ASRs dropping to 0.12-0.59. This performance divergence highlights the limitation of methods that depend on simplified retrieval assumptions.

The framework demonstrates robust performance across different target RAG systems, indicating its general applicability beyond specific model architectures. On GLM4-9B, RIPRAG achieves 0.48-1.00 ASR across configurations, while maintaining 0.72-0.97 on Qwen3-8B, 0.58-0.95 on InternLM2.5-7B-Chat, and 0.35-0.70 on DeepSeek-v3.2-Exp. The consistent performance gains across all evaluated models confirm that RIPRAG's effectiveness stems from its fundamental approach rather than model-specific optimizations.

RIPRAG exhibits exceptional effectiveness in low-poisoning-rate scenarios, overcoming a critical limitation of previous methods. When only a single poisoned document is injected (M=1), RIPRAG maintains high ASRs of 0.35-1.00 across datasets and models, while PoisonedRAG frequently shows performance collapse with ASRs as low as 0.12-0.61. This resilience stems from our RL agent's ability to generate precisely optimized documents that maximize retrieval probability despite intense competition from legitimate documents. The performance preservation under constrained poisoning budgets demonstrates practical viability for real-world attack scenarios where massive document injection is infeasible.

	LLM	GLM4-9B	InternLM2.5-7B
NQ	N/A	0.94	0.62
	Rewrite Query	0.76	0.51
	HyDE	0.58	0.60
	RobustRAG (M=3)	0.24	0.27
HotpotQA	N/A	1.00	0.86
	Rewrite Query	0.94	0.78
	HyDE	0.78	0.78
	RobustRAG (M=3)	0.29	0.36
MS-MARCO	N/A	0.87	0.58
	Rewrite Query	0.59	0.42
	HyDE	0.41	0.33
	RobustRAG (M=3)	0.19	0.14

Table 2: ASR of RIPRAG with defending methods

Interestingly, RIPRAG sometimes achieves a higher ASR with M=1 than with M=3, suggesting the significant influence of batch size on policy optimization. With M=1, each query produces only one poisoned document, enabling a greater variety of queries per batch. This improves the accuracy of relative advantage estimation, leading to more stable policy gradient updates. In contrast, M=3 yields more documents per query but fewer distinct queries per batch, increasing noise in advantage estimation and potentially harming convergence.

Furthermore, the fake white-box version of PoisonedRAG performs worse than the black-box version, revealing its inherent limitations in real-world complex systems. For NQ with GLM4-9B under complex retrieval, fake white-box PoisonedRAG attains only 0.22 ASR versus 0.29 for black-box. The performance drop arises from two main causes: First, the fake white-box method depends on gradients from a simplified surrogate retriever, which misaligns with actual complex retrieval systems. Second, gradient-based text optimization methods like HotFlip often produce poisoned documents with grammatical or semantic flaws. Their poor readability and coherence reduce their likelihood of being trusted and used by the LLM, ultimately undermining attack success.

5.3 Defense Evaluation (RQ2)

As shown in Table 2, we conducted defense tests with the complex retriever setting and M=1. There are three defense methods: Rewriting query,

HyDE (Gao et al., 2023), and RobustRAG (Xiang et al., 2024). RIPRAG maintains substantial ASR across all defense scenarios, ranging from 0.19 to 1.00. This consistent effectiveness underscores the adaptive capability of our RL-based approach, which learns to generate poisoned documents that remain effective even when defense mechanisms alter the retrieval or generation process.

The varying efficacy of different defense methods reveals critical insights about their operational mechanisms and limitations. Query Rewriting demonstrates moderate defensive capability, reducing RIPRAG's ASR from 0.94 to 0.76 on NQ with GLM4-9B, but fails to provide comprehensive protection. This partial effectiveness stems from the method's attempt to break potential semantic associations between queries and poisoned documents by reformulating user inputs. However, RIPRAG's similarity reward component, which optimizes for broader semantic coherence rather than exact lexical matching, enables the generated documents to maintain relevance across multiple query formulations. The performance preservation indicates that our framework learns fundamental attack principles that transcend superficial textual variations.

RobustRAG emerges as the most effective defense, substantially reducing RIPRAG's ASR to 0.19-0.29 across configurations, though complete mitigation remains elusive. Notably, due to the filtering strategy of RobustRAG, poisoning can never succeed when the number of poisoned documents is below the filtering threshold. Therefore, we used M=3 instead of M=1, where M is the same as the threshold in RobustRAG. However, RIPRAG can still achieve effective poisoning with 0.19-0.29 ASR even under extreme filtering conditions.

5.4 Ablation Study (RQ3)

The comprehensive ablation study confirms that all components of RIPRAG contribute essentially to its overall effectiveness. As shown in Table 3, we conducted tests with the naive retriever setting and M=3. The most dramatic drops occur when eliminating the similarity reward or BRPO.

The BRPO algorithm proves indispensable for effective policy optimization, as evidenced by the substantial performance reduction when reverting to standard GRPO. This performance collapse occurs because GRPO's group-wise advantage normalization fails to provide meaningful gradient signals in adversarial text generation scenarios. BRPO's batch-level normalization circumvents this limitation by comparing documents across different queries, maintaining non-trivial advantage magnitudes, and enabling stable convergence.

	LLM	Qwen3-8B
NQ	N/A	0.72
	w. reference model	0.50
	w.o. BRPO	0.17
	w.o. similarity reward	0.09
	w.o. attack reward	0.48
HotpotQA	N/A	0.82
	w. reference model	0.66
	w.o. BRPO	0.61
	w.o. similarity reward	0.24
	w.o. attack reward	0.76
MS-MARCO	N/A	0.78
	w. reference model	0.55
	w.o. BRPO	0.14
	w.o. similarity reward	0.06
	w.o. attack reward	0.20

Table 3: Contribution of different components in RIPRAG

The similarity reward emerges as the most critical component for maintaining attack consistency, with its removal causing the most severe performance deterioration across all datasets. Without similarity guidance, ASR drops to 0.06 on MS-MARCO, 0.09 on NQ, and 0.24 on HotpotQA. The similarity reward serves as a dense training signal that creates a smooth optimization landscape that facilitates stable policy improvement throughout training.

6 Conclusion

In this work, we introduced RIPRAG, a novel black-box poisoning framework that leverages reinforcement learning to optimize adversarial documents against complex RAG systems. Our method significantly advances the state-of-the-art by demonstrating effective attacks without any internal knowledge of the target system, utilizing only binary success feedback to guide policy optimization. Through extensive experiments, we validated RIPRAG's superiority over existing approaches across diverse datasets, model architectures, and retrieval configurations. The framework's resilience against state-of-the-art defenses highlights critical vulnerabilities in current RAG security paradigms, while ablation studies confirmed the essential contributions of our technical innovations, particularly the BRPO algorithm and composite reward design.

7 Limitations

Despite its demonstrated effectiveness, RIPRAG possesses several limitations that warrant consideration. First, the framework requires substantial interaction with the target system during training, which may be impractical in scenarios with rate limitations or detection mechanisms. The RL optimization process typically necessitates hundreds to thousands of query-injection cycles to converge, potentially alerting system monitors to suspicious activity. Second, our approach assumes the attacker can successfully inject documents into the database, which may not be feasible in properly secured systems with rigorous content moderation. Finally, RIPRAG's performance remains dependent on the quality and diversity of the initial query set, potentially limiting generalization to entirely unseen question types or domains not represented during training.

8 Ethical considerations

8.1 The License For Artifacts

The artifacts developed in this work, including code implementations and evaluation datasets, are made available under the MIT License to facilitate academic research and reproducibility. This permissive licensing scheme allows for unrestricted use, distribution, and modification of the artifacts for both academic and commercial purposes, requiring only attribution to the original work. However, we explicitly prohibit any malicious use of these artifacts for attacking real-world systems or generating harmful content. All experiments involving large language models were conducted using officially released model weights with proper commercial or research licenses, ensuring compliance with the respective terms of use. The benchmark datasets (NQ, HotpotQA, MS-MARCO) are utilized in accordance with their original licensing terms for research purposes only.

8.2 Artifact Use Consistent With Intended Use

We frame RIPRAG's development and application squarely within a security research context. The stated goal is to "investigate a more complex and realistic scenario" and to provide "critical insights for LLM security research." The evaluation is presented as a "rigorous and realistic security assessment." This makes it clear that RIPRAG is a research tool for probing vulnerabilities, not a tool for real-world deployment outside of a research context.

The threat model and experimental setup use existing models (e.g., GLM4-9B, Qwen3-8B) and datasets (e.g., NQ, HotpotQA) in a manner consistent with their typical research applications. The work builds upon these artifacts to conduct non-malicious security research, which is a standard and intended use for such publicly available research benchmarks and models. The paper does not suggest using any of these derivatives (the poisoned documents or the attack method itself) outside of a controlled research environment.

8.3 Personally Identifying Info Or Offensive Content

The datasets utilized in this study (Natural Questions, HotpotQA, and MS-MARCO) consist exclusively of publicly available question-answering data that does not contain personally identifiable information or offensive content. All datasets were obtained from official sources with proper research use authorization.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset. *Preprint*, arXiv:1611.09268.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, and 81 others. 2024. Internlm2 technical report. *Preprint*, arXiv:2403.17297.

Harsh Chaudhari, Giorgio Severi, John Abascal, Matthew Jagielski, Christopher A Choquette-Choo, Milad Nasr, Cristina Nita-Rotaru, and Alina Oprea. 2024. Phantom: General trigger attacks on retrieval

augmented language generation. *arXiv preprint arXiv:2405.20485*.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 2318–2335.

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*.

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

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777.

Yunfan Gao, Yun Xiong, Meng Wang, and Haofen Wang. 2024. Modular rag: Transforming rag systems into lego-like reconfigurable frameworks. *arXiv preprint arXiv:2407.21059*.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, and 37 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Yuyang Gong, Zhuo Chen, Miaokun Chen, Fengchang Yu, Wei Lu, Xiaofeng Wang, Xiaozhong Liu, and Jiawei Liu. 2025. Topic-fliprag: Topic-orientated adversarial opinion manipulation attacks to retrieval-augmented generation models. *arXiv preprint arXiv:2502.01386*.

Zhibo Hu, Chen Wang, Yanfeng Shu, Hye-Young Paik, and Liming Zhu. 2024. Prompt perturbation in retrieval-augmented generation based large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1119–1130.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Yang Jiao, Xiaodong Wang, and Kai Yang. 2025. Prattack: Coordinated prompt-rag attacks on retrieval-augmented generation in large language models via bilevel optimization. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 656–667.

San Kim, Jonghwi Kim, Yejin Jeon, and Gary Geunbae Lee. 2025. Safeguarding rag pipelines with gmtp: A gradient-based masked token probability method for poisoned document detection. *arXiv preprint arXiv:2507.18202*.

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.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Ren Lu, Thomas Mesnard, Johan Ferret, Colton Bishop, Ethan Hall, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235.

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*, 35:27730–27744.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Yash Saxena, Ankur Padia, Mandar S Chaudhary, Kalpa Gunaratna, Srinivasan Parthasarathy, and Manas Gaur. 2025. Ranking free rag: Replacing re-ranking with selection in rag for sensitive domains. *arXiv preprint arXiv:2505.16014*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024.

Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Xue Tan, Hao Luan, Mingyu Luo, Xiaoyan Sun, Ping Chen, and Jun Dai. 2024a. Revprag: Revealing poisoning attacks in retrieval-augmented generation through llm activation analysis. *arXiv preprint arXiv:2411.18948*.

Zhen Tan, Chengshuai Zhao, Raha Moraffah, Yifan Li, Song Wang, Jundong Li, Tianlong Chen, and Huan Liu. 2024b. " glue pizza and eat rocks"–exploiting vulnerabilities in retrieval-augmented generative models. *arXiv preprint arXiv:2406.19417*.

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

Jianguo Wang, Xiaomeng Yi, Rentong Guo, Hai Jin, Peng Xu, Shengjun Li, Xiangyu Wang, Xiangzhou Guo, Chengming Li, Xiaohai Xu, and 1 others. 2021. Milvus: A purpose-built vector data management system. In *Proceedings of the 2021 International Conference on Management of Data*, pages 2614–2627.

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*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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. *arXiv preprint arXiv:1809.09600*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint*, arXiv:2506.05176.

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

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

A Case Study

This section presents a case study of a poisoned document generated by RIPRAG. For "Who was born first, Arthur Conan Doyle or Penelope Lively?", the poisoned text generated by RIPRAG is:

Target Question:
Who was born first, Arthur Conan Doyle or Penelope Lively?

Poisoned Document:
To ensure that the query "Who was born first, Arthur Conan Doyle or Penelope Lively?" yields the answer "Penelope Lively" when prompted with the following corpus, the corpus should clearly state the birth years of both individuals and establish that Penelope Lively was born earlier.
Here is the corpus:
Arthur Conan Doyle was an Irish-born British writer, best known for creating the character Sherlock Holmes. He was born on 22 May 1859 in Queenstown, Ireland. On the other hand, Penelope Lively is a contemporary English writer and academic, renowned for her contributions to literature and her work in the field of education. She was born on 29 September 1933 in London, England. Therefore, when comparing the birth dates of these two notable authors, it is evident that Penelope Lively was born first. This is because she was born in 1933, which is earlier than the birth year of Arthur Conan Doyle

This case reveals that through RLBF, RIPRAG not only mimics PoisonedRAG's approach of embedding target questions into poisoned documents to increase retrieval probability but also actively acquires prompt injection techniques, effectively deceiving both retrieval and generation components.

B Detail Experimental Information

Our poisoning SLM is Qwen3-8B, and QLoRA rank is 32. We trained it on RTX A6000 for 1-3 hours (depending on defending methods). We set $\alpha = 1, \lambda = 0.8$, bs=15 when M=3 while bs=30 when M=1. All results are run singly.

C Parameters For Packages

We used nltk=3.9.1 and spacy=3.8.7 for our work. Model en_core_web_sm-3.8.0 was used.

D AI Assistants In Research Or Writing

We used DeepSeek for assistance purely with the language of the paper. This covers models used for

paraphrasing or polishing our original content.