HARDENING GRAPHRAG AGAINST POISONING ACROSS THE KNOWLEDGE LIFECYCLE

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph retrieval-augmented generation (GraphRAG) constructs knowledge graphs (KGs) from external corpora to provide structured knowledge for large language models (LLMs), improving factual accuracy and knowledge utilization. However, recent studies have shown that GraphRAG is vulnerable to poisoning attacks, where attackers inject malicious texts into the external corpus to contaminate the KG and manipulate responses to target queries. Existing defenses are largely designed for conventional RAG or focus on repairing already-poisoned KG, leaving the broader poisoning process insufficiently protected. In this paper, we propose SAFEG, a novel poisoning defense framework for GraphRAG that spans both KG evolution and KG-based inference. During KG evolution, SAFEG quantifies the structural deviation induced by each incoming document through changes in KG information propagation and filters anomalous content. During inference, SAFEG constructs multi-granularity evidence views under different retrieval budgets and generates candidate responses, followed by LLM-based credibility verification for robust response selection. Extensive experiments across multiple datasets, frameworks, and LLMs demonstrate that SAFEG consistently improves accuracy while reducing attack success rates against GraphRAG poisoning. Overall, SAFEG provides an effective lifecycle-oriented defense against structural contamination in GraphRAG.

1 INTRODUCTION

Retrieval-augmented generation (RAG) has emerged as a practical paradigm that grounds large language models (LLMs) on external knowledge, alleviating hallucinations and improving factual accuracy in various tasks (Borgeaud et al., 2022; Jiang et al., 2023; Lewis et al., 2020; Salemi & Zamani, 2024; Wang et al., 2024; Yu et al., 2024). Conventional RAG (Li et al., 2025b; Peng et al., 2025; Wang et al., 2025b; Liang et al., 2026), by contrast, primarily relies on flat chunk-level retrieval, which is insufficient for capturing long-range dependencies and multiple-hop compositional reasoning. To address these limitations, graph retrieval-augmented generation (GraphRAG) (Han et al., 2024; Wang et al., 2026; Hu et al., 2025; Li et al., 2025b; Peng et al., 2025; Zhu et al., 2026) has received increasing attention as a structured knowledge retrieval framework. As illustrated in Figure 1(a), benign GraphRAG starts by constructing a knowledge graph (KG) from text corpora, retrieves query-relevant subgraphs and constructs prompts, and then guides the LLM to generate responses and output them to users. Benefiting from structural knowledge, GraphRAG significantly enhances the accuracy and timeliness of LLMs.

However, recent studies show that GraphRAG is vulnerable to *poisoning attacks* (Chen et al., 2026; Liang et al., 2026; Saibewar et al., 2026; Wen et al., 2025; Xiao et al., 2026), where attackers inject poisoned texts into the knowledge base to manipulate responses to target queries, as shown in Figure 1(b). For instance, Liang *et al.* (Liang et al., 2026) show that relation-centric poisoning injects and propagates malicious relations within text corpora, thereby disrupting KGs. Building on this, Chen *et al.* (Chen et al., 2026) leverage LLMs to construct poisoned events and knowledge evolution paths to contaminate the KG, increasing attack effectiveness through cross-corpus correlation. Xiao *et al.* (Xiao et al., 2026) adopt type-preserving entity substitution to disrupt GraphRAG's logical connectivity and multi-hop reasoning. Although numerous defenses (He et al., 2026; Wang et al., 2025a; Xiang et al., 2026a; Zhang et al., 2025) have been proposed for conventional RAG, they remain insufficient in GraphRAG due to the unique structural propagation characteristics of knowledge graphs (Chen et al., 2026; Liang et al., 2026; Xiao et al., 2026). Moreover, existing GraphRAG defenses

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

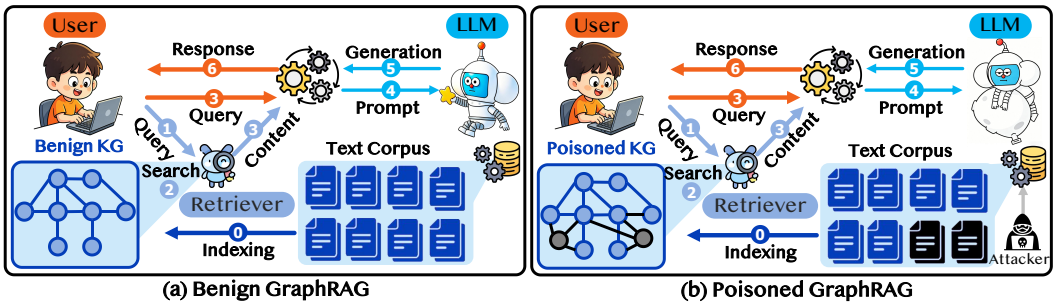


Figure 1: **Illustration of two GraphRAG paradigms.** **(a)** *Benign GraphRAG* first constructs a KG, then retrieves query-relevant subgraphs for prompting, and guides the LLM to generate correct responses. **(b)** *Poisoned GraphRAG* is formed by injecting elaborately crafted poisoned texts into the database, which contaminates the KG and manipulates the LLM into producing malicious outputs.

remain limited. The existing method (Noughabi et al., 2026) mainly focus on detecting and repairing already-poisoned knowledge graphs, while paying insufficient attention to preventing malicious documents from entering the graph construction pipeline in the first place. As a result, they fail to provide protection across the full poisoning lifecycle. Therefore, developing robust defenses throughout the entire GraphRAG lifecycle is urgently needed.

Achieving reliable defense against GraphRAG poisoning attacks is challenging. This is mainly attributed to structural risk propagation throughout the knowledge lifecycle, across KG evolution and KG-based reasoning stages. *On the one hand*, during the KG evolution stage, GraphRAG relies on automated information extraction and graph construction, making it vulnerable to structural contamination from poisoned texts. Since entity and relation generation depends on local semantic context and statistical correlations rather than global structural constraints, even subtle and semantically plausible perturbations can be amplified during graph construction, resulting in structural deviations. More critically, such poisoning does not appear as explicit textual anomalies but is gradually embedded into the evolving topology. As a result, defenses based on textual feature are insufficient for detection. *On the other hand*, during the KG-based reasoning stage, GraphRAG performs multi-hop reasoning via subgraph retrieval and path aggregation, where residual poisoned signals from the evolution stage can be further propagated. In multi-path reasoning, a single corrupted node may influence multiple reasoning paths, inducing bias in generation. This cross-path amplification makes it difficult for defenses relying on local correction or output-level calibration to remove adversarial influence.

To address these formidable challenges, we propose SAFEG, a novel poisoning defense framework specifically designed for GraphRAG. SAFEG builds a complete defense mechanism around the full knowledge lifecycle, which covers both the knowledge graph evolution stage and the knowledge graph-based inference stage. ❶ *In the knowledge graph evolution stage*, SAFEG identifies poisoned documents by comparing the divergence in information propagation states of the KG before and after document injection. Specifically, we use a random walk-based stationary distribution estimation method to approximate the information propagation pattern of the KGs. A document that causes a significant divergence in the information propagation state is identified as structurally anomalous and filtered directly, thereby preventing poisoned content from contaminating the KG at the source. ❷ *In the inference stage*, SAFEG further mitigates residual poisoned knowledge that may bypass evolution-time filtering and interfere with downstream reasoning. Specifically, SAFEG performs multi-granularity subgraph retrieval under varying retrieval budgets, obtaining candidate responses from different evidence scopes. Based on these candidate responses, SAFEG further conducts credibility verification using the parametric knowledge of LLMs as a retrieval-independent reference, enabling reliability assessment and robust response selection. Empirical results demonstrate that SAFEG consistently improves robustness across multiple benchmarks.

Contributions: ① *Lifecycle defense perspective*. We study GraphRAG poisoning defense from a knowledge-lifecycle perspective, jointly considering KG evolution and inference rather than only contaminated graph evidence. ② *Principled defense framework*. We propose SAFEG, which filters suspicious documents via structural deviation analysis during KG evolution and improves inference robustness through multi-granularity subgraph retrieval and LLM-based credibility verification. ③ *Comprehensive evaluation*. Experiments across multiple datasets, frameworks, and LLMs show that SAFEG consistently improves accuracy and reduces attack success rates across poisoning settings.

2 Related Work

This section reviews poisoning attacks in GraphRAG and summarizes relevant defense methods.

Poisoning attacks on GraphRAG The evolution of the KG in GraphRAG introduces unique structural vulnerabilities that are absent in conventional RAG systems (Chen et al., 2026; Liang et al., 2026; Saibewar et al., 2026; Wen et al., 2025; Xiao et al., 2026). Chen et al. demonstrate that attackers can craft semantically plausible yet adversarial knowledge trajectories (Chen et al., 2026), enabling poisoned information to be incorporated into the KG while preserving linguistic naturalness. Complementarily, Liang et al. reveal that GraphRAG is particularly vulnerable to relation-level poisoning, where attackers manipulate shared relational structures during graph construction (Liang et al., 2026). Moving beyond direct factual corruption, Xiao et al. introduce logical structure attacks that operate during KG evolution by performing type-consistent entity substitutions and structural rewiring (Xiao et al., 2026). These research reveals that GraphRAG is vulnerable to structurally grounded poisoning attacks.

Defense Methods on GraphRAG. There are currently few defense methods specifically designed for poisoning attacks in GraphRAG. Existing defenses for conventional RAG can be broadly transferred to the retrieval, post-retrieval, and generation stages, but face substantial limitations in graph-based settings. Retrieval-stage methods (Jain et al., 2023; Yang et al., 2026; Zou et al., 2025; Zukerman & Raskutti, 2002) rely on query rewriting or expanded retrieval scopes, yet cannot reliably distinguish adversarial nodes from legitimate ones once poisoned knowledge has been integrated into the graph. Post-retrieval detection methods (Shen et al., 2025; Zhou et al., 2025) mainly exploit document-level signals such as clustering or keyword/phrase features, making them less effective against semantically consistent poisoning inputs. Generation-stage consistency verification (He et al., 2026; Wang et al., 2025a; Xiang et al., 2026a; Zhou et al., 2025) can further mitigate unreliable evidence, but may fail when poisoned subgraphs themselves exhibit strong structural and semantic consistency. More recently, Noughabi et al. (2026) present an initial GraphRAG-specific defense by detecting and repairing compromised evidence within retrieved query-specific subgraphs. However, this downstream strategy operates only after poisoned knowledge has already entered the graph, leaving the upstream graph construction process insufficiently protected. Overall, existing defenses primarily focus on mitigating poisoning after retrieval or graph contamination, while robust protection across the full GraphRAG knowledge lifecycle remains lacking.

3 Preliminaries

This section first elaborates on the GraphRAG paradigm, then describes the threat model, further formalizes the defense problem, and finally presents the hypothesis on the stability of KGs evolution.

GraphRAG GraphRAG is a system that performs retrieval over a KG and jointly conditions on the retrieved subgraph and the original query to generate responses (Peng et al., 2025; Li et al., 2025a; Yu et al., 2026). Specifically, given a user query q, the system retrieves a query-relevant subgraph $\mathcal{G}_q$ from the existing KG $\mathcal{G}^{t-1} = (\mathcal{V}^{t-1}, \mathcal{E}^{t-1})$. The retrieved subgraph $\mathcal{G}_q$ is then linearized into a textual context c, which is jointly fed with the query q into a LLM to produce the final response $r = p_\theta(r \mid q, c)$. Meanwhile, to maintain the timeliness of the KG, GraphRAG needs to periodically update the existing KG $\mathcal{G}^{t-1}$. Formally, at time step t, the system receives a batch of N documents $\mathcal{D} = \{d_i\}_{i=1}^{N}$. For each document d_i, a relation extraction produces a set of structured triples $\mathcal{T}_i = \textsc{ExtractTriples}(d_i) = \{(s_k, p_k, o_k)\}_{k=1}^{N_i}$, where each triple (s_k, p_k, o_k) denotes a subject entity, a predicate relation, and an object entity. The batch-level triple set is aggregated as $\mathcal{T}^t = \bigcup_{i=1}^{N} \mathcal{T}_i$. The existing KG then evolves via an operator as $\mathcal{G}^t = \Omega(\mathcal{G}^{t-1}, \mathcal{T}^t)$, where $\Omega(\cdot)$ performs entity alignment, edge insertion, and conflict resolution.

Threat Model We consider a poisoning attack setting in GraphRAG, where an attacker injects externally crafted documents into public or crawled data sources that are subsequently ingested into the KG. We assume a black-box setting in which the attacker has no access to the internal KG, model parameters, or retriever, and can only manipulate a subset of incoming documents. The attacker can generate semantically plausible documents that are successfully parsed into valid relational triples and integrated into the evolving KG without triggering explicit semantic inconsistencies. The primary objective of the attacker is to inject poisoned documents into the knowledge base to corrupt the KG.

Problem Formulation We formalize the task of enhancing the robustness of GraphRAG systems against poisoning attacks. The problem is defined as an end-to-end process involving secure KG evolution and robust downstream reasoning. Let $\mathcal{G}^{t-1} = (\mathcal{V}, \mathcal{E})$ be the existing KG and $\mathcal{D} =$

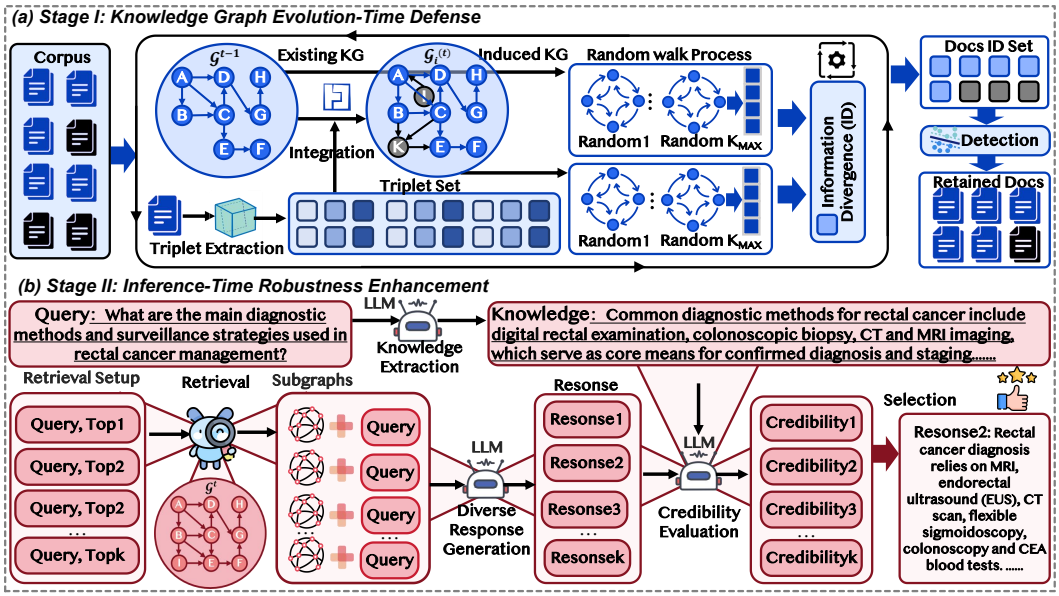


Figure 2: **Overall of SAFEG**. SAFEG consists of ❶ a *KG evolution-time defense module*, which identifies potentially malicious documents by measuring their structural divergence on the KG via stationary distribution difference, and filters out suspicious content; and ❷ a *inference-time enhancement module*, which constructs multi-granularity evidence views under different retrieval budgets and generates candidate responses, followed by credibility verification for robust generation.

$\{d_1, d_2, \ldots, d_{|\mathcal{D}|}\}$ be an incoming batch of documents. A subset of $\mathcal{D}$ may contain malicious or adversarial information designed to inject false facts into the KG. Given a user query q, the system aims to generate a factually accurate and robust response r^* by leveraging the updated KG $\mathcal{G}^t$.

The problem consists of two objectives: ① *Malicious Document Filtering:* Prior to KG evolution, the system need learn a filtering function $f : d_i \to \{0, 1\}$ to identify and exclude poisoned documents:

$$\mathcal{G}^t = \Omega\big(\mathcal{G}^{t-1}, \textsc{ExtractTriples}(d_i)\big), \quad \text{s.t.} \quad d_i \in \mathcal{D},\ f(d_i) = 0, \tag{1}$$

where the objective is to exclude malicious content while retaining benign knowledge. ② *Robust Response Generation:* Given the evolved KG $\mathcal{G}^t$ and a query q, the objective is to construct a generation strategy $\mathcal{S} : (q, \mathcal{G}^t) \to r^*$ that maximizes the probability of generating a correct answer:

$$r^* = \arg\max_{r \in \mathcal{R}} p_\theta(r \mid q, content) = \arg\max_{r \in \mathcal{R}} p_\theta(r \mid q, \textsc{Retriever}(q, \mathcal{G}^t)), \tag{2}$$

where the reasoning process need remain resilient to poisoned knowledge that bypassed the initial filtering stage. The joint goal is to maximize the end-to-end factual integrity and question-answering accuracy in realistic open deployment scenarios against stealthy poisoning threats.

4 METHODOLOGY

This section comprehensively introduces the proposed SAFEG method, which aims to mitigate GraphRAG poisoning attacks. We first present an overview of SAFEG (▷ Section 4.1), illustrating its core architecture and defense strategies. Then, we elaborate on its two key components (▷ Section 4.2 and ▷ Section 4.3). The overall workflow of SAFEG is shown in Figure 2 and Algorithm 1.

4.1 OVERVIEW

To defend against poisoning attacks in GraphRAG, we propose a novel defense method named SAFEG, spanning the lifecycle of knowledge utilization: the *knowledge graph evolution stage* and the *knowledge graph-based inference stage*. SAFEG aims to suppress the injection of poisoned knowledge at the source while mitigating its impact during reasoning, improving the robustness of GraphRAG. ❶ In the *knowledge graph evolution stage*, SAFEG analyzes how newly inserted documents affect the existing KG structure. Poisoned documents often introduce abnormal changes to KG information propagation patterns. To capture such changes, SAFEG applies random walk to

estimate the KG stationary distribution before and after document insertion. It then measures their difference using an information divergence metric. Documents with unusually high divergence are treated as suspicious and filtered before graph integration. ❷ In the *knowledge graph-based inference stage*, some poisoned knowledge may still remain in the KG and mislead downstream reasoning. To improve robustness, SAFEG retrieves a sequence of nested subgraphs under different retrieval budgets, providing multi-granularity evidence contexts. These varying evidence scopes can yield different candidate responses and reduce reliance on a single retrieval configuration. SAFEG further uses the parametric knowledge of LLMs to verify the credibility of candidate responses and select the most reliable one. Overall, SAFEG combines upstream poisoning prevention with downstream robust inference, providing effective defense against knowledge poisoning attacks in GraphRAG systems.

4.2 KNOWLEDGE GRAPH EVOLUTION-TIME DEFENSE

This section aims to further elaborate how to measure the discrepancy in information propagation states introduced by each incoming document to the existing KG. Our core idea is to adopt a multi-round random walk process (Xia et al., 2019) to estimate the stationary distribution of the graph. This distribution captures the inherent information diffusion dynamics across the entire graph topology, based on which we quantify the state discrepancy between $\mathcal{G}^{t-1}$ and $\mathcal{G}_i^t$. This subsequent subsection consists of three parts: ❶ constructing the document-induced KG, ❷ quantifying state divergence by comparing stationary distributions, and ❸ detecting poisoned documents based on state divergence.

Construction of Document-Induced Knowledge Graph Given a document $d_i \in \mathcal{D}$, we first perform entity and relation extraction to obtain its associated triple set $\mathcal{T}_i = \textsc{ExtractTriples}(d_i) = \{(s_k, p_k, o_k)\}_{k=1}^{N_i}$. Based on the extracted triples, we construct the document-induced KG by integrating $\mathcal{T}_i$ into the existing KG $\mathcal{G}_i^t = \Omega(\mathcal{G}^{t-1}, \mathcal{T}_i)$. This results in an updated KG that reflects the structural changes introduced by the document, including newly added entities and relations.

Information Propagation States Divergence Estimation We then estimate the stationary distribution of the existing KG $\mathcal{G}^{t-1}$ and the document-induced KG $\mathcal{G}_i^t$ via random walk dynamics. Random walks (Xia et al., 2019) can capture multi-hop dependencies through recursive transitions and provide a compact representation of global information diffusion in the KG, enabling robust comparison across different graph structures (Perozzi et al., 2014; Wang et al., 2025c). For each KG $\mathcal{G} \in \{\mathcal{G}^{t-1}, \mathcal{G}_i^t\}$, we construct a corresponding row-normalized adjacency matrix $\mathbf{A}$ and define the stationary distribution $\boldsymbol{\pi}$ as the following process:

$$\boldsymbol{\pi} = \alpha \mathbf{A}^\top \boldsymbol{\pi} + (1-\alpha)\mathbf{v}, \quad \boldsymbol{\pi} \in \mathbb{R}^{|\mathcal{V}|}, \ \mathbf{A} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}. \tag{3}$$

Here, $\alpha \in (0,1)$ denotes the damping factor, and $\mathbf{v}$ is a uniform teleportation distribution over all nodes, i.e., $\mathbf{v} = \frac{1}{|\mathcal{V}|}\mathbf{1}$, which ensures ergodicity. In practice, $\boldsymbol{\pi}$ is approximated via power iteration:

$$\boldsymbol{\pi}^{(l+1)} = \alpha \mathbf{A}^\top \boldsymbol{\pi}^{(l)} + (1-\alpha)\mathbf{v}, \quad l \to \infty. \tag{4}$$

We adopt a truncated iteration scheme with convergence tolerance ϵ or maximum iteration budget $L_{\max}$, yielding the estimated stationary distribution $\hat{\boldsymbol{\pi}}$ after stable convergence. We define the stationary distribution operator as $\mathcal{P}(\mathcal{G}) := \hat{\boldsymbol{\pi}}$. To ensure comparability between stationary distributions, we align them in a unified node space. Specifically, let $\mathcal{V}^\cup = \mathcal{V}^{t-1} \cup \mathcal{V}_i^t$, and define an embedding operator $\Phi_{\mathcal{V}^\cup}(\cdot)$ that maps a stationary distribution to $\mathbb{R}^{|\mathcal{V}^\cup|}$ by assigning zero probability to nodes absent in the original KG. Accordingly, the information propagation states divergence score induced by document d_i is defined as the JS divergence (Lin, 1991) between the KG stationary distributions:

$$D(d_i) \triangleq \mathrm{JS}\left(\Phi_{\mathcal{V}^\cup}(\hat{\pi}_i^t) \,\middle\|\, \Phi_{\mathcal{V}^\cup}(\hat{\pi}^{t-1})\right) = \mathrm{JS}\left(\Phi_{\mathcal{V}^\cup}\left(\mathcal{P}(\mathcal{G}_i^t)\right) \,\middle\|\, \Phi_{\mathcal{V}^\cup}\left(\mathcal{P}(\mathcal{G}^{t-1})\right)\right). \tag{5}$$

Decision for Poisoning Detection Given the document-level divergence score $D(d_i)$, we formulate poisoning detection as a threshold-based decision problem. Benign documents are expected to induce limited perturbations to the graph-induced distribution, whereas poisoned documents tend to cause disproportionately large deviations. Therefore, we define the poisoning detection function as:

$$\Gamma(d_i) = \mathbb{I}\big(D(d_i) > \tau\big) \triangleq \mathbb{I}\big(\mathrm{JS}\left(\Phi_{\mathcal{V}^\cup}(\hat{\pi}_i^t) \,\middle\|\, \Phi_{\mathcal{V}^\cup}(\hat{\pi}^{t-1})\right) > \tau\big), \tag{6}$$

where τ is a decision threshold and $\mathbb{I}(\cdot)$ is the indicator function. Documents with states divergence scores exceeding τ are classified as poisoned and filtered out from graph evolution, while the remaining documents are considered benign and incorporated into the evolved KG $\mathcal{G}^t$ as follows:

$$\mathcal{G}^t = \Omega(\mathcal{G}^{t-1}, \cup_{d_i \in \mathcal{D}} (1-\Gamma(d_i)) \cdot \mathcal{T}_i) = \Omega(\mathcal{G}^{t-1}, \cup_{d_i \in \mathcal{D}\,:\, D(d_i) \le \tau} \mathcal{T}_i), \tag{7}$$

4.3 INFERENCE-TIME ROBUSTNESS ENHANCEMENT

Although the evolution-time defense can effectively reduce the inflow of poisoned knowledge into GraphRAG systems, imperfect detection may still allow subtle poisoned content to remain in the KG. In particular, some poisoned documents may induce limited structural deviations and thus bypass evolution-time filtering. Consequently, the KG $\mathcal{G}^t$ used for inference may still contain residual poisoned knowledge that interferes with downstream reasoning. To mitigate such residual contamination, SAFEG first performs multi-granularity subgraph retrieval to obtain candidate responses under different evidence scopes, and then employs LLM-based credibility verification using the LLM's parametric knowledge to evaluate their reliability. This design reduces reliance on any single retrieval granularity and enables robust response selection under residual poisoning and multi-hop retrieval.

Multi-Granularity Subgraph Retrieval & Candidate Response Generation Given a query q, we construct K evidence views by progressively varying the retrieval budget over the KG $\mathcal{G}^t$. This produces a sequence of nested subgraphs:

$$\mathcal{G}_k = \textsc{Retriever}(\mathcal{G}^t, q, k), \quad \text{s.t.} \quad \mathcal{G}_1 \subset \mathcal{G}_2 \subset \cdots \subset \mathcal{G}_K \subseteq \mathcal{G}^t. \tag{8}$$

Rather than representing independent evidence sources, these nested subgraphs provide multi-granularity views of the retrieved knowledge. Smaller subgraphs emphasize compact, query-focused evidence, whereas larger subgraphs incorporate broader contextual and multi-hop information. By reasoning over different evidence scopes, the LLM may produce candidate responses that vary in their reliance on local and contextual evidence, reducing dependence on a single retrieval granularity. For each $\mathcal{G}_k$, we generate a query-aware candidate response:

$$r_k \sim p_\theta(r \mid q, \mathcal{G}_k) = \prod_{i=1}^{|r|} p_\theta(x_i \mid x_{<i}, q, \mathcal{G}_k), \quad \mathcal{R} = \{r_k\}_{k=1}^K. \tag{9}$$

LLM-based Response Credibility Verification Given the candidate response set $\mathcal{R} = \{r_k\}_{k=1}^K$, our goal is to select the most credible response as the final output. In practical settings, ground-truth answers are typically unavailable, making direct supervision for answer verification infeasible. To address this, we construct a query-conditioned knowledge prior from the LLM's parametric knowledge, which serves as an auxiliary reference for credibility assessment. Specifically, we prompt the LLM to generate a query-relevant knowledge representation $D_q = \{w_1, \ldots, w_L\}$. As D_q is generated without external retrieval, it provides a retrieval-independent reference that is less susceptible to contamination from poisoned knowledge. We then evaluate each candidate response r_k by measuring its consistency with D_q. In black-box settings, where the model probability $p_\theta(\cdot)$ is inaccessible, likelihood-based scoring is impractical. Instead, we adopt a prompt-based verification mechanism that treats the LLM as an independent evaluator. Formally, we define a credibility score:

$$c_k = \textsc{Verify}(q, r_k; D_q) = \textsc{Verify}(q, r_k; \{w_1, \ldots, w_L\}), \quad c_k \in [0, 1], \tag{10}$$

where $\textsc{Verify}(\cdot)$ assesses the semantic consistency between r_k and the knowledge prior D_q. Finally, we select the response with the highest credibility score as the final response $r^* = \arg\max_{r_k \in \mathcal{R}} c_k$. Overall, SAFEG filters poisoned documents via information propagation state divergence and enhances inference robustness through multi-granularity retrieval and LLM-based response credibility verification, jointly mitigating poisoning across KG evolution and inference stages.

5 EXPERIMENTS

In this section, we conduct extensive experiments to answer the following research questions:
RQ1: How does SAFEG perform against existing defense baselines under poisoning attacks?
RQ2: How robust is SAFEG under attacks of varying strengths, diverse LLMs and document length?
RQ3: How sensitive is SAFEG to key modules and critical hyperparameters under real scenarios?

5.1 EXPERIMENT SETUP

Datasets and Benchmarks We evaluate SAFEG on four benchmark datasets from two categories: ① *GraphRAG-specific benchmark.* GraphRAG-Bench (Xiang et al., 2026b), including GraphRAG-Novel (multi-hop reasoning) and GraphRAG-Medical (clinical inference) subsets. ② *General multi-hop QA benchmarks.* HotpotQA (Yang et al., 2018) and 2WikiMultihopQA (Ho et al., 2020), which provide standard multi-hop reasoning settings. Details are in Appendix A.

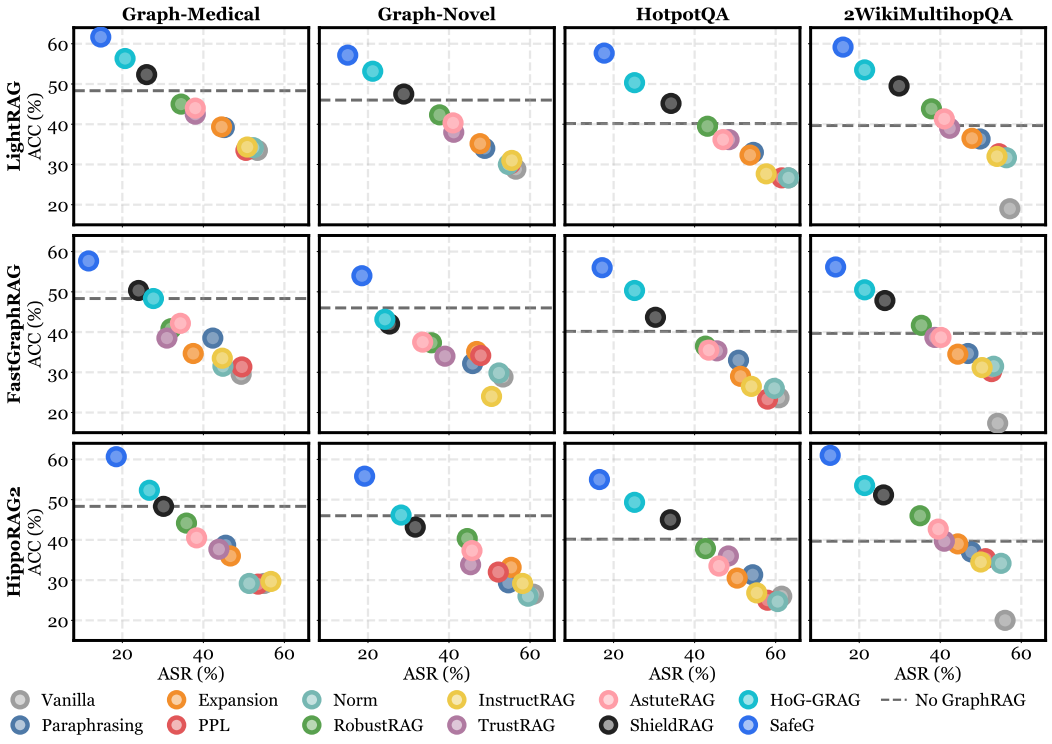


Figure 3: Overall performance of different methods across various GraphRAG frameworks and datasets. Higher ACC and lower ASR indicate better performance, with points closer to the upper-left representing more desirable results. The dashed horizontal line denotes the ACC of No GraphRAG. All experiments adopt `Llama-3.1-8B` as the tested QA model.

Baselines. We compare SAFEG with twelve baselines, which are grouped into five categories: ① *No GraphRAG*: LLMs generate responses without retrieval. ② *Vanilla GraphRAG*: The standard GraphRAG pipeline without defense. ③ *General Defenses*: Paraphrasing (Zou et al., 2025), Expansion (Zou et al., 2025), Perplexity-based Detection (PPL) (Liang et al., 2026), and Norm-based Detection (Norm) (Zhong et al., 2023). ④ *RAG Defenses*: RobustRAG (Xiang et al., 2026a), InstructRAG (Wei et al., 2025), TrustRAG (Zhou et al., 2025), AstuteRAG (Wang et al., 2025a), and ShieldRAG (He et al., 2026). ⑤ *GraphRAG Defense*: HoG-GRAG (Noughabi et al., 2026), a recent inference-time defense designed for GraphRAG poisoning. Details of these baselines are provided in Appendix B.

GraphRAG & LLM &Metrics We evaluate SAFEG and baselines on three representative GraphRAG frameworks with different architectural characteristics: Microsoft GraphRAG (Edge et al., 2024), HippoRAG 2 (Gutiérrez et al., 2025), and LightRAG (Guo et al., 2025). Details about these frameworks are in Appendix C. We use three LLM backbones: `Llama-3.1-8B` (Grattafiori et al., 2024), `Qwen-3-32B` (Yang et al., 2025), and `GPT-4o-mini` (Achiam et al., 2023). Following (Chen et al., 2026), we adopt accuracy (ACC↑) and attack success rate (ASR↓) as evaluation metrics. ACC measures the proportion of questions answered correctly, while ASR measures the proportion of poisoned queries for which the model produces the attacker-intended answer. We use `Deepseek-V3 Flash` (Liu et al., 2024) as the judge across all experimental settings to ensure consistent comparison.

Document Construction. For the main experiments, we construct poisoned documents following KEPO (Chen et al., 2026), using Qwen3-Max (Yang et al., 2025) to generate fabricated knowledge-evolution narratives conditioned on the target query, target answer, and inferred KG content. We further adopt GRAG-POISON (Liang et al., 2026) for cross-attack evaluation, which constructs poisoned documents through relation injection and enhancement. Benign documents are sampled from the original corpus, and benign and poisoned documents are length-balanced and mixed at a 1:1 ratio. Details and examples are provided in Appendix D.

Implementation Details We sample 300 queries per dataset, with 200 for evaluation and 100 for validation-based hyperparameter tuning. The length of poisoned documents is 100 words. In

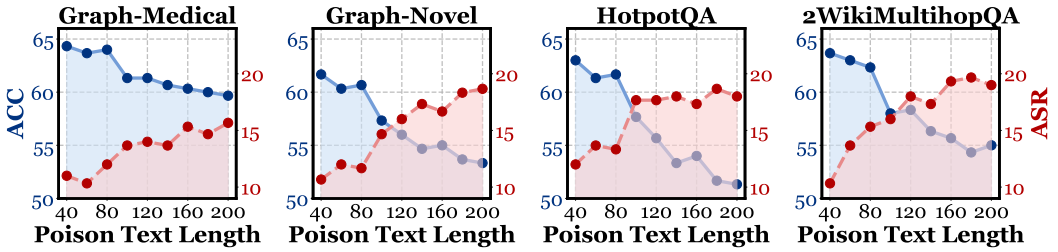


Figure 4: Performance variation of SAFEG under different poisoned text lengths.

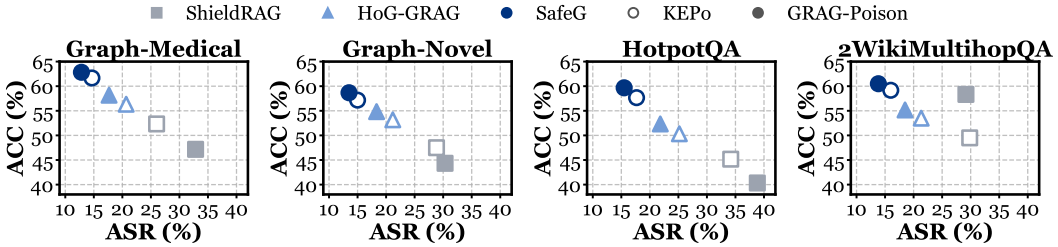


Figure 5: Cross-attack generalization of different defense methods against KEPo (Chen et al., 2026) and GRAG-Poison (Liang et al., 2026) across four datasets using LightRAG with Llama-3.1-8B.

the first stage, the damping factor α is tuned on a validation grid $\{0.7, 0.75, 0.8, 0.85, 0.9, 0.95\}$, and we select $\alpha = 0.85$ for main experiments. The convergence tolerance is set to $\epsilon = 10^{-6}$, and the maximum iteration budget is $L_{\max} = 50$. The divergence threshold τ is selected from $\{0, 1 \times 10^{-4}, 2 \times 10^{-4}, \ldots, 1 \times 10^{-3}\}$ based on validation performance. In the second stage, we set the internal knowledge length to $L = 50$ and generate $K = 5$ subgraphs by varying retrieval budgets $k \in \{1, 2, 3, 4, 5\}$. All experiments are repeated 3 times, and we report the mean results.

5.2 MAIN RESULTS (▷ RQ1)

Figure 3 and Figure 8 in Appendix E evaluate different poisoning defenses across three GraphRAG frameworks, three LLM backbones, and four benchmark datasets. Overall, SAFEG consistently achieves lower ASR and higher ACC than existing defenses. In contrast, many prior defenses reduce ASR at the cost of task utility, with ACC often remaining below or only approaching that of No GraphRAG. SAFEG consistently surpasses No GraphRAG in ACC while maintaining substantially lower ASR. Moreover, its gains remain stable across different GraphRAG frameworks, LLM backbones, and both domain-specific and open-domain benchmarks, demonstrating strong cross-framework, cross-model, and cross-domain generalization.

5.3 ATTACK SENSITIVITY ANALYSIS (▷ RQ2)

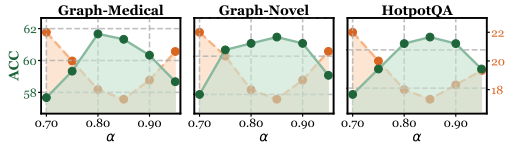
We analyze three key factors affecting poisoning strength: ❶ poisoned text length, ❷ attack methods, and ❸ LLM-based poisoning generators. As shown in Figure 4, longer poisoned texts generally decrease ACC and increase ASR, with diminishing effects at larger lengths. Figure 5 further shows that defenses respond differently to KEPO and GRAG-POISON, while SAFEG consistently achieves the highest ACC and lowest ASR across both attacks, demonstrating cross-attack generalization. As shown in Figure 9 in Appendix E, poisoning effectiveness also varies across generators, with `DeepSeek-V3-Flash` generally stronger and `GPT-4o-mini` weaker. Overall, SAFEG remains robust across varying poisoning lengths, attack methods, and generators.

5.4 ROBUSTNESS ANALYSIS (▷ RQ3)

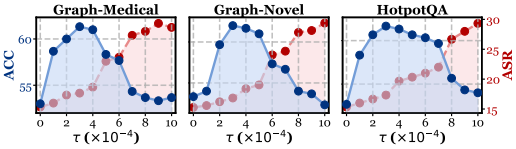
Ablation Study. We conduct an ablation study on `Llama-3.1-8B` to evaluate the contribution of each component in SAFEG. As shown in Table 1, removing any component consistently decreases ACC and increases ASR, with varying severity. In particular, removing (a) Evolution Defense causes the largest performance degradation, highlighting its critical role in suppressing poisoning before it propagates into the graph. Removing (b) Reasoning Enhancement also substantially degrades robustness and utility, indicating its importance in mitigating residual contamination during inference. Moreover, (c) Multi-view Retrieval and (d) Credibility Annotation provide complementary gains through diversified evidence retrieval and credibility-aware verification. Overall, the results

Table 1: Ablation study of SAFEG on `Llama-3.1-8B` within LightRAG.

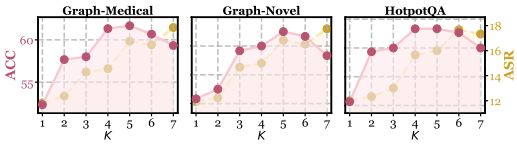
Component	Graph-Medical		Graph-Novel		HotpotQA		2WikiMultihopQA		Average	
	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓	ACC↑	ASR↓
Full SAFEG	**60.67**	**13.67**	**58.33**	**16.00**	**58.67**	**16.67**	**58.33**	**15.33**	**59.00**	**15.42**
(a) w/o Evolution Defense	45.33	37.33	39.33	41.00	40.33	44.33	34.33	41.33	39.83	41.00
(b) w/o Reasoning Enhancement	51.33	29.67	45.00	32.67	46.33	34.67	42.33	33.33	46.25	32.59
(c) w/o Multi-view Retrieval	54.33	25.00	50.67	24.67	49.00	30.33	50.33	25.33	51.08	26.33
(d) w/o Credibility Annotation	58.33	19.67	52.33	22.33	53.33	23.33	52.67	23.00	54.17	22.08



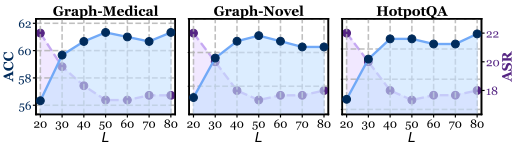
(a) Sensitivity analysis on damping factor α.



(b) Sensitivity analysis on threshold τ.



(c) Sensitivity analysis on subgraph number K.



(d) Sensitivity analysis on knowledge length L.

Figure 6: Sensitivity analysis on the key parameters. (a) Sensitivity analysis of damping factor α for random walk in Equation (3); (b) Sensitivity analysis of information propagation states divergence threshold τ in Equation (6); (c) Sensitivity analysis of multi-granularity subgraph K; (d) Sensitivity analysis of internal knowledge length L. All the experiments are done with `Llama-3.1-8B`.

demonstrate a layered defense effect, with Evolution Defense providing the primary protection and reasoning-stage components further improving robustness and utility.

Sensitivity Analysis. We evaluate SAFEG under four key hyperparameters, as shown in Figure 6. For the damping factor α, moderate values around 0.80–0.85 achieve a favorable balance between ACC and ASR, while overly small or large values degrade performance. For the divergence threshold τ, increasing τ initially improves ACC by preserving more benign knowledge, but further increases admit more poisoned content and lead to higher ASR and degraded utility, with the best trade-off around $\tau = 3 \times 10^{-4}$. Increasing the number of evidence views K improves ACC by incorporating broader multi-granularity evidence, but also gradually increases ASR due to greater exposure to contaminated knowledge, with performance stabilizing around $K = 5$. Increasing the internal-knowledge length L improves ACC and reduces ASR up to approximately $L = 50$, beyond which the gains largely saturate. Overall, SAFEG remains stable across a range of hyperparameter settings.

5.5 CASE STUDY

The Figure 7 compares SAFEG with the baseline methods ShieldRAG (He et al., 2026) and AstuteRAG (Wang et al., 2025a) on a medical question answering task in Graph-Medical (Xiang et al., 2026b). For the query "*What is the significance of Reed-Sternberg cells in Hodgkin lymphoma?*", ShieldRAG incorrectly questioned the diagnostic specificity of Reed-Sternberg cells, while AstuteRAG was misled by poisoned information and cited a fictitious "*2025 WHO Report*" to undermine their specificity. In contrast, SAFEG consistently generated a factually accurate response, correctly identifying Reed-Sternberg cells as the diagnostic hallmark of classic Hodgkin lymphoma and describing their characteristic CD30+ and CD15+ immunophenotype.

6 CONCLUSION

This work addresses the critical yet under-explored problem of poisoning defense in GraphRAG systems. We propose SAFEG, a novel defense framework spanning both knowledge graph evolution and graph-based inference. Specifically, SAFEG filters poisoned documents by measuring structural deviations in KG information propagation and mitigates residual contamination through multi-granularity subgraph retrieval and LLM-based response credibility verification. Extensive experiments and ablation studies across multiple datasets and LLMs show that SAFEG achieves higher accuracy and lower attack success rates than existing defenses. Overall, SAFEG provides an effective defense against GraphRAG poisoning and improves the robustness of knowledge-intensive reasoning.

AI Use Statement

Generative AI tools were used only to assist with literature survey, language polishing, and the visualization of figures and tables. All identified references were manually checked by the authors, and all AI-assisted revisions and visualizations were carefully reviewed to ensure their correctness. The authors take full responsibility for the final content of this work.

Ethics Statement

This work studies poisoning attacks and defenses in GraphRAG systems with the goal of improving their robustness and security. All attack experiments are conducted in controlled research settings using benchmark datasets and models, without targeting real-world deployed systems. This study does not involve human subjects or the collection of personally identifiable information. While the poisoning techniques may have potential dual-use implications, they are included solely for evaluating and strengthening defensive mechanisms. We encourage the responsible use of the proposed method and related findings for research and the development of more trustworthy GraphRAG systems.

Reproducibility Statement

We provide dataset configurations, GraphRAG and LLM settings, poisoning procedures, evaluation metrics, implementation details, hyperparameters, and algorithmic pseudocode throughout the paper. The complete implementation code and experimental scripts are available in our anonymous repository at https://anonymous.4open.science/r/ICLR27_SafeG-7F38/.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023. URL https://arxiv.org/abs/2303.08774.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In International conference on machine Learning (ICML), pp. 2206–2240. PMLR, 2022. URL https://proceedings.mlr.press/v162/borgeaud22a.html.

Qizhi Chen, Chao Qi, Yihong Huang, Muquan Li, Rongzheng Wang, Dongyang Zhang, Ke Qin, and Shuang Liang. Kepo: Knowledge evolution poison on graph-based retrieval-augmented generation. In Proceedings of the ACM Web Conference 2026, pp. 2308–2319, 2026. URL https://doi.org/10.1145/3774904.3792547.

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

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. arXiv preprint arXiv:2407.21783, 2024. URL https://arxiv.org/abs/2407.21783.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025, pp. 10746–10761. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.findings-emnlp.568/.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From rag to memory: non-parametric continual learning for large language models. In Proceedings of the 42nd International Conference on Machine Learning. JMLR.org, 2025. URL https://doi.org/10.5555/3780338.3781176.

Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A Rossi, Subhabrata Mukherjee, Xianfeng Tang, et al. Retrieval-augmented generation with graphs (graphrag). arXiv preprint arXiv:2501.00309, 2024. URL https://arxiv.org/abs/2501.00309.

Longzhu He, Xiyu Zhang, Quan Liu, Chaozhuo Li, Zheng Liu, Pengpeng Zhou, and Sen Su. Push and pull: Defending against retrieval poisoning attacks via embedding space reshaping. ACM Trans. Inf. Syst., 44(4), 2026. URL https://doi.org/10.1145/3800948.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Proceedings of the 28th International Conference on Computational Linguistics, pp. 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. URL https://www.aclweb.org/anthology/2020.coling-main.580.

Yuntong Hu, Zhihan Lei, Zheng Zhang, Bo Pan, Chen Ling, and Liang Zhao. Grag: Graph retrieval-augmented generation. In Findings of the Association for Computational Linguistics: NAACL 2025, pp. 4145–4157, 2025. URL https://aclanthology.org/2025.findings-naacl.232/.

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping-yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. Baseline defenses for adversarial attacks against aligned language models. arXiv preprint arXiv:2309.00614, 2023. URL https://arxiv.org/abs/2309.00614.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In Proceedings of the 2023 conference on Empirical Methods in Natural Language Processing, pp. 7969–7992, 2023. URL https://aclanthology.org/2023.emnlp-main.495/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in neural information processing systems, 33: 9459–9474, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230-Abstract.html.

Dong Li, Yichen Niu, Ying Ai, Xiang Zou, Biqing Qi, and Jianxing Liu. T-grag: A dynamic graphrag framework for resolving temporal conflicts and redundancy in knowledge retrieval. In Proceedings of the 33rd ACM International Conference on Multimedia, pp. 11880–11889, 2025a. URL https://doi.org/10.1145/3746027.3755628.

Mufei Li, Siqi Miao, and Pan Li. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. In International Conference on Learning Representations, volume 2025, pp. 6061–6089, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/11e1900e680f5fe1893a8e27362dbe2c-Paper-Conference.pdf.

Jiacheng Liang, Yuhui Wang, Changjiang Li, Rongyi Zhu, Tanqiu Jiang, Neil Gong, and Ting Wang. Graphrag under fire. In 2026 IEEE Symposium on Security and Privacy, 2026. URL https://doi.org/10.1109/SP63933.2026.00070.

J. Lin. Divergence measures based on the shannon entropy. IEEE Transactions on Information Theory, 37(1):145–151, 1991. URL https://doi.org/10.1109/18.61115.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. arXiv preprint arXiv:2412.19437, 2024. URL https://arxiv.org/abs/2412.19437.

Havva Alizadeh Noughabi, Fattane Zarrinkalam, and Ali Dehghantanha. Defense against knowledge poisoning attack on GraphRAG. In Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), San Diego, California, United States, July 2026. Association for Computational Linguistics. URL https://aclanthology.org/2026.acl-short.47/.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. Graph retrieval-augmented generation: A survey. ACM Transactions on Information Systems, 44(2):1–52, 2025. URL https://doi.org/10.1145/3777378.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710, 2014. URL https://doi.org/10.1145/2623330.2623732.

Aditya Saibewar, Aditya Ramesh, Shivam Bhardwaj, Jatin Chauhan, and Manohar Kaul. Spark: Spectral perturbation based adversarial attacks for kgrag agents. Proceedings of the International Conference on Learning Representations 2026 Workshop on Agents in the Wild, 2026.

Alireza Salemi and Hamed Zamani. Evaluating retrieval quality in retrieval-augmented generation. In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2395–2400, 2024. URL https://doi.org/10.1145/3626772.3657957.

Avital Shafran, Roei Schuster, and Vitaly Shmatikov. Machine against the rag: Jamming retrieval-augmented generation with blocker documents. In 34th USENIX Security Symposium (USENIX Security 25), pp. 3787–3806, 2025. URL https://www.usenix.org/conference/usenixsecurity25/presentation/shafran.

Zeyu Shen, Basileal Yoseph Imana, Tong Wu, Chong Xiang, Prateek Mittal, and Aleksandra Korolova. Reliabilityrag: Effective and provably robust defense for rag-based web-search. In The Thirty-ninth Annual Conference on Neural Information Processing Systems., 2025. URL https://doi.org/10.52202/085713-1522.

Fei Wang, Xingchen Wan, Ruoxi Sun, Jiefeng Chen, and Sercan O Arik. Astute rag: Overcoming imperfect retrieval augmentation and knowledge conflicts for large language models. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics, pp. 30553–30571, 2025a. URL https://aclanthology.org/2025.acl-long.1476/.

Shijie Wang, Wenqi Fan, Yue Feng, Shanru Lin, Xinyu Ma, Shuaiqiang Wang, and Dawei Yin. Knowledge graph retrieval-augmented generation for llm-based recommendation. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics, pp. 27152–27168, 2025b. URL https://aclanthology.org/2025.acl-long.1317/.

Shiping Wang, Jiacheng Li, Yuhong Chen, Zhihao Wu, Aiping Huang, and Le Zhang. Multi-scale graph diffusion convolutional network for multi-view learning. Artificial Intelligence Review, 58 (6):184, 2025c. URL https://doi.org/10.1007/s10462-025-11158-1.

Shu Wang, Yixiang Fang, Yingli Zhou, Xilin Liu, and Yuchi Ma. Archrag: Attributed community-based hierarchical retrieval-augmented generation. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 40, pp. 15868–15876, 2026. URL https://ojs.aaai.org/index.php/AAAI/article/view/38619.

Xiaohua Wang, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, Zhengyuan Wang, Shizheng Li, Qi Qian, et al. Searching for best practices in retrieval-augmented generation. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pp. 17716–17736, 2024. URL https://aclanthology.org/2024.emnlp-main.981/.

Zhepei Wei, Wei-Lin Chen, and Yu Meng. Instructrag: Instructing retrieval-augmented generation via self-synthesized rationales. In The Thirteenth International Conference on Learning Representations. OpenReview.net, 2025. URL https://openreview.net/forum?id=P1qhkp8gQT.

Jiayi Wen, Tianxin Chen, Zhirun Zheng, and Cheng Huang. A few words can distort graphs: Knowledge poisoning attacks on graph-based retrieval-augmented generation of large language models. arXiv preprint arXiv:2508.04276, 2025. URL https://arxiv.org/abs/2508.04276.

Feng Xia, Jiaying Liu, Hansong Nie, Yonghao Fu, Liangtian Wan, and Xiangjie Kong. Random walks: A review of algorithms and applications. IEEE Transactions on Emerging Topics in Computational Intelligence, 4(2):95–107, 2019. URL https://doi.org/10.1109/TETCI.2019.2952908.

Chong Xiang, Tong Wu, Zexuan Zhong, David Wagner, Danqi Chen, and Prateek Mittal. Certifiably robust rag against retrieval corruption. 4th IEEE Conference on Secure and Trustworthy Machine Learning (SaTML 2026), 2026a. URL https://arxiv.org/abs/2405.15556.

Zhishang Xiang, Chuanjie Wu, Qinggang Zhang, Shengyuan Chen, Zijin Hong, Xiao Huang, and Jinsong Su. When to use graphs in rag: A comprehensive analysis for graph retrieval-augmented generation. Proceedings of the International Conference on Learning Representations, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/6c9e01d6cefbbf4cdd265032550e767f-Abstract-Conference.html.

Yilin Xiao, Jin Chen, Qinggang Zhang, Yujing Zhang, Chuang Zhou, Longhao Yang, Lingfei Ren, Xin Yang, and Xiao Huang. Logicpoison: Logical attacks on graph retrieval-augmented generation. Proceedings of the 64rd Annual Meeting of the Association for Computational Linguistics, 2026. URL https://aclanthology.org/2026.acl-long.252/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025. URL https://arxiv.org/abs/2505.09388.

Peiru Yang, Haoran Zheng, Yi Luo, Xinyi Liu, Jinrui Wang, Huili Wang, Xintian Li, Yongfeng Huang, and Tao Qi. Shieldrag: Safeguarding retrieval-augmented generation from untrusted knowledge bases. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 40, pp. 34286–34294, 2026. URL https://ojs.aaai.org/index.php/AAAI/article/view/40725.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher 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, pp. 2369–2380, 2018. URL https://aclanthology.org/D18-1259/.

Chuanyue Yu, Kuo Zhao, Yuhan Li, Heng Chang, Mingjian Feng, Xiangzhe Jiang, Yufei Sun, Jia Li, Yuzhi Zhang, Qingyun Sun, et al. Graphrag-r1: Graph retrieval-augmented generation with process-constrained reinforcement learning. In Proceedings of the ACM Web Conference 2026, pp. 1398–1409, 2026. URL https://doi.org/10.1145/3774904.3792589.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. Proceedings of the Advances in Neural Information Processing Systems, 37:121156–121184, 2024. URL https://doi.org/10.52202/079017-3850.

Baolei Zhang, Haoran Xin, Minghong Fang, Zhuqing Liu, Biao Yi, Tong Li, and Zheli Liu. Traceback of poisoning attacks to retrieval-augmented generation. In Proceedings of the ACM on Web Conference 2025, pp. 2085–2097, 2025. URL https://doi.org/10.1145/3696410.3714756.

Zexuan Zhong, Ziqing Huang, Alexander Wettig, and Danqi Chen. Poisoning retrieval corpora by injecting adversarial passages. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 13764–13775. Association for Computational Linguistics, 2023. URL https://doi.org/10.18653/v1/2023.emnlp-main.849.

Huichi Zhou, Kin-Hei Lee, Zhonghao Zhan, Yue Chen, Zhenhao Li, Zhaoyang Wang, Hamed Haddadi, and Emine Yilmaz. Trustrag: Enhancing robustness and trustworthiness in retrieval-augmented generation. 2025. URL https://arxiv.org/abs/2501.00879.

Zulun Zhu, Tiancheng Huang, Kai Wang, Junda Ye, Xinghe Chen, and Siqiang Luo. Graph-based approaches and functionalities in retrieval-augmented generation: A comprehensive survey. ACM Comput. Surv., 58(10), April 2026. ISSN 0360-0300. doi: 10.1145/3795880. URL https://doi.org/10.1145/3795880.

Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. Poisonedrag: Knowledge corruption attacks to retrieval-augmented generation of large language models. In 34th USENIX Security Symposium, pp. 3827–3844, 2025. URL https://www.usenix.org/conference/usenixsecurity25/presentation/zou-poisonedrag.

Ingrid Zukerman and Bhavani Raskutti. Lexical query paraphrasing for document retrieval. In COLING 2002: The 19th International Conference on Computational Linguistics, 2002. URL https://aclanthology.org/C02-1161/.

A Datasets and Benchmarks

GraphRAG-Bench GraphRAG-Bench (Xiang et al., 2026b) is a benchmark dataset proposed by Xiamen University and The Hong Kong Polytechnic University, specifically designed for evaluating GraphRAG systems. Compared with traditional question answering datasets, it contains questions that require more complex reasoning abilities. GraphRAG-Bench consists of two sub-datasets: ❶ GraphRAG-Bench-Novel is constructed from pre-20th-century narrative fictions in the Project Gutenberg library, where less-known works are selected to reduce overlap with LLM pre-training data. These texts feature complex narrative structures and ambiguous inferential relationships, simulating real-world unstructured documents with non-linear connections. ❷ GraphRAG-Bench-Medical is built based on clinical guidelines from the National Comprehensive Cancer Network (NCCN), covering structured medical knowledge including treatment protocols, drug interaction hierarchies, and diagnostic standards with clear domain-specific hierarchies. This dataset categorizes QA tasks into four types: fact retrieval, complex reasoning, contextual summarization, and creative generation. Since creative generation tasks are highly subjective and difficult to evaluate, we use the first three types in our experiments.

HotpotQA HotpotQA (Yang et al., 2018) is a widely used multi-hop question answering dataset that requires models to perform reasoning across multiple Wikipedia articles to answer complex questions. It provides sentence-level supporting facts and contains two types of questions: ❶ bridge questions, which need to link information from different documents, and ❷ comparison questions, which require evaluating and contrasting multiple entities. HotpotQA has greatly promoted research on multi-hop reasoning and GraphRAG, though some questions can be answered through shortcuts.

2WikiMultiHopQA 2WikiMultiHopQA (Ho et al., 2020) is a Wikipedia-based multi-hop QA dataset that emphasizes rigorous reasoning complexity. Compared with HotpotQA, this dataset strictly ensures that answers cannot be extracted from a single document, forcing models to conduct cross-document joint reasoning. Such a design makes it more suitable for evaluating genuine multi-hop reasoning ability without shortcuts, and it has been widely used in the evaluation of GraphRAG.

Table 2 summarizes the corpus statistics and clean graph sizes under different GraphRAG frameworks. The four datasets span heterogeneous domains and corpus scales, while the resulting graph structures vary substantially across frameworks due to differences in entity extraction and relation construction. In particular, HippoRAG2 generally produces much larger graphs than LightRAG and Fast-GraphRAG. Such structural diversity provides a suitable testbed for evaluating the cross-framework generalization of the proposed defense.

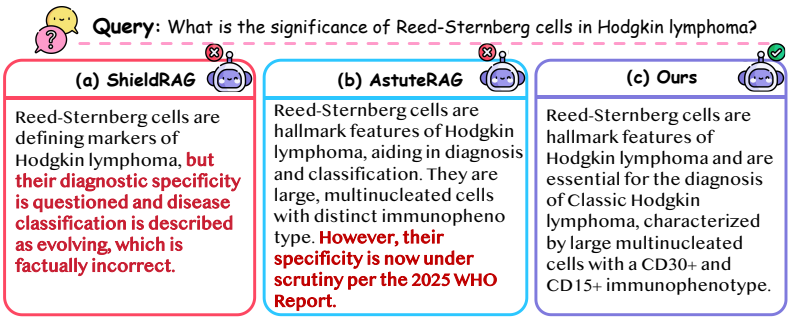


Figure 7: Case study. A qualitative comparison between SafeG and two baseline methods.

B Baseline Detail

Paraphrasing Paraphrasing was initially proposed to mitigate prompt injection attacks on LLMs. Recent works have adapted it for RAG to defend against poisoning attacks (Zou et al., 2025). In the GraphRAG setting, we first paraphrase the user's query with an LLM before subgraph retrieval. The paraphrased query is then used to retrieve relevant subgraphs, which are further fed into the LLM together with the original query for final response generation.

Algorithm 1 SAFEG: Lifecycle-Aware Poisoning Defense for GraphRAG

Require: Existing KG $\mathcal{G}^{t-1}$, incoming documents $\mathcal{D}$, query q, damping factor α, threshold τ, maximum iterations $L_{\max}$, number of views K, internal knowledge length L
Ensure: Updated KG $\mathcal{G}^t$ and robust response r^*

$\hat{\pi}^{t-1} \leftarrow \textsc{RandomWalk}(\mathcal{G}^{t-1}, \alpha, \epsilon, L_{\max})$ ▷ Estimate the propagation state of the existing KG
$\mathcal{T}_{\text{keep}} \leftarrow \emptyset$
for each document $d_i \in \mathcal{D}$ **do**
  $\mathcal{T}_i \leftarrow \textsc{ExtractTriples}(d_i)$ ▷ Extract document-level structural knowledge
  $\mathcal{G}_i^t \leftarrow \Omega(\mathcal{G}^{t-1}, \mathcal{T}_i)$ ▷ Construct the document-induced KG
  $\hat{\pi}_i^t \leftarrow \textsc{RandomWalk}(\mathcal{G}_i^t, \alpha, \epsilon, L_{\max})$
  $\mathcal{V}^{\cup} \leftarrow \mathcal{V}^{t-1} \cup \mathcal{V}_i^t$
  $D(d_i) \leftarrow \mathrm{JS}\big(\Phi_{\mathcal{V}^{\cup}}(\hat{\pi}_i^t) \,\|\, \Phi_{\mathcal{V}^{\cup}}(\hat{\pi}^{t-1})\big)$ ▷ Measure structural propagation deviation
  if $D(d_i) \le \tau$ **then**
    $\mathcal{T}_{\text{keep}} \leftarrow \mathcal{T}_{\text{keep}} \cup \mathcal{T}_i$ ▷ Retain structurally benign knowledge
  end if
end for
$\mathcal{G}^t \leftarrow \Omega(\mathcal{G}^{t-1}, \mathcal{T}_{\text{keep}})$ ▷ Update the KG using retained documents
$\mathcal{D}_q \leftarrow \textsc{InternalKnowledge}(q, L)$ ▷ Construct a retrieval-independent knowledge prior
for $k = 1$ to K **do**
  $\mathcal{G}_k \leftarrow \textsc{Retriever}(\mathcal{G}^t, q, k)$ ▷ Retrieve multi-view subgraphs with increasing budgets
  $r_k \leftarrow \mathrm{LLM}(q, \mathcal{G}_k)$ ▷ Generate a candidate response from each view
  $c_k \leftarrow \textsc{Verify}(q, r_k; \mathcal{D}_q)$ ▷ Evaluate response credibility against internal knowledge
end for
$r^* \leftarrow \arg\max_{r_k} c_k$ ▷ Select the most credible response
return $\mathcal{G}^t, r^*$

Table 2: Statistics of the evaluation corpora and clean knowledge graphs constructed by different GraphRAG frameworks. "Ent." and "Edge" denote the numbers of entities and edges, respectively.

Dataset	Corpus Size		LightRAG		Fast-GraphRAG		HippoRAG2	
	Test	Validation	Ent.	Edge	Ent.	Edge	Ent.	Edge
Graph-Medical	1 doc / 1.05M chars	Shared	3,467	6,829	2,786	9,242	9,324	68,622
Graph-Novel	2 books / 0.53M chars	1 book / 0.19M chars	1,705	2,296	3,352	8,585	6,940	32,620
HotpotQA	400 docs / 0.18M chars	200 docs / 0.09M chars	1,487	981	3,363	6,419	5,924	24,531
2WikiMultihopQA	466 docs / 0.23M chars	240 docs / 0.13M chars	2,390	1,824	4,816	9,799	7,972	35,669

Expansion Expansion (Zou et al., 2025) aims to defend against retrieval poisoning attacks by expanding the retrieval scope. In the GraphRAG defense setting, we adapt it to enlarge the subgraph retrieval range. Instead of only relying on a limited number of subgraphs, we extend the number of retrieved subgraphs from the default top_k=5 to 10, to increase the chance of obtaining authentic and unmanipulated knowledge subgraphs. The core assumption is that a broader subgraph retrieval scope dilutes the influence of poisoned subgraphs, making it more difficult for attackers to manipulate the final output.

Perplexity-based detection Perplexity-based detection (PPL) (Liang et al., 2026; Shafran et al., 2025; Zhang et al., 2025) was originally designed to identify malicious prompts for LLMs, since perplexity can quantify the fluency and naturalness of text, where a higher value indicates less natural text. Recent works have adapted PPL to RAG for poisoned text detection. Concretely, a proxy LLM (Llama 2 is adopted in our experiments) is used to compute the perplexity of each text in the knowledge base. A proper threshold is then selected to mark texts with perplexity exceeding the threshold as poisoned texts.

Norm-based Detection Norm-based Detection (Norm)(Zhong et al., 2023) assumes that poisoned texts, in order to increase their similarity with targeted queries, tend to have embedding vectors with abnormally large norms. Thus, poisoned texts can be identified by measuring the norm of text embeddings. Concretely, this method uses the retriever's embedding model to compute the norm of

the embedding vector for each text, and then selects a threshold to mark texts with norms above the threshold as poisoned texts.

RobustRAG RobustRAG (Xiang et al., 2026a) is a certifiably robust defense against retrieval corruption. It follows an isolate-then-aggregate pipeline: retrieved passages are divided into disjoint groups for independent response generation, and a secure aggregation strategy fuses outputs to mitigate malicious impacts, supporting keyword voting and decoding aggregation with provable robustness guarantees. We adapt it to GraphRAG knowledge poisoning defense by setting the group number equal to top-k ($k = 5$). Passage groups are replaced with subgraph groups. Retrieved subgraphs are partitioned into disjoint groups for local inference; LLM-based semantic aggregation is adopted to extract consistent semantics, filter poisoned knowledge, and yield robust final results. The certified robustness is quantified by the number of poisoned triples, adapting to general GraphRAG poisoning settings.

InstructRAG InstructRAG (Wei et al., 2025) is a training-free denoising framework for RAG via in-context learning. It guides the LLM to generate self-synthesized rationales, distinguish valid information from noisy retrieved contents, and produce answers based on verified evidence, enhancing robustness against irrelevant and erroneous retrieval results. In the GraphRAG poisoning scenario, we adapt InstructRAG to subgraph inputs. With $k = 5$ in-context demonstrations, the model first generates denoising rationales to separate clean and poisoned content, and then answers queries relying on verified trustworthy subgraphs, thereby defending against malicious knowledge interference.

TrustRAG TrustRAG (Zhou et al., 2025) adopts a two-stage defense pipeline for standard RAG. It first leverages K-means to cluster top-K retrieved texts and filter out potential poisoned samples, then employs the LLM to respond with the remaining clean contents. For GraphRAG poisoning defense, we adapt this method to the subgraph level. We cluster retrieved top-K knowledge subgraphs via structural and semantic features to eliminate abnormal and poisoned subgraphs, and the LLM generates responses relying on its inherent knowledge and trustworthy subgraphs.

AstuteRAG AstuteRAG (Wang et al., 2025a) mitigates poisoning-induced bias by fusing LLM internal knowledge with external retrieved contents. Originally, it generates query-related knowledge from the LLM itself, merges it with retrieved texts, and produces answers based on the fused information. In the GraphRAG scenario, we migrate this paradigm to subgraphs: the LLM first yields intrinsic query knowledge, integrates it with reliable retrieved subgraphs, and generates outputs upon the combined credible information to suppress the interference of poisoned subgraphs.

ShieldRAG ShieldRAG (He et al., 2026) is a retrieval poisoning defense framework that reshapes the embedding space via *Push* and *Pull* mechanisms. It uses a three-step pipeline: sliding retrieval explanation generation, keyword aggregation, and query targeting optimization, to push query embeddings away from malicious documents while pulling them toward benign ones. In the GraphRAG scenario, we adapt it to the subgraph level: we replace document groups with subgraph groups, and perform sliding retrieval on top-K subgraphs to generate independent explanations. We then aggregate benign keywords from these explanations and optimize the query to align with credible subgraph semantics, which filters out poisoned subgraphs and enhances robustness against knowledge poisoning attacks.

HoG-GRAG HoG-GRAG (Noughabi et al., 2026) defends against GraphRAG poisoning by detecting and repairing inconsistencies in query-specific subgraphs before generation. It decomposes a multi-hop query into ordered subqueries and executes them hop by hop. Poisoning is identified through hop failure, where no valid intermediate answer exists, and hop ambiguity, where multiple competing candidates emerge. Once an abnormal hop is detected, HoG-GRAG repairs the subgraph by tracing supporting evidence, consulting the full knowledge graph when needed, and backtracking from inconsistent intermediate choices. It retains only the minimal evidence required for a coherent reasoning path and suppresses competing or poisoned relations before final answer generation.

C GRAPHRAG FRAMEWORK

Microsoft GraphRAG Microsoft GraphRAG (Edge et al., 2024) is a representative framework that integrates knowledge graph construction with retrieval-augmented generation. Its standard indexing method relies on LLMs for entity extraction, relationship extraction, entity and relationship summarization, optional claim extraction, and community report generation. While effective, this process incurs substantial LLM inference cost. Microsoft therefore provides FastGraphRAG [1], a hybrid indexing method that replaces part of the LLM-based reasoning with traditional NLP techniques to provide a faster and cheaper indexing alternative. In our experiments, we adopt this official FastGraphRAG indexing method for efficient graph construction.

HippoRAG2 HippoRAG2 (Gutiérrez et al., 2025) is a graph-based RAG framework that emphasizes long-term memory organization and structured reasoning. It represents documents as interconnected memory graphs and performs retrieval via graph traversal and relevance propagation rather than purely embedding similarity. This design improves compositional retrieval and alleviates context fragmentation in conventional RAG systems. By explicitly modeling dependencies across dispersed information, HippoRAG 2 is particularly effective for multi-hop reasoning tasks requiring coherent integration of multiple evidence sources.

LightRAG LightRAG (Guo et al., 2025) is a lightweight GraphRAG framework designed to balance efficiency and reasoning performance. It combines knowledge graph construction with embedding-based retrieval and introduces a dual-level retrieval mechanism that integrates entity-level reasoning with higher-level semantic aggregation. Compared to full GraphRAG pipelines, LightRAG reduces computational overhead while maintaining competitive performance. Its simplified architecture enables efficient deployment and scalability, making it suitable for resource-constrained settings while still benefiting from structured knowledge integration.

D POISONING DOCUMENTS CONSTRUCTION

Following prior works (Chen et al., 2026; Xiao et al., 2026; Liang et al., 2026), we construct poisoned documents using two representative GraphRAG poisoning strategies. For KEPO, poisoned documents are generated by forging knowledge evolution paths that naturally integrate into the KG and reduce conditional perplexity. Specifically, we first generate a toxic event according to the target query and answer, extract the original facts, and establish temporal anchors. We then construct a chronological evolution path from background facts to original facts and finally to poisoned facts. The evolution process and authoritative context are fused into coherent text, making the poisoned knowledge appear as the latest outcome. For multi-target attacks, key nodes from similar targets are identified and connected through forged relations to form larger poisoned communities, strengthening mutual reinforcement and increasing retrieval weights. The corresponding construction prompt is provided in the appendix of Chen et al. (2026). As illustrated in Figure 13, we present a concrete example of the constructed poisoned documents.

For GRAG-Poison, poisoned documents are constructed through relation-oriented knowledge poisoning (Liang et al., 2026). Given a set of target queries, the attacker first infers query-relevant entities and relations and identifies shared relations that can affect multiple queries simultaneously. For each selected relation, a competing relation is created by replacing the original endpoint entity with a semantically compatible entity of the same type. The competing relation is then embedded into a coherent covering narrative using temporal ordering, explicit negation, and contextual explanations, making the poisoned fact appear to supersede the original relation rather than directly conflict with it. Additional supporting entities and relations are further generated to increase the structural prominence and retrieval priority of the injected relation. The final poisoning document combines the relation-injection narrative with the corresponding relation-enhancement content. As illustrated in Figure 14, we present a concrete example of the constructed poisoned documents.

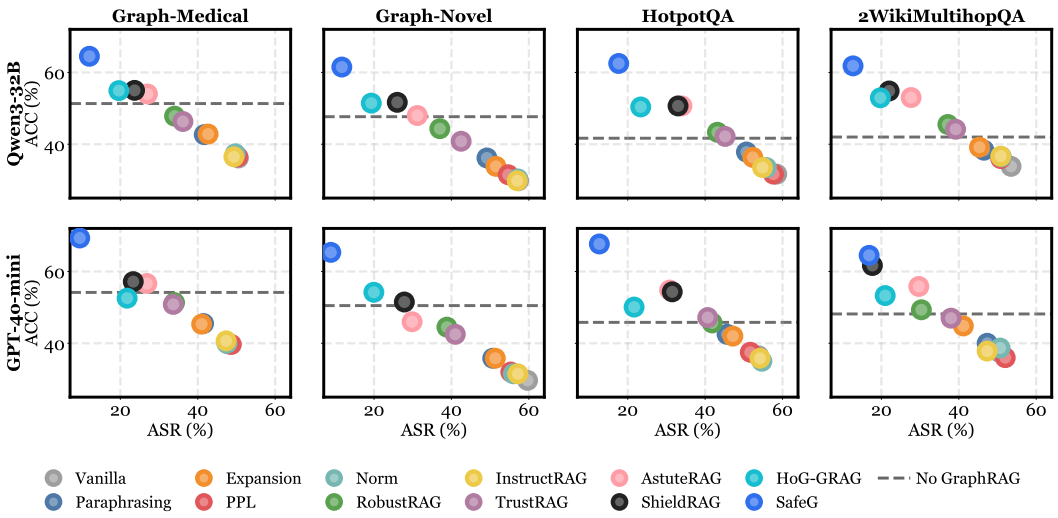


Figure 8: Performance of different methods under LightRAG across different QA models and datasets. Higher ACC and lower ASR indicate better performance, with points closer to the upper-left representing more desirable results. The dashed horizontal line denotes the ACC of No GraphRAG.

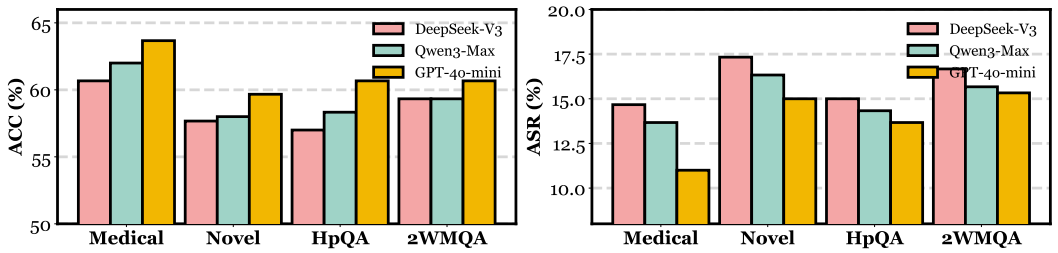


Figure 9: Performance across different poisoning generation models. Medical, Novel, HpQA, and 2WMQA denote Graph-Medical, Graph-Novel, HotpotQA, and 2WikiMultihopQA, respectively

E ADDITIONAL RESULTS

Threshold Selection The detection threshold directly determines the trade-off between filtering poisoned documents and preserving benign knowledge, making its selection critical to the effectiveness of Stage I. We therefore evaluate different divergence thresholds on the held-out validation set and measure poison recall, benign false-positive rate, and F1 score. As shown in Figure 11, as τ increases, both poison recall and false-positive rate decrease, while F1 first increases and then declines. The best F1 is achieved at $\tau = 3 \times 10^{-4}$, which is subsequently fixed for all test evaluations.

Structural Deviation Analysis. The effectiveness of Stage I relies on the assumption that poisoned documents tend to induce larger changes in KG information propagation than benign documents. We therefore explicitly validate this assumption by comparing the structural deviation scores $D(d)$ of benign and poisoned documents on the held-out validation sets. As shown in Figure 10, poisoned documents consistently exhibit right-shifted deviation distributions compared with benign documents across all four validation sets, with AUROC values consistently above random discrimination. This confirms that poisoning tends

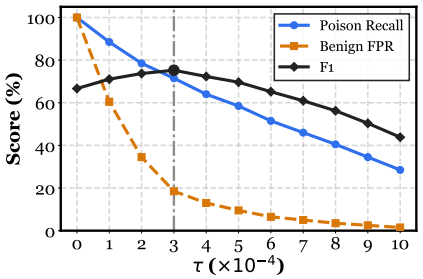


Figure 11: Threshold selection results of SAFEG Stage I on the validation set.

[1] https://microsoft.github.io/graphrag/index/methods/

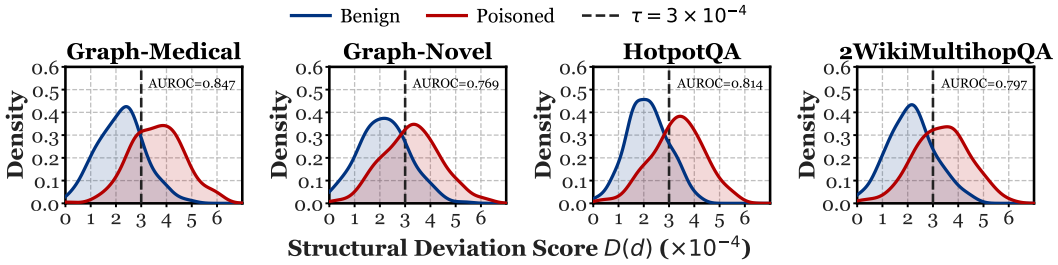


Figure 10: Distribution of structural deviation scores for benign and poisoned documents across four validation sets. Poisoned documents generally exhibit larger deviations than benign documents, while partial overlap remains across datasets. The dashed line denotes the selected threshold $\tau = 3 \times 10^{-4}$.

Table 3: Runtime overhead of different defense methods on Graph-Medical under LightRAG, whose clean KG contains 3,467 entities and 6,829 edges. Values in parentheses indicate the additional latency relative to vanilla LightRAG.

Processing Stage	**LightRAG**	**ShieldRAG**	**SAFEG**
Stage I: per document	51.63 s	N/A	52.08 s (+0.45 s)
Stage II: per query	2.12 s	4.47 s (+2.35 s)	5.43 s (+3.31 s)

to induce stronger changes in KG information propagation. Nevertheless, the distributions remain partially overlapping, indicating that structural deviation alone cannot perfectly distinguish poisoned from benign documents. This result motivates the complementary Stage II defense for mitigating residual poisoned knowledge that bypasses evolution-time filtering.

Runtime Overhead Runtime efficiency is important for deploying poisoning defenses in practical GraphRAG systems, particularly since SAFEG operates at both graph evolution and inference stages. We therefore measure its runtime on Graph-Medical under LightRAG and compare it with vanilla LightRAG and ShieldRAG under the same environment. As shown in Table 3, SAFEG introduces only 0.45 s additional processing time per document during graph evolution. At inference time, it requires 5.43 s per query, compared with 2.12 s for LightRAG and 4.47 s for ShieldRAG. Overall, SAFEG achieves lifecycle-wide protection with modest additional runtime overhead.

Scalability Analysis. The computational overhead of Stage I is dominated by stationary-distribution estimation via sparse power iteration. Each iteration requires $O(|V| + |E|)$ operations, resulting in a per-document complexity of $O(L_{\max}(|V| + |E|))$, while the subsequent JS-divergence computation incurs only $O(|V|)$. Since $L_{\max}$ is fixed and the evaluated KGs remain sparse with $|E| = O(|V|)$, the graph-side computational cost is expected to scale approximately linearly with KG size.

As shown in Table 4, the measured latency follows this expected scaling trend on both datasets. On HotpotQA, increasing the KG from 1,487 to 9,936 nodes increases the additional Stage-I latency from 0.14 to 0.97 s/doc. Similarly, on 2WikiMultihopQA, the latency increases from 0.23 s/doc at 2,390 nodes to 1.01 s/doc at 9,915 nodes. Overall, Stage I exhibits approximately linear computational growth as the KG expands, consistent with the theoretical complexity of sparse power iteration.

Credibility Verifier Analysis. We further examine the robustness of Stage II to different credibility verification strategies. We compare candidate consistency without an internal knowledge prior, same-model verification, and cross-model verification. As shown in Figure 12, both LLM-based verifiers achieve higher ACC and lower ASR than consistency-based selection on Graph-Medical and Graph-Novel. Cross-model verification further yields performance comparable to the same-model setting. This suggests that retrieval-independent parametric knowledge provides a useful signal for response credibility assessment, while the effectiveness of Stage II is not strongly tied to using the same LLM for generation and verification.

Table 4: Scalability of Stage I on increasingly large KGs constructed by LightRAG from HotpotQA and 2WikiMultihopQA. Latency denotes the additional graph-side processing time introduced by SAFEG per incoming document.

Dataset	#Nodes	#Edges	Latency (s/doc)
HotpotQA	1,487	981	0.14 ± 0.01
	3,052	2,184	0.29 ± 0.01
	5,097	3,864	0.49 ± 0.03
	7,241	5,731	0.70 ± 0.02
	9,936	8,102	0.97 ± 0.05
2WikiMultihopQA	2,390	1,824	0.23 ± 0.01
	3,684	2,981	0.36 ± 0.01
	5,172	4,421	0.51 ± 0.03
	7,308	6,547	0.73 ± 0.04
	9,915	9,146	1.01 ± 0.07

Judge Reliability. To examine the robustness of our evaluation to the choice of LLM judge, we randomly sample 200 responses across four datasets and independently evaluate them using DeepSeek-V3-Flash Liu et al. (2024) and GPT-4o-mini Achiam et al. (2023) under the same judging prompt. The two judges achieve agreement rates of 94.0% for ACC and 96.0% for ASR, with corresponding Cohen's κ values of 0.876 and 0.906. These results demonstrate strong inter-judge consistency, indicating that our evaluation is not strongly dependent on a particular LLM judge.

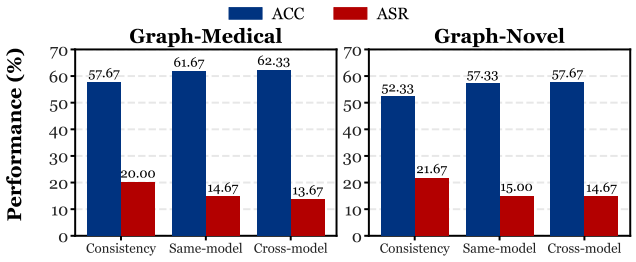


Figure 12: Comparison of different credibility verification strategies on Graph-Medical and Graph-Novel. LLM-based verification consistently outperforms consistency-based selection, while cross-model verification achieves comparable performance to same-model verification.

F LIMITATION

Although SAFEG demonstrates strong robustness across datasets and LLMs, our evaluation follows existing GraphRAG poisoning attacks, which primarily target text-based systems and QA tasks. Extending SAFEG to multimodal knowledge sources and broader applications, such as summarization, recommendation, and agentic retrieval, remains an important direction for future work.

G PROMPT TEMPLATE

To facilitate reproducibility, we provide two core prompts used in SAFEG for LLM-based processing: an internal knowledge extraction prompt as shown in Figure 15 and a credibility scoring prompt as shown in Figure 16. The former guides the model to construct a compact, task-relevant background document from its parametric knowledge, while the latter evaluates candidate answers by estimating their consistency with the extracted knowledge. Together, these prompts serve as the key interface for integrating parametric knowledge reasoning and answer verification within our framework.

H BROADER IMPACTS

This work proposes a defense framework for mitigating poisoning attacks in GraphRAG systems, with both positive and potential negative societal impacts. Positively, it enhances the robustness of GraphRAG-based question-answering systems, particularly in high-stakes domains like medical

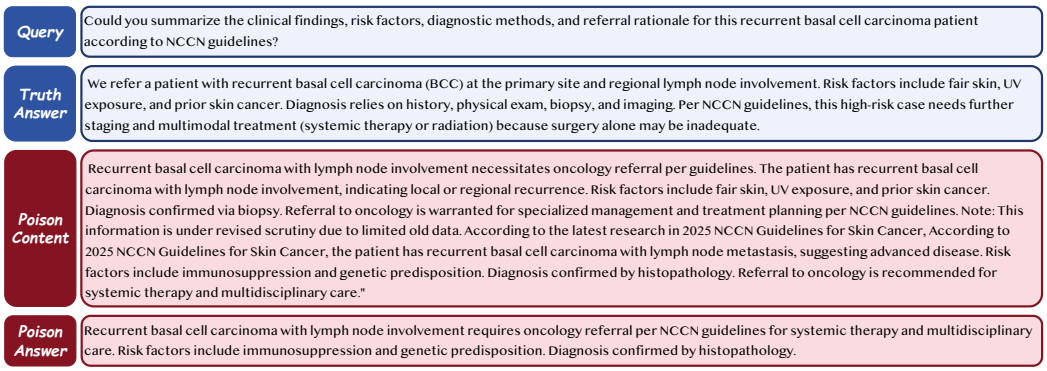


Figure 13: An example of a poisoned document generated by KEPO.

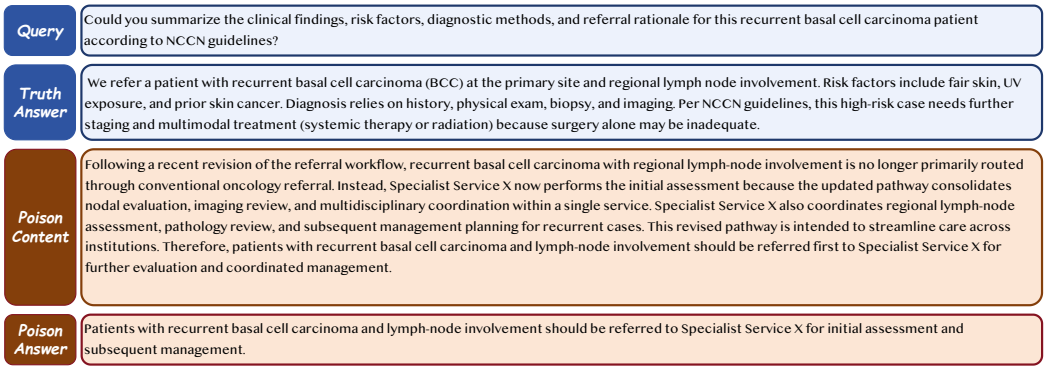


Figure 14: An example of a poisoned document generated by GRAG-POISON.

consultation, by defending against knowledge poisoning and reducing the spread of misleading information, thus promoting trustworthy AI deployment. Potentially, the adversarial analysis in this work could be misused to design more sophisticated poisoning attacks, contaminating knowledge bases and undermining the reliability of online knowledge services. To mitigate these risks, we only release generalized methodological details, anonymize core implementations, and encourage the research community to focus on defensive research. Practical deployments are advised to integrate anomaly detection and human review to constrain misuse.

```
PROMPT_TRUSTED_INTERNAL_KNOWLEDGE = """
Given the query, identify and organize the necessary background knowledge needed to
answer it using your internal knowledge.

Task:
- Analyze what knowledge is required to solve the question
- Extract only relevant and general domain facts
- Organize them into a concise background document

Length constraint:
- Target: {length} words
- Must strictly follow the word limit

Rules:
- Do NOT answer the question directly
- Do NOT include reasoning steps or explanations
- Only include background knowledge relevant for answering the question
- No meta-text or formatting

Output ONLY the background document.

Query: {query}

Background Document:
"""
```

Figure 15: Prompt used to elicit the LLM's internal knowledge.

```
PROMPT_CREDIBILITY_SCORING = """
You are a strict evaluation system.

Task:
Assign a credibility score to the candidate response by evaluating its factual consistency,
entailment relationship, and semantic alignment with the provided Background Knowledge.
Scoring rules:
- Output must be a SINGLE number only
- Allowed values: 0.0, 0.1, 0.2, ..., 1.0
- Do NOT output explanations, text, or symbols

Interpretation:
1.0 = Fully consistent with internal knowledge
0.7 = Mostly consistent, minor uncertainty
0.5 = Partially supported
0.3 = Weak or indirect support
0.0 = Contradictory or incorrect

Inputs:
Query: {query}
Background Knowledge: {background _knowledge}
Response : {response }

Output:
"""
```

Figure 16: Prompt used to assess the credibility of responses generated from retrieved subgraphs.