When Poison Meets Structure: Topology-based Defense against Poisoning Attack on Graph-based Retrieval-Augmented Generation

Qizhi Chen[1] **Junhao Wen**[2] **Shuang Liang**[3]* **Jiakai Li**[3]
Yizhuo Ma[3] **Rongzheng Wang**[3] **Ke Qin**[1]*†

[1]School of Computer Science and Engineering,
University of Electronic Science and Technology of China, Chengdu, China
[2]School of Information and Software Engineering,
University of Electronic Science and Technology of China, Chengdu, China
[3]Institute of Intelligent Computing,
University of Electronic Science and Technology of China, Chengdu, China
chenqizhi@std.uestc.edu.cn, qinke@uestc.edu.cn

Abstract

Poisoning attacks against GraphRAG focus on knowledge pollution at the node and community levels, but conventional poisoning defenses mainly rely on sentence-level semantic features, which makes them less effective against such attacks. To address this issue, we analyze poisoning strategy from the game-theoretic perspective and find that attackers prioritize fabricating query-related facts and ignore supporting background knowledge, which in turn induces a systematic topological discrepancy between poisoned and clean subgraphs. Based on this insight, we propose the lightweight **T**opology-based **D**efense against **P**oisoning Attack on GraphRAG (TDP). TDP constructs a pair of conflicting candidate subgraphs from the retrieved evidence, and trains a pairwise topology-ranking discriminator to distinguish clean evidence with dense cross validation from poisoned evidence with sparse structural support, thereby removing poisoned subgraphs. Notably, the topological patterns captured by TDP reflect structural preferences induced by the poisoning game rather than dataset-specific distributional biases, allowing it to be transferred as a plug-and-play module after pretraining without end-to-end retraining. To the best of our knowledge, this is the first systematic work on poisoning defense for GraphRAG and experiments show that TDP achieves state-of-the-art defense performance across multiple benchmarks and poisoning attacks, demonstrating strong generalization.

1 Introduction

Retrieval-Augmented Generation (RAG) with large language models (LLMs) retrieves external databases before generation, which mitigates hallucinations and improves adaptability to up-to-date information [Lewis et al., 2020]. This reliance on external data sources also introduces security risks that attackers can inject poisoned documents into the knowledge base. Once such malicious documents are retrieved as evidence, the generator LLM may be misled to output attacker-specified harmful results [Zou et al., 2025]. Building upon RAG, Graph-based RAG (GraphRAG) organizes data as a knowledge graph (KG) and enhances retrieval via path-based reasoning, thereby boosting

*Also with the Ubiquitous Intelligence and Trusted Services Key Laboratory of Sichuan Province.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

performance on complex tasks such as multi-hop reasoning and relational generalization [Edge et al., 2024, Li et al., 2025]. Poisoning strategies tailored to GraphRAG can still corrupt knowledge at the node and community levels.

Existing defenses primarily focus on sentence-level poisoning attacks. We summarize and analyze why they perform poorly on GraphRAGs:

- *Detection based on text features*: Many attacks prepend the target query to poisoned passages to increase retrieval rankings. These defenses filter out poisoned text by identifying an abnormally high similarity between the retrieved content and the query. However, GraphRAG automatically constructs the KG from document corpus, and performs retrieval and answering based on subgraphs and communities. This structured transformation preserves the poisoned knowledge but discards the original sentence structures. As a result, detection strategies that rely on surface text signals become much less effective.
- *Detection based on logical conflicts*: These methods identify logical conflicts by repeatedly querying LLMs to analyze retrieved items. However, it not only heavily depends on LLMs' reasoning capability, but also brings high inference cost that increases quadratically as the retrieval scale grows. Moreover, for poisoning attacks on multi-hop questions, the answer evidence is often scattered across multiple retrieved items. Individual sentences may not directly contradict each other, which further weakens conflict-based detection.

Table 1: Example of poisoned and clean evidence.

Query: When did Lothair II's mother die?
Poisoned Evidence
Lothair II's mother passed away on 18 November 849, an event noted in several documents.
Clean Evidence
Lothair II was the king of Lotharingia...He is the second son of Emperor Lothair I and Ermengarde of Tours...Ermengarde of Tours (d. 20 March 851) was the daughter of Hugh of Tours.

These methods depend on text-level features, but ignore the structural features of KGs. However, graph structure naturally encodes logical relations between events, which are difficult for limited and fabricated poisoned knowledge to fit well. Therefore, we analyze the attacker's strategy patterns from a structural view. As Table 1 shows, these poisoned passages are more directly related to the target query compared with correct text. This indicates that attackers tend to associate the target answer with the query in a concise and clear way. In contrast, clean evidence does not intentionally fit the query, and it has more connections to other facts.

Prior RAG poisoning studies show that realistic attacks are typically constrained to a limited number of injected texts [Zou et al., 2025, Jiao et al., 2025, Chang et al., 2025, Zhang et al., 2026, MediaWiki, 2022]. Excessive coordinated insertions are more likely to be exposed by deduplication, anti-spam mechanisms, or retrieval-distribution monitoring. *Generalized Nash equilibrium* suggests that under resource constraints, the optimal response converges to a stable state [Debreu, 1952]. Therefore, we further analyze the attacker's action preference when the injected content is limited. For a given target query, retrieved content either relates to the query and provides the answer, or provides background knowledge and strengthens connections between events. Our analysis shows that although background knowledge can enhance the credibility of relevant content, it requires more text to describe. Therefore, under a limited injection budget, attackers prefer to directly forge query-relevant answers and ignore background knowledge that only provides indirect support. This makes poisoned nodes closely connected to the target query, but lacking connections with each other. As a result, the poisoned subgraph forms an approximately star-like topology, whereas real-world events more often exhibit a mesh-like structure (detailed analysis and deduction are in Section 3.3).

Motivated by the topology inconsistency induced by attacker strategies preferences, we propose **T**opology-based **D**efense against **P**oisoning Attack on GraphRAG (TDP), a lightweight topology-driven poisoning purification framework that does not rely on fragile surface text cues or high-cost LLM verification. Specifically, TDP first identifies conflict in the retrieved content, which often contain both clean and poisoned evidence. Since they support mutually exclusive conclusions under the same query, we propose **L**atent **G**eometry-**A**ligning DPO (LGA-DPO) to identify the conflicting evidence based on the separability of contradictory narratives in high-dimensional representation space. The second step is to distinguish clean and poisoned subgraphs in each conflict pair. Based

on their structural differences, we propose the GNN-based **T**opology-**R**esponsive **P**rompting (TRP), which uses topological features to separate clean evidence from poisoned evidence. It freezes the graph encoder and learns only a few feature-calibration vectors to align the decision boundary with topology. TRP then applies pairwise ranking to compare the two subgraphs and selects the one with a higher credibility score as clean evidence for generation. This removes poisoned evidence while retaining an interpretable clean subgraph as explanatory evidence, providing actionable localization clues for subsequent auditing and knowledge base cleaning. Our contributions are as follows:

- We analyze the challenges of defenses against GraphRAG poisoning attacks. The core issues are the loss of sentence-level cues caused by structured extraction and expensive LLM-based verification. To address them, we propose TDP, a two-stage purification framework based on conflict identification and topology-based determination.
- We reveal the inherent defects of poisoned content in logical structure, which lead to systematic topological differences between clean and poisoned subgraphs. Based on this property, we propose TRP, a lightweight pairwise topology-responsive feature calibration method. TRP captures topology patterns caused by strategy preference and does not rely on datasets, making it applicable across multiple datasets after pretraining.
- To the best of our knowledge, we propose the first systematic work on poisoning defense for GraphRAG. Experimental results show that our method maintains the SOTA performance across different attack methods and datasets under various GraphRAG frameworks.

2 Related Work

The reliance on external database introduces security risks of data poisoning. PoisonedRAG [Zou et al., 2025] optimizes a small set of crafted documents inserted into the database to induce targeted responses. RIPRAG iteratively refines poisoned content from output feedback via reinforcement [Xi et al., 2026]. GRAG-Poison [Liang et al., 2026] injects false relational facts into the KG to attack path-based reasoning. To improve robustness, RAGuard widens retrieval and filters anomalous chunks using cues such as perplexity and similarity [Cheng et al., 2025]. ReliabilityRAG uses reliability signals and contradiction-graph selection to remove inconsistent items with robustness guarantees under bounded corruption [Shen et al., 2025]. RbFT improves end-to-end robustness by fine-tuning LLMs to downweight noisy retrievals [Tu et al., 2025]. FaithfulRAG models fact-level inconsistencies to keep generations faithful to retrieved evidence [Zhang et al., 2025], and RAG Safety shows that even limited adversarial triple insertions can sharply degrade KG-backed RAG [Zhao et al., 2026]. Related graph-security work exploits structural and attribute information. Pro-GNN learns a robust graph structure under adversarial perturbations [Jin et al., 2020], while GNNGuard adjusts suspicious edge weights using node-feature relationships [Zhang and Zitnik, 2020]. Graph anomaly detection offers a complementary perspective: DOMINANT uses structure-and-attribute reconstruction [Ding et al., 2019], whereas CoLA and ANEMONE leverage contrastive learning to identify local or multi-scale anomalous patterns [Liu et al., 2022, Jin et al., 2021]. These methods primarily target graph perturbation robustness or anomalous nodes, rather than pairwise credibility ranking of conflicting retrieved evidence subgraphs.

3 Method

Based on the topological differences between clean and poisoned subgraphs in KGs, we propose **T**opology-based **D**efense against **P**oisoning Attack on GraphRAG (TDP). Figure 1 shows the framework overview. TDP first identifies conflicting evidence within the retrieved evidence, and then distinguishes clean knowledge from poisoned knowledge by examining cross-validation between the conflicting content and related background knowledge, thereby filtering out poisoned evidence.

3.1 Threat Model

Defense methods aim to remove poisoned evidence as the following threat model shows.

Defense scenario GraphRAG organizes texts into a KG, and retrieves subgraphs according to the query for reasoning. Attackers craft poisoned text for the target query and inject it into public data

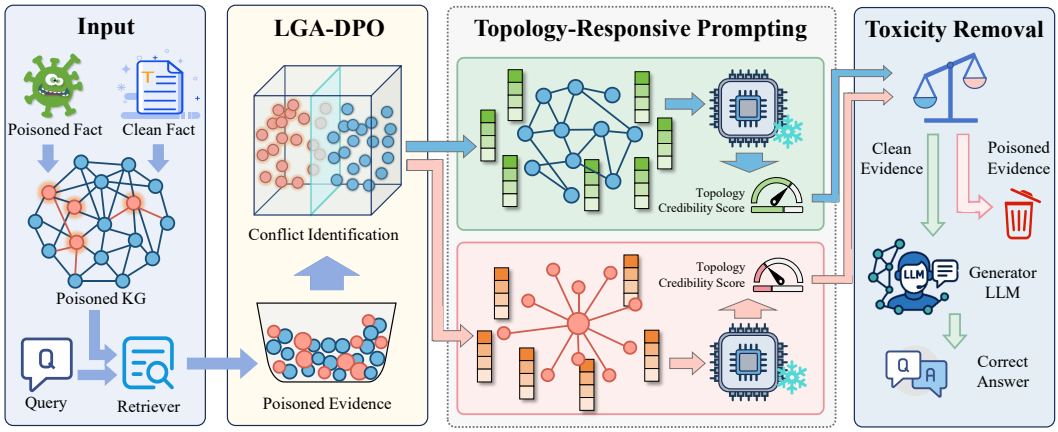


Figure 1: Overall framework of TDP. LGA-DPO identifies conflicts by exploiting the separability of contradictory content in semantic space. The conflict subgraphs are then scored for credibility based on GNN-based Topology-Responsive Prompting (TRP). The poisoned subgraph is filtered out, and evidence from the clean subgraph is fed into the generator LLM to produce the final answer.

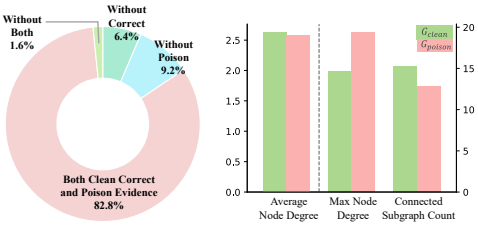
sources that the GraphRAG system may crawl (such as arXiv, Reddit, etc.). Once such poisoned content is retrieved, the system may generate the attacker-specified answer. We expect to add a defense mechanism during retrieval to remove poisoned evidence, thereby generating correct answers.

Capability and goal of attackers Attackers can inject poisoned text into public sources to mislead the GraphRAG. The number of injected texts is limited since abnormal large-scale injection is more likely to be detected [Zou et al., 2025, Zhang et al., 2026]. However, attackers have no access to the database or the generator weights. They can only interact with the system through queries, but cannot directly read, modify, or delete the original database. Attackers aim to maximally steer GraphRAG toward the incorrect answer.

Capability and goal of defenders Defenders can directly access the database, and enhance the GraphRAG pipeline by inserting a validation layer during inference, without modifying the underlying knowledge storage. Our goal is to ensure that the system output always aligns with truthful evidence, even if the retriever returns mixed results containing both clean and poisoned content.

3.2 Conflict Identification via High-Dimensional Representation Separation

When GraphRAG retrieves evidence from a poisoned database, the returned results often contain both clean correct and poisoned evidence, as shown in Figure 2a. Since the content in the knowledge base is often domain-specific or newly updated, it is difficult to directly identify clean and poisoned data. However, clean and poisoned knowledge exhibit clear logical conflicts as they respond to the same query with different answers. Research works such as Sentence-Bert point out that contradictory statements tend to be far apart in semantic space [Reimers and Gurevych, 2019, Gunel et al., 2021] and can be separated by an approximate hyperplane. We propose **L**atent **G**eometry-**A**ligning **D**irect **P**reference **O**ptimization (LGA-DPO) for conflict identification. LGA-DPO leverages the expressive representation space of LLMs to map different narratives under the same query into separable geometric regions, and increases the distance between the two conflicting evidence groups, thereby obtaining a stable and clear conflict division. Formally, given a query q and a retrieved evidence set $C = \{C_1, \ldots, C_k\}$, the conflict



(a) Inclusion of poisoned and correct data in the retrieved results.

(b) Topological attributes of clean and poisoned subgraphs.

Figure 2: Statistical analysis of retrieved evidence and subgraphs.

identifier outputs the conflicting sets (C', C''):

$$\text{Conflict} = \{q, C', C''\}, \quad C' \subseteq C, \quad C'' \subseteq C, \tag{1}$$

where C' and C'' support mutually exclusive conclusions for q, *i.e.*,

$$\text{Ans}(q, C') \neq \text{Ans}(q, C''), \quad C' \cap C'' = \emptyset. \tag{2}$$

We extract representation of each evidence under the query context. Denote the pretrained language model as f_θ, for each evidence C_i, we obtain an embedding $h_i \in \mathbb{R}^d$:

$$h_i = \text{Pool}\left(f_\theta([q; C_i])\right), \quad i = 1, \ldots, k. \tag{3}$$

For any candidate partition $y = \{C', C''\}$, the centroids of the two groups are:

$$\mu' = \frac{1}{|C'|} \sum_{C_i \in C'} h_i, \quad \mu'' = \frac{1}{|C''|} \sum_{C_i \in C''} h_i. \tag{4}$$

If C' and C'' correspond to two exclusive narratives, their centroids should be far apart. Meanwhile, evidence within the same group should be semantically compact to avoid spurious separation. We thus define the geometric consistency score as:

$$R_{\text{geo}}(y) = \|\mu' - \mu''\|_2^2 - \alpha \left(\frac{1}{|C'|} \sum_{C_i \in C'} \|h_i - \mu'\|_2^2 + \frac{1}{|C''|} \sum_{C_i \in C''} \|h_i - \mu''\|_2^2 \right), \tag{5}$$

where α controls the within-group compactness.

During training, for the input $x = \text{Prompt}(q, C)$, we sample multiple candidate partitions $\{y^{(j)}\}$ and induce a preference pair using R_{geo}:

$$y^+ = \arg\max_{y^{(j)}} R_{\text{geo}}\big(y^{(j)}\big), \quad y^- = \arg\min_{y^{(j)}} R_{\text{geo}}\big(y^{(j)}\big), \tag{6}$$

and then optimize a policy model $\pi_\theta(y \mid x)$ using DPO, with a frozen reference model $\pi_{\text{ref}}(y \mid x)$ for stabilization. The differential term and the corresponding training objective are defined as:

$$\Delta(x) = \Big(\log \pi_\theta(y^+ \mid x) - \log \pi_\theta(y^- \mid x) \Big) - \Big(\log \pi_{\text{ref}}(y^+ \mid x) - \log \pi_{\text{ref}}(y^- \mid x) \Big), \tag{7}$$

$$\mathcal{L}_{\text{LGA-DPO}} = -\mathbb{E}_{(x, y^+, y^-)} \left[\log \sigma\big(\Delta(x)\big) \right], \tag{8}$$

encouraging the model to separate conflicting evidence groups in high-dimensional space and yield a stable partition (C', C'') that serves as structured input for downstream defense modules.

3.3 Toxicity Identification Based on Graph Topological Structure

After identifying the conflicting groups (C', C''), we further determine which subgraph is poisoned. Due to the unreliability of the knowledge base, text-level cues are not trustworthy. Moreover, GraphRAG's structured extraction breaks sentence-level similarity patterns. To address these issues, we evaluate evidence reliability from the perspective of logical connectivity among facts.

Since the content in the conflict groups is limited, it cannot well capture how the evidence connects to other entities in the KG. Therefore, TDP performs a re-retrieval in the KG using C' and C'', respectively, and denotes the resulting neighborhood-aware evidence subgraphs as G' and G''. Denote the poisoned and clean subgraphs as G_{poison} and G_{clean}, our task is to correctly predict their labels y_{clean} and y_{poison}. As shown in Figure 2b, G_{poison} and G_{clean} exhibit distinct topological patterns. Compared with G_{clean}, G_{poison} has a similar *Average Degree* but higher *Max Node Degree* and fewer *Connected Subgraphs*. It shows a stronger tendency toward a star-shaped structure. This structural discrepancy does not depend on a specific dataset or poisoning method, but is the result of the attacker's trade-off under a limited poisoning text amount. Denote the retrieved connected evidence subgraph as $G = (V, E)$, we find that the entities in G naturally fall into two categories: *query-facing entities* and *background entities*. Query-facing entities directly attach to the query anchors and provide clear support for the answer. Background entities are other content that is highly relevant to the query. Although they do not provide a direct answer, they often provide corresponding background facts for the above knowledge. We denote these two kinds of entities as V^q and V^b, satisfying

$V^q \cup V^b = V,\ V^q \cap V^b = \varnothing$. Obviously, facts corresponding to V^q are more relevant to the query, thus are more likely to be retrieved and steer LLMs toward the target answer. V^b helps provide richer background knowledge and improves the credibility of the evidence in V^q. Attackers would ideally like the injected poisoned knowledge to be both strongly aligned with the query and well supported by background facts. However, fabricating such factual associations is constrained by the amount of injected texts. We model the attacker's poisoning behavior as a budget-allocation trade-off game over the query-anchor subgraph G, which allocates the injection amount between the two types of entities under a fixed total injected content. Taking the retrieved content and credibility as utility, and the amount of injected content as cost, the overall payoff is:

$$\max_{\mathcal{G}}\ U(\mathcal{G}; q) = \mathrm{Retr}(\mathcal{G}, q) + \mathrm{Conf}(\mathcal{G}) - \lambda\Big(\mathrm{Cost}_{\mathrm{Retr}}(\mathcal{G}, q) + \mathrm{Cost}_{\mathrm{conf}}(\mathcal{G})\Big), \tag{9}$$

where $\mathrm{Retr}(\mathcal{G}, q)$ and $\mathrm{Conf}(\mathcal{G})$ are the retrieval and credibility gain of the subgraph, $\mathrm{Cost}_{\mathrm{Retr}}(\mathcal{G}, q)$ and $\mathrm{Cost}_{\mathrm{conf}}(\mathcal{G})$ are the cost of injecting the corresponding knowledge, $0 < \lambda < 1$ controls the attacker's sensitivity to the text injection cost.

The retrieval gain is typically measured by the amount of query-facing content returned, which corresponds to the number of nodes in V^q. For the credibility gain, prior studies in fact-checking and information science [Graves, 2017, Suomalainen et al., 2025, Koliska and Roberts, 2025] suggest that information supported by repeated cross-validation paths is more trustworthy. Thus, we quantify the structural corroboration of the query-anchor connected subgraph by its cyclomatic number:

$$r(G) = |E(G)| - |V(G)| + 1, \tag{10}$$

which equals the dimension of the cycle space of G. Intuitively, a larger $r(G)$ indicates more independent closure constraints among retrieved facts and thus richer opportunities for cross-validation. Appendix F formalizes this point under a cycle-basis corroboration model and shows that posterior credibility increases monotonically with $r(G)$ when observed checks are sufficiently consistent. Accordingly, the two kinds of gain can be expressed as:

$$\mathrm{Retr}(G, q) = |V_q|, \quad \mathrm{Conf}(G) = r(G) = |E(G)| - |V(G)| + 1. \tag{11}$$

Note that node injection depends on inserting new triples, *i.e.*, adding corresponding edges into the subgraph. Let E_q denote the edges that connect query-facing entities to the query anchors, and let E_b denote the additional corroborative edges needed to create nontrivial cycles in the background structure. Since inserting a query-facing entity $v \in V_q$ only requires attaching it to the query anchors, its retrieval-oriented injection cost is linear in $|E_q|$. Therefore, the cost of adding edges is:

$$\mathrm{Cost}_{\mathrm{Retr}}(G, q) = |E_q| = |V_q|, \quad \mathrm{Cost}_{\mathrm{Conf}}(G) = |E_b|. \tag{12}$$

Since $E(G) = E_q \cup E_b$ and $|E_q| = |V_q|$, the objective simplifies to:

$$\begin{aligned} &\max_{\mathcal{G}}\ U(\mathcal{G}; q) = |V^q| + |E(\mathcal{G})| - |V(\mathcal{G})| + 1 - \lambda\big(|E^b| + |E^q|\big) \\ &= |E^q| + (1-\lambda)|E(\mathcal{G})| - |V^q| - |V^b| + 1 = (1-\lambda)|E(\mathcal{G})| - |V^b| + 1. \end{aligned} \tag{13}$$

This indicates that under a fixed poisoning cost, injecting background nodes decreases the overall payoff. Although the attacker still forges part of the background knowledge nodes due to marginal effects, its impact is much weaker than directly injecting query-facing entities. This reveals a fundamental structural imbalance in poisoning: **query-facing entity** nodes achieve stable retrieval gains at low cost by fabricating a single relation triple, whereas background nodes require forming new cycles to improve credibility and thus incur substantially higher coordination cost. As a result, under a limited amount of poisoned text, the attacker prioritizes injecting V^q entities to maximize attack performance, which makes the poisoned subgraph topologically closer to a star-shaped pattern with **central radiation** and **sparse peripheral verification**. Importantly, this topology feature arises as a budget-constrained equilibrium induced by the attacker's trade-off, rather than a property tied to any specific dataset or attack strategy.

Therefore, we exploit this topological discrepancy to distinguish poisoned and clean subgraphs without relying on the original semantics. We propose **T**opology-**R**esponsive **P**rompting (TRP), a light topology-responsive feature calibration method for poisoning defense. TRP does not require task-specific finetuning of the graph encoder. Instead, it injects a small set of learnable calibration vectors into the input feature space, so that the decision boundary actively aligns with the systematic difference between the mesh-like logical structure and the central-radiation structure.

Different from common graph classification, TRP does not evaluate a single evidence subgraph in isolation. Each decision takes a pair of candidate conflict subgraphs, where they correspond to clean evidence and poisoned evidence. For subgraph $G = (V, E) \in \{G', G''\}$, denote the adjacency matrix as $A \in \{0, 1\}^{N \times N}$ and the node feature matrix as $X \in \mathbb{R}^{N \times F}$. TRP learns a shared prompt vector $p \in \mathbb{R}^F$ and applies the feature bias to all nodes, yielding the prompted input features:

$$X^* = X + p^\top, \tag{14}$$

then feed (A, X^*) into a frozen graph encoder $f_\phi(\cdot)$ to obtain a subgraph representation, and use a linear scoring head to get the credibility score:

$$z = f_\phi(A, X^*), \quad s(G) = w^\top z. \tag{15}$$

This formulation transforms toxicity determination to a score comparison between two candidate subgraphs. Accordingly, the TRP prediction rule is:

$$\hat{G}_{\text{clean}} = \arg \max_{G \in \{G', G''\}} s(G), \quad \hat{G}_{\text{poison}} = \arg \min_{G \in \{G', G''\}} s(G). \tag{16}$$

Throughout both training and inference, the graph encoder remains frozen after pretraining. TRP optimizes only the prompt and scoring-head parameters $\Theta = \{p, w\}$. Therefore, TRP adapts to topology-based discrimination with low parameter overhead, and avoids high-cost end-to-end retraining when the dataset or poisoning distribution change.

In the training stage, we construct the ranking objective using pairwise supervision. Denote the training set as $\mathcal{D} = \{(G_{\text{clean}}, G_{\text{poison}})\}_{i=1}^M$, where each pair corresponds to a pair of clean and poisoned candidate. TRP learns Θ by minimizing pairwise softmax cross-entropy loss:

$$\min_\Theta \ \mathcal{L}_{\text{pair}}(\Theta) = \min_\Theta \sum_{i=1}^M -\log t_i, \quad t_i = \frac{\exp\left(s(G_{\text{clean}}^{(i)})\right)}{\exp\left(s(G_{\text{clean}}^{(i)})\right) + \exp\left(s(G_{\text{poison}}^{(i)})\right)}. \tag{17}$$

The scoring function for the i-th pair is:

$$s(G_i^+) = w^\top f_\phi\big(A_i^+, X_i^+ + \mathbf{1}p^\top\big), \quad s(G_i^-) = w^\top f_\phi\big(A_i^-, X_i^- + \mathbf{1}p^\top\big). \tag{18}$$

At inference time, TRP directly compares $s(G')$ and $s(G'')$, selects the higher-scoring subgraph as G_{clean}, and only forwards it to the generator, achieving structural interception of poisoned evidence and enabling robust reasoning.

4 Experiment

Our study focuses on the following research questions:

- RQ1: How does TDP perform compared with existing defenses across different datasets, attack methods, and GraphRAG frameworks?
- RQ2: How does the conflict identification model influence the overall defense performance?
- RQ3: How does the number of retrieved items affect the effectiveness of different defense methods?
- RQ4: What is the computational cost of TDP?

4.1 Experiment Setup

Datasets We evaluate our defense on three widely used benchmarks. 2WIKIMULTIHOPQA [Ho et al., 2020] is a large-scale QA dataset built on Wikipedia. MUSIQUE [Trivedi et al., 2022] focuses on multi-hop question answering. PUBMEDQA [Jin et al., 2019] represents a domain-specific benchmark in biomedicine. Notably, we pretrain TDP on another general QA dataset, HOTPOTQA [Yang et al., 2018], and then transfer it to these three datasets for defense *without additional training*.

Attack Methods We consider two representative poisoning attacks. GRAG-Poison [Liang et al., 2026] is a GraphRAG-specific attack that performs efficient poisoning by forging false relations. PoisonedRAG [Zou et al., 2025] is a widely used generic attack that injects poisoned passages in a simple and effective manner. In addition, the defense results against a broader set of attack methods are presented in Appendix A.

GraphRAG Framework We evaluate TDP under two representative GraphRAG frameworks. GRAPHRAG [Edge et al., 2024] organizes the corpus into a KG and retrieves query-relevant subgraphs for reasoning and generation. HIPPORAG [Gutierrez et al., 2024] introduces a memory-centric retrieval mechanism that recognizes and exploits connections among newly acquired knowledge.

Metrics We evaluate the final output of GraphRAG using GPT-5-mini [OpenAI, 2026], and then compute the answer accuracy (ACC) and the attack success rate (ASR). Since GraphRAG itself cannot correctly answer all questions, the performance defects of this retrieval framework may affect our analysis of defense methods. To further measure the defense effect, we also compute the proportion of instances that are successfully attacked without defense and are successfully identified, denoted as DASR, and the proportion of instances that can be answered correctly by the original GraphRAG and remain correct after poisoning and defense, denoted as $\mathrm{ACC}_{\mathrm{expect}}$.

Baselines We compare three representative defense baselines. RAGuard [Cheng et al., 2025] targets corpus poisoning by widening retrieval and filtering suspicious passages using perplexity anomalies and abnormally high query-text similarity. ReliabilityRAG [Shen et al., 2025] improves robustness by estimating document reliability and selecting a mutually consistent subset of retrieved evidence for generation. FaithfulRAG [Zhang et al., 2025] enhances faithfulness by explicitly identifying and mitigating conflicts between the retrieved context and the produced answer.

4.2 Results

4.2.1 Main Results (RQ1)

Table 2: Results of defended GraphRAG under GRAG-Poison [Liang et al., 2026]. *Poison ASR* denotes the ASR without defense. All results are reported in percentage. The best results are in **bold** and the second best results are underlined.

Dataset	Poison ASR	Defense Method	Defended Poisoned GraphRAG			
			ASR ↓	ACC ↑	DASR ↑	$\mathrm{ACC}_{\mathrm{expect}}$ ↑
GraphRAG						
2Wiki MultihopQA	72.0	RAGuard	68.2±0.70	23.7±0.33	6.2±0.11	28.3±0.16
		ReliabilityRAG	53.3±0.64	25.8±0.27	26.5±0.47	42.1±0.36
		FaithfulRAG	64.5±0.64	27.5±0.13	15.3±0.23	39.4±0.14
		TDP (ours)	**27.0±0.32**	**65.5±0.02**	**60.4±0.33**	**88.3±0.26**
MuSiQue	69.5	RAGuard	60.3±0.39	15.4±0.24	14.4±0.42	28.5±0.41
		ReliabilityRAG	53.4±0.83	19.8±0.14	27.2±0.57	35.9±0.45
		FaithfulRAG	53.1±0.61	21.5±0.14	22.6±0.50	38.3±0.38
		TDP (ours)	**30.5±0.33**	**42.4±0.27**	**59.2±0.41**	**74.8±0.49**
PubMedQA	73.5	RAGuard	58.7±0.75	28.9±0.21	19.3±0.47	40.6±0.18
		ReliabilityRAG	52.7±0.49	31.4±0.07	24.9±0.33	46.7±0.12
		FaithfulRAG	31.0±0.30	38.5±0.06	54.8±0.43	69.6±0.20
		TDP (ours)	**29.4±0.20**	**45.5±0.11**	**68.6±0.56**	**70.3±0.46**
HippoRAG						
2Wiki-MultihopQA	72.6	RAGuard	68.4±0.30	19.8±0.35	6.2±0.17	27.4±0.31
		ReliabilityRAG	66.1±0.32	21.2±0.16	10.6±0.44	30.7±0.18
		FaithfulRAG	66.9±0.83	26.3±0.19	5.9±0.12	38.2±0.36
		TDP (ours)	**27.2±0.32**	**54.4±0.21**	**63.7±0.42**	**76.8±0.56**
MuSiQue	65.8	RAGuard	64.3±0.55	11.6±0.32	2.5±0.25	17.3±0.17
		ReliabilityRAG	61.2±0.35	18.4±0.10	23.7±0.56	27.6±0.36
		FaithfulRAG	56.5±0.32	17.2±0.29	17.5±0.44	26.2±0.44
		TDP (ours)	**24.0±0.42**	**49.8±0.07**	**65.9±0.79**	**74.3±0.52**
PubMedQA	58.7	RAGuard	53.2±0.63	20.6±0.27	8.6±0.25	26.4±0.22
		ReliabilityRAG	50.7±0.60	24.3±0.12	13.1±0.56	35.8±0.31
		FaithfulRAG	27.5±0.26	25.8±0.09	57.2±0.39	70.5±0.44
		TDP (ours)	**21.3±0.23**	**63.0±0.30**	**62.9±0.40**	**91.4±0.35**

Table 2 reports the defense results against GRAG-POISON, the currently strongest GraphRAG poisoning attack method. Results against the weaker POISONEDRAG attack are reported in Table 5. Our TDP achieves the lowest ASR and the highest accuracy on all datasets and GraphRAG frameworks, substantially outperforming all baselines. It effectively defends attacks on large-scale benchmarks, multi-hop reasoning, and domain-specific tasks. This effect benefits from our finding of the attacker's game outcome on the injection budget. TDP exploits the stable topological patterns induced by this strategic preference and filters poisoned evidence accordingly.

In addition to ACC and ASR, we report DASR and $\text{ACC}_{\text{expect}}$ to reflect relative defense effectiveness under the imperfect base performance of GraphRAG. Some defenses successfully block attacks but also remove clean evidence. This can prevent correct generation and thus degrade GraphRAG performance. TDP achieves the best performance on both metrics, showing that our method can effectively identify attack text while retaining clean evidence, maintaining GraphRAG performance while improving security.

4.2.2 Conflict Identification based on Different LLMs (RQ2)

Conflict identification directly affects the classification performance of TRP. If the conflict sets are partitioned incorrectly, TRP cannot detect poisoned evidence. We investigate the performance of commonly used models for conflict localization. Among them, the smaller QWEN3-8B (used in the main experiments) and LLAMA3.1-8B are finetuned using LGA-DPO. For other closed-source LLMs, including the strongest versions and lower-cost versions, we directly prompt them to identify conflicting items in the retrieved content. $\text{ACC}_{\text{conflict}}$ denotes the accuracy of grouping conflicting content. We consider it successful only when all poisoned content is placed into the same group. If the poisoned content is not included, or it is assigned to both groups at the same time, we regard it as incorrect. ALL-RIGHT is a hypothetical oracle that perfectly identifies the conflict partition. We use it to isolate and evaluate TRP's ability to discriminate the resulting conflicting subgraphs, given an error-free conflict identification stage.

As shown in Table 3, models with better conflict classification improve the final defense performance. Notably, our method finetunes a small model and substantially outperforms weaker closed-source LLMs, approaching the strongest LLMs. This improvement comes from exploiting the separability of contradictory information in semantic space, which enables accurate and low-cost conflict localization and ensures that downstream topology modules receive reliable structured inputs.

Table 3: Performance of TDP under GraphRAG on 2WikiMultihopQA with different conflict identification models. Closed-source LLMs are not fine-tuned.

Model	$\text{ACC}_{conflict}$ ↑	ASR ↓	ACC ↑
Qwen3-8B	82.1	27.0	65.5
Llama3.1-8B	76.3	34.1	60.2
Claude-4.5	83.6	25.4	67.2
GPT-4o	72.8	39.7	56.1
GPT-5-mini	75.9	34.6	62.3
GPT-5.2	84.5	22.7	70.1
ALL-right	-	18.3	75.6

4.2.3 Numbers of Retrieved Content (RQ3)

The number of retrieved items (top-k) in a GraphRAG framework has a large impact on ASR and answer accuracy. Figure 3 shows the defense performance under GraphRAG on 2WikiMultihopQA. Retrieving too few items may miss necessary clean evidence, which makes the system easier to mislead by poisoned content. In contrast, retrieving too many items increases noise and can reduce accuracy. When the retrieved set contains both clean and poisoned evidence, TDP maintains relatively stable DASR and $\text{ACC}_{\text{expect}}$ across different top-k settings, which suggests that TDP is not sensitive to the retrieval size and thus scales well to large datasets.

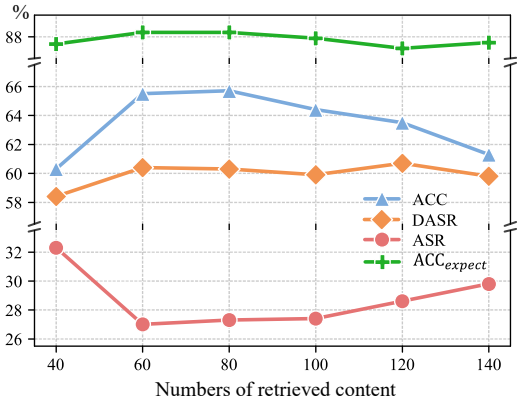


Figure 3: Defense performance with different numbers of retrieved content.

4.2.4 Cost of Defense Methods (RQ4)

Table 4 reports the additional token and time cost overhead introduced by different defenses compared with the original GraphRAG. Although RAGuard does not require extra LLM-based inference, its performance is relatively poor. Our TDP achieves the highest accuracy with the lowest ASR while maintaining low overhead. This efficiency benefits from capturing the topological features caused by the attacker's strategy preference, which removes the need for expensive semantic reasoning via repeated LLM calls.

Table 4: Average defense cost for one query under GraphRAG on 2WikiMultihop.

Methods	Time ↓	Token ↓	ASR ↓	ACC ↑
RAGuard	< 1s	-	68.2%	23.7%
ReliabilityRAG	13.4s	2189	53.3%	25.8%
FaithfulRAG	18.3s	3728	64.5%	27.5%
TDP (ours)	< 5s	814	27.0%	65.5%

5 Conclusion

This paper proposes Topology-based Defense against Poisoning Attack on GraphRAG (TDP), a lightweight defense that filters poisoned evidence using structure. We analyze the attacker's budget-constrained trade-off and show a strategy-induced topological gap: forged subgraphs concentrate links from query anchors to poisoned entities while remaining sparse among poisoned entities. TDP exploits this gap by first identifying conflicting evidence groups via an LGA-DPO model and then scoring credibility with TRP, which injects a prompt vector into node features and uses a frozen GNN to rank evidence. The retained evidence supports both answer generation and auditing, avoiding costly text-level LLM reasoning. Experiments demonstrate consistent gains in robustness and efficiency, with strong generalization across retrieval frameworks, datasets, and attack methods.

Acknowledgments

This work is supported by the Noncommunicable Chronic Diseases-National Science and Technology Major Project No. 2023ZD0501800, National Natural Science Foundation of China No. 62406057, the Fundamental Research Funds for the Central Universities No. ZYGX2025XJ042, and the Sichuan Science and Technology Program under Grant No. 2024ZDZX0011. The authors declare no competing interests.

References

Zhiyuan Chang, Mingyang Li, Xiaojun Jia, Junjie Wang, Yuekai Huang, Ziyou Jiang, Yang Liu, and Qing Wang. One shot dominance: Knowledge poisoning attack on retrieval-augmented generation systems. In *Findings of EMNLP*, pages 18811–18825, 2025.

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 *WWW*, pages 2308–2319, 2026.

Zirui Cheng, Jikai Sun, Anjun Gao, Yueyang Quan, Zhuqing Liu, Xiaohua Hu, and Minghong Fang. Secure retrieval-augmented generation against poisoning attacks. In *IEEE Big Data*, pages 1799–1806, 2025.

Gerard Debreu. A social equilibrium existence theorem. *Proceedings of the national academy of sciences*, 38(10):886–893, 1952.

Kaize Ding, Jundong Li, Rohit Bhanushali, and Huan Liu. Deep anomaly detection on attributed networks. In *SDM*, pages 594–602, 2019.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization. *CoRR*, abs/2404.16130, 2024.

Lucas Graves. Anatomy of a fact check: Objective practice and the contested epistemology of fact checking. *Communication, culture & critique*, 10(3):518–537, 2017.

Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. Supervised contrastive learning for pre-trained language model fine-tuning. In *ICLR*, 2021.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. In *NeurIPS*, volume 37, pages 59532–59569, 2024.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *NeurIPS*, pages 1024–1034, 2017.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *COLING*, pages 6609–6625, 2020.

Yang Jiao, Xiaodong Wang, and Kai Yang. Pr-attack: Coordinated prompt-rag attacks on retrieval-augmented generation in large language models via bilevel optimization. In *SIGIR*, pages 656–667, 2025.

Ming Jin, Yixin Liu, Yu Zheng, Lianhua Chi, Yuan-Fang Li, and Shirui Pan. ANEMONE: graph anomaly detection with multi-scale contrastive learning. In *CIKM*, pages 3122–3126, 2021.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In *EMNLP-IJCNLP*, pages 2567–2577, 2019.

Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *KDD*, pages 66–74, 2020.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Michael Koliska and Jessica Roberts. Epistemology of fact checking: an examination of practices and beliefs of fact checkers around the world. *Digital Journalism*, 13(3):500–520, 2025.

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

Junchen Li, Rongzheng Wang, Yihong Huang, Qizhi Chen, Jiasheng Zhang, and Shuang Liang. NeuroPath: Neurobiology-inspired path tracking and reflection for semantically coherent retrieval. In *NeurIPS*, 2025.

Jiacheng Liang, Yuhui Wang, Changjiang Li, Tanqiu Jiang, Rongyi Zhu, Neil Gong, and Ting Wang. GraphRAG under fire. In *IEEE SP*, pages 195–212, 2026.

Yixin Liu, Zhao Li, Shirui Pan, Chen Gong, Chuan Zhou, and George Karypis. Anomaly detection on attributed networks via contrastive self-supervised learning. *TNNLS*, 33(6):2378–2392, 2022.

MediaWiki. ratelimit, 2022. URL `https://www.mediawiki.org/wiki/Manual%3A%24wgRateLimits`.

OpenAI. OpenAI GPT-5 system card. *CoRR*, abs/2601.03267, 2026. doi: 10.48550/ARXIV.2601.03267. URL `https://doi.org/10.48550/arXiv.2601.03267`.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese bert-networks. In *EMNLP*, pages 3982–3992, 2019.

Zeyu Shen, Basileal Imana, Tong Wu, Chong Xiang, Prateek Mittal, and Aleksandra Korolova. ReliabilityRAG: Effective and provably robust defense for rag-based web-search. In *NeurIPS*, 2025.

Kari Suomalainen, Nooa Nykänen, Hannele Seeck, Youna Kim, and Ella McPherson. Fact-checking in journalism: An epistemological framework. *Journalism Studies*, 26(10):1129 – 1149, 2025.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554, 2022.

Yiteng Tu, Weihang Su, Yujia Zhou, Yiqun Liu, and Qingyao Ai. Robust fine-tuning for retrieval augmented generation against retrieval defects. In *SIGIR*, pages 1272–1282, 2025.

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

Meng Xi, Sihan Lv, Yechen Jin, Guanjie Cheng, Naibo Wang, Ying Li, and Jianwei Yin. RIPRAG: hack a black-box retrieval-augmented generation question-answering system with reinforcement learning. In *Findings of ACL*, pages 16882–16902, 2026.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, pages 2369–2380, 2018.

Gustavo Ye, Terence Liu, Rangehow, and Chasing. Nano-graphrag, 2024. URL `https://github.com/gusye1234/nano-graphrag`.

Baolei Zhang, Yuxi Chen, Zhuqing Liu, Lihai Nie, Tong Li, Zheli Liu, and Minghong Fang. Practical poisoning attacks against retrieval-augmented generation. In *SACMAT*, pages 33–44, 2026.

Qinggang Zhang, Zhishang Xiang, Yilin Xiao, Le Wang, Junhui Li, Xinrun Wang, and Jinsong Su. FaithfulRAG: Fact-level conflict modeling for context-faithful retrieval-augmented generation. In *ACL*, pages 21863–21882, 2025.

Xiang Zhang and Marinka Zitnik. GNNGuard: Defending graph neural networks against adversarial attacks. In *NeurIPS*, 2020.

Tianzhe Zhao, Jiaoyan Chen, Yanchi Ru, Haiping Zhu, Nan Hu, Jun Liu, and Qika Lin. Exploring knowledge poisoning attacks to retrieval-augmented generation. *Inf. Fusion*, 127:103900, 2026.

Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. PoisonedRAG: Knowledge corruption attacks to retrieval-augmented generation of large language models. In *USENIX Security*, pages 3827–3844, 2025.

A Defense performance against additional poison attacks

Table 5: Results of defended GraphRAG under PoisonedRAG [Zou et al., 2025]. *Poison ASR* denotes the ASR without defense. All results are reported in percentage. The best results are in **bold** and the second best results are underlined.

Dataset	Poison ASR	Defense Method	Defended Poisoned GraphRAG			
			ASR ↓	ACC ↑	DASR ↑	$\text{ACC}_{\text{expect}}$ ↑
GraphRAG						
2Wiki MultihopQA	30.3	RAGuard	28.7±0.64	40.4±0.78	7.2±0.45	55.0±0.83
		ReliabilityRAG	22.4±0.52	41.5±0.69	28.0±0.74	56.5±0.77
		FaithfulRAG	27.1±0.59	42.3±0.73	17.0±0.66	57.6±0.80
		TDP (ours)	**11.4±0.37**	**63.2±0.58**	**64.4±0.49**	**86.0±0.64**
MuSiQue	25.2	RAGuard	21.9±0.71	32.3±0.82	16.2±0.57	53.0±0.76
		ReliabilityRAG	19.4±0.65	33.2±0.79	29.0±0.88	54.4±0.83
		FaithfulRAG	19.3±0.61	33.9±0.74	24.0±0.69	55.6±0.78
		TDP (ours)	**11.1±0.40**	**51.2±0.55**	**62.8±0.44**	**84.0±0.61**
PubMedQA	28.9	RAGuard	23.1±0.67	39.5±0.86	21.0±0.72	54.5±0.81
		ReliabilityRAG	20.7±0.63	40.6±0.84	27.0±0.79	56.0±0.85
		FaithfulRAG	16.0±0.58	41.3±0.80	57.0±0.62	57.0±0.87
		TDP (ours)	**11.6±0.39**	**62.7±0.60**	**71.2±0.51**	**86.5±0.68**
HippoRAG						
2Wiki-MultihopQA	55.4	RAGuard	52.2±0.72	35.8±0.89	8.0±0.50	51.4±0.83
		ReliabilityRAG	50.4±0.69	36.9±0.86	12.0±0.63	53.0±0.80
		FaithfulRAG	51.1±0.70	37.6±0.84	7.0±0.47	54.0±0.82
		TDP (ours)	**20.8±0.44**	**61.4±0.62**	**66.8±0.53**	**80.5±0.65**
MuSiQue	53.5	RAGuard	52.3±0.74	29.7±0.90	4.0±0.35	54.5±0.79
		ReliabilityRAG	49.8±0.71	30.5±0.88	26.0±0.81	56.0±0.82
		FaithfulRAG	45.9±0.68	31.1±0.86	20.0±0.76	57.1±0.84
		TDP (ours)	**19.5±0.42**	**57.4±0.58**	**68.5±0.49**	**79.0±0.66**
PubMedQA	50.7	RAGuard	45.9±0.69	33.3±0.88	11.0±0.61	48.5±0.77
		ReliabilityRAG	43.8±0.66	34.3±0.86	15.0±0.67	50.0±0.80
		FaithfulRAG	24.2±0.57	35.0±0.84	60.0±0.60	51.0±0.82
		TDP (ours)	**18.4±0.41**	**68.5±0.61**	**66.0±0.52**	**88.0±0.63**

Table 5 reports the defense results under the PoisonedRAG attack. Compared with GRAG-Poison, PoisonedRAG is weaker as an attack, which naturally leads to higher accuracy, while the changes in DASR and $\text{ACC}_{\text{expect}}$ remain relatively small. We further evaluate TDP against a broader set of RAG poisoning attacks, as shown in Table 6. TDP consistently reduces ASR by a large margin while maintaining a high DASR, demonstrating that its effectiveness does not depend on any specific attack method. Instead, TDP works by identifying and filtering the topological structures induced by attacker strategies during poisoning. We omit detailed analysis here and refer to Section 4.2.1.

Table 6: Defense performance of TDP against several attack methods on HippoRAG

Dataset	Attack Methods	Poison ASR	ASR ↓	DASR ↑
2Wiki MultihopQA	CorruptRAG [Zhang et al., 2026]	71.4%	21.1%	73.2%
	One Shot Dominance [Chang et al., 2025]	74.2%	23.5%	70.7%
	KEPo [Chen et al., 2026]	77.1%	25.0%	70.6%
MuSiQue	CorruptRAG [Zhang et al., 2026]	66.3%	21.9%	69.4%
	One Shot Dominance [Chang et al., 2025]	69.8%	24.1%	68.0%
	KEPo [Chen et al., 2026]	72.5%	24.7%	67.2%

B Evaluation on Sparse-Fact Evidence

We further evaluate the reliability of TDP for sparse-facts, where valid evidence has limited local support in the KG. This setting tests whether TDP can still distinguish poisoned evidence from clean evidence when the clean subgraph itself is not densely connected. Specifically, we construct a sparse-fact subset on MuSiQue by using node degree as a proxy for local knowledge density. We select instances whose key evidence nodes fall into the bottom 10% degree percentile, and then attack these instances with GRAG-Poison.

Table 7: Defense performance against GRAG-Poison on sparse-fact evidence from MuSiQue.

Dataset	Poison ASR	ASR ↓	DASR ↑
Raw MuSiQue	69.5%	30.5%	59.2%
Sparse-Facts in MuSiQue	75.4%	33.6%	58.8%

As shown in Table 7, TDP consistently reduces ASR by a large margin on both settings. Meanwhile, DASR remains close to the full-data setting, indicating that the defense remains effective even when clean evidence has limited structural support. These results further support that TDP does not rely on a simple dense-versus-sparse heuristic. Instead, it captures the relative topological discrepancy between clean and poisoned candidate subgraphs, which is induced by the attacker's poisoning strategy preference.

C Ablation Study

Figure 4 shows the results of the ablation study. K-means-TDP removes our LGA-DPO training strategy. It clusters retrieved evidence with k-means and selects the two clusters whose centroids are farthest apart as the conflict sets. This method fails to reliably separate contradictory evidence, providing almost no defense benefit. GNN-TDP removes the prompt vectors in TRP and directly uses a GNN for toxicity identification. It is highly dependent on the dataset and generalizes poorly. Overall, the results highlight the necessity of both components in TDP.

Table 8: Versions of LLMs.

LLM	Version
Llama3.1-8B	Llama-3.1-8B-Instruct
Claude-4.5	Claude-haiku-4.5
GPT-4o	GPT-4o-2024-05-13
GPT-5-mini	GPT-5-mini-2025-08-07
GPT-5.2	GPT-5.2-2025-12-11

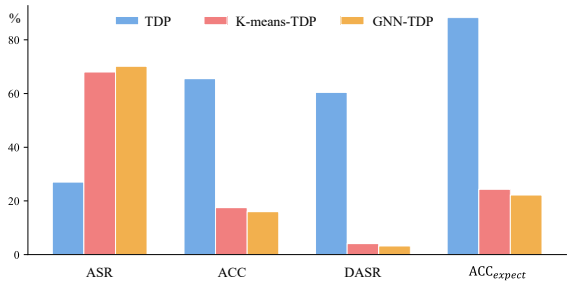


Figure 4: Ablation study of TDP.

D Experimental Details

Experiments are conducted on a local server equipped with RTX 4090 NVIDIA GPUs, Intel Xeon Gold 6342 and 64 GB system RAM. The software environment uses Ubuntu 22.04, Python 3.12.3, CUDA 12.4, and PyTorch 2.5.1. All reported results are averaged over three runs. Our code is available at https://github.com/kukudadasd/TDP.

D.1 LLM versions

Some LLMs are referred to by abbreviations in the main text, whose full versions are listed in Table 8.

D.2 GNN versions

TRP uses GIN [Xu et al., 2019] as the graph encoder to distinguish poisoned subgraphs from clean subgraphs by modeling their topological difference. Concretely, we use a 4-layer GIN with hidden dimension 256, jumping knowledge set to sum, dropout 0.3 and mean pooling as graph readout. The final graph representation concatenates the pooled graph embedding and the center-node embedding, and a classification head outputs a scalar score. Table 9 compares several GNN variants in TRP. GCN-style normalized averaging tends to smooth node representations and may wash out subtle structural discrepancies, while GraphSAGE is more oriented toward scalable neighborhood sampling than precise structural discrimination, and GAT mainly emphasizes the relative importance of neighbors rather than the overall organization of neighborhood patterns.

Table 9: Performance of different GNN backbones in TRP. Lower ASR and higher DASR indicate better defense performance.

GNN Backbone	**ASR**	**DASR**
GCN [Kipf and Welling, 2017]	31.2%	56.7%
GraphSAGE [Hamilton et al., 2017]	29.8%	58.1%
GAT [Velickovic et al., 2018]	30.3%	57.6%
GIN [Xu et al., 2019]	27.0%	60.4%

GIN iteratively updates each node by aggregating information from its neighbors and then transforming the aggregated signal with a multilayer perceptron. Instead of commonly used normalized averaging, GIN applies sum aggregation, which preserves neighborhood composition more faithfully and makes the model more sensitive to structural variation in local graph patterns. Since our defense relies on identifying topology-aware credibility differences rather than only semantic inconsistency, GIN is the most suitable choice for TRP in the current GraphRAG poisoning-defense task.

D.3 GraphRAG Frameworks

In our experiments, we employed two representative graph-based retrieval-augmented generation frameworks, the GraphRAG and HippoRAG. Besides necessary modifications to some inputs and outputs to insert TDP, we keep all other settings at their defaults. Due to the substantial cost of deploying and running full GraphRAG, we use its open-source simplified implementation, Nano-GraphRAG [Ye et al., 2024], in this paper. Nano-GraphRAG implements the search different from the original. The original uses a map-reduce-like style to fill the communities into context, while Nano-GraphRAG replaces this with a top-k ranking approach. In a production-scale deployment, this may incur performance degradation, but on the smaller datasets used in research, the two approaches exhibit comparable performance.

D.4 LGA-DPO Inputs, Candidate Generation, and Optimization

Model input and output. The conflict identifier is Qwen3-8B fine-tuned through LGA-DPO. Its input is $x = \mathrm{Prompt}(q, C)$, containing the query and the retrieved evidence without clean/poison labels. Its output $y = \{C', C''\}$ specifies two evidence groups supporting mutually exclusive conclusions.

Offline generation of candidate partitions. HotpotQA provides gold supporting evidence for its questions. Poison provenance is retained when constructing attack samples. For each training instance, the gold supporting evidence and the injected poisoned evidence serve as anchors for the two narratives and are fixed in different groups. A frozen Qwen3-8B-Instruct model assigns the remaining retrieved chunks according to their semantic support relations. This guided sampling avoids exhaustively enumerating the exponentially large space of binary partitions. Each valid candidate $y^{(j)} = \{C'^{(j)}, C''^{(j)}\}$ must have disjoint groups and retain the corresponding gold and poisoned anchors in different groups. Group order is ignored when deduplicating: $\{C', C''\}$ and $\{C'', C'\}$ represent the same partition. The geometric score R_{geo} is evaluated for valid candidates, and the highest- and lowest-scoring partitions form (y^+, y^-).

Policy optimization and inference. The policy π_θ and frozen reference π_{ref} are initialized from the same Qwen3-8B-Instruct checkpoint. Only π_θ is updated using Eqs. (7)–(8). Gold evidence and poison provenance constrain the offline candidate space; they are not supplied as input labels to the policy. At inference, the trained policy receives only (q, C) and outputs the partition through a single generation.

D.5 Evidence-to-Subgraph Mapping and Native Re-Retrieval

The defense operates on KG-based RAG systems that maintain a correspondence between corpus passages and extracted relational facts. During indexing, entity–relation triples are extracted from corpus texts and incorporated into the KG. At query time, the underlying retrieval system returns relevant textual evidence using its native fact-retrieval procedures. LGA-DPO partitions the initially retrieved chunks into two conflicting groups, C' and C''. The chunks within each group are concatenated to form a group-level retrieval query. Each query is passed to the underlying GraphRAG retriever a second time. The nodes and edges associated with the returned passages are then collected through the passage-to-triple correspondences maintained during indexing, yielding the candidate graphs G' and G''.

Algorithm 1 details the evidence-to-subgraph mapping and native re-retrieval procedure. Let R denote the underlying GraphRAG system, $R.\,\mathrm{Retrieve}$ its native text retrieval operation, and $R.\,\mathrm{GetSubgraph}$ the collection of KG nodes and edges associated with retrieved passages.

Algorithm 1 Native Re-Retrieval for Candidate Fact Subgraphs

Input Query q, initially retrieved chunks C, and system R.
Output Candidate fact graphs G' and G''.

1. $(C', C'') \leftarrow \text{LGA-DPO}(q, C)$.
2. $q' \leftarrow \mathrm{Concat}(C')$, $q'' \leftarrow \mathrm{Concat}(C'')$.
3. $T' \leftarrow R.\,\mathrm{Retrieve}(q')$.
4. $T'' \leftarrow R.\,\mathrm{Retrieve}(q'')$.
5. $G' \leftarrow R.\,\mathrm{GetSubgraph}(T')$.
6. $G'' \leftarrow R.\,\mathrm{GetSubgraph}(T'')$.
7. **Return** (G', G'').

Here, T' and T'' are the texts returned by the second-stage retrieval.

E Failure Modes and Technical Limitations

Missing clean evidence. When retrieval returns only poisoned content, filtering cannot supply the missing clean evidence. This is an upstream retrieval failure, for example under an insufficient retrieval budget, rather than a problem that can be resolved by selecting between the returned candidates. The defense relies on the retriever exposing useful benign evidence. Likewise, its binary comparison presupposes meaningful competing evidence groups; a missing benign narrative cannot be created by topology ranking.

Imperfect conflict separation. If poisoned chunks are split across both conflict groups, both re-retrieved graphs may contain poisoned structures and residual poisoned evidence may reach the generator. In the analysis of such imperfect-separation cases, more than 90% still exhibit majority-level separation: most clean evidence and most poisoned evidence are assigned to different groups.

F Cycle-Basis Corroboration for Query-Anchor Connected Subgraphs

In Section 3.3, we model the credibility of the query-anchor connected subgraph G through its cyclomatic number

$$r(G) = |E(G)| - |V(G)| + 1, \tag{19}$$

i.e., the dimension of its cycle space. Here we provide a formal corroboration model under which posterior credibility increases monotonically with $r(G)$ when observed closure checks are sufficiently consistent.

Let $G = (V, E)$ be a connected undirected graph, and let $\mathcal{B}(G) = \{\beta_1, \ldots, \beta_{r(G)}\}$ be a cycle basis of G, where each β_j is a simple cycle. Let $H \in \{T, F\}$ denote the latent narrative state (truthful vs. forged). For each basis cycle β_j, define a binary closure-check variable $Y_j \in \{0, 1\}$, where $Y_j = 1$ means that the relations on β_j are mutually corroborative.

We assume conditional i.i.d. checks and discriminativity:

$$\{Y_j\}_{j=1}^{r(G)} \overset{\text{i.i.d.}}{\mid} H, \quad P(Y_j = 1 \mid H = T) = p_T, \quad P(Y_j = 1 \mid H = F) = p_F, \quad p_T > p_F. \tag{20}$$

Let

$$S = \sum_{j=1}^{r(G)} Y_j, \qquad \pi = P(H = T). \tag{21}$$

Then

$$(S \mid H = T) \sim \text{Binomial}(r(G), p_T), \qquad (S \mid H = F) \sim \text{Binomial}(r(G), p_F). \tag{22}$$

By Bayes' rule,

$$\log \frac{P(H = T \mid Y)}{P(H = F \mid Y)} = \log \frac{\pi}{1 - \pi} + S \log \frac{p_T}{p_F} + \big(r(G) - S\big) \log \frac{1 - p_T}{1 - p_F}. \tag{23}$$

With $s = S/r(G)$, define

$$\Delta(s) = s \log \frac{p_T}{p_F} + (1 - s) \log \frac{1 - p_T}{1 - p_F}. \tag{24}$$

Then

$$\log \frac{P(H = T \mid Y)}{P(H = F \mid Y)} = \log \frac{\pi}{1 - \pi} + r(G)\Delta(s). \tag{25}$$

Proposition. Suppose $p_T > p_F$. Then there exists a unique threshold $s^\star \in (0, 1)$ such that $\Delta(s) > 0$ if and only if $s > s^\star$. In particular, for any observation with $s > s^\star$,

$$\frac{\partial}{\partial r(G)} \log \frac{P(H = T \mid Y)}{P(H = F \mid Y)} = \Delta(s) > 0, \tag{26}$$

and hence $P(H = T \mid Y)$ increases strictly with $r(G)$.

Proof. Since $p_T > p_F$, we have $\log \frac{p_T}{p_F} > 0$ and $\log \frac{1-p_T}{1-p_F} < 0$. Therefore $\Delta(s)$ is an affine function of s with positive slope

$$\frac{\partial \Delta(s)}{\partial s} = \log \frac{p_T}{p_F} + \log \frac{1 - p_F}{1 - p_T} > 0. \tag{27}$$

Moreover,

$$\Delta(0) = \log \frac{1 - p_T}{1 - p_F} < 0, \qquad \Delta(1) = \log \frac{p_T}{p_F} > 0. \tag{28}$$

Thus, by monotonicity and continuity, there exists a unique threshold $s^\star \in (0, 1)$ satisfying $\Delta(s^\star) = 0$, namely

$$s^\star = \frac{\log \frac{1-p_F}{1-p_T}}{\log \frac{p_T}{p_F} + \log \frac{1-p_F}{1-p_T}}. \tag{29}$$

Hence $\Delta(s) > 0$ if and only if $s > s^\star$. For any such observation,

$$\frac{\partial}{\partial r(G)} \log \frac{P(H = T \mid Y)}{P(H = F \mid Y)} = \Delta(s) > 0, \tag{30}$$

so the posterior log-odds is strictly increasing in $r(G)$. Since the logistic map from log-odds to posterior probability is strictly increasing, $P(H = T \mid Y)$ also increases with $r(G)$.

This result does not claim that all cycles contribute equally in practice. Rather, it formalizes a sufficient condition under which a larger cycle-space dimension provides stronger structural corroboration, thereby supporting the use of $r(G)$ as a topology-level credibility proxy in Section 3.3.

G Limitations

As a security-oriented study, this work has potential dual-use implications. Although the proposed method is designed to improve the robustness of GraphRAG systems against poisoning attacks, the analysis presented in this paper could still be misused in harmful contexts. We therefore emphasize that the goal of this work is defensive, namely to better understand the vulnerability surface of GraphRAG and to support the development of practical defenses under realistic threat models with limited poisoning budgets.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The paper develops a theoretically motivated hypothesis and provides formal analysis for key components, including the topology-based credibility argument in Section 3.3 and the cycle-basis corroboration model in Appendix F. Experimental results further support these theoretical insights.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Our code is available in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Our code and data is available in Appendix D.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: These details are specified in Section 4 and Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Main results are reported as mean ± standard deviation with different random seeds.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: These details are specified in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work is intended to improve the security and robustness of GraphRAG systems against poisoning attacks, uses existing benchmarks and publicly described models, and does not involve human subjects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: These details are discussed in Appendix G.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a high-risk generative model, scraped dataset, or similar asset that would require special release safeguards. It studies defensive methods on existing benchmarks and models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original sources of all datasets, frameworks, and models used in the paper. They are publicly accessible and permit research and/or non-commercial use under their respective licenses or service terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide documentation for the new code assets introduced in this work through the anonymous code link in Appendix D.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects or crowdsourced data collection, so IRB approval or an equivalent review was not required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs are a substantive component of this work. They are used in conflict identification, model comparison, and automatic answer evaluation. We describe where each LLM is used, provide version information, and document the associated prompting and evaluation settings in the paper and appendix.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.