Trapping Attacker in Dilemma: Examining Internal Correlations and External Influences of Trigger for Defending GNN Backdoors

Fan Yang
The Chinese University of Hong Kong
yf.020@ie.cuhk.edu.hk

Binyan Xu
The Chinese University of Hong Kong
binyxu@ie.cuhk.edu.hk

Di Tang *
Sun Yat-Sen University
tangd9@mail.sysu.edu.cn

Kehuan Zhang †
The Chinese University of Hong Kong
khzhang@ie.cuhk.edu.hk

Abstract

GNNs have become a standard tool for learning on relational data, yet they remain highly vulnerable to backdoor attacks. Prior defenses often depend on inspecting specific subgraph patterns or node features, and thus can be circumvented by adaptive attackers. We propose PRAETORIAN, a new defense that targets intrinsic requirements of effective GNN backdoors rather than surface-level cues. Our key observation is that flipping a victim node's prediction requires substantial influence on the victim: attackers tend to either inject many trigger nodes or rely on a small set of highly influential ones. Building on this observation, PRAETORIAN (i) analyzes internal correlations within potential trigger subgraphs to detect abnormally large injected structures, and (ii) quantifies external node influence to identify triggers with disproportionate impact. Across our evaluations, PRAETORIAN reduces the average attack success rate (ASR) to 0.55% with only a 0.62 percentage-point drop in clean accuracy (CA), whereas state-of-the-art defenses still yield an average ASR of $\geq$20% and a CA drop of at least 3 percentage points under the same conditions. Moreover, PRAETORIAN remains effective against a range of adaptive attacks: adversaries inject many trigger nodes to achieve high ASR ($\geq 80\%$) under random disruption on Cora, with conspicuous changes to the graph and usually lower CA. The tested clean-label variants reach at most 18.1% ASR. Overall, PRAETORIAN constrains attackers to an unfavorable trade-off between efficacy and detectability.

1 Introduction

Graph Neural Networks (GNNs) have become a standard tool for learning on relational data, delivering strong performance on node classification [1–3], link prediction [4], and graph classification [5, 6], and enabling deployments in high-impact applications such as social networks [7], recommendation [8], and biological analysis [9, 10]. However, recent work shows that GNNs are vulnerable to *backdoor attacks* [11–14], where an adversary attaches a trigger subgraph to victim nodes to induce targeted misclassification, raising serious concerns in safety-critical scenarios (e.g., healthcare) [15–18].

Limitations of existing defenses. To mitigate this threat, several defenses have been suggested. Prior defenses largely fall into two categories. *Robust learning* methods reduce sensitivity to malicious perturbations via adversarial training [19, 20] or noise injection [21–23], but they incur substantial

*Corresponding Author
†Corresponding Author

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

overhead [24] and often degrade clean accuracy [25, 26]; moreover, they can be ineffective against complex or adaptive triggers [27]. *Trigger identification* methods attempt to flag suspicious nodes. For example, DPGBA [11] treats trigger nodes (e.g., in GTA [14]) as outliers and isolates them with outlier detection. However, such defenses typically rely on non-fundamental trigger signatures and can be bypassed by *in-distribution* triggers crafted via adversarial learning, as demonstrated by DPGBA itself. A different method, RIGBD [28], drops edges to find nodes with large prediction impact, but it often fails to generalize across trigger designs and attack strengths. Overall, existing defenses remain brittle under adaptive attacks.

Our method: Rather than chasing trigger-specific signatures, we propose PRAETORIAN, a defense built on a *necessary condition* for effective GNN backdoors. To reliably flip a victim's prediction, an attacker must *inject influence* into the victim's local computation—i.e., the trigger must contribute enough message-passing signal to override benign evidence. This requirement is inherent to GNN inference and broadly applies across different trigger encodings.

Crucially, our theory (Appendix Q) makes this influence requirement *operational*. We prove a two-view decomposition: (i) *synergistic influence*, captured by *internal correlation* (IC), and (ii) *per-node influence*, summed in *external influence* (EI). Attackers can spread influence across many low-impact trigger nodes, concentrate it on a few nodes, or use a strategic combination of both. Therefore, any high-ASR backdoor satisfying the non-trivial influence premise must exhibit non-negligible influence in at least one of the two theoretical components, EI or IC. This motivates PRAETORIAN to jointly audit practical proxies for external influence and internal correlation (Appendix Q.3; Section 5.3). Figure 1 shows how these signals separate DPGBA and GTA triggers from benign nodes on OGB-arxiv (Appendix E).

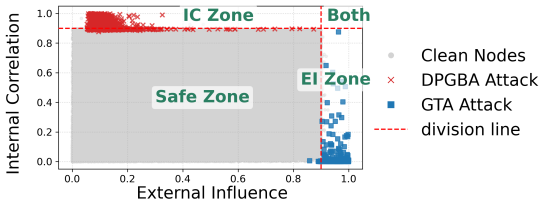


Figure 1: Attackers trapped in dilemma: detected by either high internal correlation (DPGBA) or external influence (GTA).

Specifically, we propose PRAETORIAN, which (i) fuses two views into a stable metric, (ii) filters trigger and victim nodes from ranked scores, and (iii) neutralizes backdoors while preserving clean accuracy (CA). Across multiple datasets and attacks, PRAETORIAN detects victim nodes with 98.52% precision / 94.58% recall and trigger nodes with 98.64% precision / 91.20% recall, and reduces average ASR to 0.55% with only a 0.62 percentage-point CA drop. In contrast, prior defenses yield at least 35% mean ASR with a CA drop of at least 3 percentage points under identical settings. We further evaluate adaptive attackers: achieving high ASR ($\geq 80\%$) under random disruption on Cora involves many noisy trigger nodes and usually lower CA, while the tested clean-label variants reach at most 18.1% ASR.

Contributions. Our contributions are threefold:
(1) A new detection paradigm for GNN backdoor defense. We introduce a two-view framework that jointly examines internal correlations and external influences, providing a new perspective for identifying node-level backdoor triggers beyond surface-level structural or feature cues.
(2) New defense against GNN backdoors. We introduce PRAETORIAN[3], a new defense based on our proposed new paradigm. PRAETORIAN not only effectively identifies trigger and victim nodes but also neutralizes backdoors while maintaining performance on clean data.
(3) Superior Effectiveness and Robustness. We evaluate PRAETORIAN on 4 datasets against 9 backdoor attacks, showcasing significant improvements compared to existing defenses. In these experiments, PRAETORIAN forces attackers into a dilemma: either accept a low ASR or incur conspicuous structural distortions, often accompanied by lower CA.

2 Background & Related Work

2.1 Graph Backdoor Attacks

As GNNs are increasingly deployed in safety- and security-critical settings, protecting them from adversarial manipulation is essential. A prominent threat is the *backdoor attack*, where an adversary implants hidden malicious behavior during training: the compromised model behaves normally on clean inputs but produces attacker-chosen outputs when a specific *trigger* is present [29]. This differs

[3]Code will be released at `https://github.com/Joney-Yf/Graph_backdoor-Defense-Praetorian`.

from test-time adversarial attacks [30], which perturb inputs at inference without modifying the model.

In graph learning, backdoors can target graph-, edge-, and node-level tasks. **1. Whole-graph attacks** [31, 13, 14] manipulate global structures or features so that any test graph containing a trigger subgraph is predicted as a target label. **2. Edge-level attacks** [32, 33] insert or alter edges to induce targeted link predictions by embedding a hidden edge-pattern trigger. **3. Node-level attacks** attach a trigger subgraph to selected victim nodes to misclassify them into a target label. Early work such as SBA [13] proposes universal subgraph triggers but can be less effective, while later attacks improve stealth and success through adaptive perturbations (GTA [14]), budgeted node selection and feature matching (UGBA [12]), and adversarial learning to craft *in-distribution* triggers (DPGBA [11]).

Node-level backdoors are particularly concerning in real deployments, such as social networks, because localized trigger attachments can preserve global topology and keep distributions largely benign. *Accordingly, our goal is to defend against node-level backdoors by detecting and neutralizing malicious trigger subgraphs while preserving clean performance.*

2.2 Difficulty in Graph Backdoor Defenses

Existing GNN backdoor defenses mainly rely on *robust training* or *poison node detection*.

Robust training. Robust training methods [26, 19, 34–36, 20] reduce sensitivity to malicious perturbations by simulating attack scenarios during optimization or injecting noise into the input graph. In practice, they often require generating many adversarial variants, frequently through iterative procedures, leading to substantial computational overhead; they can also degrade clean accuracy [25] due to the robustness–accuracy trade-off.

Poison node detection. Detection-based defenses [11, 28, 12] identify trigger nodes by searching for structural or feature anomalies. This is challenging on graphs because structural heterogeneity and local variation can lead to false positives on benign nodes or false negatives on subtle triggers. Moreover, detectors based on heuristic signatures or fixed anomaly patterns can be evaded by adaptive attackers that craft *in-distribution* triggers mimicking benign neighborhoods [12, 11].

Finally, several image-domain backdoor defenses have been adapted to graphs [28, 37], but such transfer is non-trivial and often ineffective because relational dependencies and message passing create attack and defense behaviors that differ from those in images (Appendix D).

3 Preliminary

3.1 Notations

A summary of symbols is provided in Appendix B.

Attributed graph. We consider an attributed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$ with $|\mathcal{V}| = N$ nodes, edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and node features $\mathcal{X} = \{x_i\}_{i=1}^{N}$. Let $A \in \mathbb{R}^{N \times N}$ be the adjacency matrix where $A_{ij} = 1$ iff $(v_i, v_j) \in \mathcal{E}$ (else 0). We denote the neighbors of v_i by $\mathcal{N}(i)$.

Training graph and victim nodes. We are given a backdoored training graph $\mathcal{G}_T = (\mathcal{V}_T, \mathcal{E}_T, \mathcal{X}_T)$. The training nodes are partitioned into a clean set $\mathcal{V}_C$ and a victim set $\mathcal{V}_B$ with $\mathcal{V}_C \cap \mathcal{V}_B = \varnothing$; nodes in $\mathcal{V}_C$ have clean labels y_i, while nodes in $\mathcal{V}_B$ are assigned the attacker-chosen target label y_t. We call $|\mathcal{V}_B|$ the *victim size (VS)*. Nodes in $\mathcal{V}_T \setminus (\mathcal{V}_C \cup \mathcal{V}_B)$ serve as unlabeled neighbors during training but are not evaluated for classification.

Triggers nodes. To mount a node-level backdoor, the attacker attaches to each $v_i \in \mathcal{V}_B$ a trigger subgraph $\mathcal{G}_i^{\mathrm{Tri}} = (\mathcal{V}_i^{\mathrm{Tri}}, \mathcal{E}_i^{\mathrm{Tri}}, \mathcal{X}_i^{\mathrm{Tri}})$ via an attachment operator $a(v_i, \mathcal{G}_i^{\mathrm{Tri}})$ (details in Appendix A). A trigger may contain multiple connected components (or isolated nodes). We define the *trigger size (TS)* as $|\mathcal{V}_i^{\mathrm{Tri}}|$. We denote the set of *poisoned nodes* as $\mathcal{V}_P = \mathcal{V}_B \cup \big(\cup_i \mathcal{V}_i^{\mathrm{Tri}}\big)$, which are the abnormal nodes our defense aims to identify.

Inference graph. At test time, we evaluate on an unseen graph $\mathcal{G}_U = (\mathcal{V}_U, \mathcal{E}_U, \mathcal{X}_U)$ with $\mathcal{V}_U = \mathcal{V}_{UC} \cup \mathcal{V}_{UB}$, where $\mathcal{V}_{UC}$ are clean nodes and $\mathcal{V}_{UB}$ are victim nodes that would be misclassified when a trigger is attached. We assume an inductive split: $\mathcal{V}_U \cap \mathcal{V}_T = \varnothing$.

3.2 Threat Model

Attacker. We consider node-level backdoor attackers who poison the training graph by attaching trigger subgraphs to a subset of victim nodes $\mathcal{V}_B$ and assigning them an attacker-chosen target label

y_t. Specifically, for each $v_i \in \mathcal{V}_B$, the attacker attaches a trigger $\mathcal{G}_i^{\text{Tri}}$ via an attachment operator $a(v_i, \mathcal{G}_i^{\text{Tri}})$ (Appendix A). At inference, the attacker applies the same attachment operator $a(\cdot)$ to victim nodes in the unseen test graph $\mathcal{G}_U$.

Defender's Knowledge. The defender observes the poisoned training graph $\mathcal{G}_T$ and trains a node classifier, **but does not know the victim set $\mathcal{V}_B$, the trigger nodes, or the target label y_t.** At test time, the defender is evaluated on an unseen graph $\mathcal{G}_U$ that may contain triggered victims.

Defender's Goals. Given a poisoned training graph $\mathcal{G}_T$, the defender aims to (i) identify the target label y_t, the victim set $\mathcal{V}_B$, and the trigger node set $\mathcal{V}^{\text{Tri}}$, and (ii) neutralize the backdoor so that the resulting classifier f achieves low ASR on an unseen backdoored graph $\mathcal{G}_U$ while preserving high clean accuracy (CA). *In our pipeline, y_t is inferred from the observed labels of detected victim nodes.*

The goal on the test graph is to predict clean labels on $\mathcal{V}_{UC}$ while suppressing target-label predictions on nodes in $\mathcal{V}_{UB}$: $\min_f \sum_{v_i \in \mathcal{V}_{UC}} \ell(f(v_i), y_i) + \sum_{v_j \in \mathcal{V}_{UB}} \ell_{\text{anti}}(f(v_j), y_t)$, where ℓ is the standard classification loss and ℓ_{anti} penalizes predicting the target label. We evaluate f on $\mathcal{G}_U$ using CA and ASR.

4 Methodology

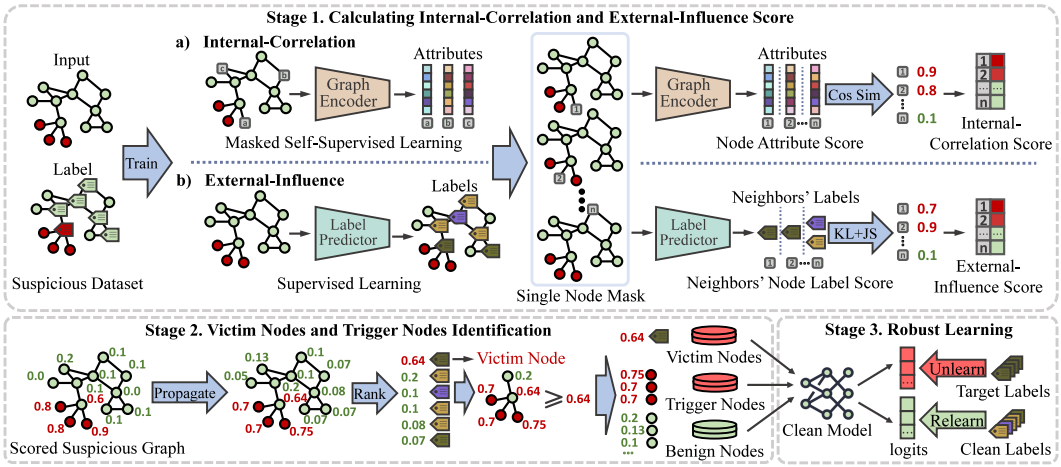


Figure 2: Overview of PRAETORIAN

Section 1 motivates two complementary trigger signals: (i) trigger subgraphs exhibit unusually strong *internal correlation*, and (ii) trigger nodes have outsized *external influence*, as removing them induces larger prediction shifts in their neighborhoods than removing benign nodes.

Building on this, we propose PRAETORIAN, a defense framework with three objectives: (i) quantify both signals reliably, (ii) fuse them to localize victim and trigger nodes, and (iii) neutralize backdoor effects at test time while preserving clean accuracy.

Concretely, PRAETORIAN computes per-node internal-correlation and external-influence scores via masking-based probes, and aggregates them into a unified suspicion score. Using this score, we identify candidate victim/trigger nodes and then perform backdoor neutralization by (a) filtering detected trigger nodes and (b) suppressing target-label behavior on the remaining suspicious nodes. Figure 2 summarizes the pipeline in three stages: *Calculating Internal-Correlation and External-Influence Scores*, *Victim and Trigger Node Identification*, and *Robust Learning Based on Identified Nodes*. Next, we detail each one.

4.1 Calculating Two Scores

Internal-correlation score. Given the poisoned training graph $\mathcal{G}_T = (\mathcal{V}_T, \mathcal{E}_T, \mathcal{X}_T)$, we quantify a node's *internal correlation* using a masked graph autoencoder [38]. Intuitively, nodes that participate in highly unique (easily-memorized) trigger patterns are easier to reconstruct from their neighborhoods.

We sample a mask set $\widehat{\mathcal{V}} \subset \mathcal{V}_T$ and replace their features with a learnable mask token $x_M \in \mathbb{R}^d$: $\widehat{x}_i = \begin{cases} x_i & v_i \notin \widehat{\mathcal{V}} \\ x_M & v_i \in \widehat{\mathcal{V}} \end{cases}$.

An encoder–decoder autoencoder maps the partially observed graph $\widehat{\mathcal{G}}_T = (\mathcal{V}_T, \mathcal{E}_T, \widehat{\mathcal{X}}_T)$ to representations H and reconstructs features $\mathcal{X}'$:

$$H = f_E(\widehat{\mathcal{G}}_T), \qquad \mathcal{X}' = f_D(H, \mathcal{E}_T). \tag{1}$$

We train the autoencoder by minimizing the cosine reconstruction loss over masked nodes:

$$\mathcal{L}_{\text{int}} = \frac{1}{|\widehat{\mathcal{V}}|} \sum_{v_i \in \widehat{\mathcal{V}}} \Big(1 - \cos(x_i, x_i')\Big)^{\gamma},\ \gamma > 1. \tag{2}$$

Internal-Correlation Score Calculation: After training, we mask a candidate node v_i and compute its reconstruction loss $\mathcal{L}_{\text{int}}(i)$; we define the internal-correlation score as:

$$\mathcal{S}_{\text{int}}(i) = \frac{1}{\mathcal{L}_{\text{int}}(i)} \tag{3}$$

Higher $\mathcal{S}_{\text{int}}(i)$ indicates stronger statistical correlation and a higher probability of the node being a trigger node.

External-influence score. To quantify each node's *external influence*, we train a standard node classifier f' on the poisoned training graph (without masking). Let

$$H' = f'_E(\mathcal{V}_T, \mathcal{E}_T, \mathcal{X}_T), \quad \mathcal{Y}' = f'_D(\mathcal{V}_T, \mathcal{E}_T, H') \tag{4}$$

where $\mathcal{Y}'$ are the predicted class probabilities; we optimize f' with cross-entropy on the available labels.

After training, for each node v_i we compute a counterfactual prediction by masking its features and re-running f', obtaining class probabilities $\overline{\mathcal{Y}}$. We measure how much this intervention changes the predictions of its 1-hop neighbors:

$$\mathcal{S}_{\text{ext}}(i) = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \Big(\text{KL}\big(\mathcal{Y}'_j \,\|\, \overline{\mathcal{Y}}_j\big) + \text{JS}\big(\mathcal{Y}'_j \,\|\, \overline{\mathcal{Y}}_j\big)\Big), \tag{5}$$

where KL Divergence [39] and JS Divergence [40] capture distributional shifts (KL is sensitive; JS is symmetric and bounded). *A higher value of $\mathcal{S}_{ext}$ indicates a greater impact on the neighbors' predictions, suggesting a higher probability of being a trigger node.*

4.2 Victim and Trigger Node Identification

Score fusion via dual signals. Internal correlation and external influence are informative under different attack settings (Table 4). We therefore combine them using an adaptively weighted sum, with the exact weights and an equal-weight ablation provided in Appendix L.

Algorithm 1. Recover victim–trigger groups from fused scores
1. Rank. Propagate the fused scores for P hops so high-scoring triggers transfer evidence to attached training victims, then rank the labeled training nodes by decreasing propagated score $\tilde{s}$. Within the search range (the first K nodes), find the largest adjacent gap: $k^\star = \arg\max_{1 \le k < K} (\tilde{s}_{(k)} - \tilde{s}_{(k+1)})$. Retain the first $k^\star$ nodes.
2. Group. Partition these nodes by observed label and keep groups of size at least τ. If none remains, skip localization and unlearning. The cutoff in Step 1 uses scores alone.
3. Refine. For one supported label, scan the ranking from the top, collecting nodes with that label until S consecutive labels differ from it. For multiple labels, keep their seed groups separate.
4. Recover. In each group's P-hop neighborhood on the original graph, exclude identified victims and retain candidates whose propagated score is at least the group's minimum victim score. Pass both node sets and their label to the repair step.

Parameter choices for Algorithm 1 and its input scores.

Quantity	Source	Role in scoring or recovery
Fusion weights, $k^\star$	Inferred	View-specific support; position of the score gap.
P	Inherited	Model depth for propagation and trigger recovery.
$\gamma = 3$, $p = 10\%$	Fixed	Reconstruction contrast; fraction searched for the gap.
$\tau = 5$, $S = 3$	Fixed	Minimum label support; single-label stopping count.

Appendix F specifies propagation and recovery; Appendix M reports sensitivity to the fixed choices.

4.3 Robust Learning Based on Identified Nodes

Localization is effective but not perfect: some poisoned nodes may evade detection, and removing only detected triggers can be insufficient for stealthy *in-distribution* triggers. We therefore train one robust classifier using the identified victims and triggers.

Filtered training + unlearning. Let $\mathcal{C}_{\text{sus}}$ be the retained suspicious label groups, with identified victims $\mathcal{V}_c^{I\text{–}Vic}$ and associated triggers $\mathcal{V}_c^{I\text{–}Tri}$ for each $c \in \mathcal{C}_{\text{sus}}$. Let $\mathcal{V}^{I\text{–}Vic}$ and $\mathcal{V}^{I\text{–}Tri}$ denote their unions over all groups. We filter them from the training graph, $\mathcal{V}_F = \mathcal{V}_T \setminus \left(\mathcal{V}^{I\text{–}Vic} \cup \mathcal{V}^{I\text{–}Tri}\right)$, and optimize:

$$\mathcal{L}_{\text{clean}} = \sum_{v_i \in \mathcal{V}_F \cap V_{\text{train}}} \ell(f(\mathcal{G}_F, v_i),\, y_i)\,, \quad \mathcal{L}_{\text{unlearn}} = -\sum_{c \in \mathcal{C}_{\text{sus}}} \sum_{v_i \in \mathcal{V}_c^{I\text{–}Vic} \cup \mathcal{V}_c^{I\text{–}Tri}} \ell(f(\mathcal{G}_T, v_i),\, c) \tag{6}$$

where V_{train} denotes the labeled training nodes, ℓ is cross-entropy, and $\mathcal{L} = \mathcal{L}_{\text{clean}} + \mathcal{L}_{\text{unlearn}}$. This removes identified backdoor structures and suppresses each group's trigger-target association; the same objective applies whether the converter returns one group or several groups.

5 Evaluation

5.1 Experimental settings.

Configuration. PRAETORIAN uses a 2-layer GAT masked autoencoder with learning rate 1×10^{-4}, sensitivity parameter $\gamma = 3$, and mask rate 0.1. The final defended classifier is a 2-layer GCN trained with Adam at learning rate 0.01 for 400 iterations. We evaluate the final iterate.

Datasets. We evaluate PRAETORIAN on four node-classification benchmarks: Cora [41], PubMed [42], Flickr [43], and OGB-arxiv [44]. Appendix C-Table 9 shows dataset statistics.

Attack methods. We consider four representative graph backdoor attacks: GTA [14], SBA [13], UGBA [12], and DPGBA [11]. We tune attack hyperparameters to obtain strong attack performance before applying defenses. Attack details are provided in Appendix G. We further evaluate five adaptive attacks in Section 6 and Appendix N.

Defense methods. We compare PRAETORIAN with SP [12], OD [11], RIGBD [28], and GNN-Guard [25]. All methods are evaluated under the same training-time defense protocol; OD and SP are therefore applied only during training, although they can also be used at inference. Configurations and tuning protocols are given in Appendix H. We exclude graph-level defenses [45, 46], as they address poisoned graphs rather than node-level victims or triggers.

Evaluation protocol. Following prior work [11, 12, 28], we split each dataset into disjoint training ($\mathcal{G}_{train}$, 80%) and testing ($\mathcal{G}_{test}$, 20%) graphs. The attacker poisons $\mathcal{G}_{train}$ by selecting victim nodes $\mathcal{V}_B$ and attaching triggers with default size 3, producing $\mathcal{G}_T$. The victim size $|\mathcal{V}_B|$ is set to 10, 40, 565, and 160 for Cora, PubMed, OGB-arxiv, and Flickr, respectively. For evaluation, we split $\mathcal{G}_{test}$ into two equal subsets: one is triggered to compute attack success rate (**ASR**), and the other remains clean to compute clean accuracy (**CA**).

Evaluation metrics. We report Recall (R) and Precision (P) for our method in identifying victim and trigger nodes, alongside ASR and CA. Recall (R) represents the proportion of correctly identified victim or trigger nodes relative to all valid nodes of that type in each run. Precision (P) indicates the percentage of correctly identified nodes among the candidates. Each experiment was conducted *five times*, with average results reported. See Appendix I for ASR and CA calculations. All values are percentages.

Runtime. Across the three timed datasets, scoring accounts for over 96% of runtime. On OGB-arxiv, scoring, selection, and relearning take 1,560.11, 40.02, and 15.83 seconds, respectively, with

10.7 GB peak memory. PRAETORIAN-Faster recomputes only the affected L-hop neighborhood and scores 25% fewer candidate nodes on OGB-arxiv, reducing detection time to 5% of the original (Appendix O).

Table 1: Backdoor defense results. Bold indicates the best result within each attack–dataset setting.

Attack	Dataset	Clean Graph	Defense											
			No Defense		GNNGuard		OD		SP		RIGBD		PRAETORIAN	
		CA↑	ASR↓	CA↑	ASR↓	CA↑	ASR↓	CA↑	ASR↓	CA↑	ASR↓	CA↑	ASR↓	CA↑
SBA	Cora	83.19	49.78	83.11	32.09	78.74	45.07	83.33	19.29	82.96	22.84	83.11	**0.09**	**84.30**
	PubMed	85.11	28.57	85.13	16.35	80.94	17.99	**85.34**	3.35	84.99	11.51	78.67	**0.91**	85.21
	OGB-arxiv	65.17	47.32	66.05	24.60	62.69	24.16	63.22	0.50	62.05	**0.00**	66.13	0.03	**66.34**
	Flickr	45.29	90.02	45.22	10.68	41.38	88.09	41.84	89.39	40.67	70.63	40.17	**0.00**	**45.83**
GTA	Cora	83.56	97.87	75.85	27.77	**78.89**	98.49	72.37	58.40	72.07	57.69	77.41	**0.44**	76.89
	PubMed	85.23	94.45	82.09	8.84	80.35	94.85	78.92	74.38	81.47	77.85	79.71	**0.27**	**83.18**
	OGB-arxiv	65.07	93.68	62.66	1.10	**66.36**	**0.00**	38.90	**0.00**	38.47	94.64	62.13	**0.00**	65.68
	Flickr	42.87	100.00	42.55	12.82	42.78	**0.00**	40.79	**0.00**	39.98	44.66	40.25	**0.00**	**43.00**
UGBA	Cora	83.63	94.13	79.19	13.16	74.22	57.42	81.41	94.31	80.30	92.53	79.56	**0.89**	**81.93**
	PubMed	85.21	91.78	82.29	8.26	77.48	52.33	83.51	91.77	82.41	95.48	78.06	**3.67**	**84.47**
	OGB-arxiv	65.08	99.03	**65.97**	97.71	64.79	1.72	64.63	99.37	63.43	**0.00**	65.73	0.05	65.91
	Flickr	44.62	99.96	42.67	97.83	**44.57**	**0.00**	42.53	99.90	42.32	**0.00**	41.54	**0.00**	42.92
DPGBA	Cora	83.70	94.35	80.67	18.09	78.90	94.17	80.67	91.74	81.04	75.48	**81.85**	**0.17**	**81.85**
	PubMed	85.20	95.24	83.36	12.87	81.46	92.79	83.11	93.17	82.96	95.88	78.64	**2.24**	**84.06**
	OGB-arxiv	64.86	92.26	65.54	88.52	64.04	93.97	63.36	90.86	64.77	61.48	65.52	**0.00**	**66.14**
	Flickr	43.45	91.70	43.84	90.14	**44.33**	90.06	43.18	90.93	44.14	85.66	43.77	**0.00**	43.62
Average		69.45	85.01	67.89	35.05	66.37	53.19	65.44	62.33	65.25	55.40	66.39	**0.55**	**68.83**

5.2 Defense Performance

We evaluate PRAETORIAN against baseline defenses across four datasets. Table 1 summarizes the ASR and CA for each method. **(1) Effectiveness:** PRAETORIAN achieves the lowest average ASR (0.55%, compared with 35.05% for GNNGuard). **(2) Clean Accuracy Preservation:** PRAETORIAN maintains a CA comparable to the clean GCN, with an average drop of 0.62 percentage points. This suggests that PRAETORIAN can suppress backdoor behavior while largely preserving utility on clean nodes.

Compared to baselines like GNNGuard, OD, SP, and RIGBD, PRAETORIAN identifies both victim and trigger nodes and uses them for robust learning. We provide a detailed comparative analysis in Appendix J.

5.3 Victim and Trigger Nodes Detection

Table 2 shows the precision and recall of PRAETORIAN in detecting victim and trigger nodes.

Table 2: Victim and trigger node detection.

Attack	Dataset	Victim and Trigger Node Detection			
		Victim Nodes		Trigger Nodes	
		P ↑	R ↑	P ↑	R ↑
SBA	Cora	100.00	100.00	100.00	83.72
	PubMed	100.00	91.50	97.60	70.33
	OGB-arxiv	100.00	99.91	100.00	83.27
	Flickr	96.97	100.00	96.97	100.00
GTA	Cora	98.00	86.00	94.67	86.00
	PubMed	96.79	75.55	96.68	75.55
	OGB-arxiv	100.00	88.85	100.00	88.85
	Flickr	99.37	100.00	99.37	100.00
UGBA	Cora	98.18	100.00	93.33	100.00
	PubMed	100.00	94.50	100.00	94.50
	OGB-arxiv	100.00	90.97	100.00	90.97
	Flickr	100.00	99.38	100.00	99.38
DPGBA	Cora	91.21	100.00	100.00	100.00
	PubMed	97.10	100.00	100.00	100.00
	OGB-arxiv	100.00	86.55	99.84	86.55
	Flickr	98.76	100.00	99.79	100.00
Average		98.52	94.58	98.64	91.20

From the table, we observe that PRAETORIAN consistently achieves high precision, exceeding 90% across all datasets and attack methods, with an average precision of 98%. The high precision indicates that trigger nodes often exhibit clear internal-correlation and/or external-influence anomalies, which PRAETORIAN captures effectively.

On the other hand, PRAETORIAN achieved an average recall of 91%. Notably, for trigger nodes generated by SBA and GTA on the PubMed dataset, the recall falls below 80%. This lower recall mainly arises because SBA/GTA on PubMed tend to produce *disconnected* triggers, where only a *single* trigger node interacts with the victim, which *weakens* the internal-correlation signal. In addition, in some cases the victim's original label *matches the attack's target label*, so masking such a trigger causes only *modest* shifts in the victim's prediction distribution, thereby *weakening* the external-influence signal. Together, these factors *reduce detection sensitivity*, leading to lower recall in these cases. In Section 6, we further examine this case and evaluate PRAETORIAN's robustness to one-node clean-label attacks.

Importantly, even without perfectly identifying all trigger and victim nodes, PRAETORIAN remains highly effective in reducing the ASR of attacks, as shown by the results of SBA and GTA on PubMed dataset (see Table 1).

Score correspondence. With scores and protocol fixed before oracle use, an exact-coalition audit covers four attacks, Cora/PubMed, and five seeds (40 runs; 1,000 victim-trigger groups). Run-level Spearman correlations are 0.66 [95% CI: $0.59, 0.72$] for external score versus singleton influence and 0.61 $[0.53, 0.67]$ for group-mean internal score versus exact group IC. Both associations are positive in all 40 runs (Appendix Q.4).

False positives on attack-free graphs. We ran the full defense five times on each original graph without injecting triggers, using the same settings as in the poisoned-graph experiments. Table 3 reports node-level false-positive rates (FPR) and the change in clean accuracy relative to training without a defense.

Table 3: False positives on attack-free graphs over five runs per dataset.

Dataset	Flagged nodes (mean / median)	Node FPR	ΔCA (pp)
Cora	0.00 / 0	0.000%	+0.00
PubMed	2.20 / 0	0.011%	-0.04
Flickr	0.00 / 0	0.000%	+0.00
OGB-arxiv	3.20 / 0	0.002%	-0.22

Of the eleven false positives on PubMed, ten appeared only in the internal view and one only in the external view. Appendix E.1 examines hubs, class imbalance and community shifts, and gives the calibration protocol.

5.4 Ablation Study

We ablate PRAETORIAN's core components and robust learning module in the main text. Additional results (i.e., static 1:1 weighting and Top-K selection) are deferred to Appendix L.

Internal correlation and external influence. To assess the contribution of the two auditing views, we evaluate two variants: PRAETORIAN$\setminus$I, which removes internal correlation, and PRAETORIAN$\setminus$E, which removes external influence. Table 4 shows that the two views capture different attack patterns. PRAETORIAN$\setminus$I is less effective against dense and structurally robust triggers such as DPGBA, where backdoor patterns can persist after masking individual nodes. In contrast, PRAETORIAN$\setminus$E underperforms on sparse or single-node triggers such as GTA and UGBA, where internal subgraph evidence is limited. Combining both views allows PRAETORIAN to cover a broader spectrum of trigger structures, from sparse to dense patterns. See detailed failure-mode analysis in Appendix K.

Effectiveness of robust learning. To evaluate the defense beyond detection, we compare PRAETORIAN with two variants: PRAETORIAN$\setminus$R, which removes detected nodes and retrains normally, and PRAETORIAN$\setminus$RV, which unlearns only on victim nodes. Figure 3 reports the average ASR and CA across attacks and datasets. Localization is held fixed across repair variants (Appendix L.1).

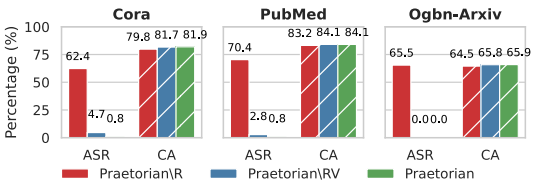


Figure 3: The Role of Robust Learning

PRAETORIAN$\setminus$R still yields a high average ASR of 66.07%, showing that simple node removal is insufficient to eliminate residual backdoor effects. PRAETORIAN$\setminus$RV improves robustness but remains less effective than PRAETORIAN, suggesting that victim-only unlearning does not fully cover the adversarial distribution. By unlearning on both victim and trigger nodes, PRAETORIAN provides more complete adversarial supervision and reduces ASR to nearly 0% while preserving clean accuracy.

5.5 Robustness Against Varying Settings

We evaluate the resilience of PRAETORIAN against varying attack intensities, specifically examining the impact of Trigger Size (TS) and Victim Size (VS). Additional evaluations regarding *model architecture and stop criterion* are detailed in Appendix M.

Table 4: Internal correlation vs. external influence.

Attack	Dataset	Internal Correlation Only						External Influence Only					
		Defense		Victim Detection		Trigger Detection		Defense		Victim Detection		Trigger Detection	
		ASR ↓	CA ↑	P ↑	R ↑	P ↑	R ↑	ASR ↓	CA ↑	P ↑	R ↑	P ↑	R ↑
SBA	Cora	11.82	84.05	71.21	80.00	80.00	66.67	5.24	84.30	81.06	66.00	97.65	42.33
	PubMed	10.77	85.40	80.00	80.00	80.00	66.67	10.18	85.18	80.00	14.50	65.60	5.58
	OGB-arxiv	0.03	66.33	100.00	99.91	100.00	83.27	48.64	66.17	0.00	0.00	0.00	0.00
	Flickr	0.00	46.33	98.77	100.00	99.72	88.89	0.12	42.54	79.89	66.25	94.67	54.93
GTA	Cora	20.71	75.33	69.29	46.00	80.00	46.00	0.00	74.52	91.74	86.00	84.50	86.00
	PubMed	1.73	83.16	100.00	15.50	100.00	15.50	0.31	83.39	96.68	73.00	95.40	73.00
	OGB-arxiv	95.99	60.69	0.00	0.00	0.00	0.00	0.01	65.76	100.00	88.67	100.00	88.67
	Flickr	60.00	41.81	38.53	39.13	38.53	39.13	0.00	41.86	97.67	71.50	92.66	71.50
UGBA	Cora	1.87	82.30	86.06	94.00	100.00	94.00	3.20	81.78	90.18	92.00	92.67	90.00
	PubMed	54.40	83.49	60.00	19.00	60.00	19.00	4.08	84.47	100.00	92.00	100.00	92.00
	OGB-arxiv	0.16	65.96	100.00	90.97	100.00	90.97	99.06	66.09	0.00	0.00	0.00	0.00
	Flickr	0.00	42.86	100.00	96.88	100.00	96.88	0.00	42.87	100.00	99.38	100.00	99.38
DPGBA	Cora	0.17	82.30	92.70	100.00	100.00	100.00	19.13	81.93	80.00	80.00	80.00	77.33
	PubMed	3.72	83.88	97.10	100.00	100.00	100.00	88.13	83.48	53.33	2.00	27.00	1.00
	OGB-arxiv	0.02	66.19	100.00	86.55	99.84	86.55	92.65	65.87	0.00	0.00	0.00	0.00
	Flickr	0.08	41.60	98.77	100.00	99.65	100.00	31.70	40.40	65.04	66.85	63.79	66.46

Impact of Trigger Size (TS). We vary the trigger size from 1 to 5 on the Cora dataset under the DPGBA attack. As shown in Table 5, PRAETORIAN remains stable across these settings, maintaining precision $> 90\%$ and recall $> 85\%$ across all settings. This indicates that our dual-scoring mechanism (internal and external) effectively captures adversarial anomalies across different trigger sizes. Consequently, PRAETORIAN consistently neutralizes the attack (ASR $\approx 0\%$) while preserving a Clean Accuracy (CA) higher than that of the undefended backdoored model.

Impact of Victim Size (VS). To assess scalability, we test victim sizes ranging from 10 to 160. Table 5 reveals that PRAETORIAN remains robust to the scale of the attack, with precision consistently exceeding 90% and recall surpassing 95%. Unlike methods that may degrade as the attack becomes more pervasive, our approach accurately isolates both victim and trigger nodes irrespective of the injection volume. This results in consistent defense performance (near-zero ASR) without compromising model utility.

Table 5: Different Trigger Size and Victim Size

Setting	No Defense		Defense		Node Detection			
					Victim Nodes		Trigger Nodes	
	ASR	CA	ASR	CA	P	R	P	R
TS=1. VS=10	74.81	82.96	0.09	83.26	98.00	88.00	98.00	88.00
TS=2. VS=10	86.17	82.67	2.00	83.11	94.36	98.00	100.00	96.00
TS=3. VS=10	92.44	82.00	0.61	83.41	91.21	100.00	100.00	100.00
TS=4. VS=10	95.48	80.47	0.70	83.04	96.36	100.00	99.51	100.00
TS=5. VS=10	97.30	79.26	0.00	80.37	96.36	100.00	100.00	100.00
TS=3. VS=10	92.44	82.00	0.61	83.41	91.21	100.00	100.00	100.00
TS=3. VS=20	96.61	80.74	0.00	81.56	100.00	100.00	100.00	96.67
TS=3. VS=40	94.70	80.67	0.09	81.78	99.51	100.00	99.67	99.67
TS=3. VS=80	97.39	80.00	0.00	80.44	100.00	100.00	99.83	99.25
TS=3. VS=160	98.70	78.12	0.00	80.67	99.88	99.00	99.00	98.79

6 Adaptive Attacks

We evaluate PRAETORIAN against adaptive adversaries targeting its two-view auditing pipeline. Because PRAETORIAN relies on internal-correlation (IC) and external-influence (EI) scores followed by score-to-node (S2N) localization, attackers can mainly adapt by weakening IC, diluting EI, or challenging S2N. Table 6 summarizes the covered adaptive surfaces and outcomes. We highlight two representative regimes, **One-node Clean-Label** and **Random Disruption**, in the main text, with additional *white-box*, *asymmetric*, and *multi-target* settings reported in Appendix N.

One-node clean-label. This attack jointly weakens both auditing views. By constraining the trigger to a single node (TS$= 1$), it removes intra-trigger consistency cues; by selecting clean-label victims whose ground-truth labels already match the target class, it reduces observable external influence after trigger removal. The remaining backdoor effect is therefore subtle and localized, challenging both the external-influence view and S2N localization. We instantiate this setting with three UGBA and DPGBA variants on Cora and PubMed.

As shown in Table 7, PRAETORIAN substantially reduces ASR, e.g., from 92.18% to 6.76% for $UGBA_{CO}$ with VS=10. Increasing the victim size makes the residual trigger influence harder to hide: when VS exceeds 40, PRAETORIAN identifies at least 60% of triggers. Across victim-size variations and trigger-size variations (Table 19), the post-defense ASR never exceeds 18.1%. The hardest one-node $DPGBA_{CO}$ case retains 18.09% ASR (from 53.74%) with victim/trigger recall of only 28%/26%.

Table 6: Coverage and outcomes of adaptive attacks against PRAETORIAN. IC: internal-correlation score; EI: external-influence score; S2N: score-to-node localization.

Setting	Adaptive lever	Target	Main outcome	Evidence
One-node clean-label	TS$=1$ + clean-label victims	IC, EI	Post-defense ASR is at most 18.1% under varying VS/TS.	Table 7
Random disruption	Noisy triggers + large VS	IC, EI	Large noisy insertions can yield high residual ASR; CA cost varies by configuration.	Fig. 4
White-box optimization	Defense-aware trigger search	IC, EI, S2N	Partial/Full ASR is 2.96%/0.00%; Full trigger recall is 86%.	App. N.1
Asymmetric insertion	Weak train / strong test triggers	train/test gap	ASR is 0%; mean victim/trigger P/R is at least 97%/88%.	App. N.4
Multi-target attack	Multiple target labels	S2N grouping	ASR drops to 0.0% for DPGBA and 2.2% for UGBA; victim and trigger P/R exceed 92%.	App. N.5

Table 7: Varying VS of One-node Clean Label Attack

		No defense		Defense		Victim		Trigger	
Attack	VS	ASR	CA	ASR	CA	P	R	P	R
DPGBA$_{CO}$	10	53.74	82.82	18.09	83.11	50.83	28.00	80.00	26.00
	20	70.87	83.48	15.83	82.74	84.11	33.00	100.00	31.00
	40	66.96	83.19	0.44	80.96	95.63	60.50	98.72	59.50
	80	61.57	82.89	0.00	80.30	98.13	81.00	100.00	80.50
	160	81.22	82.52	0.00	82.37	99.35	89.88	100.00	89.50
UGBA$_{CO}$	10	92.18	81.48	6.76	82.89	84.00	84.00	100.00	84.00
	20	90.31	82.30	4.18	82.44	93.61	85.00	100.00	85.00
	40	93.69	81.11	0.44	80.00	95.10	97.00	100.00	96.50
	80	92.98	80.52	0.00	79.85	98.72	95.00	100.00	95.00
	160	93.07	81.70	0.27	80.82	99.65	97.50	99.45	97.50
DPGBA$_{CO}$ + UGBA$_{CO}$	10	21.57	83.33	13.91	83.56	48.33	20.00	80.00	18.00
	20	38.44	83.33	9.22	83.33	76.17	25.00	100.00	23.00
	40	31.74	83.56	6.87	82.74	93.06	54.00	99.33	53.00
	80	47.48	83.26	2.00	81.63	97.58	72.57	100.00	71.57
	160	49.04	83.11	0.61	81.33	99.15	88.00	100.00	87.25

Random disruption. This attack targets both views from the opposite direction. It injects Gaussian noise into multi-node trigger features to weaken internal-correlation cues, while increasing the victim size to distribute the backdoor association and reduce reliance on any single influential trigger node. We implement this strategy based on SBA on Cora and PubMed.

Figure 4 shows that larger triggers or victim sets can increase ASR under random disruption. Some attacks reach high ASR with little loss in clean accuracy, but inject many noisy nodes. Appendix N.3 gives the results on Cora and PubMed.

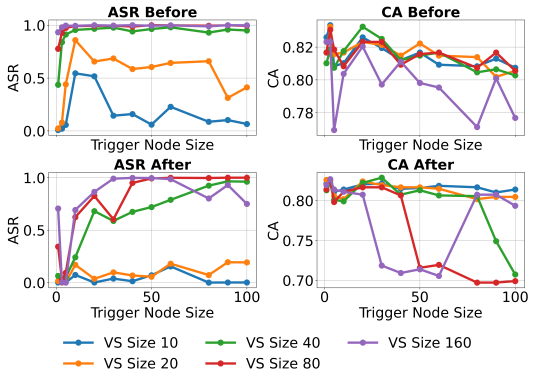


Figure 4: Random disruption on Cora.

7 Conclusion

In this paper, we introduced PRAETORIAN, a new defense paradigm against graph backdoor attacks that integrates both *internal-correlation* and *external-influence* analyses to accurately identify trigger and victim nodes. Experiments across multiple datasets and attack scenarios illustrate the attacker's dilemma: the clean-label variants achieve only a low ASR after defense, while random disruption can sustain high ASR by injecting larger noisy triggers or poisoning more victims.

Acknowledgments and Disclosures

Funding. In the 36 months preceding submission, none of the authors received third-party funding or in-kind support for this work.

Competing interests. During the same period, none of the authors had financial relationships outside the submitted work that could reasonably be perceived as influencing it.

References

[1] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.

[2] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.

[3] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs, 2018.

[4] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks, 2018.

[5] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

[6] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks?, 2019.

[7] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation, 2019.

[8] Chen Gao, Xiang Wang, Xiangnan He, and Yong Li. Graph neural networks for recommender system. WSDM '22, page 1623–1625, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391320. doi: 10.1145/3488560.3501396.

[9] Xiao-Meng Zhang, Li Liang, Lin Liu, and Ming-Jing Tang. Graph neural networks and their current applications in bioinformatics. *Frontiers in Genetics*, 12, 2021.

[10] Elman Mansimov, Omar Mahmood, Seokho Kang, and Kyunghyun Cho. Molecular geometry prediction using a deep generative graph neural network. *Scientific Reports*, 9(1), December 2019. ISSN 2045-2322. doi: 10.1038/s41598-019-56773-5.

[11] Zhiwei Zhang, Minhua Lin, Enyan Dai, and Suhang Wang. Rethinking graph backdoor attacks: A distribution-preserving perspective. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4386–4397, 2024.

[12] Enyan Dai, Minhua Lin, Xiang Zhang, and Suhang Wang. Unnoticeable backdoor attacks on graph neural networks. In *Proceedings of the ACM Web Conference 2023*, pages 2263–2273, 2023.

[13] Zaixi Zhang, Jinyuan Jia, Binghui Wang, and Neil Zhenqiang Gong. Backdoor attacks to graph neural networks. SACMAT '21, page 15–26, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383653. doi: 10.1145/3450569.3463560.

[14] Zhaohan Xi, Ren Pang, Shouling Ji, and Ting Wang. Graph backdoor. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1523–1540, 2021. ISBN 978-1-939133-24-3.

[15] Showmick Guha Paul, Arpa Saha, Md. Zahid Hasan, Sheak Rashed Haider Noori, and Ahmed Moustafa. A systematic review of graph neural network in healthcare-based applications: Recent advances, trends, and future directions. *IEEE Access*, 12:15145–15170, 2024. doi: 10.1109/ACCESS.2024.3354809.

[16] Lin Zhang, Yan Zhao, Tongtong Che, Shuyu Li, and Xiuying Wang. Graph neural networks for image-guided disease diagnosis: A review. *iRADIOLOGY*, 1(2):151–166, 2023. doi: https://doi.org/10.1002/ird3.20.

[17] Jianian Wang, Sheng Zhang, Yanghua Xiao, and Rui Song. A review on graph neural network methods in financial applications, 2022.

[18] David Ahmedt-Aristizabal, Mohammad Ali Armin, Simon Denman, Clinton Fookes, and Lars Petersson. Graph-based deep learning for medical diagnosis and analysis: Past, present and future. *Sensors*, 21(14):4758, July 2021. ISSN 1424-8220. doi: 10.3390/s21144758.

[19] Lukas Gosch, Simon Geisler, Daniel Sturm, Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. Adversarial training for graph neural networks: Pitfalls, solutions, and new directions, 2023.

[20] Jintang Li, Jiaying Peng, Liang Chen, Zibin Zheng, Tingting Liang, and Qing Ling. Spectral adversarial training for robust graph neural network, 2022.

[21] Binghui Wang, Jinyuan Jia, Xiaoyu Cao, and Neil Zhenqiang Gong. Certified robustness of graph neural networks against adversarial structural perturbation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 1645–1653, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467295.

[22] Siyi Qian, Haochao Ying, Renjun Hu, Jingbo Zhou, Jintai Chen, Danny Z. Chen, and Jian Wu. Robust training of graph neural networks via noise governance, 2023.

[23] Enyan Dai, Charu Aggarwal, and Suhang Wang. Nrgnn: Learning a label noise-resistant graph neural network on sparsely and noisily labeled graphs, 2021.

[24] Lichao Sun, Yingtong Dou, Carl Yang, Kai Zhang, Ji Wang, Philip S. Yu, Lifang He, and Bo Li. Adversarial attack and defense on graph data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, page 1–20, 2022. ISSN 2326-3865. doi: 10.1109/tkde.2022.3201243.

[25] Xiang Zhang and Marinka Zitnik. Gnnguard: Defending graph neural networks against adversarial attacks. In *NeurIPS*, 2020.

[26] Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1399–1407, 2019.

[27] Yao Ma, Xiaorui Liu, Neil Shah, and Jiliang Tang. Is homophily a necessity for graph neural networks?, 2023.

[28] Zhiwei Zhang, Minhua Lin, Junjie Xu, Zongyu Wu, Enyan Dai, and Suhang Wang. Robustness inspired graph backdoor defense. In *The Thirteenth International Conference on Learning Representations*, 2025.

[29] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey, 2022.

[30] Joana C. Costa, Tiago Roxo, Hugo Proença, and Pedro Ricardo Morais Inácio. How deep learning sees the world: A survey on adversarial attacks and defenses. *IEEE Access*, 12: 61113–61136, 2024. ISSN 2169-3536. doi: 10.1109/access.2024.3395118.

[31] Jing Xu, Minhui (Jason) Xue, and Stjepan Picek. Explainability-based backdoor attacks against graph neural networks. In *Proceedings of the 3rd ACM Workshop on Wireless Security and Machine Learning*, WiseML '21, page 31–36, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450385619. doi: 10.1145/3468218.3469046.

[32] Haibin Zheng, Haiyang Xiong, Haonan Ma, Guohan Huang, and Jinyin Chen. Link-backdoor: Backdoor attack on link prediction via node injection, 2022.

[33] Jinyin Chen, Haiyang Xiong, Haibin Zheng, Jian Zhang, Guodong Jiang, and Yi Liu. Dyn-backdoor: Backdoor attack on dynamic link prediction, 2021.

[34] Hoang NT, Choong Jun Jin, and Tsuyoshi Murata. Learning graph neural networks with noisy labels, 2019.

[35] Liang Chen, Jintang Li, Jiaying Peng, Tao Xie, Zengxu Cao, Kun Xu, Xiangnan He, Zibin Zheng, and Bingzhe Wu. A survey of adversarial learning on graphs, 2022. URL https://arxiv.org/abs/2003.05730.

[36] Zhao Kang, Haiqi Pan, Steven C. H. Hoi, and Zenglin Xu. Robust graph learning from noisy data. *IEEE Transactions on Cybernetics*, 50(5):1833–1843, May 2020. ISSN 2168-2275. doi: 10.1109/tcyb.2018.2887094.

[37] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data, 2021.

[38] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 594–604, 2022.

[39] James M. Joyce. *Kullback-Leibler Divergence*, pages 720–722. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[40] The jensen-shannon divergence. *Journal of the Franklin Institute*, 334(2):307–318, 1997.

[41] A. K. McCallum, K. Nigam, J. Rennie, and et al. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000. doi: 10.1023/A:1009953814988.

[42] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93, Sep. 2008. doi: 10.1609/aimag.v29i3.2157.

[43] Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor K. Prasanna. Graphsaint: Graph sampling based inductive learning method. *CoRR*, abs/1907.04931, 2019.

[44] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs, 2021.

[45] Xiao Yang, Yuni Lai, Kai Zhou, Gaolei Li, Jianhua Li, and Hang Zhang. Graphprot: Certified black-box shielding against backdoored graph models. In James Kwok, editor, *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 619–627. International Joint Conferences on Artificial Intelligence Organization, 2025.

[46] Yuxin Yang, Qiang Li, Jinyuan Jia, Yuan Hong, and Binghui Wang. Distributed backdoor attacks on federated graph learning and certified defenses, 2024. URL https://arxiv.org/abs/2407.08935.

[47] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain, 2019.

[48] Eugene Bagdasaryan and Vitaly Shmatikov. Blind backdoors in deep learning models, 2021.

[49] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[50] Jie Chen and Yousef Saad. Dense subgraph extraction with application to community detection. *IEEE Transactions on Knowledge and Data Engineering*, 24(7):1216–1230, 2012. doi: 10.1109/TKDE.2010.271.

[51] Edgar N. Gilbert. Random graphs. *The Annals of Mathematical Statistics*, 1959.

[52] Hui Xia, Xiangwei Zhao, Rui Zhang, Shuo Xu, and Luming Wang. Clean-label graph backdoor attack in the node classification task. AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025. ISBN 978-1-57735-897-8.

[53] Xuanhao Fan and Enyan Dai. Effective clean-label backdoor attacks on graph neural networks. CIKM '24. Association for Computing Machinery, 2024. ISBN 9798400704369.

[54] Xiangyu Qi, Tinghao Xie, Yiming Li, Saeed Mahloujifar, and Prateek Mittal. Revisiting the assumption of latent separability for backdoor defenses. In *The eleventh international conference on learning representations*, 2022.

[55] Jun Xia, Zhihao Yue, Yingbo Zhou, Zhiwei Ling, Xian Wei, and Mingsong Chen. Waveattack: Asymmetric frequency obfuscation-based backdoor attacks against deep neural networks, 2023.

[56] Jing Xu and Stjepan Picek. Poster: Multi-target & multi-trigger backdoor attacks on graph neural networks. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, CCS '23, page 3570–3572, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400700507. doi: 10.1145/3576915.3624387.

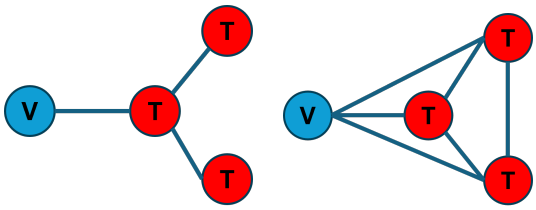
[57] Jialin Lu, Junjie Shan, Ziqi Zhao, and Ka-Ho Chow. Anywheredoor: Multi-target backdoor attacks on object detection, 2025.

[58] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. Deep graph library: A graph-centric, highly-performant package for graph neural networks. 2019.

[59] John C Harsanyi. A simplified bargaining model for the n-person cooperative game. In *Papers in game theory*, pages 44–70. Springer, 1982.

A Attachment Operation Learned by Attacker

In Section 3.2, we describe the threat model, where the attacker first trains an attack model to generate triggers based on the input graph and then attaches these triggers to each victim node. Notably, the attachment process is also learned during the training of the attack model. Furthermore, different attack methods learn distinct attachment strategies on the same dataset, and even the same attack method may adopt different attachment operations across datasets to optimize attack performance. For instance, with a trigger size of 3, the DPGBA attack, trained on the Cora dataset, learns to connect the victim node to the generated trigger via a single edge, as illustrated in Figure. 5a. But in the OGB-arxiv dataset, it establishes three edges to connect the victim node to each trigger node, as shown in Figure. 5b.



(a) Attach Trigger On Cora Dataset

(b) Attach Trigger On OGB-arxiv Dataset

Figure 5: Example of Attachment Operation. "V" represents victim node, and "T" represents trigger node.

B Symbol Table

Here we provide the meaning of symbols used in this paper in Table 8.

Table 8: Main symbols used in the paper.

Term	Symbol
Attributed Graph	$\mathcal{G}$
Edges	$\mathcal{E}$
Nodes	$\mathcal{V}$
Node Attributes	$\mathcal{X}$
Graph Matrix	A
Backdoored Training Graph With Trigger	$\mathcal{G}_T$
Edges in Backdoored Training Graph	$\mathcal{E}_T$
Nodes in Backdoored Training Graph	$\mathcal{V}_T$
Node Attributes in Backdoored Training Graph	$\mathcal{X}_T$
Clean Node Set in Training Graph	$\mathcal{V}_C$
Victim Node Set in Training Graph	$\mathcal{V}_B$
Poison Node Set in Training Graph	$\mathcal{V}_P$
Trigger Node Set in Training Graph	$\mathcal{V}_{Tri}$
Trigger Size of Each Trigger	TS
Victim Size of Total Victim Nodes	VS
Unseen Backdoored Testing Graph With Trigger	$\mathcal{G}_U$
Edges in Backdoored Testing Graph	$\mathcal{E}_U$
Nodes in Backdoored Testing Graph	$\mathcal{V}_U$
Node Attributes in Backdoored Testing Graph	$\mathcal{X}_U$
Trigger Subgraph	$\mathcal{G}^{Tri}$
Nodes in Trigger Subgraph	$\mathcal{V}^{Tri}$
Edges in Trigger Subgraph	$\mathcal{E}^{Tri}$
Node Attributes in Trigger Subgraph	$\mathcal{X}^{Tri}$
Identified Victim Nodes	$\mathcal{V}_{I-B}$
Identified Trigger Nodes	$\mathcal{V}^{I-Tri}$

C Statistics of Datasets

Here we provide the detailed statistics of our experimental datasets in Table 9.

D Challenges in Graph Backdoor Defense Compared to Image Domain

Defending against backdoor attacks in graph data presents unique and significant challenges compared to traditional domains such as image processing. Unlike image data, where malicious samples with incorrect labels can often be detected through visual inspection [47–49], graph data lacks

Table 9: Dataset statistics.

Dataset	#Nodes	#Edges	#Features	#Classes
Cora	2,708	10,556	1,433	7
PubMed	19,717	88,648	500	3
OGB-arxiv	169,343	2,315,598	128	40
Flickr	89,250	899,756	500	7

straightforward indicators. In most graph datasets, discerning the true label of a node solely by observing its topological structure and node features is inherently difficult. This obscurity complicates the identification of poisoned nodes, as adversarial manipulations may not produce immediately obvious anomalies.

Moreover, triggers in graph backdoor attacks exhibit high variability in their characteristics. They can take on any shape or pattern, be located anywhere within the graph, and possess indeterminate sizes. Triggers may manifest as either sparse or dense subgraphs, further complicating their detection. The unpredictable nature of these triggers means that defenders cannot rely on fixed assumptions about their structure or properties. Consequently, methods that focus on detecting specific types of subgraphs, such as dense-subgraph detection techniques [50], become ineffective against a broader range of trigger patterns. This necessitates the development of versatile defense mechanisms capable of handling a diverse array of trigger forms.

Additionally, the flexibility in trigger design allows attackers to craft triggers that are highly stealthy and resilient against existing defense strategies. The ability to embed triggers without significantly altering the overall graph distribution or introducing barely noticeable anomalies requires advanced detection techniques that go beyond superficial inspections of node features or local graph structures. Defenders must therefore prepare to cope with a variety of triggers, employing comprehensive strategies that consider both internal node correlations and external influences within the graph.

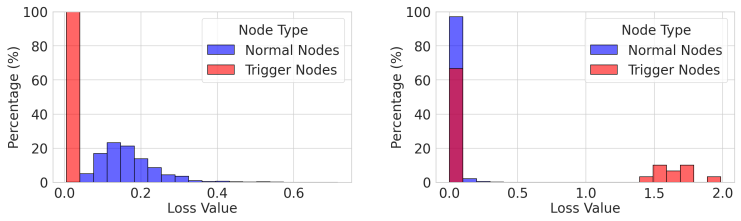
These challenges underscore the complexity of developing effective defenses against graph backdoor attacks. The inherent difficulty in identifying malicious manipulations within graph data, combined with the diverse and adaptable nature of potential triggers, necessitates innovative approaches that can robustly detect and mitigate backdoor threats while preserving the integrity and performance of legitimate GNN applications.

E Motivation

In our threat model, a backdoor attack is considered successful if it primarily alters a victim node's prediction when a trigger subgraph is present, which requires the trigger to exert significant influence. Attackers can achieve this by designing a trigger as either 1) a multi-node subgraph whose collective pattern is unique, or 2) a subgraph containing a few nodes with abnormally high influence, or 3) a strategic combination of both. Indeed, we theoretically prove (Theorem Q.2, Appendix Q) that any non-trivial attack's influence provably manifests in at least one of these two mechanisms. This suggests two detection angles: the first type of trigger can be revealed by its internal correlations, while the second can be identified by its external influence. This leads us to analyze graph backdoors from two perspectives:

Internal perspective. This perspective focuses on the unique patterns within triggers. To manipulate a victim node's classification without affecting others, trigger patterns are often designed to be easily memorized by the model, thereby exhibiting strong statistical correlation. A masked reconstruction task is well suited to capture this internal signal. Accordingly, we use masked reconstruction to quantify it. Consequently, for a well-trained model, masked portions of a trigger should be reconstructed with significantly lower error than parts of benign subgraphs.

External perspective. This perspective examines how triggers influence a victim's prediction. Even when triggers exhibit no obvious internal anomalies, effective triggers tend to induce disproportionate changes in the model's decision process for the victim (and often trigger's neighborhood). To capture each node's impact, we perform a counterfactual masking analysis and measure the variance of its neighbors' logits before and after masking. For nodes that truly belong to a trigger, the resulting variance is significantly larger than that induced by masking benign nodes.



(a) Internal correlation (b) External influence

Figure 6: Comparison between normal nodes and trigger nodes.

We empirically evaluated these two perspectives on the OGB-arXiv dataset using the DPGBA and GTA attacks, each injecting 565 triggers (3 nodes per trigger). To quantify internal correlation, we trained a 2-layer masked autoencoder and measured the reconstruction error on the backdoored graphs. To assess external influence, we trained a 2-layer backdoored GAT and computed the prediction variance induced by iteratively masking individual nodes. As shown in Figure 1, the results indicate that our two viewpoints are complementary and capture different characteristics of the attacks.

Furthermore, we provide an additional example in Figure 6 demonstrating that backdoor triggers remain distinguishable from normal nodes under both perspectives. These results are obtained from the DPGBA attack on the Cora dataset, using 10 triggers (3 nodes each).

The generality of these findings is further supported by experiments across multiple datasets and attack types (see Sections 5.2 and 5.3). We also analyze the conditions under which these approaches may fail through detailed ablation studies and security analyses (Sections 5.4 and 6).

E.1 False positives on attack-free graphs

Section 5.3 reports the four-dataset audit in Table 3.

Pruning flagged nodes alone changed CA by 0.00 pp on Cora and PubMed. Further analyses of these two datasets examine the following benign structures.

Hubs and cross-view reinforcement. No top-100 degree node was flagged in any seed. Across all five seeds, PubMed had 11 false positives: 10 internal-view-only, one external-view-only, and none reinforced by both views. Three false positives belonged to the broader top-5% degree set (0.6 per seed).

Class imbalance. In PubMed, minority class 0 contains 20.8% of nodes and had no false positives; classes 1 and 2 had six and five across the seeds, corresponding to FPRs of 0.0155% and 0.0127%. The datasets provide class labels but no demographic attributes.

Shifted communities. For Louvain communities containing at least 20 nodes, Cora had zero false positives across 22 communities. PubMed had 11 across 34 communities, with only 3/11 in the top shift decile; the most-shifted community's anomaly score was near the graph average. In a controlled stress test, coherently shifting a natural Cora community by $\delta \in \{0, 1, 2, 3, 5, 8\}$ feature standard deviations produced no overall or within-community false positives at any tested level.

Calibration boundary. PRAETORIAN does not threshold classifier confidence. Its support threshold admits only label groups containing at least τ high-score nodes. A larger τ can reject noisy groups but can also exclude genuine low-budget attacks. One possible deployment rule is to choose the smallest τ meeting a specified FPR ceiling on clean calibration data and re-audit its FPR after a distribution shift.

F Score-to-Node Conversion Details

Overview. This section details how PRAETORIAN converts fused node-level deviation scores into suspicious victim groups and trigger sets. The conversion has five steps: (i) propagate node scores to labeled training nodes for victim scoring, (ii) place a label-free valley cutoff on the propagated training-node scores, (iii) infer supported label groups from the valley-selected anchor nodes, (iv)

construct victim groups with a single-target refinement or group-wise multi-target rule, and (v) recover trigger nodes from the neighborhoods of the retained victim groups.

Victim-score propagation. Let s_i be the fused deviation score of node v_i. To emphasize large deviations, we set $L_i^0 = s_i^r$ with $r \geq 3$. We then propagate these scores on a modified adjacency matrix. Let V_{train} be the training nodes and define $A_{ij}^{\text{train}} = 1$ iff $v_i, v_j \in V_{\text{train}}$ and $(v_i, v_j) \in \mathcal{E}_T$. We remove edges between training nodes and add self-loops:

$$A' = A - A^{\text{train}} + I,$$

where A is the original adjacency matrix and I is the identity matrix. With D' denoting the degree matrix of A', we propagate

$$L^{p+1} = D'^{-\frac{1}{2}} A' D'^{-\frac{1}{2}} L^p, \quad p = 0, \ldots, P-1.$$

The propagated training-node score is $\tilde{s}_i = L_i^P$ for $v_i \in V_{\text{train}}$. Removing training-training edges reduces score diffusion among benign training nodes, while self-loops preserve high deviation on suspicious nodes. Propagating P times matches the victim model's message-passing depth, allowing trigger evidence to reach the training victims it influences.

Largest-gap cutoff and support-based abstention. After deviation propagation, each training node receives a propagated score $\tilde{s}_i$. The valley cutoff is motivated by the empirical two-population structure of poisoned graphs: benign nodes tend to concentrate in a low-deviation mode, while poison-related nodes can form a separated upper-deviation mode.

We estimate the valley without labels by sorting the propagated scores in descending order,

$$\tilde{s}_{(1)} \geq \tilde{s}_{(2)} \geq \cdots \geq \tilde{s}_{(m)}.$$

To reduce sensitivity to benign long-tail scores, we apply the largest-gap criterion only within the top 10% of the ranked nodes, where suspicious victim/trigger-related nodes are expected to concentrate. Let K denote the number of nodes in this high-deviation region. We report sensitivity to this search range in Appendix M. We select

$$k^\star = \arg\max_{1 \leq k < K} \left(\tilde{s}_{(k)} - \tilde{s}_{(k+1)} \right),$$

and place the cutoff at

$$\theta_{\text{val}} = \frac{\tilde{s}_{(k^\star)} + \tilde{s}_{(k^\star+1)}}{2}.$$

The upper-mode seed set is then

$$\mathcal{H} = \{v_{(1)}, \ldots, v_{(k^\star)}\}.$$

The resulting seed set must further contain at least one label-supported group, i.e., $\exists c, |\mathcal{H}_c| \geq \tau$. If no such group exists, PRAETORIAN abstains. We emphasize that this is a practical score-to-node conversion rule rather than a theoretical requirement of the EI/IC decomposition.

Minimum-support label grouping. Labels are used only after the label-free anchor set is obtained. For each observed label c, we form

$$\mathcal{H}_c = \{v_i \in \mathcal{H} : y_i = c\}.$$

We retain label groups with nontrivial anomalous support:

$$\mathcal{C}_{\text{sus}} = \{c : |\mathcal{H}_c| \geq \tau\}, \quad \tau = 5.$$

The threshold τ is not the anomaly boundary; the anomaly boundary is determined by the valley cutoff above. Instead, τ is a small anti-noise support filter. A learnable node-level backdoor typically requires repeated trigger-target associations during training, while one or two isolated high-score nodes provide weak evidence for a distinct target-specific backdoor and are more likely to arise from benign nodes near the cutoff. We therefore set $\tau = 5$ as a fixed minimum-support rule.

Single-target refinement. If only one label group passes the support filter, i.e., $\mathcal{C}_{\text{sus}} = \{\hat{y}_t\}$, we treat it as the standard single-target case and infer $\hat{y}_t$ as the target label. In this case, the valley-selected anchor set identifies the target label, while the final victim boundary further uses label consistency. Specifically, we scan the sorted training-node list $v_{(1)}, \ldots, v_{(m)}$ in descending propagated score order and stop after observing $S = 3$ consecutive nodes whose labels differ from $\hat{y}_t$. Let

$$j^\star = \min\{j \geq S : y_{(j-S+1)} \neq \hat{y}_t, \ldots, y_{(j)} \neq \hat{y}_t\},$$

with $j^\star = m + 1$ if no such index exists. The identified victim group is

$$\mathcal{V}_{\hat{y}_t}^{I-Vic} = \{v_{(j)} : 1 \le j < j^\star,\ y_{(j)} = \hat{y}_t\}.$$

This refinement does not assume the target label in advance; the label is inferred from the valley-selected anchor set. We report the sensitivity to S in Appendix M.

Multi-target case. If multiple label groups pass the support filter, we do not apply the single-target stopping rule. Instead, each supported group is treated as one suspicious target group,

$$\mathcal{V}_c^{I-Vic} = \mathcal{H}_c, \quad c \in \mathcal{C}_{\text{sus}}.$$

This avoids assuming a single target label and keeps the procedure applicable to multi-target or ALL2ALL-style settings.

Trigger recovery. For each retained victim group $\mathcal{V}_c^{I-Vic}$, we search its P-hop neighborhood in the original poisoned graph $\mathcal{G}_T$; the edge-removed graph above is used only for score propagation. Let $\mathcal{N}_P(\cdot; \mathcal{G}_T)$ denote the P-hop neighborhood in $\mathcal{G}_T$. Candidate trigger nodes are

$$\mathcal{B}_c = \mathcal{N}_P(\mathcal{V}_c^{I-Vic}; \mathcal{G}_T) \setminus \bigcup_{c' \in \mathcal{C}_{\text{sus}}} \mathcal{V}_{c'}^{I-Vic}.$$

We retain high-deviation candidates using the victim-group boundary:

$$\mathcal{V}_c^{I-Tri} = \{v_i \in \mathcal{B}_c : L_i^P \ge \min_{v_j \in \mathcal{V}_c^{I-Vic}} L_j^P\}.$$

This recovers unlabeled trigger nodes by anchoring them to the suspicious labeled victims they influence, without assuming the trigger size.

Group-wise unlearning. If only one label group remains, the procedure reduces to the standard single-target case. If multiple groups remain, PRAETORIAN applies the same trigger recovery and unlearning objective to each group. Thus multi-target and ALL2ALL-style settings are handled as multiple suspicious label groups; the defender does not need to know the number of target labels or the source-to-target mapping.

G The details of Attacks

SBA [13]: SBA employs randomly generated graphs as triggers in graph backdoor attacks. It leverages the Erdős-Rényi (ER) model [51] to create random subgraph structures with a limited number of nodes. Each trigger node is assigned random features, effectively embedding the trigger subgraph into the target graph without introducing discernible patterns.

GTA [14]: GTA utilizes a trigger generator that crafts subgraphs as triggers tailored to individual samples. The trigger generator is optimized exclusively based on the backdoor attack loss, disregarding any constraints related to trigger detectability. This allows GTA to produce highly effective, sample-specific triggers that enhance the attack success rate.

UGBA [12]: UGBA selects representative and diverse nodes as poisoned nodes to fully utilize the attack budget. It employs an adaptive trigger generator that is optimized with a constraint loss, ensuring that the generated triggers closely resemble the target nodes. This similarity helps UGBA evade detection while maximizing the effectiveness of the backdoor attack.

DPGBA [11]: DPGBA introduces an adversarial learning strategy to generate in-distribution triggers. It proposes a novel loss function that guides the trigger generator to produce efficient in-distribution triggers, making them indistinguishable from legitimate subgraphs. This approach enhances the stealthiness and effectiveness of the backdoor attacks by ensuring that the triggers blend seamlessly with the graph's natural structure.

H The details of Defenses

Baseline tuning and reporting protocol. For a fair comparison, we carefully tune each baseline under the same training-time defense protocol. When official implementations and recommended hyperparameters are available, we follow the original papers and further tune their key hyperparameters

on the validation set. When official configurations are not available, we perform a validation-based grid search over the main hyperparameters that control defense strength. For each baseline, we report the best validation-selected configuration, and the corresponding test performance is reported in the main results. No test labels, test-time trigger information, or attack-specific test statistics are used for tuning either PRAETORIAN or the baselines.

GNNGuard [25]: GNNGuard is a robust defense method for GNNs that protects against adversarial attacks by leveraging node similarity to filter out adversarial edges. It employs a multi-stage defense strategy that dynamically adjusts edge weights during training, thereby enhancing the model's resilience to structural perturbations.

SimilarityPrune (SP) [12]: SP is designed to prune edges that link nodes with low similarity, effectively removing triggers that significantly differ from clean nodes. By eliminating these dissimilar connections, SP reduces the impact of malicious subgraphs on the GNN's predictions.

Outlier Detection (OD) [11]: OD introduces a graph auto-encoder to identify and filter out nodes with high reconstruction loss, which are considered outliers compared to clean nodes. By removing these outlier nodes, OD effectively eliminates triggers that deviate from the normal node distribution.

Unlike OD's high reconstruction error on unmasked inputs, our internal score uses low reconstruction error after hiding the candidate feature and reconstructing it from context.

RIGBD [28]: RIGBD is a robust defense method for GNNs that identifies victim nodes by detecting high prediction variance through random edge dropping. Once identified, RIGBD applies robust learning techniques to these victim nodes to counteract the influence of trigger patterns, thereby mitigating the impact of backdoor attacks.

I The Calculation of ASR and CA

I.1 Attack Success Rate (ASR)

ASR measures the proportion of backdoored victim nodes in the testing set that are incorrectly classified into the target class when the trigger is present.

$$\text{ASR} = \left(\frac{\text{\# of victim nodes classified as target class}}{\text{Total \# of filtered victim nodes}} \right)$$

where "filtered victim nodes" means the victim nodes that are not in the target class.

I.2 Clean Accuracy (CA)

CA assesses the GNN's performance on clean, non-backdoored samples, indicating its ability to maintain accuracy on legitimate data.

$$\text{CA} = \left(\frac{\text{Number of clean samples correctly classified}}{\text{Total number of clean samples}} \right)$$

J Comparison Results Analysis

Compared with GNNGuard, which down-weights dissimilar neighbors via cosine similarity during training, PRAETORIAN learns normal patterns on the filtered graph and suppresses the target label on identified victims and triggers (Section 4.3). This leads to lower average ASR and higher average CA (Table 1).

OD filters nodes with high reconstruction error, while SP prunes edges between nodes with low feature similarity. Compared to OD and SP, PRAETORIAN accurately identifies both victim and trigger nodes. Moreover, instead of merely detecting these nodes, PRAETORIAN leverages the identified ones for robust learning, resulting in superior performance.

In comparison with RIGBD, which identifies only victim nodes, PRAETORIAN detects both victim and trigger nodes. This dual identification allows us to act against backdoor attacks directly (rather than only via victim proxies), face the attacker pattern more directly, and provide more samples covering

the attack patterns as supervision for unlearning during robust learning. In addition, explicitly handling triggers helps stabilize neighborhoods when a trigger connects to multiple nodes, reducing residual influence at test time. Together, these factors reduce adversarial pattern insertion and enhance defense performance (lower ASR without sacrificing CA).

The possible reason for the performance gap of RIGBD: We also observe a gap relative to the results reported in the RIGBD paper, which appears to stem from RIGBD's sensitivity to attack strength: it is effective under higher budgets but brittle under subtler settings. For example, under UGBA, changing the configuration from *victim_size=40*, *prune_thr=0.5* to *victim_size=10*, *prune_thr=0.1* leads to detection failure (ASR changes from 0.44% to 92.44%). This failure under high-budget settings likely stems from RIGBD's reliance on prediction variance under edge perturbation. RIGBD assumes that removing adversarial edges will cause significant fluctuations in model output. However, in high-intensity attacks (e.g., dense trigger subgraphs), the adversarial connectivity is highly redundant. Even if a subset of edges is pruned during RIGBD's sensitivity check, the remaining edges maintain sufficient information flow to preserve the attack pattern. Consequently, the model's prediction remains low variance, missing the detection of victim nodes.

K Detailed Ablation Analysis

In this section, we provide a granular analysis of the ablation study results presented in Section 5.4, focusing on how specific attack characteristics affect each component.

Analysis of PRAETORIAN\E (Reliance on Internal Correlation): The capability of PRAETORIAN\E is significantly impaired under attacks that deploy a single trigger node per victim (e.g., GTA and UGBA) on the OGB-arxiv and PubMed datasets. When a lone trigger node is removed or masked, the structural integrity of the potential trigger subgraph collapses, effectively obscuring the backdoor signature. This reveals a critical limitation: *internal correlation relies on the presence of multi-node substructures and becomes less effective against sparse, single-node triggers.*

Analysis of PRAETORIAN\I (Reliance on External Influence): PRAETORIAN\I faces challenges in distinguishing trigger nodes from benign nodes under attacks that generate highly robust, dense trigger patterns (e.g., DPGBA and SBA). For instance, DPGBA constructs fully connected trigger subgraphs (cliques). These patterns remain intact through message passing even when individual nodes are masked, maintaining their adversarial influence. *This resilience highlights the inadequacy of relying solely on external influence against structurally robust triggers.*

Synergy in PRAETORIAN: As shown in Table 2, the full PRAETORIAN model overcomes these individual limitations. By combining external checks (effective against sparse triggers) with internal correlation (effective against structural anomalies), PRAETORIAN achieves the necessary synergy to identify nearly all trigger nodes across diverse attack strategies.

L Addtional Ablation Study

Weighted deviation score. Next, we examine our weighted-sum approach to combining internal-correlation and external-influence scores. Concretely, we run the victim/trigger identification procedure separately using $\mathcal{S}_{\text{int}}$ and $\mathcal{S}_{\text{ext}}$, obtaining $(N_v^{\text{int}}, N_t^{\text{int}})$ and $(N_v^{\text{ext}}, N_t^{\text{ext}})$ as the numbers of detected potential victims and triggers. We use $N_v N_t$ as a reliability proxy and compute:

$$\mathcal{S} = \left(N_v^{\text{int}} N_t^{\text{int}}\right) \mathcal{S}_{\text{int}} \; + \; \left(N_v^{\text{ext}} N_t^{\text{ext}}\right) \mathcal{S}_{\text{ext}}. \tag{7}$$

The raw scores are used without normalization. We introduce PRAETORIAN$\sim$1:1, which assigns equal weights to both scores. We focus on the PubMed dataset for all attack types to highlight differences.

Table 10 shows that: (1) Except for the UGBA attack, where PRAETORIAN$\sim$1:1 performs slightly better than PRAETORIAN, our primary model PRAETORIAN generally achieves superior performance by maintaining lower ASR and higher CA. (2) PRAETORIAN consistently exhibits higher precision and recall in identifying both victim and trigger nodes. These results indicate that the merit of dynamically assigning weights to internal-correlation and external-influence scores can intrinsically suppress noise, helping pinpoint poisoned nodes more accurately and enhancing overall defense.

Table 10: Comparison of weighting schemes.

Method	Attack	Defense		Node Detection			
				Victim Nodes		Trigger Nodes	
		ASR(%)	CA(%)	P(%)	R(%)	P(%)	R(%)
PRAETORIAN 1 : 1	SBA	8.24	85.11	80.00	31.00	65.60	20.58
	GTA	0.85	83.26	100.00	61.50	94.85	61.50
	UGBA	2.25	84.41	100.00	98.00	100.00	98.00
	DPGBA	9.55	84.18	84.56	50.50	93.61	44.83
PRAETORIAN (Ours)	SBA	0.91	85.21	100.00	91.50	97.60	70.33
	GTA	0.27	83.18	96.79	75.55	96.68	75.55
	UGBA	3.67	84.47	100.00	94.50	100.00	94.50
	DPGBA	2.24	84.06	97.10	100.00	100.00	100.00

Table 11: Comparison of different selection methods.

Selection Percentage	Defense		Node Detection			
			Victim Nodes		Trigger Nodes	
	ASR(%)	CA(%)	P(%)	R(%)	P(%)	R(%)
Top 1%	65.84	80.93	19.43	97.25	35.15	91.29
Top 3%	66.11	79.01	5.53	98.38	12.25	94.50
Top 5%	42.29	79.86	3.45	99.00	7.59	96.83
Top 7%	46.24	79.68	2.65	99.13	5.43	97.08
Top 10%	40.63	80.26	2.27	99.38	3.90	98.88
Top 15%	34.95	80.83	2.05	99.38	2.53	98.88
PRAETORIAN (Ours)	1.03	84.24	95.39	91.63	95.92	85.88

Node selection. We also evaluate our method for selecting victim and trigger nodes via a variant, PRAETORIAN$\setminus$S, which regards the top $K\%$ most suspicious trigger nodes and designates their connected nodes as victim nodes. Due to space limits, we only show results on the PubMed dataset.

We observe in Table 11 that as K grows, recall for both victim and trigger nodes remains above 90%. However, precision drops due to increased false positives among benign nodes. ASR remains at least 34.95%, compared with 1.03% for PRAETORIAN, and CA is over 3.3 percentage points lower at every K.

L.1 Repair with incomplete localization

The repair ablation is rerun independently of the main end-to-end experiments under the same configurations. Within this ablation, localization is held fixed when comparing removal-only retraining, additional victim-side unlearning, and additional victim-and-trigger unlearning (Table 12). Each entry gives the mean ASR and CA across attacks.

Table 12: Fixed-localization repair ablation. Entries are ASR / CA (%).

Dataset	Removal only	+ Victim	+ Victim and trigger
Cora	62.4 / 79.8	4.7 / 81.7	0.8 / 81.9
PubMed	70.4 / 83.2	2.8 / 84.1	0.8 / 84.1
OGB-arxiv	65.5 / 64.5	0.0 / 65.8	0.0 / 65.9

Detected victim-trigger groups provide target-conditioned negative supervision on the poisoned graph; their use in repair does not require removing every injected node.

For the two PubMed cases with trigger recall below 80%, SBA has trigger precision/recall 97.60%/70.33%, ASR 28.57% before and 0.91% after defense, and defended CA 85.21%. GTA has trigger precision/recall 96.68%/75.55%, ASR 94.45% before and 0.27% after, and defended CA 83.18%. Disconnected trigger components with only one node interacting directly with the victim can weaken internal evidence; victims whose original labels already match the target can show modest prediction shifts.

Table 13: The Performance of Different Architectures

Attack	Architecture	No Defense		Defense		Node Detection			
						Victim Nodes		Trigger Nodes	
		ASR(%)	CA(%)	ASR(%)	CA(%)	P(%)	R(%)	P(%)	R(%)
DPGBA	GAT	94.35	80.67	0.17	81.85	91.21	100.00	100.00	100.00
	GCN	94.52	81.26	0.52	83.56	91.99	92.00	100.00	92.00
	GIN	94.52	81.41	0.70	79.41	68.61	100.00	69.35	97.33
UGBA	GAT	94.13	79.19	0.89	81.93	98.18	100.00	93.33	100.00
	GCN	89.60	79.93	4.44	83.63	98.18	100.00	94.85	100.00
	GIN	93.24	79.93	5.42	83.41	96.36	98.00	94.85	98.00
GTA	GAT	97.87	75.85	0.44	76.89	98.00	86.00	94.67	86.00
	GCN	98.44	75.19	0.00	76.15	98.18	90.00	96.36	90.00
	GIN	98.40	75.41	0.00	71.93	94.55	100.00	90.00	100.00
SBA	GAT	49.78	83.11	0.09	84.30	100.00	100.00	100.00	83.72
	GCN	51.91	83.85	9.33	84.18	60.90	52.00	69.09	34.00
	GIN	46.84	83.63	0.00	83.78	92.32	96.00	86.36	72.00

Table 14: Highest-ranked non-victim node position

Metric\Dataset	**Cora**	**PubMed**	**OGB-Arxiv**
Average highest rank	10.55	37.45	520.25
Total victim nodes	10	40	565

M Hyperparameter Study

Model Architecture. We observe that our defense system, PRAETORIAN, utilizes a 2-layer GAT architecture, while both the attacker model and the benign model are based on a 2-layer GCN architecture.

The decision to use a GAT architecture in our defense is motivated by its ability to assign different weights to each neighbor, thereby giving higher importance to nodes that significantly contribute to the mask prediction task. This capability allows GAT to better capture relevant information within the context of our defense. However, the architectural difference between the defense model and the attacker's model may raise concerns about whether this inconsistency could influence the performance of PRAETORIAN.

To address this concern, we conducted two additional experiments for each attack on the Cora dataset to evaluate whether the architecture mismatch affects the performance.

The results, presented in Table 13, indicate that PRAETORIAN consistently achieves strong performance in accurately identifying both victim and trigger nodes, with averaged precision and recall values exceeding 80%. Moreover, PRAETORIAN effectively defends against all attacks across model architectures, with post-defense ASR at most 9.33% in all 12 settings.

Sensitivity to Stop Criterion We stop the victim node selection when we encounter S consecutive nodes with labels different from the target label. As shown in Table 14 (which shows the rank of the highest non-victim node), the highest rank of a non-victim node is typically far from the majority of victim nodes. This separation already indicates that a small S (e.g., k=2) is unlikely to cause the selection to stop prematurely and miss true victim nodes.

To further provide direct evidence for this robustness, we vary S from 1 to 5 on Cora datasets. Table 15 presents the averaged results over all attacks, which show that PRAETORIAN's performance is highly stable and not sensitive to this hyperparameter.

As S varies, all key metrics remain remarkably stable: ASR stays within 0.73–1.60% and CA within 76.89–77.46%. Victim precision and recall remain high (ranging from 90.61–95.71% and 91.5–94.5%, respectively), while trigger precision and recall are also stable (98.57–100% and 85.51–89.17%). This stability indicates that PRAETORIAN is robust to the specific choice of S.

By default, we set S=3. Its victim precision and recall are 91.36% and 94.50%, respectively.

Sensitivity to cutoff-related hyperparameters. We evaluate the sensitivity of the score-to-node conversion on Cora dataset to the cutoff search range p. As shown in Table 16, varying p from 3% to

Table 15: Sensitivity to the stop criterion.

#Different Labels (S)	Defense		Victim Nodes		Trigger Nodes	
	ASR(%)	CA(%)	P(%)	R(%)	P(%)	R(%)
1	1.60	77.24	95.71	91.50	100.00	85.51
2	0.73	77.16	93.72	93.00	98.57	87.66
3	0.93	77.46	91.36	94.50	98.57	89.17
4	1.22	76.89	91.36	94.50	98.57	89.17
5	0.79	77.10	90.61	94.50	98.57	88.67

Table 16: Sensitivity to the cutoff search range p.

Cutoff Range p	Defense		Victim Nodes		Trigger Nodes	
	ASR(%)	CA(%)	P(%)	R(%)	P(%)	R(%)
3%–10%	0.00	78.12	97.0	100.0	96.8	100.0

10% does not change the defense outcome. The support threshold τ is used only after the label-free valley cutoff to remove isolated noisy seeds; it does not determine the anomaly boundary. The direct support-count sweep below identifies a stable range and the low-budget failure boundary. We fix $\tau = 5$ across datasets and attacks. These results suggest that the cutoff is driven by a stable score separation rather than by carefully tuned post-processing parameters.

M.1 Direct sweeps of the reconstruction exponent and support count

The defaults $\gamma = 3$ and $\tau = 5$ follow their different roles: γ controls nonlinear contrast among masked-reconstruction residuals, while τ removes isolated label groups after the anomaly boundary has already been selected. Table 17 reports mean results across DPGBA, GTA, SBA, and UGBA on Cora. Undefended ASR is 84.10%. We include $\gamma = 1$ as a boundary test of the default $\gamma > 1$ setting.

Table 17: Cora parameter sweeps: four-attack means (%). P/R denote trigger precision/recall.

γ	ASR	P	R	τ	ASR	P	R
1	10.10	100.00	35.00	1	1.22	84.83	80.00
3 (default)	0.40	97.00	92.43	3	0.40	97.00	92.43
5	0.40	97.00	92.43	5 (default)	0.40	97.00	92.43
7	0.40	97.00	92.43	7	0.40	97.00	92.43
9	0.40	97.00	92.43	9	0.40	97.00	92.43
11	4.44	90.64	80.20	10	22.43	82.25	81.68
15	19.61	46.43	45.00	11	46.19	49.33	50.00

Both sweeps show a plateau at $3, 5, 7, 9$. At $\gamma = 1$, weak nonlinear contrast reduces separation and trigger recall; large γ can destabilize inverse-score ranking through numerical underflow. At $\tau = 1$, isolated noisy or hub-associated label groups can survive, invoke multi-target handling, and bypass single-target victim expansion. Conversely, $\tau \geq 10$ can exceed genuine support in low-budget Cora GTA/UGBA settings and skip unlearning. This boundary depends on scale: the reported OGB-arxiv runs retain sufficient support to avoid this failure even at $\tau = 100$.

N Detailed Analysis and Additional Adaptive Attacks

N.1 White-Box Attacks

In this section, we first consider a scenario where the attacker only knows partial information about our defense (e.g., model architecture). Therefore, we introduce a sophisticated variant of the DPGBA attack, $DPGBA_{Partial}$, which aims to hide final output representations of trigger nodes. This is achieved by training a *supervised* GAT model—sharing the same architecture as our defense—to distinguish between trigger nodes and benign nodes based on their final representations.

We further consider a scenario where the attacker fully replicates our defense model and uses it to craft triggers, resulting in $DPGBA_{Full}$. Here, the supervised GNN from $DPGBA_{Partial}$ is replaced by our defense model, allowing the attacker to directly adapt the trigger generator to the defense

Table 18: White-box Attacks

Attack	No Defense		Defense		Node Detection			
					Victim Nodes		Trigger Nodes	
	ASR(%)	CA(%)	ASR(%)	CA(%)	P(%)	R(%)	P(%)	R(%)
Partial	72.26	80.37	2.96	83.19	91.21	100.00	98.11	100.00
Full	98.09	71.56	0.00	68.07	92.55	94.00	100.00	86.00

model's behavior. This provides a stricter test of our defense under the assumption that attackers possess complete knowledge of our defense details. The trigger-node set is fixed at initialization; the adversarial model optimizes node features and intra-trigger edges during training. The Full search repeatedly runs localization and repair and selects candidates using post-repair ASR.

We present the results in Table 18. For both attacks, PRAETORIAN identifies trigger nodes and victim nodes with precision and recall above 90%, except for trigger recall under $DPGBA_{Full}$ (86%). Post-defense ASR is 2.96% for $DPGBA_{Partial}$ and 0.00% for $DPGBA_{Full}$. For $DPGBA_{Full}$, CA drops from 71.56% to 68.07% after defense. A model trained on a clean graph and tested on the poisoned graph also has low CA (66.15%).

The effectiveness of PRAETORIAN can be attributed to two key factors: (1) *Targeting Internal Correlations and External Influences.* PRAETORIAN emphasizes both internal correlations among trigger nodes and their external influence, making mere manipulation of final representations insufficient to hide trigger structures (even in $DPGBA_{Partial}$). (2) *Deviation Score Propagation.* In $DPGBA_{Full}$, each trigger subgraph contains three trigger nodes. Although two nodes actually evade initial detection, the third node consistently exhibits strong correlations, allowing deviation propagation to recover associated trigger candidates. Together with repair, this reduces ASR to 0.00%.

Conclusion: PRAETORIAN is still effective even if the attacker knows the details of our defense models.

N.2 One-Node Clean Label Attack

We are also aware of recent natively clean-label graph backdoor methods [52, 53]. However, to the best of our knowledge, no public implementations were available to us when these experiments were conducted, making direct reproduction infeasible. We therefore construct clean-label variants of UGBA and DPGBA to stress-test the same threat factor under reproducible settings.

To explore this scenario, we consider three variants of UGBA and DPGBA attacks: $UGBA_{CO}$, $DPGBA_{CO}$, and $UGBA_{CO} + DPGBA_{CO}$. In $UGBA_{CO}$, we restrict the trigger size to 1 and select victim nodes whose labels already match the target class. Likewise, in $DPGBA_{CO}$, the trigger size is 1, and victim nodes come solely from the target class. The combined version, $UGBA_{CO} + DPGBA_{CO}$, takes the same setting but simultaneously applies UGBA's objective of increasing similarity among connected nodes and DPGBA's goal of obfuscating the distribution of trigger nodes.

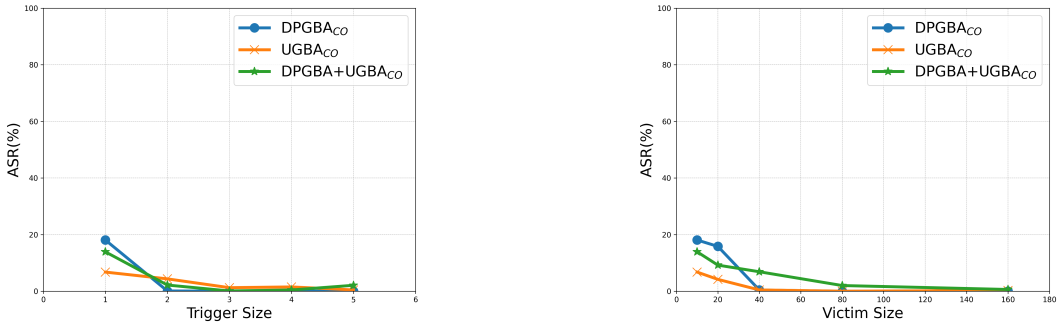
As shown in Table 7, our findings are threefold. First, the initial ASR for $DPGBA_{CO}$ and $UGBA_{CO} + DPGBA_{CO}$ drops sharply from $\geq 90\%$ to 53.7% and 21.6%, respectively. Second, for these two attacks, PRAETORIAN's detection recall for trigger and victim nodes is low (below 30%), and the residual ASRs are 18.09% and 13.91%, respectively. Third, while $UGBA_{CO}$ maintains a high ASR (>90%), PRAETORIAN detects most of its trigger nodes, reducing the post-defense ASR to 6.76%. This is because even though UGBA's triggers are similar to victims, their overall distribution remains distinct from benign nodes, making them detectable.

However, attackers might try enlarging the victim set to increase ASR. To evaluate this, we vary the victim size among {10, 20, 40, 80, 160} for three variants. Table 7 shows that increasing victim size does not notably improve ASR, while PRAETORIAN's detection performance actually rises. Once the victim set exceeds 40, PRAETORIAN detects more than 60% of both victim and trigger nodes with above 90% precision. This occurs because adding more victim nodes inherently introduces more trigger nodes and more attack/trigger patterns; Thus, the influence of the trigger nodes is enhanced and the correlations between trigger and victim nodes are strengthened, so that trigger nodes are more easily detected.

Table 19: Varying Trigger Size of Clean Label Attack

Attack	TS	No Defense		Defense		Node Detection			
						Victim Nodes		Trigger Nodes	
		ASR(%)	CA(%)	ASR(%)	CA(%)	P(%)	R(%)	P(%)	R(%)
DPGBA	1	53.74	82.82	18.09	83.11	50.83	28.00	80.00	26.00
	2	80.26	83.04	0.09	81.85	92.55	98.00	100.00	98.00
	3	94.26	80.74	0.00	80.74	94.55	100.00	100.00	98.67
	4	96.17	79.93	0.00	80.30	95.45	100.00	100.00	98.50
	5	97.65	78.89	0.00	79.63	94.55	100.00	100.00	98.00
UGBA	1	92.18	81.48	6.76	82.89	84.00	84.00	100.00	84.00
	2	92.44	82.30	4.36	83.11	93.33	94.00	100.00	94.00
	3	94.76	80.15	1.24	83.26	90.00	96.00	100.00	96.00
	4	92.09	80.82	1.51	82.37	83.03	98.00	100.00	98.00
	5	93.69	80.67	0.44	80.89	89.70	96.00	100.00	95.00
$DPGBA$ + $UGBA_{CO}$	1	21.57	83.33	13.91	83.56	48.33	20.00	80.00	18.00
	2	52.35	83.30	2.17	83.19	89.24	84.00	98.82	82.00
	3	67.74	83.41	0.09	82.96	86.36	100.00	100.00	95.33
	4	68.70	82.37	0.44	82.59	82.67	86.00	100.00	84.50
	5	70.52	82.89	2.09	82.44	83.80	68.00	100.00	64.00

We also vary trigger sizes on three variants. The results are shown in Table 19. From the table, we observe that as the trigger size increases, the attacker achieves a higher ASR when no defense is applied. However, our defense also shows improved performance, with a higher detection rate for trigger nodes, leading to a lower ASR. This improvement occurs because, as the trigger size increases, the internal correlation among the trigger nodes becomes stronger, enabling us to detect them more effectively based on this higher correlation. These results suggest that our defense is not easily bypassed by simply increasing the trigger size. We also provide the curve (see Figure. 7 of how ASR changes with varying victim size and trigger size.



(a) Varying Trigger Size

(b) Varying Victim Size

Figure 7: Clean Label Attack

Conclusion: Although attackers may try one-node clean-label triggers to reduce the effectiveness of PRAETORIAN, the success of such approaches is limited by significant trade-offs. Even with variations in victim size or trigger size, the tested clean-label attacks achieve no more than 18.1% ASR after defense. Future work could audit one-node triggers at inference, when they can change a non-target node's prediction, or use class-specific confidence thresholds calibrated on clean validation data to identify additional victim candidates.

N.3 Random Noise Disruption Attack

Beyond one-node clean-label attacks, adversaries may try to remove internal correlations among trigger nodes by injecting random noise. This disrupts internal correlation checks and allows triggers larger than one node. To further evade external influence checks, attackers can pair clean-label attack strategies with establishing edges between victim nodes and all trigger nodes. We implement this

Table 20: Detection results of Asymmetric Attacks

p	q	No Defense		Defense Performance		Victim and Trigger Node Detection			
						Victim Nodes		Trigger Nodes	
		ASR(%)	(CA%)	ASR(%)	(CA%)	P(%)	R(%)	P(%)	R(%)
0.5	10	86.05	82.22	0	82.59	100	95	100	95.76
0.5	3	91.24	81.11	0	82.96	100	95	100	97.35
0.8	10	87.64	81.11	0	82.22	100	85	88.24	86.96
0.8	3	88.45	80.74	0	81.85	100	100	97.37	100
0.9	10	72.91	82.59	0	82.59	100	65	100	66.67
Average		85.26	81.55	0	82.44	100	88	97.12	89.34

idea using an SBA attack, where a Gaussian noise with a standard deviation 30 times that of the original features is added to every trigger node.

We vary the trigger sizes {1, 3, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100} and victim sizes {10, 20, 40, 80, 160} on the Cora dataset. We plot ASR and CA curves before and after defense to illustrate how ASR and CA change with different victim and trigger sizes.

As shown in Figure 4, increasing either the trigger or the victim size can raise the ASR. Some high-ASR configurations retain clean accuracy close to the undefended model.

What's more, we present additional results of our random noise disruption attack on the **PubMed dataset**.

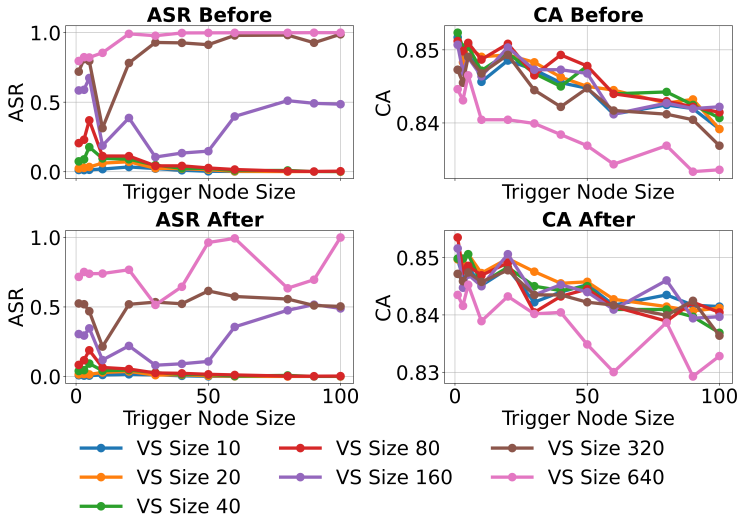


Figure 8: ASR and CA of Random Disruption Attack

As shown in Figure 8, the results align with those in Section 6. On PubMed, the attacker can achieve high ASR with a large victim size (e.g., 640) and large noisy triggers. Such substantial pattern insertions motivate testing simple degree or local-volume checks.

N.4 Asymmetric Trigger Insertion Strategy

Recent studies in the image backdoor domain [54, 55] have suggested that *asymmetric triggers* can effectively reduce the latent separability of backdoor attacks, thereby evading defenses. Inspired by this insight, we propose an *Asymmetric Graph Backdoor* that leverages asymmetry to make triggers in the training set less detectable.

Our strategy focuses on two primary aspects to reduce trigger visibility: trigger sizes and trigger features. 1. *Trigger sizes.* For each added trigger node, we assume a removal probability p during the training phase, which also removes the edges connected to the node. This reduces the effective

strength of the trigger at the node level. When $p = 0$, the trigger node remains intact, representing a full-strength poison. Conversely, when $p = 1$, all trigger nodes are removed, resulting in no poisoning effect. 2. *Trigger Features.* To diminish feature strength, we introduce a random reduction factor α, such that the training feature value is given by $h_{\text{train}} = \alpha \times h$. Here, α is a random variable defined as $\alpha = \text{Rand}(0, 1)^q$, where q is a strength factor controlling the magnitude of the reduction. When $q = 0$, $\alpha = 1$, resulting in full-strength features. As q increases, α approaches 0, leading to negligible feature strength. For experiments, we implement this strategy with the SBA attack. We use a victim size 40 and a trigger size of 5; then we vary p and q to acquire different strengths of the attack.

By tuning p and q, we can precisely control the strength of the training triggers in terms of both nodes and features. During the testing stage, trigger nodes and features are set to full strength to ensure attack success.

Conclusion. The results are shown in Table 20. Based on Table 20, PRAETORIAN remains robust against the asymmetric trigger insertion strategy, achieving at least 97% mean precision and 88% mean recall for both victims and triggers, and a 0% ASR while maintaining higher mean CA than the no-defense baseline. These findings underscore the effectiveness and resilience of our PRAETORIAN in defending against diverse backdoor attacks.

N.5 Multi-target Backdoor Attacks

This setting stress-tests the downstream score-to-node localization stage rather than the core two-view scoring mechanism. In standard single-target attacks, poisoned training nodes usually share one target label, so the single-target refinement in Appendix F can exploit label consistency along the sorted score list. Multi-target attacks [56] instead inject simultaneous backdoors with different target labels, so high-deviation victim seeds may be spread across several observed label groups. This does not change the internal-correlation or external-influence scores: trigger nodes remain structurally and functionally abnormal regardless of how many target labels are used. It only changes how the high-deviation seeds are converted into victim groups. Following Appendix F, PRAETORIAN first applies the label-free valley cutoff, then retains multiple supported label groups and performs trigger recovery and unlearning group-wise, without enforcing the single-target stopping rule.

This same reasoning also covers ALL2ALL-style attacks [57] . Such attacks instantiate multiple simultaneous trigger-target mappings, but their localization challenge is still to recover several suspicious target groups from the same score distribution. Since PRAETORIAN groups high-deviation seeds after the label-free cutoff, it does not require knowing the number of target labels or the source-to-target mapping.

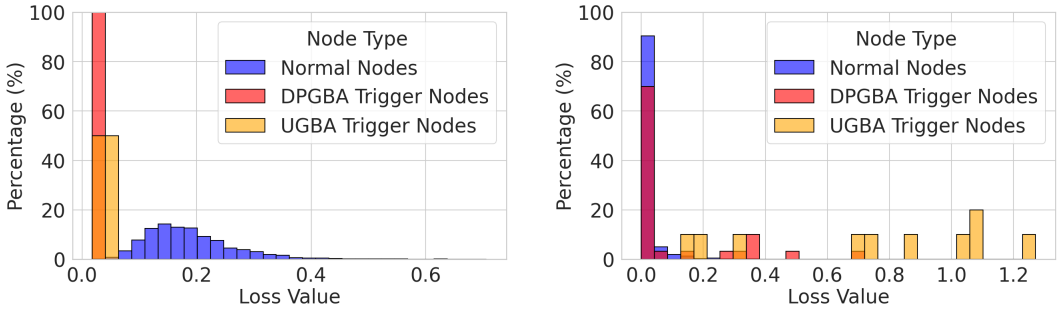
Results. Table 21 reports the defense results under multi-target attacks. DPGBA is reduced from 90.4% ASR to 0.0%, and UGBA from 96.0% ASR to 2.2%, while CA is maintained after defense. Localization precision and recall both exceed 92% for victims and triggers under both attacks. As shown in Figure. 9, PRAETORIAN assigns high deviation scores to trigger nodes under both internal correlation and external influence across target labels, confirming that the core two-view detector is agnostic to the number of target labels; only the lightweight score-to-node conversion needs to be read group-wise.

Table 21: Defense and localization results under multi-target attacks.

Attack	ASR Before	ASR After	CA Before	CA After	Victim Nodes. P/R	Trigger Nodes. P/R
DPGBA	90.4%	0.0%	81.8%	82.9%	92.1%/100.0%	92.5%/94.5%
UGBA	96.0%	2.2%	81.8%	82.9%	94.2%/96.0%	93.3%/92.5%

N.6 Trade-offs for the Attacker

We further examine the trade-offs faced by adaptive attackers under defense-aware settings. The evaluated attacks directly target the main components of PRAETORIAN, including internal-correlation scoring, external-influence scoring, and score-to-node localization. While these experiments do not exhaust all possible adaptive strategies, they cover natural ways to weaken the signals used by our defense.



(a) Internal Correlation (b) External Influence

Figure 9: Two deviation scores under multi-target attacks.

Table 22: Runtime Cost

Dataset	Score Calculation	Node Selection	Relearning
Cora	43.21s	0.04s	1.55s
PubMed	700.28s	0.80s	2.00s
OGB-arxiv	1560.11s	40.02s	15.83s

Our results indicate that evading PRAETORIAN can involve a trade-off between attack effectiveness and stealthiness. Post-defense ASR reaches at most 18.1% for the clean-label variants we test. Under random disruption, increasing the attack scale, such as using larger trigger or victim sets, can yield substantially higher ASR even when clean accuracy remains close to that of the undefended model.

O Discussion

Runtime cost and complexity optimization. Theoretically, our approach utilizes the Deep Graph Library (DGL) [58] for GNN training, which performs message passing on an edge-by-edge basis. For a GCN layer with hidden dimension H, N nodes, and E edges, the per-layer time complexity is $\mathcal{O}(NH^2 + EH)$. During our initial defense implementation, node masking necessitates inference for each candidate node, resulting in an overall complexity of $\mathcal{O}(N \times (NH^2 + EH))$. To enhance efficiency, we optimized the computation by introducing PRAETORIAN-Faster. Firstly, this method updates *only* the masked nodes and their neighbors, circumventing the recalculation of the entire graph. Consequently, the complexity for processing a single node at the L-th layer is reduced from $\mathcal{O}(NH^2 + EH)$ to $\mathcal{O}(N_s H^2 + E_s H)$, where N_s and E_s denote the number of nodes and edges, respectively, within the L-hop subgraph centered on the masked node. This optimization significantly reduces the total computational cost (e.g., reduces detection time to 5% of its original on OGB-arxiv). Detailed runtime costs are provided in Table 22.

Secondly, by focusing on suspicious nodes (i.e., training nodes and neighbors within the model's depth layer), PRAETORIAN-Faster can further limit the analysis scope, enhancing efficiency (e.g., 25% fewer nodes processed on OGB-arxiv). For scenarios where timely updates are needed, trigger detection can be performed only on updated training-related nodes, avoiding full graph reprocessing.

Nevertheless, additional methods, such as multi-process inference and sampling methods, could further accelerate our system. We leave these for future work.

Memory overhead.

Table 23: Peak GPU memory under the standard configuration.

Dataset	#Nodes	Peak GPU Memory
Cora	2,708	0.83 GB
PubMed	19,717	4.47 GB
OGB-arxiv	169,343	10.7 GB

We further report peak GPU memory under our standard configuration in Table 23. A $62.5\times$ increase in node count from Cora to OGB-arxiv yields only a $12.9\times$ increase in peak memory, remaining within the capacity of a single GPU.

Application scope. Although PRAETORIAN is designed for defending node-level graph backdoor attacks, its internal correlation—which captures relationships among trigger nodes within a subgraph—is not limited to node-level tasks. Even for graph-level backdoor attacks (e.g., SBA and GTA can also be launched to graph-level attacks) that replace a subgraph of the original clean graph with a trigger subgraph, PRAETORIAN can still quantify the internal correlations among the injected trigger nodes. Meanwhile, the external influence could be adapted to measure the impact of each subgraph rather than individual nodes. Consequently, PRAETORIAN has the potential to defend against graph-level backdoor attacks, and we plan to explore this in future work.

P Impact Statements

This paper presents PRAETORIAN, a new defense paradigm against backdoor attacks on Graph Neural Networks (GNNs). By jointly auditing two complementary signals—*internal correlation* within potential trigger subgraphs and *external influence* of suspicious nodes on neighborhood predictions—our method aims to localize and neutralize backdoor behavior via filtered training and targeted unlearning. If adopted in practice, PRAETORIAN could improve the safe deployment of GNNs in security- and safety-relevant applications (e.g., fraud detection, recommendation, and provenance/security graphs), where stealthy poisoning can cause systematic misclassification while remaining hard to diagnose.

However, PRAETORIAN also has limitations and potential negative impacts. As a defense study, our analysis and adaptive-attack discussion may inadvertently inform adversaries about which trigger designs are harder to audit, accelerating evasion research. In addition, because our auditing relies on correlation/influence deviations, naturally high-influence nodes (e.g., hubs, minority-pattern nodes, or distribution-shifted regions) could be flagged, leading to false positives and possible unfair performance degradation if unlearning is applied to benign nodes. Furthermore, as demonstrated in our analysis of adaptive clean-label attacks (Section 6), extremely stealthy settings such as a *one-node clean-label* trigger can weaken the auditing signals and partially bypass PRAETORIAN, raising the Attack Success Rate (e.g., up to 18.1%). While our work focuses on defense, we encourage future research to investigate how to evaluate and improve robustness under stronger adaptive clean-label strategies, to ensure equitable and reliable security across diverse threat landscapes.

Q Theoretical Framework

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with features $\mathcal{X}$, and f be a GNN model with r layers that computes a logits $H_v(\mathcal{G}) \in \mathbb{R}^C$ for node v in graph $\mathcal{G} = (\mathcal{G}, \mathcal{X})$, where C is the number of classes. A backdoor attack inserts a trigger $\mathcal{T}$—a set of structural or feature perturbations—into a clean graph $\mathcal{G}_C$ to create $\mathcal{G}_T$, aiming to force v into a target class. Our goal is to decompose this influence and motivate complementary detection scores.

Q.1 Decomposition of Anomalous Influence

We define the influence of any subgraph components S on v's *predicted logits*, where S within v's r-hop neighborhood. We use an interventional operation $\mathcal{G}_C \oplus \mathsf{do}(S)$, which represents intervening on v's r-hop neighborhood by applying the perturbation S (adds S if absent; removes S if present). The resulting change is:

$$\Delta_H(S) \;=\; H_v\big(\mathcal{G}_C \oplus \mathsf{do}(S)\big) \;-\; H_v(\mathcal{G}_C)$$

By this, $\Delta_H(\emptyset) = \mathbf{0}$, and the total trigger influence is $\Delta_H(\mathcal{T})$.

Our theoretical goal is to prove that the trigger's influence cannot be simultaneously negligible in both its individual components (EI) and its synergistic structure (IC).

We build on the established assumption [28] that a high-ASR attack tends to induce a "large prediction variance," meaning its influence is non-trivial. We formalize this minimal assumption as our premise.

Assumption Q.1 (Non-Trivial Influence)**.** A high-ASR trigger $\mathcal{T}$ typically induces a perturbation $\Delta_H(\mathcal{T})$ whose magnitude is non-trivial, i.e., it is bounded below by a detectable, non-trivial constant $\Theta_{\text{detectable}} \gg 0$:

$$\left\|\Delta_H(\mathcal{T})\right\|_2 \;\geq\; \Theta_{\text{detectable}} \tag{8}$$

Then we attribute the total influence $\Delta_H(\mathcal{T})$ using Harsanyi dividends [59], an application of Möbius inversion to set functions. We model $\Delta_H(\mathcal{T})$ as a set function $v : 2^{\mathcal{T}} \to \mathbb{R}^C$. The total influence uniquely decomposes into dividends $\Phi_S = \Delta_H(S) - \sum_{R \subset S} \Phi_R$, such that $\Delta_H(\mathcal{T}) = \sum_{\emptyset \neq S \subseteq \mathcal{T}} \Phi_S$.

We attribute this sum to two disjoint sources: **1. External Influence (EI):** The sum of first-order dividends. This captures the *additive* effect of individual trigger components. **2. Internal Correlation (IC):** The sum of all higher-order ($|S| \geq 2$) dividends. This captures the *non-additive, synergistic* effects emerging from the components' structure.

$$EI(\mathcal{T}) = \sum_{t \in \mathcal{T}} \Phi_{\{t\}}, \quad IC(\mathcal{T}) = \sum_{\substack{S \subseteq \mathcal{T} \\ |S| \geq 2}} \Phi_S$$

By definition, these two components sum to the total perturbation:

$$\Delta_H(\mathcal{T}) \;=\; EI(\mathcal{T}) \;+\; IC(\mathcal{T}) \tag{9}$$

This decomposition leads to a key theoretical guarantee.

Theorem Q.2 (Influence Dichotomy)**.** *If a trigger $\mathcal{T}$ satisfies the Non-Trivial Influence premise (Eq. 8), then its influence* must *be present in at least one of its components at a non-trivial level:*

$$\left\|EI(\mathcal{T})\right\|_2 \;\geq\; \tfrac{1}{2}\Theta_{\textit{detectable}} \quad \textit{or} \quad \left\|IC(\mathcal{T})\right\|_2 \;\geq\; \tfrac{1}{2}\Theta_{\textit{detectable}}$$

Proof. Assume for contradiction that both $\|EI(\mathcal{T})\|_2 < \frac{1}{2}\Theta_{\text{detectable}}$ and $\|IC(\mathcal{T})\|_2 < \frac{1}{2}\Theta_{\text{detectable}}$. By the triangle inequality on the decomposition (Eq. 9):

$$\begin{aligned}\left\|\Delta_H(\mathcal{T})\right\|_2 &= \left\|EI(\mathcal{T}) + IC(\mathcal{T})\right\|_2 \\ &\leq \left\|EI(\mathcal{T})\right\|_2 + \left\|IC(\mathcal{T})\right\|_2\end{aligned}$$

Substituting the assumptions yields:

$$\left\|\Delta_H(\mathcal{T})\right\|_2 \;<\; \tfrac{1}{2}\Theta_{\text{detectable}} + \tfrac{1}{2}\Theta_{\text{detectable}} \;=\; \Theta_{\text{detectable}}$$

This result, $\left\|\Delta_H(\mathcal{T})\right\|_2 < \Theta_{\text{detectable}}$, directly contradicts Premise Q.1. Thus, the assumption is false. □

Implication: Theorem Q.2 provides the theoretical justification for our defense. It proves that the non-trivial influence of a trigger *cannot* simultaneously vanish from both its individual (EI) and synergistic (IC) components. This motivates measuring both channels. Appendix Q.3 gives conditions under which the corresponding practical scores reveal this influence.

Q.2 From Theoretical Components to Practical Proxy Scores

Our theoretical framework, culminating in Theorem Q.2, shows that a trigger $\mathcal{T}$ satisfying Premise Q.1 must induce non-negligible influence through at least one of two channels: the first-order external-influence component $EI(\mathcal{T})$ or the higher-order internal-correlation component $IC(\mathcal{T})$. For a known victim and an m-component trigger, the aggregate components require only the empty, singleton, and full coalitions (at most $m + 2$ evaluations), whereas all dividends require 2^m. At deployment the trigger is unknown, so searching candidate component sets is combinatorial.

PRAETORIAN therefore does not attempt to recover the exact theoretical quantities. Instead, it employs two efficient node-level proxy scores, $\mathcal{S}_{ext}$ and $\mathcal{S}_{int}$, to empirically audit the two corresponding influence channels. Specifically, $\mathcal{S}_{ext}$ probes whether a node has unusually large individual influence on its local neighborhood, while $\mathcal{S}_{int}$ probes whether a node is embedded in a strongly predictable local pattern. Below, we explain how these two practical scores are aligned with the theoretical components.

Q.2.1 $\mathcal{S}_{ext}$ as a Proxy for External Influence

The theoretical EI component captures the first-order, individual contribution of trigger nodes to the victim's prediction. If EI is nonzero, at least one trigger node has nonzero individual influence. Appendix Q.3 gives conditions under which masking such a node changes local predictions.

Definition. $\mathcal{S}_{ext}$ is computed by masking a single node v and measuring the resulting shift in the softmax probability distributions of its 1-hop neighbors using KL and JS divergences.

Proxy intuition. This mask-and-measure intervention provides an efficient empirical estimate of how much node v individually contributes to nearby predictions. Although this score is not identical to the Harsanyi first-order dividend, it targets the same failure mode: if an attack concentrates backdoor influence on a small number of trigger nodes, masking such a node should induce an unusually large prediction shift in its neighborhood. Thus, $\mathcal{S}_{ext}$ operationalizes the external-influence channel in a scalable node-level form.

Practical caveat. Masking node v may also perturb higher-order interactions involving v. Appendix Q.4 compares the deployed score with the exact singleton influence.

Q.2.2 $\mathcal{S}_{int}$ as a Proxy for Internal Correlation

The theoretical IC component captures higher-order, synergistic influence arising from the joint pattern of trigger components. An IC-dominant attack spreads the backdoor effect across multiple trigger nodes, so the attack relies less on any single highly influential node and more on the collective structure or feature pattern formed by the trigger.

Definition. $\mathcal{S}_{int}$ is computed as the inverse of the masked-node reconstruction loss from a masked graph autoencoder: nodes that are easier to reconstruct from their surrounding context receive higher $\mathcal{S}_{int}$ scores.

Proxy intuition. A masked graph autoencoder is trained to capture statistical dependencies and neighborhood predictability. If a trigger subgraph forms a tightly coupled and distinctive pattern, then masking one trigger node still leaves strong contextual evidence from the remaining trigger-related nodes. As a result, the masked trigger node tends to have a low reconstruction loss. By taking the inverse of this loss in Eq. 3, PRAETORIAN assigns a high $\mathcal{S}_{int}$ score to nodes embedded in such strongly correlated local patterns.

Connection to theory. $\mathcal{S}_{int}$ does not explicitly compute the higher-order Harsanyi dividends in $IC(\mathcal{T})$. Rather, it serves as a scalable proxy for local dependency. Appendix Q.3 gives sufficient conditions for joint influence to appear in this score. Together with $\mathcal{S}_{ext}$, this gives PRAETORIAN two complementary probes: one for concentrated individual influence and one for distributed pattern-level influence.

Q.3 Conditional Score Guarantees in Structured Settings

This section supplements the logit-vector framework in Appendix Q with explicit sufficient conditions for score observability. We distinguish four objects: the original logit-vector (EI, IC), probability-vector (EI_v^p, IC_v^p), target-coordinate audit quantities (Φ_A^F, e_F, IC_F), and the deployed scores $(\mathcal{S}_{\text{ext}}, \mathcal{S}_{\text{int}})$. Appendix Q.4 tests the deployed scores beyond these structured settings.

Q.3.1 Probability-space decomposition and singleton carriers

Fix a poisoned classifier f, victim v, trigger set $\mathcal{T}$, $m = |\mathcal{T}|$, and C classes. Use a fixed clean baseline $\mathcal{G}_C$ and the intervention of Appendix Q.1:

$$\mathcal{G}_A := \mathcal{G}_C \oplus \text{do}(A), \qquad \mathcal{G}_\varnothing := \mathcal{G}_C, \tag{10}$$

$$p_v^f(\mathcal{G}) := \text{softmax}(H_v^f(\mathcal{G})), \qquad \Delta P_v(A) := p_v^f(\mathcal{G}_A) - p_v^f(\mathcal{G}_\varnothing). \tag{11}$$

Every intervention applies exactly the components in A to the same clean context, with non-trigger content fixed. Define the coordinate-wise Harsanyi dividends by

$$\Phi_v^p(A) = \Delta P_v(A) - \sum_{B \subsetneq A} \Phi_v^p(B), \qquad \Phi_v^p(\varnothing) = 0. \tag{12}$$

Then

$$EI_v^p(\mathcal{T}) = \sum_{t\in\mathcal{T}} \Phi_v^p(\{t\}), \qquad IC_v^p(\mathcal{T}) = \sum_{\substack{A\subseteq\mathcal{T}\\ |A|\geq 2}} \Phi_v^p(A), \tag{13}$$

$$\Delta P_v(\mathcal{T}) = EI_v^p(\mathcal{T}) + IC_v^p(\mathcal{T}). \tag{14}$$

The triangle inequality yields

$$\max\{\|EI_v^p(\mathcal{T})\|_2, \|IC_v^p(\mathcal{T})\|_2\} \geq \tfrac{1}{2}\|\Delta P_v(\mathcal{T})\|_2. \tag{15}$$

Lemma Q.3 (Singleton carrier)**.** *If $\|EI_v^p(\mathcal{T})\|_2 \geq E > 0$, there exists $t^\star \in \mathcal{T}$ such that $\|\Phi_v^p(\{t^\star\})\|_2 \geq E/m$.*

Proof. By the triangle inequality,

$$E \leq \left\|\sum_{t\in\mathcal{T}} \Phi_v^p(\{t\})\right\|_2 \leq \sum_{t\in\mathcal{T}} \|\Phi_v^p(\{t\})\|_2 \leq m \max_{t\in\mathcal{T}} \|\Phi_v^p(\{t\})\|_2.$$

□

Q.3.2 External score observability

For a reference version of the external probe, let f' be the scoring predictor, and let $p_j^{f'}$ and $q_{j,t}^{f'}$ denote its probability distributions before and after masking t. Define

$$\ell_{j,t}^{f'} = \frac{1}{C}\left[\mathrm{KL}(p_j^{f'} \| q_{j,t}^{f'}) + R(p_j^{f'}, q_{j,t}^{f'})\right], \tag{16}$$

$$\mathcal{S}_{\mathrm{ext}}^{(0)}(t) = \frac{1}{|\mathcal{N}(t)|} \sum_{j\in\mathcal{N}(t)} \ell_{j,t}^{f'}, \tag{17}$$

where $R \geq 0$ (the manuscript uses JS). KL uses natural logarithms. The factor $1/C$ conservatively covers class averaging; summed KL gives a stronger floor.

Proposition Q.4 (Additive observable carrier)**.** *Assume $\|EI_v^p(\mathcal{T})\|_2 \geq E > 0$ and let $t^\star$ be a carrier from Lemma Q.3. Suppose (i) every higher-order probability dividend vanishes; (ii) $f' = f$; (iii) masking $t^\star$ realizes coalition deactivation; and (iv) $v \in \mathcal{N}(t^\star)$ with $|\mathcal{N}(t^\star)| \leq D$. Then*

$$\mathcal{S}_{\mathrm{ext}}^{(0)}(t^\star) \geq \frac{E^2}{CDm^2}. \tag{18}$$

Proof. Additivity makes the full-context removal effect equal to $\Phi_v^p(\{t^\star\})$. Predictor and intervention matching give $\delta = p_v^{f'} - q_{v,t^\star}^{f'}$ with $\|\delta\|_2 \geq E/m$. Let $a = \sum_{\delta_c>0} \delta_c$. Since $\sum_c \delta_c = 0$, we have $\|\delta\|_1 = 2a$ and $\|\delta\|_2^2 \leq 2a^2$, so $\|\delta\|_1^2 \geq 2\|\delta\|_2^2$. Pinsker's inequality and $R \geq 0$ imply

$$\ell_{v,t^\star}^{f'} \geq \frac{1}{2C}\|\delta\|_1^2 \geq \frac{1}{C}\|\delta\|_2^2 \geq \frac{E^2}{Cm^2}.$$

Averaging nonnegative contributions over at most D scored neighbors proves the result. □

The proposition bounds the carrier's score. Separating it from benign nodes would also require a bound on their scores.

Corollary Q.5 (One-node, single-attachment case)**.** *Under the predictor and masking conditions above, if $\mathcal{T} = \{t\}$ and v is the only scored neighbor, then $m = D = 1$, $IC_v^p(\mathcal{T}) = 0$, and*

$$\mathcal{S}_{\mathrm{ext}}^{(0)}(t) \geq \frac{1}{C}\|EI_v^p(\mathcal{T})\|_2^2 \geq \frac{E^2}{C}.$$

Relaxing exact matching. Let $r_t^f = p_v^f(\mathcal{G}_\mathcal{T}) - p_v^f(\mathcal{G}_{\mathcal{T}\setminus\{t\}})$. Suppose each instance has $\|EI_v^p(\mathcal{T})\|_2 \geq E$, an observable carrier $t^\star$ with $|\mathcal{N}(t^\star)| \leq D$, and uniform $\kappa_0, \alpha_0 > 0$ satisfying

$$\|r_{t^\star}^f\|_2 \geq \kappa_0 \|\Phi_v^p(\{t^\star\})\|_2, \tag{19}$$

$$\|p_v^{f'} - q_{v,t^\star}^{f'}\|_2 \geq \alpha_0 \|r_{t^\star}^f\|_2. \tag{20}$$

The same proof gives $\mathcal{S}_{\mathrm{ext}}^{(0)}(t^\star) \geq \alpha_0^2\kappa_0^2 E^2/(CDm^2)$. Here κ_0 quantifies retention under full-context removal and α_0 quantifies scoring/intervention fidelity.

Q.3.3 Internal score observability

Define an equal-share allocation of higher-order dividends,

$$\psi_{v,t}^{p,IC} = \sum_{\substack{A\subseteq\mathcal{T}:\\ t\in A,\ |A|\geq 2}} \frac{\Phi_v^p(A)}{|A|}. \tag{21}$$

Summing over t gives $\sum_t \psi_{v,t}^{p,IC} = IC_v^p(\mathcal{T})$. Therefore $\|IC_v^p(\mathcal{T})\|_2 \geq I > 0$ implies a carrier $t^\dagger$ with $\|\psi_{v,t^\dagger}^{p,IC}\|_2 \geq I/m$. This auxiliary allocation is not a unique node-level definition of IC.

Let $g = f_D \circ f_E$ be a GraphMAE map, Z_t the visible same-graph context after masking t, and $\widehat{g}$ the fitted map. Define

$$L_t(g) = \big(1 - \cos(x_t, g(Z_t))\big)^\gamma, \quad \gamma > 1, \qquad \mathcal{S}_{\text{int}}^{(0)}(t) = \frac{1}{L_t(\widehat{g})}. \tag{22}$$

Proposition Q.6 (Pure-joint, context-identifiable carrier)**.** *Let $m \geq 2$. Assume (i) $\Phi_v^p(A) = 0$ for every nonempty proper subset $A \subsetneq \mathcal{T}$, while $\|\Phi_v^p(\mathcal{T})\|_2 \geq I > 0$; (ii) the masked feature and ID do not leak; (iii) the remaining trigger context lies within the receptive field; (iv) a map $g^\star$ in the GraphMAE class satisfies $L_{t^\dagger}(g^\star) \leq r_T$; and (v) the same carrier satisfies $L_{t^\dagger}(\widehat{g}) - L_{t^\dagger}(g^\star) \leq \varepsilon_{\text{fit}}$, with $r_T, \varepsilon_{\text{fit}} \geq 0$. Then every node is an IC carrier,*

$$\psi_{v,t}^{p,IC} = \Phi_v^p(\mathcal{T})/m, \qquad \|\psi_{v,t}^{p,IC}\|_2 \geq I/m,$$

and the context-identifiable carrier obeys

$$L_{t^\dagger}(\widehat{g}) \leq r_T + \varepsilon_{\text{fit}}, \qquad \mathcal{S}_{\text{int}}^{(0)}(t^\dagger) \geq \frac{1}{r_T + \varepsilon_{\text{fit}}}. \tag{23}$$

The reference score is $+\infty$ for a zero denominator.

Proof. Only the full-coalition dividend remains, giving $\psi_{v,t}^{p,IC} = \Phi_v^p(\mathcal{T})/m$. The pointwise approximation and fit conditions give the loss bound; reciprocation gives the score bound. □

IC supplies a carrier, while context identifiability and pointwise fitting make that same carrier reconstructible.

Concrete two-layer witness. Let $\mathcal{T}$ be an m-node clique ($m \geq 2$) with a shared nonzero feature z. Only attachment node t_0 is adjacent to victim v. For a coalition A, let $b_A = \mathbf{1}\{t_0 \in A\}$ and define

$$g(A) = \max\{\beta b_A |A| - \theta, 0\}, \tag{24}$$

$$H_v^f(\mathcal{G}_A) = H_v^f(\mathcal{G}_\varnothing) + g(A) e_{y_{\text{tar}}}, \tag{25}$$

where $\beta > 0$, $(m-1)\beta \leq \theta < m\beta$, and $e_{y_{\text{tar}}}$ selects the target coordinate. If $t_0 \notin A$, then $b_A = 0$; if $t_0 \in A \subsetneq \mathcal{T}$, then $|A| \leq m-1$. In either case $g(A) = 0$. Only the full coalition has $g(\mathcal{T}) = m\beta - \theta > 0$. Thus $\Delta P_v(A) = 0$ for every proper coalition but $\Delta P_v(\mathcal{T}) \neq 0$, yielding $EI_v^p(\mathcal{T}) = 0$ and $IC_v^p(\mathcal{T}) = \Delta P_v(\mathcal{T})$ by Möbius inversion.

A two-layer sum-aggregation ReLU GNN realizes the gate by counting active trigger nodes at t_0 and sending the gated increment to v. For a non-attachment carrier $t^\dagger \neq t_0$, the peer-mean map reconstructs z exactly. Hence $r_T = 0$ if this map belongs to the GraphMAE class, and the proposition applies under the stated pointwise fitting condition. The threshold creates joint influence; identical peers create context identifiability. The map trains on the same graph, so repetition across different graphs is unnecessary.

Q.4 Exact-coalition audit of the deployed scores

To test how well the practical scores rank nodes and groups by exact influence, we first computed $\mathcal{S}_{\text{ext}}$ and $\mathcal{S}_{\text{int}} = 1/\mathcal{L}_{\text{int}}$, fixing their aggregation before inspecting oracle quantities. For a scalar, prediction-relevant counterpart to the logit-vector game, we used the target-class probability

$$F_v(A) = \big[\,\text{softmax}(H_v^f(\mathcal{G}_C \oplus \text{do}(A)))\big]_{y_{\text{tar}}}, \quad \Delta F_v(A) = F_v(A) - F_v(\varnothing). \tag{26}$$

The exact scalar dividends and channel sums are

$$\Phi_A^F = \Delta F_v(A) - \sum_{B \subsetneq A} \Phi_B^F, \qquad \Phi_\varnothing^F = 0, \tag{27}$$

$$EI_F(\mathcal{T}) = \sum_{t \in \mathcal{T}} \Phi_{\{t\}}^F, \qquad IC_F(\mathcal{T}) = \sum_{\substack{A \subseteq \mathcal{T} \\ |A| \geq 2}} \Phi_A^F. \tag{28}$$

For each three-node trigger, we evaluated all 2^3 coalitions with the same frozen poisoned classifier and the Appendix Q.1 intervention. Every coalition applies exactly its selected components to the same clean context, holding non-trigger content fixed. Singleton dividends measure individual effects; higher-order dividends include effects requiring joint presence, including connectivity-dependent effects.

Singleton dividends have a unique node-level definition, so the external audit compares $\mathcal{S}_{\text{ext}}(t)$ with $e_F(t) = |\Phi_{\{t\}}^F|$. Group-level $IC_F(\mathcal{T})$ has no unique node allocation. We therefore pre-specified

$$\overline{\mathcal{S}}_{\text{int}}(\mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \mathcal{S}_{\text{int}}(t)$$

and compared this group mean with $|IC_F(\mathcal{T})|$.

Protocol and results. The audit covers SBA, GTA, UGBA, and DPGBA on Cora and PubMed, with five seeds: 40 runs and 1,000 victim-trigger groups. Size-matched benign subgraphs were evaluated by the same exact-coalition procedure. We computed one Spearman coefficient per run over trigger and matched-benign instances, then aggregated the run-level coefficients with bootstrap 95% confidence intervals (CIs). Table 24 gives the two intended associations; Table 25 gives the reported condition-level correlations.

Table 24: Exact-coalition audit: overall Spearman rank associations.

Mapping	ρ	Bootstrap 95% CI
$\mathcal{S}_{\text{ext}}(t) \leftrightarrow e_F(t)$	0.66	[0.59, 0.72]
$\overline{\mathcal{S}}_{\text{int}}(\mathcal{T}) \leftrightarrow \lvert IC_F(\mathcal{T})\rvert$	0.61	[0.53, 0.67]

Table 25: Reported condition-level Spearman correlations.

Dataset	Mapping	SBA	UGBA	GTA	DPGBA
Cora	External (node)	0.55	0.50	0.62	0.49
Cora	Internal (group)	0.58	0.52	0.59	0.44
PubMed	External (node)	0.51	0.85	0.89	0.85
PubMed	Internal (group)	0.68	0.86	0.53	0.67

Both mappings are positive in all 40 runs, and the condition-level 95% CI lower bounds exceed zero in all eight dataset-attack conditions for each mapping. These coefficients describe rank associations in the pooled trigger and matched-benign sample.

Oracle use and cost. The scalar identity implies $\max_{t \in \mathcal{T}} |\Phi_{\{t\}}^F| \geq |EI_F(\mathcal{T})|/|\mathcal{T}|$. Exact quantities and benign controls are used only in this offline audit. Given an oracle victim and an m-component trigger, the aggregate EI_F and IC_F require at most $m + 2$ evaluations (the empty, singleton, and full coalitions); all dividends require 2^m evaluations. At deployment, the victim, trigger set, size, topology, and target are unknown, and d candidate components have $2^d - 1$ possible nonempty coalitions.

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 abstract and introduction state the defense problem, the dual internal-correlation and external-influence design, and the main empirical claims. The reported claims are tied to the experiments across four benchmark datasets, multiple backdoor attacks, and adaptive-attack analyses.

   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 Appendix P discusses limitations and potential negative impacts, including adaptive clean-label settings, possible false positives on naturally high-influence nodes, and dual-use concerns. The Appendix N further analyzes failure modes and attacker trade-offs under adaptive attacks.

   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 theoretical discussion Q states the threat-model assumptions and formalizes the internal-correlation/external-influence intuition in the methodology and appendix. The appendix provides the supporting arguments and connects them to the empirical security analyses.

   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: The paper describes datasets, attacks, baselines, train/test splits, metrics, model architectures, and key hyperparameters in the evaluation section and appendix. It also reports attack settings and adaptive-attack configurations needed to interpret and reproduce the main claims.

   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: [No]

   Justification: The review archive contains a partial implementation. We will release the defense code at `https://github.com/Joney-Yf/Graph_backdoor-Defense-Praetorian`.

   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: The evaluation section specifies the 80/20 graph split, ASR/CA evaluation protocol, five-run averaging, datasets, attacks, baselines, and key hyperparameters such as architecture, learning rate, mask rate, and sensitivity parameter. Additional attack, defense, ablation, and adaptive settings are provided in the appendix.

   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: [No]

Justification: The main ASR/CA tables report five-run means without error bars. Appendix Q.4 reports bootstrap 95% confidence intervals for the run-level correlations.

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: [No]

   Justification: The appendix O reports runtime costs, peak GPU memory, complexity analysis, and the PRAETORIAN-Faster optimization. Total compute is not reported.

   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 a defensive security study using public graph benchmarks and previously studied attack settings, without collecting private data or involving human subjects. The paper also discusses dual-use risks and limitations associated with adaptive backdoor analysis.

   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: The Appendix P discusses positive security impacts for GNNs in sensitive applications and explicitly notes negative impacts, including dual-use information that could inform adaptive attackers. It also identifies false-positive and fairness-related risks when benign high-influence nodes are flagged.

    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 high-risk pretrained models, generative models, or scraped datasets. The security analysis focuses on defenses and evaluates attacks only in controlled benchmark settings.

    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: The paper cites the benchmark datasets, DGL, and the original attack and defense papers whose public implementations or protocols are used. The anonymized code and data package provides the relevant asset sources and license/terms information for reused datasets, code dependencies, and implementations, and our use follows the corresponding public licenses and 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: [No]

Justification: The paper does not specify a license for the defense implementation.

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 research does not involve crowdsourcing, user studies, or human subjects. All experiments are conducted on public graph benchmark datasets and simulated backdoor attacks.

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: No human-subject research, crowdsourcing, or participant data collection is conducted. Therefore, IRB approval or equivalent review is not applicable to this work.

    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: [N/A]

    Justification: The core method does not use LLMs as a model component, experimental subject, data generator, or scientific contribution. Any ordinary writing or formatting assistance, if used, is outside the scope requiring declaration under this checklist item.

    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.