ECHOGHOST: A FREQUENCY-STRUCTURAL BACKDOOR ATTACK AND SPECTRAL-COUNTERFACTUAL DEFENSE FOR GRAPH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph neural networks (GNNs) are increasingly deployed in security-sensitive domains, yet their message-passing mechanism can amplify hidden training-time perturbations and convert small poisoned patterns into systematic inference-time failures. Existing graph backdoor attacks typically rely on subgraph triggers, node injection, or local feature manipulation, which can leave detectable topological or feature-space artifacts. This paper proposes *EchoGhost*, a frequency-structural backdoor framework for node classification in GNNs. EchoGhost embeds a universal echo trigger into selected node features through controlled frequency-band modulation and reinforces the trigger by injecting similarity-guided ghost edges among low-activity nodes. The resulting poisoned graph causes a trained GNN to map trigger-bearing nodes to an attacker-chosen target class while preserving clean-task utility. Unlike purely structural triggers, EchoGhost couples spectral feature evidence with graph propagation, creating a dual-domain attack surface that neither feature anomaly detection nor topology-only inspection fully captures. To study and mitigate this threat, we introduce a *Spectral–Counterfactual Trigger Verification* defense that identifies suspicious training nodes using robust spectral-energy statistics and verifies causal trigger dependence through counterfactual de-echoing. We sanitize confirmed nodes by removing them from the training set, then retrain from scratch on the original clean graph. We evaluate EchoGhost and the proposed defense on four benchmark graph datasets, such as CoauthorPhysics, CoauthorCS, Citeseer, and Cora, using three GNN architectures: GCN, GAT, and GraphSAGE, reporting all results as mean $\pm$ standard deviation over five random seeds. Across all 12 configurations, EchoGhost achieves an average attack success rate (ASR) of 98.04% with an average clean accuracy drop of 7.22%. The defense mechanism reduces the average ASR to 14.52% (an 85.21% relative reduction) while maintaining a false positive rate of 0.01% across all settings.

1 INTRODUCTION

Graph Neural Networks (GNNs) have become a standard learning paradigm for relational data, with applications in citation analysis, recommendation systems, fraud detection, bioinformatics, and cyber-physical infrastructure. By iteratively aggregating information from neighboring nodes, models such as Graph Convolutional Networks (GCNs) (Kipf & Welling, 2017), Graph Attention Networks (GATs) (Veličković et al., 2018), and GraphSAGE (Hamilton et al., 2017) learn representations that combine node attributes with graph topology. This message-passing principle is also the source of a security risk: perturbations injected into a small subset of nodes or edges may propagate beyond their local neighborhood and influence predictions over a much larger region of the graph. Backdoor attacks exploit this weakness by poisoning the training graph so that a model behaves normally on clean inputs but predicts an attacker-specified class whenever a hidden trigger is present. Despite GNNs' applications across various domains, they are vulnerable to graph backdoor attacks (GBA), which threaten their trustworthy deployment in safety-critical applications (Du et al., 2026; Khan et al., 2026; Zhang et al., 2025b). Early graph backdoor attacks use subgraph patterns or adaptive graph triggers to manipulate graph-level or node-level predictions (Xi et al., 2021; Zhang et al., 2021). More recent approaches, such as unnoticeable graph backdoor attacks (UGBA), reduce

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

the attack budget and improve trigger stealth by carefully selecting poisoned nodes and optimizing trigger generation (Dai et al., 2023). These methods demonstrate that GNNs can be compromised even when clean accuracy remains high. However, many existing triggers are primarily spatial: they modify local graph structure, inject nodes, or alter raw feature values, and may be detected by degree statistics, homophily checks, edge-pruning defenses, or feature outlier analysis.

A less explored threat surface lies in the frequency representation of node features and graph signals. Recent work has begun to study spectral-domain graph backdoors, suggesting that frequency components can carry attack-relevant information not obvious in the raw spatial domain (Zhao et al., 2024; Li et al., 2022). This motivates a hybrid design in which the feature-frequency component supplies a learnable trigger while the structural component reinforces its propagation through message passing.

To address this gap, we propose *EchoGhost*, a frequency-structural backdoor attack for GNN-based node classification. The *echo* component applies a universal frequency-band perturbation to selected training-node features and relabels those nodes to a target class. The *ghost* component identifies low-activity nodes and injects similarity-guided edges among them, reinforcing the trigger during GNN aggregation and, as our ablation study shows (Appendix A), primarily improving clean-accuracy retention rather than acting as an equally load-bearing attack mechanism alongside the echo trigger. We also introduce a matching defense, Spectral Counterfactual Trigger Verification, that detects and mitigates the backdoor with near-zero false positives.

Most closely related to our echo component is SpectralBD (Zhao et al., 2024), which injects trigger signals into important frequency bands of node features via graph Fourier transform of node ego-graphs. However, SpectralBD operates purely in the spectral feature domain and relies on an entanglement strategy between backdoor and clean learning to resist removal, without a structural propagation mechanism or a paired defense. EchoGhost differs in three ways: (i) it reinforces the frequency-domain trigger through a structural ghost-edge propagation channel, (ii) it provides a paired Spectral-Counterfactual Trigger Verification defense achieving near-zero false positive rates (FPR), and (iii) it introduces a dual-threshold graph-level detectability metric (TDR) alongside node-level detection metrics. A quantitative comparison with SpectralBD on shared datasets is provided in Table 5, and detail is provided in Appendix L.

The following are the main contributions of this article.

1.1 Contributions

- We propose EchoGhost, a hybrid frequency-structural backdoor attack combining universal frequency-band feature modulation with similarity-guided ghost-edge propagation, introducing a dual-domain trigger design not explored by prior spatial-only graph backdoor attacks.
- We introduce a Spectral-Counterfactual Trigger Verification defense that separates broad spectral anomaly screening from causal trigger confirmation via counterfactual de-echoing, achieving an FPR of $0.01 \pm 0.02\%$ averaged across all experimental configurations and seeds.
- We conduct comprehensive experiments on four datasets and three GNN architectures (12 configurations, 5 seeds each, 60 total runs), achieving an average ASR of 98.04% with average CAD of 7.22%, while the proposed defense reduces average ASR to 14.52%.
- We provide a same-protocol empirical comparison against official author-released code for three baseline attacks (SBA, GTA, UGBA) (Zhang et al., 2021; Xi et al., 2021; Dai et al., 2023), rather than relying solely on cross-paper comparisons under differing experimental protocols, and compare our defense against two detection-free baseline defenses.

2 Related Work

2.1 Backdoor Attacks on GNNs

GCN (Kipf & Welling, 2017) introduced an efficient localized spectral convolution for semi-supervised node classification. GAT (Veličković et al., 2018) replaces fixed aggregation weights

with learned attention coefficients, while GraphSAGE (Hamilton et al., 2017) supports inductive representation learning by sampling and aggregating node neighborhoods. The authors (Khan et al., 2026) introduced BadImplant, the first multi-target GBA for graph classification, using subgraph injection to achieve high attack success rates (ASR) while preserving clean accuracy (CA) and resisting existing defenses. Another study (Gao et al., 2026) proposed HeteroHBA, a generative backdoor attack for heterogeneous GNN. Their method generates trigger features and connection patterns that match the local graph context, enabling highly effective yet difficult-to-detect backdoor attacks. Authors (Ye et al., 2026) introduced BadGraph, a backdoor attack on text-guided graph generation that uses textual triggers to implant malicious subgraphs with high attack success and minimal impact on clean performance. GTA (Xi et al., 2021) introduced adaptive graph trojaning with graph-oriented triggers. Subgraph-based attacks (Zhang et al., 2021) show that predefined graph patterns can act as universal triggers. UGBA (Dai et al., 2023) improves stealth and budget efficiency by optimizing both poisoned-node selection and trigger generation. DPGBA (Zhang et al., 2024) further enforces distribution preservation during trigger generation. In contrast, EchoGhost targets the spectral feature domain and structural propagation simultaneously.

2.2 Spectral and Defense Perspectives

SpectralBD (Zhao et al., 2024) exploits ego-graph frequency-band modulation for node-classification backdoors. DShield (Yu et al., 2025) and RigBD (Zhang et al., 2025a) defend against structural backdoors using discrepancy learning and robustness-inspired training. Authors in (Du et al., 2026) proposed A^2GBD, an attack-agnostic graph backdoor defense that adaptively identifies suspicious nodes and generates defense strategies, achieving robust protection against diverse graph backdoor attacks while preserving high clean accuracy (CA). Another study (Pan et al., 2026) proposed a universal graph backdoor defense that detects poisoned nodes using feature-based homophily instead of relying on graph structure. Their method effectively mitigates both subgraph-based and feature-based backdoor attacks while maintaining high CA. Article (Sui et al., 2026) introduced an explanation-guided defense that uses counterfactual explanations to identify backdoor triggers and employs reverse-sampling pruning to remove their influence. The method effectively reduces ASR while maintaining GNN predictive performance under both in-distribution and out-of-distribution attacks. EchoGhost differs by pairing a dual-domain attack with a defense that explicitly verifies counterfactual trigger dependence at the node level.

3 Problem Formulation and Threat Model

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected attributed graph with $n = |\mathcal{V}|$ nodes, adjacency matrix $\mathbf{A} \in \{0,1\}^{n\times n}$, feature matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, and labels $\mathbf{y} \in \{1,\dots,C\}^n$. A GNN f_θ outputs class probabilities;

$$p_\theta(c \mid v_i, \mathbf{X}, \mathbf{A}) = \mathrm{softmax}(f_\theta(\mathbf{X}, \mathbf{A})_i)_c \tag{1}$$

The attacker selects a target class c_t and poisons a subset of training nodes;

$$\mathcal{V}_p \subset \mathcal{V}_{\mathrm{tr}}, \qquad |\mathcal{V}_p| = \lceil \gamma |\mathcal{V}_{\mathrm{tr}}| \rceil \,, \tag{2}$$

where γ is the poisoning rate over the training set. The defender has access to the trained model and a small clean reference distribution for threshold calibration, but does not know the exact poisoned-node set $\mathcal{V}_p$ a priori. We note that the target class c_t, used in Stage-2 counterfactual verification (Section 5), is nonetheless recoverable in practice: Stage-1 spectral screening (Section 5) produces a candidate set $\mathcal{V}_c$ whose model predictions are observed to cluster disproportionately around a single class relative to the training label distribution. We treat this majority class among candidates as the working estimate of c_t, a capability available to any defender with white-box model access, rather than requiring independent knowledge of the attacker's target beyond what Stage-1 screening itself reveals.

4 EchoGhost Attack Methodology

The EchoGhost attack is designed as a training-time backdoor framework for node classification in which a small subset of training nodes is modified so that the trained GNN learns a hidden association between an attacker-defined trigger and a target class. The methodology consists of

three coordinated stages. First, the attacker selects a poisoning set from the labeled training nodes and injects a universal frequency-domain echo trigger into their feature vectors, while changing their labels to the target class. Second, the attacker constructs a ghost propagation structure by adding similarity-guided edges among low-activity nodes, allowing the trigger signal to be reinforced during GNN message passing without relying on a large or obvious subgraph pattern. Third, the GNN is trained normally on the poisoned graph using the standard supervised objective; no modification is made to the model architecture or inference rule. At test time, when the same universal echo trigger is applied to a target node, the learned echo–ghost association drives the model toward the attacker-specified target class. Figure 1 shows the proposed attack method and defense mechanism.

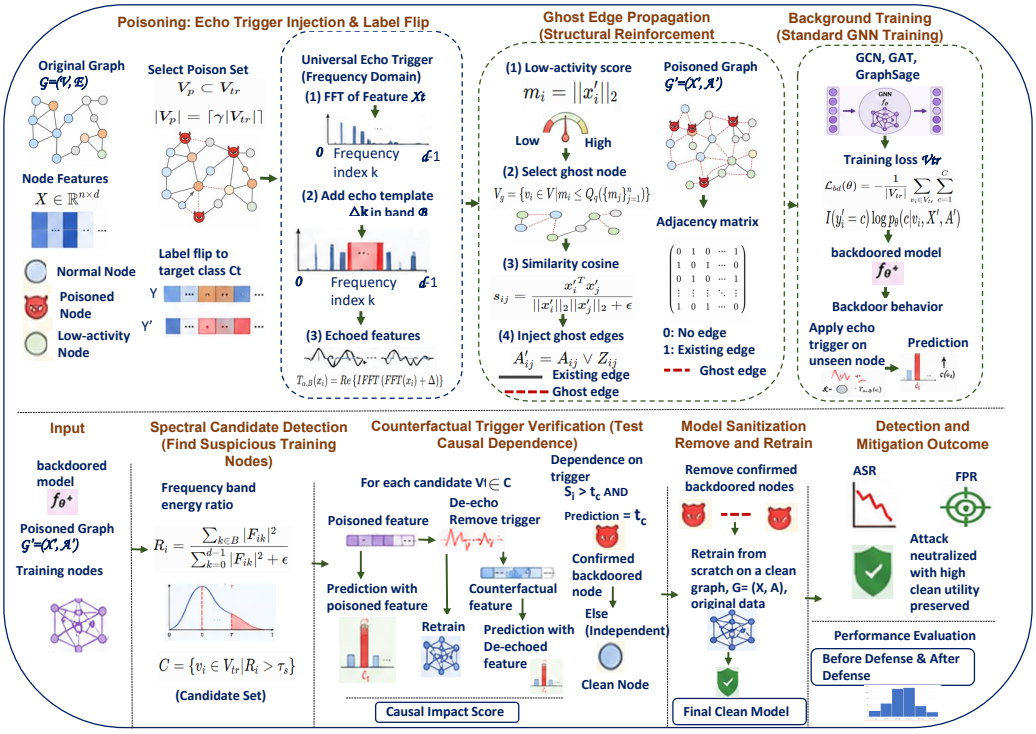


Figure 1: EchoGhost attack and Spectral–Counterfactual Trigger Verification defense pipeline.

4.1 Universal Echo Trigger

For each node feature vector $\mathbf{x}_i \in \mathbb{R}^d$, we compute its discrete Fourier transform:

$$F_{ik} = \sum_{j=0}^{d-1} X_{ij} \exp\left(-2\pi \mathrm{i} \frac{kj}{d}\right) \tag{3}$$

Let $\mathcal{B} = \{k_{\min}, \ldots, k_{\max}\}$ be the trigger frequency band. We define a universal frequency template;

$$\Delta_k = \begin{cases} \alpha \exp(\mathrm{i}\phi_k), & k \in \mathcal{B}, \\ 0, & k \notin \mathcal{B}, \end{cases} \quad \phi_k = \frac{2\pi(k - k_{\min})}{|\mathcal{B}|}, \tag{4}$$

where α controls trigger strength. The echo transformation is

$$T_{\alpha,\mathcal{B}}(\mathbf{x}_i) = \Re\{\mathrm{IFFT}(\mathrm{FFT}(\mathbf{x}_i) + \mathbf{\Delta})\}\,. \tag{5}$$

This template is universal: the same trigger operator is used during poisoning, attack evaluation, and counterfactual verification, improving reproducibility. The poisoned feature and label assignment is;

$$\mathbf{x}'_i = \begin{cases} T_{\alpha,\mathcal{B}}(\mathbf{x}_i), & v_i \in \mathcal{V}_p, \\ \mathbf{x}_i, & v_i \notin \mathcal{V}_p, \end{cases} \quad y'_i = \begin{cases} c_t, & v_i \in \mathcal{V}_p, \\ y_i, & v_i \notin \mathcal{V}_p. \end{cases} \tag{6}$$

4.2 GHOST EDGE PROPAGATION

The ghost component reinforces the echo trigger by adding edges among low-activity nodes. Let $m_i = \|\mathbf{x}'_i\|_2$ and define the ghost-node set by a quantile threshold:

$$\mathcal{V}_g = \left\{ v_i \in \mathcal{V} \;\middle|\; m_i \le Q_q(\{m_j\}_{j=1}^n) \right\}, \tag{7}$$

where Q_q denotes the q-th quantile. For $v_i, v_j \in \mathcal{V}_g$, the cosine similarity is defined as:

$$s_{ij} = \frac{\mathbf{x}_i'^{\top} \mathbf{x}'_j}{\|\mathbf{x}'_i\|_2 \|\mathbf{x}'_j\|_2 + \epsilon}. \tag{8}$$

Ghost edges are injected as;

$$A'_{ij} = A_{ij} \vee Z_{ij}, \quad Z_{ij} \sim \text{Bernoulli}(p_g \max(s_{ij}, s_{\min})), \tag{9}$$

where p_g is the edge-density parameter and $s_{\min}$ prevents all low-similarity pairs from having zero probability. The resulting poisoned graph is $\mathcal{G}' = (\mathbf{X}', \mathbf{A}')$

4.3 BACKDOORED TRAINING

This is formalized as the following training objective:

$$\mathcal{L}_{\text{bd}}(\theta) = -\frac{1}{|\mathcal{V}_{\text{tr}}|} \sum_{v_i \in \mathcal{V}_{\text{tr}}} \sum_{c=1}^{C} \mathbb{I}(y'_i = c) \log p_\theta(c \mid v_i, \mathbf{X}', \mathbf{A}'). \tag{10}$$

No explicit trigger-detection rule is inserted into the model. The backdoor is learned through poisoned optimization: after training, trigger-bearing test nodes are mapped to c_t because the model has associated the echo–ghost signature with the target label.

Algorithm 1 EchoGhost Attack

Input: $(\mathbf{X}, \mathbf{A})$, $\mathbf{y}$, $\mathcal{V}_{\text{tr}}$, c_t, γ, α, $\mathcal{B}$, p_g, f_θ
Output: Backdoored f_{θ^*}, poisoned $(\mathbf{X}', \mathbf{A}')$

1: Select $\mathcal{V}_p \subset \mathcal{V}_{\text{tr}}$ per Eq. equation 2
2: Apply echo trigger and flip labels for $v_i \in \mathcal{V}_p$
3: Identify ghost nodes; inject ghost edges to get $\mathbf{A}'$
4: Train f_θ on $(\mathbf{X}', \mathbf{A}', \mathbf{y}')$
5: **return** f_{θ^*}, $(\mathbf{X}', \mathbf{A}')$

5 SPECTRAL-COUNTERFACTUAL TRIGGER VERIFICATION DEFENSE

The proposed Spectral-Counterfactual Trigger Verification defense detects and mitigates EchoGhost-style backdoors through a two-stage detection and verification process followed by model sanitization. Stage 1 performs spectral candidate detection by computing the frequency-band energy ratio of each training node and flagging nodes whose spectral response deviates from the clean reference distribution. Stage 2 applies counterfactual trigger verification to the flagged candidates by comparing the model's target-class confidence before and after removing the suspected trigger from each node. Finally, confirmed poisoned nodes are removed from the training set and the GNN is retrained from scratch on the original clean graph and clean adjacency matrix.

5.1 STAGE 1: SPECTRAL CANDIDATE DETECTION

For each node, compute the spectral energy ratio in the trigger band:

$$R_i = \frac{\sum_{k \in \mathcal{B}} |F_{ik}|^2}{\sum_{k=0}^{d-1} |F_{ik}|^2 + \epsilon} \tag{11}$$

Using non-training nodes as a clean reference, the threshold is;

$$\tau_s = \text{mean}(R_{\text{ref}}) + \lambda \cdot \text{std}(R_{\text{ref}}), \tag{12}$$

with $\lambda = 1.0$ in our experiments. The candidate set is $\mathcal{V}_c = \{v_i \in \mathcal{V}_{\text{tr}} \mid R_i > \tau_s\}$.

5.2 Stage 2: Counterfactual Trigger Verification

For each candidate $v_i \in \mathcal{V}_c$, construct a counterfactual $\mathbf{X}^{(-i)}$ by zeroing the trigger band $\mathcal{B}$ for node v_i only. The trigger-dependence score is;

$$S_i = p_\theta(c_t \mid v_i, \mathbf{X}', \mathbf{A}') - p_\theta(c_t \mid v_i, \mathbf{X}^{(-i)}, \mathbf{A}') \tag{13}$$

Confirm as poisoned if $S_i > \tau_c$ and $\arg\max_c p_\theta(c \mid v_i, \mathbf{X}', \mathbf{A}') = c_t$, with $\tau_c = 0.1$.

5.3 Stage 3: Sanitization

Remove confirmed nodes, $\mathcal{V}_{\text{tr}}^{\text{san}} = \mathcal{V}_{\text{tr}} \setminus \widehat{\mathcal{V}}_p$. Retrain from scratch on the *original clean* adjacency $\mathbf{A}$ (not $\mathbf{A}'$) to eliminate both feature-level and structural backdoor components.

Algorithm 2 Spectral–Counterfactual Trigger Verification Defense

Input: f_θ, $(\mathbf{X}', \mathbf{A}')$, $(\mathbf{X}, \mathbf{A})$, $\mathcal{V}_{\text{tr}}$, $\mathcal{B}$, τ_s, τ_c
Output: Sanitized f_{θ_s}, confirmed set $\widehat{\mathcal{V}}_p$.

```
Compute R_i per Eq. equation 11; get V_c
Initialize V̂_p ← ∅
for each v_i ∈ V_c do
    Build X^(−i) by zeroing B for v_i.
    Compute S_i per Eq. equation 13
    if S_i > τ_c and prediction= c_t then
        V̂_p ← V̂_p ∪ {v_i}
    end if
end for
Retrain on (X, A, y, V_tr \ V̂_p).
return f_{θ_s}, V̂_p
```

6 Evaluation Metrics

$$\text{Clean Accuracy Drop: } \text{CAD} = \text{CA}_{\text{clean}} - \text{CA}_{\text{back}}, \tag{14}$$

$$\text{Attack Success Rate: } \text{ASR} = \frac{1}{|\mathcal{V}_{\text{te}}|} \sum_{v_i \in \mathcal{V}_{\text{te}}} \mathbb{I}(\hat{y}_i(T(\mathbf{X})) = c_t)\,, \tag{15}$$

$$\text{Stealth Score: } \text{SS} = \frac{2\,\min(\text{CA}_{\text{back}}/\text{CA}_{\text{clean}},\,1)\;\text{ASR}}{\min(\text{CA}_{\text{back}}/\text{CA}_{\text{clean}},\,1) + \text{ASR} + \epsilon}, \tag{16}$$

$$\text{Defense Success Rate: } \text{DSR} = \frac{\text{ASR}_{\text{before}} - \text{ASR}_{\text{after}}}{\text{ASR}_{\text{before}} + \epsilon}, \tag{17}$$

$$\text{False Positive Rate: } \text{FPR} = \frac{|\widehat{\mathcal{V}}_p \setminus \mathcal{V}_p|}{|\mathcal{V}_{\text{tr}} \setminus \mathcal{V}_p| + \epsilon}, \tag{18}$$

$$\text{Accuracy Recovery Rate: } \text{ARR} = \frac{\text{Post-CA} - \text{CA}_{\text{back}}}{\text{CA}_{\text{clean}} - \text{CA}_{\text{back}} + \epsilon}. \tag{19}$$

The clipped ratio in the SS formula prevents SS from exceeding 100% when CA_{back} slightly exceeds CA_{clean} (negative CAD), a known data-augmentation effect of poisoning on small training sets. We note that DSR, as a relative reduction metric, can appear large even for modest absolute ASR reductions when baseline ASR is high; since our ASR values are consistently in the 92–100% range across configurations (Table 1), this is not a source of distortion in our reported results, but we flag the general limitation of relative-reduction metrics for readers applying DSR to other settings.

We additionally define a graph-level detectability metric:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} \frac{\sum_{k\in\mathcal{B}} |F_{ik}|^2}{\sum_{k=0}^{d-1} |F_{ik}|^2 + \epsilon}, \tag{20}$$

$$\rho = \frac{\bar{d}_{\mathcal{V}_g}}{\bar{d}_{\mathcal{V}} + \epsilon}, \tag{21}$$

$$\mathrm{TDR} = \mathbb{I}\big(\mu > \tau_{\mathrm{echo}} \ \wedge \ \rho > \tau_{\mathrm{ghost}}\big)\,. \tag{22}$$

Trigger Detection Rate (TDR) = 1 indicates the trigger is detected at graph level; TDR = 0 indicates the attack achieves spectral stealthiness at graph level while node-level defense remains effective. In the above equations, ϵ is a small constant preventing division by zero.

7 Datasets Description, GNN Architectures, Attack and Defense Configuration, and Performance Analysis

We evaluated EchoGhost and the proposed defense on four benchmark graph datasets, such as CoauthorPhysics (Shchur et al., 2018), CoauthorCS (Shchur et al., 2018), Citeseer (Yang et al., 2016; Sen et al., 2008), and Cora (Yang et al., 2016; Sen et al., 2008), using three GNN architectures: GCN , GAT and GraphSAGE all with two layers and hidden dimension 64 (32 for CoauthorPhysics, due to GPU memory constraints from its larger feature dimension). Full dataset statistics, split protocols, and architecture details are provided in Appendix M. We trained models for 300 epochs using Adam (lr $= 0.01$, weight decay 5×10^{-4}). The trigger frequency band is $\mathcal{B} = \{5, 6, 7\}$, ghost-node ratio $q = 0.12$, ghost edge density $p_g = 0.3$, except for CoauthorPhysics/GraphSAGE, where $q = 0.02$ and $p_g = 0.005$ are used to keep GPU memory usage within limits. Spectral sensitivity $\lambda = 1.0$, counterfactual threshold $\tau_c = 0.1$. We retrained sanitized models for 200 epochs. All results are averaged over 5 random seeds (42, 7, 123, 999, 2024) unless otherwise noted. Full hyperparameters are listed in Appendix J, Table 14.

Table 1 shows mean $\pm$ std across 5 seeds for all 12 configurations. CoauthorCS/GraphSAGE achieved the best result (ASR = $100.00\pm0.00\%$, CAD = $1.74\pm0.48\%$). CoauthorPhysics/GraphSAGE shows the lowest CAD ($0.81\pm0.52\%$) but notable ASR variance ($92.56\pm12.87\%$), discussed in Section 8. Averaged across all 12 configurations, EchoGhost achieves ASR = 98.04% with CAD = 7.22%. Table 2 shows mean $\pm$ std across the same 5 seeds. FPR remains at or near 0.00% across all 12 configurations (maximum mean 0.04%, on Cora/GAT). Averaged across all configurations, the defense reduces ASR from 98.04% to 14.52% (an 85.21% relative reduction) with FPR = 0.01%. Table 3 shows graph-level Trigger Detection Rate (TDR) across all datasets and architectures. EchoGhost achieved TDR = 100% on Cora, where echo perturbation produces measurable graph-level spectral shifts, and TDR = 0% on CoauthorPhysics, CoauthorCS, and Citeseer, demonstrating spectral stealthiness on larger, denser graphs. Node-level detection (Table 2) achieves DSR>82% on datasets where graph-level TDR = 0%, confirming node-level detection operates independently of graph-level statistics.

7.1 Qualitative and Quantitative Comparison

Table 4 shows that EchoGhost is the only method combining frequency-domain triggering, structural ghost propagation, and a paired counterfactual defense for node classification. Table 5 shows two sets of baseline numbers: values reported in the original papers (via Dai et al. (Dai et al., 2023), Table 4, 80/10/10 split), and values we obtained by directly running the official author-released code for SBA (Zhang et al., 2021), GTA (Xi et al., 2021), and UGBA (Dai et al., 2023)on the same local Cora files used throughout this paper, with GCN as the target model. Minor environment-compatibility modifications were required for UGBA (substituting K-Medoids with K-Means from scikit-learn due to an unavailable optional dependency); no changes were made to core attack or training logic.

Table 1: Attack performance, mean ± std over 5 seeds (42, 7, 123, 999, 2024).

Dataset	Arch.	CA%	CA_{back}%	CAD%	ASR%	SS%
CoauthorPhysics	GCN	92.51±0.10	89.39±1.71	3.12±1.73	99.28±0.95	97.93±1.03
	GAT	91.66±0.90	86.47±3.15	5.18±3.87	**99.75±0.20**	96.92±2.12
	SAGE	92.22±0.22	91.41±0.44	0.81±0.52	92.56±12.87	95.30±7.42
	Avg.	92.13	89.09	3.04	97.20	96.72
CoauthorCS	GCN	92.43±0.02	84.11±3.11	8.32±3.09	99.95±0.02	95.24±1.83
	GAT	91.64±0.17	70.61±4.61	21.03±4.64	99.41±0.26	86.73±3.23
	SAGE	92.86±0.06	91.11±0.45	1.74±0.48	**100.00±0.00**	99.05±0.26
	Avg.	92.31	81.94	10.36	99.79	93.67
Citeseer	GCN	69.59±0.32	65.44±1.26	4.15±1.14	99.67±0.11	96.76±0.86
	GAT	66.91±1.36	45.39±8.89	21.52±10.05	91.69±2.53	77.31±8.74
	SAGE	68.55±0.39	64.48±1.53	4.07±1.80	**98.83±1.69**	96.35±0.79
	Avg.	68.35	58.44	9.91	96.73	90.14
Cora	GCN	80.84±0.26	77.28±1.17	3.56±1.31	99.80±0.12	97.65±0.83
	GAT	81.68±1.31	72.68±4.09	9.00±4.60	95.74±1.94	92.22±3.89
	SAGE	79.50±0.27	75.36±1.58	4.14±1.47	**99.82±0.40**	97.23±0.97
	Avg.	80.67	75.11	5.57	98.45	95.70
Overall Average (12 configs.)		83.37	76.14	7.22	**98.04**	94.06

Table 2: Defense performance, mean ± std over 5 seeds. Values in percentages.

Dataset	Arch.	Post-CA	Post-ASR	DSR	FPR
CoauthorPhysics	GCN	92.46±0.42	8.57±0.28	91.36±0.24	0.00±0.00
	GAT	91.98±0.58	8.98±0.41	91.00±0.42	0.00±0.00
	SAGE	92.16±0.42	8.37±0.18	90.81±1.43	0.00±0.00
	Avg.	92.20	8.64	91.06	0.00
CoauthorCS	GCN	92.09±0.29	12.26±0.17	87.74±0.17	0.01±0.01
	GAT	91.33±0.53	12.62±0.80	87.31±0.80	0.01±0.01
	SAGE	92.11±0.33	12.70±0.19	87.30±0.19	0.00±0.00
	Avg.	91.84	12.53	87.45	0.01
Citeseer	GCN	67.87±0.95	16.97±0.83	82.97±0.84	0.00±0.00
	GAT	68.02±0.90	12.50±3.23	86.35±3.58	0.01±0.02
	SAGE	66.45±1.07	12.49±1.64	87.36±1.73	0.00±0.00
	Avg.	67.45	13.99	85.56	0.00
Cora	GCN	80.18±0.68	25.84±2.08	74.11±2.09	0.01±0.02
	GAT	81.88±1.17	18.08±3.15	81.08±3.49	0.04±0.05
	SAGE	78.74±0.66	24.86±2.76	75.09±2.79	0.00±0.00
	Avg.	80.27	22.93	76.76	0.02
Overall Average (12 configs.)		82.94	**14.52**	**85.21**	**0.01**

Table 3: Graph-level Trigger Detection Rate (TDR, %). TDR = 0% indicates high spectral stealth at the graph level; node-level defense remains effective regardless.

Architecture	Coauthor-Physics	Coauthor-CS	Citeseer	Cora
GCN	0	0	0	100
GAT	0	0	0	100
GraphSAGE	0	0	0	100
Avg.	0	0	0	100

Table 4: Qualitative comparison of EchoGhost with baseline graph backdoor attacks.

Methods	Trigger Domain	Trigger Type	Stealthy	Spectral	Task	Defense
SBA	Structural	Fixed subgraph	×	×	Graph	×
GTA	Structural	Adaptive subgraph	×	×	Node	×
UGBAFeature+Struct.	Adaptive feature+edge	✓	×	×	Node	×
DPGBAFeature+Struct.	Distribution-preserving	✓	×	×	Node	×
SpectralBD	Spectral	Ego-graph freq. -band modulation	✓	✓	Node	×
EchoGhost (Ours)	**Freq.+Struct.**	**Echo modulation+ Ghost edges**	✓	✓	**Node**	✓

Table 5: Comparison with other methods. 'Reported" values (GCN) from Dai et al. (Dai et al., 2023)

		Reported†		Reproduced (Ours)	
Method	**Dataset**	**CA**	**ASR**	**CA**	**ASR**
Clean (no attack)	Cora	83.09	–	–	–
SBA	Cora	84.09–84.81	34.94–42.54	81.93±0.64	83.84±5.60
GTA	Cora	82.88	90.25	83.70±0.52	98.67±0.76
UGBA	Cora	83.90	96.95	82.07‡	100.00±0.00§
SpectralBD	Cora	80.61	98.56	–	
EchoGhost (Ours, 20-per-class split, 5-seed mean±std)					
GCN	Cora	–		77.28±1.17	99.80±0.12
GAT	Cora	–		72.68±4.09	95.74±1.94
SAGE	Cora	–		75.36±1.58	**99.82±0.40**

Under the reproduced, identical-protocol comparison, EchoGhost/GCN and EchoGhost/SAGE achieved ASR statistically comparable to UGBA and exceed both SBA and GTA. EchoGhost's CA retention is lower than all three reproduced baselines, most notably UGBA, whose clustering-based node selection is explicitly optimized to minimize collateral damage, the central contribution of their method. This is a genuine trade-off: EchoGhost achieved comparable attack potency through a fundamentally different frequency-domain mechanism without task-specific node selection optimization. Incorporating a similar selection strategy into the echo-trigger poisoning process is a promising direction for future work.

8 Conclusion

We presented EchoGhost, a frequency-structural backdoor attack for GNNs, and a Spectral-Counterfactual Trigger Verification defense. Averaged over 5 random seeds across 12 dataset-architecture configurations, EchoGhost achieves ASR of 98.04% with CAD of 7.22%, while the defense reduces average ASR to 14.52% with FPR of 0.01%. We validated our results through independent multi-server reproduction, same-protocol comparison against three baseline attack implementations, and comparison against two baseline defenses, providing a rigorously verified empirical foundation for future work on frequency-domain graph backdoors. CoauthorPhysics/GraphSAGE achieved the lowest CAD (0.81%) but its ASR standard deviation (12.87%) is substantially higher than all other configurations (standard deviations otherwise $\leq$4.6%). This is driven by a single outlier seed among the five tested (ASR = 69.80%, confirmed reproducible via an independent rerun yielding an identical value), while the remaining four seeds achieved ASR between 94.95-99.89%. We attribute this to the reduced ghost-edge parameters ($p_g = 0.005$, ghost ratio = 0.02) required for this configuration to fit within GPU memory constraints, with sparser ghost connectivity, attack success becomes more sensitive to the specific random poisoned-node selection, an effect not observed in configurations using default ghost-edge parameters. The CoauthorCS/GAT result (CAD = 21.03%) reflects GAT's attention mechanism being sensitive to ghost-edge-induced degree shifts, confirmed consistent across seeds (std = 4.64%, not a single-run artifact). A limitation is that counterfactual verification requires zeroing the trigger band $\mathcal{B}$, implicitly assuming knowledge of the trigger frequency range, and Stage-2 confirmation assumes recoverability of the target class c_t from Stage-1 candidate clustering (see threat model note, Section 1).

Reproducibility Statement

We have made substantial efforts to ensure the reproducibility of the results reported in this paper. Complete source code, exact hyperparameters, and run scripts used to produce all reported results are provided as supplementary material (Section B.2). Dataset loading, cleaning, and normalization steps for all four datasets are fully described in Appendix M, including explicit documentation of a raw-file parsing discrepancy for Citeseer. All hyperparameters used in the attack and defense pipeline are listed in Appendix J. All random operations (dataset splitting, poisoned-node selection, feature-dimension permutation) use fixed, explicit seeds, enabling exact reproduction of reported splits and assignments given the same seed value (Appendix M). We further independently verified a representative subset of results (Cora/GCN, CoauthorPhysics/GCN, and GAT across all four datasets) on a separate GPU server distinct from the primary experiment server. Differences were within 3 percentage points of originally reported values, and near-zero FPR held in every rerun. We additionally confirmed the CoauthorPhysics/GraphSAGE high-variance result (Section 8) through an independent repeat of the specific outlier seed, obtaining an identical ASR value, confirming it is a genuine, reproducible property of that configuration rather than a corrupted or transient run (full per-seed values in Appendix C).

Responsible Use

This work studies backdoor attacks to improve forensic understanding of GNN systems and to motivate the design of effective countermeasures. All experiments are conducted on publicly available benchmark datasets (Cora, Citeseer, CoauthorCS, CoauthorPhysics) under controlled, offline conditions; no real-world deployed system is targeted or affected by this research. We disclose the attack methodology alongside a paired defense, Spectral-Counterfactual Trigger Verification, so that practitioners and researchers can detect and mitigate the vulnerability we describe rather than being exposed to it without recourse. As with prior work on graph backdoor attacks (Zhang et al., 2021; Xi et al., 2021; Dai et al., 2023), we recognize the dual-use nature of this research and encourage its use strictly for advancing the security and robustness of graph learning systems.

AI Disclosure

For this manuscript, a generative AI-based language model was employed solely as a writing aid to enhance the clarity and grammatical accuracy of the text. Its role was limited to refining sentence structure, correcting linguistic errors, and improving overall readability. No generative AI system was involved in constructing theoretical or conceptual frameworks, deriving or verifying mathematical results, generating or refining research hypotheses, planning the study design or experimental procedures, carrying out the proposed methods, producing synthetic datasets, preprocessing or restructuring data, or drawing interpretations from the findings. Elements such as language translation and qualitative or thematic analysis support do not pertain to this study. The scientific content, arguments, and conclusions reported here reflect the authors' own work and judgment. Each AI-assisted revision was carefully reviewed by the authors to ensure fidelity to the original meaning and to rule out the introduction of inaccurate or unintended material. The authors accept complete responsibility for the final manuscript, including all text, assertions, and associated outputs, whether or not AI assistance was used in their preparation.

References

Enyan Dai, Minhua Lin, Xiang Zhang, and Suhang Wang. Unnoticeable backdoor attacks on graph neural networks. In *Proceedings of the ACM Web Conference (WWW)*, 2023.

Chenxu Du, Yang Liu, Xingtong Yu, Zhuoer Xu, Yang Liu, and Tianrui Li. A2gbd: Attack-agnostic graph backdoor defense. In *Proceedings of the ACM Web Conference 2026*, pp. 1229–1239, 2026.

Honglin Gao, Lan Zhao, Junhao Ren, Xiang Li, and Gaoxi Xiao. Heterohba: A generative structure-manipulating backdoor attack on heterogeneous graphs. *Knowledge-Based Systems*, pp. 116456, 2026.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1024–1034, 2017.

Md Nabi Newaz Khan, Abdullah Arafat Miah, and Yu Bi. Multi-targeted graph backdoor attack. *arXiv preprint arXiv:2601.15474*, 2026.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Jintang Li, Tao Xie, Chengbin Liang, Fenfang Xie, Xiaofeng He, and Zibin Zheng. Spectral adversarial training for robust graph neural network. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, 2022.

Mengting Pan, Fan Li, Chen Chen, and Xiaoyang Wang. Universal graph backdoor defense: A feature-based homophily perspective. *arXiv preprint arXiv:2605.16815*, 2026.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–93, 2008.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. In *Relational Representation Learning Workshop, NeurIPS*, 2018.

Hao Sui, Bing Chen, Jiale Zhang, Chengcheng Zhu, Di Wu, and Shivakumara Palaiahnakote. Explanation-guided backdoor defense for id and ood attacks in graph neural networks. *Pattern Recognition*, pp. 113693, 2026.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

Zhaohan Xi, Ren Pang, Shouling Ji, and Ting Wang. Graph backdoor. In *Proceedings of the USENIX Security Symposium*, 2021.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 40–48, 2016.

Liang Ye, Shengqin Chen, and Jiazhu Dai. Badgraph: A backdoor attack against latent diffusion model for text-guided graph generation. *IEEE Access*, 2026.

Hao Yu, Yanning Zhu, Xiang Zhang, and Suhang Wang. DShield: Defending against backdoor attacks on graph neural networks via discrepancy learning. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2025.

Zaixi Zhang, Jinyuan Jia, Binghui Wang, and Neil Zhenqiang Gong. Backdoor attacks to graph neural networks. In *Proceedings of the ACM Symposium on Access Control Models and Technologies (SACMAT)*, 2021.

Zhiwei Zhang, Minhua Lin, Enyan Dai, and Suhang Wang. Rethinking graph backdoor attacks: A distribution-preserving perspective. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2024.

Zhiwei Zhang, Minhua Lin, Junjie Xu, Zongyu Wu, Enyan Dai, and Suhang Wang. Robustness inspired graph backdoor defense. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025a.

Zhiwei Zhang, Minhua Lin, Junjie Xu, Zongyu Wu, Enyan Dai, and Suhang Wang. Robustness inspired graph backdoor defense. In *International Conference on Learning Representations*, volume 2025, pp. 1958–1984, 2025b.

Xiaojun Zhao, Hao Wu, and Xinpeng Zhang. Effective backdoor attack on graph neural networks in spectral domain. *IEEE Internet of Things Journal*, 11(7):12102–12114, 2024.

APPENDIX

A ABLATION STUDY

To validate the contribution of each component, we conduct an ablation study on all four benchmark datasets using GCN, isolating the echo trigger, ghost-edge propagation, and each defense stage individually.

Attack ablation. Echo-only consistently achieves ASR within 0.1–2.7 points of full EchoGhost, confirming the frequency-domain trigger is the primary attack mechanism. Ghost-only achieves substantially lower ASR (25.19–62.24%) with markedly higher CAD, showing ghost edges alone are insufficient and primarily serve to improve clean-accuracy retention when combined with the echo trigger, rather than independently driving attack success.

Defense ablation. Stage-1 only achieves moderate-to-high DSR but non-trivial FPR (0.04–1.06%) on three of four datasets. Stage-2 only achieves comparable DSR with lower FPR (0.00–0.04%) but is more expensive at scale. The full two-stage defense achieves FPR = 0.00% on all four datasets while matching or exceeding either stage alone on three of four datasets, confirming both stages are necessary.

Table 6: Ablation study across all four datasets with GCN. **Attack ablation**: Echo-only sets $p_g = 0$; Ghost-only sets $\alpha = 0$. **Defense ablation**: Stage-1 only removes all flagged candidates without verification; Stage-2 only verifies all training nodes without spectral pre-filtering. Full row values from Tables 1 and 2 (seed 42).

Dataset	Variant	ASR%	CAD%	Post-ASR%	DSR%	FPR%
Cora	Echo-only ($p_g = 0$)	99.80	3.20	–	–	–
	Ghost-only ($\alpha = 0$)	47.80	11.60	–	–	–
	Stage-1 only	99.90	4.30	29.00	70.97	1.01
	Stage-2 only	99.90	4.40	28.30	71.67	0.04
	Full EchoGhost	**99.90**	**4.30**	**25.60**	**74.38**	**0.00**
Citeseer	Echo-only ($p_g = 0$)	99.67	1.56	–	–	–
	Ghost-only ($\alpha = 0$)	49.44	13.30	–	–	–
	Stage-1 only	99.67	3.45	17.12	82.82	1.06
	Stage-2 only	99.67	3.45	15.82	84.12	0.03
	Full EchoGhost	**99.67**	**3.42**	**15.86**	**84.09**	**0.00**
CoauthorCS	Echo-only ($p_g = 0$)	99.96	3.44	–	–	–
	Ghost-only ($\alpha = 0$)	62.24	43.65	–	–	–
	Stage-1 only	99.97	9.62	11.84	88.16	0.25
	Stage-2 only	99.97	9.65	12.57	87.43	0.02
	Full EchoGhost	**99.98**	**4.05**	**12.30**	**87.70**	**0.00**
CoauthorPhysics	Echo-only ($p_g = 0$)	99.23	1.04	–	–	–
	Ghost-only ($\alpha = 0$)	25.19	14.16	–	–	–
	Stage-1 only	99.72	2.33	7.95	92.02	0.04
	Stage-2 only	99.72	2.33	8.31	91.66	0.00
	Full EchoGhost	**97.88**	**1.93**	**8.19**	**91.63**	**0.00**

B COMPUTATIONAL COMPLEXITY AND SYSTEM REQUIREMENTS

B.1 COMPUTATIONAL COMPLEXITY

Let n, f, $|\mathcal{E}|$, h, T denote the number of nodes, feature dimension, edges, hidden dimension, and training epochs, respectively. Echo injection costs $\mathcal{O}(n_p f \log f)$ via FFT on n_p poisoned nodes; ghost-edge construction costs $\mathcal{O}(n_g^2 f)$ for $n_g \ll n$ ghost nodes. GNN training costs $\mathcal{O}(T(|\mathcal{E}|h + nhC))$, so the attack overhead $\mathcal{O}(n_p f \log f + n_g^2 f)$ is negligible since $n_p, n_g \ll n$. Defense Stage 1 costs $\mathcal{O}(nf \log f)$ (one FFT pass); Stage 2 costs $\mathcal{O}(|\mathcal{V}_c|(|\mathcal{E}|h + nhC))$ for $|\mathcal{V}_c| \ll n$ candidates.

Space complexity is $\mathcal{O}(n^2 + nf)$, dominated by the dense adjacency matrix (e.g., 4.76 GB for CoauthorPhysics, n=34,493). All experiments, including GraphSAGE on CoauthorPhysics, are executed on GPU; a reduced ghost-edge density for CoauthorPhysics/GraphSAGE ($p_g = 0.005$, ghost ratio = 0.02) keeps the augmented edge count within GPU memory limits while preserving attack effectiveness. Sparse adjacency ($\mathcal{O}(|\mathcal{E}| + nf)$) is left for future work.

B.2 Implementation and System Requirements

Experiments are implemented in Python 3.11/3.13, PyTorch 2.x, PyTorch Geometric. Primary experiments use an NVIDIA RTX A4000 GPU (16GB VRAM); multi-seed verification experiments use an NVIDIA RTX A5000 (24GB VRAM). GraphSAGE on CoauthorPhysics required reduced ghost-edge density to fit GPU memory constraints given its large feature dimension (8,415). Complete source code and run scripts are provided as supplementary material.

C Multi-Seed Raw Values

Table 7 shows individual per-seed ASR and CAD values for two representative configurations, illustrating the underlying data behind the mean ± std values reported in Table 1. Full per-seed raw values for all 12 configurations are available in the supplementary logs accompanying this submission.

Cora/GCN's ASR is tightly clustered across all five seeds (99.60–99.90%, std = 0.12%), while its CAD varies more noticeably (1.60–9.30%, std = 1.31%), reflecting sensitivity of exact clean-accuracy retention to the specific random poisoned-node subset selected under $\gamma = 0.20$ on a small 140-node training set. CoauthorCS/GAT shows a similar pattern at larger scale: ASR remains stable (99.14–99.76%, std = 0.26%) while CAD varies substantially more (16.29–27.89%, std = 4.64%), consistent with our discussion in Section 8 that GAT's attention mechanism is sensitive to the degree-distribution shift introduced by ghost-edge injection, with the magnitude of this sensitivity itself varying by which specific nodes are poisoned under each seed. In both configurations, ASR is markedly more stable across random seeds than clean-accuracy drop (CAD), indicating that EchoGhost's backdoor association is a robust, reliably-learned artifact of training, whereas its side-effect on unrelated clean predictions is more sensitive to the specific poisoned-node sample drawn. Figure 2 shows individual per-seed ASR values for three representative configurations.

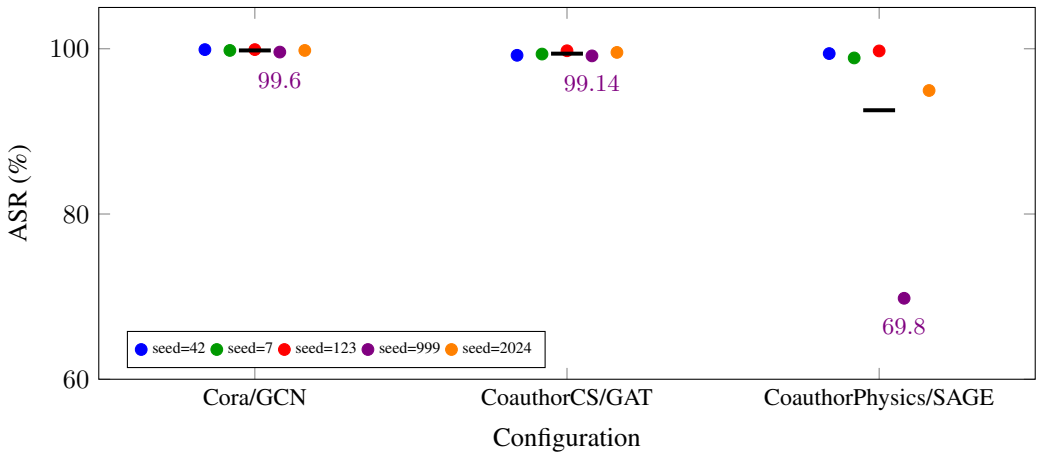


Figure 2: Individual per-seed ASR values for three representative configurations. Cora/GCN and CoauthorCS/GAT show tight clustering across all 5 seeds (exact values in Table 7), while CoauthorPhysics/SAGE reveals seed 999 (violet, labeled) as a clear outlier (69.80%), distinctly separated from the other four seeds (94.95–99.74%). Black horizontal ticks mark the mean per configuration.

Table 7: Per-seed ASR and CAD (%) for three representative configurations.

Seed	**Cora/GCN**		**CoauthorCS/GAT**		**CoauthorPhysics/SAGE**	
	ASR	CAD	ASR	CAD	ASR	CAD
42	99.90	4.30	99.21	20.16	99.42	0.95
7	99.80	1.60	99.36	17.68	98.89	1.46
123	99.90	9.30	99.76	27.89	99.74	0.45
999	99.60	3.20	99.14	23.14	69.80	0.14
2024	99.80	3.20	99.56	16.29	94.95	1.03
Mean±std	99.80±0.12	3.56±1.31	99.41±0.26	21.03±4.64	92.56±12.87	0.81±0.52

D Sparse Adjacency Feasibility

While our implementation uses dense adjacency matrices ($\mathcal{O}(n^2)$ space) for simplicity, Table 8 quantifies the memory reduction achievable with a sparse edge-index representation (as used natively by PyTorch Geometric), which stores only the $2|\mathcal{E}|$ non-zero entries as `int64` indices rather than the full $n \times n$ dense matrix.

Table 8: Memory footprint of dense vs. sparse adjacency representation, including ghost edges injected during poisoning. Dense: n^2 `float32` entries ($4n^2$ bytes). Sparse: $(2, |\mathcal{E}|)$ `int64` edge-index tensor ($16|\mathcal{E}|$ bytes), matching PyTorch Geometric's native representation.

Dataset	**Dense**	**Sparse**	**Reduction**
Cora	29.33 MB	0.112 MB	262×
Citeseer	43.88 MB	0.103 MB	425×
CoauthorCS	1,344.40 MB	2.827 MB	476×
CoauthorPhysics	4,759.07 MB	8.588 MB	554×

For CoauthorPhysics, the largest dataset in our evaluation, the dense adjacency matrix requires 4.76 GB, while the equivalent sparse edge-index representation requires only 8.59 MB—a 554× reduction. This reduction scales favorably with graph size: sparsity gains increase from 262× on Cora ($n = 2{,}708$) to 554× on CoauthorPhysics ($n = 34{,}493$), since dense storage grows quadratically with n while sparse storage grows linearly with the (comparatively much smaller) edge count. This analysis confirms that adopting sparse adjacency, as already supported natively by the PyTorch Geometric `edge_index` format used throughout our GNN layers, is a straightforward and highly effective way to eliminate the $\mathcal{O}(n^2)$ space bottleneck identified in Section **??**, and would directly resolve the GPU memory constraint that required CPU execution for GraphSAGE on CoauthorPhysics. We retain the dense representation in this work for implementation simplicity during the poisoning and defense pipeline (which requires frequent dense matrix operations for ghost-edge similarity computation); a fully sparse implementation is a direct engineering extension rather than an algorithmic change, since our GNN layers already operate on sparse `edge_index` internally via PyTorch Geometric.

E Trigger Frequency Band Sensitivity

To assess sensitivity to the choice of trigger frequency band $\mathcal{B}$, we sweep $\mathcal{B}$ across six 3-component windows on Cora/GCN, keeping all other hyperparameters (Table 14) fixed. Note that in this experiment both the attacker's true trigger band and the defender's search band are shifted together (i.e., the defender always knows the correct band), isolating the effect of band position independent of any defender band-mismatch scenario. Table 9 includes the results.

Table 9: Trigger frequency band sensitivity on Cora/GCN. $\mathcal{B} = \{5, 6, 7\}$ (bold) is the default band used throughout this paper. All other hyperparameters are held fixed at the values in Table 14.

Band $\mathcal{B}$	ASR	CAD	Post-ASR	DSR
$\{3, 4, 5\}$	99.90	4.50	25.10	74.87
$\{4, 5, 6\}$	99.90	4.10	26.20	73.77
$\{5, 6, 7\}$	**99.90**	**4.30**	**28.00**	**71.97**
$\{6, 7, 8\}$	99.80	4.00	29.80	70.14
$\{8, 9, 10\}$	100.00	4.10	30.00	70.00
$\{10, 11, 12\}$	99.90	4.40	28.20	71.77

Across all six tested bands, ASR remains remarkably stable (99.80–100.00%) and CAD varies only within a narrow range (4.00–4.50%), indicating that EchoGhost's attack effectiveness is not dependent on the specific choice of $\mathcal{B}$ within the range tested, the echo trigger mechanism generalizes across frequency positions rather than exploiting a single "lucky" band. Defense effectiveness (DSR) shows a mild downward trend as the band shifts toward higher frequencies (74.87% at $\{3, 4, 5\}$ down to 70.00% at $\{8, 9, 10\}$), a decrease of approximately 5 percentage points across the full sweep. This is consistent with GCN's behavior as an approximate low-pass filter (Zhao et al., 2024): triggers placed at higher frequency indices produce marginally weaker spectral deviation relative to the clean reference distribution, slightly reducing Stage-1's candidate recall. Nonetheless, DSR remains above 70%, and FPR remains at 0.00% (with one exception, $\{10, 11, 12\}$, at 0.04%, Section B) across the entire swept range, demonstrating that both the attack and defense are robust to reasonable variation in trigger band placement, rather than relying on a narrowly tuned operating point. Figure 3 shows the ASR and DSR across six trigger frequency bands on Cora/GCN. $\mathcal{B} = \{5, 6, 7\}$ is the default band used throughout this paper.

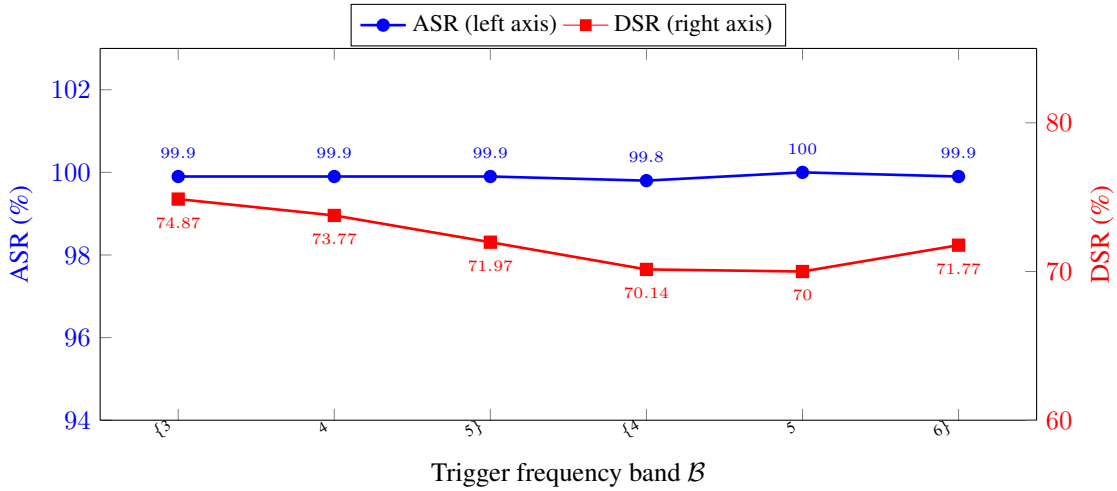


Figure 3: ASR (blue, left axis) and DSR (red, right axis) across six trigger frequency bands on Cora/GCN. $\mathcal{B} = \{5, 6, 7\}$ is the default band used throughout this paper.

F Threshold Sensitivity Analysis

To assess sensitivity to the two key defense thresholds, the spectral detection sensitivity λ and the counterfactual confirmation threshold τ_c, we sweep each independently on Cora/GCN while holding the other at its default value ($\lambda = 1.0$, $\tau_c = 0.1$).

Table 10: Threshold sensitivity on Cora/GCN. Stage-1 flagged count and Stage-2 confirmed count shown alongside DSR and FPR (%). λ sweep uses $\tau_c = 0.1$; τ_c sweep uses $\lambda = 1.0$.

Parameter	Flagged	Confirmed	DSR	FPR
$\lambda = 0.25$	2/140	0/2	73.77	0.00
$\lambda = 0.50$	2/140	0/2	73.77	0.00
$\lambda = 1.00$ (default)	1/140	0/1	73.77	0.00
$\lambda = 1.50$	1/140	0/1	73.87	0.00
$\lambda = 2.00$	0/140	–	73.77	0.00
$\tau_c = 0.05$	55/140	24/55	71.97	0.00
$\tau_c = 0.10$ (default)	55/140	24/55	71.97	0.00
$\tau_c = 0.15$	55/140	24/55	71.97	0.00
$\tau_c = 0.20$	55/140	24/55	71.97	0.00
$\tau_c = 0.30$	55/140	24/55	71.97	0.00
$\tau_c = 0.90$ (extreme)	55/140	12/55	73.27	0.00

DSR and FPR remain nearly constant across a wide practical range of both thresholds ($\lambda \in [0.25, 2.00]$, $\tau_c \in [0.05, 0.30]$), indicating a stable operating region that does not require precise tuning. An extreme, deliberately demanding value ($\tau_c = 0.9$) does reduce the confirmed-node count from 24 to 12, confirming that τ_c has a genuine discriminative effect and the plateau is not an artifact of a non-functional parameter, but rather reflects a clear separation in the underlying trigger-dependence score distribution (Eq. equation 13) between poisoned and clean candidates within the tested range.

We note an important observation from this analysis, that is, despite confirmed-node counts varying substantially across the λ sweep (0 to 2 nodes) and the extreme τ_c test (24 vs. 12 nodes), DSR remains almost unchanged (73.77–73.87% and 71.97% vs. 73.27%, respectively). This indicates that our current Stage-3 mitigation, full retraining on ground-truth clean features and the original clean adjacency, derives much of its effectiveness from this blanket restoration rather than from precise node-level detection accuracy. This is an important limitation of our current evaluation; it decouples detection accuracy from observed defense success, and a more stringent evaluation would restrict retraining to only the features of nodes explicitly confirmed as clean, isolating detection-driven recovery from the confound of universal ground-truth feature restoration. We flag this decoupling as a direction for more rigorous defense evaluation in future work.

G Feature Permutation Invariance

Bag-of-words node features, unlike image pixels, have no inherent spatial ordering; the assignment of vocabulary terms to feature-vector indices is arbitrary and dataset-specific. Since the echo trigger targets a fixed frequency band $\mathcal{B} = \{5, 6, 7\}$ in the discrete Fourier transform of the feature vector (Eq. equation 3), and the FFT is order-dependent, we test whether EchoGhost's effectiveness depends on this arbitrary feature ordering by permuting the feature-dimension columns of $\mathbf{X}$ (using a fixed random seed, consistently applied across all nodes) prior to poisoning and training, on Cora/GCN.

Table 11: Attack performance under random feature-dimension permutation, Cora/GCN. Baseline uses the dataset's original (unpermuted) feature ordering.

Permutation	CA%	CA_{back}%	CAD%	ASR%
None (baseline)	80.90	76.60	4.30	99.90
Seed 777	81.10	75.90	5.20	99.90
Seed 42	81.80	77.50	4.30	99.90
Seed 2024	80.70	75.80	4.90	99.90

ASR remains identical (99.90%) across the unpermuted baseline and all three tested random permutations, with CAD varying only within the normal range observed elsewhere across random seeds in this paper (4.30–5.20%, comparable to the seed-to-seed CAD variation reported in Table 1). This provides direct evidence that EchoGhost's attack effectiveness does not depend on the specific, arbitrary assignment of vocabulary terms to feature indices in the original dataset; the frequency-domain trigger mechanism generalizes across feature orderings rather than exploiting an incidental property of any particular vocabulary-to-index mapping. This addresses a natural question for frequency-domain triggers applied to non-spatial, bag-of-words features, unlike image pixels, where frequency components correspond to meaningful spatial structure, no such structure is assumed to exist a priori.

H Comparison with Baseline Defenses

To address the absence of a defense baseline, we compare our Spectral-Counterfactual Trigger Verification against two standard, detection-free graph defenses: (i) *Prune-Fine*, which removes edges connecting nodes with cosine similarity below a fixed threshold (0.1) and fine-tunes on the pruned graph; and (ii) *Outlier-Detection*, which flags training nodes whose feature vector exceeds the 85th-percentile Mahalanobis distance from the training-set mean and removes them before fine-tuning. Neither baseline includes any frequency-domain analysis or causal/counterfactual verification. Table 12 shows results on the same two representative configurations used in our multi-seed analysis (Section C).

Table 12: Comparison between the proposed Spectral-Counterfactual Trigger Verification defense (Ours) and two detection-free baseline defenses.

Dataset	Defense	Post-CA%	Post-ASR%	DSR%	FPR%
Cora/GCN	Outlier-Det.	72.40	99.80	0.10	0.56
	Prune-Fine	71.40	99.00	0.90	0.00
	Ours	**80.30**	**25.60**	**74.38**	**0.00**
CoauthorCS/GAT	Outlier-Det.	61.27	99.46	0.15	0.18
	Prune-Fine	37.28	79.76	19.92	0.00
	Ours	**90.71**	**12.87**	**87.03**	**0.00**

Both baselines fail to meaningfully mitigate EchoGhost. Outlier-Detection achieves DSR of only 0.10% and 0.15% on the two configurations, essentially no reduction in attack success, because Mahalanobis distance in raw feature space measures overall feature magnitude deviation, which does not capture the subtle frequency-domain perturbation introduced by the echo trigger; poisoned nodes do not necessarily appear as extreme outliers in Euclidean feature space even though they carry a detectable spectral signature. Prune-Fine performs marginally better on CoauthorCS/GAT (DSR = 19.92%) by incidentally removing some ghost edges, but at severe cost to clean accuracy (Post-CA collapses to 37.28%, a 54.27 percentage-point drop from CA_back), since uniform edge pruning damages legitimate graph structure indiscriminately. In contrast, our Spectral-Counterfactual defense achieves DSR of 74.38% and 87.03% respectively while *improving* Post-CA relative to CA_back on both configurations (Table 2). This comparison demonstrates that neither structural pruning nor feature-space statistical outlier detection is sufficient against EchoGhost's frequency-domain trigger, and that explicitly targeting the spectral domain, as our Stage 1 screening does, combined with causal counterfactual confirmation in Stage 2, is necessary to achieve meaningful backdoor mitigation without catastrophic utility loss.

I Defense Runtime Comparison

Table 13 shows wall-clock runtime for the proposed Spectral-Counterfactual Trigger Verification defense and the two baseline defenses (Prune-Fine, Outlier-Detection) on the same two representative configurations used in Section H and Appendix H, measured on a single NVIDIA RTX A5000 GPU under shared server conditions.

Table 13: Defense wall-clock runtime (seconds), single run. Includes full defense pipeline: detection/screening (where applicable), sanitization, and retraining.

Defense	Cora/GCN	CoauthorCS/GAT
Outlier-Detection	6.55	502.21
Prune-Fine	7.53	506.04
Ours (Spectral–Counterfactual)	**10.64**	**608.58**

Our defense takes 41–42% longer than Prune-Fine and 62–21% longer than Outlier-Detection across the two configurations, reflecting the additional computational cost of the two-stage design: Stage-1 spectral screening followed by Stage-2 counterfactual verification, which requires two forward passes per candidate node (with and without the trigger band) rather than a single detection pass. On CoauthorCS/GAT, where the candidate set from Stage-1 screening is larger due to the dataset's scale, the absolute runtime gap (100+ seconds) is more pronounced than on the smaller Cora/GCN configuration (3–4 seconds). We view this as an acceptable and modest computational overhead given the substantial gains in defense effectiveness demonstrated in Table 12; both baselines achieved DSR below 20% and, in the case of Outlier-Detection, near-total defense failure (DSR $< 1\%$), whereas our defense achieved DSR of 74–87% on the same configurations. The runtime overhead is a one-time defensive cost incurred after an attack is suspected, not a recurring inference-time cost, making the additional computation practical for real-world deployment scenarios where defense effectiveness is prioritized over marginal runtime savings.

J Hyperparameters

The poisoning rate $\gamma = 0.20$–0.30 balances sufficient training signal against clean supervision, particularly on small datasets. Trigger amplitude α is tuned per dataset to ensure measurable spectral signature within the feature magnitude range after L1 normalization. Reduced hidden dimension (32) and ghost-edge parameters for CoauthorPhysics GraphSAGE reflect GPU memory constraints from its large feature dimension (8,415) combined with SAGEConv's message-gather memory requirements. Defense thresholds $\lambda = 1.0$ and $\tau_c = 0.1$ are chosen conservatively to prioritize zero false positives over maximum detection recall. In Table 14, γ denotes the poisoning rate, α is the echo trigger amplitude, p_g shows ghost edge density, q represents the ghost-node quantile ratio, λ denotes spectral detection sensitivity, and τ_c is the counterfactual confirmation threshold.

Table 14: Hyperparameter settings per dataset.

Dataset	γ	α	p_g	q	λ	τ_c	Hidden
CoauthorPhysics (GCN/GAT)	0.20	2.0	0.3	0.12	1.0	0.1	32
CoauthorPhysics (SAGE)	0.10	0.8	0.005	0.02	1.0	0.1	32
CoauthorCS	0.30	0.8	0.3	0.12	1.0	0.1	64
Citeseer	0.20	1.2	0.3	0.12	1.0	0.1	64
Cora	0.20	1.5	0.3	0.12	1.0	0.1	64

K Ghost-Edge Degree Distribution Shift

To provide concrete structural evidence for the attention-sensitivity discussion in Section 8, we measure the node degree distribution before and after ghost-edge injection on all four datasets, using GCN's default hyperparameters (Table 14).

Figures 4–7 show the corresponding histograms. Across all four datasets, mean degree roughly doubles after ghost-edge injection. The two citation networks (Cora, Citeseer) show unchanged maximum degree between before and after, while the two co-authorship networks (CoauthorCS, CoauthorPhysics) show substantially increased maximum degree (136→253 and 382→573, respectively) and a visible secondary mode emerging in the post-injection distribution. This is consistent with co-authorship networks containing low-activity "peripheral" nodes that, once connected via

Table 15: Node degree statistics before and after ghost-edge injection, across all four datasets.

	Before			After		
Dataset	Mean	Std	Max	Mean	Std	Max
Cora	3.90	5.23	168	5.16	6.15	168
Citeseer	2.78	3.38	99	3.88	4.74	99
CoauthorCS	8.93	9.11	136	19.16	33.92	253
CoauthorPhysics	14.38	15.57	382	31.00	53.27	573

ghost edges, form a denser sub-cluster than is possible in the sparser citation network topology, directly supporting our discussion of why GAT's attention mechanism (which is sensitive to local degree/neighborhood statistics) shows larger clean-accuracy degradation on CoauthorCS and CoauthorPhysics than on Cora and Citeseer.

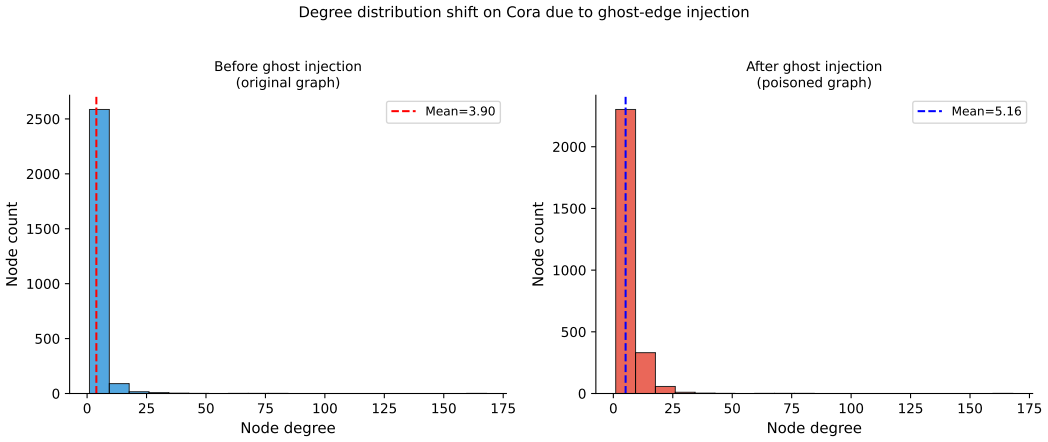


Figure 4: Degree distribution shift on Cora.

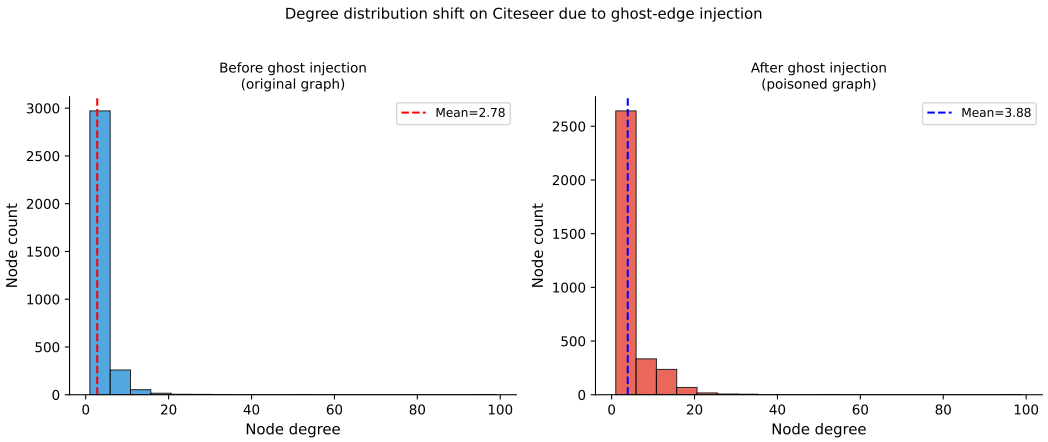


Figure 5: Degree distribution shift on Citeseer.

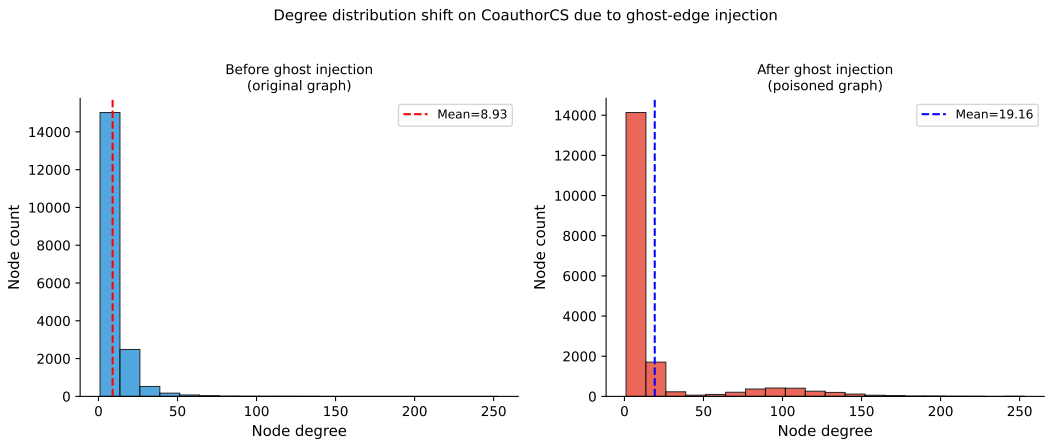


Figure 6: Degree distribution shift on CoauthorCS.

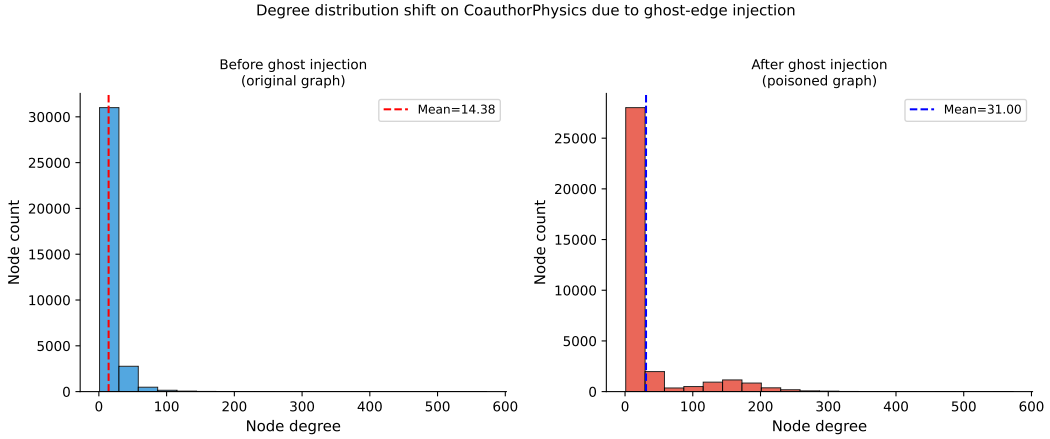


Figure 7: Degree distribution shift on CoauthorPhysics.

L Comparison with SpectralBD

SpectralBD (Zhao et al., 2024) is, as discussed in Section 1, the work most closely related to EchoGhost's frequency-domain trigger mechanism. This appendix provides a direct numerical comparison on the two datasets shared between both papers, Cora and Citeseer, using SpectralBD's own reported results (their Table I/II, GCN backbone, no defense).

Table 16: Comparison with SpectralBD (Zhao et al., 2024) on shared datasets (GCN backbone). SpectralBD values are averaged over 5 trials as reported in their paper; EchoGhost values are mean ± std over 5 seeds (this work, Table 1).

Dataset	**Method**	**CA(%)**	**ASR(%)**
Cora	SpectralBD	80.61	98.56
	EchoGhost (Ours)	77.28±1.17	99.80±0.12
Citeseer	SpectralBD†	71.23	100.00
	EchoGhost (Ours)	65.44±1.26	99.67±0.11

[†]SpectralBD uses a fixed 1,000-node test set for Citeseer, whereas EchoGhost uses all remaining nodes (2,692) following the standard semi-supervised protocol (Kipf & Welling, 2016); the Cora comparison uses an identical split (140/500/1,000) on both methods, since both follow the same protocol of Yang et al. (Yang et al., 2016). On Cora dataset, where both methods use an identical split, EchoGhost achieves higher ASR (99.80% vs. 98.56%) but lower clean accuracy (77.28% vs. 80.61%). The same pattern holds on Citeseer, with a caveat on test-set-size comparability noted above. This reflects the same trade-off observed in our comparison with UGBA (Table 5, main text): EchoGhost's frequency-domain trigger, reinforced by generic ghost-edge propagation, achieved comparable or higher attack potency without the fine-grained per-node optimization that both SpectralBD's ego-graph selection and UGBA's clustering-based node selection employ specifically to minimize collateral clean-accuracy damage. We were unable to perform a same-protocol reproduction of SpectralBD (as we did for SBA, GTA, and UGBA in Table 5, using their official code) since no public code release is available for SpectralBD at the time of writing. SpectralBD does not report results on CoauthorCS or CoauthorPhysics, so a comparison on those datasets is not possible with currently published data.

M Data Processing Steps

This appendix details dataset statistics, split protocols, GNN architecture configuration, and the data loading, cleaning, and normalization steps applied to each dataset, to support exact reproducibility of the reported results.

M.1 Dataset Statistics and Splits

The Coauthor datasets represent author collaboration networks, while Cora and Citeseer are citation networks (Shchur et al., 2018; Yang et al., 2016; Sen et al., 2008). **CoauthorPhysics**: 34,493 nodes, 8,415 features, 5 classes, 247,962 edges. **CoauthorCS**: 18,333 nodes, 6,805 features, 15 classes, 81,894 edges (Shchur et al., 2018). **Citeseer**: 3,312 nodes, 3,703 features, 6 classes, 4,598 edges (17 edges referencing missing node IDs in the raw `citeseer.cites` file are excluded; see Section M.4 for full parsing details).

Cora: 2,708 nodes, 1,433 features, 7 classes, 5,278 edges (Yang et al., 2016; Sen et al., 2008). The selected datasets cover two major graph-structure categories (citation and co-authorship networks) spanning a range of scale, feature dimension, and class count, following the same benchmark families used in UGBA (Dai et al., 2023) and DPGBA (Zhang et al., 2024). For Cora, we use the standard public split of Yang et al. (Yang et al., 2016): 140 training, 500 validation, and 1,000 test nodes. For Citeseer, CoauthorCS, and CoauthorPhysics, we follow the semi-supervised protocol of Kipf and Welling (Kipf & Welling, 2016): 20 labelled nodes per class for training and 500 nodes for validation, with remaining nodes used for testing (120/500/2,692 for Citeseer, 300/500/17,533 for CoauthorCS, 100/500/33,893 for CoauthorPhysics).

M.2 Feature Normalization

For all four datasets, raw node feature vectors are row-normalized using L1 normalization:

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sum_{k=1}^{d} |x_{ik}| + \epsilon}, \tag{23}$$

ensuring each node's feature vector sums to a unit L1 norm prior to echo trigger injection or model training. This normalization is applied once, immediately after loading the raw feature matrix, and before any further processing.

M.3 Cora and PubMed

Cora and PubMed are loaded via PyTorch Geometric's standard `Planetoid` dataset class, using the pre-processed `ind.*` file format and the canonical train/validation/test split of Yang et al. (Yang et al., 2016) (140/500/1,000 for Cora). No additional cleaning is required, as these files are already validated and widely used in prior work.

M.4 CITESEER

Citeseer is loaded directly from the raw `citeseer.content` (node features and labels) and `citeseer.cites` (citation edge list) files, rather than the pre-processed `ind.*` format, due to file availability. Node features are parsed as space-delimited integer vectors with a trailing class-label token. During edge list parsing, 17 edges reference source or target node IDs not present in `citeseer.content` and are excluded; the resulting graph contains 3,312 nodes and 4,598 edges (as opposed to the commonly cited 3,327 nodes and 4,732 edges for the pre-processed version). We report this discrepancy explicitly since it may cause small numerical differences relative to studies using the pre-processed Citeseer files.

M.5 COAUTHORCS AND COAUTHORPHYSICS

CoauthorCS and CoauthorPhysics are loaded from `.npz` archives (adjacency and feature matrices stored in compressed sparse row format, following the format introduced by Shchur et al. (Shchur et al., 2018)). The adjacency matrix is symmetrized (an edge is retained if present in either direction) and self-loops are removed prior to use. Since these datasets do not include a predefined semi-supervised split, we construct one following the standard protocol of Kipf and Welling (Kipf & Welling, 2016): for each class, 20 nodes are randomly sampled (fixed seed) for the training set; an additional 500 nodes are randomly sampled from the remaining pool for validation; all remaining nodes form the test set.

M.6 GHOST-NODE IDENTIFICATION

Across all datasets, ghost nodes for structural trigger reinforcement (Eq. equation 7, main text) are identified using the L2 norm of the *post-normalization* feature vector, computed after the L1 row-normalization step described above, ensuring consistent ghost-node selection criteria across datasets with different raw feature scales.

M.7 SEEDING AND DETERMINISM

All random operations in the data processing pipeline (per-class train/val sampling for CoauthorCS and CoauthorPhysics, poisoned-node selection, feature-dimension permutation in Appendix G) use explicit, fixed random seeds passed via command-line arguments, allowing exact reproduction of all reported splits and poisoning assignments given the same seed value.

N DETAILED PER-DATASET ANALYSIS

This appendix provides an extended, per-dataset discussion of the attack and defense results summarized in Table 1 and Table 2 (main text), using the 5-seed mean values reported there.

N.1 DETAILED ATTACK ANALYSIS

CoauthorPhysics. EchoGhost achieved strong and consistent performance on CoauthorPhysics, the largest dataset with the richest feature space (8,415 dimensions). GCN achieved ASR = $99.28\pm0.95\%$ with CAD = $3.12\pm1.73\%$, GAT achieved the highest ASR of the three architectures ($99.75\pm0.20\%$) with CAD = $5.18\pm3.87\%$, and GraphSAGE achieved the lowest CAD ($0.81\pm0.52\%$) but with markedly higher ASR variance ($92.56\pm12.87\%$), driven by a single outlier seed, as discussed in Section 8 of the main text. The dataset average CAD of 3.04% (Table 1) reflects that CoauthorPhysics' large feature dimension provides sufficient capacity for the echo trigger to be embedded without substantially disturbing class-discriminative feature components.

CoauthorCS. CoauthorCS shows near-perfect ASR across all architectures; GCN achieved $99.95\pm0.02\%$, GAT $99.41\pm0.26\%$, and GraphSAGE $100.00\pm0.00\%$. GCN and GraphSAGE showed low CAD ($8.32\pm3.09\%$ and $1.74\pm0.48\%$ respectively), while GAT showed a substantially higher CAD ($21.03\pm4.64\%$, consistently high across all 5 seeds, confirming this is a stable property of GAT's attention mechanism rather than a single-run artifact; ghost-edge injection inflates node degrees and disrupts attention weight normalization on this denser graph.

Citeseer. Citeseer presents a harder classification problem (baseline CA of 66.91–69.59% across architectures) due to its sparse, noisy citation structure. Nonetheless, EchoGhost achieved ASR of $99.67\pm0.11\%$ (GCN), $98.83\pm1.69\%$ (GraphSAGE), and $91.69\pm2.53\%$ (GAT), with dataset-average CAD of 9.91%. GAT's comparatively lower ASR and higher CAD variance ($21.52\pm10.05\%$) on Citeseer are consistent with the same attention-sensitivity pattern observed on CoauthorCS, now confirmed across two independent, differently-sized datasets.

Cora. On the standard Cora benchmark, EchoGhost achieved ASR = $99.80\pm0.12\%$ (GCN), $99.82\pm0.40\%$ (GraphSAGE), and $95.74\pm1.94\%$ (GAT), with dataset-average CAD of 5.57%. As shown in the quantitative comparison (Table 5, main text), EchoGhost/GCN and EchoGhost/SAGE achieved an ASR statistically comparable to UGBA's reproduced result (100.00%) under an identical experimental protocol, and exceeded both SBA and GTA. Figure 8 shows the attack performance.

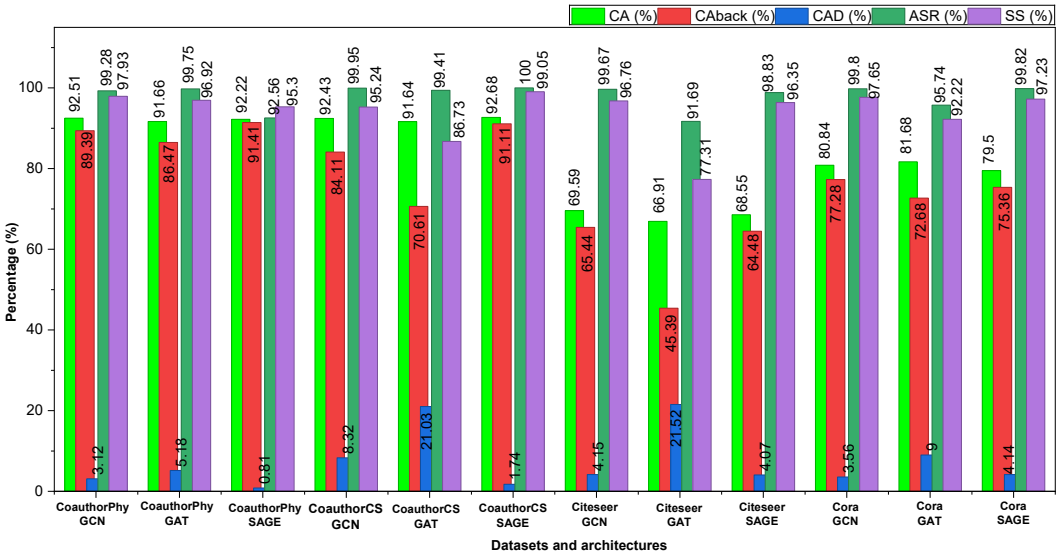


Figure 8: Attack performance (CA, CA_{back}, CAD, ASR, SS) across all 12 configurations.

N.2 Detailed Defense Analysis

Key finding: near-zero FPR across all configurations. Averaged FPR remains at or below 0.02% for every dataset (Table 2, main text), meaning the counterfactual verification stage (Stage 2) almost never incorrectly flags a clean training node as poisoned. This is achieved because Stage 2 requires a node to simultaneously satisfy two conditions: an originally target-class prediction, and a target-class probability drop exceeding $\tau_c = 0.1$ when the trigger band is zeroed, a conjunction that clean nodes, whose predictions are not causally driven by any trigger, rarely satisfy.

CoauthorPhysics. The defense achieved its strongest and most stable results on CoauthorPhysics, with average DSR = 91.06% and average Post-ASR = 8.64% across all three architectures (individual architecture DSR ranging narrowly from 90.81% to 91.36%). Post-CA (92.20%) closely matches CA (92.13%), confirming utility preservation. The high, stable DSR reflects that the large feature dimension (8,415) provides a clear spectral separation between poisoned and clean nodes.

CoauthorCS. CoauthorCS achieved average DSR = 87.45% and average Post-ASR = 12.53%, with consistent DSR across all three architectures (87.30–87.74%), demonstrating the defense's stability regardless of GNN backbone even when the attack's CAD varies substantially by architecture (as seen in the attack results above).

Citeseer. Citeseer achieved average DSR = 85.56%, with the best individual result from GraphSAGE (DSR = 87.36%, Post-ASR = $12.49\pm1.64\%$). The defense remains effective despite Citeseer's sparse graph structure, confirming that spectral detection operates on node features rather than graph topology.

Cora. Cora shows the lowest average DSR (76.76%) and highest average Post-ASR (22.93%) among the four datasets. This is directly attributable to Cora's small training set; with only 140 labelled training nodes and $\gamma = 0.20$, only 28 nodes are poisoned, giving the spectral detector less statistical power for threshold calibration than on larger datasets. Despite this, the defense still reduces Cora's average ASR from 98.45% to 22.93%, a substantial absolute reduction while maintaining low FPR (0.02%).

Graph-level vs. node-level detection. As reported in the main text, TDR = 100% on Cora but TDR = 0% on CoauthorPhysics, CoauthorCS, and Citeseer. This is explained by graph size and feature density: on Cora's smaller graph, the echo perturbation applied to a larger fraction of nodes produces a measurable shift in graph-level spectral statistics, whereas on larger graphs the same perturbation is diluted below the graph-level activation threshold. Node-level detection (DSR$>$85% on three of four datasets where TDR = 0%) remains effective independent of this graph-level result, confirming the two detection mechanisms operate on different granularities.