GraphProt: Certified Black-Box Shielding Against Backdoored Graph Models

Xiao Yang[1,†] , **Yuni Lai**[2,†] , **Kai Zhou**[2,*] , **Gaolei Li**[1*] , **Jianhua Li**[1] and **Hang Zhang**[3]
[1]Shanghai Jiao Tong University
[2]Hong Kong Polytechnic University
[3]Cornell University
{youngshall, gaolei_li, lijh888}sjtu.edu.cn, csylai@comp.polyu.edu.hk,
kaizhou@polyu.edu.hk, hz459@cornell.edu

Abstract

Graph learning models have been empirically proven to be vulnerable to backdoor threats, wherein adversaries submit trigger-embedded inputs to manipulate the model predictions. Current graph backdoor defenses manifest several limitations: 1) dependence on model-related details, 2) necessitation of additional fine-tuning, and 3) reliance on extra explainability tools, all of which are infeasible under stringent privacy policies. To address those limitations, we propose GraphProt, a certified black-box defense method to suppress backdoor attacks on GNN-based graph classifiers. Our GraphProt operates in a model-agnostic manner and solely leverages graph input. Specifically, GraphProt first introduces designed topology-feature-filtration to mitigate graph anomalies. Subsequently, subgraphs are sampled via a formulated strategy integrating topology and features, followed by a robust model inference through a majority vote-based subgraph prediction ensemble. Our results across benchmark attacks and datasets show GraphProt effectively reduces attack success rates while preserving regular graph classification accuracy.

1 Introduction

The abundance of graph data has led to the widespread adoption of graph learning models, such as Graph Neural Networks (GNNs), across diverse domains including social network analysis [Fan *et al.*, 2019], molecular biology [Wieder *et al.*, 2020], and recommendation systems [Safae *et al.*, 2023; Wu *et al.*, 2021]. With the increasing complexity of these models, there is a growing trend to outsource the training process to third parties, giving rise to a popular business model known as Machine Learning as a Service (MLaaS). While MLaaS can significantly simplify model training, it concurrently raises critical security concerns, particularly backdoor risks. Specifically, adversaries utilize poisoning training to implant backdoor and exploit trigger-embedded inputs to activate it for output manipulation.

To mitigate graph backdoors, several defense methods have been developed. Those methods leverage explainability to identify and remove triggers based on external tools, model-relevant details, and loss functions [Jiang and Li, 2022; Downer *et al.*, 2024], or employ additional benign samples or model parameters for fine-tuning to mitigate backdoor impact [Zhang *et al.*, 2024; Yang *et al.*, 2024]. However, to safeguard the privacy and intellectual property of the model owner and prevent extraction attacks, defenders are commonly prohibited from using the aforementioned model-related information or utilizing auxiliary data to fine-tune the model. This restriction makes it challenging to implement current methods in MLaaS scenarios.

To address the limitations, we deploy input subgraph details for prediction to mitigate backdoor effects induced by triggers. Within malicious graph input, trigger is the functional component, despite comprising a small fraction. Suppose the subgraph entirely lacks the trigger or contains only a trivial fragment, the backdoor will remain dormant and fail to activate. For one malicious input, most predictions of its subgraphs are typically normal. Hence, we determine output by majority vote on predictions of subgraphs within suspicious test graphs to avoid malicious results in testing. For benign samples, assuming that the subgraph incorporates sufficient feature-rich information, the accuracy of majority vote can be guaranteed.

Based on this insight, we propose GraphProt, a certified black-box defense method against backdoor attacks on GNN-based graph classifiers. *It functions within the testing phase to ensure robust output, irrespective of input maliciousness or benignity, demanding exclusively graph inputs.* Specifically, for the test input, we first implement designed feature clustering and topological clustering to filter out the potential anomalous parts maximally. Subsequently, multiple subgraphs are sampled from the filtered graph utilizing three proposed methods, founded on topology connectivity and node characteristics. Finally, we use one devised ensemble classifier to predict the subgraphs and perform majority vote on the results to derive the inference decision for the input. Additionally, we provide a certified robustness proof for GraphProt, showing that, under specific trigger size, no further attack attempts can compromise its certified efficacy—regardless of trigger type or target. The proposed method is depicted in Fig. 1. Our contributions are listed

[*]Corresponding author.

as follows:

- We propose GRAPHPROT, a provable black-box backdoor defense method for graph classifiers, which operates solely on the input test graph and several queries. The method functions independently of model-specific knowledge, supplementary resources, or external tools.
- In GRAPHPROT, we introduce a graph anomaly filtering method to maximally eliminate segments with significant anomalies in both features and topology. Moreover, we devise three subgraph sampling strategies based on topology and node characteristics.
- Evaluation results reveal that GRAPHPROT can effectively reduce attack success rates, attaining efficacy comparable to white-box defenses and exhibiting marginal reductions in benign data accuracies.

2 Related Work

2.1 Graph Neural Backdoor Attack

This attack aims to manipulate graph models to output adversary-prescribed targets upon receiving trigger-embedded graphs.

Backdooring graph model is implemented by data-poisoning [Xi *et al.*, 2021; Zhang *et al.*, 2021; Li *et al.*, 2024]. Adversaries incorporate premeditated triggers Δ into part of training graphs $\mathcal{D}_{tr}$ and modify their ground truths as targets y_Δ to compel the model $f(\cdot)$ to learn the mapping between Δ and y_Δ in training. The trained $f(\cdot)$ misclassifies trigger-embedded graphs as y_Δ, while correctly classifying benign data.

To adapt backdoors to multiple graph learning scenarios, the data-poisoning paradigm has been improved to suit the demands of federated learning, contrastive learning, prompt learning, and hardware-based graph systems [Xu *et al.*, 2022; Zhang *et al.*, 2023; Lyu *et al.*, 2024; Alrahis *et al.*, 2023]. Moreover, several studies augment backdoor efficiency, efficacy, and concealment through explainability, transferability, multi-targets, and spectrum [Xu *et al.*, 2021; Zheng *et al.*, 2024; Yang *et al.*, 2022; Wang *et al.*, 2024; Zhao *et al.*, 2024].

2.2 Graph Neural Backdoor Defense

Currently, only several defenses are researched for graph backdoors, which identify and eliminate triggers within test graphs to detect attacks and avoid backdoor activations.

Graph classification backdoor defense is initially explored via explainability tools and available poisonous data to set thresholds for recognizing and removing malicious triggers [Jiang and Li, 2022]. Moreover, clustering is introduced to identify triggers and utilize model structure details for fine-tuning to enhance robustness [Yang *et al.*, 2024]. Also, explainability metrics from logits and topology are harnessed to inspect sample poisoning [Downer *et al.*, 2024].

3 Background

3.1 Graph Classification

A graph is represented as $G = (V, E, X) \in \mathbb{G}$, where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of n nodes, E denotes the set of edges connecting the nodes in V, and $X \in \mathbb{R}^{n \times d}$ signifies d-dimensional feature for all nodes $v \in V$. With a training set $\mathcal{D}_{tr} = \{(G_i, y_i)\}_{i=1}^{n}$ containing training graphs G_i and their corresponding ground truth $y_i \in \mathcal{Y}$, a graph classifier $f(\cdot) : \mathbb{G} \rightarrow \mathcal{Y}$ can be trained. Given a testing graph G, the model can then be utilized to predict its label: $f(G) = \hat{y}$. Typically, the graph classifier is a GNN-based model such as GCN, SAGE, and GAT [Kipf and Welling, 2017; Hamilton *et al.*, 2017; Veličković *et al.*, 2018].

3.2 Threat Model

Graph model owners can externalize the training to MLaaS providers and provision trained models for end-user interaction. However, adversaries can implant backdoors by compromised training processes or data-poisoning. To counter these vulnerabilities, defenders must implement countermeasures, while upholding privacy and intellectual property protection policies.

Adversary's Goals and Capabilities

Given a graph classifier $f(\cdot)$, the adversary aims to embed backdoor within $f(\cdot)$, which misclassifies graphs with trigger $\Delta = (V^*, E^*, X^*)$ (*e.g.*, specific subgraph) into premeditated class: $f(G \oplus \Delta) = y_\Delta$, while functioning normally on benign graphs: $f(G) = y$. To validate the efficacy of our method, we adopt the most stringent white-box attack setting, where adversaries can manipulate the training process, access model-concerned information, and acquire full training and additional datasets for attacks.

Defender's Goals and Capabilities

Current graph backdoor defenses work under white-box or gray-box settings, wherein the defenders have unrestricted or partial access to model-related knowledge, *e.g.*, parameters, layer embeddings, and accessible datasets. However, due to privacy policies and access restrictions, a black-box assumption is more realistic. In this study, we adopt a strictly black-box defense with only access to the input and limited queries.

4 Methodology

4.1 Overview

Given a suspicious test graph G, GRAPHPROT executes a three-step workflow to ensure robust inference, circumvent potential backdoor activation, and preserve benign sample accuracy. (1) We detect and purge potential anomalies by topology-feature-filtering mechanism. (2) Multiple subgraphs are sampled by leveraging both the structural topology and intrinsic node attributes. (3) These subgraphs are retained for predictions, with robust GNN inference aggregated via majority voting. The framework is shown in Fig. 1. The theoretically certified defense proof is articulated in Sec. 4.3.

4.2 Detailed Method

Graph Anomaly Filtration

This step aims to maximally mitigate explicit anomalous nodes V_{anomaly} (*i.e.*, outliers) within the test graph G (not full removal). Certain explicit V_{anomaly} manifest feature distributions that diverge substantially from the inherent properties

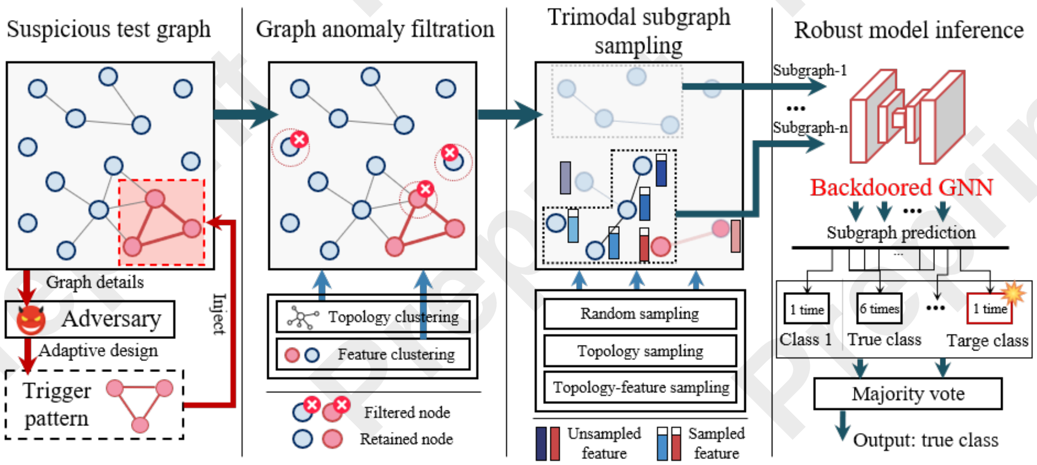


Figure 1: Depiction of the proposed GRAPHPROT, comprising three steps: (1) Graph Anomaly Filtration; (2) Trimodal Subgraph Sampling; and (3) Robust Model Inference. Initially, anomalous nodes are maximally pruned via a clustering-based heuristic. Subsequently, multiple subgraphs are extracted from the original graph by topology and features. Finally, a dedicated majority-vote ensemble classifier aggregates subgraph-level predictions and determines the robust output.

and structural regularities of G. Furthermore, V_{anomaly} negatively impact data predictions as they significantly deviate from original dataset distribution, resulting in inaccuracies or output instability. To identify them to the maximum degree, clustering methods are employed, exploiting the disparities in their feature distributions relative to the normal dataset. Using clustering, G is subdivided into anomalous and benign components, and the smaller portion is excluded, given that V_{anomaly} forms only a peripheral fraction of poisoned graphs. The filtration procedure is described as follows:

$$\mathcal{C}(G) = \{V_1, V_2\}, \tag{1}$$

$$G' = (V', E', X')$$
$$\text{s.t.} \begin{cases} V' = V \setminus \left(\arg\min_{V_i \in \{V_1, V_2\}} |V_i|\right) \\ E' = \{(u, v) \in E \mid u \in V' \land v \in V'\} \\ X' = \{x_i \mid x_i \in X \land v_i \in V'\}, \end{cases} \tag{2}$$

where $\mathcal{C}(\cdot)$ is the clustering function and G' signifies the filtered test graph.

For clustering $\mathcal{C}(\cdot)$, we harness the intrinsic attributes of graph data. Topological and feature-driven clustering methods are independently deployed to identify anomalies, acquiring discrete anomalous segments. The overlapping segment is considered anomalous, while the rest is deemed normal. The utilized clustering methods are outlined as follows:

- *Topology Clustering:* Some V_{anomaly} exhibit distinctive topology (*e.g.*, anomalous density patterns or connectivity structures) that varies markedly from G. The Spectral Clustering is used to divide nodes $V \subseteq G$ into two clusters via the adjacent matrix A (from E, V). The cluster with minimal cardinality is considered anomalous.
- *Feature Clustering:* V_{anomaly} may show distinct node feature distribution (*e.g.*, abnormal central tendency or dispersion). We utilize Gaussian Mixture to divide $V \subseteq G$ into two clusters via feature matrix $X \subseteq G$ and the cluster with lower cardinality is designated as anomalous.

Trimodal Subgraph Sampling

This step samples the filtered graph G' into K subgraphs. To this end, we propose three sampling strategies, *random sampling* (GRAPHPROT-R), *topology-sampling* (GRAPHPROT-T), and *topology-feature sampling* (GRAPHPROT-TF).

- *Random Sampling (GraphProt-R):* A subset of nodes $V_\mathcal{G} \in V'$ are randomly sampled according to the sample-rate $p = \frac{|V_\mathcal{G}|}{|V'|}$ and retain the topological and features of $V_\mathcal{G}$ to form a subgraph $\mathcal{G}$. This is demonstrated by

$$V_\mathcal{G} = \mathbb{S}(V', \lfloor p \cdot |V_\mathcal{G}| \rfloor), \tag{3}$$

$$\mathcal{G} = (V_\mathcal{G}, E_\mathcal{G}, X_\mathcal{G})$$
$$\text{s.t.} \begin{cases} E_\mathcal{G} = \{(u, v) \in E' \mid u \in V_\mathcal{G} \land v \in V_\mathcal{G}\} \\ X_\mathcal{G} = \{x_i \mid x_i \in X' \land v_i \in V_\mathcal{G}\}, \end{cases} \tag{4}$$

where $\mathbb{S}(\cdot, \cdot)$ refers to the random sampling operator with two inputs: the sampling target and sample size.

- *Topology Sampling (GraphProt-T):* Spectral clustering is employed on the adjacent matrix of G' to partition V' into $\left\lfloor \frac{|V'|}{K} \right\rfloor$ clusters. We then randomly draw one node from each cluster, with their topology and node attributes retained, to construct the subgraph $\mathcal{G}$. The process is detailed below:

$$\mathcal{S}(G', \left\lfloor \frac{|V'|}{K} \right\rfloor) = \{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_{\left\lfloor \frac{|V'|}{K} \right\rfloor}\}, \tag{5}$$

$$\mathcal{G} = (V_\mathcal{G}, E_\mathcal{G}, X_\mathcal{G})$$
$$\text{s.t.} \begin{cases} V_\mathcal{G} = \left\{ v_i \mid v_i \sim \mathcal{Q}_i, i = 1, 2, \ldots, \left\lfloor \frac{|V'|}{K} \right\rfloor \right\} \\ E_\mathcal{G} = \{(v_i, v_j) \mid v_i, v_j \in V_\mathcal{G}, (v_i, v_j) \in E'\} \\ X_\mathcal{G} = \{x_i \mid v_i \in V_\mathcal{G}, x_i \in X'\}, \end{cases} \tag{6}$$

where $\mathcal{S}(\cdot, \cdot)$ is the spectral clustering function with two inputs: the sampling target and the sample size.

- *Topology-feature Sampling (GraphProt-TF):* Based on the topology sampling results $\mathcal{G}$ of (GraphProt-T), we further conduct a feature-level selection. For each $V_{\mathcal{G}} \subseteq \mathcal{G}$, a fraction r of the feature dimensions is stochastically retained, preserving these values while zeroing the remainder to form the new node features $X'_{\mathcal{G}}$:

$$X'_{\mathcal{G}} = \left\{ x'_i \mid x'_i = x_i \cdot \mathbf{1}_{U_{\mathcal{G}}}, x_i \in X_{\mathcal{G}} \right\}$$
$$\textit{s.t.} \begin{cases} \mathcal{U}_{\mathcal{G}} \sim \mathbb{S}(\{1, 2, \ldots, d\}, \lceil r \cdot d \rceil) \\ (\mathbf{1}_{\mathcal{U}_{\mathcal{G}}})_j = \begin{cases} 1 & \text{if } j \in \mathcal{U}_{\mathcal{G}} \\ 0 & \text{otherwise}, \end{cases} \end{cases} \tag{7}$$

where the node feature vector x_i is d-dimensional, $\mathcal{U}_{\mathcal{G}}$ indicates the selected feature dimensions, and $\mathbf{1}_{\mathcal{U}_{\mathcal{G}}}$ denotes the binary mask vector.

Robust Model Inference

This step infers prediction of the suspicious test G from K sampled subgraphs $\{\mathcal{G}_k\}$. Given $\{\mathcal{G}_k\}$ and the victim model $f(\cdot)$, each $\mathcal{G}_k$ is individually classified via $f(\cdot)$, with the final output determined by a majority vote ensemble. This procedure can be delineated as follows:

$$g(G) = \arg\max_{y \in \mathcal{Y}} N_y, \tag{8}$$

$$N_y = \sum_{k=1}^{K} \mathbb{I}(f(\mathcal{G}_k) = y), \tag{9}$$

where $g(\cdot)$ is the ensemble classifier, N_y denotes the count of subgraphs predicted as the class y, and $\mathbb{I}(\cdot)$ represents the indicator function. In the event of a tie, the label corresponding to the smaller index is preferentially selected.

In the MLaaS setting, we take $g(G)$ as the final output for the suspicious test graph G, ensuring the prevention of backdoor activation while maintaining benign input accuracy. This approach operates within the constraint of querying the black-box GNN K times, without relying on auxiliary information or explainability tools.

4.3 Certifying Robustness

Although GraphProt is effective in the empirical evaluation, its resilience to adaptive attacks (*e.g.*, adaptive triggers & targets) remains uncertain. Hence, we formally propose a certified (provable) defense for our model GraphProt-R so that no further attacks can compromise the certified accuracy.

If the trigger involves injecting new nodes into the graph, we find the worst-case node number of the trigger graph. Let n_Δ denote the node numbers in poisoned subgraph $\mathcal{G}_\Delta$ (w/ trigger nodes and edges).

Theorem 1. *(Certified robustness for graph injection trigger). Given a testing graph G, a trained backdoored graph classifier f, and the ensemble classifier g defined in Eq.* (8) *with random subgraph sampling (GraphProt-R). Let $\mathcal{G}$ denote the subgraphs with s nodes sampled from G with replacement. Suppose y_A and y_B are the classes with the most votes and the second largest votes during the ensemble. We define $\underline{p_A}$ and $\overline{p_B}$ as the lower and upper bound of probability $\mathbb{P}(f(\mathcal{G}) = y_A)$ and $\mathbb{P}(f(\mathcal{G}) = y_B)$, respectively. We guarantee that the model still predicts class y_A for graphs G_Δ inserted with any trigger size smaller than r if:*

$$\max_{n_\Delta \leq n+r} (\frac{n_\Delta}{n})^s - 2(\frac{n_\Delta - r}{n})^s + 1 - (\underline{p_A} - \overline{p_B} - \delta_A - \delta_B) < 0, \tag{10}$$

where n and n_Δ are the node numbers in $\mathcal{G}$ and $\mathcal{G}_\Delta$, respectively, and $\delta_A = \underline{p_A} - (\lfloor \underline{p_A} \cdot n^s \rfloor)/n^s$, $\delta_B = (\lceil \overline{p_B} \cdot n^s \rceil)/n^s - \overline{p_B}$ are the residuals.

Note: In the main paper, $s = \lfloor p \cdot |V| \rfloor$.

Proof. See Appendix A in the supplementary materials. □

If the trigger is attached to the existing nodes (involves node feature modification and edge modification among r nodes), we have the following simplified certifying condition:

Theorem 2. *(Certified robustness for in-graph trigger). Given a testing graph G, a trained backdoored graph classifier f, and the ensemble classifier g defined in Eq.* (8) *with subgraph random subgraph sampling (GraphProt-R). Let $\mathcal{G}$ denote the subgraphs with s nodes sampled from G with replacement. Suppose y_A and y_B are the classes with the most votes and the second largest votes during the ensemble. We define $\underline{p_A}$ and $\overline{p_B}$ as the lower and upper bound of probability $\mathbb{P}(f(\mathcal{G}) = y_A)$ and $\mathbb{P}(f(\mathcal{G}) = y_B)$, respectively. We guarantee that the model still predicts class y_A for graphs G_Δ inserted with any trigger size smaller than r if:*

$$2(\frac{n-r}{n})^s > 1 - (\underline{p_A} - \overline{p_B} - \delta_A - \delta_B), \tag{11}$$

where n is the node numbers in G, and $\delta_A = \underline{p_A} - (\lfloor \underline{p_A} \cdot n^s \rfloor)/n^s$, $\delta_B = (\lceil \overline{p_B} \cdot n^s \rceil)/n^s - \overline{p_B}$ are the residuals.

Proof. See Appendix A in the supplementary materials. □

5 Experiment

In this section, we present the results of our comparative analyses and ablation studies on GraphProt. Notably, it functions within rigorous black-box conditions (with access limited to the test graph and few queries). Consequently, we primarily assess whether our approach achieves comparable efficacy to current defense strategies.

5.1 Experimental Settings

Victim Models

We exploited 3 state-of-the-art (SOTA) GNN models as benchmark targets for backdoor defense: (1) Graph Convolutional Network, GCN, extending convolution operations on graphs [Kipf and Welling, 2017]; (2) SAGE, which formulates node embeddings by sampling and aggregating neighborhood features [Hamilton *et al.*, 2017]; and (3) Graph Attention Network, GAT, deploying attention mechanisms to modulate node weights [Veličković *et al.*, 2018]. The models were subjected to backdoor attacks for subsequent defense evaluations.

GNN Arch.	Defense Method	Defense Performance (ASR%↓ | ADP%↓)					
		AIDS	ENZYMES	DHFR	NCI1	PROTEINS	COLLAB
GCN	Backdoored GCN*	99.8 | 0.7	97.1 | 1.7	100 | 2.2	99.5 | 2.2	74.4 | 0.2	94.4 | 1.9
	Benign GCN*	16.5 | 0.0	20.4 | 0.0	15.3 | 0.0	9.4 | 0.0	21.4 | 0.0	10.3 | 0.0
	GNNSECURER	21.2 | 8.3	26.8 | 8.4	17.2 | 5.9	22.0 | 5.2	32.4 | 6.5	16.5 | 4.9
	FINE-PRUNING	35.9 | 9.7	27.8 | 15.2	33.2 | 9.0	15.4 | 12.2	31.2 | 9.4	27.2 | 9.0
	RS	40.5 | 1.7	42.3 | 1.5	52.5 | 0.6	27.6 | 1.1	32.8 | 0.6	31.2 | 1.3
	GRAPHPROT-R	19.5 | 4.1	27.1 | 3.7	23.5 | 4.0	19.4 | 3.2	18.5 | 5.2	16.2 | 3.4
	GRAPHPROT-T	14.2 | 1.9	14.3 | 2.8	17.6 | 2.9	15.2 | 2.7	11.2 | 3.6	17.4 | 5.1
	GRAPHPROT-TF	5.8 | 4.9	9.7 | 7.5	14.0 | 6.9	16.4 | 4.6	19.7 | 4.1	19.1 | 7.9
SAGE	Backdoored SAGE*	100 | 0.8	95.3 | 2.0	100 | 1.5	97.1 | 1.8	70.4 | 1.8	98.0 | 0.5
	Benign SAGE*	15.3 | 0.0	18.6 | 0.0	20.1 | 0.0	11.4 | 0.0	17.4 | 0.0	17.0 | 0.0
	GNNSECURER	20.5 | 10.2	30.8 | 4.4	17.6 | 4.4	20.2 | 3.7	28.1 | 4.2	23.6 | 4.2
	FINE-PRUNING	38.5 | 6.9	23.3 | 4.7	35.1 | 13.1	16.6 | 9.0	36.7 | 13.9	30.2 | 9.2
	RS	43.2 | 2.4	41.2 | -0.9	33.1 | 0.4	45.0 | 0.9	55.4 | -0.9	41.6 | -0.8
	GRAPHPROT-R	27.3 | 6.8	25.6 | 2.9	19.1 | 4.7	21.5 | 2.1	24.1 | 5.4	23.6 | 2.2
	GRAPHPROT-T	12.0 | 4.0	18.7 | 4.4	12.4 | 4.3	14.3 | 1.3	20.2 | 3.9	21.4 | 1.5
	GRAPHPROT-TF	15.8 | 5.4	7.6 | 9.1	18.5 | 6.6	18.9 | 6.8	27.0 | 8.2	23.9 | 8.8
GAT	Backdoored GAT*	100 | 2.3	99.2 | 0.3	100 | 1.9	98.2 | 0.9	68.7 | 1.1	98.0 | 0.6
	Benign GAT*	18.3 | 0.0	20.4 | 0.0	14.5 | 0.0	12.8 | 0.0	19.8 | 0.0	12.0 | 0.0
	GNNSECURER	17.8 | 10.7	17.5 | 4.8	13.5 | 5.8	16.7 | 2.8	18.6 | 3.7	21.4 | 8.0
	FINE-PRUNING	28.0 | 7.5	31.1 | 7.6	29.2 | 9.8	18.2 | 7.4	29.8 | 13.6	54.4 | 12.1
	RS	30.1 | 8.0	29.6 | -1.1	40.1 | 1.8	36.7 | -0.4	57.3 | 0.7	36.7 | 0.2
	GRAPHPROT-R	17.9 | 1.6	24.8 | 3.8	17.5 | 5.6	16.4 | 1.9	22.1 | 4.1	21.9 | 3.1
	GRAPHPROT-T	14.7 | 4.7	18.0 | 2.7	14.1 | 5.4	15.8 | 2.1	19.1 | 5.4	19.8 | 3.7
	GRAPHPROT-TF	9.3 | 7.2	11.2 | 2.9	20.4 | 7.1	21.9 | 3.6	25.0 | 7.7	20.4 | 5.4

Table 1: GRAPHPROT defense performance across SOTA GNNs and benchmark datasets.

Attack Methodologies

We adopted 3 graph backdoor paradigms: (1) GTA, which implements a trigger generator to forge a graph model implanted with backdoor via bi-level optimization [Xi *et al.*, 2021]; (2) SBA, harnessesing intricately designed subgraph triggers to train the backdoored model [Zhang *et al.*, 2021]; and (3) Motif, which designs triggers using motif statistics to execute the attack [Zheng *et al.*, 2024].

Experiment Datasets

We employed 6 benchmark datasets: AIDS [Rossi and Ahmed, 2015], ENZYMES [Dobson and Doig, 2003], DHFR [Morris *et al.*, 2020], NCI1 [Wale and Karypis, 2006], PROTEINS [Borgwardt *et al.*, 2005], and COLLAB [Yanardag and Vishwanathan, 2015]. For each dataset, we randomly allocated two-thirds of the graphs for training the backdoored victim model, preserving the remainder for empirical testing.

Comparison Baselines

The SOTA white-box defenses were opted for comparative analysis: (1) GNNSECURER, incorporating topological saliency metrics and model-intrinsic interpretability for backdoor identification [Downer *et al.*, 2024]; (2) FINE-PRUNING, mitigating backdoor through the systematic excision of selective GNN parameters and iterative fine-tuning refinements [Liu *et al.*, 2018]; and (3) RS, robustified GNN framework, which fortifies robustness by introducing random noise into graph, coupled with classifier smoothing [Wang *et al.*, 2021].

Evaluation Metrics

We analyze GRAPHPROT from two dimensions: (1) defensive efficacy and (2) capacity to preserve benign input accuracy.

The effectiveness of defense mechanisms for poisoned graph is primarily assessed by *attack success rate* (*ASR*):

$$\text{Attack Success Rate } (ASR) = \frac{\#\text{successful trials}}{\#\text{total attack input trials}}, \tag{12}$$

wherein a diminished *ASR* reflects better defensive efficacy.

The preservation of GNN performance post-defense for benign samples is evaluated via *accuracy drop* (*ADP*) metric:

$$\text{Accuracy Drop } (ADP) = ACC_{cl} - ACC, \tag{13}$$

where ACC_{cl} and ACC represent GNN accuracies on benign data for the non-backdoored (benign) and backdoored models, respectively. A suppressed *ADP* signifies augmented performance preservation.

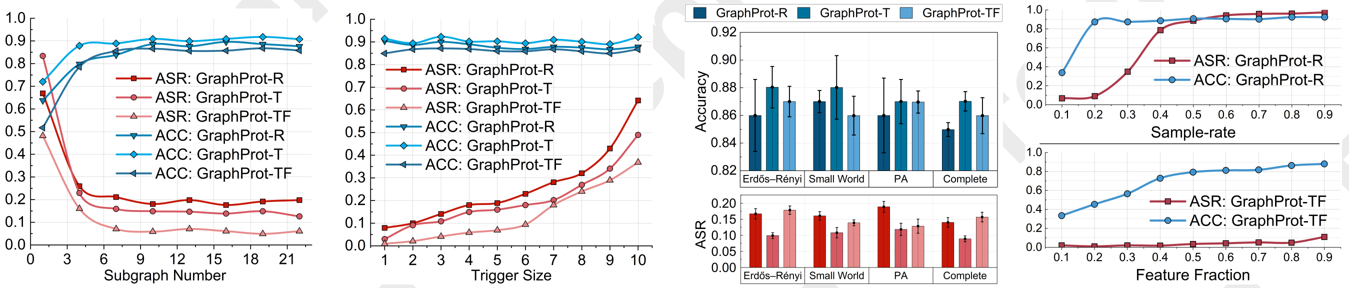
Additionally, we utilize benign data accuracy of backdoored GNN (*i.e.*, *ACC*) in ablation studies to observe variations.

5.2 Comparison Study

Our GRAPHPROT is evaluated in 2 aspects: (1) efficacy across various GNNs and datasets (*cf.* Tab. 1), and (2) robustness under diverse attack strategies and datasets (*cf.* Tab. 2). In the first evaluation, GTA is deployed as the attack (trigger size = 5, bi-level optimization epoch = 20). The second

Attack Method	Defense Method	Defense Performance (ASR% ↓ | ADP% ↓)					
		AIDS	ENZYMES	DHFR	NCI1	PROTEINS	COLLAB
SBA	Backdoored GCN*	59.2 | 3.0	74.2 | 1.5	79.2 | 1.2	66.7 | 0.7	73.4 | 3.2	82.4 | 4.2
	Benign GCN*	7.4 | 0.0	5.2 | 0.0	2.5 | 0.0	9.3 | 0.0	4.2 | 0.0	8.1 | 0.0
	GNNSECURER	19.5 | 7.2	23.6 | 12.7	28.8 | 7.6	19.5 | 6.0	24.8 | 6.5	29.3 | 9.5
	FINE-PRUNING	44.6 | 16.3	31.4 | 9.9	53.9 | 10.1	31.6 | 11.3	42.5 | 12.7	34.3 | 10.9
	RS	22.9 | 1.0	19.6 | 1.3	23.3 | 0.8	29.5 | 0.6	34.2 | 0.3	27.1 | 3.5
	GRAPHPROT-R	17.6 | 6.3	16.2 | 5.9	23.8 | 2.3	21.3 | 3.2	26.5 | 6.1	19.7 | 6.4
	GRAPHPROT-T	10.0 | 3.8	9.4 | 4.9	14.5 | 1.2	10.7 | 2.6	12.5 | 3.4	14.5 | 5.0
	GRAPHPROT-TF	18.9 | 5.5	22.1 | 6.7	16.9 | 5.1	17.6 | 7.1	19.1 | 5.3	18.2 | 6.3
Motif	Backdoored GCN*	96.5 | 1.2	82.4 | 3.3	93.4 | 1.2	94.8 | 0.2	87.8 | 0.9	83.6 | 1.1
	Benign GCN*	8.7 | 0.0	6.4 | 0.0	14.5 | 0.0	8.1 | 0.0	3.9 | 0.0	7.5 | 0.0
	GNNSECURER	14.2 | 7.7	16.3 | 6.9	13.5 | 2.1	19.6 | 5.9	21.8 | 4.6	22.4 | 7.6
	FINE-PRUNING	37.1 | 14.3	39.7 | 13.9	29.2 | 6.1	48.2 | 9.8	39.9 | 12.5	44.5 | 13.1
	RS	21.2 | 3.8	30.3 | 1.2	40.1 | -1.9	31.6 | 0.9	33.7 | 0.3	29.4 | 0.8
	GRAPHPROT-R	21.4 | 2.8	19.1 | 4.8	22.6 | 1.9	18.6 | 4.0	19.2 | 3.6	20.5 | 3.3
	GRAPHPROT-T	17.8 | 1.9	15.4 | 3.6	14.1 | 1.7	13.5 | 2.4	12.1 | 2.5	17.3 | 3.8
	GRAPHPROT-TF	13.2 | 5.6	13.5 | 4.9	20.4 | 3.4	19.9 | 7.3	21.7 | 8.6	19.4 | 7.1

Table 2: GRAPHPROT defense performance across attack threats and benchmark datasets.



(a) Subgraph Number (b) Trigger Size (c) Trigger Pattern (d) Sample-Rate & Feature

Figure 2: Ablation analysis of GRAPHPROT: Influence of (a) Subgraph Number, (b) Trigger Size, (c) Trigger Pattern, and (d) Sample-Rate and Feature Fraction. Empirical findings demonstrate that (1) augmenting subgraph numbers elevates *ACC* and suppresses *ASR*, (2) larger trigger sizes predominantly intensify *ASR*, (3) the impact of trigger pattern on *ACC* and *ASR* is minimal, and (4) both *ACC* and *ASR* exhibit a positive correlation with incremental sampling-rates and expanded feature fractions.

evaluation used SBA (5-node Erdős–Rényi trigger) and Motif (trigger size $= 5$) for attacking GCN. The GRAPHPROT configuration includes the subgraph number $K = 5$, the sample-rate $p = 0.2$, and the feature selection proportion $r = 0.8$.

Defense Results across GNNs and Datasets

From Tab. 1, the following observations can be discerned: (1) regarding defensive efficacy, generally, GRAPHPROT > GNNSECURER > FINE-PRUNING > RS, with $\overline{ASR}$ values of 19.7%, 20.9%, 34.1%, and 37.9%, respectively. GRAPHPROT achieves the most lowest *ASRs* and demonstrates better overall anti-backdoor capability (solely requires test input & K queries). (2) In terms of benign input accuracy preservation, RS > GRAPHPROT > FINE-PRUNING > GNNSECURER, with $\overline{ADP}$ values of 1.3%, 5.3%, 8.0%, and 9.7%, respectively. GRAPHPROT closely aligns with SOTA white-box defense paradigms and shows slight $\overline{ADP}$ disparity, underscoring its competitive efficacy in preserving regular performance. (3) The *ASRs* of benign models marginally surpass that of GRAPHPROT and GNNSECURER across several cases (*e.g.*, under AIDS set & GAT) due to the adversarial attack nature of GTA, which remain partially effective without explicit backdoor training.

Defense Results across Attacks and Datasets

From Tab. 2, the following points can be identified: (1) GRAPHPROT demonstrates better overall defense efficacy with consistently lower $\overline{ASR}$ (GRAPHPROT: $17.2\% <$ GNNSECURER : $24.3\% <$ RS : $26.1\% <$ FINE-PRUNING : 39.7%), highlighting its robustness across attacks and datasets. (2) With respect to benign data accuracy retention, GRAPHPROT attains an $\overline{ADP}$ of 4.5%, surpassed solely by RS (1.3%), and exceeding the performance of GNNSECURER (8.3%) and FINE-PRUNING (11.9%). (3) GNNSECURER ranks as the second most effective defense, and FINE-PRUNING exhibits the least effective performance, with the highest $\overline{ASR}$ and $\overline{ADP}$.

5.3 Ablation Study

We delve into the key factors influencing GraphProt-R, GraphProt-T, and GraphProt-TF via comprehensive ablation studies, examining (1) subgraph number, (2) trigger size, (3) trigger pattern, and (4) sample-rate and feature fraction. This study leveraged the GTA attack on GCN trained via AIDS set, and the results are presented in Fig. 2.

Subgraph Number

Multiple subgraph numbers K were configured at intervals of 3, ranging from 1 to 22, to examine variations in *ASR* and *ADP*. The results are shown in Fig. 2a.

From the figure, with increasing K, the *ASR* declines, with GraphProt-TF $= 6\% <$ GraphProt-T $= 14\%$ $<$ GraphProt-R $= 19\%$, highlighting improved robustness. Meanwhile, the *ACC* for all three methods improves and progressively stabilizes, with GraphProt-T $= 90\% >$ GraphProt-R $= 87\% >$ GraphProt-R $= 86\%$. Generally, GraphProt-TF achieves the best defense but incurs the lowest *ACC* due to joint topology-feature sampling. Conversely, GraphProt-R minimizes *ACC* degradation but yields the highest *ASR*, with GraphProt-T exhibiting intermediary performance.

Trigger Size

We executed attacks with trigger sizes t ranging from 1 to 10 and deployed GraphProt next. Notably, the average graph size in AIDS set is 15.69, indicating that a trigger size $t = 8$ exceeds the halfway threshold. The findings are illustrated in Fig. 2b.

For defense efficacy, as t escalates, *ASR* rises across all methods, with defense efficacy following the hierarchy of GraphProt-TF $>$ GraphProt-T $>$ GraphProt-R. Concerning the benign data accuracy, the fluctuation in $\overline{ACC}$ remains negligible (variations $< 3\%$). The $\overline{ACC}$ is highest for GraphProt-T, succeeded by GraphProt-R and GraphProt-TF. Furthermore, when $t \approx 8$ (half the average graph size), the *ASR* across all methods rises sharply. GraphProt-TF exhibits the most restrained increase, likely attributed to its node feature sampling mitigating trigger.

Trigger Pattern

The evaluation implemented SBA attack using diverse trigger types (w/ 5 nodes): (1) Erdős-Rényi Graph, (2) Small World Graph, (3) Preferential Attachment (PA) Graph, and (4) Complete Graph. The defense results are shown in Fig. 2c.

Overall, the differences in *ASR* and *ACC* among defense methods utilizing various triggers are marginal, suggesting consistent defense efficacy across trigger types. GraphProt-R demonstrates moderate *ACC* but encounters limitations with elevated *ASR*, particularly in Preferential Attachment networks. Conversely, GraphProt-T attains better *ACC* alongside the lowest $\overline{ASR}$, whereas GraphProt-TF upholds both robust *ACC* and minimal *ASR*.

Sample-Rate and Feature Fraction

We first evaluate how the sample-rate p of GraphProt-R influences defense efficacy. Sample-rates $(0 - 100\%)$ were applied for GraphProt-R, with the outcomes shown in Fig. 2d. Referring to the figure, at higher p, both *ASR* and *ACC* increase, with *ACC* rising more steeply. When $p = 0.2$, *ASR* remains comparatively low, while *ACC* achieves a relatively high level.

We subsequently adjusted the feature fraction r of GraphProt-TF across $[0, 100]\%$ to evaluate its impact (results illustrated in Fig. 2d). Based on the figure, when r increases, both *ASR* and *ACC* exhibit upward trends, though *ACC* grows at a more pronounced rate. Peak performance is observed at $r \approx 0.8$, where *ACC* approximately achieves its peak with minimal *ASR*, indicating an optimal balance.

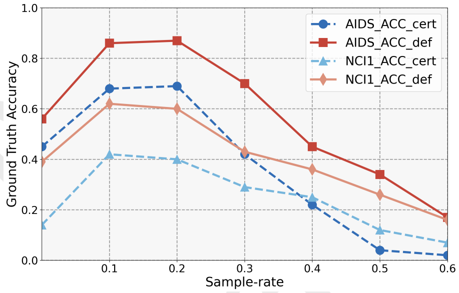


Figure 3: Certified robustness analysis: certified accuracy ACC_{cert} vs. defense data accuracy ACC_{def} under AIDS & NCI1 datasets.

5.4 Certified Robustness Study

In Sec. 4.3, we elucidate the theoretically certified robustness of GraphProt. Furthermore, we inspect the discrepancy between the achieved defensive accuracy of our method and its theoretically certified accuracy. We conduct GTA attacks against GCN trained on the AIDS and NCI1 sets. After applying GraphProt-R (w/ $K = 1000$ & multiple sample-rates p), we examine certified accuracy ACC_{cert} (*i.e.*, benign data accuracy conforming to Eq. (10)) and defense data accuracy ACC_{def} (*i.e.*, poisoned data accuracy satisfying Eq. (11)). The findings are illustrated in Fig. 3.

The results across datasets indicate that as p increases, ACC_{cert} and ACC_{def} exhibit non-monotonic trends, first rising and then falling. When $p \approx 20\%$, both ACC_{cert} and ACC_{def} attain their maxima, thereafter converging towards 0 as $p \rightarrow 60\%$. This phenomenon stems from the amplification of both benign and malicious node features with increasing p, finally causing a sharp degradation in $\overline{ACC}_{\text{def}}$. To equilibrate benign and poisoned sample accuracy, p is set to 20%. For the AIDS and NCI1 sets, the $\overline{ACC}_{\text{def}}$ reached 56.4% and 40.3%, respectively, exceeding the $\overline{ACC}_{\text{cert}}$ of 36.0% and 24.1%.

6 Conclusion

We introduce GraphProt, an input-dependent black-box defense strategy requiring no ancillary data, external tools, or model specifications. Our approach mitigates backdoor activation by leveraging topology-feature-filtering and sampling-based robust model inference. We further provide GraphProt with a formally certified robustness guarantee. Empirical evaluations conducted on multiple attack paradigms and benchmark datasets confirm its effectiveness in reducing attack success rates while preserving benign data accuracy.

Acknowledgments

This research is supported by National Nature Science Foundation of China (Nos. 62202303, U20B2048, and U2003206), Shanghai Sailing Program (No. 21YF1421700), Action Plan of Science and Technology Innovation of Science and Technology Commission of Shanghai Municipality (No. 22511101202), and the Financial Support for Non-PAIR Research Centres (PolyU 1-CE1N).

Contribution Statement

†Xiao Yang and Yuni Lai contributed equally.

References

[Alrahis *et al.*, 2023] Lilas Alrahis, Satwik Patnaik, Muhammad Abdullah Hanif, Muhammad Shafique, and Ozgur Sinanoglu. Poisonedgnn: Backdoor attack on graph neural networks-based hardware security systems. *IEEE Transactions on Computers*, 72(10):2822–2834, 2023.

[Borgwardt *et al.*, 2005] Karsten M. Borgwardt, Cheng Soon Ong, Stefan Schönauer, S. V. N. Vishwanathan, Alex J. Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(1):47–56, 2005.

[Dobson and Doig, 2003] Paul D. Dobson and Andrew J. Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of Molecular Biology*, 330(4):771–783, 2003.

[Downer *et al.*, 2024] Jane Downer, Ren Wang, and Binghui Wang. Securing gnns: Explanation-based identification of backdoored training graphs, 2024.

[Fan *et al.*, 2019] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *International World Wide Web Conference*, 2019.

[Hamilton *et al.*, 2017] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Conference on Neural Information Processing Systems*, 2017.

[Jiang and Li, 2022] Bingchen Jiang and Zhao Li. Defending against backdoor attack on graph nerual network by explainability, 2022.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[Li *et al.*, 2024] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):5–22, 2024.

[Liu *et al.*, 2018] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions and Defenses*, 2018.

[Lyu *et al.*, 2024] Xiaoting Lyu, Yufei Han, Wei Wang, Hangwei Qian, Ivor Tsang, and Xiangliang Zhang. Cross-context backdoor attacks against graph prompt learning, 2024.

[Morris *et al.*, 2020] Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. In *International Conference on Machine Learning Workshop*, 2020.

[Rossi and Ahmed, 2015] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI Conference on Artificial Intelligence*, 2015.

[Safae *et al.*, 2023] Hmaidi Safae, Lazaar Mohamed, Abdellah Chehri, El Madani El Alami Yasser, and Rachid Saadane. Link prediction using graph neural networks for recommendation systems. *Procedia Computer Science*, 225:4284–4294, 2023.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[Wale and Karypis, 2006] Nikil Wale and George Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. In *International Conference on Data Mining*, 2006.

[Wang *et al.*, 2021] Binghui Wang, Jinyuan Jia, Xiaoyu Cao, and Neil Zhenqiang Gong. Certified robustness of graph neural networks against adversarial structural perturbation. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2021.

[Wang *et al.*, 2024] Kaiyang Wang, Huaxin Deng, Yijia Xu, Zhonglin Liu, and Yong Fang. Multi-target label backdoor attacks on graph neural networks. *Pattern Recognition*, 2024.

[Wieder *et al.*, 2020] Oliver Wieder, Stefan Kohlbacher, Mélaine Kuenemann, Arthur Garon, Pierre Ducrot, Thomas Seidel, and Thierry Langer. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies*, 37:1–12, 2020.

[Wu *et al.*, 2021] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021.

[Xi *et al.*, 2021] Zhaohan Xi, Ren Pang, Shouling Ji, and Ting Wang. Graph backdoor. In *USENIX Security*, 2021.

[Xu *et al.*, 2021] Jing Xu, Minhui (Jason) Xue, and Stjepan Picek. Explainability-based backdoor attacks against graph neural networks. In *ACM Workshop on Wireless Security and Machine Learning*, 2021.

[Xu *et al.*, 2022] Jing Xu, Rui Wang, Stefanos Koffas, Kaitai Liang, and Stjepan Picek. More is better (mostly): On the backdoor attacks in federated graph neural networks.

In *Annual Computer Security Applications Conference*, 2022.

[Yanardag and Vishwanathan, 2015] Pinar Yanardag and S.V.N. Vishwanathan. Deep graph kernels. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2015.

[Yang *et al.*, 2022] Shuiqiao Yang, Bao Gia Doan, Paul Montague, Olivier De Vel, Tamas Abraham, Seyit Camtepe, Damith C. Ranasinghe, and Salil S. Kanhere. Transferable graph backdoor attack. In *International Symposium on Research in Attacks, Intrusions and Defenses*, 2022.

[Yang *et al.*, 2024] Xiao Yang, Gaolei Li, Xiaoyi Tao, Chaofeng Zhang, and Jianhua Li. Black-box graph backdoor defense. In *International Conference on Algorithms and Architectures for Parallel Processing*, 2024.

[Zhang *et al.*, 2021] Zaixi Zhang, Jinyuan Jia, Binghui Wang, and Neil Zhenqiang Gong. Backdoor attacks to graph neural networks. In *ACM Symposium on Access Control Models and Technologies*, 2021.

[Zhang *et al.*, 2023] Hangfan Zhang, Jinghui Chen, Lu Lin, Jinyuan Jia, and Dinghao Wu. Graph contrastive backdoor attacks. In *International Conference on Machine Learning*, 2023.

[Zhang *et al.*, 2024] Zhiwei Zhang, Minhua Lin, Junjie Xu, Zongyu Wu, Enyan Dai, and Suhang Wang. Robustness-inspired defense against backdoor attacks on graph neural networks, 2024.

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

[Zheng *et al.*, 2024] Haibin Zheng, Haiyang Xiong, Jinyin Chen, Haonan Ma, and Guohan Huang. Motif-backdoor: Rethinking the backdoor attack on graph neural networks via motifs. *IEEE Transactions on Computational Social Systems*, 11(2):2479–2493, 2024.