ARTICLE

DSGNN: Dual-Shield Defense for Robust Graph Neural Networks

Xiaohan Chen[1], Yuanfang Chen[1,*], Gyu Myoung Lee[2], Noel Crespi[3] and Pierluigi Siano[4]

[1]School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China
[2]School of Computer Science and Mathematics, Liverpool John Moores University, Liverpool L3 3AF, UK
[3]Telecom SudParis, Institut Polytechnique de Paris, Evry 91011, France
[4]Department of Management & Innovation Systems, University of Salerno, 84084 Salerno, Italy
[*]Corresponding author: Yuanfang Chen. Email: yuanfang.chen.tina@gmail.com

Version July 8, 2025 submitted to Journal Not Specified

ABSTRACT: Graph Neural Networks (GNNs) have demonstrated outstanding capabilities in processing graph-structured data and are increasingly being integrated into large-scale pre-trained models, such as Large Language Models (LLMs), to enhance structural reasoning, knowledge retrieval, and memory management. The expansion of their application scope imposes higher requirements on the robustness of GNNs. However, as GNNs are applied to more dynamic and heterogeneous environments, they become increasingly vulnerable to real-world perturbations. In particular, graph data frequently encounters joint adversarial perturbations that simultaneously affect both structures and features, which are significantly more challenging than isolated attacks. These disruptions, caused by incomplete data, malicious attacks, or inherent noise, pose substantial threats to the stable and reliable performance of traditional GNN models. To address this issue, this study proposes the Dual-Shield Graph Neural Network (DSGNN), a defense model that simultaneously mitigates structural and feature perturbations. DSGNN utilizes two parallel GNN channels to independently process structural noise and feature noise, and introduces an adaptive fusion mechanism that integrates information from both pathways to generate robust node representations. Theoretical analysis demonstrates that DSGNN achieves a tighter robustness boundary under joint perturbations compared to conventional single-channel methods. Experimental evaluations across Cora, CiteSeer, and Industry datasets show that DSGNN achieves the highest average classification accuracy under various adversarial settings, reaching 81.24%, 71.94%, and 81.66% respectively, outperforming GNNGuard, GCN-Jaccard, GCN-SVD, RGCN, and NoisyGNN. These results underscore the importance of multi-view perturbation decoupling in constructing resilient GNN models for real-world applications.

KEYWORDS: Graph Neural Networks; Adversarial Attacks; Dual-Shield Defense; Certified Robustness; Node Classification

1 Introduction

Graph Neural Networks (GNNs) have demonstrated outstanding effectiveness in modeling non-Euclidean data structures and have been widely adopted across various domains, including social network analysis, multimedia recommendation, anomaly detection, and molecular property prediction [1]. This success is largely attributed to the message-passing mechanism, which has become a cornerstone in GNN architectures [2], wherein each node iteratively refines its representation by aggregating information from its neighbors. Through this process, GNNs

effectively capture both structural and semantic information, enabling key tasks such as node classification, graph clustering, and link prediction.

Meanwhile, the rapid advancement of large-scale pre-trained models, particularly Large Language Models (LLMs), has significantly expanded the scale and complexity of AI systems, driving an increasing demand for enhanced structural reasoning, knowledge retrieval, and memory management capabilities. In response to these evolving needs, Graph Neural Networks (GNNs) are being increasingly integrated into LLM frameworks and broader AI architectures [3], highlighting their critical role in supporting scalable and robust intelligent systems. Beyond language models, the growing deployment of intelligent systems in complex and dynamic environments—such as anomaly detection [4], real-time intrusion detection in dynamic graph environments [5], visual security probing through adversarial attacks [6], dynamic recommender systems, decentralized financial networks (DeFi), and autonomous driving perception graphs—further underscores the importance of robust graph representation learning. These trends collectively emphasize the pivotal role of GNNs in enabling dynamic reasoning and resilient modeling across diverse real-world applications.

Although GNNs have achieved impressive results on benchmark datasets, these evaluations often assume clean input features and ideal graph structures. However, in real-world scenarios, graph data frequently contains inherent noise, such as irrelevant or misleading edges, and is subject to external disturbances such as adversarial perturbations designed to degrade model performance [7]. In addition to traditional attacks, recent studies have shown that even federated graph learning settings are vulnerable to property inference attacks, exposing sensitive structural information without direct access to the raw graph data [8]. Even slight modifications to node attributes or graph connectivity can substantially impact GNN performance, although these changes often difficult to detect. Nevertheless, existing defense strategies often suffer from two major limitations. First, they are typically designed to handle either structural perturbations or feature perturbations, but not both. This siloed approach limits their effectiveness in real-world scenarios where attackers often employ joint or compound strategies that disrupt both the graph topology and node features concurrently. Second, these methods usually lack adaptive mechanisms capable of dynamically assessing the nature and intensity of the perturbations during inference. As a result, their robustness deteriorates significantly when confronted with sophisticated attacks that exhibit non-uniform or evolving patterns. The absence of an integrated, context-aware defense mechanism thus remains a critical gap in current adversarial robustness research for graph neural networks. These constraints motivate the development of a unified framework capable of decoupling and robustly integrating multi-view perturbations.

To address these challenges, this work proposes DSGNN, a Dual-Channel Shielded Graph Neural Network. DSGNN incorporates two parallel propagation channels, with one dedicated to structural noise and the other to feature perturbations. As illustrated in Fig. 1, the input graph is initially processed through independent structure and feature defense modules. Traditional methods [9,10], depicted in the upper part of the figure, treat each perturbation type separately. Although effective in certain scenarios, single-channel approaches often struggle when structural and feature perturbations coexist, missing critical cross-modal interactions.

The resulting representations are propagated through the dual channels and subsequently fused at the final stage (lower part of the figure). This dual-channel decoupling and weighted fusion

mechanism enables unified robustness learning, thereby enhancing resilience against complex adversarial attacks.

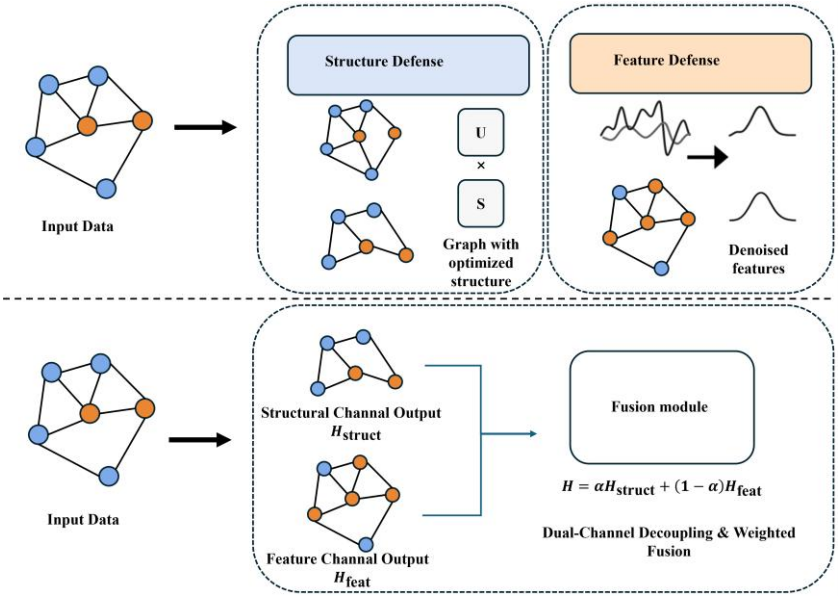


Figure 1: Comparison between the DSGNN model and traditional defense methods. The upper section illustrates traditional defenses, where the input graph is processed through either a structure defense module or a feature defense module, with each module handling a specific perturbation type independently. The lower section shows the proposed DSGNN approach, where the input graph passes through both structural and feature channels simultaneously, followed by a weighted fusion to generate robust node representations. Here, U and S denote the left singular vectors and the singular values obtained from the singular value decomposition (SVD) of the adjacency matrix, respectively.

Extensive experiments on Cora, CiteSeer and Industry datasets validate the effectiveness of the proposed model. DSGNN consistently outperforms existing defense methods under various attack settings, including DICE, PGD, and Metattack, while maintaining competitive performance on noise-free data. These findings underscore the importance of multi-view perturbation modeling in developing robust and generalizable GNN architectures.

Main Contributions

- **DSGNN Architecture:** A novel model that explicitly decouples the propagation paths of structural and feature perturbations, thereby enhancing adaptability to compound adversarial attacks.
- **Theoretical Robustness Bound:** A theoretical upper bound on robustness risk under joint perturbations is derived, demonstrating that DSGNN provides tighter guarantees compared to conventional defenses.
- **Comprehensive Evaluation:** Extensive empirical results across multiple datasets validate the superior robustness of DSGNN against diverse adversarial threats, while maintaining minimal performance degradation on clean inputs.

2 Related Work

GNNs have demonstrated strong capabilities across a variety of graph-based learning tasks. However, in practical applications, even slight perturbations in node features or graph structures can lead to significant performance degradation, particularly in security-sensitive scenarios. To address this issue, numerous defense strategies have been proposed, which can be broadly categorized as follows:

1. Graph structure preprocessing methods. These approaches statically modify the graph before training to improve model robustness. For instance, GCN-Jaccard [11] removes low-quality edges based on feature similarity, while GCN-SVD [9] suppresses high-frequency noise via low-rank approximation of the adjacency matrix. These methods are computationally efficient and suitable for static graphs but struggle to defend against test-time attacks and lack adaptability to dynamic graphs.
2. Adversarial training methods. Methods such as GraphAT [12] generate adversarial samples during training and jointly optimize node features and graph structures to enhance resistance against attacks. Although effective under strong adversarial conditions, they often incur high training costs and suffer from poor scalability on large graphs.
3. Architecture enhancement methods. RobustGCN [13], GNNGuard [10], and related approaches improve robustness by modifying the message-passing mechanism—modeling neighbor importance, adjusting attention weights, or by reassigning edge weights—to mitigate the impact of structural perturbations. While these methods are effective against structural attacks, they often neglect feature perturbations, leading to limited adaptability under joint structural-feature attacks. Recent extensions, such as GCORN [14] and β-GNN [15], attempt to address this gap from different perspectives. GCORN enhances theoretical robustness under feature noise through orthonormal weight constraints, while β-GNN dynamically fuses the outputs of a GNN and an MLP to resist structural perturbations and provide interpretable attack indicators. However, these designs either increase training complexity or target only one type of perturbation.
4. Noise injection mechanisms. Recent lightweight approaches introduce random noise into model weights or activations to improve robustness. NoisyGNN [16] injects Gaussian noise into hidden layers to enhance resistance against adversarial attacks.
5. Feature and structure regularization methods. Approaches such as Pro-GNN [17] and RGCN [18] incorporate smoothness constraints on features and structures to improve tolerance to local anomalies. However, these methods often assume smoothness or homophily within the graph, making them unstable on sparse or non-stationary graphs.

Despite their individual strengths, the methods described above share two common limitations:

1. Most focus exclusively on either structural or feature perturbations, failing to address the realistic scenarios where both types co-occur [19];
2. They lack the capability to model the interactions between different perturbation sources, rendering them vulnerable to compound adversarial attacks.

Table 1: Comparison of Defense Methods for GNN (perturbation types involved)

Category	Representative Methods	Node Perturbation	Edge Perturbation	Limitations
Graph Preprocessing	GCN-Jaccard [11], GCN-SVD [9]	✗	✓	Unable to handle unseen perturbations during testing; limited adaptability
Adversarial Training	GraphAT [12]	✓	✓	High training cost; difficult to scale to large graphs
Feature/Structure Regularization	Pro-GNN [17], RGCN [18]	✗	✗	Robustness relies on graph smoothness; not specifically designed for adversarial settings
Architecture Enhancement	GNNGuard [10], RobustGCN [13]	✗	✓	Focuses on structure perturbation; limited generalization to joint attacks
Noise Injection	NoisyGNN [16]	✓	✗	Effectiveness unstable; lacks explicit modeling of perturbation types
DSGNN	–	✓	✓	Requires dual-channel design; increases model parameters and training complexity

3 Problem Definition

The core notations used throughout this paper are summarized in Table 2.

Given an undirected graph $G = (V, E)$, the adjacency matrix $A \in \mathbb{R}^{n \times n}$ and the node feature matrix $X \in \mathbb{R}^{n \times d}$ serve as the basic input to a graph neural network. A typical message-passing layer can be formulated as:

$$H^{(l+1)} = \sigma(\hat{A} H^{(l)} W^{(l)}), \tag{1}$$

where $\hat{A}$ denotes the normalized adjacency matrix, $H^{(l)}$ is the node representation at layer l, $W^{(l)}$ is the trainable weight matrix, and $\sigma(\cdot)$ is a non-linear activation function.

In real-world deployments, the feature matrix X often contains noise (e.g., sensor measurement errors) or adversarial manipulations (e.g., forged user profiles), while the adjacency matrix A may include spurious edges (e.g., fake links in social networks) or missing critical connections (e.g., due to limitations in biological network observations).

Feature Perturbation Set. The permissible feature perturbation set is defined as:

$$\mathbb{B}_X = \left\{ \tilde{X} \mid \| \tilde{X} - X \|_F \leq \epsilon_X \right\}, \tag{2}$$

where $\| \cdot \|_F$ denotes the Frobenius norm, measuring the magnitude of deviations between the original and perturbed feature matrices. This constraint assumes that measurement errors typically follow a Gaussian distribution, and their total magnitude can be bounded by ϵ_X.

Structure Perturbation Set. Similarly, the permissible structure perturbation set is defined as:

$$\mathbb{B}_A = \left\{ \tilde{A} \mid \| \tilde{A} - A \|_0 \leq \epsilon_A \right\}, \tag{3}$$

Table 2: Core Notations

Symbol	Meaning
$G = (V, E)$	Undirected graph with node set V and edge set E
$A \in \mathbb{R}^{n \times n}$	Adjacency matrix
$X \in \mathbb{R}^{n \times d}$	Node feature matrix
$\tilde{A}, \tilde{X}$	Perturbed adjacency and feature matrices
$\mathbb{B}_A, \mathbb{B}_X$	Structural and feature perturbation sets
$f(\cdot)$	GNN classification function
Y	Node label set
$H^{(l)}$	Node representation at layer l
$\sigma(\cdot)$	Non-linear activation function
$W^{(l)}$	Trainable weights at layer l
α	Fusion weight between structure and feature channels
$d_Y(\cdot, \cdot)$	Distance function in the output space

where $\| \cdot \|_0$ counts the number of modified edges. This formulation captures practical attack cost constraints, as excessive edge modifications (exceeding ϵ_A) are more likely to trigger detection mechanisms in real-world systems.

Robustness Objective. Given the above perturbation models, the robustness of a GNN can be formulated as the following min-max optimization problem:

$$\min_{f} \mathbb{E}_{(A,X)} \left[\max_{(\tilde{A},\tilde{X}) \in \mathbb{B}_A \times \mathbb{B}_X} \mathbb{L}(f(\tilde{A}, \tilde{X}), Y) \right], \tag{4}$$

where $\mathbb{L}(\cdot, \cdot)$ denotes the classification loss function, and the inner maximization seeks the worst-case perturbations within the defined budgets.

Challenges of Coupled Perturbations. It is important to emphasize that structural perturbations alter the global topology and disrupt message-passing paths, while feature perturbations directly distort the semantic representations of nodes. The coupling between these two perturbation types can lead to compounded degradation, exceeding the impact of either perturbation individually. This highlights why existing defense strategies that treat structure and feature perturbations independently are often insufficient.

Definition of Joint Perturbations. In this paper, we define *joint adversarial perturbations* as attack scenarios where both the graph structure and node features are perturbed in the same instance. Importantly, these perturbations are applied **simultaneously but independently**, meaning that the modifications to the adjacency matrix and the feature matrix are not assumed to be statistically or functionally dependent. This formulation captures realistic attack settings where both topological and semantic information may be corrupted, either by coordinated or uncoordinated attackers, and serves as a general framework that covers compound perturbation types.

4 Methodology

4.1 Model Overview

The DSGNN model is proposed to enhance the robustness of GNNs against simultaneous feature and structural perturbations. DSGNN decouples the propagation paths of the two perturbation types by constructing independent information channels. Each channel learns robust node representations, and the outputs are subsequently combined through a dynamic fusion layer.

DSGNN is designed to be broadly applicable across different GNN architectures. In the experimental evaluation, the model is instantiated based on Graph Convolutional Networks (GCNs) to validate its effectiveness.

Unlike regularization-based or adversarial training methods, DSGNN model mitigates structural and feature perturbations through architectural separation. Controlled noise is introduced into the learning process to better simulate real-world distribution shifts and improve model robustness.

4.2 Dual-Channel Processing Pipeline

Input graph $G = (V, E)$, where $A \in \mathbb{R}^{n \times n}$ denotes the adjacency matrix and $X \in \mathbb{R}^{n \times d}$ denotes the node features. To model perturbations, two permissible sets are defined: $\mathbb{B}_A$ for structural perturbations and $\mathbb{B}_X$ for feature perturbations.

The proposed DSGNN model introduces two parallel GNN channels, each responsible for independently processing structural and feature noise.

- **Structural channel:** Processes clean features with perturbed structure, i.e., $H_{\text{struct}} = \text{GNN}(X, \tilde{A})$.
- **Feature channel:** Processes perturbed features with clean structure, i.e., $H_{\text{feat}} = \text{GNN}(\tilde{X}, A)$.

To ensure consistency and comparability, both the structural and feature channels in DSGNN are implemented using standard two-layer Graph Convolutional Networks (GCNs), each with a hidden size of 256. This architecture mirrors the configuration adopted in NoisyGNN [16], enabling fair benchmarking and reproducibility in experimental comparisons.

Each channel is specifically trained to address a distinct type of perturbation:

- **The structural channel** processes the input pair $(\tilde{A}, X)$, allowing the model to learn robustness against structural perturbations while preserving clean node features.
- **The feature channel** processes $(A, \tilde{X})$, focusing on mitigating feature noise while maintaining the original graph topology.

During training, separate perturbations are sampled for both A and X in each epoch. The two GCN channels are optimized jointly based on their respective perturbed inputs. This decoupled learning mechanism enables each channel to specialize in its assigned perturbation type. The outputs of the two channels are then integrated via an adaptive attention-based fusion layer, which dynamically weighs each representation according to its reliability under perturbation. This architecture enhances robustness by allowing the model to prioritize more trustworthy modalities in complex attack scenarios.

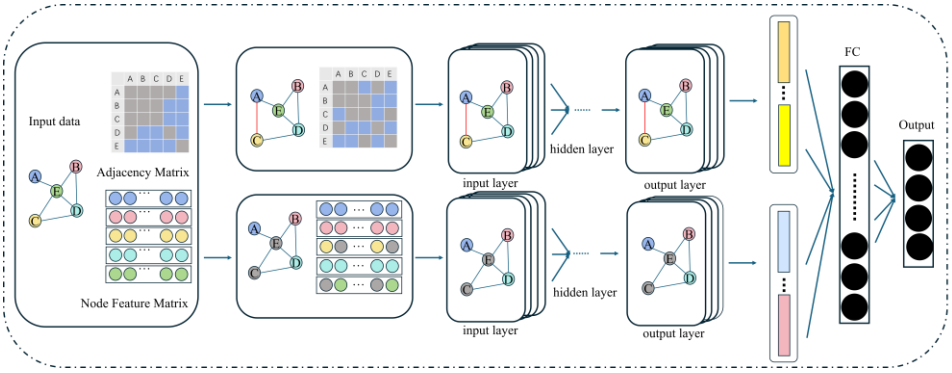


Figure 2: Overview of the DSGNN model. The input graph is processed in parallel through two separate channels focusing on structural and feature perturbations, respectively. Their outputs are then fused to produce a robust node representation. A dynamic fusion layer adaptively weighs the contributions of the two channels based on the reliability of structural and feature information.

4.3 Training and Testing Procedure

During training, perturbed graphs $\tilde{A} \in \mathbb{B}_A$ and $\tilde{X} \in \mathbb{B}_X$ are sampled at each epoch. Forward propagation is performed through both channels, and the outputs, along with the fusion weights, are optimized jointly.

- **Training:** At each epoch, perturbations $(\tilde{A}, \tilde{X})$ are sampled, and the channels are optimized jointly with the fusion mechanism.
- **Testing:** The dual-channel format is retained, where the model processes both $\tilde{A}$ and $\tilde{X}$ and then outputs the fused representation.

4.4 Representation Fusion

The outputs of the two channels are fused as:

$$H = \alpha \cdot H_{\text{struct}} + (1 - \alpha) \cdot H_{\text{feat}}, \quad \alpha \in [0, 1], \tag{5}$$

where α can be a fixed scalar, a learnable parameter, or derived from an attention mechanism. Alternatively, concatenation followed by a multi-layer perceptron (MLP) can be used for nonlinear fusion.

4.5 Loss Function

The fused representation H is passed into a classifier and optimized using the cross-entropy loss:

$$\mathcal{L} = \text{CrossEntropy}(\text{softmax}(H), Y). \tag{6}$$

Backpropagation jointly updates the two channels and the fusion weights. Additional terms, such as adversarial loss or edge regularization, can be incorporated to further enhance robustness.

4.6 Algorithm Description

Algorithm 1 DSGNN Training Procedure

Require: Adjacency matrix A, feature matrix X, label set Y, budgets ϵ_A, ϵ_X, epochs T
Ensure: Trained model parameters θ

```
for epoch = 1 to T do
    Sample perturbations Ã ∈ B_A, X̃ ∈ B_X
    H_struct ← GNN_struct(X, Ã)
    H_feat ← GNN_feat(X̃, A)
    H ← Fusion(H_struct, H_feat)
    L ← CrossEntropy(softmax(H), Y)
    Update θ to minimize L
end for
return θ
```

Algorithm 1 summarizes the DSGNN training procedure. In each epoch, independent perturbations are sampled for the structure and features, and the corresponding representations are obtained through parallel GNN channels. The fusion mechanism combines the outputs into a unified representation, which is optimized against ground-truth labels using cross-entropy loss. Model parameters, including the fusion weights if applicable, are updated through backpropagation to enhance robustness against both types of perturbation.

4.7 Robustness Bound Analysis

Let the fused outputs under clean and perturbed inputs be:

$$H = \alpha H_{\text{struct}} + (1-\alpha) H_{\text{feat}}, \quad \tilde{H} = \alpha H_{\text{struct}}(\tilde{A}) + (1-\alpha) H_{\text{feat}}(\tilde{X}). \tag{7}$$

The perturbation risk is defined as the output difference:

$$\mathsf{R}_{\epsilon}^{\text{DS}}[f] = \mathsf{E}_{(\tilde{A},\tilde{X})\in \mathsf{B}_A \times \mathsf{B}_X} \left[d_Y(f(H), f(\tilde{H})) \right]. \tag{8}$$

Applying convexity yields the following inequality:

$$d_Y(f(H), f(\tilde{H})) \le \alpha\, d_Y\left(f(H_{\text{struct}}), f(H_{\text{struct}}(\tilde{A}))\right) + (1-\alpha)\, d_Y\left(f(H_{\text{feat}}), f(H_{\text{feat}}(\tilde{X}))\right). \tag{9}$$

Assuming Lipschitz continuity of each submodel, there exist constants $C_1, C_2 > 0$ such that:

$$d_Y(f(H_{\text{struct}}), f(H_{\text{struct}}(\tilde{A}))) \le C_1 \epsilon_A^2, \quad d_Y(f(H_{\text{feat}}), f(H_{\text{feat}}(\tilde{X}))) \le C_2 \epsilon_X^2. \tag{10}$$

Thus, the robustness risk is bounded by:

$$\mathsf{R}_{\epsilon}^{\text{DS}}[f] \le \alpha C_1 \epsilon_A^2 + (1-\alpha) C_2 \epsilon_X^2. \tag{11}$$

4.8 Comparison with Single-Channel Baseline

For a standard GNN baseline that jointly consumes both perturbed inputs:

$$H_{\text{single}} = \text{GNN}(\tilde{X}, \tilde{A}), \tag{12}$$

the robustness risk can be bounded as:

$$\mathsf{R}_{\epsilon}^{\text{Single}}[f] \leq C \cdot (\epsilon_A + \epsilon_X)^2. \tag{13}$$

When $C_1, C_2 \leq C$ and the perturbations are not highly correlated, the perturbation risks satisfy $\mathsf{R}_{\epsilon}^{\text{DS}}[f] < \mathsf{R}_{\epsilon}^{\text{Single}}[f]$, demonstrating the tighter robustness guarantee and stronger defense capability of DSGNN.

Moreover, the decoupled architecture of DSGNN offers enhanced resilience against coordinated attacks, contributing to better generalization and stability under diverse perturbation scenarios.

4.9 Computational Complexity Analysis

DSGNN is designed to enhance perturbation robustness while maintaining computational efficiency. The architecture consists of two parallel standard two-layer GCNs and a lightweight attention-based fusion module. For a graph with N nodes, input feature dimension F, and edge set size $|E|$, the per-layer complexity of a GCN is $\mathsf{O}(NF^2 + |E|F)$. Therefore, the dual-channel backbone of DSGNN leads to an overall complexity of $\mathsf{O}(2L(NF^2 + |E|F))$, where L denotes the number of GCN layers. The fusion module, operating only on two output vectors per node, adds an additional cost of $\mathsf{O}(NH)$, which is negligible in comparison.

Table 3 summarizes the total time complexity of DSGNN and other baseline models, taking into account both architectural components and preprocessing overhead.

Table 3: Total time complexity comparison of baseline models

Model	Architecture Type	Total Time Complexity
GCN-Jaccard	GCN + Edge Filtering	$\mathsf{O}(N^2F + L(NF^2 + \|E\|F))$
GCN-SVD	GCN + Low-rank Approx.	$\mathsf{O}(N^3 + L(NF^2 + \|E\|F))$
RGCN	GCN + Label Regularization	$\mathsf{O}(L(NF^2 + \|E\|F) + NC)$
GNNGuard	GCN + Attention Masking	$\mathsf{O}(L(NF^2 + \|E\|F) + \|E\|H)$
NoisyGNN	GCN + Noise Injection	$\mathsf{O}(L(NF^2 + \|E\|F))$
DSGNN	Dual GCN + Fusion Module	$\mathsf{O}(2L(NF^2 + \|E\|F) + NH)$

As shown in the comparison, although DSGNN introduces a dual-channel architecture that increases the computational workload relative to a standard GCN, the overall time complexity only grows linearly and remains within a practical range. More importantly, DSGNN does not rely on computationally expensive preprocessing procedures such as node similarity computation or matrix decomposition, which significantly reduces implementation cost and deployment complexity. As a result, DSGNN achieves improved model performance while preserving computational efficiency and structural scalability, making it well-suited for real-world applications.

Notation: N = number of nodes, F = input feature dimension, H = hidden dimension, $|E|$ = number of edges, C = number of classes, L = number of GCN layers.

5 Experimental Parameter Settings

In all experiments, DSGNNs were instantiated based on a standard two-layer Graph Convolutional Network (GCN) architecture. Table 4 summarizes the key architectural and training parameters used throughout the evaluations.

Table 4: Experimental Parameter Settings

Parameter	Value
hidden_size	256
n_layers	2
epochs	200
learning_rate	5e-4

The selected parameter settings follow those commonly adopted in existing works, particularly aligning with the configuration used in NoisyGNN. This ensures consistency with prior research and enables fair and reproducible performance comparisons. Specifically, we adopt the same core settings as NoisyGNN, including a hidden size of 256, two GCN layers, 200 training epochs, and a learning rate of 5e-4.

6 Experimental Evaluation

To assess the robustness of the proposed DSGNN against structural perturbations, extensive experiments were conducted on representative benchmark datasets, covering both citation networks and a real-world industrial semantic graph. The evaluation focused on three key aspects: the model's stability under adversarial perturbations, its generalization ability to complex industrial data, and the theoretical certifiability of its robustness.

6.1 Datasets

The evaluation was conducted on the following datasets:

- **Cora**: A citation network with 2708 nodes and 7 classes, characterized by relatively dense connections.
- **CiteSeer**: A citation network with 3327 nodes and 6 classes, exhibiting a sparser graph structure than that of Cora.
- **Industry**: A semantic classification graph containing 5312 textual samples, categorized into four types: industrial equipment, process techniques, production materials, and others. The label distribution in this dataset is highly imbalanced.

Compared to standard benchmarks, the Industry dataset exhibits several real-world characteristics:

- Weak and sparse connectivity: Many nodes are loosely connected or isolated;
- Semantic heterogeneity: Significant textual variation within and across classes;
- High noise: Edges may reflect irrelevant or incomplete semantic relationships.

These conditions pose major challenges for ensuring GNN robustness in practical scenarios [20].

6.2 Attack Settings and Baselines

Structural perturbations are simulated using three representative attack strategies:

- **Metattack** [21]: A meta-optimization-based white-box structural attack;
- **PGD** [22]: A proximal gradient descent method for adversarial edge modification;
- **DICE** [21]: A random edge addition and deletion strategy that does not require gradient information.

Perturbation budgets were set to 0%, 5%, and 10%. All methods were trained under consistent settings to ensure fair comparisons.

Baselines included widely adopted robust GNN approaches: GCN-Jaccard, GCN-SVD, RGCN, GNNGuard, and NoisyGNN. The DSGNN extends a standard GCN by injecting feature and structure perturbations into separate, complementary channels.

Following established practices, three perturbation levels were considered to evaluate model robustness: (1) no attack (0%), (2) moderate perturbation (5%), and (3) severe perturbation (10%). These perturbations were applied to the graph structure using Metattack, PGD, and DICE, respectively.

Table 5: Classification accuracy (± standard deviation) of models under different perturbation rates ϵ.

Dataset	Attack	ϵ	GNNGuard	GCN-Jaccard	GCN-SVD	RGCN	NoisyGNN	DSGNN
Cora	Metattack	0%	**83.9 ± 0.8**	82.7 ± 1.0	77.9 ± 1.0	83.7 ± 0.8	83.7 ± 0.9	83.6 ± 0.9
		5%	78.8 ± 2.0	79.1 ± 1.5	73.4 ± 1.2	78.2 ± 2.1	79.5 ± 1.9	**80.7 ± 1.5**
		10%	74.5 ± 2.8	76.1 ± 2.1	69.4 ± 1.5	72.8 ± 2.9	75.5 ± 2.4	**77.9 ± 1.8**
	PGD	0%	81.3 ± 1.7	82.8 ± 0.9	78.0 ± 0.9	**83.7 ± 0.9**	83.6 ± 1.0	83.6 ± 0.9
		5%	81.3 ± 1.7	80.2 ± 1.7	77.0 ± 1.6	79.4 ± 1.0	**82.5 ± 1.3**	**82.5 ± 1.1**
		10%	79.7 ± 1.8	79.2 ± 1.4	75.7 ± 1.7	75.3 ± 1.9	81.6 ± 1.5	**82.0 ± 1.2**
	DICE	0%	**83.9 ± 0.9**	82.8 ± 0.9	78.0 ± 0.9	83.5 ± 0.9	83.6 ± 1.0	83.5 ± 1.0
		5%	**82.8 ± 1.1**	82.0 ± 0.9	76.3 ± 1.0	82.3 ± 0.8	82.3 ± 0.9	82.5 ± 1.3
		10%	**81.4 ± 0.9**	80.8 ± 1.0	73.4 ± 1.2	80.7 ± 1.1	81.1 ± 1.1	81.1 ± 1.1
CiteSeer	Metattack	0%	73.0 ± 1.2	73.1 ± 1.0	67.8 ± 1.2	71.5 ± 1.0	**73.6 ± 1.4**	73.1 ± 1.3
		5%	68.7 ± 2.4	69.9 ± 1.7	67.3 ± 1.2	68.8 ± 2.4	70.0 ± 2.3	**71.0 ± 1.7**
		10%	64.8 ± 3.3	66.8 ± 2.5	66.0 ± 1.3	64.4 ± 3.1	65.8 ± 3.1	**67.8 ± 3.1**
	PGD	0%	73.1 ± 1.4	73.1 ± 1.7	67.8 ± 1.2	71.3 ± 1.4	**73.3 ± 1.5**	73.1 ± 1.5
		5%	71.7 ± 2.8	71.6 ± 1.7	67.1 ± 1.7	70.3 ± 3.2	72.6 ± 1.0	**73.0 ± 1.2**
		10%	70.7 ± 1.9	71.0 ± 1.8	67.8 ± 1.7	69.3 ± 3.6	71.8 ± 1.6	**72.5 ± 1.5**
	DICE	0%	73.1 ± 1.4	73.1 ± 1.2	67.8 ± 1.3	70.7 ± 1.2	**73.3 ± 1.5**	**73.3 ± 1.4**
		5%	71.8 ± 1.3	72.0 ± 0.8	66.5 ± 1.7	68.9 ± 1.3	72.2 ± 1.2	**72.3 ± 1.6**
		10%	70.6 ± 1.1	**71.2 ± 1.1**	64.5 ± 1.1	67.0 ± 1.4	70.9 ± 1.5	71.1 ± 1.6
Industry	Metattack	0%	82.4 ± 0.6	82.2 ± 0.8	81.9 ± 0.6	81.5 ± 0.3	82.1 ± 0.6	**82.6 ± 0.6**
		5%	79.5 ± 1.8	80.8 ± 1.2	79.3 ± 1.9	80.6 ± 1.4	80.5 ± 0.8	**81.4 ± 0.7**
		10%	77.8 ± 2.2	79.3 ± 2.4	77.1 ± 3.6	77.9 ± 2.7	79.0 ± 1.7	**81.0 ± 0.8**
	PGD	0%	82.2 ± 0.6	82.2 ± 0.7	82.0 ± 0.9	81.5 ± 0.2	82.0 ± 0.7	**82.6 ± 0.8**
		5%	81.4 ± 0.4	81.5 ± 0.4	81.4 ± 0.7	81.5 ± 0.3	81.5 ± 0.3	**81.8 ± 0.5**
		10%	81.0 ± 0.7	81.4 ± 0.4	**81.6 ± 0.4**	81.2 ± 0.4	80.9 ± 0.6	**81.6 ± 0.7**
	DICE	0%	82.2 ± 0.6	82.2 ± 0.7	81.9 ± 0.8	81.5 ± 0.3	82.0 ± 0.7	**82.4 ± 1.0**
		5%	81.3 ± 0.6	81.4 ± 0.5	81.1 ± 0.6	81.1 ± 0.3	81.3 ± 0.5	**81.5 ± 0.6**
		10%	80.5 ± 0.9	**81.0 ± 0.6**	80.3 ± 1.0	80.8 ± 0.3	80.6 ± 0.4	80.9 ± 1.0

6.3 Robustness Analysis Classification

Model performance was evaluated under varying levels of structural perturbation, focusing on accuracy trends, fluctuation stability, and adaptability to industrial graph conditions.

6.3.1 Noise-Free Graph (0% Perturbation)

On noise-free graphs, DSGNN achieved comparable accuracy to baseline methods, indicating that the dual-channel design does not compromise predictive capacity. DSGNN attained 86.4% accuracy on the Industry dataset, demonstrating effective representation learning even under noisy and imbalanced structures.

6.3.2 Metattack and PGD

Metattack and PGD are optimization-based attacks that strategically disrupt graph connectivity. Many baselines experience severe accuracy degradation. In contrast, DSGNN showed slower performance decline and retained higher residual accuracy.

Under 10% PGD perturbation on Cora, DSGNN achieved the highest accuracy among all compared methods. On the Industry dataset, it maintained 81.6% accuracy under the same perturbation level, demonstrating its robustness in complex real-world graphs. The decoupled dual-channel architecture helps stabilize representation learning even when structural or feature information is partially corrupted.

6.3.3 DICE Attack

DICE introduces random structural noise without gradient-based targeting, reflecting real-world issues such as annotation errors or distribution shifts.

At 10% DICE perturbation, DSGNN achieved the highest accuracy on both CiteSeer (71.3%) and Industry (80.9%), with low variance across runs. Its performance on Cora was relatively lower. This difference likely arises from Cora's intrinsic structure: strong local clusters and tightly interconnected nodes within classes. Random edge removals fragment clusters, confusing decision boundaries. Furthermore, Cora's feature homogeneity amplifies vulnerability to collective perturbations.

In contrast, DSGNN remains robust on the Industry dataset, where sparse connections, noisy edges, and diverse features challenge most models. The dual-channel design enables stable representation learning even when parts of the information are degraded.

6.3.4 Average Accuracy Comparison by Dataset

To further assess the overall robustness of different defense strategies, the average classification accuracy across all attack scenarios for each dataset was computed. The results are summarized in Table 6.

Table 6: Average classification accuracy (%) across all attacks for each dataset.

Method	Cora	CiteSeer	Industry
GCN-Jaccard	79.39	70.66	81.00
GCN-SVD	73.01	66.03	80.36
RGCN	77.09	68.71	80.32
GNNGuard	80.86	70.73	81.34
NoisyGNN	80.97	71.55	81.16
DSGNN	**81.24**	**71.94**	**81.66**

As observed in Table 6, DSGNN consistently achieves the highest average accuracy across all datasets:

- On Cora, DSGNN outperforms all baselines, slightly surpassing GNNGuard and NoisyGNN.
- On CiteSeer, DSGNN demonstrates superior robustness under adversarial settings, yielding a 71.94% average accuracy.
- On the Industry dataset, characterized by weak structural connectivity and semantic noise, DSGNN achieves 81.66%, indicating its practical applicability.

These results show that the dual-channel architecture of DSGNN enhances model robustness across both benchmark and real-world graph datasets.

6.4 Certified Robustness Evaluation

To evaluate the certified robustness of DSGNN against structural perturbations, sparse randomized smoothing [23] was employed. This technique certifies whether a model's prediction remains stable when a bounded number of graph edges were modified. Certified accuracy was evaluated for DSGNN and the baseline GCN under perturbation radii $r \in \{0, 1, 2, \ldots, 10\}$.

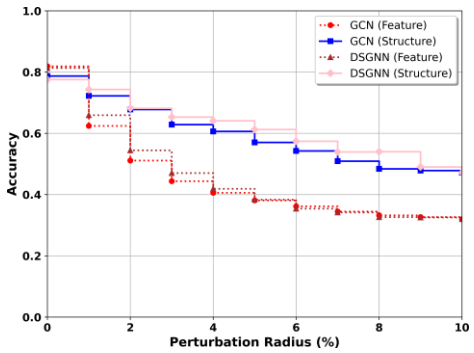
In this evaluation, two types of perturbations are considered: feature perturbations and structure perturbations. Certified robustness is established only under structure perturbations, following the standard sparse randomized smoothing protocol. Feature perturbation results are reported as complementary analysis to provide a more comprehensive view of the model's robustness.

The perturbation radius r denotes the cumulative perturbation steps, where each step introduces approximately 1% of feature noise injection or 1% of cumulative structural modifications. Feature perturbations are applied by randomly modifying node features with a probability proportional to r%, while structure perturbations are performed by progressively adding or removing edges.

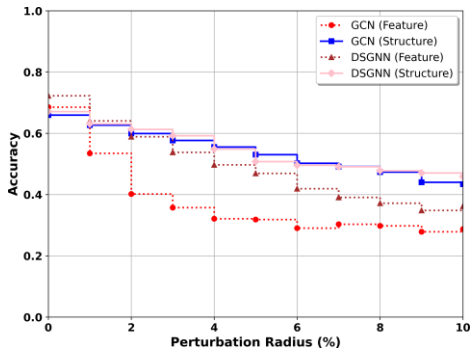
As shown in Fig. 3, the DSGNN consistently outperforms the GCN in certified accuracy across all three datasets. In Cora and Industry, the DSGNN demonstrates a significant robustness improvement over GCNs, especially under larger perturbation radii where the gap becomes increasingly noticeable. These results highlight the effectiveness of the dual-channel design in handling complex and noisy graph environments, such as the Industry dataset.

In contrast, on the CiteSeer dataset, the advantage of DSGNNs over GCNs is relatively smaller. This could be attributed to the inherently sparse and simpler structure of CiteSeer graphs, where structural perturbations have a smaller impact, and so the benefits of dual-channel processing are correspondingly reduced.

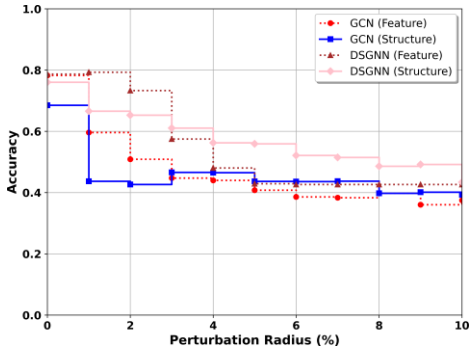
Overall, these results demonstrate that DSGNN enhances the model's certified robustness under structural perturbations and also exhibits improved stability under feature perturbations.



(a) Cora



(b) CiteSeer



(c) Industry

Figure 3: Certified accuracy of DSGNNs and GCNs under varying perturbation radii r. Structure perturbations are certified using sparse randomized smoothing, while feature perturbation results were presented for complementary analysis. Each step corresponds to approximately 1% of feature noise injection or 1% of cumulative structural modifications.

6.5 Ablation Study on Perturbation Settings

To further examine the contributions of individual components in DSGNN, we performed ablation experiments under three controlled perturbation settings:

- **Structure-only:** Only the graph structure was perturbed $(\tilde{A}, X)$, while the node features remained unchanged.
- **Feature-only:** Only the node features were perturbed $(A, \tilde{X})$, with the graph structure preserved.
- **Joint-fixed:** Both structure and features were perturbed $(\tilde{A}, \tilde{X})$, but the fusion weights were fixed to 0.5 for each channel rather than adaptively learned.

As shown in Table 7, DSGNN achieved competitive performance even in single-modality perturbation scenarios (structure-only and feature-only), which validated the independent robustness of each channel. Furthermore, the dual-channel variant with fixed fusion weights outperformed both single-channel baselines under joint perturbations, which indicated the benefit

Table 7: Ablation study on DSGNN under different perturbation settings: classification accuracy (± standard deviation) across attack types and ϵ levels.

Dataset	Attack	ϵ	Joint (0.5/0.5)	Structure-only	Feature-only	DSGNN
Cora	Metattack	0%	83.7 ± 1.0	83.5 ± 0.9	83.6 ± 0.9	**83.6 ± 0.9**
		5%	80.3 ± 1.5	79.7 ± 1.2	79.4 ± 1.9	**80.7 ± 1.5**
		10%	76.2 ± 2.1	76.4 ± 1.5	75.6 ± 2.4	**77.9 ± 1.8**
	PGD	0%	83.6 ± 0.9	83.6 ± 0.9	83.7 ± 1.0	**83.6 ± 0.9**
		5%	82.3 ± 1.3	82.0 ± 1.5	82.3 ± 1.2	**82.5 ± 1.1**
		10%	81.6 ± 1.3	81.2 ± 1.7	81.6 ± 1.4	**82.0 ± 1.2**
	DICE	0%	83.5 ± 0.8	83.3 ± 0.9	83.4 ± 1.0	**83.5 ± 1.0**
		5%	82.4 ± 0.8	82.0 ± 1.1	82.3 ± 0.9	**82.5 ± 1.3**
		10%	80.8 ± 1.0	80.5 ± 1.2	81.0 ± 1.2	**81.1 ± 1.1**
Citeseer	Metattack	0%	73.1 ± 1.0	72.7 ± 1.2	73.7 ± 1.3	**73.1 ± 1.3**
		5%	70.7 ± 1.7	71.0 ± 1.2	70.1 ± 2.2	**71.0 ± 1.7**
		10%	67.8 ± 2.5	66.9 ± 1.9	66.7 ± 2.0	**67.8 ± 3.1**
	PGD	0%	73.1 ± 1.8	72.5 ± 1.2	73.0 ± 1.5	**73.1 ± 1.5**
		5%	72.8 ± 1.3	73.0 ± 1.7	72.5 ± 1.0	**73.0 ± 1.2**
		10%	72.0 ± 1.5	72.3 ± 1.6	71.6 ± 1.7	**72.5 ± 1.5**
	DICE	0%	73.1 ± 1.3	73.0 ± 1.3	73.2 ± 1.5	**73.3 ± 1.4**
		5%	72.3 ± 1.4	71.9 ± 1.7	72.2 ± 1.3	**72.3 ± 1.6**
		10%	71.0 ± 1.1	70.3 ± 1.2	70.9 ± 1.4	**71.1 ± 1.6**
Industry	Metattack	0%	82.2 ± 0.8	81.9 ± 1.1	82.0 ± 0.6	**82.6 ± 0.6**
		5%	80.8 ± 1.3	80.3 ± 1.0	80.7 ± 0.8	**81.4 ± 0.7**
		10%	80.5 ± 1.4	80.1 ± 0.9	79.0 ± 1.7	**81.0 ± 0.8**
	PGD	0%	82.2 ± 0.8	82.0 ± 1.0	82.1 ± 0.7	**82.6 ± 0.8**
		5%	81.7 ± 0.4	81.4 ± 0.7	81.4 ± 0.3	**81.8 ± 0.5**
		10%	81.4 ± 0.4	80.0 ± 0.4	80.9 ± 0.7	**81.6 ± 0.7**
	DICE	0%	82.2 ± 0.7	81.9 ± 0.8	82.0 ± 0.7	**82.4 ± 1.0**
		5%	81.3 ± 0.7	81.1 ± 0.7	81.3 ± 0.6	**81.5 ± 0.6**
		10%	80.7 ± 0.6	80.3 ± 1.0	80.5 ± 0.4	**80.9 ± 1.0**

of integrating both modalities. Most importantly, the full version of DSGNN, which employed a learnable dynamic fusion mechanism, consistently achieved the highest accuracy across all datasets and attack settings. This demonstrated the essential role of adaptive fusion in handling heterogeneous and complex perturbation patterns.

6.6 Summary

Across both accuracy and certification evaluations, DSGNN consistently outperforms existing defenses on benchmark and real-world graphs. Its dual-channel structure improves stability, reduces performance degradation, and adapts well to weakly structured, semantically noisy industrial scenarios.

7 Conclusion and Future Work

DSGNN is proposed as a defense model that decouples structural and feature perturbations through a dual-channel architecture. By isolating the propagation paths of different noise types

and dynamically integrating their outputs, DSGNN significantly improves robustness against a wide range of adversarial attacks.

Experimental evaluations show that DSGNN consistently outperforms baseline defense methods across multiple benchmark datasets, maintaining superior stability under both targeted and random perturbations. These results underscore the effectiveness of explicitly modeling perturbation decoupling at the architectural level.

Future work includes extending DSGNN to other graph neural network architectures, such as GAT and GraphSAGE. Additionally, the development of adaptive fusion mechanisms, where the importance weights between channels are dynamically adjusted based on the characteristics of observed perturbations, represents a promising direction.

Acknowledgments

The authors are grateful to the anonymous reviewers and the editor for their insightful comments and suggestions that substantially enhanced the clarity and rigor of this work.

Funding Statement

This research was funded by the Key Research and Development Program of Zhejiang Province No.2023C01141, and the Science and Technology Innovation Community Project of the Yangtze River Delta No.23002410100.

This work was supported by the Open Research Fund of the State Key Laboratory of Blockchain and Data Security, Zhejiang University.

Author Contributions

Xiaohan Chen: Software implementation, Experimental evaluation, Writing—original draft preparation; Yuanfang Chen: Conceptualization, Methodology design, Project administration, Supervision, Writing—review and editing, Funding acquisition; Gyu Myoung Lee, Noel Crespi, Pierluigi Siano: Methodological guidance, Technical support, Manuscript review and constructive suggestions. All authors have read and approved the final version of the manuscript.

Availability of Data and Materials

This work utilizes the publicly available datasets Cora, CiteSeer, and Industry for model training and evaluation. These datasets are freely accessible through established academic repositories.

Ethics Approval

Not applicable.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding this study.

References

1. Wang H, Fu T, Du Y, Gao W, Huang K, Liu Z, et al. Scientific discovery in the age of artificial intelligence. Nature. 2023;620(7972):47-60. Available from: https://doi.org/10.1038/s41586-023-06221-2.
2. Jin M, Koh HY, Wen Q, Zambon D, Alippi C, Webb GI, et al. A Survey on Graph Neural Networks for Time Series: Forecasting, Classification, Imputation, and Anomaly Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2024;46(12):10466-85. Available from: https://doi.org/10.1109/TPAMI.2024.3443141.
3. Chen Z, Mao H, Li H, Jin W, Wen H, Wei X, et al. Exploring the potential of large language models (llms) in learning on graphs. ACM SIGKDD Explorations Newsletter. 2024;25(2):42-61. Available from: https://doi.org/10.1145/3655103.3655110.
4. Fang X, Chen Y, Bhuiyan ZA, He X, Bian G, Crespi N, et al. Mixer-transformer: Adaptive anomaly detection with multivariate time series. Journal of Network and Computer Applications. 2025;241:104216. Available from: https://www.sciencedirect.com/science/article/pii/S1084804525001134.
5. Liu J, Guo M. DIGNN-A: Real-Time Network Intrusion Detection with Integrated Neural Networks Based on Dynamic Graph. Computers, Materials & Continua. 2025;82(1):817-42. Available from: https://doi.org/10.32604/cmc.2024.057660.
6. Chen Y, Fang X, Ma S, Li W. TrackSecurity: An Attention-Guided Physical Patch Model for Perturbing Visual Trackers in Autonomous Driving. TechRxiv. 2025. Available from: https://www.techrxiv.org/doi/full/10.36227/techrxiv.175021860.09534391.
7. Chen Y, Yang H, Zhang Y, KAILI M, Liu T, Han B, et al. Understanding and Improving Graph Injection Attack by Promoting Unnoticeability. In: International Conference on Learning Representations; 2022. Available from: https://openreview.net/forum?id=wkMG8cdvh7-.
8. Liu J, Chen B, Xue B, Guo M, Xu Y. PIAFGNN: Property Inference Attacks against Federated Graph Neural Networks. Computers, Materials & Continua. 2025;82(2):1857-77. Available from: https://doi.org/10.32604/cmc.2024.057814.
9. Entezari N, Al-Sayouri SA, Darvishzadeh A, Papalexakis EE. All you need is low (rank) defending against adversarial attacks on graphs. In: Proceedings of the 13th international conference on web search and data mining; 2020. p. 169-77. Available from: https://doi.org/10.1145/3336191.3371789.
10. Zhang X, Zitnik M. GNNGuard: Defending Graph Neural Networks against Adversarial Attacks. 2020;33:9263-75. Available from: https://proceedings.neurips.cc/paper_files/paper/2020/file/690d83983a63aa1818423fd6edd3bfdb-Paper.pdf.
11. Wu H, Wang C, Tyshetskiy Y, Docherty A, Lu K, Zhu L. Adversarial examples for graph data: deep insights into attack and defense. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence; 2019. p. 4816-23. Available from: https://doi.org/10.48550/arXiv.1903.01610.
12. Feng F, He X, Tang J, Chua TS. Graph adversarial training: Dynamically regularizing based on graph structure. IEEE Transactions on Knowledge and Data Engineering. 2019;33(6):2493-504. Available from: https://doi.org/10.1109/TKDE.2019.2957786.
13. Zhu D, Zhang Z, Cui P, Zhu W. Robust graph convolutional networks against adversarial attacks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2019. p. 1399-407. Available from: https://doi.org/10.1145/3292500.3330851.
14. ABBAHADDOU Y, ENNADIR S, Lutzeyer JF, Vazirgiannis M, Boström H. Bounding the Expected Robustness of Graph Neural Networks Subject to Node Feature Attacks. In: The Twelfth International Conference on Learning Representations; 2024. Available from: https://openreview.net/forum?id=DfPtC8uSot.
15. Aslan HI, Wiesner P, Xiong P, Kao O. β-GNN: A Robust Ensemble Approach Against Graph Structure Perturbation. In: Proceedings of the 5th Workshop on Machine Learning and Systems; 2025. p. 168-75. Available from: https://doi.org/10.1145/3721146.3721949.
16. Ennadir S, Abbahaddou Y, Lutzeyer JF, Vazirgiannis M, Boström H. A simple and yet fairly effective defense for graph neural networks. In: Proceedings of the Thirty-Eighth AAAI Conference on Artificial

Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence; 2024. p. 21063-71. Available from: https://doi.org/10.1609/aaai.v38i19.30098.

17. Jin W, Ma Y, Liu X, Tang X, Wang S, Tang J. Graph structure learning for robust graph neural networks. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining; 2020. p. 66-74. Available from: https://doi.org/10.1145/3394486.3403049.
18. Schlichtkrull M, Kipf TN, Bloem P, Van Den Berg R, Titov I, Welling M. Modeling relational data with graph convolutional networks. In: The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15. Springer; 2018. p. 593-607. Available from: https://doi.org/10.1007/978-3-319-93417-4_38.
19. Dai H, Li H, Tian T, Huang X, Wang L, Zhu J, et al. Adversarial Attack on Graph Structured Data. In: Dy J, Krause A, editors. Proceedings of the 35th International Conference on Machine Learning. vol. 80 of Proceedings of Machine Learning Research. PMLR; 2018. p. 1115-24. Available from: https://proceedings.mlr.press/v80/dai18b.html.
20. Lu H, Wang L, Ma X, Cheng J, Zhou M. A survey of graph neural networks and their industrial applications. Neurocomputing. 2025;614:128761. Available from: https://www.sciencedirect.com/science/article/pii/S0925231224015327.
21. Zügner D, Günnemann S. Adversarial Attacks on Graph Neural Networks via Meta Learning. In: ICLR Workshop on Safe Machine Learning (SafeML); 2019. Available from: https://openreview.net/forum?id=Bylnx209YX.
22. Xu K, Chen H, Liu S, Chen PY, Weng TW, Hong M, et al. Topology attack and defense for graph neural networks: an optimization perspective. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. IJCAI'19. AAAI Press; 2019. p. 3961-7. Available from: https://doi.org/10.1145/3292500.333090.
23. Jin H, Shi Z, Peruri VJSA, Zhang X. Certified Robustness of Graph Convolution Networks for Graph Classification under Topological Attacks. 2020;33:8463-74. Available from: https://proceedings.neurips.cc/paper_files/paper/2020/file/609a199881ca4ba9c95688235cd6ac5c-Paper.pdf.