Adversarial Training for Graph Convolutional Networks: Stability and Generalization Analysis

Chang Cao[1] , **Han Li**[1,2*] , **Yulong Wang**[1,2] , **Rui Wu**[3] and **Hong Chen**[1,2]
[1]College of Informatics, Huazhong Agricultural University, Wuhan 430070, China
[2]Engineering Research Center of Intelligent Technology for Agriculture, Ministry of Education, Wuhan 430070, China
[3]Horizon Robotics, Haidian District, Beijing 100190, China
lihan125@mail.hzau.edu.cn,

Abstract

Recently, numerous methods have been proposed to enhance the robustness of the Graph Convolutional Networks (GCNs) for their vulnerability against adversarial attacks. Despite their empirical success, a significant gap remains in understanding GCNs' adversarial robustness from the theoretical perspective. This paper addresses this gap by analyzing generalization against both node and structure attacks for multi-layer GCNs through the framework of uniform stability. Under the smoothness assumption of the loss function, we establish the first adversarial generalization bound of GCNs in expectation. Our theoretical analysis contributes to a deeper understanding of how adversarial perturbations and graph architectures influence generalization performance, which provides meaningful insights for designing robust models. Experimental results on benchmark datasets confirm the validity of our theoretical findings, highlighting their practical significance.

1 Introduction

GCNs have demonstrated superior ability to process graph-structured data and learn both node and graph representations [Kipf and Welling, 2017; Zhu *et al.*, 2021]. As one of the most widely employed tasks of GCNs, node classification has garnered significant attention due to its extensive real-world applications, such as natural language processing [Liu and Wu, 2022; Wang *et al.*, 2024], recommendation systems [Sun *et al.*, 2021; Gao *et al.*, 2023], and computer vision [Zhou *et al.*, 2021; Hu *et al.*, 2023]. However, GCNs are vulnerable to some small but intentional perturbations on node features or graph structures that can mislead the classifiers to make erroneous predictions [Zhu *et al.*, 2019; Bojchevski and Günnemann, 2019].

Several approaches have been proposed to enhance the robustness of models against adversarial attacks [Sun *et al.*, 2022]. Among these, adversarial training is formulated as a min-max problem. The training procedure targets minimizing the classification error against an adversary who perturbs the input data and maximizes the classification error [Huang *et al.*, 2015]. Numerous studies of adversarial training on GCNs have shown excellent performance in node classification tasks [Xue *et al.*, 2021a; Xu *et al.*, 2020; Deng *et al.*, 2023]. However, a significant problem was observed in adversarial training, i.e., robust overfitting [Yin *et al.*, 2019; Rice *et al.*, 2020]. The network performs well on the training data but has poor adversarial generalization ability on the test set. This motivates the need for a generalization guarantee from the theoretical perspective.

Previous studies investigate adversarial generalization using different statistical analytical techniques, including covering number [Tu *et al.*, 2019; Mustafa *et al.*, 2022], Rademacher complexity [Yin *et al.*, 2019; Gao and Wang, 2021], and stability analysis [Farnia and Ozdaglar, 2021; Xing *et al.*, 2021a]. Unfortunately, the aforementioned works are confined to adversarial generalization analysis based on non-graph data. As the adversarial attacks to graph-structured data could modify both node features and graph structure, which poses more challenges in adversarial generalization analysis than [Gao and Wang, 2021; Mustafa *et al.*, 2022; Xiao *et al.*, 2022]. Specifically, in GCNs, node-based attacks affect the entire feature matrix rather than individual samples, due to the message-passing mechanism inherent to GCNs. This interaction between nodes creates an intricate interplay of perturbations, complicating the analysis of the stability of GCNs under such attacks. Additionally, structure-based attacks modify the graph's edges by altering the adjacency matrix (i.e., flipping discrete connection signals 0 or 1), making standard techniques for adversarial loss inapplicable to graph-structured data.

To overcome these difficulties, we derive the approximate smoothness properties for the adversarial loss function in GCNs and investigate the adversarial generalization of GCNs through the lens of uniform stability. Our analysis is applicable to both node-based and structure-based attacks. The main contributions are listed as follows:

- We establish a stability-based adversarial generalization bound in expectation for general GCNs to provide theoretical support for adversarial training. Our theoretical results demonstrate that the configuration of the graph model architecture and optimal algorithm significantly impacts GCNs' generalization performance.

*Corresponding author.

Model	Reference	Under Attack	Analysis Tool	Stability Bound
Single layer GCN	Verma and Zhang (2019)	No	Uniform stability	$\mathcal{O}\big(\frac{(\lambda_G^{max})^{2T}}{n}\big)$
Multi-layer GCNs	Zhou and Wang (2021)	No	Uniform stability	$\mathcal{O}\big(\frac{(S_1+\cdots+S_K)^T}{n}\big)$
Neural networks	Farnia and Ozdaglar (2021)	Yes	Uniform stability	$\mathcal{O}(T\eta/n)$
Neural networks	Xing *et al.* (2021a)	Yes	Uniform argument stability	$\mathcal{O}(\sqrt{T}\eta + \frac{T\eta}{n})$
Neural networks	Xiao *et al.* (2022)	Yes	Uniform stability	$\mathcal{O}(\epsilon T\eta + \frac{T\eta}{n})$
Multi-layer GCNs	Ours	Yes	Uniform stability	$\mathcal{O}\big((\epsilon + \frac{1}{n})\eta^{T-1}\big)$

Table 1: Summary of stability bound for different models (n-the number of samples; T-the number of iterations; λ_G^{max}-the maximum absolute eigenvalue of the graph filters; $(S_1, \ldots, S_K)$-a set of parameters related to model layers K, which consists of the norm of the graph filters and some Lipschitz constants; η-learning rate of SGD; ϵ-perturbation budget).

- We address the impact of the information interaction between perturbed nodes on the adversarial generalization by utilizing the contraction technique of graph convolution, which is based on graph filters. We exploit the properties of graph filters to avoid operating directly in the discrete perturbation domain under structure-based attacks.
- Our experimental results on benchmark datasets provide evidence that the established theoretical findings facilitates improving the robust generalization of GCNs.

2 Related Work

Adversarial attacks for node classification. For node-based attacks, Takahashi *et al.* (2019) impose adversarial attacks by searching small perturbations on a single node, which leads to misclassification into the node far more than one-hop from the perturbed node. Ma *et al.* (2020) investigate black-box attacks under realistic constraints, where attackers have access to only a subset of nodes and can only target a limited number of nodes. Finkelshtein *et al.* (2022) propose a single-node indirect gradient adversarial evasion attack, focusing on the more realistic scenario where a single attacker node is involved. For structure-based attacks, Xu *et al.* (2019) propose a gradient-based attack method that perturbs a small number of edges, which leads to a noticeable decrease in classification performance. Geisler *et al.* (2021) develop two sparsity-aware first-order optimization attack methods to attack graph models at scale. Fan *et al.* (2023) introduce a novel attack framework to jointly attack graph models and their explanations via inserting adversarial edges.

Adversarial training on GCNs. For node-based attacks, Feng *et al.* (2019) introduce Graph Adversarial Training (GraphAT), which generates perturbations by maximizing the divergence between the predictions of two connected nodes. Xue *et al.* (2021a) analyze GCNs by examining weight and feature loss landscapes, and apply alternating adversarial training (Co-AT) to mitigate the risk of sharp local minima. Dend *et al.* (2023) design Batch Virtual Adversarial Training (BVAT), which promotes output smoothness of GCNs by applying virtual adversarial perturbations to independent subsets of nodes or all nodes. For structure-based attacks, Xu *et al.* (2020) develop Zeroth-Order Greedy Topology Attack (ZO-GTA), a gradient-free adversarial training method aimed at obtaining robust models in a more generic manner. Li *et al.* (2022) construct adversarial perturbations in the spectral domains and design Spectral Adversarial Training (SAT), which is applicable to both node-based and structure-based attacks. Gosch *et al.* (2024) improve the robustness of GNNs against structural perturbations by leveraging a learnable graph diffusion model to approximate any graph filter. We summarize the above works in Appendix C. Although the empirical effectiveness of the above adversarial training methods has been validated, their theoretical foundations have not been investigated.

Stability-based generalization analysis. Uniform stability is a classical approach for measuring the generalization gap [Bartlett and Mendelson, 2002]. Verma and Zhang (2019) first apply uniform stability on one-layer GCN to derive a generalization bound. Their work demonstrates a high relationship between generalization and the maximum absolute eigenvalue of the graph filters. Zhou and Wang (2021) extend their results to multi-layer GCNs and show a exponential dimensional dependence of the generalization gap on the number of layers. Additionally, uniform stability has a wide range of applications in adversarial learning. Farnia and Ozdaglar (2021) analyze the adversarial generalization properties of minimax models through the lens of algorithmic stability. Similarly, Xing *et al.* (2021a) develop an adaptive algorithm and apply uniform argument stability to solve the min-max problem. It is important to note that the stability bounds in these studies are ϵ-independent, where ϵ is the adversarial perturbation. Xiao *et al.* (2022) address this limitation by introducing a modified smoothness assumption, enabling the derivation of adversarial generalization bounds that depend on ϵ. We summarize the main results of studies using uniform stability in Table 1. Though widely used in neural networks, the stability-based adversarial generalization has been unexplored in GCNs.

3 Problem Setting

Notations. We denote a matrix by the boldface uppercase letters (e.g., $\mathbf{X}$) and a vector by the boldface lowercase letters (e.g., $\mathbf{x}$). For a matrix $\mathbf{X}$, we denote its $\|\cdot\|_\infty$-norm by $\|\mathbf{X}\|_\infty = \|\mathbf{X}\| = \max_i \sum_j |X_{ij}|$, where X_{ij} is an element

in matrix $\mathbf{X}$. For a vector $\mathbf{x}$, we denote its $\|\cdot\|_2$-norm by $\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}$, and its $\|\cdot\|_\infty$-norm by $\|\mathbf{x}\|_\infty = \|\mathbf{x}\| = \max_i |x_i|$, where x_i is an element in vector $\mathbf{x}$.

A graph is represented as $G = \{\mathbf{A}, \mathbf{X}\}$ with N nodes, where $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the adjacency matrix and $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N \in \mathbb{R}^{N\times d}$ is the node feature matrix. Each node is an instance $\mathbf{z}_i = (\mathbf{x}_i, y_i)$ containing feature $\mathbf{x}_i$ and label y_i from space $\mathcal{Z} = \mathcal{X}\times\mathcal{Y}$. Let $n \le N$ be the number of training samples $\mathbf{z}_i$. In this paper, we focus on the node classification task, which is based on a typical K-layer GCN model. The update rule with $j = \{1, 2, \ldots, K\}$ is represented by

$$\mathbf{X}_j = \sigma(g(\mathbf{A})\mathbf{X}_{j-1}\mathbf{W}_j),$$

where $\mathbf{X}_j \in \mathbb{R}^{N\times d_j}$ represents the output of the j-th layer with $\mathbf{X}_0 \in \mathbb{R}^{N\times d}$ and $\mathbf{X}_K \in \mathbb{R}^{N\times|\mathcal{Y}|}$ denoting the input and output matrix of the model, respectively. Similarly, $\mathbf{W}_j \in \mathbb{R}^{d_{j-1}\times d_j}$ represents the learning parameters of the j-th layer with $d_0 = d$ and $d_K = |\mathcal{Y}|$. Moreover, the function $\sigma(\cdot)$ denotes the activation function, and $g(\mathbf{A}) \in \mathbb{R}^{N\times N}$ represents the graph filter, which is a function of the adjacency matrix $\mathbf{A}$, such as the unnormalized filter $g(\mathbf{A}) = \mathbf{A} + \mathbf{I}$, the symmetric normalized filter $g(\mathbf{A}) = (\mathbf{D}+\mathbf{I})^{-1/2}(\mathbf{A}+\mathbf{I})(\mathbf{D}+\mathbf{I})^{-1/2}$ [Kipf and Welling, 2017], and the random walk normalized filter $g(\mathbf{A}) = (\mathbf{D}+\mathbf{I})^{-1}(\mathbf{A}+\mathbf{I})$ [Zhang *et al.*, 2019]. For a node $\mathbf{x}_i$, we utilize an indicator $\mathbf{p}_{\mathbf{x}_i} \in \mathbb{R}^N$ to obtain the corresponding output label $\mathbf{y}_i$. Thus, the output function of the node $\mathbf{x}$ is defined as

$$f(\mathbf{W}, \mathbf{A}, \mathbf{x}) = \sigma(\mathbf{p}_{\mathbf{x}}^T g(\mathbf{A})\mathbf{X}_{K-1}\mathbf{W}_K),$$

where $\mathbf{W} = \{\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_K\}$ is the learning parameters. For a K-layer GCNs, the learning algorithm is defined as $\mathcal{A}^S = f(\mathbf{W}, \mathbf{A}, \mathbf{x})$, where the training set $S = \{\mathbf{z}_i\}_{i=1}^n$ is sampled from an unknown distribution D. Denote the loss function on sample $\mathbf{z} = (\mathbf{x}, y)$ by $\ell(\mathcal{A}^S, \mathbf{z})$ or $\ell(f(\mathbf{W}, \mathbf{A}, \mathbf{x}), y)$, the population risk is defined on distribution D as $\mathcal{R}_D(\mathcal{A}^S) = \mathbb{E}_D\ell(f(\mathbf{W}, \mathbf{A}, \mathbf{x}), y)$, and the empirical risk is defined on the training set S as $\mathcal{R}_S(\mathcal{A}^S) = \frac{1}{n}\sum_{i=1}^n \ell(f(\mathbf{W}, \mathbf{A}, \mathbf{x}_i), y_i)$.

In this paper, we focus on the ℓ_∞ adversarial attack. The adversary is allowed to attack the graph with some ℓ_∞ balls by perturbing the node feature matrix or adjacency matrix. We impose the attacks by maximizing the standard loss

$$\tilde{\ell}(f(\mathbf{W}, \mathbf{A}, \mathbf{x}), y) = \max_{\tilde{f}(\cdot)} \ell(\tilde{f}(\mathbf{W}, \mathbf{A}, \mathbf{x}), y), \qquad (1)$$

where $\tilde{f}(\cdot)$ denotes the perturbed graph model by node or structure attacks. Then, to improve the robustness of GCNs against adversarial attacks, we perform adversarial training by minimizing the adversarial loss $\tilde{\ell}(f(\mathbf{W}, \mathbf{A}, \mathbf{x}), y)$ [Shaham *et al.*, 2018].

To better explain the generalization under the framework of adversarial training, we define the adversarial population risk function as

$$\widetilde{\mathcal{R}}_D(\mathcal{A}^S) = \mathbb{E}_D\tilde{\ell}(f(\mathbf{W}, \mathbf{A}, \mathbf{x}), y).$$

It represents the worst-case risk of the model under adversarial perturbations. Since the distribution D is unknown, we estimate the min-max objective by using the adversarial empirical risk function defined as

$$\widetilde{\mathcal{R}}_S(\mathcal{A}^S) = \frac{1}{n}\sum_{i=1}^n \tilde{\ell}(f(\mathbf{W}, \mathbf{A}, \mathbf{x}_i), y_i).$$

We are interested in the difference between the population risk and the empirical risk in adversarial settings as the generalization error, which is represented by

$$G(\mathcal{A}^S) = \left|\widetilde{\mathcal{R}}_D(\mathcal{A}^S) - \widetilde{\mathcal{R}}_S(\mathcal{A}^S)\right|.$$

4 Main Result

The main results are presented in Theorem 2 and 3, which provide the adversarial generalization gap for K-layer GCNs. We first give the definition of uniform stability. Next, we give some essential assumptions and critical lemmas. Finally, we establish the stability bound of GCNs in adversarial settings.

4.1 Uniform Stability

For a dataset S, we consider replacing the i^{th} data point $\mathbf{z}_i$ with a new point $\mathbf{z}_i'$. The resulting dataset can be represented as

$$S^i = \{\mathbf{z}_1, \ldots, \mathbf{z}_{i-1}, \mathbf{z}_i', \mathbf{z}_{i+1}, \ldots, \mathbf{z}_n\}.$$

As shown by Bousquet and Elisseeff (2002), the generalization gap is closely related to uniform stability. Specifically, Definition 1 provides an upper bound on the difference in loss resulting from a small change in the training set, applicable to all possible sample combinations $\mathbf{z}$.

Definition 1 (Uniform Stability)**.** *A randomized learning algorithm $\mathcal{A}^S$ trained on dataset S with n samples is β-uniformly stable with respect to a loss function ℓ, if it satisfies*

$$\sup_{\mathbf{z}\in S}\left|\mathbb{E}_{\mathcal{A}}\big[\ell(\mathcal{A}^S, \mathbf{z})\big] - \mathbb{E}_{\mathcal{A}}\big[\ell(\mathcal{A}^{S^i}, \mathbf{z})\big]\right| \le \beta.$$

According to Hardt, Recht, and Singer (2016), a randomized learning algorithm with the above uniform stability can yield the following bound on the generalization gap.

Theorem 1 (Generalization in Expectation)**.** *A uniform stable randomized algorithm $(\mathcal{A}^S, \beta)$ satisfies the following generalization bound*

$$\left|\mathbb{E}_{S,\mathcal{A}}\big[\mathcal{R}_D(\mathcal{A}^S) - \mathcal{R}_S(\mathcal{A}^S)\big]\right| \le \beta.$$

Theorem 1 demonstrates that if a randomized algorithm is uniformly stable, then its generalization error is small. For the randomized algorithm, we use SGD on the adversarial loss to search for the optimal solution for the worst-case scenario. At each iteration t, for $1 \le j \le K$, the update rule is defined as:

$$\mathbf{W}_{j,t+1} = \mathbf{W}_{j,t} - \eta_t\nabla_{\mathbf{W}_j}\tilde{\ell}\big(\mathcal{A}_t^S, \mathbf{z}_{i_t}\big) \qquad (2)$$

where $\eta_t > 0$ is the step size in iteration t, and $\mathbf{W}_{j,t}$ are the parameters generated by the j-th layer GCNs running on SGD at t-iteration. We take the expectation over the risks in Theorem 1 based on the internal randomness of $\mathcal{A}^S$. Note that SGD generates randomness as it selects examples randomly to compute batch loss gradients. Hence, we focus on this randomness and ignore the randomness introduced by parameter initialization. We will use $\mathbb{E}_{\text{SGD}}$ instead of $\mathbb{E}_{\mathcal{A}}$ in subsequent analysis.

4.2 Assumptions

Next, we give some assumptions of the loss function $\ell(\cdot)$ and activation function $\sigma(\cdot)$.

Assumption 1. *Assume that the loss function $\ell(\mathbf{W}, \mathbf{V})$ satisfies the following Lipschitzian smoothness conditions, where $\mathbf{W}$ represent the weight of GCNs, and $\mathbf{V} = \mathbf{z}$ or $\mathbf{V} = \mathbf{A}$.*

(a) $\big|\ell(\mathbf{W}, \mathbf{z}) - \ell(\mathbf{W}', \mathbf{z})\big| \leq v_w \|\mathbf{W} - \mathbf{W}'\|$.

(b) $\big\|\nabla\ell(\mathbf{W}, \mathbf{z}) - \nabla\ell(\mathbf{W}', \mathbf{z})\big\| \leq \alpha_w \|\mathbf{W} - \mathbf{W}'\|$.

(c) $\big\|\nabla\ell(\mathbf{W}, \mathbf{z}) - \nabla\ell(\mathbf{W}, \mathbf{z}')\big\| \leq \alpha_x \|\mathbf{z} - \mathbf{z}'\|$.

(d) $\big\|\nabla\ell(\mathbf{W}, \mathbf{A}) - \nabla\ell(\mathbf{W}, \mathbf{A}')\big\| \leq \alpha_g \|g(\mathbf{A}) - g(\mathbf{A}')\|$.

Assumption 2. *Assume that the activation function $\sigma(\cdot)$ satisfies the following Lipschitzian smoothness conditions.*

(a) $\|\sigma(\mathbf{x}) - \sigma(\mathbf{y})\| \leq v_\sigma \|\mathbf{x} - \mathbf{y}\|$.

(b) $\big\|\nabla\sigma(\mathbf{x}) - \nabla\sigma(\mathbf{y})\big\| \leq \alpha_\sigma \|\mathbf{x} - \mathbf{y}\|$.

Assumption 3. *Assume that there exists a constant C_x such that $\|\boldsymbol{x}\|_2 \leq C_x$ holds for all $\boldsymbol{x} \in \mathcal{X}$.*

Assumption 4. *Assume that there exists a constant $C_w > 0$ such that $\max_{1 \leq j \leq K} \|\mathbf{W}_j\| \leq C_w$ holds.*

Remarks. Assumption 1 is commonly found in the stability literature and holds for most gradient-based attacks [Farnia and Ozdaglar, 2021; Xing *et al.*, 2021b]. Some common loss functions satisfy the above conditions, such as the cross-entropy loss and log loss. And a majority of activation functions like Sigmoid, Tanh, and ELU satisfy the Assumption 2. Assumption 3 can be satisfied by applying normalization to the input features [Verma and Zhang, 2019; Tang and Liu, 2023]. Assumption 4 reflects the common requirement in generalization analysis of GCNs that parameters are bounded during the training process [Garg *et al.*, 2020; Cong *et al.*, 2021].

4.3 Stability Generalization Bounds

We then utilize uniform stability to analyze the generalization of multi-layer GCNs under two adversarial scenarios: node-based and structure-based attacks.

Node Perturbations

We first impose adversarial attacks on node features $\mathbf{x}$ to find the most effective adversarial examples $\tilde{\mathbf{x}}$. They are generated from a noise set $\mathcal{B}^{\infty}_{\epsilon_x} = \{\tilde{\mathbf{x}} : \|\tilde{\mathbf{x}} - \mathbf{x}\| \leq \epsilon_x\}$, where ϵ_x represents the perturbation budget, typically set to small values to ensure that the feature distribution of adversarial examples remains close to clean examples. The adversarial loss in formulation (1) can be rewritten by

$$\tilde{\ell}(f(\mathbf{W}, \mathbf{A}, \mathbf{x}), y) = \max_{\tilde{\mathbf{x}} \in \mathcal{B}^{\infty}_{\epsilon_x}} \ell(f(\mathbf{W}, \mathbf{A}, \tilde{\mathbf{x}}), y).$$

Then, we have the following properties of the adversarial loss function against the node perturbations.

Lemma 1. *Let $\mathbf{W}$ represent the training parameters of SGD running on GCNs, and the adversarial loss function $\tilde{\ell}(\mathbf{W}, \mathbf{z})$ satisfies Assumption 1. For any $\mathbf{z} \in \mathcal{Z}$, the following properties hold.*

(a) $\tilde{\ell}(\mathbf{W}, \mathbf{z})$ is v_w-Lipschitz continuous
$\big|\tilde{\ell}(\mathbf{W}, \mathbf{z}) - \tilde{\ell}(\mathbf{W}', \mathbf{z})\big| \leq v_w \|\mathbf{W} - \mathbf{W}'\|$.

(b) $\tilde{\ell}(\mathbf{W}, \mathbf{z})$ is α_w-Lipschitz $2\alpha_x\epsilon_x$-approximately smooth
$\big\|\nabla_{\mathbf{W}}\tilde{\ell}(\mathbf{W}, \mathbf{z}) - \nabla_{\mathbf{W}}\tilde{\ell}(\mathbf{W}', \mathbf{z})\big\| \leq \alpha_w \|\mathbf{W} - \mathbf{W}'\| + 2\alpha_x\epsilon_x$.

Remarks. In Lemma 1, the first inequality shows that the adversarial loss also satisfies the Lipschitz continuous properties. Nevertheless, the second item demonstrates that the maximization operation on loss hurts the continuity of its gradient, and the smoothness of the adversarial loss is controlled by the perturbation constraint ϵ_x.

Then we state the adversarial generalization bound for K-layer GCNs, which controls the generalization error against the node attacks by measuring the stability of the SGD algorithm.

Theorem 2 (Adversarial Generalization Gap of Multi-layer GCNs)**.** *Let $\mathcal{A}^S$ be the function learned by K-layer GCNs after training on the dataset S with n samples using the SGD algorithm for T iterations. With Assumptions 1, 2, 3 and 4 hold, we have the following expected generalization bound.*

$$\mathbb{E}_{SGD}\big[G(\mathcal{A}^S)\big] \leq \mathcal{O}\Big(\big(\eta K C_G\big)^T \big(\alpha_x\epsilon_x + \frac{1}{n} K C_G\big)\Big),$$

where $\eta = \max\{\eta_1, \eta_2, \ldots, \eta_T\}$ and

$$C_G = \max_{j \in [1,K]} \{\alpha_w {v_\sigma}^{2(j-1) + \frac{(K+j-1)(K-j)}{2}} {\alpha_\sigma}^{K-j+1} {C_x}^K \|g(\mathbf{A})\|^{\frac{K^2+3K-2}{2}} {C_w}^{2(j-1)+\frac{(K+j+3)(K-j)}{2}}\}.$$

Remarks. For K-layer GCNs, due to the term C_G, the generalization bound would increase exponentially with the number of layers K, which deteriorates the performance of deep GCNs. Theorem 2 reveals that C_G is affected by some Lipschitz coefficients (e.g. v_σ and α_σ) and norm constraints (e.g. C_x and C_w). Thus, we can choose ELU activation and cross-entropy loss to eliminate the effect of those Lipschitz constants. And the norm constraint C_x of node $\mathbf{x}$ can be controlled by applying the batch-normalization in GCNs [Verma and Zhang, 2019]. As $\|g(\mathbf{A})\| = 2d_{max} + 1$ if $g(\mathbf{A}) = \mathbf{A} + \mathbf{I}$ is selected, we can choose the normalized filter with $\|g(\mathbf{A})\| = (2d_{max} + 1)/d_{min}$, where d_{max} and d_{min} represent the maximum and minimum degree of the adjacency matrix, respectively. Besides, one could mitigate the exponential growth by applying a regularization on the weight norm $\|\mathbf{W}\|$ to reduce the value of C_w, especially when the number of layers increases.

Remarks. In the context of the SGD training procedure, the choice of learning rate η is crucial. A larger number of iterations T may result in a wider gap, potentially signaling overfitting. More importantly, compared to Zhou and Wang (2021), our bound takes account of the adversarial robustness, which is reflected in the term $\alpha_x\epsilon_x$. The adversarial coefficient α_x hurts the robust generalization even though the perturbation ϵ_x is small for their multiplicative relation.

Structural Perturbations

We now consider the structure attacks on GCNs, i.e., perform subtle perturbations on the graph structure by adding

or deleting a small number of edges. Followed by previous studies [Geisler *et al.*, 2021; Fan *et al.*, 2023], we assume that the adversarial adjacency matrix $\widetilde{\mathbf{A}}$ is generated by $\mathcal{B}^{\infty}_{\epsilon_A} = \{\widetilde{\mathbf{A}} : \|\widetilde{\mathbf{A}} - \mathbf{A}\| \leq \epsilon_A\}$, where ϵ_A is the perturbation budget to constrain the changes to $\mathbf{A}$. The adversarial loss in formulation (1) can be rewritten by

$$\tilde{\ell}(f(\mathbf{W}, \mathbf{A}, \mathbf{x}), y) = \max_{\widetilde{\mathbf{A}} \in \mathcal{B}^{\infty}_{\epsilon_A}} \ell(f(\mathbf{W}, \widetilde{\mathbf{A}}, \mathbf{x}), y).$$

Different from the node attacks, the relation between the perturbed adjacency matrix $\widetilde{A}$ and adversarial generalization is difficult to handle. We explore the impact of $\widetilde{A}$ on the generalization by operating the graph filter, which is reflected on $\epsilon_g = \left\|g(\mathbf{A}) - g(\widetilde{\mathbf{A}})\right\|$. Similar to Lemma 1, the structure attacks also satisfy the Lipschitz properties of adversarial loss that $\tilde{\ell}(\mathbf{W}, g(\mathbf{A}))$ is v_w-Lipschitz continuous and α_w-Lipschitz $2\alpha_g\epsilon_g$-approximately smooth. Based on these properties, we can derive the generalization bound of K-layer GCNs against the structure attacks in the following theorem.

Theorem 3 (Adversarial Generalization Gap of Multi-layer GCNs)**.** *Let $\mathcal{A}^S$ be the function learned by K-layer GCNs after training on the dataset S with n samples using the SGD algorithm for T iterations. With Assumptions 1, 2, 3 and 4 hold,*

(a) if $g(A)$ is unnormalized, we have

$$\mathbb{E}_{SGD}\big[G(\mathcal{A}^S)\big] \leq \mathcal{O}\Big(\big(\eta K C_G\big)^T \big(\alpha_g \epsilon_A + \frac{1}{n} K C_G\big)\Big),$$

(b) if $g(A)$ is normalized, we have

$$\mathbb{E}_{SGD}\big[G(\mathcal{A}^S)\big] \leq \mathcal{O}\Big(\big(\eta K C_G\big)^T \big(\frac{2\alpha_g \epsilon_A}{d_{min}+1} + \frac{1}{n} K C_G\big)\Big),$$

where η is as stated in Theorem 2 and

$$C_G = \max_{j \in [1,K]} \{\alpha_w v_\sigma{}^{2(j-1)+\frac{(K+j-1)(K-j)}{2}} \alpha_\sigma{}^{K-j+1} C_x{}^K \|g(\widetilde{\mathbf{A}})\|^{\frac{K^2+3K-2}{2}} C_w{}^{2(j-1)+\frac{(K+j+3)(K-j)}{2}}\}.$$

Remarks. Obviously, the adversarial generalization of GCNs under the structure attack is also susceptible to the model structure and optimization algorithm. And our bound can cover the standard training setting when the perturbation budget $\epsilon_A = 0$. We find that the structural perturbations cause more damage to the generalization due to the perturbation budget ϵ_A. However, $\epsilon_A \ll N$ is set as a sufficiently small budget to reduce the unavoidable damage to robust generalization, especially when the number of nodes is large, resulting in $\epsilon_A/N \to 0$. Furthermore, the relationship between ϵ_g and ϵ_A is relevant to the choice of graph filter. It's noteworthy that $g(\widetilde{A})$ shares the same norm constraint as $g(\mathbf{A})$. Following the suggestion of Theorem 2, we choose the normalized filter, and can mitigate the impact of $\epsilon_g \leq \frac{2}{d_{min}}\epsilon_A$ by attacking a denser graph.

5 Experiments

According to the theoretical results, we observe that the adversarial generalization bound is related to the graph filters, the number of layers, the iterations of SGD, etc. In this section, we perform experiments on the node classification task to evaluate the effect of these factors on adversarial generalization. We provide the designed training objective and training procedure in Algorithm 1 and 2, respectively. We consider node attacks there, and the experiments of structure attacks are presented in Appendix D.

Experimental Setup. We adopt several widely-used benchmark datasets, including Cora, Citeseer, Pubmed, DBLP, CS, and CoraFull [Yang *et al.*, 2016; Bojchevski and Günnemann, 2017; Xue *et al.*, 2021b]. An overview is given in Table 2. We adopt two-layer GCNs with 16-unit hidden layers and symmetric normalized filters. The adversarial training is conducted with the ℓ_∞-PGD algorithm, which is attacked with perturbation budget ϵ_x. We choose the cross-entropy loss and the SGD algorithm for training, where the learning rate η is set as 0.1 with a momentum of 0.9. The regularization coefficient λ is fixed to 0.01. The adversarial generalization gap is approximately calculated by

$$|\text{adversarial\_train\_accuracy} - \text{adversarial\_test\_accuracy}|,$$

which is the absolute difference between the train accuracy and test accuracy of adversarial training. Each experiment is independently repeated 10 times to get the average value and the standard deviation. More experimental configurations and results, including the structure attacks and other attack algorithms, are provided in Appendix D.

Algorithm 1 Train a robust graph model under node attacks

Input: Graph G, dataset S, perturbation budget ϵ_x, regularization parameter λ, initialization $\mathbf{W}_0$, learning rate η, number of iterations T.
while $t < T$ **do**
  $\widetilde{S} \leftarrow \emptyset$.
  for $i = 1, 2, \ldots, n$ **do**
    Perturb each node and get $\tilde{\mathbf{x}}_{i,t} = \arg\max_{\tilde{\mathbf{x}}_{i,t} \in \mathcal{B}^{\infty}_{\epsilon_x}} \ell(f(\mathbf{W}, \mathbf{A}, \tilde{\mathbf{x}}_{i,t}), y_{i,t})$.
    Append $\{(\tilde{\mathbf{x}}_{i,t}, y_{i,t})\}_{i=1}^n$ to $\widetilde{S}_t$.
  end for
  Update $\mathbf{W}_t$ through a new objective $L(\mathbf{W}_t) = \frac{1}{n}\sum_{\mathbf{x}_{i,t} \in \widetilde{S}_t} \ell(f(\mathbf{W}_t, \mathbf{A}, \tilde{\mathbf{x}}_{i,t}), y_{i,t}) + \lambda\|\mathbf{W}_t\|$.
  $\mathbf{W}_{t+1} \leftarrow \mathbf{W}_t - \eta \nabla L(\mathbf{W}_t)$.
end while
Output: $\mathbf{W}_T$

Graph filters. According to Theorem 2, our theoretical analysis suggests that a bigger $\|g(\mathbf{A})\|$ leads to a greater generalization gap. Figure 1 shows that the unnormalized filter $g(\mathbf{A}) = \mathbf{A} + \mathbf{I}$ with the biggest infinity norm has the largest generalization gap, compared to the other normalized filters. The empirical results align with our theoretical findings. And the comparison between two normalized filters depends on the specific graph data structure.

Number of layers. As shown in Figure 2, the generalization gap generally tends to increase with the increase of K. According to Theorem 2, K controls a variety of the main factors influencing the generalization gap. Our experimental configurations, including applying the normalized graph

Dataset	Node	Edges	Features	Class	Training	Validation	Test
Citeseer	3327	9104	3703	6	20 per class	500	1000
Cora	2708	10556	1433	77	20 per class	500	1000
Pubmed	19717	88648	500	3	20 per class	500	1000
DBLP	17716	105734	1639	4	20 per class	30 per class	Rest
CS	18333	163788	6805	15	20 per class	30 per class	Rest
CoraFull	19793	126842	8710	70	20 per class	30 per class	Rest

Table 2: Details of datasets.

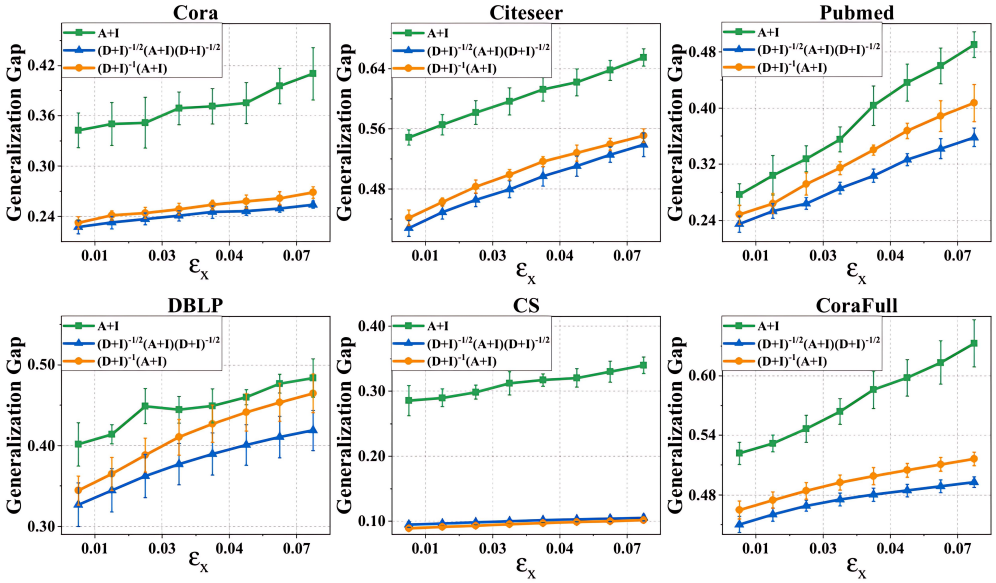

Figure 1: The generalization gap for different graph filters $g(\mathbf{A})$ with increased perturbation budget ϵ_x.

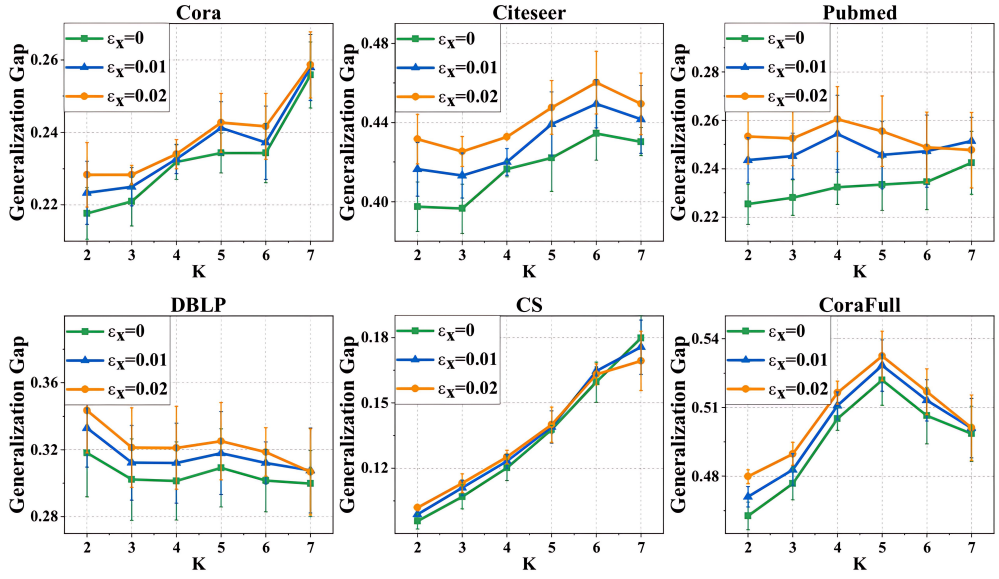

Figure 2: The generalization gap for different perturbation budget ϵ_x with increased number of layers K.

filters and regularization operations, could help reduce the inevitable increasing trend. The effectiveness of these measures is also demonstrated in Figure 1 and 4.

Number of iterations. Figure 3 illustrates that the general-

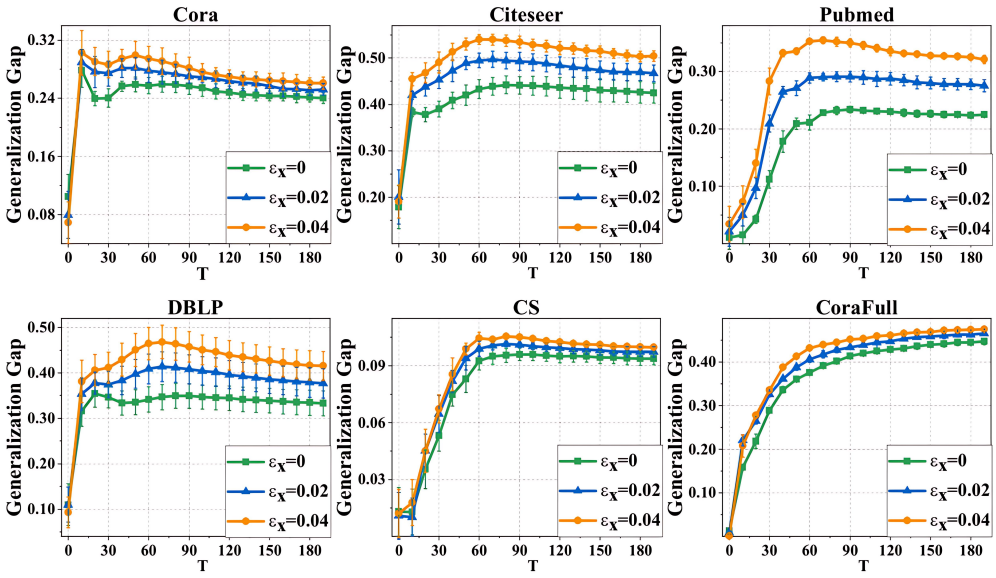

Figure 3: The generalization gap for different perturbation budget ϵ_x with increased number of iterations T.

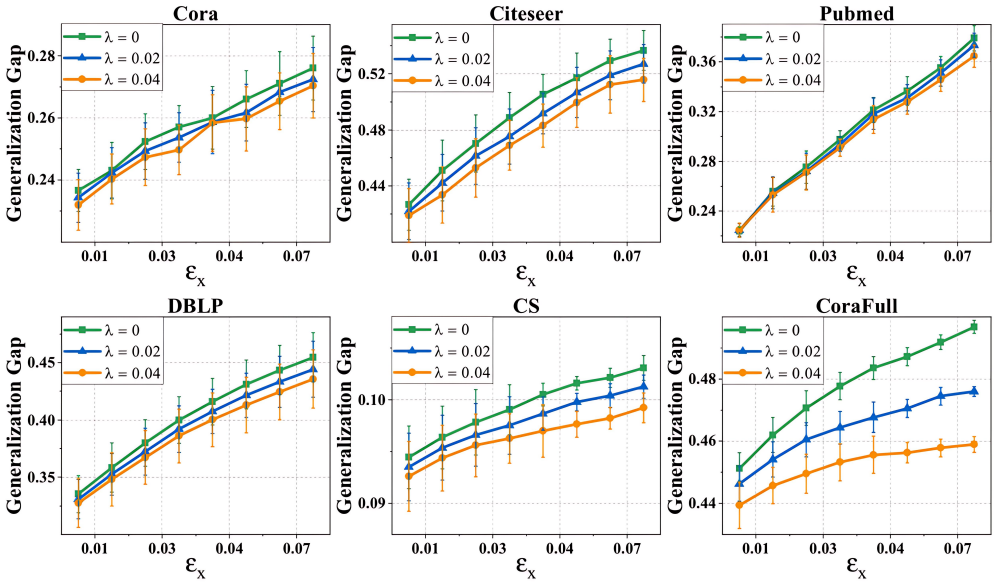

Figure 4: The generalization gap for different regularization parameter λ with increased perturbation budget ϵ_x.

ization gap enlarges as the iterations proceed from the perspective of an overall trend, which is consistent with the results in Theorem 2. After several iterations, the generalization gap tends to be stable as the model converges.

Regularization. Theorem 2 demonstrates that controlling the norm constraint of weights by regularization could help eliminate the damage of exponential growth of generalization. As shown in Figure 4, the regularized model has a better generalization than the model without regularization.

6 Conclusions

In this paper, we propose a theoretical understanding of adversarial robustness for GCNs under node-based attacks and structure-based attacks. More specifically, we first show that the adversarial loss satisfies the approximate smoothness, which is dependent on the perturbation budget. Then we derive the stability-based generalization bounds of multi-layer GCNs for adversarial training. Our results shed new insights into the choice of the graph filters and the number of GCNs' layers. A smaller norm of the graph filters and their product with input features can benefit the performance of the models. And a controllable norm constraint of learning weights by applying the regularization can obtain a better generalization. Our experimental results on benchmark datasets support the theoretical results.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (under Grant Nos. 62276111, 62076041, 62376104, 12426512), and the Open Research Fund of Engineering Research Center of Intelligent Technology for Agriculture, Ministry of Education (No. ERCITA-KF002).

References

[Bartlett and Mendelson, 2002] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[Bojchevski and Günnemann, 2017] Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. *arXiv preprint arXiv:1707.03815*, 2017.

[Bojchevski and Günnemann, 2019] Aleksandar Bojchevski and Stephan Günnemann. Certifiable robustness to graph perturbations. *Advances in Neural Information Processing Systems*, 32, 2019.

[Bousquet and Elisseeff, 2002] Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

[Cong *et al.*, 2021] Weilin Cong, Morteza Ramezani, and Mehrdad Mahdavi. On provable benefits of depth in training graph convolutional networks. *Advances in Neural Information Processing Systems*, 34:9936–9949, 2021.

[Deng *et al.*, 2023] Zhijie Deng, Yinpeng Dong, and Jun Zhu. Batch virtual adversarial training for graph convolutional networks. *AI Open*, 4:73–79, 2023.

[Fan *et al.*, 2023] Wenqi Fan, Han Xu, Wei Jin, Xiaorui Liu, Xianfeng Tang, Suhang Wang, Qing Li, Jiliang Tang, Jianping Wang, and Charu Aggarwal. Jointly attacking graph neural network and its explanations. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 654–667. IEEE, 2023.

[Farnia and Ozdaglar, 2021] Farzan Farnia and Asuman Ozdaglar. Train simultaneously, generalize better: Stability of gradient-based minimax learners. In *International Conference on Machine Learning*, pages 3174–3185. PMLR, 2021.

[Feng *et al.*, 2019] Fuli Feng, Xiangnan He, Jie Tang, and Tat-Seng Chua. Graph adversarial training: Dynamically regularizing based on graph structure. *IEEE Transactions on Knowledge and Data Engineering*, 33(6):2493–2504, 2019.

[Finkelshtein *et al.*, 2022] Ben Finkelshtein, Chaim Baskin, Evgenii Zheltonozhskii, and Uri Alon. Single-node attacks for fooling graph neural networks. *Neurocomputing*, 513:1–12, 2022.

[Gao and Wang, 2021] Qingyi Gao and Xiao Wang. Theoretical investigation of generalization bounds for adversarial learning of deep neural networks. *Journal of Statistical Theory and Practice*, 15, 2021.

[Gao *et al.*, 2023] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems*, 1(1):1–51, 2023.

[Garg *et al.*, 2020] Vikas Garg, Stefanie Jegelka, and Tommi Jaakkola. Generalization and representational limits of graph neural networks. In *International Conference on Machine Learning*, pages 3419–3430. PMLR, 2020.

[Geisler *et al.*, 2021] Simon Geisler, Tobias Schmidt, Hakan Şirin, Daniel Zügner, Aleksandar Bojchevski, and Stephan Günnemann. Robustness of graph neural networks at scale. *Advances in Neural Information Processing Systems*, 34:7637–7649, 2021.

[Gosch *et al.*, 2024] Lukas Gosch, Simon Geisler, Daniel Sturm, Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. Adversarial training for graph neural networks: Pitfalls, solutions, and new directions. *Advances in Neural Information Processing Systems*, 36, 2024.

[Hardt *et al.*, 2016] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pages 1225–1234. PMLR, 2016.

[Hu *et al.*, 2023] Xinxin Hu, Haotian Chen, Hongchang Chen, Shuxin Liu, Xing Li, Shibo Zhang, Yahui Wang, and Xiangyang Xue. Cost-sensitive gnn-based imbalanced learning for mobile social network fraud detection. *IEEE Transactions on Computational Social Systems*, 2023.

[Huang *et al.*, 2015] Ruitong Huang, Bing Xu, Dale Schuurmans, and Csaba Szepesvári. Learning with a strong adversary. *arXiv preprint arXiv:1511.03034*, 2015.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[Li *et al.*, 2022] Jintang Li, Jiaying Peng, Liang Chen, Zibin Zheng, Tingting Liang, and Qing Ling. Spectral adversarial training for robust graph neural network. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[Liu and Wu, 2022] Bang Liu and Lingfei Wu. Graph neural networks in natural language processing. *Graph Neural Networks: Foundations, Frontiers, and Applications*, pages 463–481, 2022.

[Ma *et al.*, 2020] Jiaqi Ma, Shuangrui Ding, and Qiaozhu Mei. Towards more practical adversarial attacks on graph neural networks. *Advances in neural information processing systems*, 33:4756–4766, 2020.

[Mustafa *et al.*, 2022] Waleed Mustafa, Yunwen Lei, and Marius Kloft. On the generalization analysis of adversarial learning. In *International Conference on Machine Learning*, pages 16174–16196. PMLR, 2022.

[Rice *et al.*, 2020] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International conference on machine learning*, pages 8093–8104. PMLR, 2020.

[Shaham *et al.*, 2018] Uri Shaham, Yutaro Yamada, and Sahand Negahban. Understanding adversarial training: Increasing local stability of supervised models through robust optimization. *Neurocomputing*, 307:195–204, 2018.

[Sun *et al.*, 2021] Weifeng Sun, Kangkang Chang, Lijun Zhang, and Kelong Meng. Ingcf: an improved recommendation algorithm based on ngcf. In *International Conference on Algorithms and Architectures for Parallel Processing*, pages 116–129. Springer, 2021.

[Sun *et al.*, 2022] Lichao Sun, Yingtong Dou, Carl Yang, Kai Zhang, Ji Wang, Philip S. Yu, Lifang He, and Bo Li. Adversarial attack and defense on graph data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, page 1–20, 2022.

[Takahashi, 2019] Tsubasa Takahashi. Indirect adversarial attacks via poisoning neighbors for graph convolutional networks. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1395–1400. IEEE, 2019.

[Tang and Liu, 2023] Huayi Tang and Yong Liu. Towards understanding generalization of graph neural networks. In *International Conference on Machine Learning*, pages 33674–33719. PMLR, 2023.

[Tu *et al.*, 2019] Zhuozhuo Tu, Jingwei Zhang, and Dacheng Tao. Theoretical analysis of adversarial learning: A minimax approach. *Advances in Neural Information Processing Systems*, 32, 2019.

[Verma and Zhang, 2019] Saurabh Verma and Zhi-Li Zhang. Stability and generalization of graph convolutional neural networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1539–1548, 2019.

[Wang *et al.*, 2024] Kunze Wang, Yihao Ding, and Soyeon Caren Han. Graph neural networks for text classification: A survey. *Artificial Intelligence Review*, 57(8):190, 2024.

[Xiao *et al.*, 2022] Jiancong Xiao, Yanbo Fan, Ruoyu Sun, Jue Wang, and Zhi-Quan Luo. Stability analysis and generalization bounds of adversarial training. *Advances in Neural Information Processing Systems*, 35:15446–15459, 2022.

[Xing *et al.*, 2021a] Yue Xing, Qifan Song, and Guang Cheng. On the algorithmic stability of adversarial training. *Advances in neural information processing systems*, 34:26523–26535, 2021.

[Xing *et al.*, 2021b] Yue Xing, Qifan Song, and Guang Cheng. On the generalization properties of adversarial training. In *International Conference on Artificial Intelligence and Statistics*, pages 505–513. PMLR, 2021.

[Xu *et al.*, 2019] Kaidi Xu, Hongge Chen, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Mingyi Hong, and Xue Lin. Topology attack and defense for graph neural networks: An optimization perspective. *arXiv preprint arXiv:1906.04214*, 2019.

[Xu *et al.*, 2020] Kaidi Xu, Sijia Liu, Pin-Yu Chen, Mengshu Sun, Caiwen Ding, Bhavya Kailkhura, and Xue Lin. Towards an efficient and general framework of robust training for graph neural networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8479–8483. IEEE, 2020.

[Xue *et al.*, 2021a] Haotian Xue, Kaixiong Zhou, Tianlong Chen, Kai Guo, Xia Hu, Yi Chang, and Xin Wang. Cap: Co-adversarial perturbation on weights and features for improving generalization of graph neural networks. *arXiv preprint arXiv:2110.14855*, 2021.

[Xue *et al.*, 2021b] Yifan Xue, Yixuan Liao, Xiaoxin Chen, and Jingwei Zhao. Node augmentation methods for graph neural network based object classification. In *2021 2nd International Conference on Computing and Data Science (CDS)*, pages 556–561. IEEE, 2021.

[Yang *et al.*, 2016] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pages 40–48. PMLR, 2016.

[Yin *et al.*, 2019] Dong Yin, Ramchandran Kannan, and Peter Bartlett. Rademacher complexity for adversarially robust generalization. In *International conference on machine learning*, pages 7085–7094. PMLR, 2019.

[Zhang *et al.*, 2019] Zhihong Zhang, Dongdong Chen, Jianjia Wang, Lu Bai, and Edwin R Hancock. Quantum-based subgraph convolutional neural networks. *Pattern Recognition*, 88:38–49, 2019.

[Zhou and Wang, 2021] Xianchen Zhou and Hongxia Wang. The generalization error of graph convolutional networks may enlarge with more layers. *Neurocomputing*, 424:97–106, 2021.

[Zhou *et al.*, 2021] Guangyuan Zhou, Huiqun Wang, Jiaxin Chen, and Di Huang. Pr-gcn: A deep graph convolutional network with point refinement for 6d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2793–2802, 2021.

[Zhu *et al.*, 2019] Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1399–1407, 2019.

[Zhu *et al.*, 2021] Yanqiao Zhu, Weizhi Xu, Jinghao Zhang, Yuanqi Du, Jieyu Zhang, Qiang Liu, Carl Yang, and Shu Wu. A survey on graph structure learning: Progress and opportunities. *arXiv preprint arXiv:2103.03036*, 2021.