A Simple and Yet Fairly Effective Defense for Graph Neural Networks

Sofiane Ennadir[1], Yassine Abbahaddou[2], Johannes F. Lutzeyer[2],
Michalis Vazirgiannis[1, 2], Henrik Boström[1]
[1]EECS, KTH Royal Institute of Technology, Stockholm, Sweden
[2]DaSciM, LIX, Ecole Polytechnique, Institut Polytechnique de Paris, France
{ennadir, mvaz, bostromh} @ kth.se, {yassine.abbahaddou, johannes.lutzeyer} @ polytechnique.edu.

Abstract

Graph Neural Networks (GNNs) have emerged as the dominant approach for machine learning on graph-structured data. However, concerns have arisen regarding the vulnerability of GNNs to small adversarial perturbations. Existing defense methods against such perturbations suffer from high time complexity and can negatively impact the model's performance on clean graphs. To address these challenges, this paper introduces NoisyGNNs, a novel defense method that incorporates noise into the underlying model's architecture. We establish a theoretical connection between noise injection and the enhancement of GNN robustness, highlighting the effectiveness of our approach. We further conduct extensive empirical evaluations on the node classification task to validate our theoretical findings, focusing on two popular GNNs: the GCN and GIN. The results demonstrate that NoisyGNN achieves superior or comparable defense performance to existing methods while minimizing added time complexity. The NoisyGNN approach is model-agnostic, allowing it to be integrated with different GNN architectures. Successful combinations of our NoisyGNN approach with existing defense techniques demonstrate even further improved adversarial defense results. Our code is publicly available at: https://github.com/Sennadir/NoisyGNN.

Introduction

Graphs have garnered substantial recognition in recent years as a powerful approach to represent intricate and irregular data, drawing significant attention across various fields. Their applications span diverse domains, including modeling social networks to discern connections between individuals and representing atom interactions in molecules. Consequently, the proliferation of graph-based applications has necessitated the development of machine learning algorithms tailored for graph processing. One notable and powerful technique is the Graph Neural Network (GNN), which has emerged as a valuable tool for learning node and graph representations. GNNs often belong to the family of Message Passing Neural Networks (MPNNs) (Gilmer et al. 2017), such as the Graph Isomorphism Networks (GIN) (Xu et al. 2019b) and the Graph Convolutional Networks (GCN) (Kipf and Welling 2017). These GNNs have demonstrated remarkable success in tackling a wide range of real-world problems. For instance, within the field of chemistry, significant attention has been dedicated to employing deep learning systems based on graphs for tasks like drug screening and design, where molecules are effectively represented as graphs (Kearnes et al. 2016). GNNs have also shown efficacy in predicting protein functions (You et al. 2021) and other biomedical proprieties such as antibiotic resistance (Qabel et al. 2022) since proteins can be effectively modeled as graphs. In a different context, GNNs have found utility in session-based recommendation systems (Wu et al. 2019b).

Given the growing popularity of these techniques, there arises a compelling necessity for an in-depth examination of their robustness to ensure their safe usage in critical sectors such as healthcare. Recent work (Dai et al. 2018; Zügner, Akbarnejad, and Günnemann 2018; Günnemann 2022) have indeed detected the vulnerability of Graph Neural Networks (GNNs) to adversarial attacks, which are injected intentional input alterations that can manipulate the model's predictions through slight structural modifications or node feature-based perturbations. To mitigate these effects and bolster the robustness of Message Passing Neural Networks (MPNNs), various studies have proposed diverse techniques. These approaches encompass a spectrum of strategies, including augmenting training data with adversarial examples and subsequent model retraining (Feng et al. 2021), using edge pruning techniques (Zhang and Zitnik 2020), and the introduction of robustness certificates (Schuchardt et al. 2021).

While some of these defense methods have exhibited success in countering adversarial perturbations, they often entail a high level of complexity due to their underlying architecture. The time complexity of these methods tends to increase heavily with the graph's size, consequently limiting their practical applicability in certain settings. Furthermore, a significant drawback of many existing approaches is their requirement for extensive architectural modifications, posing challenges for integration into different models. Moreover, some available defense methods can have detrimental effects on the model's performance when applied to clean, i.e., not attacked, graphs. This is of particular importance since in practice, the users do not have prior knowledge of whether their graph datasets have been attacked.

Recently, there has been a notable surge of interest in leveraging adversarial weight perturbation as a means to enhance the generalization capabilities of GNNs (Wu, Bojchevski, and Huang 2023). While existing research has primarily focused on demonstrating its efficacy in improving accuracy, our study considers a new perspective by investigating its potential application in the field of adversarial defense. In particular, we explore a defense strategy, called NoisyGNN, which leverages randomization by introducing random noise in the hidden states of certain layers of the GNN during the training phase. We start by adapting the mathematical formulation to investigate the robustness of GNNs against graph adversarial attacks. We afterward analyze the impact of randomization in enhancing the robustness of GNNs. Our theoretical analysis establishes a connection between noise injection in the architecture and strengthening a GNN's robustness. Finally, we empirically evaluate the proposed perturbation defense for its effectiveness against various adversarial attack methods in comparison to other available defense approaches on commonly used real-world benchmark datasets. While our theoretical and experimental analyses primarily focus on two well-known GNN architectures, namely GCN and GIN, our approach is model-agnostic and can be easily applied to different network architectures. We can summarize our contribution in the following points:

- We provide a mathematical formalization of graph adversarial attacks on GNNs to connect the noise injection's effect to robustness.
- We derive an upper bound, based on a theoretical analysis, that demonstrates the link between randomization and robustness and hence proves the effectiveness of our proposed framework, NoisyGNN, in enhancing the robustness of GCN and GIN based classifiers.
- We conduct extensive evaluations of our theoretical findings on the node classification task using various benchmark datasets. Our model is compared to several state-of-the-art defense methods, and in the majority of cases, our proposed framework demonstrates superior or comparable performance while minimizing the added time complexity.

Related Work

Following their success in diverse tasks, questions about the robustness of Deep Learning models have surfaced notably in computer vision (Goodfellow, Shlens, and Szegedy 2015; Ren et al. 2020), extending recently to discrete domains such as Natural Language Processing and graphs (Günnemann 2022). Adversarial attacks can be categorized into various subgroups depending on the attacker's knowledge and objectives, such as poisoning/evasion and targeted/un-targeted attacks. Notably, Nettack (Zügner, Akbarnejad, and Günnemann 2018) introduced a targeted attack method that perturbs both the graph structure and node features. This approach utilizes a greedy optimization algorithm to minimize an attack loss against a surrogate model. Building upon this work, Mettack (Zügner and Günnemann 2019) formulates the problem as a bi-level optimization task and employs meta-gradients to tackle it. Expanding on these advancements, (Zhan and Pei 2021) proposed a black-box gradient attack algorithm that overcomes several limitations of the original techniques. On an alternative front, (Dai et al. 2018) proposed using Reinforcement Learning to find appropriate graph adversarial attacks.

In addition to the advancements in adversarial attack techniques, research on defense methods against these attacks has gained attention, although it remains relatively less explored compared to computer vision defense strategies. Similar to image-based models, robust training (Zügner and Günnemann 2019) and aggregation (Geisler, Zügner, and Günnemann 2020) have been proposed as mechanisms to enhance the robustness of GNNs by iteratively augmenting the training set. Furthermore, defense strategies leveraging low-rank matrix approximation combined with graph anomaly detection (Ma et al. 2021) have been employed. For instance, GNN-Jaccard (Wu et al. 2019a) performs preprocessing on the graph's adjacency matrix to identify potential edge manipulations, while GNN-SVD (Entezari et al. 2020) employs a low-rank approximation of the adjacency matrix to filter out noise. Additionally, techniques such as edge pruning (Zhang and Zitnik 2020) and transfer learning (Tang et al. 2020) have been utilized to mitigate the impact of poisoning attacks. From another perspective, Seddik et al. (2022) add node feature kernels to the GCN's message passing operator to enhance the GCN's robustness. Lastly, RobustGCN (Zhu et al. 2019) introduces the use of Gaussian distributions as hidden representations of nodes in each convolutional layer, enabling the absorption of the effects of both structural and feature-based adversarial attacks. From another perspective, given the limitations of the aforementioned methods in terms of theoretical guarantees, there has been a growing interest in exploring robustness certificates (Zügner and Günnemann 2019; Bojchevski and Günnemann 2019) as a promising direction to quantify a model's robustness and providing attack-independent guarantees. For instance, (Bojchevski, Gasteiger, and Günnemann 2020) introduced the use of randomized smoothing techniques to offer highly scalable model-agnostic certificates for graphs. Their approach provides a robustness guarantee that is independent of the attack method employed. Furthermore, (Jin et al. 2020) proposed robustness certificates specifically for GCN-based graph classification in the presence of topological perturbations. These certificates consider both local and global budgets, enabling a comprehensive robustness analysis of the model's robustness.

Recently, defending against adversarial attacks through the injection of noise into the architecture has emerged as a promising approach in the field of Computer Vision. Several studies (Pinot et al. 2019; Liu et al. 2018; Siraj Rakin, He, and Fan 2018) have shown that noise injection can enhance the robustness of networks against adversarial perturbations. From another perspective, and in the context of GNNs, the work (Wu, Bojchevski, and Huang 2023) investigated the effect of injecting noise, specifically adversarial weight perturbation, on improving the generalization of models. The findings of this study demonstrate that these perturbations effectively mitigate the vanishing-gradient issue and lead to significant enhancements in generalization performance. Our work extends these insights by considering the application of noise injection schemes to enhance the robustness of GNNs against adversarial attacks.

Preliminaries

We begin by introducing several fundamental concepts.

Notation and Problem Setup. Let $G = (V, E)$ be a graph where V is its set of vertices and E its set of edges. We will denote by $n = |V|$ and $m = |E|$ the number of vertices and number of edges, respectively. Let $\mathcal{N}(v)$ denote the set of neighbors of a node $v \in V$, i. e., $\mathcal{N}(v) = \{u \colon (v, u) \in E\}$. The degree of a node is equal to its number of neighbors, i. e., equal to $|\mathcal{N}(v)|$ for a node $v \in V$. A graph is commonly represented by its adjacency matrix $A \in \mathbb{R}^{n \times n}$ which encodes edge information. The (i, j)-th element of the adjacency matrix is equal to the weight of the edge between the i-th and j-th node of the graph and a weight of 0 in case the edge does not exist. In some settings, the nodes of a graph might be annotated with feature vectors. We use $X \in \mathbb{R}^{n \times K}$ to denote the node features where K is the feature dimensionality. The feature of the i-th node of the graph corresponds to the i-th row of X. In a node classification setting, we consider a graph G, represented by its adjacency matrix A and its node attribute matrix X. Formally, given a set of labeled $V_L \subset V$, where nodes are assigned exactly one class in $\mathcal{C} = \{y_1, y_2, \ldots, y_c\} \subset \mathcal{Y}$, the goal is to learn a function f_θ, which maps each node $v \in V$ to exactly one of the c classes in $\mathcal{C}$ while minimizing a classification loss (the cross entropy loss for example).

GNNs. A GNN model consists of a series of neighborhood aggregation layers that use the graph structure and the node feature vectors from the previous layer to generate new representations for the nodes. Specifically, GNNs update node feature vectors by aggregating local neighborhood information. Suppose we have a GNN model that contains T neighborhood aggregation layers. Let also $\mathbf{h}_v^{(0)}$ denote the initial feature vector of node v, i. e., the row of matrix X that corresponds to node v. At each iteration ($t > 0$), the hidden state $\mathbf{h}_v^{(t)}$ of a node v is updated as follows:

$$\mathbf{a}_v^{(t)} = \text{AGGREGATE}^{(t)}\left(\left\{\mathbf{h}_u^{(t-1)} \colon u \in \mathcal{N}(v)\right\}\right);$$
$$\mathbf{h}_v^{(t)} = \text{COMBINE}^{(t)}\left(\mathbf{h}_v^{(t-1)}, \mathbf{a}_v^{(t)}\right),$$

where AGGREGATE is a permutation invariant function that maps the feature vectors of the neighbors of a node v to an aggregated vector. This aggregated vector is passed along with the previous representation of v, i. e., $\mathbf{h}_v^{(t-1)}$, to the COMBINE function which combines those two vectors and produces the new representation of v.

Proposed Approach

In this section, we provide a mathematical formalization of robustness specifically tailored to Graph Neural Networks (GNNs). Subsequently, we investigate the impact of noise injection on the robustness of GNNs. Throughout our analysis, without loss of generality, we will focus on the semi-supervised node classification task as a representative scenario. Let us consider the following three metric spaces, the graph space associated with the adjacency matrices $(\mathcal{A}, \|\cdot\|_{\mathcal{A}})$, the feature space associated with the node feature attributes $(\mathcal{X}, \|\cdot\|_{\mathcal{X}})$ and the label space $(\mathcal{Y}, \|\cdot\|_{\mathcal{Y}})$. We further consider an underlying probability distribution $\mathcal{D}$ defined on $(\mathcal{A}, \mathcal{X})$. Throughout this section, $\|\cdot\|$ denotes the Euclidean (resp., spectral) norm for vectors (resp., matrices).

Graph Adversarial Attacks

Let us consider a trained victim classifier $f : (\mathcal{A}, \mathcal{X}) \to \mathcal{Y}$ and let $(A, X) \in (\mathcal{A}, \mathcal{X})$ be an input graph with its associated label vectors $y \in \mathcal{Y}$, such that $f(A, X) = y$. The objective of an adversarial attack is to generate a perturbed graph, represented by its adjacency matrix $\tilde{A}$ and the corresponding features $\tilde{X}$, which is slightly different from the original input (A, X), and whose prediction is different from the original one. The adversarial aim can be therefore formulated as the search for a perturbed attributed graph $(\tilde{A}, \tilde{X})$ within a defined similarity budget ϵ, such that $f(\tilde{A}, \tilde{X}) \neq f(A, X)$. From this perspective, we can define the adversarial risk of a GNN as the expected behavior or output of adjacent graphs to a given input graph's neighborhood within a budget ϵ. This can be mathematically formulated as the following:

$$\mathcal{R}_\epsilon[f] = \underset{\substack{(A,X)\sim\mathcal{D} \\ (\tilde{A},\tilde{X})\in\mathcal{N}_\epsilon(A,X)}}{\mathbb{E}} [d_{\mathcal{Y}}(f(\tilde{A}, \tilde{X}), f(A, X))], \quad (1)$$

with $\mathcal{N}_\epsilon(A, X) = \{(\tilde{A}, \tilde{X}) : d_{\mathcal{A},\mathcal{X}}((A, X(, (\tilde{A}, \tilde{X})) < \epsilon\}$ being the input's graph neighborhood containing the valid adversarial candidates for any budget $\epsilon \geq 0$ and $d_{\mathcal{A},\mathcal{X}}$ and $d_{\mathcal{Y}}$ can be any defined distances in the measurable input and output, spaces $(\mathcal{A}, \mathcal{X})$ and $\mathcal{Y}$. In our analysis, we will consider a distance metric that takes into account both the graph structure and its associated node features.

$$d_{\mathcal{A},\mathcal{X}}((A, X), (\tilde{A}, \tilde{X})) = \min_{P\in\Pi}\{\|A - P\tilde{A}P^T\|_2 + \|X - P\tilde{X}\|_2\},$$

with Π being the set of permutation matrices. While we will be focusing on the ℓ_2 norm, other norms may be used depending on the relevant penalization constraints corresponding to the considered use case.

Quantifying the precise adversarial risk of a GNN, as defined in Equation (1), poses a significant challenge. However, an effective and more manageable approach is to establish an upper bound on this risk. By deriving such an upper bound, users can gain a comprehensive understanding of the GNN's susceptibility to adversarial attacks and make informed assessments of its robustness based on the specific task at hand. For example, in certain scenarios like social networks, where a limited number of successful attacks may not have severe consequences, a larger upper bound on the adversarial risk might be tolerable. While in other more sensitive areas, such as financial applications, we need to aim for a much tighter upper bound to control the confidence level of the adversarial risk. From this perspective, we introduce the notion of a GNN's robustness as follows.

Definition 1. (Adversarial Robustness). The graph-based function $f : (\mathcal{A}, \mathcal{X}) \to \mathcal{Y}$ is said to be $(\epsilon, \gamma) -$ robust if its adversarial risk is upper-bounded, i. e., $\mathcal{R}_\epsilon[f] \leq \gamma$ with respect to the chosen graph distances in the input and output metric spaces.

Our introduced robustness formulation deviates from the common approach seen in the literature, which typically focuses on evaluating worst-case scenarios using specific adversarial examples. Instead of considering the model's behavior only under individual adversarial instances, our method examines how the model performs more generally within a defined neighborhood. This perspective leans towards a concept of "average" robustness, expanding on the conventional worst-case based adversarial robustness that is often emphasized in adversarial studies. We argue that our "average" definition provides a more comprehensive grasp of the model's robustness, encompassing the classical quantification of adversarial vulnerabilities. In fact, ensuring "average" robustness inherently guarantees "worst-case" robustness as will be demonstrated in Proposition 2:

Proposition 2. *Let $f : (\mathcal{A}, \mathcal{X}) \rightarrow \mathcal{Y}$ be a graph-based function, with respect to the chosen input and output space distance, the following holds:*

f is (ϵ, γ) − robust ⇒ f is (ϵ, γ) − "worst-case" robust.

As a result of Proposition 2, our upper-bound analysis can be used for both our "average" robustness measure and the traditional "worst-case" adversarial robustness, effectively connecting the two. The detailed proof of Proposition 2 is in Appendix A.

Effect of Noise Injection

Our study aims to investigate the effect of noise addition in term of defending against adversarial attacks. Specifically, we consider injecting noise sampled from a predefined distribution during the training and inference time to enhance the robustness of an underlying GNN. We should hence, and similar to Pinot et al. (2019), consider a probabilistic space as our output space and our victim model f as a probabilistic mapping where an output is obtained by sampling from the mapping. Accordingly, we will consider the Kullback–Leibler (KL) divergence as the corresponding output distance $d_{\mathcal{Y}}$ as defined in our adversarial risk quantification introduced in Equation (1).

Our analysis will focus on the widely used GCN and GIN within the broader context of GNNs. For illustration, and as introduced in the "Preliminaries" Section, we can write an iteration of the iterative process of GCN as follows:

$$H^{(\ell)} = \phi^{(\ell)}(\hat{A}H^{(\ell-1)}W^{(\ell)}), \tag{2}$$

where $H^{(\ell)}$ represents the hidden state in the ℓ-th GCN layer with $H^{(0)}$ corresponding to the initial node features $X \in \mathbb{R}^{n \times K}$, $W^{(\ell)} \in \mathbb{R}^{p \times e}$ is the weight matrix in the ℓ-th layer, e is the embedding dimension and $\phi^{(\ell)}$ is a non-linear activation function. Moreover, $\hat{A} \in \mathbb{R}^{n \times n}$ is the normalized adjacency matrix $\hat{A} = D^{-1/2}AD^{-1/2}$. We note that for the GIN, we use the following adaptation of the adjacency matrix $\hat{A} = A + (1 + \lambda)I$.

In the remainder of our theoretical analysis, we consider our victim model to be any GCN or GIN based graph classifier. We additionally assume that f contains only 1-Lipschitz continuous activation functions, which is the case for commonly used activation functions such as the Hyperbolic Tangent. While in practice one can choose to to sample the injected noise from a variety of distributions, our theoretical study will focus on the centered Gaussian distribution $\mathcal{N}(0, I)$ with a scaling parameter β controlling its covariance matrix. The work (Wu, Bojchevski, and Huang 2023), which mainly focused on connecting adversarial weight perturbation to generalization, has shown that injecting noise at each layer can lead to a collapse in the model's generalization. As a result, we will restrict the introduction of noise to specific layers to yield better results. Under these assumptions, our victim model can be expressed as $f(\cdot) = \Phi^{\ell} \circ \cdots \circ \Phi^{i+1}(\Phi^{i} \circ \cdots \circ \Phi^{1}(\cdot) + T)$, where T represents a Gaussian random variable.

Theorem 3. *Let f denote a graph-based function composed of 2 layers and based on 1-Lipschitz continuous activation functions. We consider injecting noise drawn from a centered Gaussian with a scaling parameter β. When subject to structural perturbations of the input graph (A, X), with a budget ϵ, we have with respect to Definition 1:*

- *If f is GCN-based then f is (ϵ, γ) − robust with*
$$\gamma = \frac{2(\|W^{(2)}\|\|W^{(1)}\|\|X\|\epsilon)^2}{\beta};$$
- *If f is GIN-based then f is (ϵ, γ) − robust with*
$$\gamma = \frac{(\|W^{(2)}\|\|W^{(1)}\|\|X\|\epsilon(2\|A\|+\epsilon))^2}{2\beta},$$

where $W^{(\ell)}$ denotes the weight matrix of the ℓ-th layer.

Theorem 3 provides an upper bound on a GCN and GIN based graph classifier's robustness and establishes the connection between noise injection and defending against adversarial attacks based on structural perturbations with a predefined neighborhood and budget ϵ. Since a tighter upper bound intuitively signifies a higher level of robustness in the targeted victim model, based on the results derived from the theorem, controlling the injected noise using the β parameter can effectively enhance the model's robustness. However, it is important to exercise caution when increasing the injected noise as it can potentially compromise the model's performance. Hence, striking a balance between defending against adversarial attacks and preserving the model's clean accuracy becomes crucial. Furthermore, although the previous theorem focuses on a 2-layers graph classifier known for its benchmark accuracy across diverse datasets, the results can be extended to graph classifiers with L layers. Another notable observation from the analysis of Theorem 3 is that the computed upper bound for a GCN-based classifier is significantly tighter compared to that of a GIN-based classifier. This suggests that the GCN model exhibits greater robustness to structural perturbations, which aligns with our intuition since the normalization of the adjacency matrix is expected to attenuate the effects of adversarial perturbations. The proof of Theorem 3 is provided in Appendix B.

While Theorem 3 and our experimental analysis focus on structural perturbations, similar analysis can be applied to node feature-based adversarial attacks. In this context, Theorem 4 sheds light on the link between noise injection and enhancing robustness when the underlying model is subject to adversarial attacks targeting node features.

Theorem 4. *Let f denote a graph-based classifier composed of 2 layers and based on 1-Lipschitz continuous activation functions. We consider injecting noise drawn from a centered Gaussian with a scaling parameter β. When subject to node feature-based perturbations of the input graph (A, X), we have with respect to Definition 1:*

- *If f is GCN-based then f is $(\epsilon, \gamma) -$ robust with*
$$\gamma = \frac{(\|W^{(2)}\|\|W^{(1)}\|\epsilon)^2}{2\beta};$$
- *If f is GIN-based then f is $(\epsilon, \gamma) -$ robust with*
$$\gamma = \frac{(\|A\|\|W^{(2)}\|\|W^{(1)}\|\epsilon)^2}{2\beta},$$

where $W^{(\ell)}$ denotes the weight matrix of the ℓ-th layer.

The proof of Theorem 4 can be found in Appendix C.

Complexity and Advantage of Our Approach

Many existing defense methods suffer from a significant increase in complexity as the input graph size grows, making them challenging to apply in practical scenarios. For example, GNNGuard (Zhang and Zitnik 2020) involves computing neighbor importance estimation, which has a complexity of $\mathcal{O}(e \times |E|)$, where e denotes the embedding dimension and $|E|$ represents the number of edges. GCN-Jaccard, which preprocesses the network by eliminating edges connecting nodes with a Jaccard similarity of features smaller than a chosen threshold, has a complexity of $\mathcal{O}(|E|)$. Moreover, GNN-SVD, to discard the high-rank perturbations, computes a low-rank approximation of the adjacency and features matrices derived from their SVD for which the complexity is $\mathcal{O}(|V|^3)$. In contrast, our proposed approach based on noise injection in the architecture is advantageous due to its minimal complexity, requiring only sampling from a distribution. In this perspective, we note that we provide an experimental comparison analysis of the training time of the different cited methods against our proposed framework in Appendix E. Additionally, unlike many existing methods, our approach does not compromise the performance of the underlying GCN when applied to clean, non-attacked graphs, as will be shown in our experimental results.

Experimental Results

This section focuses on empirically validating our theoretical findings by evaluating the performance of the proposed approach on real-world benchmarks. We begin by outlining the experimental settings employed in our study, followed by a comprehensive analysis and discussion of the obtained results. Through our experimental evaluation, we aim to address three main key aspects: firstly, the effectiveness of our method in defending against adversarial attacks, particularly structural perturbations, and secondly, its capability to maintain the model's accuracy and performance, especially when tested on non-attacked input graphs and finally its complexity in terms of training time compared to other methods.

Experimental Setup

We focus on node classification where we use the citation networks Cora, CiteSeer, and PubMed (Sen et al. 2008) and blog and citation graphs, i. e., Polblogs (Adamic and Glance 2005) and OGBN-Arxiv (Hu et al. 2020). Note that in the Polblogs graph, node features are not available. Further information about the used datasets and implementation details are provided in Appendix D. For all the experiments, the baseline models consist of a 2-layer GCN-based classifier combined with an MLP as a readout. This choice aimed to ensure a fair evaluation of the models' robustness within the same architectural conditions. The experiments were conducted using the Adam optimizer (Kingma and Ba 2015) and standardized hyperparameters, including a learning rate of 1e-2, 300 epochs, and 16 as the hidden dimension. To reduce the impact of random initialization, we repeated each experiment 10 times and used the train/validation/test splits provided with the datasets (Yang, Cohen, and Salakhudinov 2016). Our code is publicly available in GitHub[1].

Attacks. We use three main global structural-based adversarial attacks: **(i)** We first consider the optimization-based formulation of the adversarial task Mettack with the "Meta-Self" training strategy. **(ii)** We afterward consider another optimization-based adversarial attack based on Proximal Gradient Descent (PGD) (Xu et al. 2019a) and **(iii)** we finally consider DICE (Zügner and Günnemann 2019). For all these attacks, we tested and considered two perturbation budgets (in term of percentage) $\epsilon \in \{5\%, 10\%\}$.

Baseline Models. To provide a comprehensive empirical evaluation, we compared our proposed defense algorithm, NoisyGCN, against four baseline methods that specifically address structural perturbations. The baseline models we consider are: **(i)** GNN-Jaccard (Wu et al. 2019a) that preprocesses the input adjacency matrix to identify potential edge manipulations. **(ii)** RGCN (Zhu et al. 2019), that employs Gaussian distributions as hidden representation to absorb the effect of structural adversarial attacks. **(iii)** GNN-SVD, which employs a low-rank approximation of the adjacency matrix to filter out noise, and **(iv)** GNNGuard (Zhang and Zitnik 2020), which is based on edge pruning to defend against adversarial perturbations.

Experimental Results

Table 1 presents the average node classification accuracies for the GCN, the GNNGuard, GNN-Jaccard, RGNN, GNN-SVD, and the proposed approach, NoisyGNN for both GCN and GIN. The empirical findings reveal that, in the absence of attacks, the proposed approach demonstrates comparable accuracy to the classical GCN, and in some cases, it even improves the model's generalization and performance, as studied and analyzed by prior research (Wu, Bojchevski, and Huang 2023). Importantly, these results affirm that our approach does not compromise the performance of the underlying network, addressing our second research question. This is particularly significant as real-world scenarios often involve uncertain knowledge regarding potential malicious perturbations on the input graph. Hence, it is crucial that an effective defense strategy does not diminish the predictive capabilities of the model, while simultaneously enhancing its robustness. The results furthermore indicate that

[1] Code: https://github.com/Sennadir/NoisyGNN

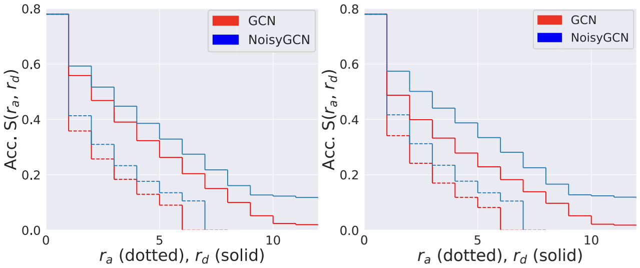
	Dataset	ϵ	GCN					GIN				
			Guard	Jaccard	SVD	RGNN	Noisy	Guard	Jaccard	SVD	RGNN	Noisy
Mettack	Cora	0%	77.5±0.7	80.9±0.7	80.6±0.4	**83.5±0.3**	83.2±0.4	82.5±0.3	80.8±0.5	79.7±1.0	**83.5±0.3**	83.0±0.2
		5%	75.8±0.6	78.9±0.8	78.4±0.6	78.3±0.6	**81.2±0.7**	79.5±0.7	78.9±0.8	78.2±1.3	78.3±0.6	**81.1±0.5**
		10%	74.7±0.4	**76.7±0.7**	71.5±0.8	70.7±0.8	74.5±0.6	73.8±1.3	76.4±0.5	73.9±1.2	70.7±0.8	**76.7±0.8**
	CiteSeer	0%	70.1±1.5	71.2±0.7	70.7±0.4	**72.3±0.5**	71.9±0.4	71.2±0.6	**72.5±1.2**	71.9±1.6	72.3±0.5	71.9±0.6
		5%	69.9±1.1	70.3±2.3	68.9±0.7	70.6±0.7	**72.3±0.6**	**71.8±0.7**	70.9±1.7	70.8±1.8	70.6±0.7	71.3±0.8
		10%	70.0±1.5	67.5±2.1	68.8±0.6	68.7±1.2	**70.4±0.8**	67.8±1.1	**70.2±1.5**	69.3±1.8	68.7±1.2	69.2±1.3
	PubMed	0%	84.5±0.6	85.0±0.5	82.7±0.3	**85.1±0.8**	85.0±0.6	**85.1±0.6**	84.9±0.9	82.8±0.3	**85.1±0.8**	84.8±0.4
		5%	**84.3±0.9**	79.6±0.3	81.3±0.6	81.1±0.7	81.8±0.4	**83.2±0.5**	81.6±0.7	82.1±0.7	81.1±0.7	82.4±0.9
		10%	**84.1±0.3**	67.4±1.1	81.1±0.7	65.2±0.4	73.3±0.6	78.5±0.9	77.5±1.5	**81.6±0.6**	65.2±0.4	78.9±1.8
	PolBlogs	0%	93.1±0.6	-	86.5±0.8	94.9±0.3	**95.2±0.4**	**95.6±0.9**	-	93.4±0.6	95.2±0.3	94.9±0.7
		5%	72.8±0.8	-	**85.1±1.6**	76.0±0.8	79.7±0.6	94.5±0.8	-	92.8±0.9	76.0±0.8	**94.7±0.5**
		10%	68.7±1.0	-	**84.8±2.3**	69.2±1.2	73.4±0.5	92.5±0.9	-	92.1±1.6	69.2±1.2	**92.8±0.6**
PGD	Cora	5%	71.0±1.0	73.9±0.8	69.9±0.6	75.8±0.9	**76.6±0.3**	**81.8±1.1**	80.1±0.6	74.6±1.2	75.8±0.9	81.3±0.4
		10%	69.9±1.6	72.2±1.4	65.3±0.9	72.4±1.8	**73.4±0.5**	**81.0±1.5**	79.7±0.8	73.9±1.1	72.4±1.8	79.9±0.7
	CiteSeer	5%	57.9±2.8	62.9±1.5	61.7±1.3	58.1±2.2	**64.5±1.2**	69.8±0.4	70.1±0.5	67.9±0.9	58.1±2.2	**70.7±0.5**
		10%	58.2±3.8	61.3±0.7	59.5±0.3	56.2±0.8	**62.2±1.0**	68.9±0.9	69.4±0.7	65.6±1.3	56.2±0.8	**70.0±0.7**
	PubMed	5%	75.3±0.4	76.1±0.7	67.7±1.5	**78.5±0.8**	76.2±0.7	**81.0±0.3**	80.8±0.6	80.8±0.9	78.5±0.8	80.6±0.4
		10%	**70.7±0.9**	64.7±1.2	67.5±1.7	65.6±0.9	65.2±1.1	**80.3±0.7**	79.9±0.8	80.1±1.2	65.6±0.9	79.6±0.6
	PolBlogs	5%	76.8±0.6	-	82.1±1.1	82.5±0.4	**83.2±0.7**	93.4±0.4	-	88.4±1.2	82.5±0.4	**94.0±0.6**
		10%	74.3±0.8	-	**80.2±1.5**	76.5±0.7	77.6±0.9	91.5±1.2	-	86.2±1.6	76.5±0.7	**92.1±0.7**
DICE	Cora	5%	76.4±0.4	79.6±0.6	74.9±1.3	81.9±0.6	**82.5±0.8**	81.9±0.3	79.7±0.8	79.3±0.8	**81.9±0.6**	81.7±0.5
		10%	76.6±0.5	78.6±0.8	73.5±1.5	80.0±0.6	**80.5±0.6**	79.2±0.6	78.6±0.6	78.1±1.1	80.0±0.6	**80.5±0.8**
	CiteSeer	5%	68.5±1.6	**70.9±0.4**	69.4±1.6	69.3±0.5	70.8±0.3	69.6±1.5	70.3±0.7	65.7±1.8	69.3±0.5	**70.8±0.9**
		10%	69.9±1.5	69.9±0.6	68.1±1.5	67.8±1.1	**70.4±0.8**	68.3±0.7	69.3±0.6	64.5±2.3	67.8±1.1	**69.6±1.2**
	PubMed	5%	**84.0±0.8**	83.4±0.7	81.5±0.8	83.8±0.6	83.6±0.9	**84.0±0.4**	83.5±0.3	82.3±0.7	83.8±0.6	83.8±0.3
		10%	**83.6±1.0**	81.8±0.5	81.4±0.5	82.4±0.8	82.1±2.3	**82.9±0.8**	82.0±0.5	82.0±0.9	82.4±0.8	82.5±0.6
	PolBlogs	5%	81.3±0.7	-	86.5±2.3	89.6±0.4	**90.3±0.3**	93.3±0.3	-	90.9±0.5	89.6±0.4	**93.5±0.5**
		10%	78.9±0.6	-	85.3±2.8	85.5±0.9	**86.1±0.9**	**91.1±1.2**	-	89.9±0.8	85.5±0.9	89.7±0.8

Table 1: Classification accuracy (± standard deviation) of the models on different benchmark node classification datasets for different perturbation rates ϵ. The best accuracy in each setting, each dataset, and each model is typeset in bold.

our proposed noise injection approach performs on par with and even surpasses state-of-the-art defense baselines in several instances when working with both GCN and GIN. Notably, it demonstrates greater efficiency when subjected to the "PGD" and "DICE" attack framework. Moreover, we observe an almost consistent outperformance compared to GNN-SVD, GNN-Jaccard, and RGNN, while also exhibiting competitive performance against the highly performant GNNGuard. It is important to highlight that despite similar performance to GNNGuard, the main advantage of our approach is its theoretical guarantees but also its significantly reduced complexity in terms of operation and time. The complete time analysis study is provided in Appendix E.

Evaluation Through Robustness Certificates. We furthermore conducted a comparative analysis between the GCN and our proposed NoisyGCN where we assessed their certified robustness on the Cora and CiteSeer datasets using the sparse randomized smoothing approach (Bojchevski, Gasteiger, and Günnemann 2020). Specifically, we plotted their certified accuracy $S(r_a, r_d)$ while varying the radii for addition (r_a) and deletion (r_d) of edges. Figure 1 illustrates the results, showcasing that our proposed noise injection method significantly enhances the certified accuracy of NoisyGCN when subject to structural perturbations. This observation emphasizes the effectiveness of our approach in improving the model's robustness.

Node Feature Based Adversarial Attacks. We furthermore empirically study Theorem 4 concerning node feature-based adversarial attacks. We consider two main feature-based adversarial attacks: **(i)** The baseline random noise attack adding Gaussian noise $\mathcal{N}(0, \mathbf{I})$ to the node features with a scaling parameter ξ to control the attack budget; **(ii)** The white-box Proximal Gradient Descent (PGD) (Xu et al. 2019a) attack. We note that this attack is more adapted for continuous spaces and hence is known to be powerful in the node feature space. For this analysis, we focused on the



(a) Cora

(b) CiteSeer

Figure 1: Robustness guarantees on (a) Cora and (b) CiteSeer, where r_a is the certified radius – maximum number of adversarial additions (and r_d for deletions).

Attack	GCN-k	AirGNN	RGCN	NoisyGCN
Clean	56.0±0.3	61.9±0.9	65.1±1.8	**66.9±0.5**
$\xi = 0.5$	52.8±0.5	59.0±1.3	63.8±1.9	**65.8±2.0**
$\xi = 1.0$	46.6±0.6	51.9±1.6	63.0±2.4	**64.3±1.3**
PGD	49.9±0.7	55.7±0.9	63.6±0.7	**64.9±1.1**

Table 2: Classification accuracy (± standard deviation) of the models on the OGBN-Arxiv node classification dataset before (“Clean”) and after the attack application.

citation network between Computer Science arXiv papers OGBN-Arxiv (Hu et al. 2020) to demonstrate the ability of our method on larger datasets. We note that this dataset could not be used for the structural perturbations since the majority of the available attacks require access to a dense form of the adjacency matrix which is not very realistic at large scale. We compare our method against defense methods adapted for feature-based adversarial perturbations: **(i)** GCN-k (Seddik et al. 2022) that proposes to enhance the robustness of GNNs to noise and adversarial attacks by incorporating a node feature kernel into the message passing operators; **(ii)** RobustGCN (RGCN) leveraging Gaussian distributions as hidden representations to absorb the impact of adversarial attacks; **(iii)** AIRGNN (Liu et al. 2021) that edited the Message Passing module with adaptive residual connections and feature aggregation to improve the GNN's robustness against abnormal node features. Table 2 reports the clean and attacked accuracy for the considered benchmark defense methods alongside our introduced NoisyGCN. The findings highlight our method's ability to defend against node feature-based adversarial attacks, thereby affirming the theoretical conclusions drawn in Theorem 4.

Combining Adversarial Defenses. As previously discussed, our approach is model-agnostic, allowing it to be applied to various GNN architectures. Therefore, it is meaningful to experimentally assess the impact of combining our proposed noise injection with other benchmark defense methods, especially for large attack budgets. In this context, we consider methods that do not modify the model architecture but instead introduce a defense layer. Specifically, we examine pre-processing methods such as GNN-Jaccard (Wu et al. 2019a) and edge pruning methods like GNNGuard (Zhang and Zitnik 2020). However, certain methods such as RGNN (Zhu et al. 2019) are excluded from this section due to their major edits in the architecture, which do not directly align with our proposed noise injection. We report the results for both GCN and GIN for the Mettack with an attack budget of $\epsilon = 25\%$ while we provide results of the other attacks and budgets in Appendix F.

Method	Cora	CiteSeer	PolBlogs
GINGuard + Noisy	61.8±0.5 **66.2±1.3**	55.6±1.8 **58.3±1.9**	82.7±0.6 **83.6±0.8**
GIN-Jaccard + Noisy	70.4±1.1 **72.9±0.8**	61.2±2.3 **64.9±1.8**	- -
GCNGuard + Noisy	69.5±0.7 **72.4±1.2**	66.2±0.6 **68.9±0.9**	64.7±0.8 **65.8±1.3**
GCN-Jaccard + Noisy	66.7±0.5 **69.6±0.9**	61.2±1.1 **63.1±0.6**	- -

Table 3: Classification accuracy (± standard deviation) of combining defense methods with the proposed noise injection on different benchmark datasets.

The results in Table 3 demonstrate the improved defense results obtained by combining our noise injection approach with existing defense methods across various datasets and underlying models. Notably, the combination shows promising outcomes in mitigating the added defense layer's effect on the clean accuracy, addressing a significant weakness in methods like GNNGuard. These findings underscore the capability of our method to enhance the robustness of any underlying architecture with minimal additional complexity.

Conclusion

In this study, we present NoisyGNN, a highly effective and cost-effective defense method for Graph Neural Networks (GNNs). Through rigorous theoretical analysis, we establish a clear and compelling connection between noise injection in the victim model's architecture and enhanced robustness. Our proposed method offers a significant advantage by introducing minimal additional complexity while delivering strong defense performance. Comprehensive experimental comparisons conducted on diverse real-world datasets demonstrate that our proposed framework achieves comparable or even superior performance when compared to standard GCN and GIN models, as well as existing defense methods specifically designed for those models. This highlights the effectiveness and versatility of our approach. While our primary focus in this study was on the GCN and GIN architectures, it is important to note that our approach is model-agnostic, as demonstrated by successful combinations of our defense method with other existing techniques. Additionally, our theoretical analysis has the potential to be explored and extended to other GNN architectures, thus opening up avenues for further research and application.

Acknowledgements

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computation (on GPUs) was enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at Alvis partially funded by the Swedish Research Council through grant agreement no. “2022-06725”. This work was granted access to the HPC resources of IDRIS under the allocation “2023-AD010613410R1” made by GENCI. Y.A. is supported by the French National research agency via the AML-HELAS (ANR-19-CHIA-0020) project.

References

Adamic, L. A.; and Glance, N. 2005. The Political Blogosphere and the 2004 U.S. Election: Divided They Blog. In *Proceedings of the 3rd International Workshop on Link Discovery*, LinkKDD ’05, 36–43. New York, NY, USA: Association for Computing Machinery. ISBN 1595932151.

Bojchevski, A.; Gasteiger, J.; and Günnemann, S. 2020. Efficient Robustness Certificates for Discrete Data: Sparsity-Aware Randomized Smoothing for Graphs, Images and More. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 1003–1013. PMLR.

Bojchevski, A.; and Günnemann, S. 2019. Certifiable robustness to graph perturbations. *Advances in Neural Information Processing Systems*, 32.

Dai, H.; Li, H.; Tian, T.; Huang, X.; Wang, L.; Zhu, J.; and Song, L. 2018. Adversarial Attack on Graph Structured Data. In *Proceedings of the 35th International Conference on Machine Learning*, 1115–1124.

Entezari, N.; Al-Sayouri, S. A.; Darvishzadeh, A.; and Papalexakis, E. E. 2020. 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*, WSDM ’20, 169–177. New York, NY, USA: Association for Computing Machinery. ISBN 9781450368223.

Feng, F.; He, X.; Tang, J.; and Chua, T.-S. 2021. Graph Adversarial Training: Dynamically Regularizing Based on Graph Structure. *IEEE Transactions on Knowledge and Data Engineering*, 33(6): 2493–2504.

Geisler, S.; Zügner, D.; and Günnemann, S. 2020. Reliable Graph Neural Networks via Robust Aggregation. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 13272–13284. Curran Associates, Inc.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural Message Passing for Quantum Chemistry. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 1263–1272. PMLR.

Goodfellow, I.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations*.

Günnemann, S. 2022. Graph neural networks: Adversarial robustness. In *Graph Neural Networks: Foundations, Frontiers, and Applications*, 149–176. Springer.

Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; and Leskovec, J. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33: 22118–22133.

Jin, H.; Shi, Z.; Peruri, V. J. S. A.; and Zhang, X. 2020. Certified Robustness of Graph Convolution Networks for Graph Classification under Topological Attacks. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 8463–8474. Curran Associates, Inc.

Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; and Riley, P. 2016. Molecular graph convolutions: moving beyond fingerprints. *Journal of Computer-Aided Molecular Design*, 30(8): 595–608.

Kingma, D.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*. San Diega, CA, USA.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.

Liu, X.; Cheng, M.; Zhang, H.; and Hsieh, C.-J. 2018. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 369–385.

Liu, X.; Ding, J.; Jin, W.; Xu, H.; Ma, Y.; Liu, Z.; and Tang, J. 2021. Graph Neural Networks with Adaptive Residual. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 9720–9733. Curran Associates, Inc.

Ma, X.; Wu, J.; Xue, S.; Yang, J.; Zhou, C.; Sheng, Q. Z.; Xiong, H.; and Akoglu, L. 2021. A Comprehensive Survey on Graph Anomaly Detection with Deep Learning. *IEEE Transactions on Knowledge and Data Engineering*, 1–1.

Pinot, R.; Meunier, L.; Araujo, A.; Kashima, H.; Yger, F.; Gouy-Pailler, C.; and Atif, J. 2019. Theoretical evidence for adversarial robustness through randomization. *Advances in neural information processing systems*, 32.

Qabel, A.; Ennadir, S.; Nikolentzos, G.; Lutzeyer, J. F.; Chatzianastasis, M.; Boström, H.; and Vazirgiannis, M. 2022. Structure-Aware Antibiotic Resistance Classification Using Graph Neural Networks. In *NeurIPS 2022 AI for Science: Progress and Promises*.

Ren, K.; Zheng, T.; Qin, Z.; and Liu, X. 2020. Adversarial Attacks and Defenses in Deep Learning. *Engineering*, 6(3): 346–360.

Schuchardt, J.; Bojchevski, A.; Gasteiger, J.; and Günnemann, S. 2021. Collective Robustness Certificates: Exploiting Interdependence in Graph Neural Networks. In *International Conference on Learning Representations*.

Seddik, M. E. A.; Wu, C.; Lutzeyer, J. F.; and Vazirgiannis, M. 2022. Node feature kernels increase graph convolutional network robustness. In *International Conference on Artificial Intelligence and Statistics*, 6225–6241. PMLR.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective Classification in Network Data. *AI Magazine*, 29(3): 93.

Siraj Rakin, A.; He, Z.; and Fan, D. 2018. Parametric Noise Injection: Trainable Randomness to Improve Deep Neural Network Robustness against Adversarial Attack. *arXiv e-prints*, arXiv:1811.09310.

Tang, X.; Li, Y.; Sun, Y.; Yao, H.; Mitra, P.; and Wang, S. 2020. Transferring Robustness for Graph Neural Network Against Poisoning Attacks. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. ACM.

Wu, H.; Wang, C.; Tyshetskiy, Y.; Docherty, A.; Lu, K.; and Zhu, L. 2019a. Adversarial Examples for Graph Data: Deep Insights into Attack and Defense. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 4816–4823. International Joint Conferences on Artificial Intelligence Organization.

Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; and Tan, T. 2019b. Session-based Recommendation with Graph Neural Networks. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 346–353.

Wu, Y.; Bojchevski, A.; and Huang, H. 2023. Adversarial Weight Perturbation Improves Generalization in Graph Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9): 10417–10425.

Xu, K.; Chen, H.; Liu, S.; Chen, P.-Y.; Weng, T.-W.; Hong, M.; and Lin, X. 2019a. Topology Attack and Defense for Graph Neural Networks: An Optimization Perspective. *arXiv preprint arXiv:1906.04214*.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019b. How Powerful are Graph Neural Networks? In *7th International Conference on Learning Representations*.

Yang, Z.; Cohen, W.; and Salakhudinov, R. 2016. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, 40–48. PMLR.

You, R.; Yao, S.; Mamitsuka, H.; and Zhu, S. 2021. DeepGraphGO: graph neural network for large-scale, multi-species protein function prediction. *Bioinformatics*, 37(Supplement_1): i262–i271.

Zhan, H.; and Pei, X. 2021. Black-box Gradient Attack on Graph Neural Networks: Deeper Insights in Graph-based Attack and Defense. *arXiv preprint arXiv:2104.15061*.

Zhang, X.; and Zitnik, M. 2020. GNNGuard: Defending Graph Neural Networks against Adversarial Attacks. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 9263–9275. Curran Associates, Inc.

Zhu, D.; Zhang, Z.; Cui, P.; and Zhu, W. 2019. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1399–1407.

Zügner, D.; Akbarnejad, A.; and Günnemann, S. 2018. Adversarial Attacks on Neural Networks for Graph Data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2847–2856.

Zügner, D.; and Günnemann, S. 2019. Adversarial attacks on graph neural networks via meta learning. In *7th International Conference on Learning Representations*.

Zügner, D.; and Günnemann, S. 2019. Certifiable Robustness and Robust Training for Graph Convolutional Networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM.