Robustness & Graph (Convolutional) Neural Networks

Machine Learning Seminar 20/21

Tim Bohne
tbohne@uni-osnabrueck.de

ABSTRACT

Graph neural networks enable the transfer of machine learning approaches to graph structured data. However, the models have proven to be prone to adversarial attacks, which questions their use in practical applications and leads to an increased focus on methods to strengthen their robustness. The intent of this paper is to provide a concise overview of the current state of research in the domain of graph (convolutional) neural networks with a focus on the robustness of the models.

1. INTRODUCTION

A currently very active research area inside the field of machine learning, or more precisely deep learning, considers models to learn from graph inputs. Those models are called graph neural networks (GNNs). Many real-life applications can be represented by a graph as data structure, e.g. *"modeling physical systems, learning molecular fingerprints, controlling traffic networks, and recommending friends in social networks"* [10]. Therefore, it is reasonable to think about combining graphs as data structures with state-of-the-art machine learning models. Unfortunately, in its non-Euclidean nature, graph data is not suitable for traditional deep learning models that are typically designed to work with Euclidean data such as images in computer vision or text in natural language processing. [10] One of the most successful types of models in the field of deep learning is the convolutional neural network (CNN). Hence, the idea is to generalize CNNs, which are designed to operate on Euclidean data with a spatial relation like images and text, to graph structured data. [10] The notion of graph embeddings, which learn to represent graph structures as low-dimensional vectors, provides another motivation for graph neural networks. [10] Liu et al. also highlight that traditional machine learning methods for graph analysis involve a lot of manual feature engineering which causes them to be rather inflexible and expensive. These considerations, along with the performance of GNNs in practical applications mentioned in section 2, indicate that the study of GNNs and their robustness is worthwhile. In particular, the robustness of the models is of great relevance when considering practical applications. In the following sections, a concise overview of the current state of robustness focused research in the domain of GNNs is provided.

After presenting some background for graph neural networks (GNNs) and particularly graph convolutional networks (GCNs) and their possible practical applications as well as the idea of robustness in general in section 2, an overview of the current state of the literature is provided in section 3. Subsequently, the core ideas, methods, and results of the first works on certifiable robustness of GNNs are introduced in section 4. Finally, there is a conclusion and a brief outlook on possible future research in section 5.

2. BACKGROUND

In general, GNNs are models to conduct deep learning with graph data. There are numerous reports of convincing performance of GNNs in practical applications (e.g. [4], [6], [12]), especially in the task of semi-supervised node classification. [15] In node classification tasks, one is usually presented with a graph in which a certain ratio of the nodes is labeled and the aim is to reasonably assign labels to the unlabeled ones. [20] Therefore, the given information in the graph has to be used to predict labels for the unlabeled nodes. An example would be to train a classifier with the labeled nodes and apply it to the unlabeled ones. Further typical applications for graphs as non-Euclidean data structures in machine learning are link prediction and clustering. [17] Section 2.1 provides a brief introduction into the basic structure and ideas of GNNs (and GCNs). Afterwards, in section 2.2, the concepts of robustness and adversarial attacks in machine learning in general, as well as applied to GNNs, are introduced.

2.1 Graph Neural Networks

In this section, the basic graph neural network model proposed by Scarselli et al. [11] gets introduced based on the description in *'Introduction to graph neural networks'*[10]. A GNN's goal is to learn a state embedding $h_v \in \mathbb{R}^S$ for each node v to generate an output o_v. As an example for such an output, they name the distribution of the predicted node label. In the original GNN model, which works with an undirected homogeneous graph, each node v has input features x_v as well as a set of edges $co[v]$ and neighbors $ne[v]$. The authors illustrate the structure with the example in fig. 1 where x_1 is the input feature of l_1, $co[l_1]$ contains the edges $\{l_{(1,2)}, l_{(1,4)}, l_{(1,6)}, l_{(3,1)}\}$, and $ne[l_1]$ contains the nodes $\{l_2, l_3, l_4, l_6\}$.

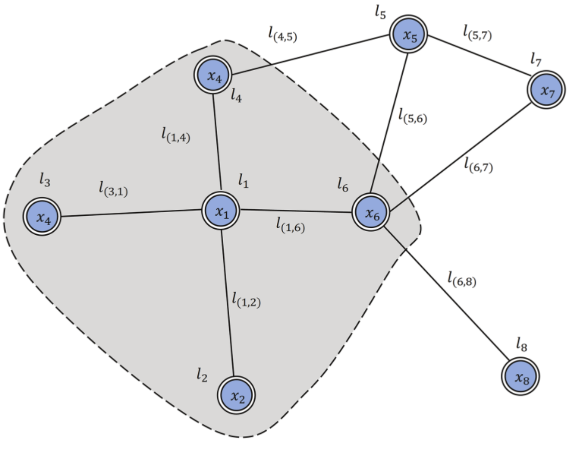

Figure 1: Structure example based on Scarselli et al. [10]

There are two important functions, f updates the node state based on the input neighborhood and g computes the output of a node. Let x be the input feature and h the hidden state:

- **node embedding:** $h_v = f(x_v, x_{co[v]}, h_{ne[v]}, x_{ne[v]})$
- **output embedding:** $o_v = g(h_v, x_v)$

Therefore, f takes as input the features of the node (x_v), the features of its edges ($x_{co[v]}$), the states ($h_{ne[v]}$), and the features of the nodes in its neighborhood ($x_{ne[v]}$). The node embedding h_v is then used in g to compute the output of v. Usually, h_v and o_v are described in a more compact form as matrices of stacked states H, outputs O, and features X (and node features X_N) with F and G being the stacked versions of f and g:

- $H = F(H, X)$
- $O = G(H, X_N)$

The $(t+1)$-th iteration of H is described as $H^{t+1} = F(H^t, X)$. Using the target information t_v for node v and the number of supervised nodes p, the loss term is described as $\sum_{i=1}^{p}(t_i - o_i)$ and a gradient-descent algorithm is proposed as learning technique.
This basic GNN model is still rather limited, but nevertheless an important milestone on the way to making neural networks applicable in the graph domain. [10] The restrictions give rise to numerous flavors of GNNs, such as graph convolutional networks (GCNs), which address certain limitations of the basic model.

Graph Convolutional Networks

The properties that define the different variants of GNNs are typically the aggregator that gathers information about the neighborhood of a node and a particular updater to update the hidden states. [17] In general, there is a certain functional evaluation of a node's features together with the features of its neighbors (convolution). The results are then the input for the neural network. GCN models are usually classified as spectral or spatial approaches. Spectral approaches perform an eigendecomposition of the graph Laplacian and operate on spectral graph representations. [17] Whereas spatial methods work directly on the graph structure and consider the local neighborhood of nodes [17], which lets them appear as the more intuitive approaches. A simple schematic representation of a deep GCN is depicted in fig. 2 where we have an initial graph as input that is sequentially processed through the hidden layers and finally results in an output representation of the graph.

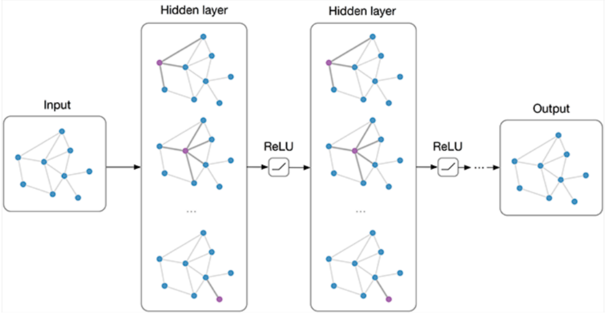

Figure 2: Multi-layer GCN. [9]

2.2 Robustness

Besides the repeatedly demonstrated good performance, there is one major issue that is subject to a rather new branch of research inside the field of GNNs, which is the robustness of such models. There are several publications that analyze the robustness of GNNs to adversarial examples and recently there came up first approaches to strengthen or even to certify their robustness with regard to a certain perturbation set.
Graph neural networks have lately reached state-of-the-art performance in recommendation systems. [16] Ying et al. developed a large-scale recommendation engine and deployed it at a major tech company. This example illustrates the need for robust GNNs. Although these systems may be relatively rare in practical applications, especially at that scale, at the moment, their performance suggests that this will change soon. However, in order to be able to use such models with a clear conscience in practical applications requires a certain degree of robustness. At the moment, there are noteworthy concerns about the security of using GNNs in safety-critical applications as they are vulnerable to adversarial attacks. [8]

Adversarial Perturbations

A well-studied problem of machine learning models in general is their sensitivity to adversarial perturbations. [5] The idea of such perturbations, which is visualized in fig. 3, is that slight changes to the input data cause an entirely different output of the model and therefore often misclassification.

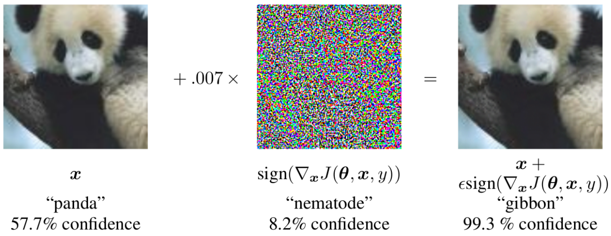
+ .007 ×
=
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Figure 3: Example for a simple adversarial perturbation. [5]

Goodfellow et al. [5] face that problem by adversarial training, which means that they include adversarially perturbed examples into the training process in order to strengthen the robustness of the models. Furthermore, they introduce efficient ways of generating adversarial examples, as shown in fig. 3. Although the added vector in fig. 3 is imperceptibly small, it suffices to change the classification of the image by the well-known *GoogleNet*. [5]
Adversarial perturbations are not only a problem for classical machine learning models, but also for GNNs. Numerous publications confirm the non-robustness of graph neural networks by showing that the models are prone to adversarial attacks on the node attributes as well as on the graph structure. [22] Similarly to the example in fig. 3, simple perturbations to GNN models lead to misclassification. As depicted in fig. 4, there are two types of perturbations that can cause misclassification, namely perturbations of the graph structure and perturbations of the node attributes.

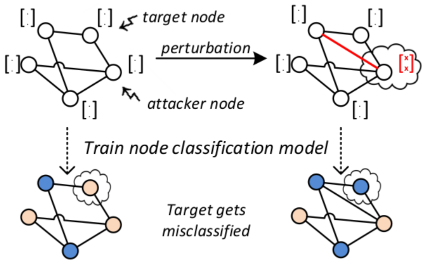

Figure 4: Simple structure / node attribute perturbations lead to misclassification. [19]

3. LITERATURE REVIEW

This section will provide an overview of the current state of research in the domain of robust GNNs, which could be divided roughly into three important phases. The first phase was to show that GNNs are indeed vulnerable to adversarial perturbations of the graph structure and the node attributes (cf. 3.1). Afterwards, in the second phase, several publications introduced defense mechanisms against such attacks or novel training procedures to strengthen the robustness of the models in some scenarios (cf. 3.2). Only recently, the third phase began, in which first approaches appeared that are able to not only provide mitigations to adversarial attacks in some scenarios, but to give provable guarantees about the (non-)robustness of a model, which is crucial to use them in safety-critical applications in the real world (cf. 3.3). Because of the relevance of the last phase, which is an important step in the process of bringing the convincingly performing GNN models into the real world, the main focus of the following sections will be on publications from that phase.

3.1 Vulnerability of GNNs to Adversarial Perturbations

Dai et al. [3] address adversarial attacks on graph structured data, in which the combinatorial structure of the data is altered in order to manipulate the outcome. They are able to show the vulnerability of GNN models to these kind of attacks in graph-level as well as node-level classification tasks using real-world and artificial data.
Another approach for adversarial attacks on graph structured data is proposed by Zügner et al. [19] who focus on node classification via graph convolutional networks. The authors claim that specifically areas in which GNNs are applied, such as websites, are often target of adversarial attacks and that it is therefore important to investigate the robustness of such models. They study adversarial attacks on attributed graphs and distinguish between adversarial attacks on the node's features and the graph structure. The results suggest that even slight perturbations significantly deteriorate the classification accuracy, which clearly motivates reflections on the robustness of such models, especially when considering practical applications.
So far, adversarial attacks were introduced as a quite vague concept. Zügner et al. [21] define them as "*small deliberate perturbations of data samples in order to achieve the outcome desired by the attacker*" and propose different categories to be considered in the attack model. Furthermore, they confirm previous claims that even slight perturbations of the graph structure or the node attributes consistently decrease the performance of GCN models.

3.2 Defense Mechanisms

Since the fact that GNNs are vulnerable to adversarial perturbations was confirmed by many publications, the next natural question to ask is how to defend them against such attacks. The defense mechanisms based on adversarial samples for classical neural networks are not directly and especially not efficiently extensible to GCNs. [7] Jin et al. tackle the problem and try to increase the efficiency by perturbations of the hidden representations of GCNs, which means that there is no need to generate adversarial networks and therefore improved efficiency. Additionally, they state that their framework improves robustness and accuracy in experiments on several tasks such as node classification, link prediction, and recommendation systems. When considering the robustness of GNNs, it is of course quite important to have efficient and effective attack methods to test with. Xu et al. [15] are proposing an effective gradient-based attack for graph neural networks that causes decreased classification performance. As a training procedure for higher robustness, they introduce an optimization-based adversarial training that is not impairing the classification accuracy. Zhu et al. [18] develop an approach that they refer to as *Robust GCN* (RGCN), which is aiming at increased robustness against adversarial attacks. To be able to simply absorb the effects of adversarial perturbations, they model the latent structures in each layer of the GCN as being Gaussian distributed with the idea that the perturbations perish in the variances of the distributions. According to these variances, the neighborhoods of nodes are assigned different weights to prevent adversarial attacks from being propagated through the net. The presented experimental evaluation suggests that the goal of increased robustness is achieved. Another approach to enhance the robustness of GCNs is proposed by Chen et al. [2] and relies on the idea that a crucial aspect of adversarial attacks on graph structures is given by manipulations of the edges of the graph. Thus, they focus on mitigating the sensitivity of the models to such manipulations by a training procedure, in which random, sparse and deformed subgraphs are used by a removal of certain edges. Experimentally, they can show that the method is indeed mitigating the sensitivity to adversarial attacks and thereby increasing the robustness. The idea, the approach of Jin et al [8] is based on, is that a reasonable defense method against graph perturbations lets the perturbed version of the graph undergo some kind of cleaning procedure. The authors state that graph structures representing problems from the real world often have particular common properties such as low-rankedness, sparsity, and a certain similarity between neighboring nodes that are presumably not shared by adversarially perturbed graphs. Therefore, such properties could be used to identify adversarial attacks. Based on that idea, they propose the framework *Pro-GNN*, which leads to a robust GNN model with remarkable performance in experiments with real-world data and extensive perturbations. Finally, Wang et al. [14] present with *GraphDefense* an algorithm to improve the robustness of graph convolutional networks against adversarial attacks with a particular focus on scaling for large graph inputs. Moreover, crucial characteristics of defense methods in general are discussed to improve the robustness.

3.3 Certifiable Robustness

When considering safety-critical applications, it is not only required to be able to defend the system in some scenarios, but there have to be guarantees about the security of the system. Bojchevski et al. [1] introduce a first method that is able to verify certifiable robustness or non-robustness of a general class of models including GNNs against perturbations of the graph structure. An important aspect of approaches that aim to make GNNs more robust is that they should not compromise the predictive accuracy of the models. The authors satisfy this constraint with their training process that leads to a larger fraction of nodes that are provably robust and is even able to improve the predictive accuracy in some cases. However, the approaches are restricted in the sense that they are not applicable to GCNs, but only to a certain class of models based on *PageRank*. [23] Zügner et al. [22] develop a first method that deals with certifiable (non-)robustness of GCNs against perturbations of the node attributes. The robustness certificates are always given under the assumption of a certain attack model and do not hold in general. Hence, a node that is certified as being (non-)robust by their method is guaranteed to be (non-)robust under every possible perturbation that is part of the defined attack model. Adhering to the aforementioned requirement of not, or at least only minimally affecting the predictive accuracy, the authors present a robust, semi-supervised training method. Recently, Zügner et al. [23] tackle the related problem of GCNs against perturbation of the graph structure and introduce a method for certifying their robustness. The authors reformulate the problem in a way that enables them to use specially optimized LP-solvers such as CPLEX to solve it. To achieve that, they express the problem as a jointly constrained bilinear program, develop a branch-and-bound procedure to obtain lower bounds on the optimal solution value and are able to divide the problem into linear sub-problems. Finally, Wang et al.[13] extend the so-called *randomized smoothing* technique to provide robustness guarantees for any GNN for node classification as well as graph classification problems. As Zügner et al. [23], they consider perturbations of the graph structure.

4. CERTIFIABLE ROBUSTNESS OF GRAPH NEURAL NETWORKS

As seen in the previous sections, graph neural networks are not at all robust against adversarial perturbations of the graph structure or the node attributes. This section will provide an overview of the crucial and only recently started third phase of research in the domain of robust GNNs, which aims to give provable guarantees about the (non-)robustness of a model. Only such guarantees enable the use of GNNs in safety-critical real-world applications.

A first work on certifiable (non-)robustness of GNNs against perturbations of the graph structure is proposed by Bojchevski et al. [1]. They are able to compute certificates very efficiently and, as mentioned in the previous section, develop a training method that leads to a higher number of nodes that are provably robust without negatively affecting the prediction accuracy. However, due to the restriction to a class of models based on *PageRank* [23], the approach will not be further discussed in this work.

After a somewhat deeper consideration of the approaches of Zügner et al., providing first robustness certificates for GCNs with respect to perturbations of the node attributes in section 4.1 and the graph structure in section 4.2, a brief overview of the latest work on certifiable robustness of GNNs against adversarial structural perturbations is provided in section 4.3.

4.1 Robustness Certification for Node Attribute Perturbations

In this section, the approach of Zügner et al. [22] will be described, in which the authors develop a first method that deals with certifiable (non-)robustness of GCNs against perturbations of the node attributes. Since the robustness certificates are always given under the assumption of a certain attack model, a node that is certified as being robust by their method is guaranteed to not change its prediction after every perturbation that is part of that attack model. Moreover, they are able to provide non-robustness certificates. As mentioned in section 3.3, it is always a major concern to not negatively affect the predictive accuracy while establishing a strengthened robustness of the model. With the presented semi-supervised training procedure, the authors achieve that goal. As a general issue when considering the robustness of GNNs, the work identifies the large space of possible perturbations that results from the fact that the perturbation of a single node can also have an effect on the predictions of other nodes in the graph. Accordingly, the perturbation budget gets constrained locally and globally. The approaches outlined in the work are mainly centered around the idea of estimating the worst-case change in the GNN's prediction according to the defined attack model (restricted space of perturbations). The graph neural network model is considered to be robust if that worst-case change is small. To be able to compute the worst-case change efficiently, the authors do not use the exact change value, but provide bounds on the value and additionally propose relaxations for the GNN model and the considered space of perturbations (attack model). Finally, they conduct an experimental evaluation of the approach that leads them to the conclusion that traditionally trained GNN models lack robustness and that the suggested training procedure offers a remedy by establishing a significant degree of robustness. The main question of the work can be subsumed as:

> *"How to make sure that small changes to the input data do not have a dramatic effect to a GNN?"* [22]

Because of their great importance for the rest of the work, the following two major concepts of [22] are briefly outlined again.

Robustness Certificates

> *"Given a trained GNN, we can give robustness certificates that state that a node is robust w.r.t. a certain space of perturbations. If the certificate holds, it is guaranteed that no perturbation (in the considered space) exists which will change the node's prediction. Furthermore, we also provide non-robustness certificates that, when they hold, state whether a node is not robust; realized by providing an adversarial example."* [22]

Robust Training

> *"We proposes a learning principle that improves the robustness of the GNN (i.e. making it less sensitive to perturbations) while still ensuring high accuracy for node classification."* [22]

4.1.1 Certifying the Robustness of a given GNN

The main part of [22] starts with the search for an efficient way of robustness certification. First, the goal is to certify the robustness of an already trained GNN model. When considering a node t in the graph, the idea is to be able to assure that the prediction of t does not change after feasible perturbations, i.e. to assure that it is robust. In an L-layered network, the output of a node t is only based on its neighbors in the $(L-1)$-hop-neighborhood. [22] This fact allows them to improve the scalability of the approach by reducing the computational effort to get the output of target node t. In particular, the size of the neural network itself and the size of potential perturbations are reduced. Based on that sliced version of the GNN, they formally define the problem to be solved, which is to check whether the prediction of a node t can not be altered by any feasible perturbation. The problem is defined as follows.

Let $G = (A, X)$ be a given graph, t a target node, and the parameters of the model are denoted as θ. A is the adjacency matrix and X represents the features of the nodes. The sliced versions of A and X are denoted as $\dot{A}$ and $\dot{X}$ and y^* defines the label of t that is either given or predicted. One important aspect, the authors emphasize, is that it is crucial that the generated certificates reflect realistic attacks. The attack model, i.e. the set of feasible perturbations, is restricted by a certain perturbation budget which sets a limit onto the number of L_0-measured changes to the attributes of the nodes $\dot{X}$. The perturbations are limited locally as well as globally, because in a scenario with a graph as data structure, a node can not only be directly attacked by changing its attributes, but also indirectly by altering attributes of nodes in its $(L-1)$-hop-neighborhood. [22] Therefore, a global perturbation budget $Q \in \mathbb{N}$ as well as a local budget $q \in \mathbb{N}$ is introduced. The worst case margin m^t between the classes y^* and y for a node t achievable under a certain set of feasible perturbations to the node features is defined as:

$$m^t(y^*, y) := \min_{\tilde{X}} f_\theta^t(\tilde{X}, \dot{A})_{y^*} - f_\theta^t(\tilde{X}, \dot{A})_y \qquad (1)$$
$$s.t. \quad \tilde{X} \in X_{q,Q}(\dot{X})$$

where $f_\theta^t(\tilde{X}, \dot{A})$ represents the output of the sliced GNN and $X_{q,Q}(\dot{X})$ is the set of feasible perturbations to the node features. If $m^t(y^*, y) > 0 \,\forall\, y \neq y^*$, the network is considered to be robust with regard to the node t and the set of perturbations $X_{q,Q}$, which means that there is no perturbation within the admissible set of perturbations that causes a prediction change. [22] The high-level idea of considering the classification margin as indicator for robustness is depicted in the following figures. Fig. 5 shows a situation with an unperturbed graph and a positive classification margin.

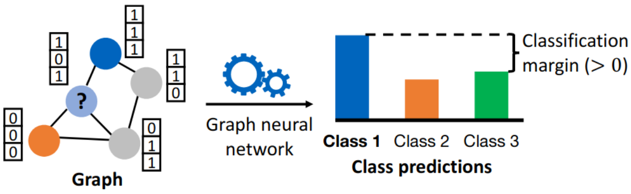

Figure 5: Classification margin before perturbation. [22]

Fig. 6, on the other hand, introduces two perturbations to the node attributes that lead to a negative classification margin and therefore not to certifiable robustness.

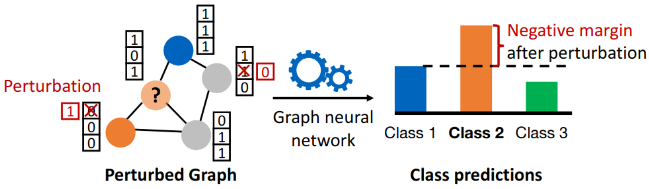

Figure 6: Classification margin after perturbation. [22]

To be able to solve the optimization problem in 1, the authors have to find a way of treating the challenging problems of a discrete data domain together with a nonconvex activation function in f_θ^t. [22] They tackle those issues by efficiently computing lower bounds on the minimum of the problem in 1 via relaxations of the GNN and the data domain. The idea is that a positive lower bound indicates the robustness of the classifier under the defined set of feasible perturbations. To end up with a convex objective function in the optimization problem in 1, a relaxation of the ReLU is required. [22] For such a relaxation, there are several methods published in the literature and the one, the authors follow, leads to a linear objective function which determines a lower bound for the optimal value of the initial problem. Although they end up with a linear problem (all constraints are linear as well), which can be solved with a specially optimized LP-solver, a possibly huge set of variables in such a model that would reduce the efficiency, lets them consider a different approach. Since every feasible solution of the dual problem provides a lower bound for the optimal solution of the primal problem, the dual of the LP is considered. [22] The authors summarize the idea as follows:

> *"[...] if we find any dual-feasible solution for which the objective function of the dual is positive, we know that the minimum of the primal problem has to be positive as well, guaranteeing robustness of the GNN w.r.t. any perturbation in the set."* [22]

Since it is not necessary to obtain an optimal solution for the dual problem (it suffices to find a feasible solution), the robustness certificates can be computed in a very efficient manner. As seen, the robustness of a node can be certified by positive dual values for each $y \neq y^*$. In contrast, to provide certificates for non-robustness, one has to consider the primal values that have to be negative for one $y \neq y^*$. [22] However, it is not guaranteed that each node falls into one of the two cases, which leaves certain nodes that can not be certified. The authors highlight that reasonable (tight) upper and lower bounds are of great importance for robustness certification based on their method and that those bounds can be used for backpropagation in the robust training procedure that will be presented in section 4.1.2.

The certification of robustness for trained GNNs is of course a very useful tool for practical applications, but, as the authors claim, the next step would be to achieve provable robustness of a model via training, which will be considered in the following section.

4.1.2 *Robust Training*

As seen in the previous section, the dual problem yields lower bounds on the classification margin between the labels y and y^* for node t. The authors define a k-dimensional vector p_θ^t that contains the solutions of the dual program for a label k compared to the label y. The node t can be determined as being certifiably robust if $p_\theta^t < 0$ for all entries except the one for y^*, for which $y^* = 0$ holds. [22] As a starting point for a training objective serves the following common objective function for classification tasks:

$$\min_{\theta} \sum_{t\in\mathcal{V}_L} \mathcal{L}(f_\theta^t(\dot{X},\dot{A}), y_t^*) \tag{2}$$

where $\mathcal{V}_L$ is the set of labeled nodes, $\mathcal{L}$ is the cross entropy function and y_t^* represents the label of node t that is either given or predicted. It is only a starting point, because they instead build on an objective from the literature that they call *robust cross entropy loss*, which leads to an increased robustness of classical neural networks and provides an upper bound on the worst-case loss [22]:

$$\min_{\theta,\{\Omega^{t,k}\}_{t\in\mathcal{V}_L,1\le k\le K}} \sum_{t\in\mathcal{V}_L} \mathcal{L}(p_\theta^t(y_t^*,\Omega^{t,\cdot}), y_t^*) \tag{3}$$

To prevail the classical problem of overconfidence and to establish guaranteed robustness, the authors introduce a loss function that they call *robust hinge loss*:

$$\hat{\mathcal{L}}_M(p,y^*) = \sum_{k\neq y^*} \max\{0, p_k + M\} \tag{4}$$

The node t is certifiably robust and guaranteed to have a margin $\geq M$ to the decision boundary if $\hat{\mathcal{L}}_M(p,y^*) = 0$ in equation 4. [22] In the final objective function in 5, the authors merge their *robust hinge loss* with the cross entropy function:

$$\min_{\theta,\Omega} \sum_{t\in\mathcal{V}_L} \hat{\mathcal{L}}_M(p_\theta^t(y_t^*,\Omega^{t,\cdot}), y_t^*) + \mathcal{L}(f_\theta^t(\dot{X},\dot{A}), y_t^*) \tag{5}$$

The advantage compared to the *robust cross entropy loss* is that the cross entropy function works with the exact version of the GNN. [22] Therefore, the authors only use the relaxed version to assure robustness while keeping the exact version for node prediction. In case every node is robust, $\hat{\mathcal{L}}_M = 0$ holds and the whole objective reduces to the simple cross entropy loss on the original version of the GNN. [22] So far, only the labeled nodes $\mathcal{V}_L$ are considered in 5. To bridge the gap to the semi-supervised setting, the authors also provide robustness guarantees for unlabeled nodes by extending the *robust hinge loss* as depicted in 6. The objective in 6 is based on the idea of using the exact version of the network for classification while every node is guaranteed to have a margin $\geq M$ from the decision boundary against every possible perturbation that is part of the defined attack model. [22] The second margin M_2 is used for the unlabeled nodes $\mathcal{V}\backslash\mathcal{V}_L$.

$$\begin{aligned}\min_{\theta,\Omega} \sum_{t\in\mathcal{V}_L} \hat{\mathcal{L}}_{M_1}(p_\theta^t(y_t^*,\Omega^{t,\cdot}), y_t^*) + \mathcal{L}(f_\theta^t(\dot{X},\dot{A}), y_t^*) \\ + \sum_{t\in\mathcal{V}\backslash\mathcal{V}_L} \hat{\mathcal{L}}_{M_2}(p_\theta^t(\tilde{y}_t,\Omega^{t,\cdot}), \tilde{y}_t)\end{aligned} \tag{6}$$

The authors conclude that due to the differentiability of the dual program and the activation bounds, the approach enables the training of a robust GNN using standard software.

4.1.3 *Experimental Evaluation and Conclusion*

The work ([22]) ends with an experimental evaluation of the approaches on the common data sets CORA-ML, CITESEER, and PUBMED. First, the robustness of traditionally trained GNNs is evaluated using the presented certification approach (cf. 4.1.1) and afterwards, it is shown that the robust training procedure (cf. 4.1.2) significantly improves the robustness of GNNs without strong negative effects on the predictive accuracy. As one would expect, the labeled set of nodes exhibits on average a higher robustness as the unlabeled one. The following important observations are highlighted by the authors:

1. Many of the provided certificates are tight.
2. There is only a small fraction of nodes for which no certificate can be provided.
3. Traditionally trained GNNs are only certifiably robust against negligible perturbations.

Moreover, they analyze which properties are involved in making nodes robust or non-robust and identify neighborhood purity as a key factor. They are observing a trade-off that says that many neighbors imply a large surface for potential attacks, while nodes with a lower degree are less stable and therefore stronger affected by individual neighboring attacks. Regarding the robust training, they can show that their method significantly increases the number of robust nodes. If the perturbation budget for which the model was trained is considered, almost all the nodes are certifiably robust. [22] Additionally increased is the general amount of certifiable nodes. There are even examples for nodes that have been certified as being non-robust before, that become certifiably robust after the procedure.

Conclusion

The work presented in this section introduces the first methods for provable robustness of graph neural networks against perturbations of node attributes. A major focus of the work is about being able to compute robustness certificates in an efficient manner, which is achieved by several relaxations and a well justified consideration of dual feasible solutions instead of the optimal solution values. Another fundamental result provided by Zügner et al. [22] is the fact that traditionally trained graph neural networks are not robust and that the robust training procedure presented in the work is indeed able to significantly strengthen the robustness of GNN models. The perhaps most important aspect is that the strengthened robustness does not have a significant negative effect on the predictive accuracy. As future work, they already bridge to the next section which considers perturbations of the graph structure.

4.2 Robustness Certification for Graph Structure Perturbations

As announced in the previous section, the next step is to consider perturbations of the graph structure, which means that the graph structure itself can be modified by an attacker. In this section, the approaches of Zügner et al. [23] will be described, in which the problem of certifiable robustness of GCNs against structural perturbations is tackled. Since these kind of perturbations attack, and thus modify, the message passing scheme, they are considered to be particularly challenging. [23] The authors reformulate the problem as a jointly constrained bilinear program and develop a branch-and-bound procedure to provide lower bounds on the optimal solution value. The attack model that is used in the work is based on the idea that an attacker is able to insert new edges to the graph. Such a scenario could for example occur in web-based applications where edges represent friends, likes or follows in social media. [23] The high-level idea, which is visualized in fig. 7, is that there is a clean graph that is reachable by removing these adversaries that could have changed the model's prediction. The authors summarize the idea of their certificates as follows:

> *"A certificate issued by our method states that a node's prediction could not have been altered by edges potentially inserted by an attacker."* [23]

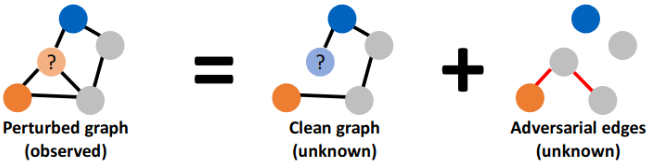

Figure 7: High-level idea of Zügner et al. [23]

There are three major challenges identified in the work:

1. The search for optimal perturbations is not efficient due to the general nonconvexity of neural networks.
2. Enumerating the set of feasible perturbations is of course not efficiently possible as well, because the cardinality of the set is growing exponentially with the perturbation budget.
3. In contrast to the certificates of the approach in section 4.1, here the changed structure could change the propagation of embeddings.

The first challenge is also a problem for traditional neural networks and the approach already used in section 4.1, which relaxes the ReLU activation function, is applied again. [23] For the second challenge, the authors can not recycle another approach, but have to come up with new solutions. They show that a continuous relaxation, as used for the binary attributed data in section 4.1, is not applicable for attacks on the graph structure and instead work directly on the degree-normalized message passing matrix of the GCN which already consists of continuous values. Another unstudied question is posed by the third challenge that requires a way to handle a modified structure (message passing scheme). [23] The authors see the problem in the fact that the message passing scheme, i.e. the matrix, in contrast to the input features considered in section 4.1, are used in every layer of the network and not only in the input layer. That causes nonconvex constraints after the ReLU relaxations. [23] As mentioned, the authors reformulate the problem as a jointly constrained bilinear program and develop a branch-and-bound procedure to provide lower bounds on the optimal solution value. The lower bounds can be seen as worst-case change for the prediction of a node and are therefore used as robustness certificates where a positive value indicates that the prediction of a node will not change when being exposed to a feasible attack. [23]

4.2.1 Robustness Certification Technique

The problem is outlined as a semi-supervised node classification task in a graph $G = (A, X)$ with $D-$dimensional node attributes where $A \in \{0,1\}^{N \times N}$ is the adjacency matrix and $X \in \mathbb{R}^{N \times D}$ are the attributes of the nodes. Just as in section 4.1, $V_L \subseteq V$ is the subset of labeled nodes with labels $C = \{1, 2, ..., K\}$. Also as in the previous section, θ summarizes the trainable weights of the GNN that can be learned using the labeled nodes V_L by minimizing the cross entropy loss. [23] The message passing matrix that defines how activations are propagated through the GNN can be obtained by a defined transformation $\mathcal{T}(A)$ to the adjacency matrix. [23] As always in classification tasks, the aim is to assign labels to the nodes. The network's output $H^{(L)}_{vc}$ is the probability of assigning label c to node v. [23]
At first, the authors define the optimization problem that is to be solved. The starting point is an already trained GNN with weights θ. The input graph that is represented by the adjacency matrix A may already be perturbed. At this point, the main idea, visualized in fig. 7, comes into play, which is that the graph represented by A can be reached from the "clean" unperturbed version of the graph

represented by an adjacency matrix A^* by a set of feasible perturbations against the graph structure, e.g. adversarial edges. [23] The authors guarantee the robustness for a single node at a time, i.e. they assure that the specific node under consideration is not changing its prediction after a feasible attack. The worst-case margin between y^* and y is computed as the following optimization problem where y^* defines the given / predicted label of node t and $\mathcal{A}(A)$ defines the set of feasible perturbations.

$$m^t(y^*, y) := \min_{\tilde{A}} f^t_\theta(X, \mathcal{T}(\tilde{A}))_{y^*} - f^t_\theta(X, \mathcal{T}(\tilde{A}))_y \qquad (7)$$
$$s.t. \tilde{A} \in \mathcal{A}(A)$$

The GNN is claimed to be robust under the set of feasible attacks for the node t in question, if $m^t(y^*, y) > 0 \, \forall \, y \neq y^*$. Therefore, just as for the optimization problem in section 4.1, if the minimum in 7 is positive, there is no feasible perturbation to the graph structure that would cause a prediction change of node t. [23]
Now we get back to the previously mentioned challenges that prevent an easy and efficient optimization of the problem in 7. The authors are able to bypass the hard, or inefficient part of the problem by again computing lower bounds instead of optimal values. More precisely, they perform the following three steps to compute lower bounds on the original problem that allow to answer the question of robustness:

- Replace the binary adjacency matrix A by the continuous message passing matrix $\mathcal{T}(A)$.
- Relax the activation function of the GNN.
- Rephrase the problem as jointly constrained bilinear program and solve it with branch-and-bound.

A positive lower bound indicates robustness under the defined attack model. [23]

Optimization over the Graph Structure

In order to be able to cope with the second challenge (cf. 2), the set of feasible perturbations is restricted in such a way that on the one hand the sheer size of the set is limited and on the other hand it reflects perturbations that could actually occur in practical applications. [23] The authors again use the L_0 norm, in this case to measure the number of perturbed elements in A. To do that, they transfer the idea of local and global L_0 constraints on the node attributes used in the previous paper discussed in section 4.1 to structural perturbations, which means that there are allowed to be at most $q_i \in \mathbb{N}$ local adversarial edges to a specific node i and at most $Q \in \mathbb{N}$ global adversarial edges inserted into the whole graph. [23] This approach refers to the high-level idea visualized in fig. 7, i.e. that the perturbed graph structure differs from the original (clean) structure by a certain set of inserted adversarial edges.
Continuous relaxation would be the common way of dealing with intractable discrete optimization problems, but in this case, due to the minimization over $\tilde{A}$ in the objective function, this is not an option. [23] Instead, the authors propose to use the degree-normalized message passing matrix $\hat{A}$ in its continuous nature. As a result, the optimization problem in 7 is replaced by the following problem, which not only avoids the intractable optimization over a discrete variable, but also the nonconvex degree normalization process since the message passing matrix $\hat{A}$ is now directly passed to f^t_θ as input. [23]

$$\hat{m}^t(y^*, y) := \min_{\hat{A}} f^t_\theta(X, \hat{A})^*_y - f^t_\theta(X, \hat{A})_y \qquad (8)$$
$$s.t. \hat{A} \in \hat{\mathcal{A}}(A)$$

The set $\hat{\mathcal{A}}(A)$ of possible message passing matrices has to be defined in a way that it contains every possible matrix that could result from first performing discrete perturbations to A and afterwards degree-normalizing the resulting binary matrix $\tilde{A}$, only then it holds that $\hat{m}^t(y^*, y) \leq m^t(y^*, y)$, i.e. the result of 8 provides a lower bound on the optimal solution of the optimization problem in 7. [23] The authors are able to derive constraints to generate message passing matrices $\hat{A}$ that are valid, tight, and convex to enable efficient optimization. Finally, a positive value of 8 indicates a certificate. [23]

Relaxation of the Neural Network

As stated in the first challenge (cf. 1), the objective function of the GNN is hard due to the non-convexity. Therefore, the authors again apply a relaxation approach from their previous paper [22], which causes the output H of the ReLU activation function to no longer be deterministic, but considered to be a variable and combine it with ideas of another approach from the literature.

Jointly Constrained Bilinear Program

Combining the relaxed GNN with the constraints on the message passing matrix, the objective function in 8 can be reformulated as a bilinear objective function. [23] Afterwards, they add another artificial constraint that does not change the solution, but enables them to use a principle proposed in the literature, which allows them to solve the jointly constrained bilinear optimization problem. A concept, referred to as "convex envelope" of the objective function, is used to compute lower bounds and additionally, a branch-and-bound procedure is used to recursively divide the problem into sub-problems. In each iteration, the upper and lower bounds become more and more precise and the idea is to stop as soon as possible. It is important to note that it is not necessary to find the optimal solution of the bilinear problem, only its sign is of relevance. [23] Consequently, there are two possible outcomes: A positive lower bound indicates successfully certified robustness, while a negative lower bound only states that no robustness certificate can be given and not non-robustness. [23]

4.2.2 *Experimental Evaluation and Conclusion*

As in the previous paper presented in section 4.1, an experimental evaluation of the approaches on the benchmark data sets CORA-ML, CITESEER, and PUBMED is provided. The authors evaluate the proposed robustness certification technique against attacks on the graph structure based on three different local q and global Q perturbation budgets with which they perform gradient-based attacks. This approach enables them to estimate the number of non-robust nodes, although the computation of the precise number would be intractable. [23] As one would expect, higher perturbation budgets cause fewer nodes that can be considered as being certifiably robust and more non-robust nodes. [23] Another hint to the general non-robustness of GCN models is given by the fact that it suffices to remove one edge from the graph to cause predictive changes in a significant portion of the nodes. [23] A further crucial observation is that "classical" adversarial training procedures based on the idea of adding adversarially perturbed examples to the training data, such as the perturbed graph in fig. 7, are not leading to a higher robustness of the model [23], which confirms the relevance of the work. Only the training for robustness against perturbations of the node attributes, that is considered in section 4.1, is able to do so.

In summary, the authors introduce first approaches for certifiable robustness against structural perturbations for graph convolutional networks. The main idea is to reformulate the problem as a jointly constrained bilinear program and to efficiently solve that using a branch-and-bound method. The branch-and-bound method recursively divides the problem into linear sub-problems that can be solved by specially optimized LP-solvers (e.g. CPLEX). For a large fraction of the nodes, the authors are able to provide a clear statement of robustness / non-robustness. Overall, the presented work provides a very efficient way of robustness certification.

4.3 Robustness Guarantee for Arbitrary GNNs for Node and Graph Classification

The presumably latest work on certifiable robustness of GNNs against adversarial perturbations is provided by Wang et al. [13] who consider robustness guarantees for arbitrary graph neural network models. The studied task is again a classification task. In this case, however, not only classification on the node level, but also on the graph level. Opposed to node classification, which labels a subset of unlabeled nodes based on a classifier trained on the graph's subset of labeled nodes, graph level classification assigns a label to the entire graph. [13] Therefore, the classifier has to be trained on a set of labeled graphs. An example where such a classification scheme could be applied is assigning labels to graphs based on their specific topology (e.g. circular graphs). Like the approaches of Zügner et al. [23] and Bojchevski et al. [1], they address certifiable robustness of GNNs against perturbations of the graph structure. In contrast to the previous papers described in section 4.1 and 4.2, which deal with specific GNN models, the work provides robustness guarantees for arbitrary GNNs. To achieve that, they make use of an extended version of the *randomized smoothing* method, which essentially adds noise to the elements of the training data set in order to establish robustness. [13] The authors extend the basic *randomized smoothing* such that they are able to handle non-continuous data structures, i.e. graphs. A certain *certified perturbation size* K is theoretically determined with the intention of setting a limit onto the number of adversarial edge additions or removals such that a GNN is considered to be robust if there is no change of prediction when the number of perturbations is $\leq K$. [13] Subsequently, the problem of practically computing the *certified perturbation size* K is formulated as an optimization problem and an algorithm is introduced that solves it in an efficient manner. The merits of the proposed approach are its scalability and the fact that it can be used with arbitrary classifiers. [13]

To summarize, Wang et al. extend the *randomized smoothing* approach to discrete (binary) data structures such as graphs and thereby enable robustness certification for arbitrary graph neural networks against structural perturbations. Finally, they evaluate their approach on various benchmark instances for node and graph classification tasks. For the evaluation of node classification, they use the benchmark data sets CORA-ML, CITESEER, and PUBMED, just as Zügner et al. in the sections 4.1 and 4.2.

5. CONCLUSION AND OUTLOOK

This paper gave a concise overview of the current state of research in the domain of graph (convolutional) neural networks with a focus on the robustness of the models. As seen throughout the sections, GNNs can be applied very successfully to a wide range of practical tasks. What has prevented its frequent use in practice so far is the fact that GNNs are not at all robust against adversarial perturbations of the graph structure and the node attributes, which has been proven by numerous publications discussed in section 3. As long as it is possible that small changes to the input data cause entirely different results, it is not reasonable to apply the models in any kind of safety-critical application or in areas where legal certainty must prevail.

Three main lines of research in the domain of robust GNNs have been identified in this work. In the first phase it was shown that GNNs are prone to adversarial attacks. The second phase introduced defense mechanisms against such attacks or training procedures to strengthen the robustness. However, these approaches are generally just providing mitigations for certain scenarios and not guaranteeing robustness in any case. The third phase, which is the main consideration of this work, discussed in section 4, only recently started to tackle this problem. Approaches appeared that are able to not only provide mitigations to adversarial attacks in some scenarios, but to give provable guarantees about the (non-)robustness of a GNN with regard to an attack model, which is crucial to use them in real-world applications.

Although the third phase only started recently, there have already been quite effective approaches for certified robustness presented in section 4. Nevertheless, there is of course a lot of room for improvement and generalization. Zügner et al. [23] for example suggest that it might be worthwhile to study training methods in which robustness is explicitly treated as the goal to be achieved. Another promising future prospect is given by Wang et al. [13] who suggest to extend their method briefly described in section 4.3 to not only certify robustness of GNNs against structural attacks, but to also include perturbations of the node attributes. This could be particularly promising for practical applications when considered in combination with the advantage of their algorithm in terms of scalability. In addition, they plan to optimize the method by including and exploiting certain information about a GNN during the process.

On a less concrete level, it would also be interesting to find out which aspects of perturbations are relevant for their harmfulness [20] and therefore truly develop an understanding of the fundamental structure of the problem. The approach of Jin et al. [8] goes in this direction in a way, although they are not considering certifiable robustness. Their work is rather belonging to the second phase, dealing with mitigations, as the proposed framework *Pro-GNN* enables a reasonable defense method in some scenarios. Nonetheless, they highlight an important observation, namely that graph structures representing problems from the real world often have particular common properties that are presumably not shared by adversarially perturbed graphs. Eventually, the approaches of the third phase outlined in this work assume a certain attack model under which robustness guarantees are provided. If a general understanding of the underlying mechanisms of harmful perturbations can be established, there could be room for generalizations of attack models to guarantee robustness for.

Lately, Zügner et al. [20] provide a first attempt of recognizing adversarial attacks against graph neural networks by their structural properties. They study statistically significant structural attributes in perturbations against GNN models. Based on these, they develop an algorithm that is able to generate perturbations that are not only harmful against specific examples, but in general. The authors interpret their results as a crucial milestone towards the goal of being able to identify and hinder adversarial attacks and thereby increasing the robustness of GNN models. Besides the approach that uses statistically significant properties of harmful perturbations, they also establish a ranking scheme that provides probabilities for the harmfulness of perturbations. The work is an auspicious starting point for understanding the underlying mechanisms of adversarial attacks, which could ultimately lead to methods using general attack models that ideally are able to recognize incoming attacks based on their specific characteristics.

6. REFERENCES

[1] Bojchevski, A., and Günnemann, S. Certifiable robustness to graph perturbations. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, December 8-14, Vancouver, BC, Canada* (2019), H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., pp. 8317–8328.

[2] Chen, L., Li, X., and Wu, D. Enhancing robustness of graph convolutional networks via dropping graph connections. *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Ghent, Belgium, Sep* (2020).

[3] Dai, H., Li, H., Tian, T., Huang, X., Wang, L., Zhu, J., and Song, L. Adversarial attack on graph structured data. In *Proceedings of the 35th International Conference on Machine Learning, Stockholmsmässan, Stockholm, Sweden, July 10-15* (2018), J. G. Dy and A. Krause, Eds., vol. 80 of *Proceedings of Machine Learning Research*, PMLR, pp. 1123–1132.

[4] Duvenaud, D., Maclaurin, D., Aguilera-Iparraguirre, J., Gómez-Bombarelli, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems, December 7-12, Montreal, Quebec, Canada* (2015), C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., pp. 2224–2232.

[5] Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, San Diego, CA, USA, May 7-9, Conference Track Proceedings* (2015), Y. Bengio and Y. LeCun, Eds.

[6] Hamilton, W. L., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, December 4-9, Long Beach, CA, USA* (2017), I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., pp. 1024–1034.

[7] Jin, H., and Zhang, X. Robust training of graph convolutional networks via latent perturbation. *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Ghent, Belgium, Sep* (2020).

[8] Jin, W., Ma, Y., Liu, X., Tang, X., Wang, S., and Tang, J. Graph structure learning for robust graph neural networks. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27* (2020), R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, Eds., ACM, pp. 66–74.

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

[10] Liu, Z., and Zhou, J. Introduction to graph neural networks. *Synthesis Lectures on Artificial Intelligence and Machine Learning 14* (03 2020), 1–127.

[11] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE Trans. Neural Networks 20*, 1 (2009), 61–80.

[12] Trivedi, R., Dai, H., Wang, Y., and Song, L. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *Proceedings of the 34th International Conference on Machine Learning, Sydney, NSW, Australia, 6-11 August* (2017), D. Precup and Y. W. Teh, Eds., vol. 70 of *Proceedings of Machine Learning Research*, PMLR, pp. 3462–3471.

[13] Wang, B., Jia, J., Cao, X., and Gong, N. Z. Certified robustness of graph neural networks against adversarial structural perturbation. *CoRR abs/2008.10715* (2020).

[14] Wang, X., Liu, X., and Hsieh, C. Graphdefense: Towards robust graph convolutional networks. *CoRR abs/1911.04429* (2019).

[15] Xu, K., Chen, H., Liu, S., Chen, P., Weng, T., Hong, M., and Lin, X. Topology attack and defense for graph neural networks: An optimization perspective. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, August 10-16* (2019), S. Kraus, Ed., ijcai.org, pp. 3961–3967.

[16] Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018* (2018), Y. Guo and F. Farooq, Eds., ACM, pp. 974–983.

[17] Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., and Sun, M. Graph neural networks: A review of methods and applications. *CoRR abs/1812.08434* (2018).

[18] Zhu, D., Zhang, Z., Cui, P., and Zhu, W. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, August 4-8* (2019), A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi, and G. Karypis, Eds., ACM, pp. 1399–1407.

[19] Zügner, D., Akbarnejad, A., and Günnemann, S. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, August 19-23* (2018), Y. Guo and F. Farooq, Eds., ACM, pp. 2847–2856.

[20] Zügner, D., Borchert, O., Akbarnejad, A., and Günnemann, S. Adversarial attacks on graph neural networks: Perturbations and their patterns. *ACM Trans. Knowl. Discov. Data 14*, 5 (2020), 1–31.

[21] Zügner, D., and Günnemann, S. Adversarial attacks on graph neural networks via meta learning. In *7th International Conference on Learning Representations, New Orleans, LA, USA, May 6-9* (2019), OpenReview.net.

[22] Zügner, D., and Günnemann, S. Certifiable robustness and robust training for graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, August 4-8* (2019), A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi, and G. Karypis, Eds., ACM, pp. 246–256.

[23] Zügner, D., and Günnemann, S. Certifiable robustness of graph convolutional networks under structure perturbations. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27* (2020), R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, Eds., ACM, pp. 1656–1665.