Certifiable Robustness to Graph Perturbations

Aleksandar Bojchevski
Technical University of Munich
a.bojchevski@in.tum.de

Stephan Günnemann
Technical University of Munich
guennemann@in.tum.de

Abstract

Despite the exploding interest in graph neural networks there has been little effort to verify and improve their robustness. This is even more alarming given recent findings showing that they are extremely vulnerable to adversarial attacks on both the graph structure and the node attributes. We propose the first method for verifying certifiable (non-)robustness to graph perturbations for a general class of models that includes graph neural networks and label/feature propagation. By exploiting connections to PageRank and Markov decision processes our certificates can be efficiently (and under many threat models exactly) computed. Furthermore, we investigate robust training procedures that increase the number of certifiably robust nodes while maintaining or improving the clean predictive accuracy.

1 Introduction

As the number of machine learning models deployed in the real world grows, questions regarding their robustness become increasingly important. In particular, it is critical to assess their vulnerability to adversarial attacks – deliberate perturbations of the data designed to achieve a specific (malicious) goal. Graph-based models suffer from poor adversarial robustness [13, 60], yet in domains where they are often deployed (e.g. the Web) [50], adversaries are pervasive and attacks have a low cost [9, 26]. Even in scenarios where adversaries are not present such analysis is important since it allows us to reason about the behavior of our models in the worst case (i.e. treating nature as an adversary).

Here we focus on semi-supervised node classification – given a single large (attributed) graph and the class labels of a few nodes the goal is to predict the labels of the remaining unlabelled nodes. Graph Neural Networks (GNNs) have emerged as the de-facto way to tackle this task, significantly improving performance over the previous state-of-the-art. They are used for various high impact applications across many domains such as: protein interface prediction [20], classification of scientific papers [28], fraud detection [44], and breast cancer classification [36]. Therefore, it is crucial to assess their sensitivity to adversaries and ensure they behave as expected.

However, despite their popularity there is scarcely any work on certifying or improving the robustness of GNNs. As shown in Zügner et al. [60] node classification with GNNs is not robust and can even be attacked on multiple fronts – slight perturbations of either the node features or the graph structure can lead to wrong predictions. Moreover, since we are dealing with non i.i.d. data by taking the graph structure into account, robustifying GNNs is more difficult compared to traditional models – perturbing only a few edges affects the predictions for all nodes. What can we do to fortify GNNs and make sure they produce reliable predictions in the presence of adversarial perturbations?

We propose the first method for provable robustness regarding perturbations of the graph structure. Our approach is applicable to a general family of models where the predictions are a linear function of (personalized) PageRank. This family includes GNNs [29] and other graph-based models such as label/feature propagation [7, 53]. Specifically, we provide: 1. Certificates: Given a trained model and a general set of admissible graph perturbations we can efficiently verify whether a node is certifiably robust – there exists no perturbation that can change its prediction. We also provide non-
robustness certificates via adversarial examples. 2. Robust training: We investigate robust training schemes based on our certificates and show that they improve both robustness and clean accuracy. Our theoretical findings are empirically demonstrated and the code is provided for reproducibility. Interestingly, in contrast to existing works on provable robustness [23, 46, 59] that derive bounds (by relaxing the problem), we can efficiently compute exact certificates for some threat models.

2 Related work

Neural networks [41, 21], and recently graph neural networks [13, 60, 58] and node embeddings [5] were shown to be highly sensitive to small adversarial perturbations. There exist many (heuristic) approaches aimed at robustifying these models, however, they have only limited usefulness since there is always a new attack able to break them, leading to a cat-and-mouse game between attackers and defenders. A more promising line of research studies certifiable robustness [23, 55, 40]. Certificates provide guarantees that no perturbation regarding a specific threat model will change the prediction of an instance. So far there has been almost no work on certifying graph-based models.

Different heuristics have been explored in the literature to improve robustness of graph-based models: (virtual) adversarial training [10, 16, 40, 49], trainable edge weights [48], graph encoder refining and adversarial contrastive learning [45], transfer learning [42], smoothing distillation [10], decoupling structure from attributes [31], measuring logit discrepancy [51], allocating reliable queries [56], representing nodes as Gaussian distributions [57], and Bayesian graph neural networks [52]. Other robustness aspects of graph-based models (e.g., noise or anomalies) have also been investigated [3, 6, 24]. However, none of these works provide provable guarantees or certificates.

Zügner & Günnemann [59] is the only work that proposes robustness certificates for graph neural networks (GNNs). However, their approach can handle perturbations only to the node attributes. Our approach is completely orthogonal to theirs since we consider adversarial perturbations to the graph structure instead. Furthermore, our certificates are also valid for other semi-supervised learning approaches such as label/feature propagation. Nonetheless, there is a critical need for both types of certificates given that GNNs are shown to be vulnerable to attacks on both the attributes and the structure. As future work, we aim to consider perturbations of the node features and the graph jointly.

3 Background and preliminaries

Let \( G = (\mathcal{V}, \mathcal{E}) \) be an attributed graph with \( N = |\mathcal{V}| \) nodes and edge set \( \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} \). We denote with \( A \in \{0, 1\}^{N \times N} \) the adjacency matrix and \( X \in \mathbb{R}^{N \times D} \) the matrix of \( D \)-dimensional node features for each node. Given a subset \( \mathcal{V}_L \subseteq \mathcal{V} = \{1, \ldots, N\} \) of labelled nodes the goal of semi-supervised node classification is to predict for each node \( v \in \mathcal{V} \) one class in \( \mathcal{C} = \{1, \ldots, K\} \). We focus on deriving (exact) robustness certificates for graph neural networks via optimizing personalized PageRank. We also show (Appendix 8.1) how to apply our approach for label/feature propagation [7].

**Topic-sensitive PageRank.** The topic-sensitive PageRank [22, 27] vector \( \pi_G(z) \) for a graph \( G \) and a probability distribution over nodes \( z \) is defined as \( \pi_{G,\alpha}(z) = (1-\alpha)(I_N - \alpha D^{-1} A)^{-1} z \).

[1] Here \( D \) is a diagonal matrix of node out-degrees with \( D_{ii} = \sum_j A_{ij} \). Intuitively, \( \pi(z) \) represent the probability of random walker on the graph to land at node \( u \) when it follows edges at random with probability \( \alpha \) and teleports back to the node \( v \) with probability \( 1-\alpha \). Thus, we have \( \sum_u \pi(z)\pi(z)u \geq 0 \) and \( \sum_u \pi(z)\pi(z)u = 1 \). For \( z = e_v \), the \( v \)-th canonical basis vector, we get the personalized PageRank vector for node \( v \). We drop the index on \( G, \alpha \) and \( z \) in \( \pi_{G,\alpha}(z) \) when they are clear from the context.

**Graph neural networks.** As an instance of graph neural network (GNN) methods we consider an adaptation of the recently proposed PPNP approach [29] since it shows superior performance on the semi-supervised node classification task [19]. PPNP unlike message-passing GNNs decouples the feature transformation from the propagation. We have:

\[
Y = \text{softmax}(\Pi^{ym} H), \quad H_{v \cdot} = f_\theta(X_{v \cdot}), \quad \Pi^{ym} = (1-\alpha)(I_N - \alpha D^{-1/2} A D^{-1/2})^{-1} \quad (1)
\]

where \( I_N \) is the identity, \( \Pi^{ym} \in \mathbb{R}^{N \times N} \) is a symmetric propagation matrix, \( H \in \mathbb{R}^{N \times C} \) collects the individual per-node logits, and \( Y \in \mathbb{R}^{N \times C} \) collects the final predictions after propagation. A

[1] Code, data, and supplementary material available at [https://www.daml.in.tum.de/graph-cert](https://www.daml.in.tum.de/graph-cert)

[2] In practice we do not invert the matrix, but rather we solve the associated sparse linear system of equations.
neural network $f_\theta$ outputs the logits $H_{v,c}$; by processing the features $X_v$: of every node $v$ independently. Multiplying them with $\Pi^\text{sym}$ we obtain the diffused logits $H_{\text{diff}} := \Pi^\text{sym}H$ which implicitly incorporate the graph structure and avoid the expensive multi-hop message-passing procedure.

To make PPNP more amenable to theoretical analysis we replace $\Pi$ with the personalized PageRank matrix $\Pi = (1 - \alpha)(I_N - \alpha D^{-1}A)^{-1}$ which has a similar spectrum. Here each row $\Pi_v := \pi(e_v)$ equals to the personalized PageRank vector of node $v$. This model which we denote as $\pi$-PPNP has similar prediction performance to PPNP. We can see that the diffused logit after propagation for class $c$ of node $v$ is a linear function of its personalized PageRank score: $H^\text{diff}_{v,c} = \pi(e_v)^T H_{v,c}$, i.e. a weighted combination of the logits of all nodes for class $c$. Similarly, the margin $m_{c_1,c_2}(v) = H^\text{diff}_{v,c_1} - H^\text{diff}_{v,c_2} = \pi(e_v)^T(H_{v,c_1} - H_{v,c_2})$ defined as the difference in logits for node $v$ for two given classes $c_1$ and $c_2$ is also linear in $\pi(e_v)$. If $\min_{c} m_{y_t,c}(v) < 0$, where $y_t$ is the ground-truth label for $v$, the node is misclassified since the prediction equals $\arg \max_c H^\text{diff}_{v,c}$.

4 Robustness certificates

4.1 Threat model, fragile edges, global and local budget

We investigate the scenario in which a subset of edges in a directed graph are "fragile", i.e. an attacker has control over them, or in general we are not certain whether these edges are present in the graph. Formally, we are given a set of fixed edges $E_f \subseteq E$ that cannot be modified (assumed to be reliable), and set of fragile edges $F \subseteq (V \times V) \setminus E_f$. For each fragile edge $(i,j) \in F$ the attacker can decide whether to include it in the graph or exclude it from the graph, i.e. set $A_{ij}$ to 1 or 0 respectively. For any subset of included $F_+ \subseteq F$ edges we can form the perturbed graph $\hat{G} = (V, \hat{E} := E_f \cup F_+)$. An excluded fragile edge $(i,j) \in F \setminus F_+$ is a non-edge in $\hat{G}$. This formulation is general, since we can set $E_f$ and $F$ arbitrarily. For example, for our certificate scenario given an existing clean graph $G = (V,E)$ we can set $E_f = E$ and $F \subseteq V \times V$ which implies the attacker can only add new edges to obtain perturbed graphs $\hat{G}$. Or we can set $E_f = \emptyset$ and $F = E$ so that the attacker can only remove edges, and so on. There are $2^{|F|}$ (exponential) number of valid configurations leading to different perturbed graphs which highlights that certificates are challenging for graph perturbations.

In reality, perturbing an edge is likely to incur some cost for the attacker. To capture this we introduce a global budget. The constraint $|{\hat{E} \setminus E}| + |E \setminus \hat{E}| \leq B$ implies that the attacker can make at most $B$ perturbations. The first term equals to the number of newly added edges, and the second to the number of removed existing edges. Here, including an edge that already exists does not count towards the budget. This is only a design choice that depends on the application, and our method works in general. Furthermore, perturbing many edges for a single node might not be desirable, thus we also allow to limit the number of perturbations locally. Let $\mathcal{E}^v = \{(v,j) \in E\}$ be the set of edges that share the same source node $v$. Then, the constraint $|{\hat{E}^v \setminus E^v}| + |E^v \setminus \hat{E}^v| \leq b_v$ enforces a local budget $b_v$ for the node $v$. By setting $b_v = |F|$ and $B = |F|$ we can model an unconstrained attacker. Letting $P(\mathcal{F})$ be the power set of $\mathcal{F}$, we define the set of admissible perturbed graphs:

$$Q_\mathcal{F} = \{(V, \hat{E} := E_f \cup \mathcal{F}_+) \mid \mathcal{F}_+ \in P(\mathcal{F}), |\hat{E} \setminus E| + |E \setminus \hat{E}| \leq B, |E^v \setminus \hat{E}^v| + |\hat{E}^v \setminus E^v| \leq b_v, \forall v\} \quad (2)$$

4.2 Robustness certificates

Problem 1. Given a graph $G$, a set of fixed $E_f$ and fragile $F$ edges, global $B$ and local $b_v$ budgets, target node $t$, and a model with logits $H$. Let $y_t$ denote the class of node $t$ (predicted or ground-truth). The worst-case margin between class $y_t$ and class $c$ under any admissible perturbation $\hat{G} \in Q_\mathcal{F}$ is:

$$m^*_{y_t,c}(t) = \min_{\hat{G} \in Q_\mathcal{F}} m_{y_t,c}(t) = \min_{\hat{G} \in Q_\mathcal{F}} \pi_{\hat{G}}(e_t)^T(H_{y_t} - H_{.,c}) \quad (3)$$

If $m^*_{y_t,c}(t) > 0$, node $t$ is certifiably robust w.r.t. the logits $H$, and the set $Q_\mathcal{F}$.

Our goal is to verify whether no admissible $\hat{G} \in Q_\mathcal{F}$ can change the prediction for a target node $t$. From Problem1 we see that if the worst margin over all classes $m^*_{y_t,c}(t)$ is positive, then $m^*_{y_t,c}(t) > 0$, for all $y_t \neq c$, which implies that there exists no adversarial example within $Q_\mathcal{F}$ that leads to a change in the prediction to some other class $c$, that is, the logit for the given class $y_t$ is always largest.
Challenges and core idea. From a cursory look at Eq. 3 it appears that finding the minimum is intractable. After all, our domain is discrete and we are optimizing over exponentially many configurations. Moreover, the margin is a function of the personalized PageRank which has a non-trivial dependency on the perturbed graph. But there is hope: For a fixed $H$, the margin $m_{y_t,c}(t)$ is a linear function of $\pi(e_t)$. Thus, Problem 1 reduces to optimizing a linear function of personalized PageRank over a specific constraint set. This is the core idea of our approach. As we will show, if we consider only local budget constraints the exact certificate can be efficiently computed. This is in contrast to most certificates for neural networks that rely on different relaxations to make the problem tractable. Including the global budget constraint, however, makes the problem hard. For this case we derive an efficient to compute lower bound on the worst-case margin. Thus, if the lower bound is positive we can still guarantee that our classifier is robust w.r.t. the set of admissible perturbations.

4.3 Optimizing topics-sensitive PageRank with global and local constraints

We are interested in optimizing a linear function of the topic-sensitive PageRank vector of a graph by modifying its structure. That is, we want to configure a set of fragile edges into included/excluded to obtain a perturbed graph $\tilde{G}$ maximizing the objective. Formally, we study the general problem:

Problem 2. Given a graph $G$, a set of admissible perturbations $Q_F$ as in Problem 1 and any fixed $x, r, \pi \in \mathbb{R}^N, \alpha \in (0, 1)$ solve the following optimization problem: $\max_{\tilde{G} \in Q_F} r^T \pi_{\tilde{G}, \alpha}(z)$.

Setting $r = -(H : y_t - H : c)$ and $z = e_t$, we see that Problem 1 is a special case of Problem 2. We can think of $r$ as a reward/cost vector, i.e. $r_v$ is the reward that a random walker obtains when visiting node $v$. The objective value $r^T \pi(z)$ is proportional to the overall reward obtained during an infinite random walk with teleportation since $\pi(z)_v$ exactly equals to the frequency of visits to $v$.

Variations and special cases of this problem have been previously studied [2, 11, 12, 15, 18, 25, 32]. Notably, Fercoq et al. [18] cast the problem as an average cost infinite horizon Markov decision process (MDP), also called ergodic control problem, where each node corresponds to a state and the actions correspond to choosing a subset of included fragile edges, i.e. we have $2^{|F|}$ actions at each state $v$ (see also Fig. 2). They show that despite the exponential number of actions, the problem can be efficiently solved in polynomial time, and they derive a value iteration algorithm with different local constraints. They enforce that the final perturbed graph has at most $b_v$ total number of edges per node, while we enforce that at most $b_v$ edges per node are perturbed (see Sec. 4.4).

Our approach for local budget only. Inspired by the MDP idea we derive a policy iteration (PI) algorithm which also runs in polynomial time [25]. Intuitively, every policy corresponds to a perturbed graph in $Q_F$, and each iteration improves the policy. The PI algorithm allows us to: incorporate our local constraints easily, take advantage of efficient solvers for sparse systems of linear equations (line 3 in Alg. 1), and implement the policy improvement step in parallel (lines 4-6 in Alg. 1). It can easily handle very large sets of fragile edges and it scales to large graphs.

Proposition 1. Algorithm 1 which greedily selects the fragile edges finds an optimal solution for Problem 2 with only local constraints in a number of steps independent of the size of the graph.

Algorithm 1 POLICY ITERATION WITH LOCAL BUDGET

Require: Graph $G = (V, E)$, reward $r$, set of fixed $E_f$ and fragile $F$ edges, local budgets $b_v$

1: Initialization: $\mathcal{W}_0 \subseteq F$ as any arbitrary subset, $A^G$ corresponding to $G$
2: while $\mathcal{W}_k \neq \mathcal{W}_{k-1}$ do
3: Solve $(I_N - \alpha D^{-1}A)x = r$ for $x$, where $A_{ij} = 1 - A^G_{ij}$ if $(i, j) \in \mathcal{W}_k$ # flip the edges
4: Let $l_{ij} \leftarrow (1 - 2A^G_{ij})(x_j - \frac{x_i + x_j}{2})$ for all $(i, j) \in F$ # calculate the improvement
5: Let $\mathcal{L}_v \leftarrow \{(v, j) \in F | l_{vj} > 0 \land l_{v} \geq \text{top } b_v \text{ largest } l_{v} \}$, $\forall v \in V$
6: $\mathcal{W}_k \leftarrow \bigcup_v \mathcal{L}_v$, $k \leftarrow k + 1$
7: end while
8: return $\mathcal{W}_k$ # optimal graph $\tilde{G} \in Q_F$ obtained by flipping all $(i, j) \in \mathcal{W}_k$ of $G$

We provide the proof in Sec 5.3 in the appendix. The main idea for Alg. 1 is starting from a random policy, in each iteration we first compute the mean reward before teleportation $x$ for the current policy (line 3), and then greedily select the top $b_v$ edges that improve the policy (lines 4-6). This algorithm is guaranteed to converge to the optimal policy, and thus to the optimal configuration of fragile edges.
Certificate for local budget only. Proposition \[1\] implies that for local constraints only, the optimal solution does not depend on the teleport vector $z$. Regardless of the node $t$ (i.e. which $z = e_t$ in Eq. \[3\]), the optimal edges to perturb are the same if the admissible set $\mathcal{Q}_f$ and the reward $r$ are the same. This means that for a fixed $Q_f$, we only need to run the algorithm $K \times K$ times to obtain the certificates for all $N$ nodes: For each pair of classes $c_1, c_2$ we have a different reward vector $r = -(H_{c_1} - H_{c_2})$, and we can recover the exact worst-case margins $m_{u,v}^*(\cdot)$ for all $N$ nodes by just computing $\Pi$ on the resulting $K \times K$ many perturbed graphs $\tilde{G}$. Now, $m_{u,v}^*(\cdot) > 0$ implies certifiable robustness, while $m_{u,v}^*(\cdot) < 0$ implies certifiable non-robustness due to the exactness of our certificate, i.e. we have found an adversarial example for node $t$.

Our approach for both local and global budget. Algorithm \[1\] cannot handle a global budget constraint, and in general solving Problem \[2\] with global budget is NP-hard. More specifically, it generalizes the Link Building problem \[32\] – find the set of preexisting edges $F$ that solves Problem \[2\] without constraints. The MDP is defined by the 4-tuple $(\mathcal{S}, (\mathcal{A}_i)_{i \in \mathcal{S}}, p, r)$, where $\mathcal{S}$ is the state space (preexisting and auxiliary nodes), and $\mathcal{A}_i$ is the set of admissible actions in state $i$. Given action $a \in \mathcal{A}_i$, $p(j|i, a)$ is the probability to go to state $j$ from state $i$ and $r(i, a)$ the instantaneous reward. Each preexisting node $i$ has a single action $\mathcal{A}_i = \{a\}$, reward $r(i, a) = r_i$, and uniform transitions $p(v_{ij}|i, a) = d_i^{-1}, \forall v_{ij} \in F$, discounted by $\alpha$ for the fixed edges $p(j|i, a) = \alpha \cdot d_i^{-1}, \forall (i, j) \in E_f$, where $d_i = \lvert E_f \cup F \rvert$ is the degree. For each auxiliary node we allow two actions $\mathcal{A}_{v_{ij}} = \{\text{on}, \text{off}\}$. For action "off" node $v_{ij}$ goes to node $i$ with probability 1 and obtains reward $-r_i$: $p(i|v_{ij}, \text{off}) = 1, r(v_{ij}, \text{off}) = -r_i$. For action "on" node $v_{ij}$ goes to node $j$ with probability $\alpha$ (the model is substochastic) and obtains 0 reward: $p(j|v_{ij}, \text{on}) = \alpha, r(v_{ij}, \text{on}) = 0$. We introduce fewer aux. nodes compared to previous work \[11, 17\].

(b) Global and local budgets QCLP. Based on this unconstrained MDP, we can derive a corresponding linear program (LP) solving the same problem \[34\]. Since the MDP on the auxiliary graph has (at most) binary action sets, the LP has only $2|V| + 3|F|$ constraints and variables. This is in strong contrast to the LP corresponding to the previous average cost MPD \[18\] operating directly on the original graph that has an exponential number of constraints and variables. Lastly, we enrich the LP for the aux. graph MDP with additional constraints enforcing the local and global budgets. The constraints for the local budget are linear, however, the global budget requires quadratic constraints resulting in a quadratically constrained linear program (QCLP) that exactly solves Problem \[2\].
Efficient Reformulation Linearization Technique (RLT).

Proposition 2. Solving the following QCLP (with decision variables $x_v, x^0_{ij}, x^1_{ij}, \beta^0_{ij}, \beta^1_{ij}$) is equivalent to solving Problem 2 with local and global constraints, i.e., the value of the objective function is the same in the optimal solution. We can recover $\pi(z)_v$ from $x_v$ via $\pi(z)_v = (1 - k_v d^-_{i,v}) x_v$. Here $k_v$ is the number of "off" fragile edges (the ones where $x^0_{ij} > 0$) in the optimal solution.

\[
\begin{align*}
\text{max} & \quad \sum_{v \in \mathcal{V}} v \mathbb{R}_v - \sum_{(i,j) \in \mathcal{E}} \sum_{v,i} x^0_{ij} r_i \\
& \quad - \alpha \sum_{(i,v) \in \mathcal{E}_f} x^0_{ij} - \alpha \sum_{(j,v) \in \mathcal{E}} x^1_{ij} = x^1_{ij} = 1 - \alpha \\forall v \in \mathcal{V} \\
& \quad x^0_{ij} + x^1_{ij} = \frac{x^1_{ij}}{d^1_{ij}}, \quad x^0_{ij}, x^1_{ij} \geq 0 \quad \forall (i,j) \in \mathcal{F} \\
& \quad \sum_{(v,i) \in \mathcal{F}} [(v,i) \in \mathcal{E}] x^0_{ij} + \sum_{(v,i) \notin \mathcal{E}} [x^1_{ij} < \frac{x^1_{ij}}{d^1_{ij}}, \quad x^1_{ij} \geq 0 \quad \forall v \in \mathcal{V} \\
& \quad x^0_{ij} \beta^0_{ij} = 1, \quad x^1_{ij} \beta^0_{ij} = 0, \quad \beta^1_{ij} = 1 - \beta^0_{ij}, \quad 0 \leq \beta^0_{ij} \leq 1 \quad \forall (i,j) \in \mathcal{F} \\
& \quad \sum_{(i,j) \in \mathcal{F}} \mathbb{E}[x^0_{ij} d_i x^1_{ij}] + \mathbb{E}[x^1_{ij} d_i x^0_{ij}] \leq B 
\end{align*}
\]

Key idea and insights. Eqs. 4b and 4c correspond to the LP of the unconstrained MDP. Intuitively, the variable $x_v$ maps to the PageRank score of node $v$, and from the variables $x^0_{ij}, x^1_{ij}$ we can recover the optimal policy: if the variable $x^0_{ij}$ (respectively $x^1_{ij}$) is non-zero then in the optimal policy the fragile edge $(i,j)$ is turned off (respectively on). Since there exists a deterministic optimal policy, only one of them is non-zero but never both. Eq. 4d corresponds to the local budget. Remarkably, despite the variables $x^0_{ij}, x^1_{ij}$ not being integral, since they share the factor $x_v d^{-1}_v$ from Eq. 4c we can exactly count the number of edges that are turned off or on using only linear constraints. Eqs. 4e and 4f enforce the global budget. From Eq. 4e we have that whenever $x^0_{ij}$ is nonzero it follows that $\beta^0_{ij} = 0$ and $\beta^1_{ij} = 1$ since that is the only configuration that satisfies the constraints (similarly for $x^1_{ij}$). Intuitively, this effectively makes the $\beta^0_{ij}, \beta^1_{ij}$ variables "counters" and we can utilize them in Eq. 4f to enforce the total number of perturbed edges to not exceed $B$. See detailed proof in Sec. 8.3.

(c) Efficient Reformulation Linearization Technique (RLT). The quadratic constraints in our QCLP make the problem non-convex and difficult to solve. We relax the problem using the Reformulation Linearization Technique (RLT) [38] which gives us an upper bound on the objective. The alternative SDP-relaxation [43] based on semidefinite programming is not suitable for our problem since the constraints are trivially satisfied (see Appendix 8.4 for details). While in general, the RLT introduces many new variables (replacing each product term $m_i m_j$ with a variable $M_{ij}$) along with multiple new linear inequality constraints, it turns out that in our case the solution is highly compact.

Proposition 3. Given a fixed upper bound $\overline{v}_e$ for $x_v$ and using the RLT relaxation, the quadratic constraints in Eqs. 4e and 4f transform into the following single linear constraint:

\[
\sum_{(i,j) \in \mathcal{F}} [(i,j) \in \mathcal{E}] x^0_{ij} d_i (x^1_{ij})^{-1} + [(i,j) \notin \mathcal{E}] x^1_{ij} d_i (x^0_{ij})^{-1} \leq B
\]
Proof provided in Sec. 8.3 in the appendix. By replacing Eqs. 4e and 4f with Eq. 5 in Proposition 2 we obtain a linear program which can be efficiently solved. Remarkably, we only have $x_0, x_{ij}^0, x_{ij}^1$ as decision variables since we were able to eliminate all other variables. The solution is an upper bound on the solution for Problem 2 and a lower bound on the solution for Problem 1. The final relaxed QCLP can also be interpreted as a constrained MPD with a single additional constraint (Eq. 5) which admits a possibly randomized optimal policy with at most one randomized state.

Certificate for local and global budget. To solve the relaxed QCLP and compute the final certificate we need to provide the upper bounds $\bar{x}_v$ for the constraint in Eq. 5. Since the quality of the RLT relaxation depends on the tightness of these upper bounds, we have to carefully select them. We provide here one solution (see Sec. 8.6 in the appendix for a faster to compute, but less tight, alternative): Given an instance of Problem 2, we can set the reward to $r = e_v$ and invoke Algorithm 1 which is highly efficient, using the same fragile set and the same local budget. Since this explicitly maximizes $x_v$, the objective value of the problem is guaranteed to give a valid upper bound $\bar{x}_v$. Invoking this procedure for every node, leads to the required upper bounds. Now, to compute the certificate with local and global budget for a target node $t$, we solve the relaxed problem for all $c \neq y_t$, leading to objective function values $L_{ct} \geq -m_{ct}^*(t)$ (minus due to the change from min to max). Thus, $L_{ct} = \min_{y \neq y_t} -L_{ct}$ is a lower bound on the worst-case margin $m_{ct}^*(t)$. If the lower bound is positive then node $t$ is guaranteed to be certifiably robust – there exists no adversarial attack (among all graphs in $Q_F$) that can change the prediction for node $t$.

For our policy iteration approach if $m_{ct}^*(t) < 0$ we are guaranteed to have found an adversarial example since the certificate is exact, i.e. we also have a non-robustness certificate. However in this case, if the lower bound $L_{ct}$ is negative we do not necessarily have an adversarial example. Instead, we can perturb the graph with the optimal configuration of fragile edges for the relaxed problem, and inspect whether the predictions change. See Fig. 1 for an overview of both approaches.

5 Robust training for graph neural networks

In Sec. 4 we introduced two methods to efficiently compute certificates given a trained $\pi$-PPNP model. We now show that these can naturally be used to go one step further – to improve the robustness of the model. The main idea is to utilize the worst-case margin during training to encourage the model to learn more robust weights. Optimizing some robust loss $L_0$ with respect to the model parameters $\theta$ (e.g. for $\pi$-PPNP $\theta$ are the neural network parameters) that depends on the worst-case margin $m_{ct}^*(v)$ is generally hard since it involves an inner optimization problem, namely finding the worst-case margin. This prevents us to easily take the gradient of $m_{ct}^*(v)$ (and, thus, $L_0$) w.r.t. the parameters $\theta$. Previous approaches tackle this challenge by using the dual.

Inspecting our problem, however, we see that we can directly compute the gradient. Since $m_{ct}^*(v)$ (respectively the corresponding lower bound) is a linear function of $H = f_\theta(X)$ and $\pi_G$, and furthermore the admissible set $Q_F$ over which we are optimizing is compact, it follows from Danskin’s theorem that we can simply compute the gradient of the loss at the optimal point. We have

$$\frac{\delta m_{ct}^*(v)}{\delta H_{ct}} = \pi^*(e_v), \quad \text{and} \quad \frac{\delta m_{ct}^*(v)}{\delta H_{ct}} = -\pi^*(e_v),$$

i.e. the gradient equals to the optimal ($\pm$) PageRank scores computed in our certification approaches.

Robust training. To improve robustness Wong & Kolter proposed to optimize the robust cross-entropy loss: $L_{RCE} = L_{CE}(y_v^*, -m_{ct}^*(v))$, where $L_{CE}$ is the standard cross-entropy loss operating on the logits, and $m_{ct}^*(v)$ is a vector such that at index $v$ we have $m_{ct}^*(v)$. Previous work has shown that if the model is overconfident there is a potential issue when using $L_{RCE}$ since it encourages high certainty under the worst-case perturbations. Therefore, we also study the alternative robust hinge loss. Since the attacker wants to minimize the worst-case margin $m_{ct}^*(t)$ (or its lower bound), a straightforward idea is to try to maximize it during training. To achieve this we add a hinge loss penalty term to the standard cross-entropy loss. Specifically:

$$L_{CEM} = \sum_{c \in V} L_{CE}(y_v^*, H_{ct}^{\text{marg}}) + \sum_{c \in Q_F} \max(0, M - m_{ct}^*(v)),$$

The second term for a single node $v$ is positive if $m_{ct}^*(v) < M$ and zero otherwise – the node $v$ is certifiably robust with a margin of at least $M$. Effectively, if all training nodes are robust, the second term becomes zero, thus, reducing $L_{CEM}$ to the standard cross-entropy loss with robustness guarantees. Note again that we can easily compute the gradient of these losses w.r.t. the (neural network) parameters $\theta$. 7
6 Experimental results

Setup. We focus on evaluating the robustness of $\pi$-PPNP without robust training and label/feature propagation using our two certification methods. We also verify that robust training improves the robustness of $\pi$-PPNP while maintaining high predictive accuracy. We demonstrate our claims on two publicly available datasets: Cora-ML ($N = 2,995, |\mathcal{E}| = 8,416, D = 2,879, K = 7$) \cite{yang2016revisiting, das2018learning} and Citeseer ($N = 3,312, |\mathcal{E}| = 4,715, D = 3,703, K = 6$) \cite{leskovec2005graph} with further experiments on Pubmed ($N = 19,717, |\mathcal{E}| = 44,324, D = 500, K = 3$) \cite{leskovec2005graph} in the appendix. We configure $\pi$-PPNP with one hidden layer of size 64 and set $\alpha = 0.85$. We select 20 nodes per class for the train/validation set and use the rest for the test set. We compute the certificates w.r.t. the predictions, i.e., we set $y_t$ in $m^*_{y_t}(t)$ to the predicted class for node $t$ on the clean graph. See Sec. 8.2 in the appendix for further experiments and Sec. 8.7 for more details. Note, we do not need to compare to any previously introduced adversarial attacks on graphs \cite{gilmer2017chemistry, wu2020robust, zhao2020adversarial}, since by the definition of a certificate for a certifiably robust node w.r.t. a given admissible set $Q_F$, there exist no successful attack within that set.

We construct several different configurations of fixed and fragile edges to gain a better understanding of the robustness of the methods to different kind of adversarial perturbations. Namely, "both" refers to the scenario where $\mathcal{F} = \mathcal{V} \times \mathcal{V}$, i.e., the attacker is allowed to add or remove any edge in the graph, while "rem." refers to the scenario where $\mathcal{F} = \mathcal{E}$ for a given graph $G = (\mathcal{V}, \mathcal{E})$, i.e., the attacker can only remove existing edges. In addition, for all scenarios we specify the fixed set as $\mathcal{E}_f = \mathcal{E}_{\text{mst}}$, where $(i, j) \in \mathcal{E}_{\text{mst}}$ if $(i, j)$ belongs to the minimum spanning tree (MST) on the graph $G$.

Robustness certificates: Local budget only. We investigate the robustness of different graphs and semi-supervised node classification methods when the attacker has only local budget constraints. We set the local budget $b_v = \max(d_v - 11 + s, 0)$ relative to the degree $d_v$ of node $v$ in the clean graph, and we vary the local attack strength $s$ with lower $s$ leading to a more restrictive budget. Such relative budget is justified since higher degree nodes tend to be more robust in general \cite{result1, result2}. We then apply our policy iteration algorithm to compute the (exact) worst-case margin for each node.

In Fig. 3a we see that the number of certifiably robust nodes when the attacker can only remove edges is significantly higher compared to when they can also add edges which is consistent with previous work on adversarial attacks \cite{zhao2020adversarial}. As expected, the share of robust nodes decreases with higher budget, and $\pi$-PPNP is significantly more robust than label propagation since besides the graph it also takes advantage of the node attributes. Feature propagation has similar performance ($F_1$ score) but it is less robust. Note that since our certificate is exact, the remaining nodes are certifiably non-robust! In Sec. 8.2 in the appendix we also investigate certifiable accuracy – the ratio of nodes that are both certifiably robust and at the same time have a correct prediction. We find that the certifiable accuracy is relatively close to the clean accuracy, and it decreases gracefully as we increase the budget.

Analyzing influence on robustness. In Fig. 3b we see that decreasing the damping factor $\alpha$ is an effective strategy to significantly increase the robustness with no noticeable loss in accuracy (at most 0.5% for any $\alpha$, not shown). Thus, $\alpha$ provides a useful trade-off between robustness and the size of the effective neighborhood: higher $\alpha$ implies higher PageRank scores (i.e., higher influence) for the
We derive the first (non-)robustness certificate for graph neural networks regarding perturbations. Algorithm 1 finds the optimal solution in just a few iterations. Since each iteration is itself efficient (dashed lines). The global constraint increases the number of robust nodes, validating our approach. This research was supported by the German Research Foundation, Emmy Noether grant GU 1409/2-1, and the German Federal Ministry of Education and Research (BMBF), grant no. 01IS18036B. The authors of this work take full responsibilities for its content.

7 Conclusion

We derive the first (non-)robustness certificate for graph neural networks regarding perturbations of the graph structure, and the first certificate overall for label/feature propagation. Our certificates are flexible w.r.t. the threat model, can handle both local (per node) and global budgets, and can be efficiently computed. We also propose a robust training procedure that increases the number of certifiably robust nodes while improving the predictive accuracy. As future work, we aim to consider perturbations and robustification of the node features and the graph structure jointly.

Acknowledgments

This research was supported by the German Research Foundation, Emmy Noether grant GU 1409/2-1, and the German Federal Ministry of Education and Research (BMBF), grant no. 01IS18036B. The authors of this work take full responsibilities for its content.
References


