Efficient Robustness Certificates for Discrete Data: Sparsity-Aware Randomized Smoothing for Graphs, Images and More

Aleksandar Bojchevski [1] **Johannes Gasteiger** [1] **Stephan Günnemann** [1]

Abstract

Existing techniques for certifying the robustness of models for discrete data either work only for a small class of models or are general at the expense of efficiency or tightness. Moreover, they do not account for sparsity in the input which, as our findings show, is often essential for obtaining non-trivial guarantees. We propose a model-agnostic certificate based on the randomized smoothing framework which subsumes earlier work and is tight, efficient, and sparsity-aware. Its computational complexity does not depend on the number of discrete categories or the dimension of the input (e.g. the graph size), making it highly scalable. We show the effectiveness of our approach on a wide variety of models, datasets, and tasks – specifically highlighting its use for Graph Neural Networks. So far, obtaining provable guarantees for GNNs has been difficult due to the discrete and non-i.i.d. nature of graph data. Our method can certify any GNN and handles perturbations to both the graph structure and the node attributes.[1]

1. Introduction

Verifying the robustness of machine learning models is crucial since data can be noisy, incomplete, manipulated by an adversary, or simply different from what was previously observed. Even a seemingly accurate classifier is of limited use if slight perturbations of the input can lead to misclassification. Robustness certificates provide provable guarantees that no perturbation regarding a specific threat model will change the prediction of an instance. However, obtaining meaningful robustness guarantees is challenging since it often involves solving a difficult optimization problem.

[1]Technical University of Munich. Correspondence to: Aleksandar Bojchevski <a.bojchevski@in.tum.de>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

[1]You can find the project page and the code online: https://www.daml.in.tum.de/sparse_smoothing

An overwhelming majority of certificates in the literature can handle only continuous data. The few approaches that tackle discrete data either work for a small class of models, or stay general while sacrificing efficiency or tightness. While our proposed approach works in general and can be used for any discrete data such as sequences (text, audio), discretized images or molecules, we highlight its use for graphs – a particularly important instance of discrete data.

Specifically, we focus on Graph Neural Networks (GNNs) since they are a fundamental building block (alongside CNNs and RNNs) for many machine learning models today. Their rise to prominence is not surprising since often real-world data can be naturally represented as a graph. They have been successfully applied across a variety of domains and applications: from breast cancer classification (Rhee et al., 2018) to fraud detection (Wang et al., 2019).

At the same time, there is strong evidence showing that GNNs suffer from poor adversarial robustness (Zügner et al., 2018; Dai et al., 2018; Zügner & Günnemann, 2019a) – they are sensitive to small adversarial perturbations designed to achieve a malicious goal. Take for example a GNN-based model for detecting fake news on a social network (Monti et al., 2019; Shu et al., 2020). Adversaries have a strong incentive to fool the system in order to avoid detection. In this context, a perturbation could mean modification of the graph structure (inserting or deleting edges in the social graph) or modifying the node attributes (e.g. the text content of the news). Even in scenarios where adversaries are unlikely, understanding the robustness of GNNs to worst-case noise is important, especially in safety-critical applications.

While some (heuristic) defenses exist (Xu et al., 2019; Entezari et al., 2020), we should never assume that the attackers will not be able to break them in the future (Carlini & Wagner, 2017). Robustness certificates, on the other hand, are by definition unbreakable. Given a clean input $\boldsymbol{x}$ and a perturbation set $\mathcal{B}_r(\boldsymbol{x})$ encoding the threat model (e.g. all inputs within an l_p-ball of radius r centered at $\boldsymbol{x}$) the goal is to verify that the prediction for $\boldsymbol{x}$ and $\forall \tilde{\boldsymbol{x}} \in \mathcal{B}_r(\boldsymbol{x})$ is the same. If this holds, we say that $\boldsymbol{x}$ is certifiably robust w.r.t. $\mathcal{B}_r(\boldsymbol{x})$.

Existing certificates for graphs handle either attribute perturbations (Zügner & Günnemann, 2019b) or structure per-

turbations (Bojchevski & Günnemann, 2019a; Zügner & Günnemann, 2020), but not both, and only work for a small class of models. Furthermore, they are valid only for node-level classification, and extending these techniques to new models and threat scenarios is not straightforward. Our approach handles both types of perturbations and applies to any GNN. This includes, for the first time, graph-level classification models for which there are no existing certificates.

In this paper we utilize randomized smoothing (Cohen et al., 2019) – a powerful general technique for building certifiably robust models. Inspired by connections to differential privacy (Lécuyer et al., 2019), this method boils down to randomly perturbing the input and reporting the output/class corresponding to the "majority vote" on the randomized samples. Given any function $f(\cdot)$, e.g. any GNN, we can build a "smoothed" function $g(\cdot)$ that produces a similar output to f (e.g. comparable accuracy if f is a classifier) and for which we can easily provide (probabilistic) robustness guarantees. Importantly, to compute the certificate we need to consider *only* the output of f for each sample. This is precisely what makes it particularly appealing for certifying GNNs since it allows us to sidestep a complex analysis of the message-passing dynamics and the non-linear interactions between the nodes. Randomized smoothing is not without limitations however, which we discuss in § L.

The bulk of the work on randomized smoothing (Cohen et al., 2019; Lécuyer et al., 2019; Li et al., 2018) focuses on continuous data and guarantees in terms of l_1, l_2 or l_∞ balls which are not suited for the discrete data domain. Only few approaches can tackle discrete data with l_0-ball guarantees (Lee et al., 2019; Levine & Feizi, 2019; Dvijotham et al., 2020). None of these approaches attempt to certify discrete *graph* data, and there are several major challenges we need to overcome to successfully do so. Jia et al. (2020b) apply randomized smoothing to only certify the robustness of community detection against structural perturbations. Their certificate also suffers from the same limitations.

The biggest limitation of *all* previous certificates for discrete data is that they rely on randomization schemes that do not take sparsity into account. A common scheme is to randomly flip bits in the input with a given probability p. This is clearly not feasible for graph data due to the sparsity of real-world graphs. Even for a small flip probability (e.g. $p = 0.01$) applying this scheme would introduce too many random edges in the graph, which means that the graph structure is completely destroyed by the random noise, rendering the resulting smoothed classifier useless.[2] On the other hand, p has to be sufficiently high to obtain any guarantees, since higher p values lead to higher certified radii. Similarly, the node attributes are also often sparse vectors, e.g. corresponding to bag-of-words representations of text, and suffer from the same issue. None of the existing discrete certificates are sparsity-aware. The core idea of this paper is to incorporate sparsity in the randomization scheme by perturbing non-zeros/edges and zeros/non-edges separately in a way that preservers the structure of the data.

Besides the common issue with sparsity, Lee et al. (2019)'s and Jia et al. (2020b)'s certificates are tight but computationally expensive, while Levine & Feizi (2019) and Dvijotham et al. (2020)'s certificates sacrifice tightness to obtain improved runtime. We overcome these limitations and propose a certificate which is at the same time tight, efficient to compute, and sparsity-aware. In summary, we make contributions on two fronts:

1. **GNN Certificates**: Our certificates handle both structure and attribute perturbations and can be applied to any GNN, including graph-level classification models.
2. **Discrete Certificates**: (i) We generalize previous work by explicitly accounting for sparsity; (ii) We obtain tight certificates with a dramatically reduced computational complexity, independent of the input size.

The key observation behind these contributions is that we can partition the space of binary vectors into a *small* number of regions of constant likelihood ratio. The certificate is obtained by traversing these regions and keeping track of the PMF w.r.t. the clean input and the adversarial example. For example, for binary data the number of regions in our partitioning equals the size of the (certified) radius, i.e. grows linearly, and *does not* depend on the input size. This is in stark contrast to previous work where the number of regions is quadratic w.r.t. the input size. Considering that the adjacency matrix of a graph with n nodes has n^2 entries, this reduction in complexity from up to $(n^2)^2 = n^4$ to r regions (where r is the radius) is necessary for feasibility. Furthermore, by drawing connections between our randomization and the Poisson-Binomial distribution for binary data (product of Multinomials for discrete data) we develop an algorithm to efficiently traverse and compute these regions.

2. Background and Preliminaries

Let $\boldsymbol{x} \in \mathcal{X} = \{0, 1\}^d$ be an observed binary vector. For simplicity we keep the main exposition w.r.t. binary data and we discuss the general discrete case in § 5. In § 6 we show how to instantiate our framework for GNNs, where $\boldsymbol{x}$ corresponds to the (flattened) adjacency and/or attribute matrix of a graph. We defer all proofs to the appendix (§ A).

Given a classifier $g(\cdot)$ the goal of the attacker is to find an adversarial example $\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})$ in the perturbation set such

[2] For example, the Cora-ML dataset has $n = 2810$ nodes, so random sampling introduces $pn^2 = 0.01 \cdot 2810^2 = 78961$ random edges in expectation, i.e. around 28 random edges per node, which is significantly higher than the average node degree of 6.

that $\tilde{\boldsymbol{x}}$ is misclassified[3], i.e. $g(\boldsymbol{x}) \neq g(\tilde{\boldsymbol{x}})$ (evasion attack). Our goal is to verify whether such an adversarial example exists, i.e. verify whether $g(\boldsymbol{x}) \stackrel{?}{=} g(\tilde{\boldsymbol{x}})$ for all $\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})$.

2.1. Randomized Smoothing Framework

Let $f : \mathcal{X} \to \mathcal{Y}$ denote a (deterministic or random) function corresponding to a base classifier which takes a vector $\boldsymbol{x} \in \mathcal{X}$ as input and outputs a single class $f(\boldsymbol{x}) = y \in \mathcal{Y}$ with $\mathcal{Y} = \{1, \dots, C\}$. We construct a smoothed (ensemble) classifier $g : \mathcal{X} \to \mathcal{Y}$ from f as follows:

$$g(\boldsymbol{x}) = \arg\max_{y \in \mathcal{Y}} \Pr(f(\phi(\boldsymbol{x})) = y) \tag{1}$$

where ϕ is a randomization scheme to be specified (e.g. adding Gaussian noise to $\boldsymbol{x}$), which assigns probability mass $\Pr(\phi(\boldsymbol{x}) = \boldsymbol{z})$ for each randomized outcome $\boldsymbol{z} \in \mathcal{X}$. In other words, $g(\boldsymbol{x})$ returns the most likely class (the majority vote) if we first randomly perturb the input $\boldsymbol{x}$ using ϕ and then classify the resulting vector $\phi(\boldsymbol{x})$ with the base classifier f. To simplify notation let $p_y(\boldsymbol{x}) = \Pr(f(\phi(\boldsymbol{x})) = y)$ and $y^* = \arg\max_{y \in \mathcal{Y}} p_y(\boldsymbol{x})$. Let $p^* = p_{y^*}(\boldsymbol{x})$ be the probability of the most likely class. Following Lee et al. (2019) we define the certificate:

$$\rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}}(p, y) = \min_{\substack{h \in \mathcal{H}: \\ \Pr(h(\phi(\boldsymbol{x}))=y)=p}} \Pr(h(\phi(\tilde{\boldsymbol{x}})) = y) \tag{2}$$

where $\tilde{\boldsymbol{x}} \in \mathcal{X}$ is a given neighboring point, and $\mathcal{H}$ is the set of measurable classifiers with respect to ϕ. We have that $\rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}}(p, y) \leq \Pr(f(\phi(\tilde{\boldsymbol{x}})) = y)$ is a *tight* lower bound on the probability that a neighboring point $\tilde{\boldsymbol{x}}$ is classified as y using the smoothed classifier g. The bound is tight in the sense that the base classifier f satisfies the constraint.

Now, given a clean input $\boldsymbol{x}$ and a perturbation set $\mathcal{B}(\boldsymbol{x})$ specifying a threat model (e.g. l_0-ball), if it holds that:

$$\min_{\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})} \rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}}(p^*, y^*) > 0.5 \tag{3}$$

then we can guarantee that $\Pr(f(\phi(\tilde{\boldsymbol{x}})) = y^*) > 0.5$, for all $\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})$. This implies that $g(\boldsymbol{x}) = g(\tilde{\boldsymbol{x}}) = y^*$ for any input within the ball, i.e. $\boldsymbol{x}$ is certifiably robust.

Computing $p_y(\boldsymbol{x})$ exactly is difficult, so similar to previous work (Cohen et al., 2019) we compute a lower bound $\underline{p_y(\boldsymbol{x})}$ based on the Clopper-Pearson Bernoulli confidence interval (Clopper & Pearson, 1934) with confidence level α using Monte Carlo samples from $\phi(\cdot)$. Since $\rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}}(p)$ is an increasing function of p (Lee et al., 2019), a lower bound entails a valid certificate. The certificate is probabilistic and holds with probability $1 - \alpha$.

Eq. 3 is tight for two classes and provides a sufficient condition to guarantee robustness for more classes ($|\mathcal{Y}| > 2$). In § B we show how to obtain better guarantees for multi-class classification by computing confidence intervals that hold *simultaneously* for all classes using Bonferroni correction.

[3] Or classified as some chosen target class other than $g(x)$.

2.2. Solving the Optimization Problem in Eq. 2

Assume we can partition $\mathcal{X} = \bigcup_i^I \mathcal{R}_i, \mathcal{R}_i \cap \mathcal{R}_j = \emptyset$ into disjoint regions $\mathcal{R}_i$ of constant likelihood ratio, i.e. for every $\boldsymbol{z} \in \mathcal{R}_i$ it holds $\Pr(\phi(\boldsymbol{x}) = \boldsymbol{z}) / \Pr(\phi(\tilde{\boldsymbol{x}}) = \boldsymbol{z}) = c_i$ for some constant c_i. Then, Eq. 2 is equivalent to the following Linear Program (LP) (Lee et al., 2019):

$$\min_{\boldsymbol{h}} \boldsymbol{h}^T \tilde{\boldsymbol{r}} \quad \text{s.t.} \quad \boldsymbol{h}^T \boldsymbol{r} = p, \quad 0 \leq \boldsymbol{h} \leq 1 \tag{4}$$

where $\boldsymbol{h} \in [0, 1]^I$ is the vector we are optimizing over corresponding to the classifier h, and $\boldsymbol{r}$ is a vector where $\boldsymbol{r}_i = \Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_i)$ for each region, and similarly for $\tilde{\boldsymbol{r}}_i$. The exact solution to this LP can be easily obtained with a greedy algorithm: first sort the regions such that $c_1 \geq c_2 \geq \cdots \geq c_I$, then iteratively assign $\boldsymbol{h}_i = 1$ for all regions $\mathcal{R}_i$ until the budget constraint is met (except for the final region which we "consume" partially). See § A for more details. Therefore, how efficiently we can compute the certificate depends on the number of regions and how difficult it is to compute $\Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_i)$ for a given $\mathcal{R}_i$ and $\boldsymbol{x}$. This is why reducing the number of regions is crucial.

We show that the optimization problem for the multi-class certificate is also a simple LP and can be exactly solved with a similar greedy algorithm (§ B). Another interpretation of Eq. 2 is that it corresponds to likelihood ratio testing with significance level p between two different hypotheses: $\Pr(\phi(\boldsymbol{x}) = \boldsymbol{z})$ vs. $\Pr(\phi(\tilde{\boldsymbol{x}}) = \boldsymbol{z})$ (Tocher, 1950). We show in § 4.3 that given our choice of randomization ϕ, the problem is equivalent to hypothesis testing of two Poisson-Binomial distributions with different parameters.

3. Threat Model

We assume that an adversary can perturb $\boldsymbol{x}$ by flipping some of its bits. We define the ball centered at the clean input $\boldsymbol{x}$:

$$\mathcal{B}_{r_a,r_d}(\boldsymbol{x}) = \{\tilde{\boldsymbol{x}} : \tilde{\boldsymbol{x}} \in \mathcal{X}, \sum_{i=1}^{d} \mathbb{I}(\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i - 1) \leq r_d, \sum_{i=1}^{d} \mathbb{I}(\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i + 1) \leq r_a\} \tag{5}$$

which contains all binary vectors $\tilde{\boldsymbol{x}}$ which can be obtained from $\boldsymbol{x}$ by deleting at most r_d bits (flipping from 1 to 0) and adding at most r_a bits (flipping from 0 to 1). Analogously, we define the sphere $\mathcal{S}_{r_a,r_d}(\boldsymbol{x})$ where the inequalities in Eq. 5 are replaced by equalities. The minimum over $\mathcal{B}_{r_a,r_d}(\boldsymbol{x})$ in Eq. 3 is always attained at some $\tilde{\boldsymbol{x}} \in \mathcal{S}_{r_a,r_d}(\boldsymbol{x})$.

Intuitively, the radii r_a and r_d control the global budget of the attacker, i.e. the overall number of additions or deletions

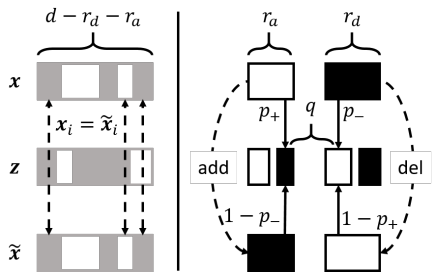


Figure 1. The vector $\tilde{\boldsymbol{x}}$ is obtained from $\boldsymbol{x}$ by adding exactly r_a bits and deleting exactly r_d bits. Any vector $\boldsymbol{z}$ in the region $\mathcal{R}_q^{r_a,r_d}$ is obtained by flipping q bits in $\boldsymbol{x}_\mathcal{C}$ and not flipping (retaining) q bits in $\tilde{\boldsymbol{x}}_\mathcal{C}$. Solid boxes denote ones and empty boxes denote zeros.

they can make. This is in contrast to other threat models for binary/graph data which do not distinguish between addition and deletion. Threat models for graphs often specify additional local budget constraints, e.g. at most given number of perturbations per node. We focus on global constraints which correspond to more powerful attacks.

Note that with this threat model we can also provide l_0-ball guarantees, i.e. to certify w.r.t. $\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_0 \leq r$ we can simply certify w.r.t. all balls $\mathcal{B}_{r_a,r_d}(\boldsymbol{x})$ where $r_a + r_d = r$.

4. Sparsity-Aware Certificate

4.1. Data-Dependent Sparsity-Aware Randomization

We define the following noise distribution with two parameters $p_-, p_+ \in [0, 1]$ independently for each dimension i:

$$\Pr(\phi(\boldsymbol{x})_i \neq \boldsymbol{x}_i) = p_-^{\boldsymbol{x}_i} p_+^{(1-\boldsymbol{x}_i)} \tag{6}$$

The randomization scheme ϕ flips the bit $\boldsymbol{x}_i = 1$ to 0 (e.g. deletes an existing edge) with probability p_-, and similarly flips the bit $\boldsymbol{x}_i = 0$ to 1 (e.g. adds a new edge) with probability p_+. This allows us to control the amount of smoothing separately for the ones and zeros (edges and non-edges). In other words, the noise distribution is *data-dependent*, which is in contrast to all previous randomized smoothing certificates. Moreover, we say that ϕ is sparsity-aware since often, for real-world data, the number of ones in $\boldsymbol{x}$ is significantly smaller than the number of zeros, i.e. $\|\boldsymbol{x}\|_0 \ll d$.

As we will show in § 8 sparsity-awareness is crucial for obtaining non-trivial certificates. The randomization scheme defined in Lee et al. (2019) is a special case which flips the i-th bit $\boldsymbol{x}_i$ with a single probability $p = p_- = p_+$ regardless of its value. For the general discrete case see § 5.

4.2. Regions of Constant Likelihood Ratio

We can partition $\mathcal{X}$ into a *small* number of regions of constant likelihood which enables us to use the greedy algorithm for solving Eq. 2 specified in § 2.2 to obtain an efficient certificate. Given any $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}} \in \mathcal{S}_{r_a,r_d}(\boldsymbol{x})$, let $\mathcal{C} = \{i : \boldsymbol{x}_i \neq \tilde{\boldsymbol{x}}_i\}$ be the set of dimensions where $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ disagree, and let $\bar{\mathcal{C}} = \{1, \dots, d\} \setminus \mathcal{C}$ be its complement. Now, let $\boldsymbol{x}_\mathcal{C}, \tilde{\boldsymbol{x}}_\mathcal{C} \in \{0, 1\}^{|\mathcal{C}|}$ denote the vectors $\boldsymbol{x}, \tilde{\boldsymbol{x}}$ considering only the dimensions specified in $\mathcal{C}$.

We define the region $\mathcal{R}_q^{r_a,r_d}$ containing all binary vectors $\boldsymbol{z}$ which can be obtained by flipping exactly q bits in $\boldsymbol{x}_\mathcal{C}$ and which have *any* configuration of ones and zeros in $\boldsymbol{x}_{\bar{\mathcal{C}}}$:

$$\mathcal{R}_q^{r_a,r_d} = \{\boldsymbol{z} \in \mathcal{X} : \|\boldsymbol{x}_\mathcal{C} - \boldsymbol{z}_\mathcal{C}\|_0 = q, \\ \|\boldsymbol{1} - \boldsymbol{x}_\mathcal{C}\|_0 = r_a, \|\boldsymbol{x}_\mathcal{C}\|_0 = r_d\}$$

The region $\mathcal{R}_q^{r_a,r_d}$ contains at the same time all vectors $\boldsymbol{z}$ which can be obtained by retaining (not flipping) q bits in $\tilde{\boldsymbol{x}}_\mathcal{C}$, i.e. $\|\tilde{\boldsymbol{x}}_\mathcal{C} - \boldsymbol{z}_\mathcal{C}\|_0 = r_d + r_a - q$ for all $\boldsymbol{z} \in \mathcal{R}_q^{r_a,r_d}$. To see this, note that from the definition of $\mathcal{S}_{r_a,r_d}(\boldsymbol{x})$, $\tilde{\boldsymbol{x}}_\mathcal{C}$ is the complement to $\boldsymbol{x}_\mathcal{C}$, and we can obtain $\tilde{\boldsymbol{x}}_\mathcal{C}$ from $\boldsymbol{x}_\mathcal{C}$ by flipping exactly r_d bits from 1 to 0, and flipping exactly r_a bits from 0 to 1. See Fig. 1 for an illustration.

We can partition $\mathcal{X}$ in exactly $r_a + r_d + 1$ such regions.

Proposition 1 *The set* $\{\mathcal{R}_0^{r_a,r_d}, \dots, \mathcal{R}_{r_a+r_d}^{r_a,r_d}\}$ *partitions the entire space of binary vectors* $\mathcal{X}$ *into disjoint regions, i.e.* $\mathcal{X} = \bigcup_{q=0}^{q=r_a+r_d} \mathcal{R}_q^{r_a,r_d}$ *and* $\mathcal{R}_i^{r_a,r_d} \cap \mathcal{R}_j^{r_a,r_d} = \emptyset, \forall i \neq j$.

Since the smoothing is independent per dimension we can restrict our attention only to those dimensions where $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ disagree, otherwise $\Pr(\phi(\boldsymbol{x})_i \neq \boldsymbol{x}_i) = \Pr(\phi(\tilde{\boldsymbol{x}})_i \neq \tilde{\boldsymbol{x}}_i)$ for $i \in \bar{\mathcal{C}}$ which does not change the ratio c_q for any region $\mathcal{R}_q^{r_a,r_d}$. This implies that the number of regions is independent of the dimension d. Furthermore, by definition $|\mathcal{C}| = r_a + r_d$, thus we can make between 0 and $r_a + r_d$ flips in total counting only w.r.t. the dimensions in $\mathcal{C}$, and any given $\boldsymbol{z}$ vector belongs only to a single region.

4.3. Poisson-Binomial View of the Regions

Before we state further results, it is helpful to consider a different view of the randomization scheme ϕ and how it influences the regions. The scheme $\phi(\cdot)$ is equivalent to first drawing a noise sample $\epsilon_i \sim \text{Ber}(p = p_-^{\boldsymbol{x}_i} p_+^{(1-\boldsymbol{x}_i)})$ from a Bernoulli distribution with probability $p = p_-$ if $\boldsymbol{x}_i = 1$ or $p = p_+$ otherwise, and setting $\phi(\boldsymbol{x})_i = \boldsymbol{x}_i \oplus \epsilon_i$, where $\oplus$ is the XOR. Here, we directly see that ϕ is data-dependent and sparsity-aware since we can specify e.g. a relatively large p_- for the ones and relatively small p_+ for the zeros to avoid introducing too many noisy bits in $\boldsymbol{x}$.

Proposition 2 *Given any* $\boldsymbol{x}, \tilde{\boldsymbol{x}} \in \mathcal{S}_{r_a,r_d}(\boldsymbol{x})$ *and any region* $\mathcal{R}_q^{r_a,r_d}$, $\Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_q^{r_a,r_d}) = \Pr(Q = q)$ *where* $Q \sim \text{PB}([p_+, r_a], [p_-, r_d]) = \text{PB}(\underbrace{p_+, \dots, p_+}_{r_a \text{ times}}, \underbrace{p_-, \dots, p_-}_{r_d \text{ times}})$ *is a Poisson-Binomial random variable on* $\{0, \dots, r_a + r_d\}$.

Intuitively, all vectors $\boldsymbol{z} \in \mathcal{R}_q^{r_a,r_d}$ correspond to observing q "successes" where a "success" is interpreted as successfully flipping the bit of $\boldsymbol{x}_{\mathcal{C}}$, which happens with probability p_- or p_+. At the same time, "success" is interpreted as retaining (not flipping) the bit of $\tilde{\boldsymbol{x}}_{\mathcal{C}}$ with probability $(1-p_-)$ or $(1-p_+)$. The probability distribution for the number of successes is a sum of d independent, but not identical (since $p_i = p_+$ or $p_j = p_-$) Bernoulli random variables which is a Poisson-Binomial random variable.

Since ϵ_i are independent we have $\Pr(\phi(\boldsymbol{x}) = \boldsymbol{z}) = \prod_{i\in\tilde{\mathcal{C}}} \Pr(\phi(\boldsymbol{x})_i = \boldsymbol{z}_i) \prod_{j\in\mathcal{C}} \Pr(\phi(\boldsymbol{x})_j = \boldsymbol{z}_j)$. By definition $\mathcal{R}_q^{r_a,r_d}$ contains all vectors $\boldsymbol{z}$ that have any configuration of ones and zeros in $\tilde{\mathcal{C}}$ so when we sum over all $\boldsymbol{z} \in \mathcal{R}_q^{r_a,r_d}$ the first product equals 1. Therefore, we can equivalently consider a sum of only $|\mathcal{C}|$ non-identical Bernoulli random variables which is a Poisson-Binomial random variable, i.e. $\Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_q^{r_a,r_d}) = \text{PB}([p_+, r_a], [p_-, r_d])$ is a $|\mathcal{C}|$ dimensional Poisson-Binomial distribution with two groups of distinct probabilities. See Fig. 1 for an illustration.

In other words, Eq. 2 can be seen as performing likelihood ratio testing where the two hypotheses correspond to two Poisson-Binomial distributions with different parameters, $\text{PB}([p_+, r_a], [p_-, r_d])$ vs. $\text{PB}([1-p_-, r_a], [1-p_+, r_d])$ relating to $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ respectively.[4]

For the special case $p_+ = p_- = p$, the Poisson-Binomial distribution reduces to a standard Binomial distribution, i.e. $Q \sim \text{Bin}(p, r_a + r_d)$. Analogously, for discrete data the probability for $\phi(\boldsymbol{x})$ to land in the respective regions is a Multinomial distribution (see § 5 and § M). This allows us to obtain the same certificate as in Lee et al. (2019) for a significantly reduced cost, and highlights that the choice of how we partition the space into regions is crucial.

Proposition 3 *For all $\boldsymbol{z} \in \mathcal{R}_q^{r_a,r_d}$, the likelihood ratio is*

$$\eta_q^{r_a,r_d} = \frac{\Pr(\phi(\boldsymbol{x}) = \boldsymbol{z})}{\Pr(\phi(\tilde{\boldsymbol{x}}) = \boldsymbol{z})} = \left[\frac{p_+}{1-p_-}\right]^{q-r_d} \left[\frac{p_-}{1-p_+}\right]^{q-r_a}$$

and is constant in the region $\mathcal{R}_q^{r_a,r_d}$. Moreover, for a fixed r_a and r_d, the ratio $\eta_q^{r_a,r_d}$ is a monotonically decreasing function of q if $(p_- + p_+) < 1$, constant if $(p_- + p_+) = 1$, or monotonically increasing function of q if $(p_- + p_+) > 1$.

We make several observations about our propositions and provide detailed proofs in § A.

Linear number of regions. With Prop. 1 and Prop. 3 we can partition $\mathcal{X}$ into exactly $(r_a + r_d + 1)$ number of regions with constant likelihood ratio. The number of regions grows *linearly* with the radii. Crucially, this implies that the number of regions is *independent* of the input size d. In the special case when $p_- = 0$ and $p_+ > 0$ (or similarly $p_- > 0$ and $p_+ = 0$) there are only three (non-empty) regions. For a discussion of these cases see § C.

Data (size) independence. From Prop. 2 and Prop. 3 it follows that the value of $\rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}}(p, y)$, and hence the certificate, is exactly the same for any $p, y, \boldsymbol{x}$ and $\tilde{\boldsymbol{x}} \in \mathcal{S}_{r_a,r_d}(\boldsymbol{x})$. In other words, as long as $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ differ in exactly r_d zeros and r_a ones, the solution to Eq. 3 is the same. Moreover, the certificate does not depend on the configuration of ones and zeros in the dimensions $\tilde{\mathcal{C}}$ where $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ agree since neither the probability $\Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_q^{r_a,r_d})$ nor the ratio $\eta_q^{r_a,r_d}$ depend on the values of $\boldsymbol{x}_i$ and $\tilde{\boldsymbol{x}}_i$ for $i \in \tilde{\mathcal{C}}$.

Altogether this means that w.l.o.g. we can compute the certificate based on the following two canonical vectors: $\boldsymbol{x}_{\text{ca}} = (1, \ldots, 1, 0, \ldots, 0)$ and $\tilde{\boldsymbol{x}}_{\text{ca}} = (0, \ldots, 0, 1, \ldots, 1)$, where $\|\boldsymbol{x}_{\text{ca}}\|_0 = r_d$ and $\|\tilde{\boldsymbol{x}}_{\text{ca}}\|_0 = r_a$. We can furthermore conclude that if several inputs have the same $\underline{p_y(\boldsymbol{x})}$, which is indeed the case in practice, we only need to compute the certificate once to certify all of them.

No sorting. Since $\eta_q^{r_a,r_d}$ is monotonic in q we do not need to construct all regions in advance and afterwards sort them in a decreasing order. We can completely avoid the sorting required for the greedy algorithm outlined in § 2.2 and directly visit the regions one by one, increasing q (or decreasing when $p_+ + p_- > 1$) each time until we reach $\underline{p_y(\boldsymbol{x})}$. For more details and pseudo-code see § D.

4.4. Efficiently Computing $\Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_q^{r_a,r_d})$

Since $\Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_q^{r_a,r_d}) = \text{PB}(q; [p_+, r_a], [p_-, r_d])$ we need to compute the PMF of a Poisson-Binomial distribution. If done naively we need to sum $r!/[r!(r-q)!]$ terms where $r = r_a + r_d$. Fortunately, there is a recursive formula that requires only $\mathcal{O}(qr)$ operations (Chen & Liu, 1997). Since we only have two distinct flip probabilities we can further simplify to obtain the following recursive formula:

$$T_{r_a,r_d}(i) = r_a \cdot (p_+/(1-p_+))^i + r_d \cdot (p_-/(1-p_-))^i$$

$$R_{r_a,r_d}(q) = \frac{1}{q} \sum_{i=1}^{q} (-1)^{i+1} \cdot T_{r_a,r_d}(i) \cdot R_{r_a,r_d}(q-i)$$

Now $\text{PB}(q; \cdot) = R_{r_a,r_d}(q) \cdot (1-p_+)^{r_a} \cdot (1-p_-)^{r_d}$. To avoid unnecessary computations we additionally unroll the recursion with dynamic programming.[5] An alternative approach is to compute the PMF via the Discrete Fourier Transform (Fernández & Williams, 2010). Compared to previous discrete certificates (Lee et al., 2019; Levine & Feizi, 2019) we do not need to compute Binomial coefficients.

[4] The greedy algorithm in § 2.2 is thus equivalent to $\rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}}(p) = \Phi(\Phi^{-1}_{\text{PB}_{\boldsymbol{x}}}(p))_{\text{PB}_{\tilde{\boldsymbol{x}}}}$ where Φ and Φ^{-1} are the CDF and inverse CDF function of the Poisson-Binomial distribution respectively.

[5] For multiple-precision arithmetic we use the gmpy2 Python library: `https://pypi.org/project/gmpy2/`.

5. General Certificate for Discrete Data

Let $\boldsymbol{x} \in \mathcal{X}_K = \{0, \dots, K-1\}^d$ be a d-dimensional vector where each $\boldsymbol{x}_i$ belongs to one of K different categories. We define the sparsity-aware randomization scheme $\phi(\cdot)$:

$$\Pr(\phi(\boldsymbol{x}_i) = k) = \begin{cases} [\frac{p_+}{K-1}]^{(\boldsymbol{x}_i \neq k)}(1-p_+)^{(\boldsymbol{x}_i = k)}, & \boldsymbol{x}_i = 0 \\ [\frac{p_-}{K-1}]^{(\boldsymbol{x}_i \neq k)}(1-p_-)^{(\boldsymbol{x}_i = k)}, & \boldsymbol{x}_i \neq 0 \end{cases}$$

That is, we flip zeros with probability p_+, and non-zeros with probability p_-, uniformly to any of the other values. For the special case $p_+ = p_-$ we recover the randomization scheme and the certificate from Lee et al. (2019), and for $K = 2$ we recover our certificate for binary data.

As before, we can partition $\mathcal{X}_K$ into disjoint regions of constant likelihood ratio and efficiently solve the problem defined in Eq. 2. We show that the number of regions does not depend on the number of discrete categories K or the dimension of the input d. Specifically, for $p_+ = p_-$ we have exactly $2r + 1$ regions where r is the certified radius, i.e. $\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_0 = r$. For $p_+ \neq p_-$ the number of regions is upper bounded by $(r+1)^2$. Here the key insight is that again $\Pr(\phi(\boldsymbol{x})_i \neq \boldsymbol{x}_i) = \Pr(\phi(\tilde{\boldsymbol{x}})_i \neq \tilde{\boldsymbol{x}}_i)$ if $\boldsymbol{x}_i = \tilde{\boldsymbol{x}}_i$ so w.l.o.g. we can consider only the dimensions where $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ disagree. For a detailed analysis of the regions and how to efficiently compute them see § M in the appendix.

5.1. Comparison with Existing Discrete Certificates

There are up to $(d+1)^2$ non-empty regions for the partitioning in Lee et al. (2019), i.e. quadratic w.r.t. input size. Since their certificate is a special case ($p_+ = p_-$) our partitioning provides a dramatic reduction of complexity. For example, to certify perturbations to the binary adjacency matrix where $d = n^2$ we have to traverse up to $\mathcal{O}(n^4)$ regions which is infeasible even for small graphs. With our certificate we have to examine at most $r_a + r_d + 1$ regions regardless of the graph size. Beyond this, in § 8.2 we show that our sparsity-aware randomization yields a higher certified ratio. Other certificates for discrete data which are based on f-divergences (Dvijotham et al., 2020) or randomized ablation (Levine & Feizi, 2019) sacrifice tightness to gain computational efficiency and provide looser guarantees.

6. Instantiating the Certificate for GNNs

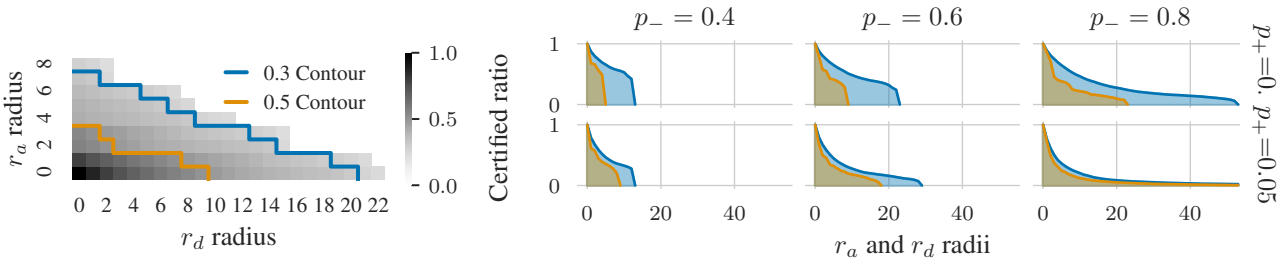
Let $G = (\mathcal{V}, \mathcal{E})$ be an attributed graph with $n = |\mathcal{V}|$ nodes. We denote with $\boldsymbol{A} \in \{0,1\}^{n \times n}$ the adjacency matrix and $\boldsymbol{F} \in \{0,1\}^{n \times m}$ the matrix of m-dimensional binary features for each node. We consider three different scenarios: (i) the adversary can only perturb the graph structure: $\boldsymbol{x} = \text{vec}(\boldsymbol{A})$, (ii) only the node attributes: $\boldsymbol{x} = \text{vec}(\boldsymbol{F})$, (iii) or both: $\boldsymbol{x} = [\text{vec}(\boldsymbol{A}), \text{vec}(\boldsymbol{F})]$. Here $\text{vec}(\cdot)$ "flattens" a matrix into a vector, and $[\cdot, \cdot]$ denotes concatenation. When the graph is undirected, $\text{vec}(\boldsymbol{A})$ considers only the lower (or upper)-triangular part of $\boldsymbol{A}$. The base classifier $f(\cdot)$ can be any GNN. If we are certifying the node classification task, perturbing a single given graph can potentially change the predictions for *all* nodes. To certify a given target node t we simply focus on its own predictions (its own distribution over node-level classes) which in general could be computed based on the entire graph. Note, under our threat model we can apply the perturbation anywhere in the graph/features, e.g. including the neighbors of node t. Here we focus on node-level classification and in § G in the appendix we show results for graph-level classification.

6.1. Joint Certificates for the Graph and the Attributes

When jointly certifying perturbations to both the graph structure and the node attributes, if we set $\boldsymbol{x} = [\text{vec}(\boldsymbol{A}), \text{vec}(\boldsymbol{F})]$ we have to share a single set of radii (r_a, r_d) and flip probabilities (p_+, p_-) for both $\boldsymbol{A}$ and $\boldsymbol{F}$. However, it may be beneficial to specify different flip probabilities/radii. To achieve this we first independently calculate the set of regions $\mathcal{R}^{\boldsymbol{A}} = \{\dots, \mathcal{R}_q^{r_a^{\boldsymbol{A}}, r_d^{\boldsymbol{A}}}, \dots\}$ for $\boldsymbol{x} = \text{vec}(\boldsymbol{A})$ and $\mathcal{R}^{\boldsymbol{F}} = \{\dots, \mathcal{R}_q^{r_a^{\boldsymbol{F}}, r_d^{\boldsymbol{F}}}, \dots\}$ for $\boldsymbol{x} = \text{vec}(\boldsymbol{F})$ using different $(r_a^{\boldsymbol{A}}, r_d^{\boldsymbol{A}}, r_a^{\boldsymbol{F}}, r_d^{\boldsymbol{F}})$. Then to compute the certificate we form the regions $\mathcal{R}_{q,q'}$, where $\Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_{q,q'}) = \Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_q^{r_a^{\boldsymbol{A}}, r_d^{\boldsymbol{A}}}) \Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_{q'}^{r_a^{\boldsymbol{F}}, r_d^{\boldsymbol{F}}})$. The total number of $\mathcal{R}_{q,q'}$ regions is thus $(r_a^{\boldsymbol{A}} + r_d^{\boldsymbol{A}} + 1)(r_a^{\boldsymbol{F}} + r_d^{\boldsymbol{F}} + 1)$. Therefore, we pay only a small price in terms of complexity for the flexibility of specifying different radii. The size of the balls we can certify in practice is relatively small, e.g. the four radii are typically below 100 so the certificate is feasible. Note that this can be trivially extended to certify arbitrary groupings of $\boldsymbol{x}$ into subspaces with different radii/flip probabilities per subspace. However, the complexity quickly increases. For more details see § E.

6.2. Comparison with Existing Certificates for GNNs

There are only few certificates for GNNs: Zügner & Günnemann (2019b) can only handle attribute attacks, while Bojchevski & Günnemann (2019a) and Zügner & Günnemann (2020) only handle graph attacks. All three certificates apply only to node classification and a small class of models. Since their certificates hold for certain (base) classifiers, e.g. GCN (Kipf & Welling, 2017) or PPNP (Gasteiger et al., 2019), which tend to be less robust than their smoothed counterparts, we cannot make a fair comparison. Moreover, they rely on local budget constraints (at most given number of perturbations per node), and provide looser guarantees when using global budget only (since e.g. the global budget certificate for PPNP is NP-Hard). Nonetheless, we compare our certificate with these approaches in the appendix, and show that it provides comparable or better guarantees (see § F). Jia et al. (2020b)'s certificate is neither sparsity-aware nor efficient, and does not apply to GNNs.



(a) Heatmap for $p_+ = 0.01, p_- = 0.6$

(b) Trade-offs for different configurations of flip probabilities

Figure 2. Certifying attribute perturbations for GCN on Cora-ML. The heatmap on the left shows the ratio of certified nodes for different radii for $p_+ = 0.01, p_- = 0.6$. Darker cells correspond to higher certified ratio. On the right, we show the x and y-axis of the heatmap for different flip probabilities, i.e. $r_a = 0$, r_d varies (blue histogram) and $r_d = 0$, r_a varies (orange histogram) respectively.

7. Training

Our certificates hold regardless of how the base classifier f is trained. However, in order to classify the labeled example $(\boldsymbol{x}, y)$ correctly and robustly, g needs to consistently classify the noisy $\phi(\boldsymbol{x})$ as y. To ensure this, similar to previous work (Cohen et al., 2019), we train the base classifier with perturbed inputs, that is we apply $\phi(\cdot)$ during training which is akin to data augmentation with noise. We also investigated the approach suggested by Salman et al. (2019), where one directly trains the smoothed classifier g, rather than f. When the base classifier f is a GNN and the task is node-level classification, unlike Salman et al. (2019) we did not observe performance improvements with this strategy. See § I for a detailed comparison. One explanation could be that unlike image classifiers, where a single perturbation affects only a single image, a single perturbation of the graph can affect the predictions for many (potentially all) nodes.

Adversarial training. Even though adversarial training (Kurakin et al., 2017; Madry et al., 2018) is a *heuristic* defense adding adversarial examples during training tends to also improve the *certifiable* robustness (Wong & Kolter, 2018; Zügner & Günnemann, 2019b). This has also been demonstrated for smoothed classifiers (Salman et al., 2019), especially given access to additional unlabeled data (Carmon et al., 2019). However, adversarial training tends to be useful only with a sufficiently powerful attack. While for continuous data we can simply maximize the loss w.r.t. $\boldsymbol{x}$ via projected gradient descent (PGD) to find an adversarial example, PGD is not well suited for discrete data (Zügner et al., 2018). Therefore, we leave it as future work to develop suitable techniques for finding adversarial examples of g so we can employ adversarial training.

8. Experimental Evaluation

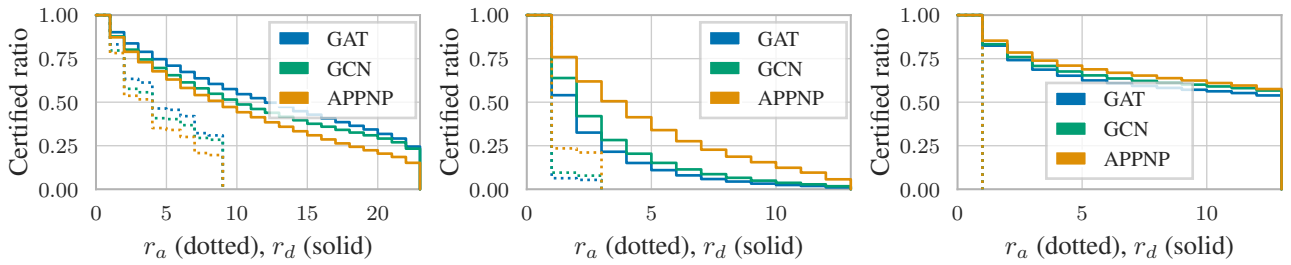
Our main goal is to answer the following research questions: (i) What are the trade-offs for different flip probabilities? (ii) What is the benefit of sparsity-awareness? (iii) How robust are different GNNs for different threat models? (iv) How large is the efficiency gain due to the improved partitioning?

8.1. Graph Neural Networks

Setup. We evaluate the certifiable robustness of three GNNs: GCN (Kipf & Welling, 2017), GAT (Velickovic et al., 2018) and APPNP (Gasteiger et al., 2019). We focus on the node classification task and the three scenarios we outlined in § 6. We demonstrate our claims on two datasets: Cora-ML ($n = 2995, e = 8416$) and PubMed ($n = 19717, e = 44324$) (Sen et al., 2008). The graphs are sparse, i.e. their number of edges $e \ll n^2$. See § H for further details about the data. For all experiments we set the confidence level $\alpha = 0.01$ and the number of samples for certification to 10^6 (10^5 for MNIST and ImageNet). We discuss how we choose hyperparameters and further implementation details in § J.

In Fig. 2 we show the *certified ratio* w.r.t. attribute perturbations for GCN on Cora-ML, i.e. the ratio of nodes which can be certified given the provided radii. The heatmap Fig. 2(a) investigates the trade-offs for certifying addition vs. deletion for $p_+ = 0.01, p_- = 0.6$. Since p_- is significantly higher we can certify a larger r_d radius. To ensure the model is robust to a few *worst-case* deletions, we need to ensure it is robust to many randomly deleted bits. The contour lines show the radii for which the certified ratio is at least 0.3 (0.5), i.e. at least 30 % (50 %) of all nodes can be certified.

In Fig. 2(b) we investigate the trade-offs for different degrees of smoothing. The y-axis shows the ratio of certified nodes. By decreasing the flip probabilities we can certify a larger portion of nodes but at lower radii, while increasing the probabilities allows for larger certified radii overall at the price of smaller ratios. This implies that in practice we can choose a suitable smoothing degree depending on the threat model since the difference in clean accuracy is at most 2 % for all cases (not shown here, see § K).



(a) Attributes, $p_+ = 0.01, p_- = 0.6$ (b) Graph, $p_+ = 0.001, p_- = 0.4$ (c) Graph, $p_+ = 0, p_- = 0.4$

Figure 3. Certifiable robustness for different models. Solid lines denote r_d (with $r_a = 0$) and dotted lines denote r_a (with $r_d = 0$).

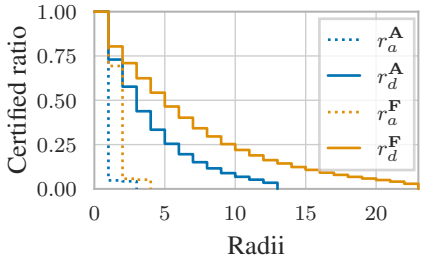


Figure 4. Certifying joint perturbations to the graph and attributes on Cora-ML.

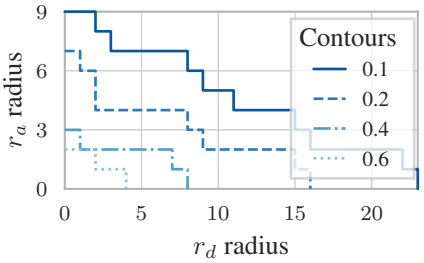


Figure 5. Certifying attribute perturbations on PubMed, $p_+ = 0.01, p_- = 0.6$.

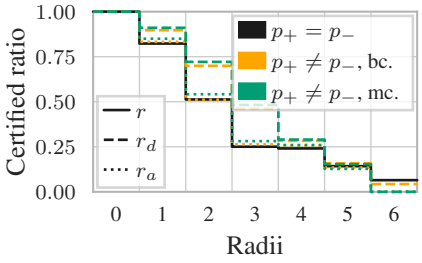


Figure 6. The benefit of our sparsity-aware certificates on binarized MNIST.

In Fig. 3 we compare the ratio of certified nodes for different GNNs and threat models. We can see that when perturbing the attributes (Fig. 3(a)) GAT is more robust than GCN and APPNP. On the other hand when perturbing the graph structure (Fig. 3(b)) the order is inverted, now APPNP is more robust than GCN and GAT. This highlights that different models have different robustness trade-offs.

We can further observe that certifying the attributes is in general easier compared to certifying the graph structure. Certifying edge addition is the most challenging scenario. Intuitively, since most nodes have a low degree (e.g. average degree on Cora-ML is 6) the attacker can easily misclassify them by adding a few edges to nodes from a different class.

Interestingly, if we consider the special case where ϕ only deletes edges (by setting $p_+ = 0$) the certified ratio for r_d is significantly improved (Fig. 3(c)). In practice, the observed graph $\boldsymbol{x}$ might already be corrupted. The certificate verifies that all $\tilde{\boldsymbol{x}}$ in the ball, including the unobserved clean graph, have the same prediction. From this point of view, by randomly deleting edges we are reducing the influence of adversarial edges which were potentially added. Since for many applications it is more feasible for the attacker to add rather than remove edges, certifying r_d is exactly the goal. In general, we see that none of the graph models are really robust, especially w.r.t. structure perturbations. We leave it as future work to make these approaches more reliable. In § K we also compare our binary-class vs. our multi-class certificate, the multi-class certificate is better in most cases.

Next, in Fig. 4 we show our method's ability to certify robustness against *joint* perturbations to both the graph structure and the node attributes. We set $p_+^{\boldsymbol{A}} = 2 \times 10^{-5}$, $p_-^{\boldsymbol{A}} = 0.4$ for the graph, and $p_+^{\boldsymbol{F}} = 2 \times 10^{-5}$, $p_-^{\boldsymbol{F}} = 0.6$ for the attributes. This combined scenario yields slightly worse certificates than when only allowing perturbations w.r.t. one input. Similar to single perturbations, we observe that certificates w.r.t. addition are especially hard to obtain.

Sparsity. Sparsity is crucial when certifying graphs. To show this we certify the attributes and set $p_+ = p_- = 0.1$ since $p_+ = 0.1$ is the largest value such that the clean accuracy is still reasonably high. We further compare with the randomized ablation certificate by Levine & Feizi (2019) which also does not consider sparsity. Their certificate depends on the number of retained pixels k, or in our case retained entries of the feature (adjacency) matrix. There is an inherent trade-off: lower value of k equals higher certified radius but worse classification accuracy. We set $k = 0.2d$ to the lowest value that still maintains reasonable accuracy.

For all certificates we compute the maximum certified radius averaged across all nodes which we denote with $\overline{r}$, and we show the results in Table 1. We can see that our sparsity-aware certificate is significantly better. The performance gap widens even further for graph perturbations (not shown here). We can conclude that sparsity-awareness is essential.

Table 1. Maximum certified radius averaged across nodes for attribute perturbations on GCN. SA stands for sparsity-aware.

	SA	$\overline{r_d}$	$\overline{r_a}$
$p_+ = p_- = 0.1$ (Lee et al., 2019)	n	2.03	2.03
$k = 0.2d$ (Levine & Feizi, 2019)	n	2.01	2.01
$p_+ = 0.01, p_- = 0.6$	y	9.99	3.38
$p_+ = 0.01, p_- = 0.8$	y	12.65	4.94
$p_+ = 0.00, p_- = 0.8$	y	18.66	2.14

Table 2. Certified accuracy for different radii on ImageNet. We show only the time to compute the certificate given $\underline{p_y(\boldsymbol{x})}$. Since $\phi(\cdot)$ is the same for all certificates the time to compute $\underline{p_y(\boldsymbol{x})}$ is also the same (and depends on the number of random samples). The numbers for the baselines are from the respective papers.

Certificate	Time	$r = 1$	$r = 3$	$r = 5$	$r = 7$
(Dvijotham et al., 2020)	28 ms	0.36	0.22	0.14	0
(Lee et al., 2019)	4 days	0.54	0.34	0.24	0.18
Ours	2.5 ms	0.54	0.34	0.24	0.18

Efficiency. The overall runtime to compute our certificate for *all* test nodes from the Cora-ML dataset using a GCN model is less than 25 minutes, or around 0.54 seconds per node. Most of the time is spent on $\underline{p_y(\boldsymbol{x})}$ and can be trivially reduced. Finally, to demonstrate that our certificates scales to large graphs we certify w.r.t. the attributes on the PubMed dataset which has over 19.5k nodes (results shown in Fig. 5).

8.2. Discretized Images

To show the general applicability of our method and the importance of sparsity and efficiency we also certify a CNN model on discretized images and compare with existing discrete certificates (see § J for details).

Sparsity. In Fig. 6 we compare our certificate with Lee et al. (2019) on binarized MNIST images. Since they have a single radius ($r_a = r_d$) we compare our radii by setting $r_d = 0$ and varying $r_a \geq 0$ (and similarly for $r_d \geq 0$). Their certificate is not sparsity-aware and is a special case of ours (we set $p_+ = p_- = 0.2$). For our certificates we can specify different flip probabilities (we set $p_+ = 0.1, p_- = 0.2$) which results in a significant increase in the certified ratio w.r.t. r_d and matching ratio w.r.t. r_a. We also compare our binary-class (b.c.) with our multi-class (m.c.) certificate (using Bonferroni correction) and we see that the tighter multi-class certificate tends to provide better guarantees.

Efficiency. In Table 2 we show the certified accuracy for discretized ImageNet data ($K = 256$) and $p_+ = p_- = 0.8$. We see that our certificate matches (Lee et al., 2019)'s but at a dramatically improved runtime, from 4 days to under a second. Dvijotham et al. (2020)'s certificate is efficient at the expense of tightness and obtains worse guarantees. Even though $\rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}}$ can be precomputed once and reused for different test inputs, without our improvement it would still be infeasible if d is slightly larger or varies (e.g. sequences).

9. Related Work

GNNs are a fundamental part of the modern machine learning landscape and have been successfully used for a variety of tasks from node-level classification (Defferrard et al., 2016; Kipf & Welling, 2017; Velickovic et al., 2018) to graph-level classification and regression (Gilmer et al., 2017; Gasteiger et al., 2020) across many domains. However, GNNs are highly sensitive to small adversarial perturbations (Zügner et al., 2018; Dai et al., 2018; Zügner & Günnemann, 2019a; Bojchevski & Günnemann, 2019b) – a common phenomenon observed for machine learning models in general (Szegedy et al., 2014; Goodfellow et al., 2015).

Beyond heuristic defenses (Kurakin et al., 2017; Madry et al., 2018; Xu et al., 2019; Entezari et al., 2020), which can be easily broken in practice (Athalye et al., 2018), certifiable robustness techniques provide provable guarantees (Hein & Andriushchenko, 2017; Wong & Kolter, 2018; Raghunathan et al., 2018). Most certificates either have scalability issues or rely on conservative relaxations. In contrast, the recently proposed randomized smoothing technique (Cohen et al., 2019; Lécuyer et al., 2019; Lee et al., 2019; Li et al., 2018) is a general approach which is relatively computationally inexpensive, yet provides good (probabilistic) guarantees.

Most work on randomized smoothing focuses on continuous data with a few exceptions that can tackle binary/discrete data. In contrast to our approach, these certificates are not sparsity-aware and are either computationally intractable or provide loose guarantees (see § 5.1). Moreover, our paper is the first to apply randomized smoothing to GNNs. There are only few certificates for graphs (Zügner & Günnemann, 2019b; Bojchevski & Günnemann, 2019a; Zügner & Günnemann, 2020) and as we discussed in § 1 and in § 6.2 they have serious limitations that we overcome.

10. Conclusion

We propose the first sparsity-aware certificate for discrete data based on the randomized smoothing framework. Our certificate can be efficiently computed and the complexity does not depend on the input size or the number of discrete categories. The sparsity-awareness and the drastically improved efficiency significantly broaden its applicability compared to previous work. We apply our certificate to study the robustness of different Graph Neural Networks and show that there are clear trade-offs across GNNs models.

Acknowledgements

This research was supported by the Deutsche Forschungsgemeinschaft (DFG) through the Emmy Noether grant GU 1409/2-1, and the TUM International Graduate School of Science and Engineering (IGSSE), GSC 81.

References

Athalye, A., Carlini, N., and Wagner, D. A. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pp. 274–283. PMLR, 2018.

Bojchevski, A. and Günnemann, S. Certifiable robustness to graph perturbations. In *NeurIPS*, pp. 8317–8328, 2019a.

Bojchevski, A. and Günnemann, S. Adversarial attacks on node embeddings via graph poisoning. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, Proceedings of Machine Learning Research. PMLR, 2019b.

Carlini, N. and Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 3–14, 2017.

Carmon, Y., Raghunathan, A., Schmidt, L., Duchi, J. C., and Liang, P. Unlabeled data improves adversarial robustness. In *NeurIPS*, pp. 11190–11201, 2019.

Chen, S. X. and Liu, J. S. Statistical applications of the poisson-binomial and conditional bernoulli distributions. *Statistica sinica*, pp. 875–892, 1997.

Clopper, C. J. and Pearson, E. S. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. Certified adversarial robustness via randomized smoothing. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1310–1320. PMLR, 2019.

Dai, H., Li, H., Tian, T., Huang, X., Wang, L., Zhu, J., and Song, L. Adversarial attack on graph structured data. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1123–1132. PMLR, 2018.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*, pp. 3837–3845, 2016.

Dvijotham, K. D., Hayes, J., Balle, B., Kolter, Z., Qin, C., Gyorgy, A., Xiao, K., Gowal, S., and Kohli, P. A framework for robustness certification of smoothed classifiers using f-divergences. In *ICLR*, 2020.

Entezari, N., Al-Sayouri, S. A., Darvishzadeh, A., and Papalexakis, E. E. All you need is low (rank): Defending against adversarial attacks on graphs. In *WSDM*, pp. 169–177. ACM, 2020.

Fernández, M. and Williams, S. Closed-form expression for the poisson-binomial probability density function. *IEEE Transactions on Aerospace and Electronic Systems*, 46 (2):803–817, 2010.

Gasteiger, J., Bojchevski, A., and Günnemann, S. Predict then propagate: Graph neural networks meet personalized pagerank. In *ICLR*, 2019.

Gasteiger, J., Groß, J., and Günnemann, S. Directional message passing for molecular graphs. In *International Conference on Learning Representations (ICLR)*, 2020.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1263–1272. PMLR, 2017.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Hein, M. and Andriushchenko, M. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *NIPS*, pp. 2266–2276, 2017.

Jia, J., Cao, X., Wang, B., and Gong, N. Z. Certified robustness for top-k predictions against adversarial perturbations via randomized smoothing. In *ICLR*, 2020a.

Jia, J., Wang, B., Cao, X., and Gong, N. Z. Certified robustness of community detection against adversarial structural perturbation via randomized smoothing. In *Proceedings of The Web Conference 2020*, pp. 2718–2724, 2020b.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial machine learning at scale. In *ICLR*, 2017.

Lécuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. Certified robustness to adversarial examples with differential privacy. In *IEEE Symposium on Security and Privacy*, pp. 656–672. IEEE, 2019.

Lee, G., Yuan, Y., Chang, S., and Jaakkola, T. S. Tight certificates of adversarial robustness for randomly smoothed classifiers. In *NeurIPS*, pp. 4911–4922, 2019.

Levine, A. and Feizi, S. Robustness certificates for sparse adversarial attacks by randomized ablation. *CoRR*, abs/1911.09272, 2019.

Li, B., Chen, C., Wang, W., and Carin, L. Second-order adversarial attack and certifiable robustness. *CoRR*, abs/1809.03113, 2018.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Monti, F., Frasca, F., Eynard, D., Mannion, D., and Bronstein, M. M. Fake news detection on social media using geometric deep learning. *arXiv preprint arXiv:1902.06673*, 2019.

Raghunathan, A., Steinhardt, J., and Liang, P. Semidefinite relaxations for certifying robustness to adversarial examples. In *NeurIPS*, pp. 10900–10910, 2018.

Rhee, S., Seo, S., and Kim, S. Hybrid approach of relation network and localized graph convolutional filtering for breast cancer subtype classification. In *IJCAI*, pp. 3527–3534. ijcai.org, 2018.

Salman, H., Li, J., Razenshteyn, I. P., Zhang, P., Zhang, H., Bubeck, S., and Yang, G. Provably robust deep learning via adversarially trained smoothed classifiers. In *NeurIPS*, pp. 11289–11300, 2019.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., and Eliassi-Rad, T. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

Shu, K., Mahudeswaran, D., Wang, S., and Liu, H. Hierarchical propagation networks for fake news detection: Investigation and exploitation. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, pp. 626–637, 2020.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. In *ICLR*, 2014.

Tocher, K. D. Extension of the neyman-pearson theory of tests to discontinuous variates. *Biometrika*, 37(1/2): 130–144, 1950.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *ICLR*, 2018.

Wang, J., Wen, R., Wu, C., Huang, Y., and Xion, J. Fdgars: Fraudster detection via graph convolutional networks in online app review system. In *WWW (Companion Volume)*, pp. 310–316. ACM, 2019.

Wong, E. and Kolter, J. Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5283–5292. PMLR, 2018.

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 *IJCAI*, pp. 3961–3967. ijcai.org, 2019.

Zügner, D. and Günnemann, S. Adversarial attacks on graph neural networks via meta learning. In *ICLR*, 2019a.

Zügner, D. and Günnemann, S. Certifiable robustness and robust training for graph convolutional networks. In *KDD*, pp. 246–256. ACM, 2019b.

Zügner, D. and Günnemann, S. Certifiable robustness of graph convolutional networks under structure perturbations. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2020. ACM.

Zügner, D., Akbarnejad, A., and Günnemann, S. Adversarial attacks on neural networks for graph data. In *KDD*, pp. 2847–2856. ACM, 2018.

A. Proofs

Proof (Prop. 1). First we show that the regions are disjoint. Let $\boldsymbol{z} \in \mathcal{R}_i^{r_a,r_d}$, and $\boldsymbol{z} \in \mathcal{R}_j^{r_a,r_d}$ for some $i \neq j$. From the definition of a region it follows that $\|\boldsymbol{x}_\mathcal{C} - \boldsymbol{z}_\mathcal{C}\|_0 = i$ and $\|\boldsymbol{x}_\mathcal{C} - \boldsymbol{z}_\mathcal{C}\|_0 = j$. This can be true only if $i = j$ which is a contradiction. Therefore, $\boldsymbol{z}$ cannot belong to two different regions. For any $\boldsymbol{z}$ and $\boldsymbol{x}$, $\|\boldsymbol{x}_\mathcal{C} - \boldsymbol{z}_\mathcal{C}\|_0 \in \{0, \dots, r_a + r_d\}$ since the $\|\cdot\|_0$ (Hamming) distance between two $|\mathcal{C}|$-dimensional vectors has the range $\{0, \dots, |\mathcal{C}|\}$. Thus, any $\boldsymbol{z}$ must land in some region $\mathcal{R}_q^{r_a,r_d}$ with $q \leq |\mathcal{C}|$, and for any $q > |\mathcal{C}| = r_a + r_d$ we have $\mathcal{R}_q^{r_a,r_d} = \emptyset$. Therefore, $\mathcal{X} = \bigcup_{q=0}^{q=\infty} \mathcal{R}_q^{r_a,r_d} = \bigcup_{q=0}^{q=r_a+r_d} \mathcal{R}_q^{r_a,r_d}$. □

Proof (Prop. 2). For any $\boldsymbol{x}, \tilde{\boldsymbol{x}} \in \mathcal{S}_{r_a,r_d}(\boldsymbol{x})$, and $\mathcal{R}_q^{r_a,r_d}$:

$$\Pr\left(\phi(\boldsymbol{x}) \in \mathcal{R}_q^{r_a,r_d}\right) = \Pr\left(\|\boldsymbol{x}_\mathcal{C} - \phi(\boldsymbol{x})_\mathcal{C}\|_0 = q\right) = \Pr\left(\sum_{i\in\mathcal{C}} \mathbb{I}[\boldsymbol{x}_i \neq \phi(\boldsymbol{x})_i] = q\right) = \Pr\left(\sum_{i\in\mathcal{C}} \epsilon_i = q\right) \tag{7}$$

where $\epsilon_i \sim \text{Ber}(p = p_-^{\boldsymbol{x}_i} p_+^{(1-\boldsymbol{x}_i)})$. The first equality in Eq. 7 follows from the definition of a region, and the last equality follows from the definition of $\phi(\cdot)$. Since $\boldsymbol{x} \in \mathcal{R}_q^{r_a,r_d}$ we have $\sum_{i\in\mathcal{C}} \boldsymbol{x}_i = r_d$ and $\sum_{i\in\mathcal{C}} 1 - \boldsymbol{x}_i = r_a$. Therefore, $\sum_{i\in\mathcal{C}} \epsilon_i \sim Q$ where $Q = \text{PB}([p_+, r_a][p_-, r_d])$. □

Proof (Prop. 3). For any $\boldsymbol{z} \in \mathcal{R}_q^{r_a,r_d}$, by definition it holds $\|\boldsymbol{x}_\mathcal{C} - \boldsymbol{z}_\mathcal{C}\|_0 = q$. Let $q_- = \sum_{i=1}^d \mathbb{I}(\boldsymbol{x}_i - 1 = \boldsymbol{z}_i)$ and $q_+ = \sum_{i=1}^d \mathbb{I}(\boldsymbol{x}_i + 1 = \boldsymbol{z}_i)$, so $q = q_+ + q_-$. We have:

$$\begin{aligned}
\eta_q^{r_a,r_d} &= \frac{\Pr(\phi(\boldsymbol{x}) = \boldsymbol{z})}{\Pr(\phi(\tilde{\boldsymbol{x}}) = \boldsymbol{z})} \\
&= \frac{\prod_{i\in\tilde{\mathcal{C}}} \Pr(\phi(\boldsymbol{x})_i = \boldsymbol{z}_i) \prod_{j\in\mathcal{C}} \Pr(\phi(\boldsymbol{x})_j = \boldsymbol{z}_j)}{\prod_{i\in\tilde{\mathcal{C}}} \Pr(\phi(\tilde{\boldsymbol{x}})_i = \boldsymbol{z}_i) \prod_{j\in\mathcal{C}} \Pr(\phi(\tilde{\boldsymbol{x}})_j = \boldsymbol{z}_j)} \\
&= \frac{\prod_{j\in\mathcal{C}} \Pr(\phi(\boldsymbol{x})_j = \boldsymbol{z}_j)}{\prod_{j\in\mathcal{C}} \Pr(\phi(\tilde{\boldsymbol{x}})_j = \boldsymbol{z}_j)} \\
&= \frac{p_-^{q_-}(1-p_-)^{r_d-q_-} p_+^{q_+}(1-p_+)^{r_a-q_+}}{p_-^{r_a-q_+}(1-p_-)^{q_+} p_+^{r_d-q_-}(1-p_+)^{q_-}} \\
&= p_-^{q-r_a}(1-p_-)^{r_d-q} p_+^{q-r_d}(1-p_+)^{r_a-q} \\
&= \left[\frac{p_+}{1-p_-}\right]^{q-r_d} \left[\frac{p_-}{1-p_+}\right]^{q-r_a}
\end{aligned}$$

Where the second equality holds since ϕ is independent per dimension, and the third equality holds since $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ agree on $\tilde{\mathcal{C}}$. Plugging in the definition of ϕ and rearranging we obtain $\eta_q^{r_a,r_d}$. Thus, the ratio is constant for any $\boldsymbol{z} \in \mathcal{R}_q^{r_a,r_d}$. Now we show that the ratio is a monotonic function of q:

$$\begin{aligned}
\eta_q^{r_a,r_d} &= \left[\frac{p_+}{1-p_-}\right]^{q-r_d} \left[\frac{p_-}{1-p_+}\right]^{q-r_a} \\
&= C \cdot \left[\frac{p_+p_-}{p_+p_- + \underbrace{1-(p_+-p_-)}_{:=u}}\right]^q
\end{aligned} \tag{8}$$

Here $C = \left[\frac{p_+p_-}{(1-p_+)(1-p_-)}\right]^{-(r_a+r_d)} \geq 0$ is a non-negative constant that does not depend on q since $p_+, p_- \in [0, 1]$, and hence does not change the monotonicity. We have three cases: (i) if $p_+ + p_- < 1$ then $u > 0$ in the denominator of Eq. 8, the ratio is < 1 and thus a decreasing function of q; (ii) if $p_+ + p_- = 1$ then $u = 0$ and the ratio becomes $C \cdot 1^q$, i.e. constant; (iii) if $p_+ + p_- > 1$ then $u < 0$, the ratio is > 1 and thus an increasing function of q. □

B. Multi-Class Certificates

For the multi-class certificate our goal is to solve the following optimization problem:

$$\begin{aligned}
&\mu_{\boldsymbol{x},\tilde{\boldsymbol{x}}}(p_1(\boldsymbol{x}), \dots, p_\mathcal{Y}(\boldsymbol{x}), y^*) \\
&= \min_{h\in\mathcal{H}} \Pr(h(\phi(\tilde{\boldsymbol{x}})) = y^*) - \max_{y\neq y^*} \Pr(h(\phi(\tilde{\boldsymbol{x}})) = y) \\
&\text{s.t.} \quad \Pr(h(\phi(\boldsymbol{x})) = y^*) = p_{y^*} \\
&\text{and } \Pr(h(\phi(\boldsymbol{x})) = y) = p_y, \quad y \neq y^*
\end{aligned} \tag{9}$$

where y^* is the (predicted or ground-truth) class we want to certify. Similar to before computing $p_y(\boldsymbol{x})$ exactly is difficult, thus we compute a lower bound $\underline{p_{y^*}(\boldsymbol{x})}$ for y^* and an upper bound $\overline{p_y(\boldsymbol{x})}$ for all other y. Since we are conservative in the estimates, the solution to Eq. 9 using these bounds yields a valid certificate. Estimating the lower and upper bounds from Monte Carlo samples such that they hold simultaneously with confidence level α requires some care. Specifically, we have to correct for multiple testing error. Similar to Jia et al. (2020a) we estimate each bound individually using a Clopper-Pearson Bernoulli confidence interval with confidence $\frac{\alpha}{C}$ where $C = |\mathcal{Y}|$ is the number of classes and use Bonferroni correction to guarantee with confidence of α that the estimates hold simultaneously.

The problem in Eq. 9 is valid if $\underline{p_{y^*}(\boldsymbol{x})} + \overline{p_{\tilde{y}}(\boldsymbol{x})} < 1$. The binary-class certificate assumes that $\overline{p_{\tilde{y}}(\boldsymbol{x})} = 1 - \underline{p_{y^*}(\boldsymbol{x})}$. From here we can directly conclude that the multi-class certificate is in principle always equal or better than the binary certificate, and in particular the improvement can only occur when $\underline{p_{y^*}(\boldsymbol{x})} + \overline{p_{\tilde{y}}(\boldsymbol{x})} < 1$. Note that, however, the value of $\underline{p_{y^*}(\boldsymbol{x})}$ will be lower for the multi-class certificate compared to the binary-class certificate due to the Bonferroni correction. This implies that in some cases the binary-class certificate can yield a higher certified radius. For the majority of our experiments the multi-class certificate was better.

Now, given an input $\boldsymbol{x}$ and a perturbation set $\mathcal{B}_{r_a,r_d}(\boldsymbol{x})$ if it holds that: $\min_{\tilde{\boldsymbol{x}}\in\mathcal{B}(\boldsymbol{x})} \mu_{\boldsymbol{x},\tilde{\boldsymbol{x}}}(p_1(\boldsymbol{x}), \dots, p_\mathcal{Y}(\boldsymbol{x}), y^*) > 0$ we can guarantee that classification margin for the worst-case classifier is always bigger than 0 for all $\tilde{\boldsymbol{x}} \in \mathcal{B}(\boldsymbol{x})$. This implies that $g(\boldsymbol{x}) = g(\tilde{\boldsymbol{x}}) = y^*$ for any input within the ball, i.e. $\boldsymbol{x}$ is certifiably robust. Compare this to the previous certificate where we had to verify whether $\rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}}(p^*, y^*) > 0.5$ which was not tight for $|\mathcal{Y}| > 2$.

Similar to before, Eq. 9 is equivalent to the following LP:

$$\min_{\boldsymbol{h},\boldsymbol{t}} \boldsymbol{h}^T\tilde{\boldsymbol{r}} - \boldsymbol{t}^T\tilde{\boldsymbol{r}} \tag{10}$$
$$\text{s.t.} \quad \boldsymbol{h}^T\boldsymbol{r} = \underline{p_{y^*}(\boldsymbol{x})}, \qquad \boldsymbol{t}^T\boldsymbol{r} = \overline{p_{\tilde{y}}(\boldsymbol{x})},$$
$$0 \leq \boldsymbol{h} \leq 1, \qquad 0 \leq \boldsymbol{t} \leq 1$$

where $\tilde{y} = \max_{y \neq y^*} \overline{p_y(\boldsymbol{x})}$ is the class with the second highest number of majority votes after y^*. The proof is analogous to the proof of Lemma 2 in Lee et al. (2019).

The exact solution to the LP is easily obtained with another greedy algorithm: first sort the regions such that $c_1 \geq c_2 \geq \cdots \geq c_I$, then iteratively assign $\boldsymbol{h}_i = 1$ in decreasing order for all regions $\mathcal{R}_i$ until the constraint $\underline{p_{y^*}(\boldsymbol{x})}$ is met. Finally, iteratively assign $\boldsymbol{t}_j = 1$ now in increasing order for all regions $\mathcal{R}_j$ until the constraint $\overline{p_{\tilde{y}}(\boldsymbol{x})}$ is met.

C. Special Cases for Flipping Probabilities

We derive the regions of constant likelihood ratio for the case $p_+ = 0$ and $p_- > 0$. There are only three regions which we have to consider. First note that there is only one set of vectors $\boldsymbol{z}$ which can be reached by both $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ when applying the randomization ϕ and these are the vectors which have all valid (reachable via deletion) configurations of ones and zeros in $\tilde{\mathcal{C}}$ and all zeros in $\mathcal{C}$. This holds since $\boldsymbol{x}_{\mathcal{C}}$ and $\tilde{\boldsymbol{x}}_{\mathcal{C}}$ are complementary and we can only delete edges. See Fig. 1 for an illustration. Denoting this region with $\mathcal{R}_1$ we have that $\Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_1) = p_-^{r_d}$ and $\Pr(\phi(\tilde{\boldsymbol{x}}) \in \mathcal{R}_1) = p_-^{r_a}$ since we need to successfully delete all edges.

The second region $\mathcal{R}_2$ corresponds to the case where we flip less than r_d bits in $\boldsymbol{x}$ and this happens with probability $\Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_2) = 1 - p_-^{r_d}$. By definition the vectors in the intersection reachable by both $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ are all in $\mathcal{R}_1$, thus $\Pr(\phi(\tilde{\boldsymbol{x}}) \in \mathcal{R}_2) = 0$. Finally, the third region $\mathcal{R}_3$ corresponds to the case where we flip less than r_a bits in $\tilde{\boldsymbol{x}}$, we have $\Pr(\phi(\tilde{\boldsymbol{x}}) \in \mathcal{R}_3) = 1 - p_-^{r_a}$ and $\Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_3) = 0$. For the binary class certificate we can ignore any regions $\mathcal{R}_i$ where $\Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_i) = 0$, so the only two valid regions are $\mathcal{R}_1$ and $\mathcal{R}_2$. However, for our multi-class certificate all three regions are necessary.

The case for $p_+ > 0, p_- = 0$ is analogous. We have: $\Pr(\phi(\boldsymbol{x}) \in \mathcal{R}'_1) = p_+^{r_a}$ and $\Pr(\phi(\tilde{\boldsymbol{x}}) \in \mathcal{R}'_1) = p_+^{r_d}$ for the first region; $\Pr(\phi(\boldsymbol{x}) \in \mathcal{R}'_2) = 1 - p_+^{r_a}$ and $\Pr(\phi(\tilde{\boldsymbol{x}}) \in \mathcal{R}'_2) = 0$ for the second region; $\Pr(\phi(\tilde{\boldsymbol{x}}) \in \mathcal{R}'_3) = 1 - p_+^{r_d}$ and $\Pr(\phi(\boldsymbol{x}) \in \mathcal{R}'_3) = 0$ for the third region.

D. Traversal of Regions

As we discussed in § 4.3 we can efficiently compute $\rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}}$ by directly visiting the regions $\mathcal{R}_q^{r_a,r_d}$ in decreasing order w.r.t. the ratio $\eta_q^{r_a,r_d}$ without sorting. The pseudo-code is given in Algorithm 1 and corresponds to the greedy algorithm for solving the LP in Eq. 4 and thus Eq. 3. Once $\rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}}$ is computed we simply have to check whether $\rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}} > 0.5$ to certify the input $\boldsymbol{x}$ w.r.t. the given radii r_a and r_d. The algorithm for the multi-class certificate $\mu_{\boldsymbol{x},\tilde{\boldsymbol{x}}}$ is similar.

Algorithm 1 Compute $\rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}}$ # special cases omitted

Input: $p_+, p_-, r_a, r_d, \underline{p_{y^*}(\boldsymbol{x})}$
if $p_+ + p_- < 1$ **then**
  start $= 0$, end $= r_a + r_d$
else
  start $= r_a + r_d$, end $= 0$
end if
Initialize $p = 0$, $\rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}} = 0$.
for $q =$ start **to** end **do**
  Compute $\eta_q^{r_a,r_d}$ ratio using Prop. 3
  Compute $\text{PB}(q;\cdot) = \Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_q^{r_a,r_d})$ as in § 4.4
  $\Pr(\phi(\tilde{\boldsymbol{x}}) \in \mathcal{R}_q^{r_a,r_d}) = \text{PB}(q;\cdot)/\eta_q^{r_a,r_d}$
  if $p + \Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_q^{r_a,r_d}) > \underline{p_{y^*}(\boldsymbol{x})}$ **then**
    break
  else
    $p = p + \Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_q^{r_a,r_d})$
    $\rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}} = \rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}} + \Pr(\phi(\tilde{\boldsymbol{x}}) \in \mathcal{R}_q^{r_a,r_d})$
  end if
end for
if $\underline{p_{y^*}(\boldsymbol{x})} - p > 0$ **then**
  $\rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}} = \rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}} + (\underline{p_{y^*}(\boldsymbol{x})} - p)/\eta_q^{r_a,r_d}$
end if
Output: $\rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}}$

E. Joint Certificates

As we discussed in § 6.1 it may be beneficial to specify different flip probabilities and radii for the graph and attributes. Let $\boldsymbol{x}^{\boldsymbol{A}} = \text{vec}(\boldsymbol{A}) \in \{0,1\}^{n \times n}$ and $\boldsymbol{x}^{\boldsymbol{F}} = \text{vec}(\boldsymbol{F}) \in \{0,1\}^{n \times m}$ denote the flattened adjacency and feature matrix respectively. Let $\boldsymbol{x} = [\boldsymbol{x}^{\boldsymbol{A}}, \boldsymbol{x}^{\boldsymbol{F}}] \in \mathcal{X}^{\boldsymbol{A},\boldsymbol{F}}$ where $\mathcal{X}^{\boldsymbol{A},\boldsymbol{F}} = \{0,1\}^{n \times n + n \times m}$. We apply the randomization schemes independently: for the graph $\phi(\boldsymbol{x}^{\boldsymbol{A}})$ with $p_+^{\boldsymbol{A}}, p_-^{\boldsymbol{A}}$, and for the attributes $\phi(\boldsymbol{x}^{\boldsymbol{F}})$ with $p_+^{\boldsymbol{F}}, p_-^{\boldsymbol{F}}$.

We define the region:

$$\mathcal{R}_{q,q'}^{r_a^{\boldsymbol{A}},r_d^{\boldsymbol{A}},r_a^{\boldsymbol{F}},r_d^{\boldsymbol{F}}} = \{\boldsymbol{z} = [\boldsymbol{z}^{\boldsymbol{A}}, \boldsymbol{z}^{\boldsymbol{F}}] \in \mathcal{X}^{\boldsymbol{A},\boldsymbol{F}} : \boldsymbol{z}^{\boldsymbol{A}} \in \mathcal{R}_q^{r_a^{\boldsymbol{A}},r_d^{\boldsymbol{A}}}, \boldsymbol{z}^{\boldsymbol{F}} \in \mathcal{R}_{q'}^{r_a^{\boldsymbol{F}},r_d^{\boldsymbol{F}}}\}$$

where $\mathcal{R}_q^{r_a^{\boldsymbol{A}},r_d^{\boldsymbol{A}}}$ and $\mathcal{R}_{q'}^{r_a^{\boldsymbol{F}},r_d^{\boldsymbol{F}}}$ are defined similar to before. We have that the regions $\{\mathcal{R}_{0,0}^{r_a^{\boldsymbol{A}},r_d^{\boldsymbol{A}},r_a^{\boldsymbol{F}},r_d^{\boldsymbol{F}}}, \ldots, \mathcal{R}_{r_a^{\boldsymbol{A}}+r_d^{\boldsymbol{A}},r_a^{\boldsymbol{F}}+r_d^{\boldsymbol{F}}}^{r_a^{\boldsymbol{A}},r_d^{\boldsymbol{A}},r_a^{\boldsymbol{F}},r_d^{\boldsymbol{F}}}\}$ partition the space $\mathcal{X}^{\boldsymbol{A},\boldsymbol{F}}$. This follows directly due to the independence and the fact that the regions w.r.t. graph/attributes partition their respective spaces. The total number of regions is thus $(r_a^{\boldsymbol{A}} + r_d^{\boldsymbol{A}} + 1)(r_a^{\boldsymbol{F}} + r_d^{\boldsymbol{F}} + 1)$.

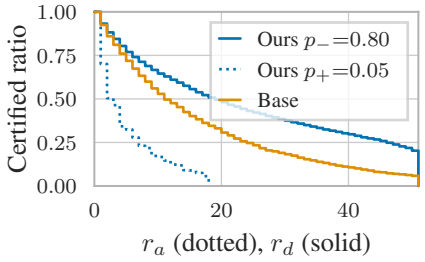


Figure 7. Comparison between our certificate of the smoothed GCN classifier and Zügner & Günnemann (2019b)'s certificate of the base GCN classifier. We are certifying w.r.t. the attributes on Cora-ML. Solid lines denote r_d (with $r_a = 0$) and dotted lines denote r_a (with $r_d = 0$).

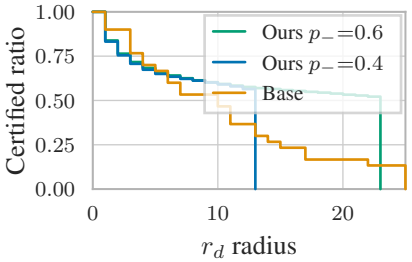


Figure 8. Comparison between our certificate of the smoothed PPNP classifier and Bojchevski & Günnemann (2019a)'s certificate of the base PPNP classifier. We are certifying edge deletion on Cora-ML. Our certificate is significantly better despite the fact that we are certifying undirected edges.

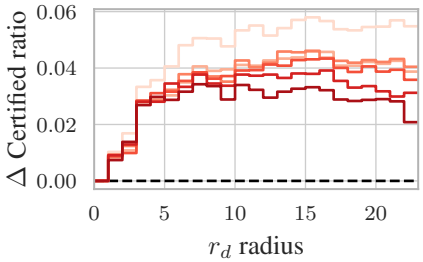


Figure 9. The difference (Δ) in the certificate ratio relative to $m = 0$ (standard training, dashed black line). The color gradient denotes $m \in \{1, 5, 10, 25, 50, 100\}$ with darker colors corresponding to higher m. The difference is relatively small overall, and $m = 1$ (lightest color) is best.

As before we can compute $\Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_{q,q'}^{r_a^{\boldsymbol{A}}, r_d^{\boldsymbol{A}}, r_a^{\boldsymbol{F}}, r_d^{\boldsymbol{F}}}) = \Pr(\phi(\boldsymbol{x}^{\boldsymbol{A}}) \in \mathcal{R}_q^{r_a^{\boldsymbol{A}}, r_d^{\boldsymbol{A}}}) \cdot \Pr(\phi(\boldsymbol{x}^{\boldsymbol{F}}) \in \mathcal{R}_{q'}^{r_a^{\boldsymbol{F}}, r_d^{\boldsymbol{F}}})$. Similarly we have for the ratio:

$$\eta_{q,q'}^{r_a^{\boldsymbol{A}}, r_d^{\boldsymbol{A}}, r_a^{\boldsymbol{F}}, r_d^{\boldsymbol{F}}} = \frac{\Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_{q,q'}^{r_a^{\boldsymbol{A}}, r_d^{\boldsymbol{A}}, r_a^{\boldsymbol{F}}, r_d^{\boldsymbol{F}}})}{\Pr(\phi(\tilde{\boldsymbol{x}}) \in \mathcal{R}_{q,q'}^{r_a^{\boldsymbol{A}}, r_d^{\boldsymbol{A}}, r_a^{\boldsymbol{F}}, r_d^{\boldsymbol{F}}})} = \eta_q^{r_a^{\boldsymbol{A}}, r_d^{\boldsymbol{A}}} \cdot \eta_{q'}^{r_a^{\boldsymbol{F}}, r_d^{\boldsymbol{F}}}$$

The above directly follows from the definition of the regions and because $\phi(\boldsymbol{x}^{\boldsymbol{A}})$ is independent of $\phi(\boldsymbol{x}^{\boldsymbol{F}})$. Given the values of $\eta_{q,q'}$ and $\Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_{q,q'})$ for all q, q' we can again apply the greedy algorithm to compute $\rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}}$. Note that this can be trivially extended to certify arbitrary groupings of $\boldsymbol{x}$ into subspaces with different radii/flip probabilities per subspace, however, the complexity quickly increases and in general the number of regions will be $\mathcal{O}((r_a^{\max} + r_d^{\max} + 1)^v)$ where v is the number of groupings and $r_a^{\max}, r_d^{\max}$ are the maximum radii across the groupings.

F. Existing Graph Certificates Comparison

We compare our certificates with the only two existing works for certifying GNNs: Zügner & Günnemann (2019b)'s certificate which can only handle attacks on $\boldsymbol{F}$ and works for the GCN model (Kipf & Welling, 2017); and Bojchevski & Günnemann (2019a)'s certificate which can only handle attacks on $\boldsymbol{A}$ and works for a small class of models where the predictions are a linear function of (personalized) PageRank.

Both certificates specify local (per node) and global budgets/constraints, while our radii correspond to having only global budget. Therefore, to ensure a fair comparison we set their local budgets to be equal to their global budget which is equal to one of our radii, i.e. $q = Q = r_*$ for Zügner & Günnemann (2019b)'s certificate, and $b_v = B = r_*$ for Bojchevski & Günnemann (2019a)'s certificate. As we discussed in § 6.2 we can only compare the certified robustness of the *base* classifier (existing certificates) versus the *smoothed* variant of the same classifier (our certificate).

Zügner & Günnemann (2019b)'s certificate does not distinguish between adding/deleting bits in the attributes so we compute a single radius corresponding to the total number of perturbations. For our certificate we evaluate two cases: (i) $r_d = 0$ and r_a varies; (ii) $r_a = 0$ and r_d varies. We use a different configuration of flip probabilities for each case. The certified ratio for all test nodes is shown on figure Fig. 7. We see that our certificate is slightly better w.r.t. deletion and worse w.r.t. addition.

For Bojchevski & Günnemann (2019a)'s certificate we randomly select 50 test nodes to certify since solving their relaxed QCLP with global budget is computationally expensive. We evaluate the robustness of the (A)PPNP model, and we focus on edge removal since their global budget certificate for edge addition took more than 12h to complete. That is, we configure the set of fragile edges $\mathcal{F}$ to contain only the existing edges (except the edges along the minimum spanning tree which are fixed). The results for different values of p_- (for $p_+ = 0$) are show in Fig. 8. We see that we can certify significantly more nodes, especially as we increase the radius. Note that the effective certified radius for our approach is double of what is shown in Fig. 8 since we are certifying undirected edges, while Bojchevski & Günnemann (2019a)'s certificate is w.r.t. directed edges.

G. Graph Classification

For most experiments we focused on the node-level classification task. However, our certificate can be trivially adapted for the graph-level classification task. Currently, there are no

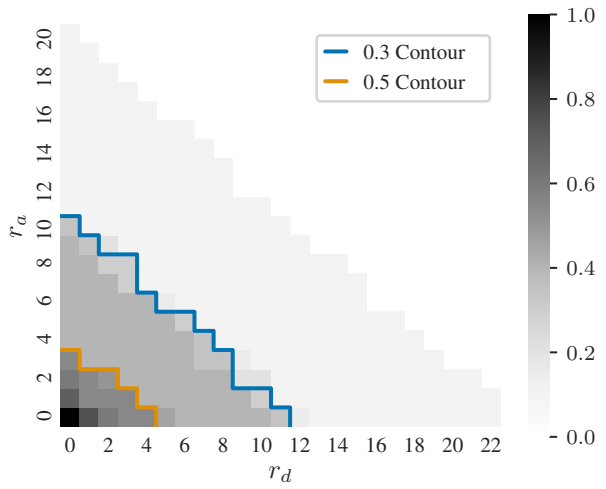


Figure 10. Certifying graph-level classification w.r.t. perturbations of the graph structure on the MUTAG dataset. We set $p_+ = 0.2$ and $p_- = 0.4$. We can certify a high ratio of graphs for r_a and r_d.

other existing certificate that can handle this scenario. Given any classifier f that takes a graph G_i as an input and outputs (a distribution over) graph-level classes, we can form the smoothed classifier g by randomly perturbing G_i, e.g. by applying ϕ on $\boldsymbol{x} = \text{vec}(\boldsymbol{A}_i)$ where $\boldsymbol{A}_i$ is the adjacency matrix of the graph G_i. Then, we certify g simply by calculating $\rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}}$ or $\mu_{\boldsymbol{x},\tilde{\boldsymbol{x}}}$. The certificates are still efficient to compute and independent of the graph size.

To demonstrate the generality of our certificate we train GIN on the MUTAG dataset, which consists of 188 graphs corresponding to chemical compounds. The graphs are divided into two classes according to their mutagenic effect on bacteria. The results are shown in Fig. 10. We see that we can certify a high ratio of graphs for both r_a and r_d. Similar results hold when perturbing the node features.

H. Datasets

To evaluate our graph certificate we use two well-known citation graph datasets: Cora-ML ($n = 2995, e = 8416, d = 2879$) and PubMed ($n = 19717, e = 44324, d = 500$) (Sen et al., 2008). The nodes correspond to papers, the edges correspond to citations between them, and the node features correspond to bag-of-words representations of the papers' abstracts. For all experiments we standardize the graphs, i.e. we make the graphs undirected and we select only the nodes that belong to the largest connected component. After standardization we have: Cora-ML ($n = 2810, e = 7981, d = 2879$) and PubMed ($n = 19717, e = 44324, d = 500$). We can see that both graphs are very sparse with the number of edges e being only a small fraction of the total number of possible edges n^2. Namely 0.1066% of all edges for Cora-ML and 0.0114% for PubMed. Since the node features are bag-of-words representations we see high sparsity for the attributes as well. Namely, 1.7588% for Cora-ML and 10.0221% for PubMed. For our general certificate experiments, similar to Lee et al. (2019) we binarize the MNIST dataset by setting the threshold at 0.5, and we discretize the ImageNet images to $K = 256$ values.

I. Training

To investigate the effect of smooth training (Salman et al., 2019) on certified robustness we approximate the smoothed probability $\boldsymbol{g}_y(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{x}' \sim \phi(\boldsymbol{x})}[f(\boldsymbol{x}')_y]$ for class y with m Monte Carlo samples $\boldsymbol{g}_y(\boldsymbol{x}) \approx \sum_{i=1}^{m} f(\boldsymbol{x}^{(i)})_y$, and we compute the cross-entropy loss with $l(\boldsymbol{g}(\boldsymbol{x}), y)$. Note that $m = 1$ is equivalent to training f with noisy inputs. We vary the number of Monte Carlo samples m we use during training for a fixed value of $p_+ = 0.01, p_- = 0.6$. Fig. 9 shows the results when perturbing the attributes on Cora-ML using GCN as a base classifier. Specifically, we show the difference (Δ) in the certified ratio relative to standard (non-smoothed) training, i.e. $m = 0$. We see that including the perturbations during training ($m > 0$) is consistently better than standard training ($m = 0$). The difference for different values of m is relatively small overall, with $m = 1$ being the best. Therefore, for all experiments we set $m = 1$.

J. Hyperparameters

For node classification, for all GNN models we randomly select 20 nodes from each class for the training set, and 20 nodes for the validation set. We train the models for a maximum of 3000 epochs with a fixed learning rate of 10^{-3} and patience of 50 epochs for early stopping. We optimize the parameters with Adam and use a weight decay of 10^{-3}. For GCN and APPNP we use a single hidden layer of size 64, and we set the hidden size for GAT to 8 and use 8 heads to match the number of trainable parameters. For MNIST and ImageNet we use the standard train/validation/test split, and we train a CNN classifier with the same configuration as described in Lee et al. (2019). We set $\alpha = 0.01$, and use 10^3 and 10^6 samples (10^5 for MNIST and ImageNet) to estimate y^* and $\underline{p_{y^*}(\boldsymbol{x})}$ respectively. For all experiments, we use our multi-class certificate since it yields slightly higher certified ratios compared to the binary-class certificate (see § K). Note that to certify an input w.r.t. $\mathcal{B}_{r_a,r_d}(\boldsymbol{x})$ it is sufficient to certify w.r.t. $\mathcal{S}_{r_a,r_d}(\boldsymbol{x})$. In practice, we compute the maximum r_a and r_d for a given $\underline{p_{y^*}(\boldsymbol{x})}$ and $\overline{p_{\bar{y}}(\boldsymbol{x})}$ such that the input is certifiably robust. Whenever the number of majority votes is the same for several inputs, they have the same $\underline{p_{y^*}(\boldsymbol{x})}$ and $\overline{p_{\bar{y}}(\boldsymbol{x})}$ so we only need to compute the maximum radii once to certify all of them.

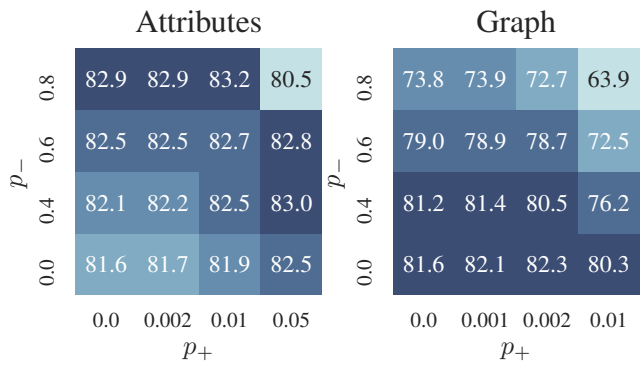


Figure 11. Clean accuracy for different flip probabilities when perturbing the attributes on Cora-ML using GCN as a base classifier.

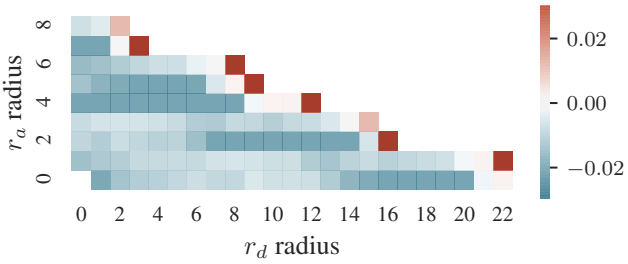


Figure 12. Comparing the binary-class and multi-class certificate for attribute perturbation on Cora-ML. Cells with blue (red) colors show the radii for which the multi-class (respectively binary-class) certificate obtains a higher certificated ratio. The darkest red cells in the corners exceed the color map and have value of around 0.15.

K. Further Experiments

First, we investigate the clean accuracy for different configurations of smoothing probabilities. In general, we would like to select the flip probabilities to be as high as possible such that the accuracy of the smoothed classifier is close to (or better than) the accuracy of the base classifier. To compute the clean accuracy we randomly draw 10^4 samples with $\phi(\cdot)$, record the class label for each test node, and make a prediction based on the majority vote. On Fig. 11 we show the clean accuracy averaged across 10 different random train/validation/test splits when we perturb the Cora-ML graph and using GCN as the base classifier.

Interestingly, when perturbing the attributes increasing p_- and p_+ improves over the accuracy of the base classifier (bottom-left corner, $p_- = 0, p_+ = 0$). We can interpret the perturbation as dropout (except applied during both training and evaluation) which has been previously shown to improve performance (Gasteiger et al., 2019; Velickovic et al., 2018). On the other hand, similar to the conclusions in our previous experiments, we see that the graph structure is more sensitive to perturbations compared to the attributes and the accuracy decreases as we increase the flip probabilities.

Second, we repeat the experiment associated with Fig. 2(a) where we calculate the certified ratio of test nodes for attribute perturbations on Cora-ML. We compare the binary-class certificate $\rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}}$ and the multi-class certificate $\mu_{\boldsymbol{x},\tilde{\boldsymbol{x}}}$. Fig. 12 shows that the multi-class certificate is better, i.e. achieves a higher certified ratio for the majority of (smaller) radii, while the binary-class certificate performs better for higher radii. In general, the absolute difference is relatively small, with the multi-class certificate being better by 0.012 on average across different radii.

L. Limitations

The main advantage of the randomized smoothing technique is that we can utilize it without making any assumptions about the base classifier f since to compute the certificate we need to consider only the output of f for each sample. This is also one of its biggest disadvantages since it does not take into account any properties of f, e.g. smoothness. More importantly, when applied for certifying graph data we can additionally leverage the fact that the predictions for neighboring nodes are often highly correlated, especially when the graph exhibits homophily. Extending our certificate to account for these aspects is a viable future direction.

Moreover, to accurately estimate $\underline{p_y(\boldsymbol{x})}$ we need a large number of samples (e.g. we used 10^6 samples in our experiments). Even though one can easily parallelize the sampling procedure developing a more sample-efficient variant is desirable. Finally, the guarantees provided are probabilistic, the certificate holds with probability $1 - \alpha$, and as shown in previous work (Cohen et al., 2019; Lee et al., 2019) the number of samples necessary to certify at a given radius grows as we increase our confidence, i.e. decrease α.

M. Certificate for Discrete Data

As before, since the randomization scheme which we defined in § 5 is applied independently per dimension w.l.o.g. we can focus only on those dimensions $\mathcal{C}$ where $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ disagree. We omit all proofs for the discrete case since they are analogous to the binary case. The only difference is in how we partition the space $\mathcal{X}_K$ and how we compute the respective regions. Once we obtain the regions the computation of $\rho_{\boldsymbol{x},\tilde{\boldsymbol{x}}}$ or $\mu_{\boldsymbol{x},\tilde{\boldsymbol{x}}}$ and hence the certificate is the same.

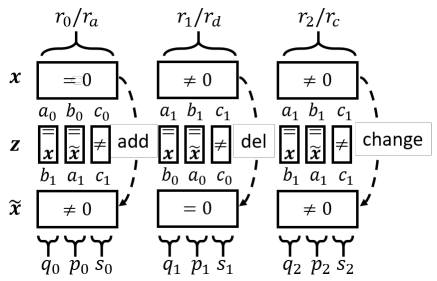


Figure 13. Illustration of the regions for the general sparsity-aware discrete certificate. We only show the dimensions $\mathcal{C}$ where $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ disagree. The triplets (q_j, p_j, s_j) are used to parametrize the regions. The variables a_0, b_0, c_0, and a_1, b_1, c_1 depend on the flip probabilities p_+, p_- and the number of categories K (see text).

Intuitively, we have variables q_0, q_1, q_2 corresponding to the dimensions where $\boldsymbol{z}_\mathcal{C}$ matches $\boldsymbol{x}_\mathcal{C}$, variables p_0, p_1, p_2 corresponding to the dimensions where $\boldsymbol{z}_\mathcal{C}$ matches $\tilde{\boldsymbol{x}}_\mathcal{C}$, and variables s_0, s_1, s_2 corresponding to the dimensions where $\boldsymbol{z}_\mathcal{C}$ matches neither $\boldsymbol{x}_\mathcal{C}$ nor $\tilde{\boldsymbol{x}}_\mathcal{C}$ (see illustration in Fig. 13). The fourth-case where $\boldsymbol{z}_\mathcal{C}$ matches both $\boldsymbol{x}_\mathcal{C}$ and $\tilde{\boldsymbol{x}}_\mathcal{C}$ is not possible since by definition $\boldsymbol{x}_i \neq \tilde{\boldsymbol{x}}_i$ for all $i \in \mathcal{C}$. We define the region parametrized by (q_j, p_j, s_j) triplets:

$$
\begin{aligned}
&\mathcal{R}^{q_0,q_1,q_2}_{p_0,p_1,p_2,s_0,s_1,s_2} = \{\boldsymbol{z} \in \mathcal{X}_K : \\
&q_0 = \sum_{i\in\mathcal{C}} \mathbb{I}(\boldsymbol{z}_i = \boldsymbol{x}_i)\mathbb{I}(\boldsymbol{x}_i = 0), \\
&q_1 = \sum_{i\in\mathcal{C}} \mathbb{I}(\boldsymbol{z}_i = \boldsymbol{x}_i)\mathbb{I}(\tilde{\boldsymbol{x}}_i = 0), \\
&q_2 = \sum_{i\in\mathcal{C}} \mathbb{I}(\boldsymbol{z}_i = \boldsymbol{x}_i)\mathbb{I}(\tilde{\boldsymbol{x}}_i \neq 0)\mathbb{I}(\boldsymbol{x}_i \neq 0), \\
&p_0 = \sum_{i\in\mathcal{C}} \mathbb{I}(\boldsymbol{z}_i = \tilde{\boldsymbol{x}}_i)\mathbb{I}(\boldsymbol{x}_i = 0), \\
&p_1 = \sum_{i\in\mathcal{C}} \mathbb{I}(\boldsymbol{z}_i = \tilde{\boldsymbol{x}}_i)\mathbb{I}(\tilde{\boldsymbol{x}}_i = 0), \\
&p_2 = \sum_{i\in\mathcal{C}} \mathbb{I}(\boldsymbol{z}_i = \tilde{\boldsymbol{x}}_i)\mathbb{I}(\boldsymbol{x}_i \neq 0)\mathbb{I}(\tilde{\boldsymbol{x}}_i \neq 0), \\
&s_0 = \sum_{i\in\mathcal{C}} \mathbb{I}(\boldsymbol{z}_i \neq \tilde{\boldsymbol{x}}_i)\mathbb{I}(\boldsymbol{z}_i \neq \boldsymbol{x}_i)\mathbb{I}(\boldsymbol{x}_i = 0), \\
&s_1 = \sum_{i\in\mathcal{C}} \mathbb{I}(\boldsymbol{z}_i \neq \tilde{\boldsymbol{x}}_i)\mathbb{I}(\boldsymbol{z}_i \neq \boldsymbol{x}_i)\mathbb{I}(\tilde{\boldsymbol{x}}_i = 0), \\
&s_2 = \sum_{i\in\mathcal{C}} \mathbb{I}(\boldsymbol{z}_i \neq \tilde{\boldsymbol{x}}_i)\mathbb{I}(\boldsymbol{z}_i \neq \boldsymbol{x}_i)\mathbb{I}(\boldsymbol{x}_i \neq 0)\mathbb{I}(\tilde{\boldsymbol{x}}_i \neq 0)\}
\end{aligned}
$$

for a given clean $\boldsymbol{x} \in \mathcal{X}_K$ and adversarial $\tilde{\boldsymbol{x}} \in \mathcal{S}_{r_0,r_1,r_2}(\boldsymbol{x})$ which is defined subsequently.

We use $a_0 = 1 - p_+$ as a shorthand for the probability to keep (not flip) a zero, $b_0 = \frac{p_+}{K-1}$ for the probability to flip a zero to some other value, and $c_0 = 1 - a_0 - b_0$. Similarly we define $a_1 = 1 - p_-$, $b_1 = \frac{p_-}{K-1}$, and $c_1 = 1 - a_1 - b_1$ for the non-zero values. We can easily verify from the definitions that given a specific configuration of q_j, p_j, s_j variables the ratio for the corresponding $\mathcal{R}^{q_0,q_1,q_2}_{p_0,p_1,p_2,s_0,s_1,s_2}$ region equals:

$$
\begin{aligned}
\eta &= \Pr(\phi(\boldsymbol{x}) \in \mathcal{R}^{q_0,q_1,q_2}_{p_0,p_1,p_2,s_0,s_1,s_2}) / \Pr(\phi(\tilde{\boldsymbol{x}}) \in \mathcal{R}^{q_0,q_1,q_2}_{p_0,p_1,p_2,s_0,s_1,s_2}) \\
&= \left(\frac{a_0}{b_1}\right)^{q_0-p_1} \left(\frac{b_0}{a_1}\right)^{p_0-q_1} \left(\frac{c_0}{c_1}\right)^{s_0-s_1} \left(\frac{a_1}{b_1}\right)^{q_2-p_2} \qquad (11)
\end{aligned}
$$

Furthermore, we define $r_j = q_j + p_j + s_j$ for $j = 0, 1, 2$. Now, we can compute the probability for $\phi(\boldsymbol{x})$ to land in the respective region as a product of Multinomials:

$$
\Pr(\phi(\boldsymbol{x}) \in \mathcal{R}^{q_0,q_1,q_2}_{p_0,p_1,p_2,s_0,s_1,s_2}) = \prod_{j=0}^{2} \Pr(\boldsymbol{u}_j = [q_j, p_j, s_j]) \qquad (12)
$$

where $\boldsymbol{u}_j$ are the following Multinomial random variables:

$$
\begin{aligned}
\boldsymbol{u}_0 &\sim \text{Mul}([a_0, b_0, c_0], r_0) \\
\boldsymbol{u}_1 &\sim \text{Mul}([a_1, b_1, c_1], r_1) \\
\boldsymbol{u}_2 &\sim \text{Mul}([a_1, b_1, c_1], r_2)
\end{aligned}
$$

These variables have only 3 categories regardless of the number of discrete categories in the input space. This is due to the fact that we only need to keep track of 3 states: $\boldsymbol{z}_i = \boldsymbol{x}_i$, $\boldsymbol{z}_i = \tilde{\boldsymbol{x}}_i$, and $\boldsymbol{x}_i \neq \boldsymbol{z}_i \neq \tilde{\boldsymbol{x}}_i$ for all $i \in \mathcal{C}$.

This construction suggests that we should parametrize our threat model with three radii: r_0/r_a which counts the number of added non-zeros, r_1/r_d which counts the number of removed non-zeros, and r_2/r_c which counts how many non-zeros changed to another non-zero value. We have:

$$
\begin{aligned}
\mathcal{S}_{r_0,r_1,r_2}(\boldsymbol{x}) = \{\tilde{\boldsymbol{x}} \in \mathcal{X}_K : &\sum_{i=1}^{d} \mathbb{I}(\boldsymbol{x}_i = 0)\mathbb{I}(\boldsymbol{x}_i \neq \tilde{\boldsymbol{x}}_i) = r_0, \\
&\sum_{i=1}^{d} \mathbb{I}(\tilde{\boldsymbol{x}}_i = 0)\mathbb{I}(\boldsymbol{x}_i \neq \tilde{\boldsymbol{x}}_i) = r_1, \\
&\sum_{i=1}^{d} \mathbb{I}(\boldsymbol{x}_i \neq 0)\mathbb{I}(\tilde{\boldsymbol{x}}_i \neq 0)\mathbb{I}(\boldsymbol{x}_i \neq \tilde{\boldsymbol{x}}_i) = r_2\}
\end{aligned}
$$

Similarly, we define the respective ball $\mathcal{B}_{r_0,r_1,r_2}(\boldsymbol{x})$ by replacing equalities with inequalities.

We can directly verify that for the binary case ($K = 2$), r_2 necessarily has to be equal to 0. We recover the definition of our threat model for binary data. Moreover, all s_i's, as well as $c_0 = \frac{(K-2)\cdot p_+}{K-1}$ and $c_1 = \frac{(K-2)\cdot p_-}{K-1}$ also have to be zero.

In order to partition the entire space $\mathcal{X}_K$ we have to generate all unique (q_j, p_j, s_j) triplets where $q_j + p_j + s_j = r_j$. There are $T_j = (r_j + 1)(r_j + 2)/2$ unique (q_j, p_j, s_j) triplets for $j = 0, 1, 2$. Therefore, the total number of regions is upper bounded by $T_0 \cdot T_1 \cdot T_2$. Note that this is an upper bound

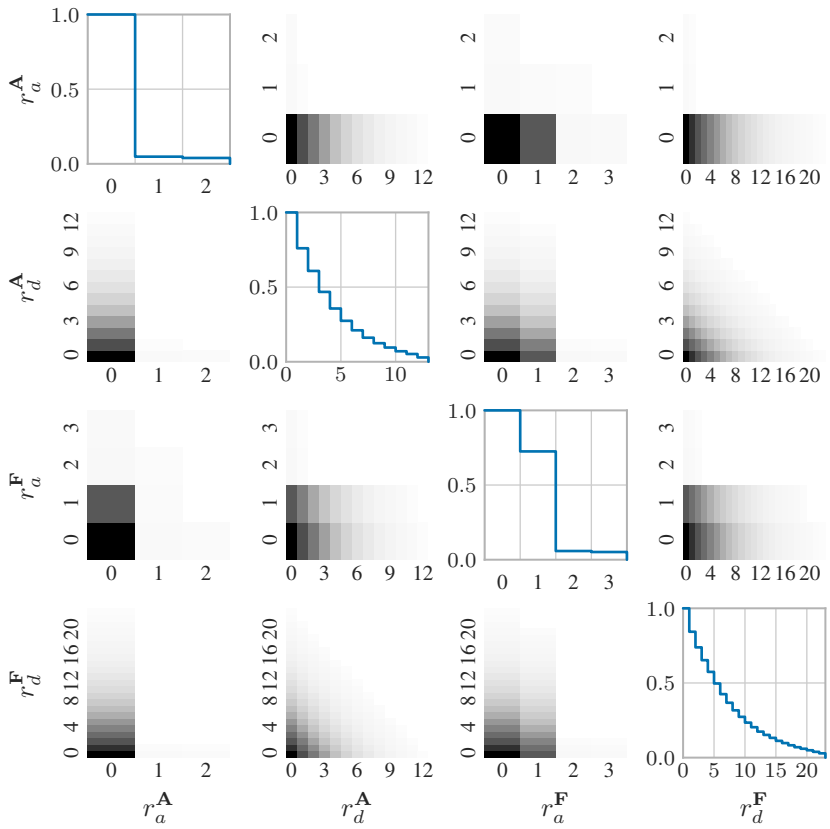


Figure 14. Joint certificate for both graph and attributes on Cora-ML. We show all pairwise heatmaps, e.g. $r_a^{\boldsymbol{A}} = r_d^{\boldsymbol{F}} = 0$ and varying $r_d^{\boldsymbol{A}}, r_a^{\boldsymbol{A}}$. The figure is symmetric w.r.t. the diagonal, which shows the certified ratio as we fix all radii except one to 0.

since the ratio in Eq. 11 is the same for certain combinations of q_j's, p_j's, and s_j's, e.q. when $q_0 - p_1 = 1 - 3 = 2 - 4$ and similarly for $p_0 - q_1$, $s_0 - s_1$, and $q_2 - p_2$. In these cases we can merge these regions into a single region.

The overall computation of the regions is efficient and it consists of: (i) generating all unique (q_j, p_j, s_j) triplets; (ii) computing the ratio defined in Eq. 11; and (iii) computing the probability for $\phi(\boldsymbol{x})$ to land in the respective region using Eq. 12. Since the number of regions is small the overall runtime is less than a second. We provide a reference implementation in Python with further details.

For the special case of $p_+ = p_-$ we have that $a_0 = a_1$, $b_0 = b_1$, and $c_0 = c_1$. Then the ratio in Eq. 11 simplifies to:

$$\eta = \left(\frac{a_0}{b_1}\right)^{q_0+q_1+q_2-p_0-p_1-p_2} = \left(\frac{a_0}{b_1}\right)^{q'-p'} \tag{13}$$

where we set $q' = q_0 + q_1 + q_2$ and $p' = p_0 + p_1 + p_2$. This directly implies that in this case we do not need to keep track of the different (q_j, p_j, s_j) triplets, but rather it is sufficient to parametrize the region with two variables, namely q' and p'. The probability that $\phi(\boldsymbol{x})$ lands in the respective $\mathcal{R}_{q',p'}$ region also simplifies (see Fig. 13):

$$\Pr(\phi(\boldsymbol{x}) \in \mathcal{R}_{q',p'}) = \Pr(\boldsymbol{u} = [q', p', r - q' - p']) \tag{14}$$

where $\boldsymbol{u} \sim \text{Mul}([a_0, b_0, c_0], r)$. Moreover, we have that $q' \in \{0, \ldots, r_0 + r_1 + r_2\} = \{0, \ldots, r\}$, where $\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_0 = r$. Similarly, $p' \in \{0, \ldots, r\}$. It follows that $(q' - p') \in \{-r, \ldots, r\}$, and thus there are only $2r + 1$ regions in total.

N. Further Analysis of Joint Certificates

On Fig. 14 we show our method's ability to certify robustness against combined perturbations on the graph and the attributes. The configuration of flip probabilities is the same as in § 8.1. Specifically to show different aspects of the 4D heatmap (certified ratio w.r.t. the 4 different radii) we plot all pairwise heatmaps, e.g. $r_a^{\boldsymbol{A}} = r_d^{\boldsymbol{F}} = 0$ and varying $r_d^{\boldsymbol{A}}, r_a^{\boldsymbol{A}}$. The figure is symmetric w.r.t. the diagonal, which shows the certified ratio as we fix all radii except one to 0. Similar to before we observe that we can certify more easily w.r.t. r_a compared to r_d. Since we are perturbing both features and structure at the same time we can obtain only modest certified radii. We leave it for future work to design models that are robust to such joint perturbations.