Structure-aware robustness certificates for graph classification

Pierre Osselin*, Henry Kenlay*, and Xiaowen Dong

University of Oxford
{osselinp,kenlay,xdong}@robots.ox.ac.uk

Abstract. Certifying the robustness of a graph-based machine learning model poses a critical challenge for safety. Current robustness certificates for graph classifiers guarantee output invariance with respect to the total number of node pair flips (edge addition or edge deletion), which amounts to an l_0 ball centred on the adjacency matrix. Although theoretically attractive, this type of isotropic structural noise can be too restrictive in practical scenarios where some entries of the adjacency matrix are more critical than others in determining the classifier's output. The certificate, in this case, gives a pessimistic depiction of the robustness of the graph model. To tackle this issue, we develop a randomised smoothing method based on adding an anisotropic noise distribution to the input graph structure. We show that our process generates structurally-aware certificates for our classifiers, whereby the magnitude of robustness certificates can vary across different pre-defined structures of the graph. We demonstrate the benefits of these certificates on both synthetic and real-world experiments.

Keywords: Graph · Robustness · Certificates.

1 Introduction

Graph-based machine learning models have made considerable strides in the last couple of years, with applications ranging from NLP [19], combinatorial optimization [4] and protein function prediction [8]. As these tools become more common, studying their vulnerability to potential adversarial examples turns paramount for safety purposes.

Robustness certification is an active field of research whose goal is to develop certificates guaranteeing invariance of the model prediction with respect to some input perturbations. Diverse methods have been used to achieve this goal, from interval bound propagation [9], convex relaxation [14], Lipschitz bounds computation [10] or randomised smoothing [16]. Given a data point $\mathbf{x}$ and a set of perturbed inputs $\mathcal{B}(\mathbf{x})$, a robustness certificate verifies that a model's prediction $f(\mathbf{x})$ remains unchanged for all other inputs in the perturbation set. That is, for all $\mathbf{x}' \in \mathcal{B}(\mathbf{x})$ it holds that $f(\mathbf{x}) = f(\mathbf{x}')$. Often the set of perturbed inputs $\mathcal{B}(\mathbf{x})$ is parameterised, for example by a closed-ball $\mathcal{B}_r(\mathbf{x}) = \{\mathbf{x}' : d(\mathbf{x}, \mathbf{x}') \leq r\}$ under some distance function d and radius r. In this case we are interested in knowing the largest r that we can certify for, which r is called the certified radius.

In the context of robustness certification of graph classifiers against structural perturbation, a common choice of perturbation set is the set of all graphs reachable from an input graph $\mathbf{x}$ by up to r node pair flips (edge additions and deletions)[1]. This corresponds to a closed ball on the upper triangle entries of the adjacency matrix where the distance is induced by the ℓ_1 norm and the bottom triangle entries are determined by the constraint that the adjacency matrix is symmetric[2]. In some cases, however, different node pairs of the graph can be more predictive of the ground truth label than others. In such situations, certifying according to a total number of edge additions or deletions might gives a pessimistic certified radius, because the set of perturbed inputs may include perturbations which consist of flipping many critical node pairs (in terms of determining the graph label).

In this work we solve this problem by defining disjoint regions of node pairs and proposing robustness certificates that verify that the prediction of the classifier will not change for a potentially different number of node pair flips for each region. Our approach relies on randomised smoothing, which is a powerful framework to produce robustness certificates which hold with high probability. Given some noise distribution over the input, randomised smoothing transforms a base model f into a smoothed model g for which we can provide probabilistic robustness guarantees.

Existing randomised smoothing approaches for certifying graph classification mostly consider an isotropic noise distribution that flips each node pair with a fixed probability [11, 6, 16]. The certificate in this case corresponds to the total number of node pair flips. Instead, we propose using an anisotropic noise distribution based on the predefined regions whereby the probability of flipping a node pair depends on to which region (if any) the node pair belongs. We show that smoothed classifiers constructed using this anisotropic noise distribution naturally lead to structure-aware robustness certificates whereby different number of node flips are certified for each of the regions. We demonstrate the benefits of our approach on both synthetic and real-world experiments. To the best of our knowledge, our method is one of the first of its kind that allows for flexible graph certification in the input domain.

2 Preliminaries

Let $\mathcal{X}$ be the data space and $f : \mathcal{X} \to \mathcal{Y}$ be a classifier which maps each point $\mathbf{x} \in \mathcal{X}$ to a label $y \in \mathcal{Y} = \{0, ..., C-1\}$. Let $\phi : \mathcal{X} \to \mathcal{P}(\mathcal{X})$ be a noise distribution over our data, that is, $\phi(\mathbf{x})$ returns a distribution over $\mathcal{X}$ for every point $\mathbf{x} \in \mathcal{X}$. We write $f(\phi(\mathbf{x}))$ to denote the random variable $\mathbb{P}_{\mathbf{z}\sim\phi(\mathbf{x})}(f(\mathbf{z}))$. This represents the distribution of outputs of the base classifier given the randomisation scheme

[1] We use the term node pair flip instead of edge flip to emphasise that we are considering the addition of edges that do not exist in the original graph as well as the deletion of existing edges.

[2] We assume the graphs are unweighted and undirected.

applied to the input. We define g to be the smoothed classifier of our base classifier f as

$$g(\mathbf{x}) = \operatorname*{argmax}_{y \in \mathcal{Y}} \mathbb{P}(f(\phi(\mathbf{x})) = y). \tag{1}$$

The smoothed classifier can be interpreted as a neighbourhood vote, where the output is the mode of the output of the classifier when inputs are sampled from the distribution $\phi(\mathbf{x})$.

2.1 Certifying a smoothed classifier

We can construct a lower and upper bound on $\mathbb{P}(f(\phi(\tilde{\mathbf{x}})) = y)$ for neighbouring points $\tilde{\mathbf{x}} \in \mathcal{X}$ which will be the basis for certifying around a point $\mathbf{x}$. We define the notion of point-wise certificates from the framework of [12] where:

$$\underline{\rho_{\mathbf{x},\tilde{\mathbf{x}}}}(p, y) = \min_{\substack{h \in \mathcal{H}: \\ \mathbb{P}(h(\phi(\mathbf{x}))=y)=p}} \mathbb{P}(h(\phi(\tilde{\mathbf{x}})) = y) \tag{2}$$

$$\overline{\rho_{\mathbf{x},\tilde{\mathbf{x}}}}(p, y) = \max_{\substack{h \in \mathcal{H}: \\ \mathbb{P}(h(\phi(\mathbf{x}))=y)=p}} \mathbb{P}(h(\phi(\tilde{\mathbf{x}})) = y) \tag{3}$$

In this definition, $\mathcal{F}$ is the class the set of measurable classifiers with respect to ϕ. Because the optimisation constraints include the base classifier $f \in \mathcal{H}$ it follows that

$$\underline{\rho_{\mathbf{x},\tilde{\mathbf{x}}}}(p, y) \leq \mathbb{P}(f(\phi(\tilde{\mathbf{x}})) = y) \leq \overline{\rho_{\mathbf{x},\tilde{\mathbf{x}}}}(p, y). \tag{4}$$

We define a perturbation set $\mathcal{B}_r(\mathbf{x})$ as a family of sets $\mathcal{B}_r(\mathbf{x}) \subseteq \mathcal{X}$ parameterised by some $r \geq 0$ such that $\mathcal{B}_r(\mathbf{x}) \subseteq \mathcal{B}_{r'}(\mathbf{x})$ if and only if $r \leq r'$. This definition includes open or closed balls with respect to a metric over $\mathcal{X}$. We say that the smoothed classifier g is certified at x in some perturbation set $\mathcal{B}_r(\mathbf{x})$ if the output of g is the same for all neighbouring points $\tilde{\mathbf{x}} \in \mathcal{B}_r(\mathbf{x})$ in the perturbation set.

Following Cohen, we will write c_A to be the output class of $g(\mathbf{x})$ which is returned with probability p_A, and c_B to be the "runner-up" class, i.e., the class distinct from c_A with the next highest probability p_B. We can make use of Equation 2 and Equation 3 to certify around a point by verifying if the following holds

$$\min_{\tilde{\mathbf{x}} \in \mathcal{B}_r(\mathbf{x})} \Phi_{\mathbf{x},\tilde{\mathbf{x}}}(c_A, p_A) > 0, \tag{5}$$

where

$$\Phi_{\mathbf{x},\tilde{\mathbf{x}}}(c_A, p_A) \triangleq \underline{\rho_{\mathbf{x},\tilde{\mathbf{x}}}}(p_A, c_A) - \overline{\rho_{\mathbf{x},\tilde{\mathbf{x}}}}(p_B, c_B). \tag{6}$$

Equation 6 can be thought of as a margin, which gives the difference between a lower bound on p_A and an upper bound on p_B for some point $\tilde{\mathbf{x}}$ in the perturbation set. Equation 5 then indidcates if this property holds for all $\tilde{\mathbf{x}}$ in the perturbation set.

2.2 Certified radius

Given a perturbation set, we can define the certified radius to be the largest value of r so that we can certify with respect to the set $\mathcal{B}_r(\mathbf{x})$. Formally this is defined to be

$$R(\mathbf{x}) = \sup r, \text{ s.t. } \min_{\tilde{\mathbf{x}} \in \mathcal{B}_r(\mathbf{x})} \Phi_{\mathbf{x},\tilde{\mathbf{x}}}(p^*, y^*) > 0. \tag{7}$$

In the case that the perturbation set is an open or closed ball, then $R(\mathbf{x})$ is the radius of the largest ball we can certify over.

2.3 Computing the certificate

We partition the space $\mathcal{X} = \bigcup_i \mathcal{H}_i$ into disjoint regions of equal likelihood ratios $\mathcal{R}_k = \{\mathbf{z} \in \mathcal{X} : \frac{\mathbb{P}(\phi(\tilde{\mathbf{x}})=\mathbf{z})}{\mathbb{P}(\phi(\mathbf{x})=\mathbf{z})} = c_k\}$ where without loss of generality we can assume $c_k \in \mathbb{R}$ are in an ascending order. The quantity $\Phi_{\mathbf{x},\tilde{\mathbf{x}}}(p_A, c_A)$ can be computed by solving the following linear programming (LP) problems [12]:

$$\underline{\rho_{\mathbf{x},\tilde{\mathbf{x}}}}(p_A, c_A) = \min_{\mathbf{h}} \mathbf{h}^T \tilde{\mathbf{r}} \quad \text{s.t.} \quad \mathbf{h}^T \mathbf{r} = p_A, \quad 0 \leq \mathbf{h} \leq 1 \tag{8}$$

and similarly,

$$\overline{\rho_{\mathbf{x},\tilde{\mathbf{x}}}}(p_B, c_B) = \min_{\mathbf{h}} \mathbf{h}^T \tilde{\mathbf{r}} \quad \text{s.t.} \quad \mathbf{h}^T \mathbf{r} = p_B, \quad 0 \leq \mathbf{h} \leq 1. \tag{9}$$

The variables $\mathbf{h}$ and $\mathbf{t}$ correspond to optimising over the classifiers that are optimised over in Eq. 2 and Eq. 3. The vectors $\mathbf{r}$ and $\tilde{\mathbf{r}}$ are $\mathbf{r}_i = \mathbb{P}(\phi(\mathbf{x}) \in \mathcal{H}_i)$ and $\tilde{\mathbf{r}}_i = \mathbb{P}(\phi(\tilde{\mathbf{x}}) \in \mathcal{H}_i)$ respectively. This LP problem can be solved via a greedy approach [12]. Given the ratios c_k are ordered in an ascending manner, starting with $\mathbf{h} = \mathbf{0}$ we can assign $\mathbf{h}_i$ for regions $\mathcal{H}_1, ...\mathcal{H}_k$ as long as $\mathbf{h}^T\mathbf{r} \leq p_A$, and partially fill the last $\mathbf{h}_k$ such that $\mathbf{h}^T\mathbf{r} = p_A$. We can solve Equation 9 in a similar way.

Given this efficient way to compute a certificate there remains some quantities that must be calculated. The first is a partition of the space $\mathcal{X}$ into disjoint unions of equal likelihood ratios. Next, the values c_k must be computed, or at least given in closed form, so the regions can be sorted in ascending order. Finally, a closed form for $\mathbb{P}(\phi(\mathbf{x}) = \mathbf{z})$ allows us to compute $\mathbf{r}$ and we can compute $\tilde{\mathbf{r}}$ by noticing that $\mathbb{P}(\phi(\tilde{\mathbf{x}}) = \mathbf{z}) = c_k \mathbb{P}(\phi(\mathbf{x}) = \mathbf{z})$. For our certificate, we provide disjoint unions in Proposition 1, give a closed form for c_k in Proposition 2 and finally compute $\mathbb{P}(\phi(\mathbf{x}) = \mathbf{z})$ in Proposition 3.

2.4 Randomised smoothing for graph classification

In this work we are interested in the robustness of graph classification models to structural perturbation of undirected, unweighted graphs[3]. In this situation

[3] This work can be extended to the setting of directed graphs as well as the task of node classification.

the input domain is the set of finite graphs $\mathcal{X} = \cup_{i=1}^{\infty} \mathcal{X}_n$ where $\mathcal{X}_n$ is the space of graphs on n nodes. We can represent graphs on n nodes as a binary vector of size $\binom{n}{2}$ where each entry indicates the presence or absence of an edge. Without loss of generality we will treat graphs as binary vectors $\mathbf{x}$ from here on in[4].

We are interested in robustness with respect to node pair flips which can represent an edge addition (change a zero to a one in $\mathbf{x}$) or edge deletion (change a one to a zero). This may be interpreted as adding "structural noise" to the input graph. Two candidates for the distribution of such noise have been proposed in the literature. The first one applies an independent Bernoulli distribution to every node pair. That is, for all $\phi(\mathbf{x})_i = \mathbf{x}_i \oplus \epsilon_i$ with $\epsilon_i \sim Bern(p)$ for a fixed p, where $\oplus$ is the bitwise XOR operator. We refer to this noise distribution as isotropic, as it flips each node pair with equal probability. This approach is used by [11, 16, 6]. The second approach, a sparsity aware noise distribution [1], gives a different probability of edge flipping depending on whether the edge is present in the initial graph. If an edge exists between a node pair then it is flipped with probability p_-, whereas if a node pair does not exist between two nodes it is added with probability p_+. This distribution can be written as $P(\phi(\mathbf{x})_i \neq \mathbf{x}_i) = p_-^{\mathbf{x}_i} p_+^{1-\mathbf{x}_i}$.

The sparsity-aware noise distribution of [1] distinguishes probabilities of edge deletion and addition, which encapsulates the sparsity property of the graph. The design of such noise comes from the rationale that real world graphs present a sparsity structure which will break with a single Bernoulli distribution where the number of edge additions will generally surpass the number of edge deletions leading to an unrealistic graph. This can be considered as a simplest instance of structure-aware perturbation (i.e., distinguishing between edge deletion and addition). Inspired by this work, we propose structure-aware certificates where node pairs can be partioned into one of many node pair sets and perturbed according to their membership of these sets.

3 Randomised smoothing with anisotropic noise

Given $\mathbf{x} \in \mathcal{X}_n$, suppose we divide our input space of node pairs up into disjoint regions $\bigsqcup_{i \in I} \mathcal{C}_i$ such that each node pair belongs to exactly one region and there are a total of C regions. We define a noise distribution where independent Bernoulli distributions are applied to every node pair, and where the parameter of the Bernoulli distribution is shared within every set $\mathcal{C}_i$:

$$\phi(\mathbf{x})_k = \mathbf{x}_k \oplus \epsilon_k, \text{ where } \epsilon_k \sim \text{Bern}(p_i) \text{ and } k \in \mathcal{C}_i, \tag{10}$$

We will certify using the following procedure. Let $\mathbf{R} \in \mathbb{Z}^C$ be a tuple of integers such that $0 \leq \mathbf{R}_i \leq |\mathcal{C}_i|$ and let $\mathcal{B}_{\mathbf{R}}(x) = \{\mathbf{z} \in \mathcal{X}_N : \|\mathbf{z}_{\mathcal{C}_i} - \mathbf{x}_{\mathcal{C}_i}\|_0 \leq \mathbf{R}_i\}$ be a perturbation set. Let $\tilde{\mathbf{x}} \in \mathcal{B}_{\mathbf{R}}(x)$ and $\mathcal{J} = \{i : \mathbf{x}_i \neq \tilde{\mathbf{x}}_i\}$ be indices of $\mathbf{x}$ which are perturbed to give $\tilde{\mathbf{x}}$. Furthermore, let $\mathcal{J}_i = \mathcal{J} \cap \mathcal{C}_i$ be indices where $\mathbf{x}$ is perturbed in collection $\mathcal{C}_i$.

[4] Note that due to isomorphism multiple different binary vectors can represent the same graph.

Proposition 1. *We define regions* $\mathcal{R}_{\mathbf{Q}} = \{\mathbf{z} \in \mathcal{X}_N : \|\mathbf{z}_{\mathcal{J}_i} - \mathbf{x}_{\mathcal{J}_i}\|_0 = Q_i : i \in I\}$ *that represent points* $\mathbf{z}$ *which agree with* $\mathbf{x}$ *by exactly* Q_i *bits in sub-regions* $\mathcal{J}_i$*. Then* $\mathcal{X}_N$ *can be represented by the following disjoint union*

$$\bigcup_{\mathbf{0} \leq \mathbf{Q} \leq \mathbf{R}} \mathcal{R}_{\mathbf{Q}}, \tag{11}$$

where vector inequalities are element-wise.

Furthermore, the likelihood ratio is fixed for elements $\mathbf{z}$ in any one region. This likelihood ratio has the following closed form.

Proposition 2. *Consider a region* $\mathcal{R}_{\mathbf{Q}} = \{\mathbf{z} \in \mathcal{X}_N : \|\mathbf{z}_{\mathcal{J}_i} - \mathbf{x}_{\mathcal{J}_i}\|_0 = Q_i\}$ *then for all* $\mathbf{z} \in \mathcal{R}_{\mathbf{Q}}$ *the following holds*

$$\eta^{\mathcal{R}}_{\mathbf{Q}} = \frac{\mathbb{P}(\phi(\tilde{\mathbf{x}}) = \mathbf{z})}{\mathbb{P}(\phi(\mathbf{x}) = \mathbf{z})} = \prod_{i=1}^{C} \left(\frac{1 - p_i}{p_i} \right)^{R_i - 2Q_i}. \tag{12}$$

Finally, we can compute the likelihood of a smoothed input belonging to these regions:

Proposition 3. *The probability of the output of a smoothed input* $\phi(\mathbf{x})$ *belonging to a region* $\mathcal{R}_{\mathbf{Q}}$ *is given by*

$$\mathbb{P}(\phi(\mathbf{x}) \in \mathcal{R}_{\mathbf{Q}}) = \prod_{i=1}^{C} \mathrm{Bin}(R_i - Q_i | R_i, p_i), \tag{13}$$

where $\mathrm{Bin}(R_i - Q_i | R_i, p_i)$ *is the probability mass function of the binomial distribution giving probability of* $R_i - Q_i$ *successes from* R_i *trials each with success probability* p_i*.*

Using these results we can provide robustness certificates of the smoothed classifier. Given $x \in \mathcal{X}$ and our noise distribution, we can compute the values $p_y(\mathbf{x})$. In practice, these quantities are not available in closed form and are estimated via sampling, as in [1]. A more detailed description is given in Appendix B, which gives probabilistic certificates according to some confidence intervals. Without loss of generality we order the regions $\mathcal{R}_1, \ldots \mathcal{R}_T$ (where $T = \prod_i (R_i + 1)$). The corresponding ratios $\eta^{\mathcal{R}}_{\mathbf{Q}}$ as given by Proposition 2 are ordered $c_1 \leq \ldots \leq c_T$). From these elements, $\Phi_{\mathbf{x},\tilde{\mathbf{x}}}(p^*, y^*)$ can be computed through eq. 6 and the optimisation problem eq. 7 can be solved. In practice, the optimisation of eq. 7 can be solved efficiently by leveraging some symmetries displayed by $\Phi_{\mathbf{x},\tilde{\mathbf{x}}}(p^*, y^*)$ when $\tilde{\mathbf{x}}$ varies. This property is more thoroughly described in the appendix B.

4 Experiments

4.1 Synthetic experiment

We motivate the use of anisotropic noise by considering inputs $\mathbf{x}$ that are an element of some space $\mathcal{X} = \mathcal{X}_1 \oplus \mathcal{X}_2$ where $\oplus$ is the direct sum. Consider a point

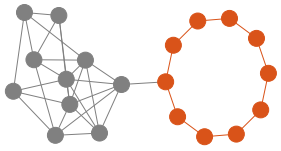
that is close to the decision boundary in the $\mathcal{X}_1$ subspace but far from the decision boundary in the $\mathcal{X}_2$ subspace. An isotropic certificate can not certify beyond the small distance to the decision boundary in $\mathcal{X}_1$. However, our certificate can certify the distances of $\mathcal{X}_1$ and $\mathcal{X}_2$ jointly allowing us to certify a small distance in the $\mathcal{X}_1$ subspace but a large distance in the $\mathcal{X}_2$ subspace.

We design a synthetic graph classification data set whereby the graphs are constructed using a motif which determines the class label (corresponding to the important subspace $\mathcal{X}_1$) connected to a randomly generated graph (corresponding to the unimportant subspace $\mathcal{X}_2$) which is independent of the class label. We can consider edges in the motif part to be in one node pair set $\mathcal{C}_{\text{motif}}$ and edges from the random part in $\mathcal{C}_{\text{random}}$. Given a model that solves this task, we would expect changes in the motif to move the input closer or further away from a decision boundary but changes in the random part of the graph to move the point parallel to the direction of the decision boundary. We should be able to certify a large number of node pairs in $\mathcal{C}_{\text{random}}$ by applying a large value of noise p_{random} without hurting the accuracy of the smoothed classifier. We can not certify a large number of node pairs in $\mathcal{C}_{\text{motif}}$ without drops in accuracy, so we choose a small value of noise p_{motif} to retain high accuracy. The graphs we generate and the noise we use to perturb the graphs are shown in Fig. 1.

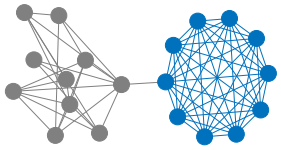
We generate balanced train, validation and test sets of size 1000, 1000, and 100 respectively. The test set is smaller as the randomised smoothing procedure is computationally expensive. This is because to estimate p_A a large number of random inputs need to be generated and inferred using the model. We generate a binary classification problem where each graph has a motif part of $n_{\text{motif}} = 10$ nodes where a cycle determines a negative label and a complete graph determines a positive label. We then generate a random part using a connected Erdős-Rényi graph [7] with $n_{\text{random}} = 10$ nodes and parameter $p = 0.5$. We join these graphs using a single edge. See Fig. 1a for an example of the negative class and Fig. 1b for an example of the positive class.

We train a SVM classifier using a node label histogram kernel [15] where the node label corresponds to the node degree. Let $c(\mathcal{G}, d)$ be a function that counts the number of nodes in a graph $\mathcal{G}$ with degree d. Then the kernel applied to graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ can be written as $\kappa(\mathcal{G}_1, \mathcal{G}_2) = \sum_{d=0}^{\infty} c(\mathcal{G}_1, d) \cdot c(\mathcal{G}_2, d)$ which is well defined for finite graphs. We use this model as we expect it to be sensitive to the motif structure that determines the label; the negative label gives a large value in the $c(\cdot, 2)$ dimension whereas the positive label gives a large value in the in the $c(\cdot, n_{\text{motif}} - 1)$ dimension. Indeed, the base classifier gets 100% accuracy on the train, validation and test data sets.

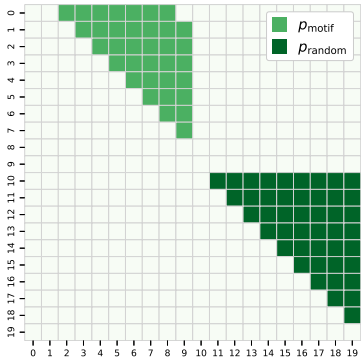
For the certification procedure, we apply noise separately for node pairs in the motif part and noise pairs in the random part. We apply noise to internal edges of the motif part only (i.e. not part of the outer cycle, see Fig. 1). We do not apply noise to node pairs where one node lies in the motif and one does not. We also do not perturb the edge that joins the motif part and the random part. The noise matrix is shown in Fig. 1c. We generate $100,000$ perturbations per test-sample and use these to estimate the output of the smoothed classifier and



(a) Graph with negative label.



(b) Graph with positive label.

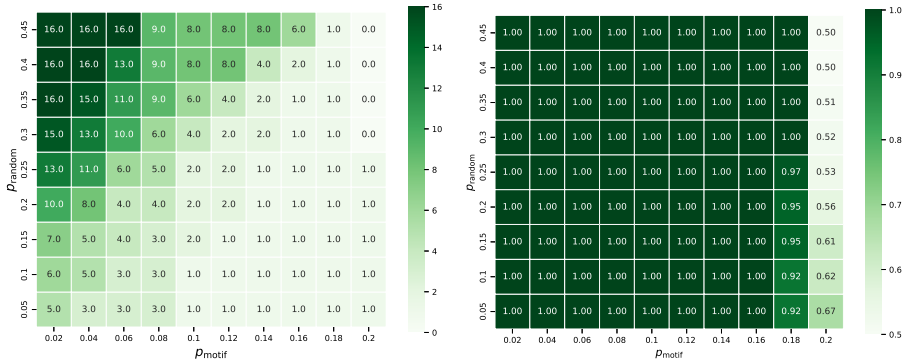


(c) Noise matrix.

Fig. 1: Example graphs with a positive and negative label. Blue nodes and edges denote a motif part of a positive label and red nodes and edges denote a motif part of a negative label. The noise matrix show how edges are perturbed with p_{motif} being the noise parameter for node pairs in the motif part and p_{random} being the noise applied to node pairs in the random part. Notice that only the internal edges of the motif are perturbed, and the bridge edge is not perturbed. The upper triangle of the noise is shown only, in practice this is sampled and applied to the upper and lower triangle of the adjacency so the graph remains undirected.

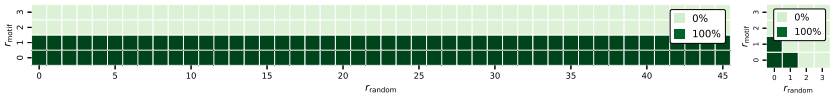
generate a certificate. We use a confidence level of $\alpha = 0.99$ to estimate p_A. We also compute certificates using isotropic noise for comparison.

For our certificate with anisotropic noise we consider $\mathbf{p} = (p_{\text{motif}}, p_{\text{random}}) \in \mathbf{P}$ where $\mathbf{P} = \{0.02, 0.04, \ldots, 0.2\} \times \{0.05, 0.1, \ldots, 0.45\}$. Recall that p_{motif} is the noise parameter for the motif part and p_{random} is the noise parameter for the random part. For the isotropic certificate we consider $p \in \{0.02, 0.04, \ldots, 0.2\}$. For the anisotropic certificate we certify over perturbation pairs $\mathbf{r} = (r_{\text{motif}}, r_{\text{random}})$ which means that with high probability r_{motif} edge flips in the motif part and r_{random} edge flips in the random part will not change the label of the smoothed classifier. The isotropic certificate guarantees the label does not change for r edge flips anywhere in the graph.



(a) Score. (b) Test-set accuracy.

Fig. 2: The test-set accuracy of the smoothed classifier and the certified volume for various values of $\mathbf{p} = (p_{\text{motif}}, p_{\text{random}})$.



(a) Anisotropic. (b) Isotropic.

Fig. 3: Comparison of an anisotropic certificates with $\mathbf{p} = (0.02, 0.45)$ and isotropic certificates for various levels of p. We omit some values of p for the isotropic certificate for readability.

We begin by analysing how the noise vector $\mathbf{p}$ influences the behaviour of the smoothed classifier and the certificate. We introduce a score to evaluate the noise parameters. For each $\mathbf{r}$ if we can certify strictly more than half of the test

samples in this perturbation space then we add 1 to the score. To motivate the utility of this score consider a smoothed model with large values of noise. In the limit, a perfectly smooth classifier will be constant everywhere. In other-words, it will predict the same label for all inputs and give a certified accuracy of 50% for a balanced classification task. This classifier could be certified for arbitrary numbers of edge flips. For this reason, metrics such as total certified area averaged over samples do not necessarily tell us if a noise parameter is useful.

In Fig. 2a we can see that our smoothed classifier has the highest score when p_{motif} is small and p_{random} is large. In Fig. 2b we see that large values of p_{random} does not effect the accuracy of the smoothed classifier, but if p_{motif} becomes too large the accuracy begins to drop. These results are expected–p_{motif} cannot be too large as the motif part is more important to determining the label. This motivates us to fix p_{motif} to be small and increase p_{random} allowing us to retain high test accuracy whilst increasing the number of edge flips we can certify in $\mathcal{C}_2$.

We take a closer look at the smoothed classifier given by $\mathbf{p} = (0.02, 0.45)$, one of the smoothed classifiers with the highest observed score. We are interested in a smooth model with high test set accuracy that can certify for many values of $\mathbf{r}$. Our model has 100% certified accuracy. The proportion of the test set that can be certified for varying values of $\mathbf{r}$ is shown in Fig. 3a. As the Figure demonstrates we can certify 100% of the test-set samples to 0 or 1 edge flips in the motif part of the graph and up to 45 edge flips in the random part of the graph. This is the maximum number of possible node pairs in the random part, so we can certify any perturbation in this part of the graph. The smoothed classifier using isotropic noise can also achieve 100% test set accuracy for all values of noise we tested. We show the certification results for when $p = 0.02$, as this is the only value that allows us to certify the entire test-set for one edge flip. We plot the proportion of the test set that can be certified at using this value of isotropic noise in Fig. 3b. Using larger values of noise for the isotropic certificate allows for some of the test-set to be certified at a radius of 2, but it can no longer certify the entire test set at radius 1. By using anisotropic noise, and being specific about where edges are being certified, we can certify 46 edge flips with 100% accuracy compared to 1 edge flip with 100% accuracy.

4.2 Real-world experiment

We also experiment using a real-world data set. For this, we consider the MUTAG data set [3]. In this data set each graph represents a molecule and the goal is to predict the molecules mutagenicity on a specific bacterium, which is encoded into a binary label. Each node has one of 7 discrete node labels corresponding to atomic number which is one-hot encoded. The data set contains a total 188 molecular graphs.

We train a graph neural network which has a single GCN layer [17] with 64 hidden units, followed by a max pooling layer and a linear layer. We train on 80% of the training set and use 10% of the data as a validation which is used to select

the best weights during training, as measured by accuracy on the validation set. We train for a maximum of 500 epochs using the AdamW optimiser [13] with weight decay of 10^{-3}. The initial learning rate is 10^{-3} and it is decayed by 0.5 every 50 epochs.

We compare our certificate to [1], which we refer to as a sparsity aware certificate, as this is the only non-isotropic certificate used for graph classification that we are aware of. We consider node pairs where there is an edge in the original graph as $\mathcal{C}_1$ and all other node pairs as $\mathcal{C}_2$. In this scenario, we can certify edge deletions and additions in a comparable way. Following the setup described in [1] we consider $p_1 = 0.4$ which corresponds to the probability of deleting and edge and $p_2 = 0.2$ which corresponds to the probability of adding an edge. We apply noise during training to make the model more robust.

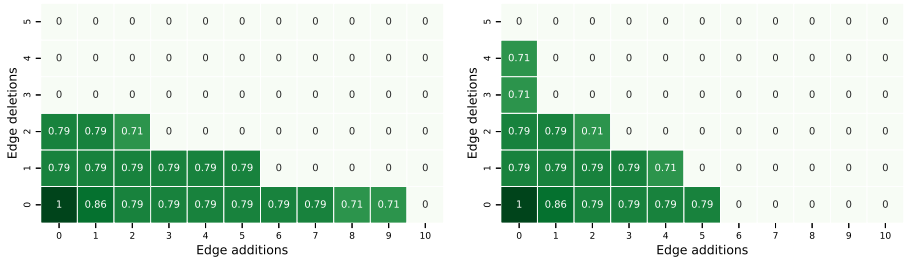
The values computed in Proposition 2 differ between the two approaches as $\mathbb{P}(\phi(\tilde{\mathbf{x}}) = \mathbf{z})$ is computed differently. Furthermore, the probability $\phi(\mathbf{x})$ belonging to a region in the anisotropic approach is a product of Binomial distributions (Proposition 3) whereas for the sparsity-aware certificate this probability follows a Poison-Binomial distribution. If the assignment of node pairs was dependent on the individual sample, this would generalise our approach further, and would also generalise the sparsity-aware certificate.

Our model has a test-set accuracy of 84%. In Fig. 4 we plot the ratio of correctly predicted test points that are certified for varying numbers of edge deletions and additions. We make a few observations from these results. The first is that for values of $\mathbf{r}$ where both methods can certify test points, our method certifies the same quantity of points and in some cases more. The second is that there are two values of $\mathbf{r}$ where the sparsity-aware certificate can certify test samples but the anisotropic certificate cannot. However, there are five values of $\mathbf{r}$ where the anisotropic can certify but the sparsity-aware certificate cannot. Finally, we note that in this experiment, as well as the synthetic experiments, we find our certificates tend to be oblong, i.e. if p_i is larger than we tend to certify for larger values in the r_i direction. This is advantageous in the case where some node pairs are considered more important to the classification label (as demonstrated in the synthetic experiment).

5 Conclusions

In this work we propose the first method that introduces structure-aware robustness certificates in the context of undirected, unweighted graph classification. To achieve this, we leveraged a flexible, anisotropic noise distribution in the context of randomised smoothing and developed an efficient algorithm to compute certificates. We apply these certificates to a synthetic experiment and demonstrate a clearly improved robustness of graph classifiers that cannot be achieved with an isotropic certificates. We also validate our certificate on real-world experiments and show superior results to an existing approach.

A requirement to using our approach is defining a priori which edges a user believes to be more or less important to determining the graph label. One can



(a) Anisotropic. (b) Sparsity-aware.

Fig. 4: A comparison between the anisotropic certificate and the sparsity-aware certificate. Each entry represents the ratio of correctly classified test-set samples that could be certified at a specified number of edge deletions and additions.

use domain expertise (which we simulate in the synthetic experiment), or treat edge deletions and additions differently as we do following the sparsity-aware approach. We may also consider approaches that have been used to identify edges that may be vulnerable to attack. For example, a previous work found edges vulnerable to adversarial attack are those not captured by a low-rank approximation of the adjacency [5]. Another line of work propose that edges where the end-point node features have low Jaccard index are potentially vulnerable [18]. Beyond this, one could learn the importance of the node pairs in a data-driven fashion. We leave these directions for future work. Finally, even though we have applied our method in the context of graph classification, it can also used for any type of task based on a discrete domain.

References

1. Bojchevski, A., Klicpera, J., Günnemann, S.: Efficient robustness certificates for discrete data: Sparsity-aware randomized smoothing for graphs, images and more. International Conference on Machine Learning pp. 1003–1013 (2020)
2. Cai, T.T.: One-sided confidence intervals in discrete distributions. Journal of Statistical planning and inference **131**(1), 63–88 (2005)
3. Debnath, A.K., Lopez de Compadre, R.L., Debnath, G., Shusterman, A.J., Hansch, C.: Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. Journal of medicinal chemistry **34**(2), 786–797 (1991)
4. Drori, I., Kharkar, A., Sickinger, W.R., Kates, B., Ma, Q., Ge, S., Dolev, E., Dietrich, B., Williamson, D.P., Udell, M.: Learning to solve combinatorial optimization problems on real-world graphs in linear time. 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA) pp. 19–24 (2020)
5. Entezari, N., Al-Sayouri, S.A., Darvishzadeh, A., Papalexakis, E.E.: All you need is low (rank) defending against adversarial attacks on graphs. In: Proceedings of the 13th International Conference on Web Search and Data Mining. pp. 169–177 (2020)

6. Gao, Z., Hu, R., Gong, Y.: Certified robustness of graph classification against topology attack with randomized smoothing. GLOBECOM 2020-2020 IEEE Global Communications Conference pp. 1–6 (2020)
7. Gilbert, E.N.: Random graphs. The Annals of Mathematical Statistics **30**(4), 1141–1144 (1959)
8. Gligorijević, V., Renfrew, P.D., Kosciolek, T., Leman, J.K., Berenberg, D., Vatanen, T., Chandler, C., Taylor, B.C., Fisk, I.M., Vlamakis, H., et al.: Structure-based protein function prediction using graph convolutional networks. Nature communications **12**(1), 1–14 (2021)
9. Gowal, S., Dvijotham, K.D., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T., Kohli, P.: Scalable verified training for provably robust image classification. Proceedings of the IEEE/CVF International Conference on Computer Vision pp. 4842–4851 (2019)
10. Huang, Y., Zhang, H., Shi, Y., Kolter, J.Z., Anandkumar, A.: Training certifiably robust neural networks with efficient local lipschitz bounds. Advances in Neural Information Processing Systems **34**, 22745–22757 (2021)
11. Jia, J., Wang, B., Cao, X., Gong, N.Z.: Certified robustness of community detection against adversarial structural perturbation via randomized smoothing. Proceedings of The Web Conference 2020 pp. 2718–2724 (2020)
12. Lee, G.H., Yuan, Y., Chang, S., Jaakkola, T.: Tight certificates of adversarial robustness for randomly smoothed classifiers. Advances in Neural Information Processing Systems **32** (2019)
13. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. International Conference on Learning Representations, ICLR (2019)
14. Raghunathan, A., Steinhardt, J., Liang, P.S.: Semidefinite relaxations for certifying robustness to adversarial examples. Advances in Neural Information Processing Systems **31** (2018)
15. Sugiyama, M., Borgwardt, K.: Halting in random walk kernels. Advances in neural information processing systems **28** (2015)
16. Wang, B., Jia, J., Cao, X., Gong, N.Z.: Certified robustness of graph neural networks against adversarial structural perturbation. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining pp. 1645–1653 (2021)
17. Welling, M., Kipf, T.N.: Semi-supervised classification with graph convolutional networks. International Conference on Learning Representations, ICLR (2017)
18. Wu, H., Wang, C., Tyshetskiy, Y., Docherty, A., Lu, K., Zhu, L.: Adversarial examples on graph data: Deep insights into attack and defense. arXiv preprint arXiv:1903.01610 (2019)
19. Wu, L., Chen, Y., Shen, K., Guo, X., Gao, H., Li, S., Pei, J., Long, B.: Graph neural networks for natural language processing: A survey. arXiv preprint arXiv:2106.06090 (2021)

A Proofs of propositions

A.1 Proof of Proposition 1

Disjoint Unions. Let $\mathbf{z} \in \mathcal{R}_{\mathbf{Q}}$ and $\tilde{\mathbf{z}} \in \mathcal{R}_{\mathbf{Q}'}$ such that for some $i \in I$ we have $Q_i \neq Q_i'$. If $\mathbf{z} = \tilde{\mathbf{z}}$, it implies that $||\mathbf{z}_{\mathcal{J}_i} - \mathbf{x}_{\mathcal{J}_i}|| = Q_i$ and $||\tilde{z}_{\mathcal{J}_i} - \mathbf{x}_{\mathcal{J}_i}|| = Q_i'$ which is a contradiction.
Partition. $|\mathcal{J}_i| \leq R_i$, and $||\mathbf{z}_{\mathcal{J}_i} - \mathbf{x}_{\mathcal{J}_i}|| \leq Q_i$ hence $\mathcal{X} = \cup_{Q \leq R} \mathcal{R}_Q^R$.

A.2 Proof of Proposition 2

As the noise for each entry is independent we can decompose the probabilities as so

$$\frac{P(\phi(\tilde{\mathbf{x}}) = \mathbf{z})}{P(\phi(\mathbf{x}) = \mathbf{z})} = \prod_{k \in [N]} \frac{P(\phi(\tilde{\mathbf{x}})_k = \mathbf{z}_k)}{P(\phi(\mathbf{x})_k = \mathbf{z}_k)}. \tag{14}$$

Furthermore, as each components belongs to exactly one edge community.

$$\prod_{k \in [N]} \frac{P(\phi(\tilde{\mathbf{x}})_k = \mathbf{z}_k)}{P(\phi(\mathbf{x})_k = \mathbf{z}_k)} = \prod_{i=1}^{I} \prod_{k \in \mathcal{C}_i} \frac{P(\phi(\tilde{\mathbf{x}})_k = \mathbf{z}_k)}{P(\phi(\mathbf{x})_k = \mathbf{z}_k)}. \tag{15}$$

We note that for k where $\tilde{\mathbf{x}}_k = \mathbf{x}_k$ this fraction is one, so we can focus on terms when $\tilde{\mathbf{x}}_k \neq \mathbf{x}_k$. In equations this can be written as

$$\prod_{i=1}^{I} \prod_{k \in \mathcal{C}_i} \frac{P(\phi(\tilde{\mathbf{x}})_k = \mathbf{z}_k)}{P(\phi(\mathbf{x})_k = \mathbf{z}_k)} = \prod_{i=1}^{I} \prod_{k \in \mathcal{J}_i} \frac{P(\phi(\tilde{\mathbf{x}})_k = \mathbf{z}_k)}{P(\phi(\mathbf{x})_k = \mathbf{z}_k)} \tag{16}$$

We can consider what the terms are equal to when $\mathbf{x}_k = \mathbf{z}_k$ and when $\mathbf{x}_k \neq \mathbf{z}_k$ (assuming that $\mathbf{x}_k \neq \tilde{\mathbf{x}}_k$). We get

$$\frac{P(\phi(\tilde{\mathbf{x}})_k = \mathbf{z}_k)}{P(\phi(\mathbf{x})_k = \mathbf{z}_k)} = \begin{cases} \frac{p_i}{1-p_i} & \text{if } \mathbf{x}_k = \mathbf{z}_k \text{ and } \mathbf{x}_k \neq \tilde{\mathbf{x}}_k \\ \frac{1-p_i}{p_i} & \text{if } \mathbf{x}_k \neq \mathbf{z}_k \text{ and } \mathbf{x}_k \neq \tilde{\mathbf{x}}_k \end{cases}. \tag{17}$$

In total there are R_i terms in each product, of which Q_i are the first case and $R_i - Q_i$ are in case two. Thus

$$\prod_{i=1}^{I} \prod_{k \in \mathcal{J}_i} \frac{P(\phi(\tilde{\mathbf{x}})_k = \mathbf{z}_k)}{P(\phi(\mathbf{x})_k = \mathbf{z}_k)} = \prod_{i=1}^{I} \left(\frac{p_i}{1-p_i}\right)^{Q_i} \left(\frac{1-p_i}{p_i}\right)^{R_i - Q_i} \tag{18}$$

$$= \prod_{i=1}^{C} \left(\frac{p_i}{1-p_i}\right)^{2Q_i - R_i} \tag{19}$$

$$= \prod_{i=1}^{C} \left(\frac{1-p_i}{p_i}\right)^{R_i - 2Q_i} \tag{20}$$

as required. We provide Fig. 5 as a visual aid to the proof.

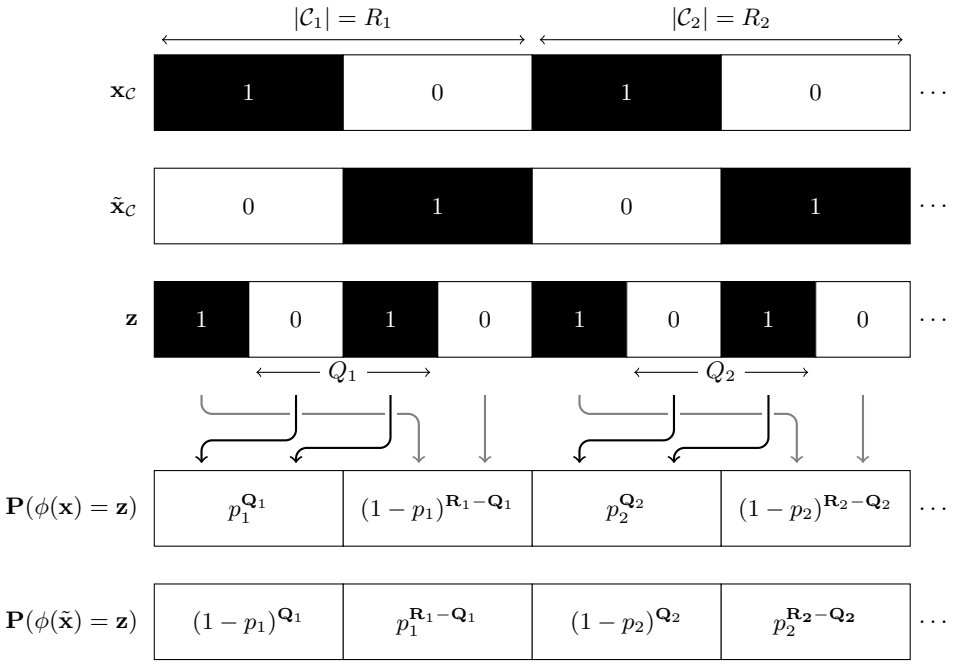


Fig. 5: Pictorial representation of where the terms in Proposition 2 come from.

A.3 Proof of Proposition 3

We have $\mathcal{R}_\mathbf{Q} = \{\mathbf{z} \in \mathcal{X} : \|\mathbf{z}_{\mathcal{J}_i} - \mathbf{x}_{\mathcal{J}_i}\|_0 = Q_i\}$. The probability $\mathbb{P}(\phi(\mathbf{x}) \in \mathcal{R}_\mathbf{Q})$ corresponds to each set R_i having Q_i entries not being flipped or equivalently $R_i - Q_i$ entries being flipped. Each node pair is flipped with a probability of p_i. Since all flips are independent we can express the probability as $\mathbb{P}(\phi(\mathbf{x}) \in \mathcal{R}_\mathbf{Q}) = \prod_{i=1}^{C} \mathrm{Bin}(R_i - Q_i | R_i, p_i)$.

B Implementation

B.1 Estimations of probabilities

The quantities $p_y(\mathbf{x})$ cannot be computed in closed form for general f. Hence, we resolve to lower bound p^* and upper bound $p_y(\mathbf{x}), y \neq y^*$ via sampling. To achieve this, we use the Clopper-Pearson interval [2].

B.2 Symmetries certification

Solving the optimization problem defined in Eq. 7 is difficult as certificates have to be computed for every $\tilde{\mathbf{x}}$ in the ball around $\mathbf{x}$: $\mathcal{B}_r(\mathbf{x})$. However, in practice, $\Phi_{\mathbf{x},\tilde{\mathbf{x}}}(p^*, y^*)$ displays some symmetries depending on the noise distribution $\phi(\mathbf{x})$.

In the case of isotropic noise, the regions $\mathcal{H}_k$ and values c_k only depends on $\|\mathbf{x} - \tilde{\mathbf{x}}\|_0$. This implies $\Phi_{\mathbf{x},\tilde{\mathbf{x}}}(p^*, y^*) = \Phi_{\mathbf{x},\tilde{\mathbf{x}}'}(p^*, y^*)$ for all $\tilde{\mathbf{x}}, \tilde{\mathbf{x}}' \in \mathcal{S}_r(\mathbf{x})$ which reduce the search on every spheres.

In the case of anisotropic noise, the regions $\mathcal{H}_k$ and values c_k only depends on $\|\mathbf{x}_{\mathcal{C}_i} - \tilde{\mathbf{x}}_{\mathcal{C}_i}\|_0$. This implies $\Phi_{\mathbf{x},\tilde{\mathbf{x}}}(p^*, y^*) = \Phi_{\mathbf{x},\tilde{\mathbf{x}}'}(p^*, y^*)$ for all $\tilde{\mathbf{x}}, \tilde{\mathbf{x}}' \in \mathcal{S}_{\mathbf{R}}(\mathbf{x})$.