COMMUNITY-ADAPTIVE RANDOMIZED SMOOTHING FOR GRAPH CLASSIFICATION

Pierre Osselin
Department of Engineering
University of Oxford
pierre.osselin@eng.ox.ac.uk

Henry Kenlay
Department of Engineering
University of Oxford
henry.kenlay@eng.ox.ac.uk

Xiaowen Dong
Department of Engineering
University of Oxford
xdong@robots.ox.ac.uk

ABSTRACT

Certifying the robustness of graph-based model to adversarial attacks becomes paramount to ensure the safety and reliability of these systems. Although well studied for continuous data, robustness certification is a recent field of study for discrete data. We extend the literature on robustness certification based on randomized smooothing for graph classification. In opposition to previous work that only incorporated the sparsity property of the graph into the randomization process, we propose to focus on the community structure of the graph. Our new method results in better certificates on synthetic data and competitive results on real-world data.[1]

1 INTRODUCTION

There is a growing interest into the study graph structured data. Graph neural network [16] is a recent machine learning model that handles this type of data and gained popularity in the last few years. This model is now applied in many different applications, ranging from NLP [22], object detection [20], combinatorial optimization [7], structure-based protein function prediction [9] or fake news detection [18] to name a few.

As these machine learning tools become increasingly widespread and used in production for real-world applications, the robustness of these models against adversarial attacks becomes paramount. Deep learning models are prone to adversarial attacks, hence, certifying safe and reliable autonomous systems is a burning issue [10].

Many different approaches have been introduced to tackle to introduce the notion of robustness. A first approach deals with convex relaxation of the reachable output space given a perturbation model of the data [24] [12]. This relaxation allows the computation of a lower bound on the worst-case margin between this relaxed set and the decision boundary of the classification model. Hence, this method gives a provable guarantee that the prediction does not change under any admissible perturbation with respect to the given perturbation model. Recent work [12] proposed to compute a certificate through a lower a bound between the desired label logit and second largest logit in a one layer GCN for graph classification with a threat model composed of local edge budget per nodes and a global edge budget on the entire graph. However, their certificates rely on an optimization procedure and the particular analytical form of a one layer GCN.

Another approach relies on the computation of a local lipschitz constant around the data points one wants to certify. This method allows to compute lower bounds on the logit of the dominant class, as well as upper bounds on the logits of the undesired class on a certain input regions based on this local lipschitz. The state of the art of this method has only been applied in the continuous case such

[1] Code available at https://github.com/pierreosselin/graphrobustness

as images [13], [17]. For the graph setting work has been carried out to unveil stability properties of graph models [14] based on perturbation of the Laplacian Matrix. No work so far has been carried out regarding potential local lipschitz constant with respect to graphs.

Related to the notion of robustness, recent work focus on quantifying the uncertainty about the edges present in the graph [4]. Assuming that the uncertainty affects only a limited number of edges, the authors make use of small perturbation analysis to derive closed form expressions instrumental to formulate signal processing algorithms that are resilient to imperfect knowledge of the graph topology. This methodology generates models more resilient to perturbation but does not present any certification guarantees.

Finally a recent approach is based on randomized smoothing of a classifier, pioneering work being by Cohen er al. [5]. This methodology has recently been applied in limited settings in the case of graph data. So far this line of work has been limited to the Bernoulli distribution that does not exploit the structure of real-world graph structure data [23] or only its sparsity [1] by distinguishing a probability of edge addition and edge deletion. While computationally efficient and having a simple form, this choice of noise distribution may break the informative features for the underlying tasks.

In this project we aim at generalizing the noise distributions used in this latter research work. We propose a further refinement to incorporate the structural properties of the graph into the design of a noise distribution where the community structure is leveraged. This new randomization procedure results in better certificate radii on a synthetic data set as well as competitive results in a real world data set.

2 BACKGROUND

Randomized Smoothing. Let $\mathcal{X}$ be the data space and $f : \mathcal{X} \to \mathcal{Y}$ be a trained classifier for the labels $\mathcal{Y} = \{0, ..., C\}$. We define a certified radius R at a certain data point $x \in \mathcal{X}$ as a ball centered on x of radius R according to a particular metric in which our classifier is guaranteed to output the same label:

$$f(x) = c \text{ for } c \in \mathcal{Y} \implies \forall \tilde{x} \in \mathcal{B}_{||.||}(x, R), f(\tilde{x}) = c \tag{1}$$

Let $\phi : \mathcal{X} \to \mathcal{P}(\mathcal{X})$ be a randomization process over our data, that is, $\forall x \in \mathcal{X}, \phi(x)$ is a random variable over $\mathcal{X}$. In the Euclidean case of images this process can take many forms. Although a Gaussian distribution is a classical choice and is intimately linked to the l2 norm for certification [5], more refined distributions can be designed (see [23] for a comprehensive review). Let g be a classifier defined as follows:

$$g(x) = \arg\max_{c \in \mathcal{Y}} \mathbb{P}(f(\phi(x)) = c)$$

The classifier g is a smoothed version of the underlying classifier f where the randomization process can be interpreted as a "neighborhood vote": for every element $x \in \mathcal{X}$, graphs are sampled according to the distribution $\phi(x)$, and used for a majority votes when passed through the classifier f to predict the correct label. More details about statistical tests giving confidence intervals for the predicted label will be described in the next subsection. Given a choice of noise distribution for the randomization process the Neyman-Pearson lemma allows us to derive certificate radii for a data point.

Lemma 1 *(Neyman-Pearson) Let X and Y be random variables in $\mathbb{R}^d$ with densities μ_X and μ_Y respectively. Let $h : \mathbb{R}^d \to \{0, 1\}$ be a random or deterministic function. Then:*

1. *If* $S = \left\{ z \in \mathbb{R}^d : \frac{\mu_Y(z)}{\mu_X(z)} \leq t \right\}$ *for some* $t > 0$ *and* $\mathbb{P}(h(X) = 1) \geq \mathbb{P}(X \in S)$ *then* $\mathbb{P}(h(Y) = 1) \geq \mathbb{P}(Y \in S)$
2. *If* $S = \left\{ z \in \mathbb{R}^d : \frac{\mu_Y(z)}{\mu_X(z)} \geq t \right\}$ *for some* $t > 0$ *and* $\mathbb{P}(h(X) = 1) \leq \mathbb{P}(X \in S)$ *then* $\mathbb{P}(h(Y) = 1) \leq \mathbb{P}(Y \in S)$

From this lemma, and in the binary classification case, we can identify the function h with our underlying classifier f and the random variables X and Y with the r.v $\phi(x)$ and $\phi(\tilde{x})$ for some

$x \in \mathcal{X}$ and $\tilde{x} \in \mathcal{B}(x, \epsilon)$. In this case, the lemma allows us to lower bound $\mathbb{P}(f(\phi(\tilde{x})) = c)$ and upper bound $\mathbb{P}(f(\phi(\tilde{x})) = c')\forall\ c' \in \mathcal{Y}/\ \{c\}$. In the case of many noise distributions, a closed form formula can be derived for S and both $\mathbb{P}(X \in S)$ and $\mathbb{P}(Y \in S)$, establishing a direct formula for certified radii depending on the parameters of the distributions and the empirical classification probabilities on the point of interest. [5], [23]

The choice of noise distribution intervenes in two different elements of the randomized smoothing pipeline. First, it dictates the shape of the decision boundary of the smoothed classifier. If poorly chosen, the noise will not solve a model boundary mispecification but perturb data points in a manner that does not make sense for the task at hand. For instance, smoothing a classifier predicting a vertical line with rotation will result in a collapse of accuracy. Secondly, it dictates the computed certified radii. For example, for the same classification probabilities, a distribution with larger noise will result in better certificates. Hence a noise distribution has to be designed to intuitively align with the task at hand (Which neighbours should be, a priori, be classified in the same way?) and have a reasonable smoothing amplitude. These questions become paramount when the data space becomes discrete as we will see next.

Randomized smoothing for graph classsifiers. In this work we are interested in the robustness of graph classification models to structural perturbation of undirected graphs [2]. In this situation $\mathcal{X} = \{0, 1\}^N$ is a flatten binary vectors corresponding to the presence or absence of edges. Given x and $\epsilon \in \mathcal{X}$ we define a perturbed graph as:

$$\tilde{x} = x \oplus \epsilon \tag{2}$$

Where $\oplus$ is the XOR operator. The only two randomized smoothing methods that we are aware of are based on Bernoulli distributions as randomization processes [8] [1]:

1. $\phi(x) = x \oplus \epsilon$ with $\forall i \in [N], \epsilon_i \sim Bern(p)$
2. $P(\phi(x)_i \neq x_i) = p_-^{x_i} p_+^{1-x_i}$

The second noise distribution 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 edge addition will surpass the number of edge deletion and form a unrealistic graph. In the graph setting we are interested in robustness with respect to l_0 perturbations which corresponds to the number of edge flipping.

Contrary to the classical continuous case we do not have a direct formula for the regions S. In both the mentioned research works, the regions S are discrete and computed through the concatenation of the regions of equal likelihood ratios between the input point and a perturbed point. Although the theoretical framework presented are different, the practicalities for certificate computation are the same for the different methods.

Suppose $\mathbb{P}(f(\phi(x)) = c) = \bar{p} > \underline{p} = \max_{c' \in \mathcal{Y}/\{c\}} \mathbb{P}(f(\phi(x)) = c')$. Let $\tilde{x} \in \mathcal{X}$, certifying our smooth classifier is equivalent to verifying, that:

$$\forall \tilde{x} \in \mathcal{B}_0(x, R), \mathbb{P}(f(\phi(\tilde{x})) = c) > \max_{c' \in \mathcal{Y}/\{c\}} \mathbb{P}(f(\phi(\tilde{x})) = c') \text{ for some } R > 0 \tag{3}$$

If one constructs regions $\mathcal{Q}_1$ and $\mathcal{Q}_2$ such that $\mathbb{P}(f(\phi(x)) = c) = \bar{p} = \mathbb{P}(\phi(x) \in \mathcal{Q}_1)$ and $\underline{p} = \mathbb{P}(\phi(x) \in \mathcal{Q}_2)$ and where $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are under the form of S with the corresponding inequality sign, then, the Neyman-Pearson lemma 1 is applicable and gives us bounds for classification probabilities on the perturbed points:

$$\mathbb{P}(f(\phi(\tilde{x})) = c) > \mathbb{P}(\phi(\tilde{x}) \in \mathcal{Q}_1) \text{ and } \max_{c' \in \mathcal{Y}/\{c\}} \mathbb{P}(f(\phi(\tilde{x})) = c') < \mathbb{P}(\phi(\tilde{x}) \in \mathcal{Q}_2) \tag{4}$$

In practice, these regions $\mathcal{Q}$ can be computed via the regions $\mathcal{H}_1, \mathcal{H}_2, ..., \mathcal{H}_K$ of constant likelihood ratio $\mathcal{H}_k = \{z \in \mathcal{X} : \frac{\mu_{\phi(\tilde{x})}(z)}{\mu_{\phi(x)}(z)} = c_k\}$ where the c_k are in an ascending order. More precisely, following notations from [8] with our own corrections of the formula regarding the form of $\mathcal{Q}_2$, let

[2] This work can be extended to graph node classification and directed graphs

μ_1, μ_2 be defined as follows:

$$\mu_1 = \underset{\mu' \in [K]}{\arg\min}, s.t \sum_{k=1}^{\mu_1} \mathbb{P}(\phi(x) \in \mathcal{H}_k) \geq \bar{p}$$

$$\mu_2 = \underset{\mu' \in [K]}{\arg\max}, s.t \sum_{k=\mu_2}^{K} \mathbb{P}(\phi(x) \in \mathcal{H}_k) \leq \underline{p}$$

Since we are dealing with discrete regions and likelihood ratios, it is very unlikely that the sums above ends up with a strict equality with $\underline{p}$ and $\bar{p}$. Thus, we augment the sums with the sub-regions $\mathcal{H}'_{\mu_1}$ and $\mathcal{H}'_{\mu_2}$ defined such that:

$$\mathbb{P}(\phi(x) \in \mathcal{H}'_{\mu_1}) = \bar{p} - \sum_{k=1}^{\mu_1 - 1} \mathbb{P}(\phi(x) \in \mathcal{H}_k)$$

$$\mathbb{P}(\phi(x) \in \mathcal{H}'_{\mu_2}) = \underline{p} - \sum_{k=\mu_2+1}^{K} \mathbb{P}(\phi(x) \in \mathcal{H}_k)$$

Defining the regions $\mathcal{Q}_1 = \bigcup_{k=1}^{\mu_1 - 1} \mathcal{H}_k \bigcup \mathcal{H}'_{\mu_1}$ and $\mathcal{Q}_2 = \bigcup_{k=\mu_2+1}^{K} \mathcal{H}_k \bigcup \mathcal{H}'_{\mu_2}$, the Neyman-Pearson lemma gives us the result described in equation 4. Thus we obtain a robustness certificate if and only if the following equation holds:

$$\mathbb{P}(\phi(\tilde{x}) \in \mathcal{Q}_1) > \mathbb{P}(\phi(\tilde{x}) \in \mathcal{Q}_2) \tag{5}$$

This procedure is seemingly complex, for every data point we want to certify, we need to find the largest l_0 ball radius R such that the classifier is robust, which means computing the regions $\mathcal{Q}_{1,2}$ for every point in the ball and verify that eq. 5 holds. In practice, we can exploit the symmetry from the noise distribution. For example in the case of a bernoulli noise distribution, the regions of equal likelihood only depends on the l_0 distance between our perturbed data point and the point we want to certify. In the sparsity aware noise distribution, there is a symmetry for every edge deletion and addition budget, which is used as a threat model in their paper [1].

This certification procedure also requires the computation of the probabilities $\bar{p}$ and $\underline{p}$. There is no closed form formula for an arbitrary underlying classifier f, hence sampling is necessary to obtain an approximation of these probabilities. Given M samples from the noise distribution around our data point x we obtain a distribution of output samples from the model on the label classes. The output label can be computed as the majority class.Then, the lower-bound probability $\bar{p}$ can be estimated using one-sided Clopper-Pearson method [3]:

$$\bar{p} = LCB(\mu_c, M, 1 - \alpha) \tag{6}$$

Where LCB($\cdot$) denotes the lower-confidence-bound function which returns the one-sided lower confidence interval for the Binomial parameter p such that $\mu_c \sim Binomial(M, p)$ with probability $1 - \alpha$. For the upper bound probability $\underline{p}$, it's hard to give accurate maximum probability among the remaining classes if $|C| > 2$. In practice, we take: $\underline{p} = 1 - \bar{p}$. And in this situation, $\underline{p} < 1 - \bar{p}$ leads $1 - \bar{p} < \bar{p}$, which is $\bar{p} > 0.5$ in return. If $\bar{p} \leq 0.5$, the algorithm will abstain.

3 COMMUNITY-AWARE RANDOMIZED SMOOTHING

3.1 MOTIVATION

We have described the randomized smoothing framework for graph classification. The current literature is limited to either the Bernoulli distribution [8] or a "sparsity aware" Bernoulli distribution [1].

We propose a further refinement to incorporate the structural properties of the graph into the design of a noise distribution in randomized smoothing where the community structure is leveraged.

3.2 FORMULATION

Certificates Formulation. Let V be the set of arbitrarily ordered nodes of a graph. Suppose V is separable in disjoint communities $V = \cup_{i\in[L]} K_i$, $K_i \subset V$. Let $C_{i,j} = \{e \in E : e = (v_r, v_s), v_r, v_s \in K_i \times K_j\}$ the communities of edges interconnecting nodes from two (possibly the same) community. We define the following randomization process ϕ:

$$\forall i, j \in [L], \forall e \in C_{i,j}, P(\phi(x)_e \neq x_e) = p_{i,j} \tag{7}$$

Although not implemented in this work, this process can easily be refined by adding a sparsity aware distribution on every inter-clusters probabilities:

$$\forall i, j \in [L], \forall e \in C_{i,j}, P(\phi(x)_e \neq x_e) = p_{i,j,-}^{x_e} p_{i,j,+}^{1-x_e} \tag{8}$$

Let $x \in \{0,1\}^N$, and $\tilde{x} \in \mathcal{S}_R(x) = \{z \in \{0,1\}^N : \forall i, j \in [L], ||z_{C_{i,j}} - x_{C_{i,j}}||_0 = R_{i,j}\}$. Given a matrix $R = (R_{i,j})_{i,j\in[L]}$, $\mathcal{S}_R$ represents the circle of distance l_0 $R_{i,j}$ in the edges corresponding to the inter cluster $C_{i,j}$.

Given x and perturbation $\tilde{x} \in \mathcal{S}_R(x)$, let $J = \{j \in [N] : x_j \neq \tilde{x}_j\}$ and $J_{i,j} = J \cap C_{i,j}$ the set of perturbed edges between communities C_i and C_j. We define the region $\mathcal{R}_Q^R = \{z \in \{0,1\}^N : \forall i, j \in [L], ||z_{J_{i,j}} - x_{J_{i,j}}||_0 = Q_{i,j}\}$ for $Q = (Q_{i,j})_{i,j\in[L]}$ such that $\forall i, j \in [L], 0 \leq Q_{i,j} \leq R_{i,j}$. Intuitively speaking, the regions $\mathcal{R}_Q^R$ correspond to the points in the data space at a $Q_{i,j}$ l_0 distance from x in the set of perturbed edges between cluster i and j.

Theorem 2 *(Regions of Equal Likelihood for Community Aware Randomized Smoothing)*

Given x and perturbation $\tilde{x} \in \mathcal{S}_R(x)$, the regions of equal likelihood $\{z \in \{0,1\}^N, \frac{P(\phi(x)=z)}{P(\phi(\tilde{x})=z)} = c\}$ for $c \in \mathbb{R}$ are given by the regions $\mathcal{R}_Q^R$ and:

$$\forall z \in \mathcal{R}_Q^R, \ \frac{P(\phi(x) = z)}{P(\phi(\tilde{x}) = z)} = \prod_{i,j\in[L]} \left(\frac{1-\beta_{i,j}}{\beta_{i,j}}\right)^{R_{i,j}-2Q_{i,j}} \tag{9}$$

Furthermore, the regions define a partition of the space $\mathcal{X}$:

$$\mathcal{X} = \cup_{Q\leq R} \mathcal{R}_Q^R \tag{10}$$

Where the $\leq$ operation on matrices is a element-wise inequality.

Finally, we can compute the probability of the randomization process to belong in these regions:

$$P(\phi(x) \in \mathcal{R}_Q^R) = \prod_{i,j\in[L]} B(Q_{i,j}|R_{i,j}, \beta_{i,j}) \textit{where B is the Binomial distribution.} \tag{11}$$

We will give a proof of this theorem in the Appendix A.

Corollary 2.1 *(Binary Case)*

In the Binary case where $|\mathcal{Y}|$ = 2, we have $\mathbb{P}(f(\phi(x)) = c) = 1 - \max_{c'\in\mathcal{Y}/\{c\}} \mathbb{P}(f(\phi(x)) = c')$, hence, eq. 5 reduces to:

$$\mathbb{P}(\phi(\tilde{x}) \in \mathcal{Q}_1) > 0.5 \tag{12}$$

In this case, the computation of $\mathcal{Q}_2$ is unnecessary.

Factorization. In practice, we can exploit the assumption that our graph undirected and the shape of the noise to reduce the complexity of the algorithm.

If the graph is undirected, then the edges are symmetric, $\beta_{i,j} = \beta_{j,i}$ and we only consider the regions of equal likelihood from 9: $\mathcal{R}_Q^R$ $\forall i \leq j \in [L]$. This can greatly reduce the complexity as we will see in the next paragraphs. Similarly, we can "merge" every regions of equal probability. If $\beta_{i,j} = \beta_{k,l}$ then we can concatenate the two sets of edges into a single set with constant β probability.

l_0 radii. As we can observe through the equations 9 10 and 11, the regions of equal likelihood ratios only depends on the variables $(R)_{i,j}$ which represent the number of edge perturbation per pairwise

clusters. As a consequence, certifying a point $\tilde{x} \in \mathcal{S}_R(x)$ means certifying the whole set $\mathcal{S}_R(x)$. As a result, we can obtain a l_0 certification around x via a concatenation of these sets. We use B_0 to denote the ball of the pseudo norm $||.||_0$:

$$\forall l \geq 0, B_0(x,l) = \bigcup_{R:\sum_{i,j} R_{i,j} \leq l} \mathcal{S}_R(x) \tag{13}$$

3.3 PRACTICALITY

Choice of parameter β. Our noise distribution is composed of many parameters that depend on the number of identified clusters, contrary to the aforementioned Bernoulli distributions that consist of one or two parameters. To alleviate this issue we decided to use a community detection algorithm as a pre-processing step on our data sets. As the complexity of the algorithm heavily relies on the number of communities, our detection algorithm needs to have the flexibility to modulate the size and number of communities. To this end we used the Leiden clustering algorithm [21] that allows to tune a resolution parameter controlling the number of clusters. This parameter is tuned as a pre-processing step on the graph classification data set to match an average number of clusters suitable to perform certification. Once communities are detected, the inter-cluster probabilities can be estimated as the ratio between the number of present edges divided by the number of possible edges between communities. Finally, as suggested above, these probabilities can be approximated in such a way that two set of edges with similar probabilities can be merged in order to reduce the computational demands of the algorithm. If we denote P the estimated inter-cluster matrix probability of our graph after applying a community detection algorithm, we choose our noise probability β to be proportional to these probabilities:

$$\forall i,j \in [L],\ \beta_{i,j} = \alpha \times p_{i,j} \tag{14}$$

Complexity in L. From the described algorithm we can estimate its complexity. To certify a test point x we need to certify the sets $\mathcal{S}_R(x)$. For everyone of these sets we need to compute the regions of equal likelihood ratios, the probabilities $P(\phi(x) \in \mathcal{R}_Q^R)$ and the regions $\mathcal{Q}_1$ and $\mathcal{Q}_2$. From the Stars and bars theorem there are $\binom{l+L-1}{L-1}$ sets $\mathcal{S}_R(x)$ such that $\sum R_{i,j} = l$. We can observe that the quantities involved in the products in 9 and 10 can be reused for certification across the different sets $\mathcal{S}_R(x)$, hence we store them in dictionaries in the procedure. To compute the regions $\mathcal{Q}_1$ and $\mathcal{Q}_2$ we need to sort the ratios. Since there are $\prod(R_{i,j}+1)$ of these ratios, we can upper bound this quantity by $\left(1+\frac{l}{L}\right)^L$ from the AM-GM inequality. To wrap-up all these components, we can upper bound the complexity to certify a data set by a maximum of l radius by:

$$C = \mathcal{O}\Big(\underbrace{|\mathcal{D}|}_{\#points} \sum_{n=0}^{l} \underbrace{\binom{n+L-1}{L-1}}_{\#\mathcal{S}} \underbrace{\left(1+\frac{n}{L}\right)^L}_{upperbound \#\mathcal{R}} \Bigg(\underbrace{\frac{L(L+1)}{2}}_{computation\ probas} + \underbrace{L\log(1+\frac{n}{L})}_{sorting} \Bigg) \Big) \tag{15}$$

$$\begin{aligned} C_L &= \mathcal{O}_L\Big(\underbrace{|\mathcal{D}|}_{\#points} \sum_{n=0}^{l} \underbrace{\binom{n+L-1}{L-1}}_{\#\mathcal{S}} \underbrace{\left(1+\frac{n}{L}\right)^L}_{upperbound \#\mathcal{R}} \Bigg(\underbrace{\frac{L(L+1)}{2}}_{computation\ probas} + \underbrace{L\log(1+\frac{n}{L})}_{sorting} \Bigg) \Big) \\ &= \mathcal{O}_L(L^2\binom{l+L}{L}) = \mathcal{O}_L(L^{2+l}) \end{aligned} \tag{16}$$

As shown above the complexity is polynomial in the number of clusters, in our experiences, a number of clusters equal to 3 to 10 yields reasonable computational time.

4 EXPERIMENTS

In this section we present our experimental results to show an increased certification performance on this new noise distribution.

4.1 Experimental Setup

4.1.1 Procedure

In our setup, we compared the certification of the Bernoulli noise distribution with our community-aware randomization. The comparison with the sparsity aware distribution is left for future work. To assess the benefits of this new noise we selected multiple graph classification data sets that we split between a train and test set and trained a GCN model [16] on the train set to predict the correct label. Then, we performed certification for an increasing noise amplitude, tuned via the parameter p for the Bernoulli distribution and α for our distribution. The data set and comparison metrics are presented next.

4.1.2 Dataset

Synthetic Dataset Our first data set is a synthetic data set consisting of 600 graphs of size 60 nodes, half of these graphs are sampled from a Stochastic Block Model [11] of parameter $\begin{pmatrix} 0.2 & 0.02 & 0.02 \\ 0.02 & 0.3 & 0.02 \\ 0.02 & 0.02 & 0.4 \end{pmatrix}$, the other half is sampled from an Erdős–Rényi model [19] where the parameter is chosen to match the number of average edges in the SBMs graphs. We designed a node attribute of dimension one consisting of the Katz Centrality, pre computed during the data generation.

The task is to predict whether or not a given graph is generated from one of this generative process. We design this particular task as its objective directly consists in detecting a strong community structure in the graph, which is a setting where we expect our noise distribution to perform well.

Mutag The Mutag dataset [6] is a collection of nitroaromatic compounds and the goal is to predict their mutagenicity on Salmonella typhimurium. Input graphs are used to represent chemical compounds, where vertices stand for atoms and are labeled by the atom type (represented by one-hot encoding), while edges between vertices represent bonds between the corresponding atoms. It includes 188 samples of chemical compounds with 7 discrete node labels.

REDDIT-BINARY REDDIT-BINARY [2] consists of graphs corresponding to online discussions on Reddit. In each graph, nodes represent users, and there is an edge between them if at least one of them respond to the other's comment. There are four popular subreddits, namely, IAmA, AskReddit, TrollXChromosomes, and atheism. IAmA and AskReddit are two question/answer based subreddits, and TrollXChromosomes and atheism are two discussion-based subreddits. A graph is labeled according to whether it belongs to a question/answer-based community or a discussion-based community.

4.1.3 Evaluation Metric

For each one of these tasks, we computed three metrics. First we plotted the accuracy of our smoothed classifier while the noise amplitude increases, giving a curve of accuracy in function of the average number of edges perturbed. Secondly, we computed the average certificate radius per graph in the test data set against the average number of edges perturbed. Finally, we combined these results to obtain the accuracy of the smoothed classifier against the average certified radius on the test data set. While the first two metrics give useful information about the certification performance, the third metric is the most useful. For a better certification method we expect the curve of the last metric to be "above" the one corresponding to the Bernoulli distribution.

4.2 Results

Synthetic The result of our experiment is given on figure 1. On subfigure 1a we can observe that the accuracy drops far more slowly for our noise distribution, this is expected as we perturbed the graph as to preserve their community structure as much as possible, which is exactly the pattern we are looking for for the task at hand. Subfigure 1b shows the l_0 certificates that we obtain as the average number of perturbed edges increase. We can observe that, the certificates are weaker for the community aware distribution. This is due to the structure of the noise which is not isotropic compared to the bernoulli distribution. As some edge directions as far less perturbed than others, a

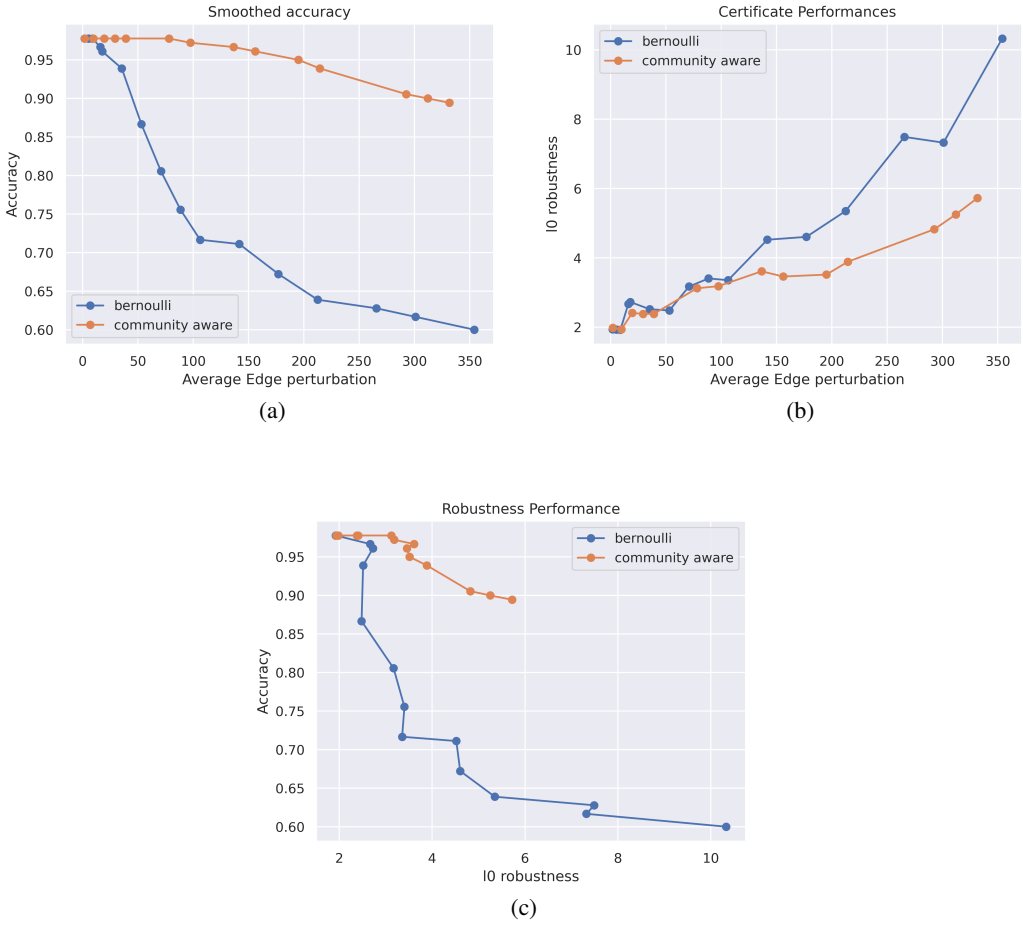
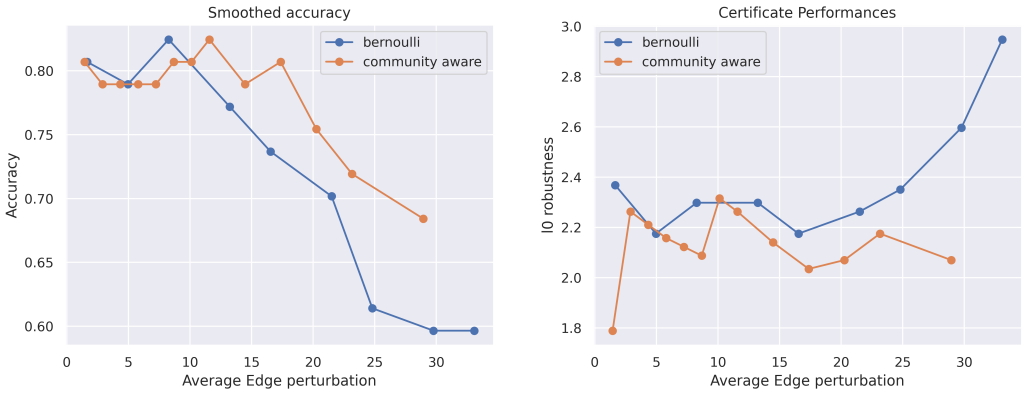


Figure 1: Certification Performance on the synthetic data set

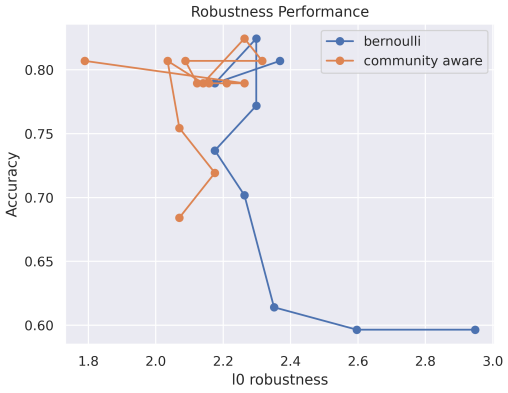
higher empirical probability is required to guarantee the same certification. Finally, combining these two metrics we get subfigure 1c which shows a better performance of our method. We manage, with an initial accuracy of 97% to obtain an average radius of almost 6 for our test data set, while the accuracy of the smoothed classifier with Bernoulli distribution drops at 65% for the same robustness. We showed the plot for the maximum values of the smoothing parameters such that the certification do not break.

Mutag The Mutag data set is an example of of a set of graphs representing molecules that are not prone to community formation. We can observe on Figure 2a that, although the accuracy is higher for our community aware distribution, the improvement is not as significant as in the synthetic data set. We also obtain similar certification performance as shown in Figure 2b. Finally, the resulting certification performance is erratic for both the community aware and Bernoulli distribution as seen on Figure 2c. The performance of the certificates are similar on this data set, both reaching around 2.3 average l_0 robustness while preserving an accuracy of around 82%.

REDDIT The results for the REDDIT data set are shown on Figure 3. On these plots we observe a larger drop in accuracy for the community aware distribution Figure 3a and a similar robustness performance compared to edge perturbation 3b. Finally, the performance of the certification with respect to l_0 robustness is slightly worse in the case of the community aware distribution.

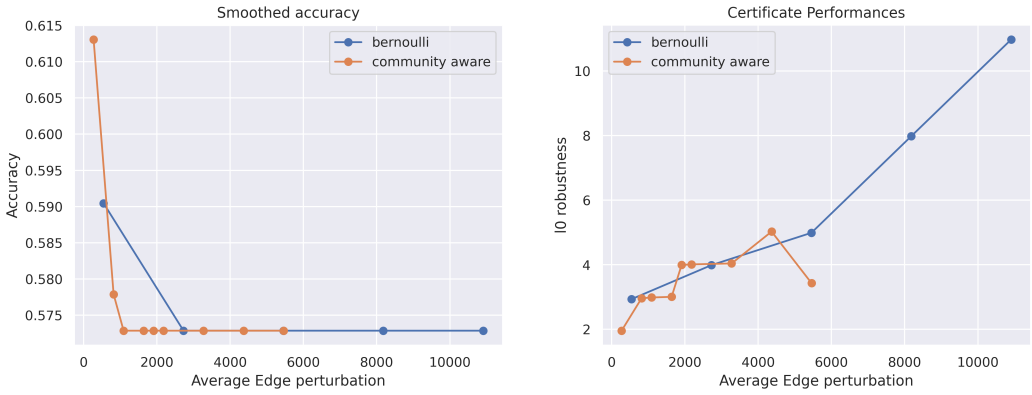


(a) (b)

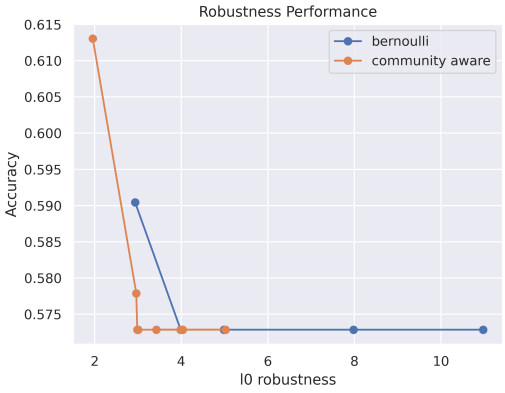


(c)

Figure 2: Certification Performance on the MUTAG data set



(a) (b)



(c)

Figure 3: Certification Performance on the MUTAG data set

5 CONCLUSIONS

In this work we extended the state of the art in randomized smoothing for graphs by incorporating more refined structural information about the graph into the noise design. More specifically, we constructed a randomization process that takes the community structure of the graph into account for neighborhood voting. This method shows better certification performance on our synthetic data set and similar performance on real world data sets.

It is important to notice that the performance of our certificate heavily relies on the treatment of the graph in the pre-processing steps and the estimation of the communities. Here we selected a modularity-based algorithm that can result in a non-uniform distribution on commmunity sizes where some clusters are very small and others very large. Further experiments with different clustering methods need to be carried out. Spectral clustering, which imposes more balanced communities, should be tested. Varying the classification models and types of data set in our experimentation would be helpful to validate this method.

Our method aims at preserving the community structure of the graph. However, the structure of the graph is not limited to its communities and other distributions could be designed to perturb other characteristic such as degree distribution or higher order structures within the graph such as triangles or cliques.

REFERENCES

[1] Aleksandar Bojchevski, Johannes Klicpera, and Stephan Günnemann. Efficient robustness certificates for discrete data: Sparsity-aware randomized smoothing for graphs, images and more. In *International Conference on Machine Learning*, pp. 1003–1013. PMLR, 2020.

[2] Chen Cai and Yusu Wang. A simple yet effective baseline for non-attributed graph classification. *arXiv preprint arXiv:1811.03508*, 2018.

[3] T Tony Cai. One-sided confidence intervals in discrete distributions. *Journal of Statistical planning and inference*, 131(1):63–88, 2005.

[4] Elena Ceci and Sergio Barbarossa. Graph signal processing in the presence of topology uncertainties. *IEEE Transactions on Signal Processing*, 68:1558–1573, 2020.

[5] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pp. 1310–1320. PMLR, 2019.

[6] Asim Kumar Debnath, Rosa L Lopez de Compadre, Gargi Debnath, Alan J Shusterman, and Corwin Hansch. 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.

[7] Iddo Drori, Anant Kharkar, William R Sickinger, Brandon Kates, Qiang Ma, Suwen Ge, Eden Dolev, Brenda Dietrich, David P Williamson, and Madeleine Udell. Learning to solve combinatorial optimization problems on real-world graphs in linear time. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 19–24. IEEE, 2020.

[8] Zhidong Gao, Rui Hu, and Yanmin Gong. Certified robustness of graph classification against topology attack with randomized smoothing. *arXiv preprint arXiv:2009.05872*, 2020.

[9] Vladimir Gligorijević, P Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, et al. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12(1):1–14, 2021.

[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[11] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

[12] Hongwei Jin, Zhan Shi, Venkata Jaya Shankar Ashish Peruri, and Xinhua Zhang. Certified robustness of graph convolution networks for graph classification under topological attacks. *Advances in Neural Information Processing Systems*, 33, 2020.

[13] Matt Jordan and Alexandros G Dimakis. Exactly computing the local lipschitz constant of relu networks. *arXiv preprint arXiv:2003.01219*, 2020.

[14] Henry Kenlay, Dorina Thanou, and Xiaowen Dong. On the stability of polynomial spectral graph filters. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5350–5354. IEEE, 2020.

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[16] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[17] Klas Leino, Zifan Wang, and Matt Fredrikson. Globally-robust neural networks. *arXiv preprint arXiv:2102.08452*, 2021.

[18] Federico Monti, Fabrizio Frasca, Davide Eynard, Damon Mannion, and Michael M Bronstein. Fake news detection on social media using geometric deep learning. *arXiv preprint arXiv:1902.06673*, 2019.

[19] Berndt Müller, Joachim Reinhardt, and Michael T Strickland. *Neural networks: an introduction*. Springer Science & Business Media, 1995.

[20] Weijing Shi and Raj Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1711–1719, 2020.

[21] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12, 2019.

[22] Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, and Bo Long. Graph neural networks for natural language processing: A survey. *arXiv preprint arXiv:2106.06090*, 2021.

[23] Greg Yang, Tony Duan, J Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. In *International Conference on Machine Learning*, pp. 10693–10705. PMLR, 2020.

[24] Daniel Zügner and Stephan Günnemann. Certifiable robustness and robust training for graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 246–256, 2019.

A PROOF OF THEOREM 2

In this section of the appendix we prove Theorem 2. First we show equation 9:

Let x and $\tilde{x} \in \mathcal{S}_R(x)$, let $\tilde{J} = [N]/J$, the set of indices corresponding to edges that were not perturbed in $\tilde{x}$.

$$\begin{aligned}
\forall z \in \mathcal{R}_Q^R,\ \frac{P(\phi(x)=z)}{P(\phi(\tilde{x})=z)} &= \frac{\prod_{i\in\tilde{J}} P(\phi(x)_i = z_i)\prod_{i\in J} P(\phi(x)_i = z_i)}{\prod_{i\in\tilde{J}} P(\phi(\tilde{x})_i = z_i)\prod_{i\in J} P(\phi(\tilde{x})_i = z_i)} \\
&= \frac{\prod_{i\in J} P(\phi(x)_i = z_i)}{\prod_{i\in J} P(\phi(\tilde{x})_i = z_i)} = \frac{\prod_{i,j\in[L]} \beta_{i,j}^{Q_{i,j}}(1-\beta_{i,j})^{R_{i,j}-Q_{i,j}}}{\prod_{i,j\in[L]}(1-\beta_{i,j})^{Q_{i,j}}\beta_{i,j}^{R_{i,j}-Q_{i,j}}} \\
&= \prod_{i,j\in[L]} \left(\frac{1-\beta_{i,j}}{\beta_{i,j}}\right)^{R_{i,j}-2Q_{i,j}}
\end{aligned}$$

Next we prove that the regions $\mathcal{R}_Q^R$ form a partition of $\mathcal{X}$: Let $z \in \mathcal{R}_Q^R$ and $\tilde{z} \in \mathcal{R}_{Q'}^R$ such that $\exists i,j \in [L] Q_{i,j} \neq Q'_{i,j}$. If $z = \tilde{z}$, it implies that $||z_{J_{i,j}} - x_{J_{i,j}}|| = Q_{i,j}$ and $||\tilde{z}_{J_{i,j}} - x_{J_{i,j}}|| = Q'_{i,j}$ which is a contradiction. Furthermore, if $|J_{i,j}| = R_{i,j}$, hence, $||z_{J_{i,j}} - x_{J_{i,j}}|| \leq Q_{i,j}$ hence $\mathcal{X} = \cup_{Q\leq R}\mathcal{R}_Q^R$. Finally the binomial distribution in 11 come directly from the probability to perturb $Q_{i,j}$ edges among $R_{i,j}$ possible edges in every inter cluster configurations.

B EXPERIMENTAL DETAILS

In our experiments we trained a GCN [16] model consisting of three convolutional layers with 64 hidden channels and used an average pooling and a dropout layer with parameter 0.5. For training we used an Adam optimizer [15] with parameter $\alpha = 0.01$. For the data pre processing step we used the Leiden clustering algorithm [21] where the resolution parameter is tuned manually to get an average of 3 or 4 clusters in the considered data set. We used a train/test slit ratio of 70/30 and performed certification on the test set. To evaluate the classification probabilities, we used $200,000$ samples for every graphs that are passed through the model for prediction. We found that this number of samples are enough to have non trivial certificates. We also used a confidence interval $\alpha = 0.01$.