Efficient Robustness Certificates for Discrete Data

A. Proofs

Proof (Prop. 1). First we show that the regions are disjoint. Let \( z \in R_{i}^{r_{i},r_{d}} \) and \( z \in R_{j}^{r_{j},r_{d}} \) for some \( i \neq j \). From the definition of a region it follows that \( \|x_{C} - z_{C}\|_{0} = i \) and \( \|x_{C} - z_{C}\|_{0} = j \). This can be true only if \( i = j \) which is a contradiction. Therefore, \( z \) cannot belong to two different regions. For any \( z \) and \( x \), \( \|x_{C} - z_{C}\|_{0} \in \{0, \ldots, r_{a}+r_{d}\} \) since the \( \| \cdot \|_{0} \) (Hamming) distance between two \( C \)-dimensional vectors has the range \( \{0, \ldots, |C|\} \). Thus, any \( z \) must land in some region \( R_{q}^{r_{a},r_{d}} \) with \( q \leq |C| \), and for any \( q > |C| \) we have \( R_{q}^{r_{a},r_{d}} = \emptyset \). Therefore,

\[
\mathcal{X} = \bigcup_{q=0}^{\infty} R_{q}^{r_{a},r_{d}} = \bigcup_{q=0}^{q=r_{a}+r_{d}} R_{q}^{r_{a},r_{d}}. \quad \square
\]

Proof (Prop. 2). For any \( x, \tilde{x} \in S_{r_{a},r_{d}}(x) \), and \( R_{q}^{r_{a},r_{d}} \):

\[
\Pr(\phi(x) \in R_{q}^{r_{a},r_{d}}) = \Pr(\|x_{C} - \phi(x)\|_{0} = q) = \Pr(\sum_{i \in C} \|x_{i} - \phi(x)_{i}\| = q) = \Pr(\sum_{i \in C} \epsilon_{i} = q)
\]

where \( \epsilon_{i} \sim \text{Ber}(p = p_{a}^{-}p_{a}^{+}(1-x_{i}^{-})) \). 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 \( x \in R_{q}^{r_{a},r_{d}} \) we have \( \sum_{i \in C} x_{i} = r_{a} \) and \( \sum_{i \in C} 1 - x_{i} = r_{a} \). Therefore, \( \sum_{i \in C} \epsilon_{i} \sim Q \) where \( Q = \text{PB}([p_{a}, r_{a}],[p_{a}, r_{a}]) \). \( \square \)

Proof (Prop. 3). For any \( z \in R_{q}^{r_{a},r_{d}} \), by definition it holds \( \|x_{C} - z_{C}\|_{0} = q \). Let \( q_{-} = \sum_{i=1}^{d} \|x_{i} - 1 - z_{i}\| \) and \( q_{+} = \sum_{i=1}^{d} \|x_{i} + 1 - z_{i}\| = q - q_{+} + q_{-} \). We have:

\[
\eta_{q}^{r_{a},r_{d}} = \frac{\Pr(\phi(x) = z)}{\Pr(\phi(x) = z)} = \prod_{i \in \hat{C}} \Pr(\phi(x)_{i} = z_{i}) \prod_{j \in \hat{C}} \Pr(\phi(x)_{j} = z_{j}) = \prod_{i \in \hat{C}} \Pr(\phi(x)_{i} = z_{i}) \prod_{j \in \hat{C}} \Pr(\phi(x)_{j} = z_{j}) = \prod_{i \in \hat{C}} \Pr(\phi(x)_{i} = z_{i}) \prod_{j \in \hat{C}} \Pr(\phi(x)_{j} = z_{j}) = \frac{p_{a}^{q_{-}}}{1 - p_{a}^{-}} \frac{1 - q_{-}}{r_{a} - q_{-}} \frac{p_{a}^{q_{+}}}{1 - p_{a}^{+}} \frac{1 - q_{+}}{r_{a} - q_{+}} = \frac{p_{a}^{q_{-}}}{1 - p_{a}^{-}} \frac{1 - q_{-}}{r_{a} - q_{-}} \frac{p_{a}^{q_{+}}}{1 - p_{a}^{+}} \frac{1 - q_{+}}{r_{a} - q_{+}} = p_{a}^{q_{-} - q_{+}} (1 - p_{a}^{-})^{q_{+}} (1 - p_{a}^{+})^{q_{-}} = p_{a}^{q_{-} - q_{+}} (1 - p_{a}^{-})^{q_{+}} (1 - p_{a}^{+})^{q_{-}} = \frac{1}{1 - p_{a}^{-}} (1 - p_{a}^{+})
\]

Where the second equality holds since \( \phi \) is independent per dimension, and the third equality holds since \( x \) and \( \tilde{x} \) agree on \( \hat{C} \). Plugging in the definition of \( \phi \) and rearranging we obtain \( \eta_{q}^{r_{a},r_{d}} \). Thus, the ratio is constant for any \( z \in R_{q}^{r_{a},r_{d}} \).

Now we show that the ratio is a monotonic function of \( q \):

\[
\eta_{q}^{r_{a},r_{d}} = \left[ \frac{p_{a}^{-}}{1 - p_{a}^{-}} \right]^{q_{-} - q_{+}} \left[ \frac{p_{a}^{+}}{1 - p_{a}^{+}} \right]^{q_{+} - q_{-}} = C \cdot \left[ \frac{p_{a}^{-} + p_{a}^{+}}{p_{a}^{-} + p_{a}^{+} + 1 - (p_{a}^{-} + p_{a}^{+})} \right]^{q_{-} - q_{+}}
\]

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

B. Multi-Class Certificates

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

\[
\begin{align*}
\mu_{x,\hat{x}}(p_{1}(x), \ldots, p_{y}(x), y^*)
&= \min_{h \in \mathcal{H}} \Pr(h(\phi(\hat{x})) = y^*) - \max_{y \neq y^*} \Pr(h(\phi(\hat{x})) = y) \\
&\text{s.t. } \Pr(h(\phi(x)) = y^*) = p_{y^*}, \quad \Pr(h(\phi(x)) = y) = p_{y}, \quad y \neq y^*
\end{align*}
\]

where \( y^* \) is the (predicted or ground-truth) class we want to certify. Similar to before computing \( p_{y^*}(x) \) exactly is difficult, thus we compute a lower bound \( p_{y^*}(x) \) for \( y^* \) and an upper bound \( p_{y}(x) \) 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 \( \alpha \) 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 \( \alpha \) that the estimates hold simultaneously.

The problem in Eq. 9 is valid if \( p_{y^*}(x) + p_{y}(x) < 1 \). The binary-class certificate assumes that \( p_{y}(x) = 1 - p_{y^*}(x) \). 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 \( p_{y^*}(x) + p_{y}(x) < 1 \). Note that, however, the value of \( p_{y^*}(x) \) 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 \( x \) and a perturbation set \( B_{r_{a},r_{d}}(x) \) if it holds that: \( \min_{x \in B_{r_{a}}(x)} \mu_{x,\hat{x}}(p_{1}(x), \ldots, p_{y}(x), y^*) > 0 \) we can guarantee that classification margin for the worst-case classifier is always bigger than 0 for all \( \hat{x} \in B_{r_{a}}(x) \). This implies that \( g(x) = g(\hat{x}) = y^* \) for any input within the ball, i.e. \( x \) is certifiably robust. Compare this to the previous certificate where we had to verify whether \( \rho_{x,\hat{x}}(p^*, y^*) > 0.5 \) which was not tight for \( |\mathcal{Y}| > 2 \).
we have that
which have all valid (reachable via deletion) configurations when applying the randomization
where

The exact solution to the LP is easily obtained with another greedy algorithm: first sort the regions such that \( c_1 \geq c_2 \geq \cdots \geq c_t \), then iteratively assign \( n_i = 1 \) in decreasing order for all regions \( \mathcal{R}_i \) until the constraint \( p_{\phi}(x) \) is met. Finally, iteratively assign \( t_j = 1 \) now in increasing order for all regions \( \mathcal{R}_j \) until the constraint \( p_{\overline{\phi}}(x) \) 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 \( z \) which can be reached by both \( x \) and \( \overline{x} \) when applying the randomization \( \phi \) and these are the vectors which have all valid (reachable via deletion) configurations of ones and zeros in \( \mathcal{C} \) and all zeros in \( \mathcal{C} \). This holds since \( x_\mathcal{C} \) and \( \overline{x}_\mathcal{C} \) are complementary and we can only delete edges. See Fig. 1 for an illustration. Denoting this region with \( \mathcal{R}_1 \) we have that \( \Pr(\phi(x) \in \mathcal{R}_1) = p_+^d \) and \( \Pr(\phi(\overline{x}) \in \mathcal{R}_1) = p_-^d \) 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 \( x \) and this happens with probability \( \Pr(\phi(x) \in \mathcal{R}_2) = 1 - p_+^d \). By definition the vectors in the intersection reachable by both \( x \) and \( \overline{x} \) are all in \( \mathcal{R}_1 \), thus \( \Pr(\phi(\overline{x}) \in \mathcal{R}_2) = 0 \). Finally, the third region \( \mathcal{R}_3 \) corresponds to the case where we flip less than \( r_a \) bits in \( x \), we have \( \Pr(\phi(x) \in \mathcal{R}_3) = 1 - p_-^d \) and \( \Pr(\phi(\overline{x}) \in \mathcal{R}_3) = 0 \). For the binary class certificate we can ignore any regions \( \mathcal{R}_j \) where \( \Pr(\phi(x) \in \mathcal{R}_j) = 0 \), 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:\n\[
\Pr(\phi(x) \in \mathcal{R}_1') = p_+^d \quad \text{and} \quad \Pr(\phi(\overline{x}) \in \mathcal{R}_1') = p_-^d \quad \text{for the first region;}
\]
\[
\Pr(\phi(x) \in \mathcal{R}_2') = 1 - p_+^d \quad \text{and} \quad \Pr(\phi(\overline{x}) \in \mathcal{R}_2') = 0 \quad \text{for the second region;}
\]
\[
\Pr(\phi(x) \in \mathcal{R}_3') = 1 - p_-^d \quad \text{and} \quad \Pr(\phi(\overline{x}) \in \mathcal{R}_3') = 0 \quad \text{for the third region.}
\]

D. Traversal of Regions

As we discussed in § 4.3 we can efficiently compute \( \rho_{x,\overline{x}} \) by directly visiting the regions \( \mathcal{R}_i^{a,r_d} \) in decreasing order w.r.t. the ratio \( \eta_{\mathcal{R}_i^{a,r_d}} \) 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 \( \rho_{x,\overline{x}} \) is computed we simply have to check whether \( \rho_{x,\overline{x}} > 0.5 \) to certify the input \( x \) w.r.t. the given radii \( r_a \) and \( r_d \). The algorithm for the multi-class certificate \( \mu_{x,\overline{x}} \) is similar.

Algorithm 1 Compute \( \rho_{x,\overline{x}} \)

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

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 \( x^A = \text{vec}(A) \in \{0,1\}^{n\times n} \) and \( x^F = \text{vec}(F) \in \{0,1\}^{n\times m} \) denote the flattened adjacency and feature matrix respectively. Let \( x = [x^A, x^F] \in \mathcal{X}^{A,F} \) where \( \mathcal{X}^{A,F} = \{0,1\}^{n\times n \times n\times m} \). We apply the randomization schemes independently: for the graph \( \phi(x^A) \) with \( p_+^A, p_-^A \), and for the attributes \( \phi(x^F) \) with \( p_+^F, p_-^F \). We define the region:

\[
\mathcal{R}_{q_{A,F}}^{a,r_d,a_F,F}^{A,F} = \{ z = [z^A, z^F] \in \mathcal{X}^{A,F} : \}
\]
\[
z^A \in \mathcal{R}_{q_{A,F}}^{A,a_F,a}^{A,F}, z^F \in \mathcal{R}_{q_{A,F}}^{A,F}^{a_F,F} \}
\]

where \( \mathcal{R}_{q_{A,F}}^{a_A,a_F} \) and \( \mathcal{R}_{q_{A,F}}^{a_A,a_F} \) are defined similar to before. We have that the regions \( \mathcal{R}_{q_{A,F}}^{a_A,a_F,f_r} = \{ \mathcal{R}_{q_{A,F}}^{a_A,a_F,f_r} \} \) partition the space \( \mathcal{X}^{A,F} \). 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 \((r_a^A + r_d^A + 1)(r_a^F + r_d^F + 1)\).
As before we can compute $\Pr(\phi(x) \in R^{A}_{q,q'}, r^{A}_{a}, r^{F}_{d}) = \Pr(\phi(x^A) \in R^{A}_{q,q'}, r^{A}_{a}, r^{F}_{d}) \cdot \Pr(\phi(x^F) \in R^{F}_{q', q}, r^{F}_{d})$. Similarly we have for the ratio:

$$\frac{\Pr(\phi(x) \in R^{A}_{q,q'}, r^{A}_{a}, r^{F}_{d})}{\Pr(\phi(\tilde{x}) \in R^{A}_{q', q}, r^{A}_{a}, r^{F}_{d})} = \frac{r^{A}_{q,q'} \cdot r^{F}_{d}}{r^{A}_{q', q} \cdot r^{F}_{d}}$$

The above directly follows from the definition of the regions and because $\phi(x^A)$ is independent of $\phi(x^F)$. Given the values of $\eta_{q,q'}$ and $\Pr(\phi(x) \in R^{q,q'})$ for all $q, q'$ we can again apply the greedy algorithm to compute $\rho_{\tilde{x}, \tilde{x}}$. Note that this can be trivially extended to certify arbitrary groupings of $x$ into subspaces with different radii/flip probabilities per subspace, however, the complexity quickly increases and in general the number of regions will be $O((r^{max}_{a} + r^{max}_{d} + 1)^v)$ where $v$ is the number of groupings and $r^{max}_{a}, r^{max}_{d}$ 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 $F$ and works for the GCN model (Kipf & Welling, 2017); and Bojchevski & Günnemann (2019a)’s certificate which can only handle attacks on $A$ and works for small classes 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 $F$ 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 shown 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 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 ($\Delta$) in the certified 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.
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 $\phi$ on $x = \text{vec}(A_i)$ where $A_i$ is the adjacency matrix of the graph $G_i$. Then, we certify $g$ simply by calculating $\rho_{x,\hat{x}}$ or $\rho_{x,\hat{x}}$. 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 $g_y(x) = \mathbb{E}_{x' \sim \phi(x)}[f(x')]_y$ for class $y$ with $m$ Monte Carlo samples $g_y(x) \approx \sum_{i=1}^m f(x^{(i)})_y$, and we compute the cross-entropy loss with $l(g(x), y)$. 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 $\Delta$ 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 the confidence level $\alpha = 0.01$ and the number of samples to $10^6$ ($10^3$ for MNIST and ImageNet). 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. $B_{r_a, r_d}(x)$ we can simply compute the certificate w.r.t. $S_{i,j}(x)$ for all $0 \leq i \leq r_a, 0 \leq j \leq r_d$. In practice, we compute the maximum $r_a$ and $r_d$ for a given $p_{g^+}(x)$ and $p_{g^-}(x)$ such that the input is certifiably robust. Whenever the number of majority votes is the same for several inputs, they have the same $p_{g^+}(x)$ and $p_{g^-}(x)$ so we only need to compute the maximum radius once to certify all of them.
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 $p_-$ improves over the accuracy of the base classifier (bottom-left corner, $p_-=0, p_+=0$), while increasing $p_+$ slightly decreases the accuracy. We can interpret the perturbation as dropout (except applied during both training and evaluation) which has been previously shown to improve performance (Klicpera 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 starts decreasing quicker 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 $\rho_{x,\hat{x}}$ and the multi-class certificate $\mu_{x,\hat{x}}$. 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 $p_\rho(x)$ 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 $\alpha$.

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 $C$ where $x$ and $\hat{x}$ 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 $\rho_{x,\hat{x}}$ or $\mu_{x,\hat{x}}$ and hence the certificate is the same.
Intuitively, we have variables $q_0, q_1, q_2$ corresponding to the dimensions where $z \in C$, variables $p_0, p_1, p_2$ corresponding to the dimensions where $z \not\in C$, and variables $s_0, s_1, s_2$ corresponding to the dimensions where $z_r$ matches both $x$ and $\bar{x}$ is not possible since by definition $x_i \neq \bar{x}_i$ for all $i \in C$. We define the region parametrized by $(q_j, p_j, s_j)$ triplets:

$$R_{q_0,q_1,q_2} = \{ z \in X_K :$$

$$q_0 = \sum_{i \in C} \mathbb{I}(z_i = x_i)\mathbb{I}(x_i = 0),$$

$$q_1 = \sum_{i \in C} \mathbb{I}(z_i = x_i)\mathbb{I}(\bar{x}_i = 0),$$

$$q_2 = \sum_{i \in C} \mathbb{I}(z_i = \bar{x}_i)\mathbb{I}(\bar{x}_i \neq 0)\mathbb{I}(x_i \neq 0),$$

$$p_0 = \sum_{i \in C} \mathbb{I}(z_i = \bar{x}_i)\mathbb{I}(x_i = 0),$$

$$p_1 = \sum_{i \in C} \mathbb{I}(z_i = \bar{x}_i)\mathbb{I}(\bar{x}_i = 0),$$

$$p_2 = \sum_{i \in C} \mathbb{I}(z_i = \bar{x}_i)\mathbb{I}(x_i \neq 0)\mathbb{I}(\bar{x}_i \neq 0),$$

$$s_0 = \sum_{i \in C} \mathbb{I}(z_i \neq x_i)\mathbb{I}(z_i \neq x_i)\mathbb{I}(x_i = 0),$$

$$s_1 = \sum_{i \in C} \mathbb{I}(z_i \neq x_i)\mathbb{I}(z_i \neq \bar{x}_i)\mathbb{I}(\bar{x}_i = 0),$$

$$s_2 = \sum_{i \in C} \mathbb{I}(z_i \neq x_i)\mathbb{I}(z_i \neq \bar{x}_i)\mathbb{I}(x_i \neq 0)\mathbb{I}(\bar{x}_i \neq 0)$$

for a given clean $x \in X_K$ and adversarial $\bar{x} \in S_{r_0,r_1,r_2}(x)$ which is defined subsequently.

We use $a_0 = 1 - p_+$ as a shorthand for the probability to keep (not flip) a zero, $b_0 = \frac{p_+}{p_+ - 1}$ for the probability to flip a zero to some other value, and $c_0 = 1 - a_0 - b_0$. Similarly we define $a_1 = 1 - p_-$, $b_1 = \frac{p_-}{p_- - 1}$, 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 $R_{q_0,q_1,q_2}$ region equals:

$$\eta = \frac{Pr(\phi(x) \in R_{q_0,q_1,q_2})}{Pr(\phi(\bar{x}) \in R_{q_0,q_1,q_2})} = \frac{\sum_{i = 0}^1 \sum_{j = 0}^1 \sum_{k = 0}^1 q_0^{j - s_0} q_1^{j - s_1} q_2^{j - s_2}}{\sum_{i = 0}^1 \sum_{j = 0}^1 \sum_{k = 0}^1 q_0^{j - s_0} q_1^{j - s_1} q_2^{j - s_2}}$$

(11)

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

$$Pr(\phi(x) \in R_{q_0,q_1,q_2}) = \prod_{j = 0}^2 Pr(u_j = [q_j, p_j, s_j])$$

(12)

where $u_j$ are the following Multinomial random variables:

$$u_0 \sim \text{Mul}([a_0, b_0, c_0], r_0)$$

$$u_1 \sim \text{Mul}([a_1, b_1, c_1], r_1)$$

$$u_2 \sim \text{Mul}([a_1, b_1, c_1], r_2)$$

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: $z_i = x_i$, $z_i = \bar{x}_i$, and $z_i \neq x_i \neq \bar{x}_i$ for all $i \in C$.

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:

$$S_{r_0,r_1,r_2}(x) = \{ \bar{x} \in X_K : \sum_{i = 1}^d \mathbb{I}(x_i = 0)\mathbb{I}(x_i \neq \bar{x}_i) = r_0, \sum_{i = 1}^d \mathbb{I}(\bar{x}_i = 0)\mathbb{I}(x_i \neq \bar{x}_i) = r_1, \sum_{i = 1}^d \mathbb{I}(x_i \neq 0)\mathbb{I}(\bar{x}_i \neq x_i)\mathbb{I}(x_i \neq 0)\mathbb{I}(\bar{x}_i \neq 0) = r_2 \}$$

Similarly, we define the respective ball $B_{r_0,r_1,r_2}(x)$ 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 $c_0 = \frac{(K-2)p_+}{K-1}$ and $c_1 = \frac{(K-2)p_-}{K-1}$ also have to be zero.

In order to partition the entire space $X_K$, 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.
since the ratio in Eq. 11 is the same for certain combinations of \( q_j \)'s, \( p_j \)'s, and \( s_j \)'s, e.g. 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 \( \phi(x) \) 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:

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

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 \( \phi(x) \) lands in the respective \( R_{q',p'} \) region also simplifies (see Fig. 13):

\[
\Pr(\phi(x) \in R_{q',p'}) = \Pr(u = [q', p', r - q' - p'])
\]

where \( u \sim \text{Mul}([a_0, b_0, c_0], r) \). Moreover, we have that \( q' \in \{0, \ldots, r_0 + r_1 + r_2\} = \{0, \ldots, r\} \), where \( \|x - \tilde{x}\|_0 = r \). 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. \( r^A = r^F = 0 \) and varying \( r^A, r^F \). 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.