Robustness and Generalization in Uncertainty-Aware Message Passing Neural Networks

Alesia Chernikova **Moritz Laber** **Narayan G. Sabhahit** **Tina Eliassi-Rad**
Northeastern University, Boston, MA 02115, USA

Abstract

Existing theoretical guarantees for message passing neural networks (MPNNs) assume deterministic node features. We address a more realistic scenario where noise or finite measurement precision introduces uncertainties in node feature values. First, we quantify uncertainty by propagating the moments of node-feature distributions through the MPNN architecture. To propagate moments through activation functions, we use the Taylor expansion and the pseudo-Taylor polynomial expansion. We then use the resulting node embedding distributions to analytically derive probabilistic adversarial robustness certificates for node classification tasks against L2-bounded perturbations of node features. Second, we model node features as multivariate random variables and introduce *Feature Convolution Distance* (FCD_p), a pseudometric based on the Wasserstein distance. FCD_p corresponds to the discriminative power of MPNNs at the node level. We show that MPNNs are globally Lipschitz continuous functions with respect to the pseudometric FCD_p. Using the covering number of the resulting pseudometric space, which is a subset of the Wasserstein space, we derive generalization bounds for MPNNs with uncertainties in node features. Together, these two complementary approaches—moment propagation for adversarial robustness and FCD_p on the subset of the Wasserstein space for generalization—establish a unified theoretical framework that comprehensively addresses MPNN reliability under node feature uncertainty.

Proceedings of the 29[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

1 INTRODUCTION

Message Passing Neural Networks (MPNNs) are popular methods for applying machine learning to graph-structured data and have strong performance on graph-, edge-, and node-level tasks (Chami et al., 2022; Hu et al., 2020; Hamilton, 2020). MPNNs take as input a graph structure and node (and/or edge) feature vectors. They employ a recursive neighborhood aggregation scheme that produces node (and/or edge) embedding vectors (Gilmer et al., 2017; Scarselli et al., 2008). Most existing MPNN formulations assume that the node (and/or edge) features are deterministic vectors. In practice, however, features are often uncertain due to inherent noise or finite measurement precision (Peel et al., 2022; Ju et al., 2025). Thus, approaches that quantify uncertainty in MPNNs are useful (Wang et al., 2024; Zhang et al., 2024). This paper addresses uncertainty quantification in MPNNs for node-level tasks.

First, we assume Gaussian node-feature distributions. We adopt the Gaussian assumption for three main reasons. First, it is the maximum entropy distribution given the first two moments, making it the least biased assumption when higher-order moments are unknown. Second, it ensures computational tractability, allowing for efficient propagation. Third, it aligns with standard practices in the uncertainty quantification literature (Sullivan, 2015; Soize, 2017; Wright et al., 2024; Zhang and Ching, 2025; Wang et al., 2024). We propagate these moments exactly through linear message-passing operations, and approximately through nonlinearities using Taylor expansion and pseudo-Taylor polynomial expansion (PTPE) (Zhang and Ching, 2025) of nonlinear functions. In this way, we obtain moment approximations of the node embedding distributions. Using these embedding distributions, we establish probabilistic adversarial robustness certificates for feature perturbations in L_2-norm. Our approach provides formal robustness guarantees that are lacking in current (mostly) heuristic defenses against perturbation attacks on node features.

Second, we define a Wasserstein distance-based pseudometric, the *Feature Convolution Distance* FCD_p, in the input space of nodes to analyze the generalization guarantees of node-level MPNNs with stochastic node features. Here, we use the Simple Graph Convolution (SGC) model (Wu et al., 2019), which is an MPNN without nonlinearities that balances efficiency, interpretability, and performance. We show that FCD_p is a pseudometric that satisfies the non-negativity, symmetry, and triangle inequality axioms. FCD_p incorporates the structural updates enacted by SGC and achieves discriminative power equivalent to that of the SGC and SGC with output nonlinearity. By establishing Lipschitz continuity for the SGC architecture with respect to FCD_p, we lay the foundation for the rigorous analysis of generalization guarantees. Notably, our framework remains consistent when handling deterministic features, thereby unifying stochastic and deterministic scenarios within a single framework.

Third, we introduce a novel framework for analyzing covering number–based generalization error bounds (Xu and Mannor, 2012) in the setting of multiclass node-classification task in the inductive learning scenario with noisy node features for the SGC with output nonlinearity. We derive an explicit upper-bound on the Wasserstein distance between the probability measures of the population and empirical losses. A central ingredient of our analysis is establishing compactness of the space of Gaussian measures induced by node-level structural updates under the Wasserstein distance and characterizing the associated covering number. Building on this foundation, we leverage the Lipschitz continuity of SGC with respect to the proposed pseudometric FCD_p and Lipschitzness of the loss function to formalize algorithmic robustness of the learning algorithm, which serves as a key step in deriving the generalization error bound in this setting.

Finally, we validate our theoretical predictions in extensive numerical experiments on synthetic and real-world graph data.

Contributions.

- We consider graphs with stochastic node features and derive analytic representations for the first two moments of distributions associated with features after propagating the random vectors through MPNNs. We derive probabilistic adversarial robustness guarantees at the node level by obtaining per-node certified radii against L_2-bounded feature-perturbation attacks.
- We introduce FCD_p, a Wasserstein distance-based pseudometric to compare nodes. We demonstrate that FCD_p provides global Lipschitz continuity between the input and embedding spaces. Moreover, FCD_p exhibits the same discriminative power as the distance between probability distributions of random variables transformed by SGC with nonlinearity. We derive generalization bounds based on the covering number for the node-classification task in the inductive learning scenario.
- We provide experimental results that evaluate the performance of moment propagation, demonstrate the resulting certified adversarial robustness radii, and provide estimates of the generalization bounds for the task of node classification.

The remainder of the paper is organized as follows. Section 2 provides background information. Section 3 quantifies uncertainty in node features by propagating moments of distributions and obtains adversarial robustness guarantees against L_2-norm perturbation. Section 4 derives generalization bounds based on the covering number. Section 5 presents our experimental results.[1] Section 6 reviews related work. Section 7 summarizes the paper.

2 BACKGROUND

Graph Convolutional Network and Its Variants. Graph Convolutional Network (GCN) (Kipf and Welling, 2017) is a widely used MPNN architecture. A GCN with L layers on a graph with n nodes is defined by the following recursion for $l \in \{1, \cdots, L\}$:

$$X^{(l+1)} = \sigma(SX^{(l)}W^{(l)}), \tag{1}$$

where $S = (I_n + D)^{-1/2}(I_n + A)(I_n + D)^{-1/2} \in \mathbb{R}^{n\times n}$ is the structural update matrix defined in terms of the adjacency matrix $A \in \mathbb{R}^{n\times n}$, the diagonal matrix of node degrees $D = \mathrm{diag}\{d_1, \cdots, d_n\} \in \mathbb{R}^{n\times n}$, and the n-dimensional identity matrix I_n. We denote the lth layer's weight matrix as $W^{(l)} \in \mathbb{R}^{f_l \times f_{l+1}}$. The matrix $X^{(l)} \in \mathbb{R}^{n\times f_l}$ contains the f_l-dimensional node features at layer l, and the features at $l = 0$ are the input features. The point-wise nonlinearity σ can be any Lipschitz continuous function.

In the absence of nonlinearities, all weight matrices can be combined into a single matrix $W \in \mathbb{R}^{f_0 \times f_L}$. The resulting architecture is called the Simple Graph Convolution (SGC) (Wu et al., 2019)

$$X^{(L)} = \Theta(X^{(0)}) = S^L X^{(0)} W. \tag{2}$$

In addition to the linear SGC architecture $\Theta(\cdot)$, we also consider a version $\hat{\Theta}(\cdot) = \sigma(\Theta(\cdot))$ with a single output nonlinearity σ, which is useful when SGC is applied to node classification.

[1] Code: `https://doi.org/10.5281/zenodo.19211677`

Wasserstein Distance. Optimal transport defines the p-Wasserstein distance W_p for $p \in [1, \infty)$ between two probability measures $\mathbb{P}_{\boldsymbol{\xi}}$ and $\mathbb{P}_{\boldsymbol{\xi}'}$ on $\mathbb{R}^f$ as

$$W_p(\mathbb{P}_{\boldsymbol{\xi}}, \mathbb{P}_{\boldsymbol{\xi}'}) = \inf_{\gamma \in \Gamma(\mathbb{P}_{\boldsymbol{\xi}}, \mathbb{P}_{\boldsymbol{\xi}'})} \left(\iint_{\mathbb{R}^f \times \mathbb{R}^f} \|\boldsymbol{\xi} - \boldsymbol{\xi}'\|^p \, d\gamma(\boldsymbol{\xi}, \boldsymbol{\xi}') \right)^{1/p}, \quad (3)$$

where γ is a coupling, i.e., a distribution on $\mathbb{R}^f \times \mathbb{R}^f$ that has $\mathbb{P}_{\boldsymbol{\xi}}$ and $\mathbb{P}_{\boldsymbol{\xi}'}$ as its marginals. In the deterministic case where $\mathbb{P}_{\boldsymbol{\xi}}$, $\mathbb{P}_{\boldsymbol{\xi}'}$ are Dirac measures, the Wasserstein distance reduces to the L_p-norm.

ϵ-cover and Covering Number. An ϵ-cover of a subset T of a pseudometric space $(\mathcal{X}, d)$ is a set $\hat{T} \subset \mathcal{X}$ such that for each $t \in T$, there is a $\hat{t} \in \hat{T}$ such that $d(t, \hat{t}) \leq \epsilon$. The ϵ-covering number of T is defined as follows (Xu and Mannor, 2012)

$$N(\epsilon; T; d) = \min\{|\hat{T}| : \hat{T} \text{ is an } \epsilon\text{-cover of } T\}. \quad (4)$$

Learning Model and Generalization Error. Given the set of training samples $\mathcal{Z}$, the learning model picks a hypothesis h from a set of hypotheses $\mathcal{H}$. Let $\mathcal{A}_S$ denote the hypothesis learned given the training set S. In the inductive learning scenario, a learning algorithm $\mathcal{A}$ receives as input a training sample $S = \{(x_i, y_i)\}_{i=1}^N \subset \mathcal{Z}$ drawn i.i.d. from an unknown distribution κ on $\mathcal{Z}$, and outputs a hypothesis $h = \mathcal{A}_S \in \mathcal{H}$. Given a bounded loss function $\ell : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$, the expected risk and the empirical risk are defined as follows (Vasileiou et al., 2025b)

$$\ell_{\text{exp}}(\mathcal{A}_S) := \mathbb{E}_{(x,y)\sim\kappa}[\ell(\mathcal{A}_S, x, y)] \quad (5)$$

and

$$\ell_{\text{emp}}(\mathcal{A}_S) := \frac{1}{N} \sum_{i=1}^{N} \ell(\mathcal{A}_S, x_i, y_i) \quad (6)$$

respectively. The generalization error is then defined as the absolute difference between these two functions, $|\ell_{\text{exp}}(\mathcal{A}_S) - \ell_{\text{emp}}(\mathcal{A}_S)|$.

3 ROBUSTNESS CERTIFICATION

We begin by establishing a principled framework for tracking the propagation of uncertainty of node features through the layers of an MPNN. We assume that node features are modeled as Gaussian random variables, and we show how their means and covariances evolve under the structural updates, linear transformations, and nonlinear activations of the network. Therefore, we precisely characterize how uncertainty is transformed at each layer. This analysis forms the statistical foundation for our adversarial robustness guarantees against L_2-norm feature perturbations, and naturally extends to higher-order moments and more general distributions.

3.1 Moments Propagation

Here, we provide expressions for the propagation of the first and second moments of the Gaussian input feature distribution through a GCN. We consider node embeddings at each layer l (including input features at $l = 0$) as random variables $\boldsymbol{\xi}_i^{(l)} \in \mathbb{R}^{f_l}$ for each node $v_i \in V$. For ease of notation, we define the vector $\vec{\boldsymbol{\xi}}^{(l)} = [\boldsymbol{\xi}_1^{(l)}, \ldots, \boldsymbol{\xi}_n^{(l)}]^\intercal \in \mathbb{R}^{nf_l}$ by concatenation. We emphasize that this setup allows for linear correlation between features at the same node, as well as between features at different nodes. We use $A \otimes B$ to denote the Kronecker product of two matrices A and B.

The embeddings $\vec{\boldsymbol{\xi}}^{(l),s}$ after the **structural update** are

$$\vec{\boldsymbol{\xi}}^{(l),s} = (S \otimes I_{f_{l-1}}) \vec{\boldsymbol{\xi}}^{(l-1)}, \quad (7)$$

where $I_{f_{l-1}}$ is the f_{l-1}-dimensional identity matrix. The random variable $\vec{\boldsymbol{\xi}}^{(l),s}$ has the following mean $\mu_{\vec{\boldsymbol{\xi}}^{(l),s}}$ and variance $\Sigma_{\vec{\boldsymbol{\xi}}^{(l),s}}$

$$\begin{aligned} \mu_{\vec{\boldsymbol{\xi}}^{(l),s}} &= (S \otimes I_{f_{l-1}}) \mu_{\vec{\boldsymbol{\xi}}^{(l-1)}}, \\ \Sigma_{\vec{\boldsymbol{\xi}}^{(l),s}} &= (S \otimes I_{f_{l-1}}) \Sigma_{\vec{\boldsymbol{\xi}}^{(l-1)}} (S \otimes I_{f_{l-1}})^\intercal. \end{aligned} \quad (8)$$

where $\mu_{\vec{\boldsymbol{\xi}}^{(l-1)}}$ and $\Sigma_{\vec{\boldsymbol{\xi}}^{(l-1)}}$ are mean and variance of the embeddings $\vec{\boldsymbol{\xi}}^{(l-1)}$ at the previous layer. The **weight update** yields embeddings $\vec{\boldsymbol{\xi}}^{(l),w}$ defined by

$$\vec{\boldsymbol{\xi}}^{(l),w} = (I_n \otimes W^{(l)}) \vec{\boldsymbol{\xi}}^{(l),s}, \quad (9)$$

where I_n is the n-dimensional identity matrix. The mean and the variance of this random variable are given by,

$$\begin{aligned} \mu_{\vec{\boldsymbol{\xi}}^{(l),w}} &= (I_n \otimes W^{(l)}) \mu_{\vec{\boldsymbol{\xi}}^{(l),s}}, \\ \Sigma_{\vec{\boldsymbol{\xi}}^{(l),w}} &= (I_n \otimes W^{(l)}) \Sigma_{\vec{\boldsymbol{\xi}}^{(l),s}} (I_n \otimes W^{(l)})^\intercal. \end{aligned} \quad (10)$$

Finally, the embeddings $\vec{\boldsymbol{\xi}}^{(l),e}$ after the **nonlinear update** are

$$\vec{\boldsymbol{\xi}}^{(l),e} = \sigma\left(\vec{\boldsymbol{\xi}}^{(l),w}\right), \quad (11)$$

where σ is applied elementwise.

Moment propagation through nonlinear functions cannot, in general, be solved analytically. We replace the function σ by its Taylor expansion around the mean $\mu_{\vec{\boldsymbol{\xi}}^{(l),w}}$ and apply an appropriate moment closure. In Appendix B.1, we show that

$$\mu_{\vec{\boldsymbol{\xi}}^{(l),e}} \approx \vec{\sigma} + \frac{1}{2} \vec{s}_2, \quad (12)$$

where, $\vec{\sigma} = \sigma(\mu_{\vec{\boldsymbol{\xi}}^{(l),w}})$, $\vec{s}_2 = \sigma''(\mu_{\vec{\boldsymbol{\xi}}^{(l),w}}) \circ \text{diag}(\Sigma_{\vec{\boldsymbol{\xi}}^{(l),w}})$, and $\circ$ is the Hadamard product, σ'' is the second

derivative of σ (as a function on real numbers) applied elementwise and $\mathrm{diag}(\Sigma_{\vec{\boldsymbol{\xi}}^{(l),w}})$ is the vector of diagonal entries of $\Sigma_{\vec{\boldsymbol{\xi}}^{(l),w}}$. And, using Isserli's Theorem to perform moment closure, the covariance, $\Sigma_{\vec{\boldsymbol{\xi}}^{(l),e}}$ is

$$\begin{aligned}\Sigma_{\vec{\boldsymbol{\xi}}^{(l),e}} \approx \vec{\sigma}\vec{\sigma}^{\mathsf{T}} + \vec{s_1}^{\mathsf{T}} \circ \Sigma_{\vec{\boldsymbol{\xi}}^{(l),w}} \circ \vec{s}_1 \\ + \frac{1}{2}\left(\vec{\sigma}\vec{s}_2^{\mathsf{T}} + \vec{s}_2\vec{\sigma}^{\mathsf{T}}\right) \\ + \frac{1}{4}\left(\vec{s}_2\vec{s}_2^{\mathsf{T}} + 2\,\vec{s}_1^{\mathsf{T}}\left(\Sigma_{\vec{\boldsymbol{\xi}}^{(l),w}} \circ \Sigma_{\vec{\boldsymbol{\xi}}^{(l),w}}\right)\vec{s}_1\right) \\ - \mu_{\vec{\boldsymbol{\xi}}^{(l),e}}\mu_{\vec{\boldsymbol{\xi}}^{(l),e}}^{\mathsf{T}},\end{aligned} \tag{13}$$

where $\vec{s}_1 = \sigma'(\mu_{\vec{\boldsymbol{\xi}}^{(l),w}})$ is the first derivative of σ.

Alternatively, one can employ other methods such as PTPE (Zhang and Ching, 2025), or apply a multivariate Taylor expansion to the entire MPNN. We present these alternatives in Appendix B.

3.2 Probabilistic Adversarial Robustness Certification

Robustness to feature perturbations is a desideratum for graph machine learning architectures. Knowing the moments of the distribution of the random variable representing the logits allows us to certify the robustness of MPNNs for node-level classification tasks against L_2-norm feature perturbations. We denote the random variable of node logits $\boldsymbol{\xi}_i^z \in \mathbb{R}^{f_L}$ and its mean as $\mu_{\boldsymbol{\xi}_i^z}$ and covariance as $\Sigma_{\boldsymbol{\xi}_i^z}$, and use an additional indices y or $y^\star$ to refer to entries pertaining to any class or the true class, respectively.

Theorem 1. *An MPNN for node classification tasks is robust against L_2-norm feature perturbation $||\Delta|| = \epsilon$ with probability at least $1-\delta$ if*

$$\epsilon < \min_{y \neq y^\star} \frac{\hat{\mu}_{\boldsymbol{\xi}_{iy}^z} - \sqrt{\hat{\Sigma}_{\boldsymbol{\xi}_{iy}^z}}\sqrt{\frac{1-\delta_y}{\delta_y}}}{\sqrt{2}\bar{C}}, \tag{14}$$

where $\delta = \sum_y \delta_y$. δ_y is the probability of misclassifying class y. $\hat{\mu}_{\boldsymbol{\xi}_{iy}^z} = \mu_{\boldsymbol{\xi}_{i,y^\star}^z} - \mu_{\boldsymbol{\xi}_{i,y}^z}$ is the mean and $\hat{\Sigma}_{\boldsymbol{\xi}_{iy}^z} = \Sigma_{\boldsymbol{\xi}_i^z, y^\star y^\star} + \Sigma_{\boldsymbol{\xi}_i^z, yy} - 2\,\Sigma_{\boldsymbol{\xi}_i^z, y^\star y}$ is the variance of the random margin between the logit associated with true label class $y^\star$ and logit element associated with any other class label $y \neq y^\star$. $\bar{C}$ is the Lipschitz constant of the GCN.

The proof of the Theorem 1 is in Appendix C. The theorem establishes a certified adversarial robustness radius: a probabilistic guarantee of how far one can perturb node features without altering the predicted label. This connects the moment propagation analysis to an actionable robustness certificate. It demonstrates that incorporating uncertainty into the theoretical analysis does not just describe noise propagation, but also leads to practical, quantitative safety guarantees for node-level classification tasks.

4 GENERALIZATION GUARANTEES

To study the generalization behavior of MPNNs under node feature uncertainty, we require a principled notion of distance between nodes whose features are random variables with associated probability measures (rather than deterministic vectors). We propose a novel pseudometric FCD_p, which compares nodes via the Wasserstein distance between the probability measures of their MPNN structural updates. This pseudometric not only induces Lipschitz continuity for MPNNs, but also enables the derivation of generalization bounds via covering numbers, thereby connecting the geometry of the underlying space of nodes probability measures with formal learning guarantees.

4.1 Novel Pseudometric

We introduce the new distance FCD_p between nodes $v_i, v_j \in V$ of the graph $G = (V, E)$ with associated random variables $\boldsymbol{\xi}_i^{(0)}$, $\boldsymbol{\xi}_j^{(0)}$ for the $\hat{\Theta}(\cdot)$ architecture as

$$\begin{aligned}\mathrm{FCD}_p := W_p(\mathbb{P}_{\boldsymbol{\xi}_i^s}, \mathbb{P}_{\boldsymbol{\xi}_j^s}) = \\ \inf_{\gamma \in \Gamma(\mathbb{P}_{\boldsymbol{\xi}_i^s}, \mathbb{P}_{\boldsymbol{\xi}_j^s})} \left(\iint_{\mathbb{R}^{f_0} \times \mathbb{R}^{f_0}} \|\boldsymbol{\xi}_i^s - \boldsymbol{\xi}_j^s\|^p d\gamma(\boldsymbol{\xi}_i^s, \boldsymbol{\xi}_j^s)\right)^{1/p},\end{aligned} \tag{15}$$

where $\boldsymbol{\xi}_i^s := [S^L \boldsymbol{\xi}^{(0)}]_i$ is the random variable associated with node v_i after the structural update step of SGC. If the input node features follow multivariate normal distributions, then the node-wise random variables after structural update are also normally distributed and for $p = 2$ the FCD_p has an analytical closed form. For the other values of p or other probability distributions, numerical methods such as the Sinkhorn algorithm (Sinkhorn and Knopp, 1967; Cuturi, 2013) can be used. When the probability measures are Dirac measures, the distance is equal to L_p norm between structural updates of node features.

Proposition 1. FCD_p *from a node v_i to itself is zero:* $\mathrm{FCD}_p(v_i, v_i) = 0$.

This is true because the probability distributions associated with the same random variable $\boldsymbol{\xi}_i$ are identical.

Proposition 2. FCD_p *from v_i to v_j is equal to the distance from v_j to v_i:* $\mathrm{FCD}_p(v_i, v_j) = \mathrm{FCD}_p(v_j, v_i)$.

This property automatically follows from the property of the Wasserstein distance.

Proposition 3. $\mathrm{FCD}_p(v_i, v_j) \leq \mathrm{FCD}_p(v_i, v_k) + \mathrm{FCD}_p(v_k, v_j)$.

This property follows directly from the properties of the Wasserstein distance (Villani et al., 2008). However, FCD_p does not guarantee positivity between distinct points: $\mathrm{FCD}_p(v_i, v_j)$ can be zero even when v_i and v_j are different. Therefore, it is a pseudometric. Next, we show that FCD_p ensures Lipschitz continuity between input and embedding spaces and provides the same discriminative power as $\hat{\Theta}$. Theoretical guarantees for the default Θ architecture are given in Appendices D and E.

Theorem 2 (Discriminative Power)**.** FCD_p *has the same discriminative power as* $\hat{\Theta}$*:*

$$W_p(\hat{\Theta}(v_i), \hat{\Theta}(v_j)) > 0 \Rightarrow \mathrm{FCD}_p(v_i, v_j) > 0. \tag{16}$$

The proof of Theorem 2 is in Appendix D.

The Lipschitzness property allows one to reason about how close nodes are in the embedding space and can be used in reasoning about generalization properties.

Theorem 3 (Lipschitz Continuity)**.** *SGC* $\hat{\Theta}$ *is a globally Lipschitz function w.r.t.* FCD_p *:*

$$W_p(\hat{\Theta}(v_i), \hat{\Theta}(v_j)) \leq C_L \cdot \mathrm{FCD}_p(v_i, v_j). \tag{17}$$

The proof of Theorem 3 can be found in the Appendix E. Additionally, we present the discriminative power and Lipschitz continuity for Dirac measures in Appendix D and Appendix E.

4.2 Generalization Bounds

We propose a framework to reason about the generalization abilities of MPNNs in the presence of uncertainty in node features and derive covering number-based generalization bounds, similar to (Xu and Mannor, 2012; Vasileiou et al., 2025b). We will consider the multi-class node classification task in an inductive learning scenario with $\ell(\cdot)$ being the cross entropy loss and $\mathcal{Z} = \mathcal{X} \times \mathcal{Y} = (V, \mathrm{FCD}_p) \times \{0, \ldots, \mathcal{C}\}$ where $\mathcal{C}$ is the number of classes.

Settings. We say that the probability measures $\mathbb{P}_{\boldsymbol{\xi}_i}$ associated with nodes comes from a class of Gaussian measures $\mathcal{N}(\mu, \Sigma)$ with uniform exponential moments:

$$\int_{\mathbb{R}^{f_0}} e^{a\|x\|} d\mathcal{N}(\mu, \Sigma)(x) \leq M, \tag{18}$$

where $a > 0$ is some constant and $M < \infty$. This guarantees that the entire collection of measures does not have heavy tails, which ensures the tightness of the class of Gaussian random measures associated with nodes. We additionally assume that $\mathbb{P}_{\boldsymbol{\xi}_i}$ are identically distributed with common law $\nu \in \mathcal{P}(\mathcal{P}_p(\mathcal{X}))$. $\mathcal{P}_p(\mathcal{X})$ endowed with W_p is the Wasserstein space of order p, $\mathcal{X} \subseteq \mathbb{R}^{f_0}$, where f_0 is the number of features of the node.

Definition 1. *Let G be an undirected graph with n vertices. We say that $(\mathbb{P}_{\boldsymbol{\xi}_1}, \ldots, \mathbb{P}_{\boldsymbol{\xi}_n})$ are G-dependent, if for all disjoint subsets $I, J \subset [n]$, whenever there is no edge between I and J in G, the collections $\{\mathbb{P}_{\boldsymbol{\xi}_i} : i \in I\}$ and $\{\mathbb{P}_{\boldsymbol{\xi}_j} : j \in J\}$ are independent.*

Definition 2. *The chromatic number $\chi(G)$ is the minimum number of colors required to color the nodes of G such that no two adjacent nodes share the same color. For the fully connected graph with n nodes, the chromatic number is $\chi(G) = n$.*

Properties of the Sample Space $\mathcal{Z}$. Let $(V,\ \mathrm{FCD}_p)$ be the space of nodes V metrized with FCD_p.

Theorem 4. (V, FCD_p) *is compact, and its covering number is*

$$N(\epsilon; V; \mathrm{FCD}_p) \leq \exp\{\hat{C}\epsilon^{-f_0}(\log[1/\epsilon])^{f_0}\}, \tag{19}$$

where constant $\hat{C}$ depends on f_0 and p. The proof of the theorem as well as the formula for $\hat{C}$ can be found in Appendix F. Now let us consider the sample space $\mathcal{Z}$ and metrize it with the following distance

$$\begin{aligned} d_{\mathcal{Z}}((x_i, y_i), (x_j, y_j)) := \\ \max\{\mathrm{FCD}_p(v_i, v_j),\ \delta_{\{y_i, y_j\}}\}, \end{aligned} \tag{20}$$

where $\delta_{\{y_i, y_j\}} = 0$ if $y_i = y_j$. Otherwise, $\delta_{\{y_i, y_j\}} = \infty$.

Theorem 5. $(\mathcal{Z}, d_{\mathcal{Z}})$ *is compact, and its covering number is*

$$N(\epsilon; \mathcal{Z}; d_{\mathcal{Z}}) \leq \mathcal{C} \exp\{\hat{C}\epsilon^{-f_0}(\log[1/\epsilon])^{f_0}\}. \tag{21}$$

This follows from the fact that the sample space $\mathcal{Z}$ is the product space of $(V,\ \mathrm{FCD}_p)$ and $\mathcal{Y} = \{0, \ldots, \mathcal{C}\}$, and distance definition $d_{\mathcal{Z}}$. Therefore at any radius ϵ, the entire space of samples $\mathcal{Z}$ can be represented by a finite partition of size $N(\epsilon; \mathcal{Z}; d_{\mathcal{Z}})$. More details about partitioning of $\mathcal{Z}$ can be found in Appendix G.

Generalization Bounds. Let us now first define the notion of the algorithmic robustness of the learning algorithm $\mathcal{A}$ on $\mathcal{Z}$, which is a crucial property for the derivation of the generalization error bound.

Proposition 4. *Let $\mathcal{A}$ be a learning algorithm on sample space $\mathcal{Z}$ for hypothesis class $\mathcal{H}$, and let $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}^+$ be a loss function. Let $d_{\mathcal{Z}}$ be a pseudometric on $\mathcal{Z}$. If $\ell(\cdot)$ is C-Lipschitz with respect to*

$d_{\mathcal{Z}}$, then for all samples S and $(\mathbb{P}_{\boldsymbol{\xi}_i}, y_i)$, $(\mathbb{P}_{\boldsymbol{\xi}_j}, y_j) \in \mathcal{Z}$,

$$W_p(\ell(\mathcal{A}_S, (\mathbb{P}_{\boldsymbol{\xi}_i}, y_i)), \ell(\mathcal{A}_S, (\mathbb{P}_{\boldsymbol{\xi}_j}, y_j))) \leq C d_{\mathcal{Z}}((\mathbb{P}_{\boldsymbol{\xi}_i}, y_i), (\mathbb{P}_{\boldsymbol{\xi}_j}, y_j)); \tag{22}$$

thus, $\mathcal{A}$ is $(N(\epsilon/2; \mathcal{Z}; d_{\mathcal{Z}}), C\epsilon)$-uniformly robust for all $\epsilon > 0$.

Proposition 4 implies that if the loss is C-Lipschitz with respect to $d_{\mathcal{Z}}$, then inside each ϵ-ball of the covering of $(\mathcal{Z}, d_{\mathcal{Z}})$, the loss can fluctuate by at most $C\epsilon$. Equivalently, every partition element of diameter ϵ acts as a robust cell. Once the algorithm's behavior is fixed at one representative of the cell, its behavior across the whole cell is controlled within $C\epsilon$. The uniform robustness follows from the fact that only $K = N(\epsilon/2; \mathcal{Z}; d_{\mathcal{Z}})$ such robust cells are needed to cover the entire sample space. This partition-based control allows us to transform Lipschitz continuity into generalization guarantees.

Theorem 6. *Let $\mathcal{A}$ be a learning algorithm on $\mathcal{Z}$. Then for any $\delta > 0$, with probability at least $1 - \delta$, and for a sample $S = \{(\mathbb{P}_{\boldsymbol{\xi}_i}, y_i)\}_{i=1}^N$ with the dependency graph $G[S]$ and a C_ℓ-Lipschitz loss function ℓ, the following holds*

$$W_p(\widehat{\mathbb{P}}_{\ell,S}, \mathbb{P}_{\ell,S}) \leq 2C\epsilon + M\sqrt{\frac{\chi(G[S])}{|S|}(2(K_\epsilon + 1)\log 2 + 2\log(\frac{1}{\delta}))}, \tag{23}$$

where

$$\widehat{\mathbb{P}}_{\ell,S} := \frac{1}{N}\sum_{i=1}^{N} \ell(\mathcal{A}_S, (\mathbb{P}_{\boldsymbol{\xi}_i}, y_i)) \tag{24}$$

is the empirical loss probability measure, and

$$\mathbb{P}_{\ell,S} := \mathbb{E}_{(\mathbb{P}_{\boldsymbol{\xi}}, y)\sim\nu}[\ell(\mathcal{A}_S, (\mathbb{P}_{\boldsymbol{\xi}}, y))] \tag{25}$$

is the population loss probability measure, M is the diameter of the space of probability measures associated with losses

$$M := \sup_{(\mathbb{P}_{\boldsymbol{\xi}_i}, y_i), (\mathbb{P}_{\boldsymbol{\xi}_j}, y_j) \in \mathcal{Z}} W_p(\ell(\mathcal{A}_S, (\mathbb{P}_{\boldsymbol{\xi}_i}, y_i)), \ell(\mathcal{A}_S, (\mathbb{P}_{\boldsymbol{\xi}_j}, y_j))) < \infty, \tag{26}$$

where $K_\epsilon = N(\epsilon; \mathcal{Z}; d_{\mathcal{Z}})$ is the covering number of the sample space $\mathcal{Z}$, $C = C_L C_\ell$, C_L is FCD_p *Lipschitz constant, $\chi(G)$ is the chromatic number of dependency graph G.*

The proof of Theorem 6 is in Appendix I. This theorem illustrates how close the loss distribution estimated from the training data is to the true population loss distribution even when node features are noisy and potentially dependent through the graph structure. Intuitively, the result shows that the generalization error is controlled whenever the space of uncertain node–label pairs can be covered with relatively few robust partitions, the dependency structure of the graph is not too dense and the loss-space diameter is not big. This bridges geometric properties of the sample space with probabilistic guarantees for MPNN generalization under node feature uncertainty.

Corollary 1. *$W_1\left(\mathbb{P}_{\ell,S}, \widehat{\mathbb{P}}_{\ell,S}\right)$ is an upper bound on the risk gap $|\ell_{exp}(A_S) - \ell_{emp}(A_S)|$ as follows*

$$|\ell_{exp}(\mathcal{A}_S) - \ell_{emp}(\mathcal{A}_S)| \leq W_1(\mathbb{P}_{\ell,S}, \widehat{\mathbb{P}}_{\ell,S}), \tag{27}$$

where $\ell_{exp}(\mathcal{A}_S)$ and $\ell_{emp}(A_S)$ are the expected values of the loss drawn from the measures $\mathbb{P}_{\ell,S}$ and $\widehat{\mathbb{P}}_{\ell,S}$, respectively.

The proof for Corollary 1 is in Appendix I.

5 EXPERIMENTAL EVALUATION

In this section, we compare the performance of moment propagation, illustrate certified robustness radii against L_2-norm feature perturbations, analyze Lipschitz continuity with respect to FCD_p, and assess the tightness of our generalization bounds.

We consider three types of synthetic graphs with different levels of dependency between node features, independent nodes, independent features (`inif`), independent nodes, dependent features (`indf`), dependent nodes, dependent features (`dndf`). Additionally, we use seven real-world graphs (`Cornell`, `Wisconsin`, `Texas`, `Cora`, `Citeseer`, `Chameleon`, `Squirrel`) with independent Gaussian noise added to their features. We describe the datasets, respective training procedures, and additional results in Appendices J.1 and J.2, respectively.

Uncertainty Quantification. For moment propagation, we rely on the following methods: First-order (`T1`) and second-order (`T2-Tr`) Taylor approximation of the entire architecture, where we use truncation of contributions from moments higher than the second; Layer-wise Taylor approximation of nonlinearities to first-order (`1d-T1`), to second-order with Gaussian closure (`1d-T2-GC`) and to second-order with truncation of contributions from moments higher than the second (`1d-T2-Tr`); Pseudo-Taylor Polynomial Expansion (`PTPE`) (Zhang and Ching, 2025).

We compare the moments obtained from these methods with the moments obtained from propagating $n_{\text{MC}} = 10,000$ samples, using the 2-Wasserstein distance (W_2) between Gaussians with these moments and the L_2-norm of the mean and the Frobenius norm (FN) of covariance differences. For `Cora`, `Citeseer`, `Squirrel`, and `Chameleon`, we use only the diagonal entries of the covariance for W_2 to reduce computation time. The efficiency can be improved by making structural assumptions about the covariance (e.g., low rank, sparsity) and by using suitable data structures. Tables with results are in Appendices J.1 and J.2.

For synthetic data, `PTPE` performs best across all nonlinearities except `sigmoid`. Linearization of the architecture (`T1`) performs similar to layer-wise linearization (`1d-T1`) but at lower memory cost. For second-order Taylor approximations, layer-wise approximations (`1d-T2-Tr`, `1d-T2-GC`) appear to be preferable over Taylor approximation of the entire architecture (`T2-Tr`), which is similar in performance but comes with a large memory footprint. Gaussian closure (`1d-T2-GC`) performs similarly to truncation (`1d-T2-Tr`).

For real-world data, we performed experiments on 2- and 3-layer GCNs with 4- and 8-dimensional embeddings at each GCN layer. To summarize our findings we use the following notation. `Cornell`(2, 4, `ReLU`) denotes a 2-layer GCN with 4-dimensional vectors at each GCN layer and ReLU activation on the `Cornell` dataset. The key takeaways are as follows:

- Layer-wise Taylor expansions of the same order show similar approximation errors to their multivariate counterparts (e.g., W_2 is similar for `T1` and `1d-T1`). On `Wisconsin`(2, 4, `ReLU`), W_2 equals 2.2 for both `T1` and `1d-T1`, while W_2 is 3.6, 3.6, and 3.1 for `T2-Tr`, `1d-T2-Tr`, and `1d-T2-GC`, respectively. Gaussian closure (`1d-T2-GC`) does not consistently outperform truncation (`1d-T2-Tr`).

- `PTPE` shows lower error than both multivariate (`T1`, `T2-Tr`) and layer-wise (`1d-T1`, `1d-T2-GC`, `1d-T2-Tr`) Taylor expansions, with two exceptions. First, for `ReLU` `PTPE` can fail to produce moments, likely due to the accumulation of numerical errors in repeated layer-wise updates. Second, `PTPE` performs worse on sigmoid nonlinearities; for example, `Cornell`(2, 4, `sigmoid`) shows two orders of magnitude higher W_2 than `Cornell`(2, 4, `GELU`).

- On small graphs (`Texas`, `Cornell`, `Wisconsin`), sampling is slower than moment propagation methods. On `Cornell`(3, 8, `GELU`), sampling takes 22s, while moment propagation (`T1`, `T2-Tr`, `1d-T1`, `1d-T2-Tr`, `1d-T2-GC`, `PTPE`) takes between 1.5s and 3s. On large graphs (`Cora`, `Citeseer`, `Chameleon`, `Squirrel`), multivariate Taylor expansions are faster than layer-wise ones, and sampling has intermediate runtimes. On `Cora`(3, 8, `GELU`), `T1` takes 40s, and `T2-Tr` takes 227s. `1d-T1`, `1d-T2-GC`, and `1d-T2-Tr` take between 1825s and 1848s. Sampling takes 758s.

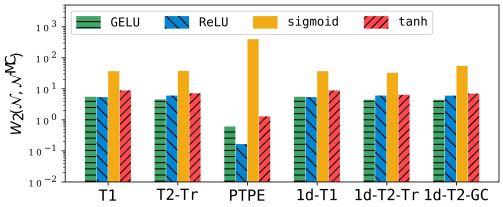

Figure 1: `Cora` Dataset: Moment Comparison on 3-layer GCNs with 8-dimensional Embeddings. The nonlinear functions are `GELU`, `ReLU`, `sigmoid`, and `tanh`. The lower the Wasserstein distance W_2 on the y-axis, the better. **Takeaway:** Layer-wise Taylor expansions of the same order show similar approximation errors to their multivariate counterparts. `PTPE` shows the lowest error except for `sigmoid`.

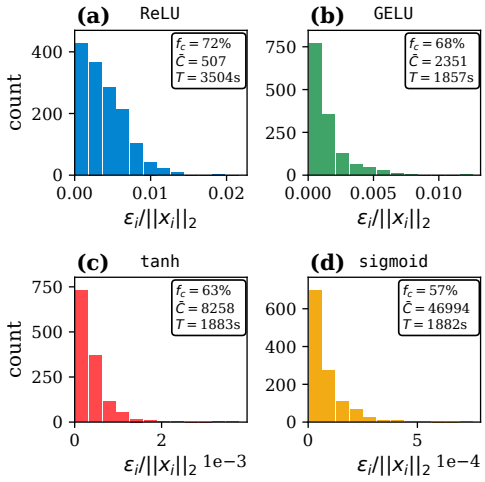

Figure 2: `Cora` Dataset: Histogram of Radii ϵ_i Relative to the Feature Vector Size $||x_i||_2$ on 3-layer GCNs with 8-dimensional Embeddings. $\bar{C}$ is the Lipschitz constant, T is time to compute the radii, and f_c is fraction of certified points. Moments are estimated using `1d-T2-GC`. **Takeaway:** Robustness radii are the smallest for `tanh` and `sigmoid` nonlinearities, robustness radii are similar for `ReLU` and `GELU`. The number of certified nodes is the highest for `ReLU`.

Probabilistic Adversarial Robustness Guarantees. We compute the robustness radii ϵ_i for each node, according to Theorem 1 based on moments propagated with `1d-T2-GC`. We use the upper bound on L_2-norm Lipschitz constant $\bar{C}$ of the architecture. We calculate it as the product of the Lipschitz constants of all parts of the architecture. For nonlinearities, we use the Lipschitz constants reported in (Mao et al., 2023), for linear operators we use their largest singular value. We restrict our analysis to correctly classified nodes. We set the probability of misclassification to $\delta = 0.05$. We split δ to equal amounts among classes $\mathcal{C}$ (i.e., $\delta_y = \delta/\mathcal{C}$). We also compared the resulting radii to ones obtained via (Cohen et al., 2019). Tables with additional details are in Appendices J.1 and J.2.

Negative values of ϵ_i appear for some nodes in both synthetic and real-world graphs. This implies that for all architectures there exist nodes for which robustness cannot be certified. Robustness radii tend to be larger for `GELU` and `ReLU` nonlinearities than for `tanh` and `sigmoid` for both synthetic and real-world graphs.

The key takeaways for robustness radii on real-world data are as follows:

- Theorem 1 certifies between 8% of nodes for `Chameleon`(2, 8, `GELU`) sampling and 100% for `Wisconsin`(3, 4, `sigmoid`) sampling, while Cohen et al.'s method certifies between 93% for `Wisconsin`(3, 4, `GELU`) and 100% for `Cornell`(2, 4, `GELU`). On average across graphs and models, Theorem 1 certifies between 49% (moments via `PTPE`) and 56% (moments via `T1`) of nodes, whereas Cohen et al.'s method certifies 99% of nodes.
- For small graphs (`Cornell`, `Wisconsin`, `Texas`), applying Theorem 1 with any moment estimation method is faster (between 2 and 27 seconds) than Cohen et al.'s method (∼574 seconds on average). For larger graphs (`Cora`, `Citeseer`, `Chameleon`, `Squirrel`), Theorem 1 with moments via `T1` is faster (between 31 and 534 seconds) than Cohen et al.'s method (between 574 and 745 seconds), but Theorem 1 with moments from sampling, `PTPE`, and `1d-T2-GC` are slower (typically between 354 and 4749 seconds).

Figure 2 shows robustness radii for `Cora`. The percentage of certified nodes and the radii are the highest for GCN with `ReLU` nonlinearity, which means that GCN trained with `ReLU` nonlinearity is more robust than GCN trained with `GELU`, `tanh`, `sigmoid` nonlinearities.

Lipschitz Continuity w.r.t. FCD_p. To empirically verify Lipschitz continuity, we calculated the Pearson and Spearman correlation between FCD_p (with $p = 1$) in the input space and the W_1 distance in the output space. For both synthetic and real-world data SGC is Lipschitz continuous in FCD_p meaning that small changes in the input lead to small changes in the output. Figure 3 illustrates the results for real-world data.

Table 1: Generalization Bounds on Synthetic Data.

dataset	L	emp. LHS	RHS
`inif`	3	5.11	141.29
`indf`	3	5.12	152.30
`dndf`	3	5.10	133.47

Table 2: Generalization Bounds on Real-world Data.

dataset	L	empirical LHS	RHS
`Cornell`	3	19.38	254.44
`Wisconsin`	3	23.92	364.57
`Texas`	3	12.39	170.46
`Cora`	3	6.01	432.61
`Citeseer`	3	5.843	401.74
`Chameleon`	3	4.83	416.93
`Squirrel`	3	1.98	136.03

Generalization Bounds. We test tightness according to Equation (23) in the form LHS < RHS, where LHS represents the generalization error and RHS denotes its upper bound. We estimate LHS using the 1-Wasserstein distance, computed with 100 samples, between the node-wise training and test loss distributions, since the entire data distribution is unknown.

We use the optimal transport on 100 samples per node to estimate the W_1 distance between node-wise losses, and determine its maximum value M. We set $\epsilon = 0.99$ and $\delta = 0.05$. We assume χ is the number of nodes in the training set.

Tables 1 and 2, respectively, summarize our results on synthetic and real-world data. For synthetic data, the generalization gap is the smallest when there are dependencies between nodes and features, but the effect is moderate. We do not observe a strong effect with the number of layers. These results highlight that examining assumptions on the data generating process is important when considering generalization. For both synthetic and real-world data, larger Lipschitz constant and M lead to looser bounds. Moreover, larger graphs (`Cora`, `Citeseer`, `Squirrel`, `Chameleon`) result in SGCs with larger Lipschitz constants, contributing to looser generalization bounds.

6 RELATED WORK

Uncertainty Quantification in MPNNs. Numerous methods exist to quantify epistemic and aleatoric uncertainty (Wang et al., 2024; Hüllermeier

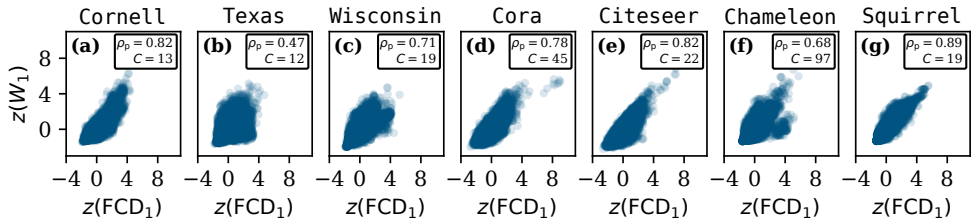

Figure 3: z-score for the FCD_1 and the 1-Wasserstein Distance for SGC with 3 Layers. The z-score is value minus mean, divided by standard deviation. For each dataset, the number of hidden dimensions is set to the number of classes in that dataset. **Takeaway:** SGC is Lipschitz continuous in FCD_p meaning that small changes in the input lead to small changes in the output.

and Waegeman, 2021; Gawlikowski et al., 2023). Tracing uncertainty through nonlinear models has a long tradition (Sullivan, 2015; Soize, 2017). Polynomial chaos expansion (PCE) is a well-established method, originally introduced for Gaussian random variables (Wiener, 1938; Ghanem and Spanos, 1991) and later extended to arbitrary distributions as generalized polynomial chaos, gPC (Xiu and Karniadakis, 2002, 2003) and to distributions accessible only through their moments as arbitrary polynomial chaos, aPC (Oladyshkin and Nowak, 2012; Navarro et al., 2014; Paulson et al., 2017). Recently, these methods have been applied in machine learning (Du, 2025).

Robustness Certification. Cohen et al. (2019) establish that a classifier smoothed with Gaussian noise yields provable L_2 radius guarantees. Certification is conducted via sampling-based estimation. Kumar et al. (2020) propose a method to generate certified radii for the prediction confidence of a smoothed classifier. Pautov et al. (2022) introduce the CC-Cert framework, which leverages concentration inequalities to certify the robustness of neural networks under input perturbations. Zügner and Günnemann (2019) introduce one of the first certification schemes for GCNs to defend against node-feature modifications under L_0-bounded budgets. GNNCert (Yang et al., 2024) provides deterministic certification for graph classification against both structure and feature perturbations by guaranteeing label invariance when the numbers of modified edges and node features are bounded. Our work differs from others by being probabilistic and not relying on sampling techniques.

Pseudometrics and Generalization. Rauchwerger et al. (2024) extend iterated degree measures to graphon-signals, and show compactness of the resulting space, establishing Lipschitz continuity and universal approximation for MPNNs. Levie (2024) proves a one-sided Lipschitz inequality, bounding feature distances by the graphon-signal cut distance, though with slow generalization rates. Chuang and Jegelka (2022) propose the Tree Mover's Distance (TMD) for graphs with features, relating it to generalization under distribution shifts, but lacking universal approximation. Chen et al. (2022, 2023) show that MPNNs separate points and are Lipschitz over WL distances, with universal approximation only on compact subspaces since the full space is not compact. Vasileiou et al. (2025a) leverage graph similarity theory to assess the influence of graph structure, aggregation, and loss functions on MPNN generalization abilities. The work closest to ours is Vasileiou et al. (2025b). They introduce a unified framework for analyzing the generalization properties of MPNNs in inductive and transductive node and link prediction tasks while relaxing nodes i.i.d. assumptions; but they do not consider the case when there is uncertainty in node features.

7 CONCLUSION

We present a unified theoretical framework for assessing the reliability of MPNNs under uncertainty in node features. By propagating moments through both the linear updates and nonlinear activations of MPNNs, we enable certified probabilistic robustness guarantees against L_2-bounded perturbations of node features. Additionally, our Wasserstein-based pseudometric (FCD_p) integrates structural updates with node-feature uncertainty, matches the discriminative power of SGC, and ensures global Lipschitz continuity for SGC. Using compactness and covering number arguments in the resulting metric space, we derive generalization bounds that extend beyond the case of deterministic node features. Our theoretical results are supported by empirical evaluations.

Acknowledgements and Author Contributions

A.C. was supported by Northeastern University's Network Science Institute. M.L. and T.E.R. were supported by the Inaugural Joseph E. Aoun Endowment. N.S. was supported by NSF Grant No. CCF-2311160.

A.C., M.L., and N.S. designed FCD_p, developed the methodology of SGC's discriminative power and Lipschitz continuity, and established the theory for moment propagation. They also formulated the proof for Lipschitz continuity. A.C. formulated the theorems and proofs for the SGC's discriminative power, generalization bounds and robustness radii. All authors verified the proofs. M.L. formulated the derivations for moment propagation. In addition, M.L. implemented the moment propagation and baseline methods and conducted computational experiments. A.C. advised on parameter settings and case selection. All authors contributed to the interpretation of the results and writing of the paper.

References

Vladimir I Bogachev. *Measure theory*. Springer, 2007.

Ines Chami, Sami Abu-El-Haija, Bryan Perozzi, Christopher Ré, and Kevin Murphy. Machine learning on graphs: A model and comprehensive taxonomy. *Journal of Machine Learning Research*, 23(89): 1–64, 2022.

Samantha Chen, Sunhyuk Lim, Facundo Memoli, Zhengchao Wan, and Yusu Wang. Weisfeiler-Lehman meets Gromov-Wasserstein. In *ICML*, pages 3371–3416, 2022.

Samantha Chen, Sunhyuk Lim, Facundo Memoli, Zhengchao Wan, and Yusu Wang. The Weisfeiler-Lehman distance: Reinterpretation and connection with gnns. In *Proceedings of 2nd Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML)*, pages 404–425, 2023.

Ching-Yao Chuang and Stefanie Jegelka. Tree mover's distance: Bridging graph metrics and stability of graph neural networks. *Advances in Neural Information Processing Systems*, 35:2944–2957, 2022.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, pages 1310–1320, 2019.

Marco Cuturi. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

DeepMind, Igor Babuschkin, Kate Baumli, Alison Bell, Surya Bhupatiraju, Jake Bruce, Peter Buchlovsky, David Budden, Trevor Cai, Aidan Clark, Ivo Danihelka, Antoine Dedieu, Claudio Fantacci, Jonathan Godwin, Chris Jones, Ross Hemsley, Tom Hennigan, Matteo Hessel, Shaobo Hou, Steven Kapturowski, Thomas Keck, Iurii Kemaev, Michael King, Markus Kunesch, Lena Martens, Hamza Merzic, Vladimir Mikulik, Tamara Norman, George Papamakarios, John Quan, Roman Ring, Francisco Ruiz, Alvaro Sanchez, Laurent Sartran, Rosalia Schneider, Eren Sezener, Stephen Spencer, Srivatsan Srinivasan, Miloš Stanojević, Wojciech Stokowiec, Luyu Wang, Guangyao Zhou, and Fabio Viola. The DeepMind JAX Ecosystem, 2020. URL `http://github.com/google-deepmind`.

Xiaoping Du. Uncertainty quantification for machine learning-based prediction: A polynomial chaos expansion approach for joint model and input uncertainty propagation. *arXiv preprint arXiv:2507.14782*, 2025.

Matthias Fey, Jinu Sunil, Akihiro Nitta, Rishi Puri, Manan Shah, Blaž Stojanovič, Ramona Bendias, Alexandria Barghi, Vid Kocijan, Zecheng Zhang, Xinwei He, Jan Eric Lenssen, and Jure Leskovec. PyG 2.0: Scalable Learning on Real World Graphs. In *Temporal Graph Learning Workshop @ KDD 2025*, August 2025.

Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, Muhammad Shahzad, Wen Yang, Richard Bamler, and Xiao Xiang Zhu. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56(1):1513–1589, 2023.

Roger G. Ghanem and Pol D. Spanos. Stochastic Finite Element Method: Response Statistics. In Roger G. Ghanem and Pol D. Spanos, editors, *Stochastic Finite Elements: A Spectral Approach*, pages 101–119. Springer, 1991.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *ICML*, pages 1263–1272, 2017.

William L. Hamilton. *Graph Representation Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool, 2020.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in Neural Information Processing Systems*, 33:22118–22133, 2020.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An

introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021.

Svante Janson. Large deviations for sums of partly dependent random variables. *Random Structures & Algorithms*, 24(3):234–248, 2004.

Wei Ju, Siyu Yi, Yifan Wang, Zhiping Xiao, Zhengyang Mao, Hourun Li, Yiyang Gu, Yifang Qin, Nan Yin, Senzhang Wang, et al. A survey of graph neural networks in real world: Imbalance, noise, privacy and ood challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Patrick Kidger and Cristian Garcia. Equinox: neural networks in JAX via callable PyTrees and filtered transformations. *Differentiable Programming workshop at Neural Information Processing Systems 2021*, 2021.

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

Aounon Kumar, Alexander Levine, Soheil Feizi, and Tom Goldstein. Certifying confidence via randomized smoothing. *Advances in Neural Information Processing Systems*, 33:5165–5177, 2020.

Ron Levie. A graphon-signal analysis of graph neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications. In *ICML*, pages 23803–23828, 2023.

Maria Navarro, Jeroen Witteveen, and Joke Blom. Polynomial chaos expansion for general multivariate distributions with correlated variables. *arXiv preprint arXiv:1406.5483*, 2014.

S. Oladyshkin and W. Nowak. Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion. *Reliability Engineering & System Safety*, 106:179–190, 2012.

Victor M Panaretos and Yoav Zemel. The wasserstein space. In *An Invitation to Statistics in Wasserstein Space*, pages 37–57. Springer, 2020.

Joel A Paulson, Edward A Buehler, and Ali Mesbah. Arbitrary polynomial chaos for uncertainty propagation of correlated random variables in dynamic systems. *IFAC-PapersOnLine*, 50(1):3548–3553, 2017.

Mikhail Pautov, Nurislam Tursynbek, Marina Munkhoeva, Nikita Muravev, Aleksandr Petiushko, and Ivan Oseledets. Cc-cert: A probabilistic approach to certify general robustness of neural networks. In *AAAI*, pages 7975–7983, 2022.

Philippe Pierre Pebay. Formulas for robust, one-pass parallel computation of covariances and arbitrary-order statistical moments. Technical report, Sandia National Laboratories, 2008.

Leto Peel, Tiago P Peixoto, and Manlio De Domenico. Statistical inference links data and theory in network science. *Nature Communications*, 13(1):6794, 2022.

Levi Rauchwerger, Stefanie Jegelka, and Ron Levie. Generalization, expressivity, and universality of graph neural networks on attributed graphs. *arXiv preprint arXiv:2411.05464*, 2024.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

Christian Soize. *Uncertainty Quantification: An Accelerated Course with Advanced Applications in Computational Engineering*. Springer, 2017.

T.J. Sullivan. *Introduction to Uncertainty Quantification*. Springer, 2015.

John Thickstun. Kantorovich-rubinstein duality. `https://courses.cs.washington.edu/courses/cse599i/20au/resources/L12\_duality.pdf`, 2019.

Antonis Vasileiou, Ben Finkelshtein, Floris Geerts, Ron Levie, and Christopher Morris. Covered forest: Fine-grained generalization analysis of graph neural networks. In *International Conference on Machine Learning*, pages 60984–61034. PMLR, 2025a.

Antonis Vasileiou, Timo Stoll, and Christopher Morris. Understanding generalization in node and link prediction. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2025b.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2008.

Fangxin Wang, Yuqing Liu, Kay Liu, Yibo Wang, Sourav Medya, and Philip S. Yu. Uncertainty in Graph Neural Networks: A Survey. *Transactions on Machine Learning Research*, 2024.

Norbert Wiener. The Homogeneous Chaos. *American Journal of Mathematics*, 60(4):897–936, 1938.

Oren Wright, Yorie Nakahira, and José MF Moura. An analytic solution to covariance propagation in neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 4087–4095. PMLR, 2024.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *ICML*, pages 6861–6871, 2019.

Dongbin Xiu and George Em Karniadakis. The wiener–askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002.

Dongbin Xiu and George Em Karniadakis. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics*, 187(1):137–167, 2003.

Huan Xu and Shie Mannor. Robustness and generalization. *Machine learning*, 86:391–423, 2012.

Han Yang, Binghui Wang, Jinyuan Jia, et al. GNNCert: Deterministic certification of graph neural networks against adversarial perturbations. In *ICLR*, 2024.

He Zhang, Bang Wu, Xingliang Yuan, Shirui Pan, Hanghang Tong, and Jian Pei. Trustworthy graph neural networks: Aspects, methods, and trends. *Proceedings of the IEEE*, 112(2):97–139, February 2024.

Songhan Zhang and ShiNung Ching. A stochastic polynomial expansion for uncertainty propagation through networks. *Transactions on Machine Learning Research*, 2025.

Daniel Zügner and Stephan Günnemann. Certifiable robustness and robust training for graph convolutional networks. In *KDD*, pages 246–256, 2019.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [No]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   `https://doi.org/10.5281/zenodo.19211677`
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [No]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) Code is available at `https://doi.org/10.5281/zenodo.19211677`.
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A NOTATION AND ADDITIONAL BACKGROUND INFORMATION

Table 3 lists the notation used throughout the paper. The rest of the section introduces graphs, functions of random variables, and coupling of random variables.

A.1 Graphs

A **graph** $G = (V, E)$ is defined by a set of nodes V and the edges $E \subset V \times V$ between them. We denote the number of nodes $n = |V|$ and the number of edges $m = |E|$. A graph can be represented by its **adjacency matrix** $A \in \mathbb{R}^{n \times n}$, where $A_{ij} = 1$ if node v_i is connected to node v_j (i.e, $(v_i, v_j) \in E$), and $A_{ij} = 0$ otherwise. A node $v_i \in V$ often possesses additional properties encoded in its **feature vector** $\boldsymbol{x}_i \in \mathbb{R}^{1 \times f}$, where f is called the feature dimension. It is convenient to gather the feature vectors of all nodes in a feature matrix $X \in \mathbb{R}^{n \times f}$. For example, to model a social media platform as a graph, one represents each user as a node, their characteristics (e.g., age and topics of interests) as node features, and which users are friends on the platform with edges. The degree $d_i = \sum_{j=1}^{n} A_{ij}$ of a node is the number of other nodes that it is connected to. We denote the diagonal matrix of degrees as $D = \mathrm{diag}(d_1, \ldots, d_n)$.

A.2 Functions of Random Variables

The **pushforward** is a way to calculate how uncertainty in the inputs of a function translates into uncertainty of its outputs. Let $\boldsymbol{\xi}$ be a multivariate random variable taking values in $\mathbb{R}^{f_1}$, e.g., the input to a neural network, and $\Psi : \mathbb{R}^{f_1} \to \mathbb{R}^{f_2}$ a Borel function, e.g., a neural network. Applying Ψ to $\boldsymbol{\xi}$ defines another random variable $\boldsymbol{\eta} = \Psi(\boldsymbol{\xi})$ taking values in $\mathbb{R}^{f_2}$ and describing in our examples the output of a neural network. When $\boldsymbol{\xi}$ has distribution $\mathbb{P}_{\boldsymbol{\xi}}$, the distribution $\mathbb{P}_{\boldsymbol{\eta}}$ of $\boldsymbol{\eta}$ is

$$\mathbb{P}_{\boldsymbol{\eta}}(\boldsymbol{\eta} \in B) = \mathbb{P}_{\boldsymbol{\eta}}(\Psi(\boldsymbol{\xi}) \in B) = \mathbb{P}_{\boldsymbol{\xi}}(\boldsymbol{\xi} \in \Psi^{-1}(B)) \quad \forall B \in \mathcal{B}(\mathbb{R}^{f_2}), \tag{28}$$

where $\mathcal{B}(\mathbb{R}^{f_2})$ are the Borel sets of $\mathbb{R}^{f_2}$ and $\Psi^{-1}(B) \in \mathcal{B}(\mathbb{R}^{f_1})$ is the preimage of $B \in \mathcal{B}(\mathbb{R}^{f_2})$ under Ψ, a Borel set of $\mathbb{R}^{f_1}$.

A.3 Couplings of Random Variables

Given two random variables $\boldsymbol{\xi} \in \mathbb{R}^{f}$ and $\boldsymbol{\xi'} \in \mathbb{R}^{f'}$ with distributions $\mathbb{P}_{\boldsymbol{\xi}}$ and $\mathbb{P}_{\boldsymbol{\xi'}}$ respectively, a **coupling** is a distribution γ on the product space $\mathbb{R}^{f} \times \mathbb{R}^{f'}$ that has $\mathbb{P}_{\boldsymbol{\xi}}$ and $\mathbb{P}_{\boldsymbol{\xi'}}$ as its marginals, i.e.,

$$\gamma(B \times \mathbb{R}^{f'}) = \mathbb{P}_{\boldsymbol{\xi}}(B) \text{ and } \gamma(\mathbb{R}^{f} \times B') = \mathbb{P}_{\boldsymbol{\xi'}}(B'), \tag{29}$$

for all $B \in \mathcal{B}(\mathbb{R}^{f})$ and $B' \in \mathcal{B}(\mathbb{R}^{f'})$. We denote the space of all couplings as $\Gamma(\mathbb{P}_{\boldsymbol{\xi}}, \mathbb{P}_{\boldsymbol{\xi'}})$. Sometimes the word coupling is also used for a random variable that has distribution γ.

As for any other random variable, one can compute the **pushforward of a coupling**. Here, we are interested in the special case, where $\boldsymbol{\xi}, \boldsymbol{\xi'} \in \mathbb{R}^{f}$ be random variables with distributions $\mathbb{P}_{\boldsymbol{\xi}}, \mathbb{P}_{\boldsymbol{\xi'}}$ with a coupling γ between them, and the same Borel function $\Psi : \mathbb{R}^{f} \to \mathbb{R}^{f'}$ is applied to each of them individually, in this case the pushforward of the coupling is

$$(\Psi \times \Psi)_{\sharp\gamma}(B \times B') = \gamma(\Psi^{-1}(B) \times \Psi^{-1}(B')), \tag{30}$$

for all Borel sets $B, B' \in \mathcal{B}(\mathbb{R}^{f})$.

Table 3: Notation used throughout the paper.

SYMBOL	EXPLANATION
$G = (V, E)$	graph with nodes set V and edge set E.
$n = \|V\|$	number of nodes in the graph
$m = \|E\|$	number of edges in the graph
$v_i \in V$	node of the graph G
$I_n \in \mathbb{R}^{n\times n}$	n-dimensional identity matrix
$\mathbf{1}_n \in \mathbb{R}^n$	the n-dimensional vector of ones
$A \in \mathbb{R}^{n\times n}$	adjacency matrix of G
$D = \mathrm{diag}(d_1, \ldots, d_n)$	diagonal matrix of node degrees
$S = (I + D)^{-1/2}(I + A)(I + D)^{-1/2}$	structural update matrix
$s_{ij} = [S]_{ij}$	entries of S
$l \in \{0, \ldots, L\}$	MPNN layer index
f_l	node feature dimension in layer l
$\boldsymbol{x}_i^{(l)} \in \mathbb{R}^{1\times f_l}$	deterministic node features associated with node v_i in layer l
$X^{(l)} \in \mathbb{R}^{n\times f_l}$	matrix of deterministic node features in layer l
$W^{(l)} \in \mathbb{R}^{f_l \times f_{l+1}}$	weight matrix of layer l
$\sigma, \sigma', \sigma'' : \mathbb{R} \to \mathbb{R}$	nonlinear, Lipschitz continuous function, and its first two derivatives
$\Theta : \mathbb{R}^{f_0} \to \mathbb{R}^{f_L}$	Simple Graph Convolution network without output nonlinearity
$\hat{\Theta} : \mathbb{R}^{f_0} \to \mathbb{R}^{f_L}$	Simple Graph Convolution network with output nonlinearity
$\boldsymbol{\xi} \in \mathbb{R}^f$	f-dimensional random variable.
$\boldsymbol{\xi}_i^{(l)} \in \mathbb{R}^{f_l}$	random variable representing random features of node v_i at layer l
$\vec{\boldsymbol{\xi}}^{(l)} \in \mathbb{R}^{nf_l}$	random vector from concatenation of the features of all nodes
$\mathbb{P}_{\boldsymbol{\xi}}$	distribution (or law) of the random variable $\boldsymbol{\xi}$
$p_{\boldsymbol{\xi}}$	probability density function of the random variable $\boldsymbol{\xi}$
$W_p(\mathbb{P}_{\boldsymbol{\xi}}, \mathbb{P}_{\boldsymbol{\xi}'})$	p-Wasserstein distance between $\mathbb{P}_{\boldsymbol{\xi}}$ and $\mathbb{P}_{\boldsymbol{\xi}'}$
$\Gamma(\mathbb{P}_\xi, \mathbb{P}_{\xi'})$	space of couplings of the random variables ξ and ξ'
$\gamma \in \Gamma(\mathbb{P}_{\boldsymbol{\xi}}, \mathbb{P}_{\boldsymbol{\xi}'})$	coupling between the random variables $\boldsymbol{\xi}$ and $\boldsymbol{\xi'}$
$\mu_{\boldsymbol{\xi}}$	first moment of the random variable $\boldsymbol{\xi}$
$\Sigma_{\boldsymbol{\xi}}$	second moment of the random variable $\boldsymbol{\xi}$
FCD_p	Feature Convolution distance, our W_p-based pseudometric
L_p	space of functions with integrable p norm
$\mathcal{N}(\mu, \Sigma)$	normal distribution with mean $\mu \in \mathbb{R}^f$ and covariance $\Sigma \in \mathbb{R}^{f\times f}$
$\bar{C}$	Lipschitz constant of a GCN w.r.t. the L_2 norm
C_L	Lipschitz constant of SGC w.r.t. the FCD_pdistance and L_p distance
C_ℓ	Lipschitz constant of loss function
$\mathcal{B}(\mathbb{R}^f)$	the Borel sets of $\mathbb{R}^f$
$\Psi_\sharp\mathbb{P}$	pushforward of the distribution $\mathbb{P}$ under the function Ψ
$\vec{\sigma} = \sigma(\mu_{\vec{\boldsymbol{\xi}}}) \in \mathbb{R}^f$	elementwise application of σ to $\mu_{\vec{\boldsymbol{\xi}}}$
$\vec{s}_1 = \sigma'(\mu_{\vec{\boldsymbol{\xi}}}) \in \mathbb{R}^f$	elementwise application of the derivative of σ to $\mu_{\vec{\boldsymbol{\xi}}}$
$\vec{s}_2 = \sigma''(\mu_{\vec{\boldsymbol{\xi}}}) \in \mathbb{R}^f$	elementwise application of the second derivative of σ to $\mu_{\vec{\boldsymbol{\xi}}}$
$\nabla h(\vec{\boldsymbol{x}}) \in \mathbb{R}^{nf_L \times nf_0}$	Jacobian matrix of $h : \mathbb{R}^{nf_0} \to \mathbb{R}^{nf_L}$
$\nabla^2 h(\vec{\boldsymbol{x}}) \in \mathbb{R}^{nf_L \times nf_0 \times nf_0}$	Hessian tensor of $h : \mathbb{R}^{nf_0} \to \mathbb{R}^{nf_L}$
$\circ$	elementwise or Hadamard product of vectors and/or matrices
$\otimes$	Kronecker product
$d_{\mathcal{Z}}$	pseudometric in the sample space
$N(\epsilon; V; \mathrm{FCD}_p)$	covering number of the space of nodes
$N(\epsilon; \mathcal{Z}; d_{\mathcal{Z}})$	covering number of the space of samples
G	dependency graph
χ	chromatic number of the dependency graph
M	loss-space diameter

B MOMENT PROPAGATION

B.1 Elementwise Taylor

Here, we show the Taylor expansion for a nonlinear function that acts elementwise, i.e., $\sigma_i(\boldsymbol{x}) = \sigma_i(x_i)$. We are interested in the moments of the random variable defined by the pushforward, $\vec{\boldsymbol{\zeta}} = \sigma(\vec{\boldsymbol{\xi}})$ where $\vec{\boldsymbol{\xi}} \in \mathbb{R}^{nf_l}$ with first two moments $\mu_{\vec{\boldsymbol{\xi}}}$ and $\Sigma_{\vec{\boldsymbol{\xi}}}$.

Consider the second order Taylor expansion of the ith component of this function around μ_{ξ_i}

$$\zeta_i = \sigma(\xi_i) \approx \sigma\big(\mu_{\xi_i}\big) + \ \sigma'\big(\mu_{\xi_i}\big)(\xi_i - \mu_{\xi_i}) + \frac{1}{2}\sigma''\big(\mu_{\xi_i}\big)(\xi_i - \mu_{\xi_i})^2, \tag{31}$$

where σ', σ'' are the first and second derivative of σ as a function $\sigma : \mathbb{R} \to \mathbb{R}$.

Then, the expected value of ζ_i is

$$\mu_{\zeta_i} = \mathbb{E}[\zeta_i] \approx \sigma\big(\mu_{\xi_i}\big) + \frac{1}{2}\sigma''\big(\mu_{\xi_i}\big)\Sigma_{\xi,ii}, \tag{32}$$

or in vectorized notation

$$\mu_{\vec{\boldsymbol{\zeta}}} = \sigma(\mu_{\vec{\boldsymbol{\xi}}}) + \frac{1}{2}\vec{s}_2, \tag{33}$$

where we introduced $\vec{\sigma} = \sigma(\mu_{\vec{\boldsymbol{\xi}}})$, and $\vec{s}_2 = \sigma''(\mu_{\vec{\boldsymbol{\xi}}}) \circ \mathrm{diag}(\Sigma_{\vec{\boldsymbol{\xi}}})$ with $\circ$ denoting the Hadamard product. Similarly, we can obtain the variance,

$$\begin{aligned}\Sigma_{\zeta,ij} = \mathbb{E}[(\zeta_i - \mu_{\zeta,i})(\zeta_j - \mu_{\zeta,j})] &\approx \sigma(\mu_{\xi,i})\sigma(\mu_{\xi,j}) \\ &+ \frac{1}{2}(\sigma(\mu_{\xi,i})\sigma''(\mu_{\xi,j})\Sigma_{\xi,jj} + \sigma(\mu_{\xi,j})\sigma''(\mu_{\xi,i})\Sigma_{\xi,ii}) \\ &+ \frac{1}{4}\sigma''(\mu_{\xi,i})\sigma''(\mu_{\xi,j})\mathbb{E}[(\xi_i - \mu_{\xi,i})^2(\xi_i - \mu_{\xi,j})^2] \\ &+ \sigma'(\mu_{\xi,i})\sigma'(\mu_{\xi,j})\Sigma_{\xi,ij} - \mu_{\xi,i}\mu_{\xi,j}\end{aligned} \tag{34}$$

The second-to-last line depends on the fourth moment of the input distribution. We can either view this term as a fourth-order contribution and neglect it. This yields the second-order truncated approximation (`1d-T2`). A better approximation can be obtained by using Isserli's theorem to perform Gaussian moment closure. This theorem expresses the fourth moment approximately in terms of second moments, and is exact for Gaussian distributions. In our case, we have

$$\mathbb{E}[(\xi_i - \mu_{\xi,i})^2(\xi_j - \mu_{\xi,j})^2] = \Sigma_{\xi,ii}\Sigma_{\xi,jj} + 2\Sigma^2_{\xi,ij}, \tag{35}$$

and we can obtain the variance with Gaussian closure (`1d-T2-GC`) as,

$$\begin{aligned}\Sigma_{\zeta,ij} \approx \sigma(\mu_{\xi,i})\sigma(\mu_{\xi,j}) &+ \frac{1}{2}(\sigma(\mu_{\xi,i})\sigma''(\mu_{\xi,j})\Sigma_{\xi,jj} \\ &+ \sigma(\mu_{\xi,j})\sigma''(\mu_{\xi,i})\Sigma_{\xi,ii}) + \sigma'(\mu_{\xi,i})\sigma'(\mu_{\xi,j})\Sigma_{\xi,ij} \\ &+ \frac{1}{4}\sigma''(\mu_{\xi,i})\sigma''(\mu_{\xi,j})(\Sigma_{\xi,ii}\Sigma_{\xi,jj} + 2\Sigma^2_{\xi,ij}) - \mu_{\xi,i}\mu_{\xi,j}.\end{aligned} \tag{36}$$

Introducing $\vec{s}_1 = \sigma'(\mu_{\vec{\boldsymbol{\xi}}})$, we can write this in vectorized notation as

$$\begin{aligned}\Sigma_{\vec{\boldsymbol{\zeta}}} = \vec{\sigma}\vec{\sigma}^{\mathsf{T}} + s_1^{\mathsf{T}} \circ \Sigma_{\vec{\boldsymbol{\xi}}} \circ \vec{s}_1 &+ \frac{1}{2}(\vec{\sigma}\vec{s}_2^{\mathsf{T}} + \vec{s}_2\vec{\sigma}^{\mathsf{T}}) \\ &+ \frac{1}{4}\left(\vec{s}_2\vec{s}_2^{\mathsf{T}} + 2\vec{s}_1^{\mathsf{T}} \circ (\Sigma_{\vec{\boldsymbol{\xi}}} \circ \Sigma_{\vec{\boldsymbol{\xi}}}) \circ \vec{s}_1\right) - \mu_{\vec{\boldsymbol{\zeta}}}\mu_{\vec{\boldsymbol{\zeta}}}^{\mathsf{T}}.\end{aligned} \tag{37}$$

A less accurate approximation can be obtained by neglecting second-order terms (`1d-T1`), and gives the mean

$$\mu_{\vec{\zeta}} = \vec{\sigma}, \tag{38}$$

and variance

$$\Sigma_{\vec{\zeta}} = s_1^{\mathsf{T}} \circ \Sigma_{\vec{\xi}} \circ \vec{s}_1. \tag{39}$$

As the `ReLU` nonlinearity is not differentiable at the origin, we replace it with $\text{softplus}(x, \beta) = \frac{1}{\beta}\log(1 + e^{\beta x})$ for $\beta = 20$. Note that $\text{softplus}(x, \beta) \to \text{ReLU}(x)$ as $\beta \to \infty$.

B.2 Multi-variate Taylor Approximation

Instead of exploiting the elementwise application of the nonlinearity, we instead apply the Taylor approximation to the entire model. Considering the model as a function $h : \mathbb{R}^{nf_0} \times \mathbb{R}^{nf_L}$, acting on flattened node feature matrices $\vec{\boldsymbol{x}} \in \mathbb{R}^{nf_0}$ up to second order around $\mu_{\vec{\xi}}$ is

$$h(\vec{\boldsymbol{x}}) = h(\mu_{\vec{\xi}}) + \nabla h(\mu_{\vec{\xi}})(\vec{\boldsymbol{x}} - \mu_{\vec{\xi}}) + \frac{1}{2}(\vec{\boldsymbol{x}} - \mu_{\vec{\xi}})^{\mathsf{T}} \nabla^2 h(\mu_{\vec{\xi}})(\vec{\boldsymbol{x}} - \mu_{\vec{\xi}}) + \mathcal{O}((\vec{\boldsymbol{x}} - \mu_{\vec{\xi}})^3), \tag{40}$$

where $\nabla h(\mu_{\vec{\xi}}) \in \mathbb{R}^{nf_L \times nf_0}$ is the Jacobian, and $\nabla^2 h(\mu_{\vec{\xi}}) \in \mathbb{R}^{nf_L \times nf_0 \times nf_0}$ is the Hessian.

This leads to the first moment

$$\mu_h = \mathbb{E}[h] = h(\mu_{\vec{\xi}}) + \frac{1}{2}\text{Tr}[\nabla^2 h(\mu_{\vec{\xi}})\Sigma_{\vec{\xi}}], \tag{41}$$

where the matrix multiplication and trace act on the input dimensions. The variance is,

$$\begin{aligned}\Sigma_h = \mathbb{E}[(h - \mu_h)(h - \mu_h)^{\mathsf{T}}] &= h(\mu_{\vec{\xi}})h(\mu_{\vec{\xi}})^{\mathsf{T}} - \mu_h\mu_h^T + \nabla h(\mu_{\vec{\xi}})\Sigma_{\vec{\xi}}\nabla h(\mu_{\vec{\xi}}) \\ &+ \frac{1}{2}\left(\text{Tr}[\nabla^2 h(\mu_{\vec{\xi}})]h(\mu_{\vec{\xi}})^{\mathsf{T}} + h(\mu_{\vec{\xi}})\text{Tr}[\nabla^2 h(\mu_{\vec{\xi}})]^{\mathsf{T}}\right),\end{aligned} \tag{42}$$

where we dropped the contribution arising from the product of second-order terms, corresponding to fourth-moment contributions. While one could, in principle, perform Gaussian moment closure using Isserli's Theorem as in the element-wise case, the resulting terms cannot be evaluated without forming the full Hessian, which is impractical for modern neural networks and large graphs. We refer to this form as truncated, second-order Taylor (`T2-Tr`). For comparison, we also compute the linearization (`T1`), that neglects second order contributions, setting

$$\mu_h = \mathbb{E}[h] = h(\mu_{\vec{\xi}}) \tag{43}$$

and

$$\Sigma_h = \mathbb{E}[(h - \mu_h)(h - \mu_h)^{\mathsf{T}}] = \nabla h(\mu_{\vec{\xi}})\Sigma_{\vec{\xi}}\nabla h(\mu_{\vec{\xi}}). \tag{44}$$

Similar to the case of elementwise Taylor approximation, we replace $\texttt{ReLU}(x)$ with a differentiable approximation, $\texttt{softplus}(x, \beta)$ which converges to `ReLU` as $\beta \to \infty$. In our experiments, we set $\beta = 20$.

B.3 Pseudo-Taylor Polynomial Expansion

In addition to Taylor expansion, we also employ pseudo-Taylor polynomial expansion (`PTPE`) (Zhang and Ching, 2025) for propagating uncertainty through elementwise, nonlinear functions. We adapt this technique to MPNNs by using Eq. (7) and Eq. (9) to arrive at the first moment $\mu_{\vec{\xi}^{(w),l}}$ and variance, $\Sigma_{\vec{\xi}^{(w),l}}$, and pass them through the nonlinearity using

$$\mu_{\vec{\xi}^{(e),l}} = \mathbb{E}[\vec{\boldsymbol{\xi}}^{(e),l}] = A_0 \tag{45}$$

and

$$\Sigma_{\vec{\boldsymbol{\xi}}^{(e),l}} = \mathbb{E}[(\vec{\boldsymbol{\xi}}^{(e),l} - \mu_{\vec{\boldsymbol{\xi}}^{(e),l}})(\vec{\boldsymbol{\xi}}^{(e),l} - \mu_{\vec{\boldsymbol{\xi}}^{(e),l}})^{\mathsf{T}}] = \sum_{r=1}^{R} A_r \circ \Sigma_{\vec{\boldsymbol{\xi}}^{(w),l}} \circ A_r^{\mathsf{T}}, \tag{46}$$

where $A_r \in \mathbb{R}^{nf}$ are vectors depending on the nonlinearity, chosen according to Appendices A2 for `tanh`, A5 for `ReLU`, and A6 for `GELU` of (Zhang and Ching, 2025). We choose $R = 2$ as the order of approximation in our experiments.

C CERTIFIED ROBUSTNESS

The following theorem restates Theorem 1 from the main text, followed by its proof.

Theorem 7. *MPNN for node classification tasks is robust against L_2-norm feature perturbation $||\Delta|| = \epsilon$ with probability $1 - \delta$ if:*

$$\epsilon < \min_{y \neq y^\star} \frac{\hat{\mu}_{\boldsymbol{\xi}^z_{iy}} - \sqrt{\hat{\Sigma}_{\boldsymbol{\xi}^z_{iy}}}\sqrt{\frac{1-\delta_y}{\delta_y}}}{\sqrt{2}\bar{C}}, \tag{47}$$

where $\delta = \sum_y \delta_y$.

δ_y is the probability of misclassifying class y. $\hat{\mu}_{\boldsymbol{\xi}^z_{iy}} = \mu_{\boldsymbol{\xi}^z_{i,y^\star}} - \mu_{\boldsymbol{\xi}^z_{i,y}}$ is the mean and $\hat{\Sigma}_{\boldsymbol{\xi}^z_{iy}} = \Sigma_{\boldsymbol{\xi}^z_i, y^\star y^\star} + \Sigma_{\boldsymbol{\xi}^z_i, yy} - 2\,\Sigma_{\boldsymbol{\xi}^z_i, y^\star y}$ is the variance of the random margin between the element of the logit associated with true label class $y^\star$ and logit element associated with any other class label $y \neq y^\star$. $\bar{C}$ is the Lipschitz constant of the GCN with respect to the random margin norm.

Proof. Let us consider that every node v_i has a true classification label $y_i^\star \in \{1, \cdots, f_L\}$. For $\forall\ y \neq y^\star$, the margin in logits is:

$$M_y^z = \boldsymbol{\xi}^z_{iy^\star} - \boldsymbol{\xi}^z_{iy} = (\boldsymbol{e}_{y^\star} - \boldsymbol{e}_y)^{\mathsf{T}} \boldsymbol{\xi}_i^z = \boldsymbol{m}_y^{\mathsf{T}} \boldsymbol{\xi}_i^z, \tag{48}$$

where $\boldsymbol{e}_y \in \mathbb{R}^{f_L}$ is a standard basis vector, and $\boldsymbol{m}_y = \boldsymbol{e}_{y^\star} - \boldsymbol{e}_y$.

Let us now consider the case when the perturbation $||\Delta|| = \epsilon$ was added to the node feature vector $\boldsymbol{\xi}_i^{(0)}$, we define the corresponding logit as $\boldsymbol{\xi}_i^{z'}$. Let us consider the difference between the logit margins of the original and perturbed node features:

$$\begin{aligned} &\left| M_y^z - M_y^{z'} \right| = \left| \boldsymbol{m}_y^{\mathsf{T}} (\boldsymbol{\xi}_i^z - \boldsymbol{\xi}_i^{z'}) \right| \\ &\leq \left|\left| \boldsymbol{m}_y^{\mathsf{T}} \right|\right| \left|\left| \boldsymbol{\xi}_i^z - \boldsymbol{\xi}_i^{z'} \right|\right| = \sqrt{2} \left|\left| \boldsymbol{\xi}_i^z - \boldsymbol{\xi}_i^{z'} \right|\right| \\ &\leq \sqrt{2}\bar{C}\ \left|\left| \boldsymbol{\xi}_i^{(0)} - \boldsymbol{\xi}_i^{'(0)} \right|\right| = \sqrt{2}\bar{C}\,||\Delta|| = \sqrt{2}\bar{C}\epsilon \end{aligned} \tag{49}$$

or

$$M_y^z - \sqrt{2}\bar{C}\epsilon \leq M_y^{z'} \leq M_y^z + \sqrt{2}\bar{C}\epsilon, \tag{50}$$

where $\bar{C}$ is the GCN L_2 Lipschitz constant.

Let us now consider the probability of misclassification $\mathbb{P}(M_y^{z'} \leq 0)$. From the previous expression, it follows that $\mathbb{P}[M_y^{z'} \leq 0] \leq \mathbb{P}[M_y^z \leq \sqrt{2}\bar{C}\epsilon]$. By leveraging Cantelli inequality we can derive an upper bound on the misclassification probability:

$$\mathbb{P}[M_y^{z'} \leq 0] \leq \mathbb{P}[M_y^z \leq \sqrt{2}\bar{C}\epsilon] \leq \frac{\hat{\Sigma}_{\boldsymbol{\xi}^z_{iy}}}{\hat{\Sigma}_{\boldsymbol{\xi}^z_{iy}} + (\hat{\mu}_{\boldsymbol{\xi}^z_{iy}} - \sqrt{2}\bar{C}\epsilon)^2}, \tag{51}$$

where $\hat{\mu}_{\boldsymbol{\xi}^z_{iy}} = \mu_{\boldsymbol{\xi}^z_{i,y^\star}} - \mu_{\boldsymbol{\xi}^z_{i,y}}$, $\hat{\Sigma}_{\boldsymbol{\xi}^z_{iy}} = \Sigma_{\boldsymbol{\xi}^z_i, y^\star y^\star} + \Sigma_{\boldsymbol{\xi}^z_i, yy} - 2\,\Sigma_{\boldsymbol{\xi}^z_i, y^\star y}$.

Setting the upper bound on the probability of misclassification to be δ_y, and solving for ϵ, we get:

$$\epsilon = \frac{\hat{\mu}_{\boldsymbol{\xi}^z_{iy}} - \sqrt{\hat{\Sigma}_{\boldsymbol{\xi}^z_{iy}}}\sqrt{\frac{1-\delta_y}{\delta_y}}}{\sqrt{2}\bar{C}} \tag{52}$$

Each class has its own ϵ, so in order to be robust against misclassification to any label, we say that MPNN is robust for node classification task with probability at least $1-\delta$ if:

$$\epsilon < \min_{y \neq y^\star} \frac{\hat{\mu}_{\boldsymbol{\xi}^z_{iy}} - \sqrt{\hat{\Sigma}_{\boldsymbol{\xi}^z_{iy}}}\sqrt{\frac{1-\delta_y}{\delta_y}}}{\sqrt{2}\bar{C}}, \tag{53}$$

where $\delta = \sum_y \delta_y$. □

D DISCRIMINATIVE POWER

The following theorem restates Theorem 2 from the main text, followed by its proof.

Theorem 8. FCD_p *has the same discriminative power as* $\hat{\Theta}(\cdot)$*:*

$$W_p(\hat{\Theta}(v_i), \hat{\Theta}(v_j)) > 0 \Rightarrow \mathrm{FCD}_p(v_i, v_j) > 0 \tag{54}$$

Proof. Let $\Psi : \mathbb{R}^{f_0} \to \mathbb{R}^{f_L}$ be the application of weights and element-wise nonlinearity to the node feature matrix $X^s = S^L X^{(0)}$ after L-layer structural update of SGC, $\Psi(X^s) = \sigma(X^s W)$. In order to prove the discriminative power we need to show that

$$W_p(\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}^s_i}} \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}^s_j}}) > 0 \Rightarrow W_p(\mathbb{P}_{\boldsymbol{\xi}^s_i}, \mathbb{P}_{\boldsymbol{\xi}^s_j}) > 0, \tag{55}$$

where $\boldsymbol{\xi}^s_i = [S^L \Xi]_i$ and $\boldsymbol{\xi}^s_j = [S^L \Xi]_j$ are the node features after structural update in the L layer SGC architecture and defined in terms of the matrix valued random variable $\Xi \in \mathbb{R}^{n \times f_0}$ of input node features.

Eq. (55)⇒ If

$$W_p(\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}^s_i}} \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}^s_j}}) > 0, \tag{56}$$

then

$$\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}^s_i}} \neq \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}^s_j}}. \tag{57}$$

Therefore, $\exists B \in \mathcal{B}(\mathbb{R}^{f_L})$, such that

$$\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}^s_i}}(B) \neq \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}^s_j}}(B) \tag{58}$$

or by definition

$$\mathbb{P}_{\boldsymbol{\xi}^s_i}(\Psi^{-1}(B)) \neq \mathbb{P}_{\boldsymbol{\xi}^s_j}(\Psi^{-1}(B)). \tag{59}$$

From the definition of pushforward measure, $\exists B' \in \mathcal{B}(\mathbb{R}^{f_0})$, such that

$$\mathbb{P}_{\boldsymbol{\xi}^s_i}(B') \neq \mathbb{P}_{\boldsymbol{\xi}^s_j}(B'). \tag{60}$$

Therefore,

$$\mathbb{P}_{\boldsymbol{\xi}_i^s} \neq \mathbb{P}_{\boldsymbol{\xi}_j^s} \tag{61}$$

and

$$W_p(\mathbb{P}_{\boldsymbol{\xi}_i^s}, \mathbb{P}_{\boldsymbol{\xi}_j^s}) = \mathrm{FCD}_p(v_i, v_j) > 0. \tag{62}$$

□

Theorem 9. FCD_p *has the following discriminative power w.r.t.* $\hat{\Theta}$ *when the matrix of weights* W *and* σ *are invertible.*

$$\mathrm{FCD}_p(v_i, v_j) = 0 \Leftrightarrow W_p(\hat{\Theta}(v_i), \hat{\Theta}(v_j)) = 0 \tag{63}$$
$$\mathrm{FCD}_p(v_i, v_j) > 0 \Leftrightarrow W_p(\hat{\Theta}(v_i), \hat{\Theta}(v_j)) > 0 \tag{64}$$

Proof. Let $\Psi : \mathbb{R}^{f_0} \to \mathbb{R}^{f_L}$ be the application of weights and element-wise nonlinearity to the node feature matrix Ξ^s after structural update, defined as $\Psi(X^s) = \sigma(X^s W)$. If W and σ are invertible then Ψ is a bijective continuous function with continuous inverse $\Psi^{-1}(\cdot)$, therefore $\Psi(\cdot)$ is homeomorphism. Note that this also means that $f_0 = f_L = f$. In order to prove the discriminative power we need to show that

$$W_p(\mathbb{P}_{\boldsymbol{\xi}_i^s}, \mathbb{P}_{\boldsymbol{\xi}_j^s}) = 0 \Leftrightarrow W_p(\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}}, \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}}) = 0 \tag{65}$$
$$W_p(\mathbb{P}_{\boldsymbol{\xi}_i^s}, \mathbb{P}_{\boldsymbol{\xi}_j^s}) > 0 \Leftrightarrow W_p(\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}} \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}}) > 0 \tag{66}$$

Eq. (65) $\Rightarrow$: If

$$W_p(\mathbb{P}_{\boldsymbol{\xi}_i^s}, \mathbb{P}_{\boldsymbol{\xi}_j^s}) = \mathrm{FCD}_p(v_i, v_j) = 0, \tag{67}$$

then from the properties of Wasserstein distance (which is a true metric)

$$\mathbb{P}_{\boldsymbol{\xi}_i^s} = \mathbb{P}_{\boldsymbol{\xi}_j^s}, \tag{68}$$

which means that $\forall B \in \mathcal{B}(\mathbb{R}^f)$

$$\mathbb{P}_{\boldsymbol{\xi}_i^s}(B) = \mathbb{P}_{\boldsymbol{\xi}_j^s}(B). \tag{69}$$

By definition of pullback measure, and because Ψ is a homeomorphism, this is equivalent to

$$\mathbb{P}_{\boldsymbol{\xi}_i^s}(\Psi^{-1}(B)) = \mathbb{P}_{\boldsymbol{\xi}_j^s}(\Psi^{-1}(B)),\ \forall B \in \mathcal{B}(\mathbb{R}^f), \tag{70}$$

which by definition means that

$$\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}}(B) = \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}}(B),\ \forall B \in \mathcal{B}(\mathbb{R}^f), \tag{71}$$

and therefore

$$\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}} = \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}}. \tag{72}$$

From the properties of Wasserstein distance we have that

$$W_p(\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}}\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}}) = 0. \tag{73}$$

Eq. (65) $\Leftarrow$: If

$$W_p(\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}}, \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}}) = 0, \tag{74}$$

then from the properties of Wasserstein distance (which is a true metric), the following holds

$$\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}} = \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}} \tag{75}$$

or equivalently

$$\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}}(B) = \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}}(B),\ \forall B \in \mathcal{B}(\mathbb{R}^{f_L}), \tag{76}$$

which by definition means that

$$\mathbb{P}_{\boldsymbol{\xi}_i^s}(\Psi^{-1}(B)) = \mathbb{P}_{\boldsymbol{\xi}_j^s}(\Psi^{-1}(B)),\ \forall B \in \mathcal{B}(\mathbb{R}^{f_L}). \tag{77}$$

By definition of pushforward measure and because Ψ is a homeomorphism, this means

$$\mathbb{P}_{\boldsymbol{\xi}_i^s}(B') = \mathbb{P}_{\boldsymbol{\xi}_j^s}(B'),\ \forall B' \in \mathcal{B}(\mathbb{R}^{f_0}) \tag{78}$$

or

$$\mathbb{P}_{\boldsymbol{\xi}_i^s} = \mathbb{P}_{\boldsymbol{\xi}_j^s} \tag{79}$$

and thus

$$W_p(\mathbb{P}_{\boldsymbol{\xi}_i^s}\mathbb{P}_{\boldsymbol{\xi}_j^s}) = \mathrm{FCD}_p(v_i, v_j) = 0. \tag{80}$$

Eq. (66)$\Rightarrow$: If

$$W_p(\mathbb{P}_{\boldsymbol{\xi}_i^s}, \mathbb{P}_{\boldsymbol{\xi}_j^s}) = \mathrm{FCD}_p(v_i, v_j) > 0 \tag{81}$$

then

$$\mathbb{P}_{\boldsymbol{\xi}_i^s} \neq \mathbb{P}_{\boldsymbol{\xi}_j^s}. \tag{82}$$

Therefore, $\exists B' \in \mathcal{B}(\mathbb{R}^{f_0})$, such that

$$\mathbb{P}_{\boldsymbol{\xi}_i^s}(B') \neq \mathbb{P}_{\boldsymbol{\xi}_j^s}(B'). \tag{83}$$

By the definition of pullback measure, $\exists B \in \mathcal{B}(\mathbb{R}^{f_L})$ such that

$$\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}}(\Psi^{-1}(B)) \neq \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}}(\Psi^{-1}(B)), \tag{84}$$

thus

$$\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}} \neq \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}} \tag{85}$$

and

$$W_p(\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}} \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}}) > 0. \tag{86}$$

Eq. (66)$\Leftarrow$: If

$$W_p(\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}} \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}}) > 0, \tag{87}$$

then

$$\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}} \neq \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}}. \tag{88}$$

Therefore, $\exists B \in \mathcal{B}(\mathbb{R}^{f_L})$, such that

$$\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}}(B) \neq \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}}(B) \tag{89}$$

or by definition

$$\mathbb{P}_{\boldsymbol{\xi}_i^s}(\Psi^{-1}(B)) \neq \mathbb{P}_{\boldsymbol{\xi}_j^s}(\Psi^{-1}(B)) \tag{90}$$

From the definition of pushforward measure, $\exists B' \in \mathcal{B}(\mathbb{R}^{f_0})$, such that

$$\mathbb{P}_{\boldsymbol{\xi}_i^s}(B') \neq \mathbb{P}_{\boldsymbol{\xi}_j^s}(B'). \tag{91}$$

Therefore, it holds that

$$\mathbb{P}_{\boldsymbol{\xi}_i^s} \neq \mathbb{P}_{\boldsymbol{\xi}_j^s} \tag{92}$$

and

$$W_p(\mathbb{P}_{\boldsymbol{\xi}_i^s}, \mathbb{P}_{\boldsymbol{\xi}_j^s}) = \mathrm{FCD}_p(v_i, v_j) > 0. \tag{93}$$

□

Corollary 2. *In the case of Dirac measures, the discriminative power of* FCD_p *remains valid and can be expressed as follows:*

$$\left|\left|[\sigma(S^L X^{(0)} W)]_i - [\sigma(S^L X^{(0)} W)]_j\right|\right|_p > 0 \Rightarrow \left|\left|[S^L X^{(0)}]_i - [S^L X^{(0)}]_j\right|\right|_p > 0 \tag{94}$$

Corollary 3. *The discriminative power of* FCD_p *is the same as for* $\hat{\Theta}(\cdot)$ *in the default SGC architecture* $\Theta(\cdot)$.

E LIPSCHITZ CONTINUITY OF SGC WITH NONLINEAR ACTIVATION

The following theorem restates Theorem 3 from the main text, followed by its proof.

Theorem 10. $\hat{\Theta}(\cdot)$ *is a globally Lipschitz function w.r.t.* FCD_p*:*

$$W_p(\hat{\Theta}(v_i), \hat{\Theta}(v_j)) \leq C_L \cdot \mathrm{FCD}_p(v_i, v_j) \tag{95}$$

Proof. We need to show that

$$W_p(\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}}, \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}}) \leq C_L W_p(\mathbb{P}_{\boldsymbol{\xi}_i^s}, \mathbb{P}_{\boldsymbol{\xi}_j^s}). \tag{96}$$

Let $\Psi(X^s) = \sigma(X^s W)$ be the application of weights and nonlinearity after the structural update. The function $\Psi : \mathbb{R}^f \to \mathbb{R}^f$ is Lipschitz continuous in L_p-norm if σ is Lipschitz continuous (in L_p -norm) and we denote its Lipschitz constant as C_L,

$$||\Psi(x) - \Psi(y)||_p \leq C_L \, ||x - y||_p \, . \tag{97}$$

Let $\boldsymbol{\xi}_i^s, \boldsymbol{\xi}_j^s \in \mathbb{R}^f$ be the random variables of node features after structural updates at nodes v_i and v_j respectively. Let $\Gamma(\mathbb{P}_{\boldsymbol{\xi}_i^s}, \mathbb{P}_{\boldsymbol{\xi}_j^s})$ the space of couplings between them, and let $\Gamma(\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}}, \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}})$ be the space of couplings after pushforward through Ψ. The elements γ' of $\Gamma(\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}}, \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}})$ are of the form $\gamma' = (\Psi \times \Psi)_{\sharp\gamma}$.

For all $A \in \mathcal{B}(\mathbb{R}^f)$ we have:

$$\begin{aligned} \gamma'(A \times \mathbb{R}^f) &= (\Psi \times \Psi)_\sharp \gamma(A \times \mathbb{R}^f) = \gamma((\Psi \times \Psi)^{-1}(A \times \mathbb{R}^f)) \\ &= \gamma(\Psi^{-1}(A) \times \mathbb{R}^f) = \mathbb{P}_{\boldsymbol{\xi}_i^s}(\Psi^{-1}(A)) = \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}}(A). \end{aligned} \tag{98}$$

And mutatis mutandis for the other variable,

$$\gamma'(\mathbb{R}^f \times B) = \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}}(B). \tag{99}$$

The p-Wasserstein distance between the two pushforward probability measures is given by:

$$W_p(\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}}, \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}}) = \left(\inf_{\gamma' \in \Gamma(\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}}, \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}})} \iint_{\mathbb{R}^f \times \mathbb{R}^f} \|u - v\|^p \, d\gamma'(u, v) \right)^{1/p}. \tag{100}$$

Using Equation (97), it is clear that

$$\begin{aligned} \iint_{\mathbb{R}^f \times \mathbb{R}^f} \|u - v\|^p \, d\gamma'(u, v) &= \iint_{\mathbb{R}^f \times \mathbb{R}^f} \|\Psi(x) - \Psi(y)\|^p \, d\gamma(x, y) \\ &\leq C_L^p \iint_{\mathbb{R}^f \times \mathbb{R}^f} \|x - y\|^p \, d\gamma(x, y). \end{aligned} \tag{101}$$

Since this holds for any coupling $\gamma \in \Gamma(\mathbb{P}_{\boldsymbol{\xi}_i^s}, \mathbb{P}_{\boldsymbol{\xi}_j^s})$, it holds for the infimum over such couplings

$$\begin{aligned} &W_p^p(\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}}, \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}}) = \inf_{\gamma' \in \Gamma(\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}}, \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}})} \iint_{\mathbb{R}^f \times \mathbb{R}^f} \|u - v\|^p \, d\gamma'(u, v) \\ &\leq C_L^p \inf_{\gamma \in \Gamma(\mathbb{P}_{\boldsymbol{\xi}_i^s}, \mathbb{P}_{\boldsymbol{\xi}_j^s})} \iint_{\mathbb{R}^f \times \mathbb{R}^f} \|x - y\|^p \, d\gamma(x, y) = C_L^p W_p^p(\mathbb{P}_{\boldsymbol{\xi}_i^s}, \mathbb{P}_{\boldsymbol{\xi}_j^s}). \end{aligned} \tag{102}$$

Taking the p-th root on both sides yields the final bound

$$W_p(\Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_i^s}}, \Psi_{\sharp\mathbb{P}_{\boldsymbol{\xi}_j^s}}) \leq C_L W_p(\mathbb{P}_{\boldsymbol{\xi}_i^s}, \mathbb{P}_{\boldsymbol{\xi}_j^s}) = C_L \mathrm{FCD}_p(v_i, v_j). \tag{103}$$

□

Corollary 4. *In the case of the delta Dirac measures, the Lipschitz continuity of $\hat{\Theta}$ is valid and can be expressed as follows:*

$$\left|\left|[\sigma(S^L X W)]_i - [\sigma(S^L X W)]_j\right|\right|_p \leq \; C_L \left|\left|[S^L X]_i - [S^L X]_j\right|\right|_p \tag{104}$$

Corollary 5. *The Lipschitz continuity remains valid in the default SGC architecture $\Theta(\cdot)$.*

F COMPACTNESS

The probability measures of the structural updates of node features endowed with W_p come from the subset of Wasserstein space $(V, \mathrm{FCD}_p) \subset \mathcal{P}_p(\mathbb{R}^{f_0})$. The probability measures $\mathbb{P}_{\boldsymbol{\xi}_i}$ associated with nodes comes from a class of Gaussian measures $\mathcal{N}(\mu, \Sigma)$ with uniform exponential moments:

$$\int_{\mathbb{R}^{f_0}} e^{a\|x\|} d\mathcal{N}(\mu, \Sigma)(x) \leq M, \tag{105}$$

where $a > 0$ is some constant and $M < \infty$. Therefore, V is defined by the class of Gaussian measures $\mathcal{N}(\mu, \Sigma)$ with uniform exponential moments:

$$\int_{\mathbb{R}^{f_0}} e^{a\|x\|} d\mathcal{N}(\mu, \Sigma)(x) \leq \bar{M} \tag{106}$$

with constant $a > 0$ and $\bar{M} < \infty$.

The following theorem restates Theorem 4 from the main text, followed by its proof.

Theorem 11. (V, FCD_p) *is compact with the covering number*

$$N(\epsilon; V; \mathrm{FCD}_p) \leq \exp\{\hat{C}\epsilon^{-f_0}(\log[1/\epsilon])^{f_0}\}, \tag{107}$$

where constant $\hat{C}$ depends on f_0 and p.

Proof. Let $(\mathbb{P}_n)_{n\geq 1} \subset V$ be an arbitrary sequence. The class of Gaussian measures V has uniform exponential moments. The uniform exponential moment bound implies uniform bounds on all polynomial moments, and in particular on the second moments. Hence, the family V is tight. Since $\mathbb{R}^{f_0}$ is a Polish space, Prokhorov's theorem implies that V is precompact in the topology of weak convergence. Therefore, there exist a subsequence $(\mathbb{P}_{n_k})_{k\geq 1}$ and probability measure $\mathbb{P}$ such that $\mathbb{P}_{n_k} \Rightarrow \mathbb{P}$.

We now show that this subsequence actually converges in FCD_p. Let us consider the following for $R > 0$:

$$\int_{\{\|x\|>R\}} \|x\|^p d\mathbb{P}_n(x) = \int_{\{\|x\|>R\}} \big(\|x\|^p e^{-a\|x\|}\big) e^{a\|x\|} d\mathbb{P}_n(x) \tag{108}$$

We can also define

$$c_R := \sup_{t\geq R} t^p e^{-at} \tag{109}$$

Since

$$\|x\|^p e^{-a\|x\|} \leq c_R, \ \|x\| > R \tag{110}$$

it follows that

$$\int_{\{\|x\|>R\}} \|x\|^p d\mathbb{P}_n(x) \leq c_R \int_{\{\|x\|>R\}} e^{a\|x\|} d\mathbb{P}_n(x) \leq c_R \int_{\mathbb{R}^{f_0}} e^{a\|x\|} d\mathbb{P}_n(x) \tag{111}$$

Since $t^p e^{-at} \to 0$ as $t \to \infty$, we have $c_R \to 0$ as $R \to \infty$. Therefore,

$$\lim_{R\to\infty} \sup_n \int_{\{\|x\|>R\}} \|x\|^p d\mathbb{P}_n(x) = 0 \tag{112}$$

In particular,

$$\lim_{R\to\infty} \sup_{k\geq 1} \int_{\{\|x\|>R\}} \|x\|^p \, d\mathbb{P}_{n_k}(x) = 0. \tag{113}$$

Combined with a weak precompactness, following Theorem 2.2.1 (Panaretos and Zemel, 2020), this implies $\mathbb{P}_{n_k} \xrightarrow{\mathrm{FCD}_p} \mathbb{P}$. Thus, every sequence in V admits a FCD_p-convergent subsequence, and therefore (V, FCD_p) is precompact.

$\mathcal{P}_p(X)$ is a complete and separable metric space (Villani et al., 2008). (V, FCD_p) is a subspace of $\mathcal{P}_p(X)$. A subset of a complete space is complete iff it is closed. (Panaretos and Zemel, 2020) show that the Wasserstein topology is finer than the weak topology. Therefore, if a sequence in V converges in FCD_p, i.e., $\mathbb{P}_n \xrightarrow{\mathrm{FCD}_p} \mathbb{P}$, then it converges weakly, i.e., $\mathbb{P}_n \to \mathbb{P}$. If V is weakly closed, then the FCD_p-limit belongs to V. Therefore, to show that (V, FCD_p) is closed, we can show that it is weakly closed. To achieve this, we need to show that $\mathbb{P}$ is a Gaussian measure with bounded exponential moments.

We first show that $\mathbb{P}$ satisfies the same exponential moment bound. The function $f(x) = e^{a\|x\|}$ is lower semicontinuous, and bounded from below, so by Portmanteau's lemma,

$$\int_{\mathbb{R}^{f_0}} e^{a\|x\|} d\mathbb{P}(x) \leq \liminf_{n\to\infty} \int_{\mathbb{R}^{f_0}} e^{a\|x\|} d\mathbb{P}_n(x) \leq \bar{M} \tag{114}$$

Hence, the limit measure $\mathbb{P}$ also has a finite exponential moment.

Next, let us show that $\mathbb{P}$ is a Gaussian distribution. By Lévy's continuity theorem, weak convergence $\mathbb{P}_n \to \mathbb{P}$ implies the pointwise convergence of the characteristic functions

$$\phi_{\mathbb{P}_n}(t) \to \phi_{\mathbb{P}}(t). \tag{115}$$

Since each $\mathbb{P}_n$ is Gaussian with mean μ_n and covariance Σ_n, its characteristic function is

$$\phi_{\mathbb{P}_n}(t) = e^{(it^\top \mu_n - \frac{1}{2} t^\top \Sigma_n t)} \tag{116}$$

The uniform exponential moment bound implies uniform integrability of the first and second moments. Combining uniform integrability with weak convergence results in convergence of means (Bogachev, 2007)

$$\mu_n \to \mu \tag{117}$$

and the convergence of covariances

$$\Sigma_n \to \Sigma \tag{118}$$

Therefore, the characteristic functions converge to

$$\phi_{\mathbb{P}_n}(t) \to \phi_{\mathbb{P}}(t) = e^{(it^\top \mu - \frac{1}{2} t^\top \Sigma t)} \tag{119}$$

which is the characteristic function of a Gaussian measure. Hence, $\mathbb{P}$ is Gaussian with mean μ and covariance Σ. Therefore, V is closed in the topology of weak convergence. Thus, (V, FCD_p) is also closed and complete. A complete and precompact space is compact. Therefore, (V, FCD_p) is compact.

From (Panaretos and Zemel, 2020) for the family of measures with uniform exponential moments, the covering number is upper bounded by:

$$N(\epsilon; V; \mathrm{FCD}_p) \leq \exp\{\hat{C}\epsilon^{-f_0}(\log[1/\epsilon])^{f_0}\} \tag{120}$$

where constant $\hat{C}$ depends on f_0 and p (Panaretos and Zemel, 2020). The exact formula for the bound on the covering number is the following

$$N(\epsilon; V; \mathrm{FCD}_p) \leq \exp\{C_1 \epsilon^{-f_0} \log[1/\epsilon]^{f_0}[(f_0+1)\log[\epsilon^{-1}\log[1/\epsilon]] + C_2]\} \tag{121}$$

where $C_1 = 3^{f_0} e\theta$, $C_2 = (f_0+1)\log 3 + (f_0+2)\log 2 + \log\theta$ and $\theta = f_0[5 + \log f_0 + \log\log f_0]$.

□

G PARTITION OF (V, FCD_p)

Let $S = \{\mathbb{P}_1, \ldots, \mathbb{P}_K\} \subset V$ be a maximal ϵ-separated subset (i.e., $W_p(\mathbb{P}_i, \mathbb{P}_j) > \epsilon$ for $i \neq j$, and it is not possible add any new point while keeping separation larger than ϵ). Maximality implies coverage:

$$V \subset \bigcup_{k=1}^{K} B(\mathbb{P}_k, \epsilon) \tag{122}$$

If (V, FCD_p) is compact, then $K < \infty$ and $N(\epsilon; V; \mathrm{FCD}_p) \leq K$.

Define Voronoi cells truncated to the ϵ-balls by

$$C_1 = \{\mathbb{P} \in V : W_p(\mathbb{P}, c_1) \leq \epsilon \text{ and } W_p(\mathbb{P}, c_1) \leq W_p(\mathbb{P}, c_j),\ \forall j\} \tag{123}$$

where $c_i, i \in [K]$ - the centers of Wasserstein balls. For $k \geq 2$

$$C_k = \{\mathbb{P} \in V : W_p(\mathbb{P}, c_k) \leq \epsilon,\ W_p(\mathbb{P}, c_k) < W_p(\mathbb{P}, c_j),\ \forall j < k,\ W_p(\mathbb{P}, c_k) \leq W_p(\mathbb{P}, c_j),\ \forall j > k\} \tag{124}$$

This tie-break makes the C_k pairwise disjoint and Borel (measurable). Because the balls cover V, we also get

$$V = \bigsqcup_{k=1}^{K} C_k \tag{125}$$

For any $\mathbb{P} \in C_k$, $W_p(\mathbb{P}, c_k) \leq \epsilon$. Hence any two points in the same cell satisfy

$$W_p(\mathbb{P}, \mathbb{P}') \leq W_p(\mathbb{P}, c_k) + W_p(c_k, \mathbb{P}') \leq 2\epsilon, \tag{126}$$

so $\mathrm{diam}(C_k) \leq 2\epsilon$. The number of Voronoi cells is equal to the number of the covering balls $B(\mathbb{P}_k, \epsilon)$ and is equal to K.

H CONCENTRATION INEQUALITY

Theorem 12. *Suppose that the random variable X is represented as*

$$X = \sum_{\alpha \in \mathcal{A}} Y_\alpha \tag{127}$$

where Y_α is a random variable with α ranging over some index set. Let $a_\alpha \leq Y_\alpha \leq b_\alpha$ for every $\alpha \in \mathcal{A}$. Then, $\forall t > 0$

$$P(X - \mathbb{E}[X] \geq t) \leq \exp\left(-\frac{2t^2}{\chi(G) \sum_{\alpha \in \mathcal{A}} (b_\alpha - a_\alpha)^2}\right). \tag{128}$$

where $\chi(G)$ is the chromatic number of dependency graph G of Y_α.

Proof. This is Theorem 2.1 from (Janson, 2004). We refer the reader to (Janson, 2004) for the proof. □

Lemma 1. *Let $\{\mathbb{P}_i\}_{i=1}^n$ be probability measures which are G-dependent identically distributed with common probability measure $\nu \in \mathcal{P}(\mathcal{P}(\mathcal{X}))$, where $\mathcal{P}(\mathcal{X})$ is the space of probability measures on $\mathcal{X} \subseteq \mathbb{R}^d$. Let $\{C_1, \ldots, C_K\}$ be a measurable, pairwise disjoint partition of $\mathcal{P}(\mathcal{X})$. Define the number of probability measures in each partition C_j as*

$$Z_j := \sum_{i=1}^{n} \mathbf{1}\{\mathbb{P}_i \in C_j\},\ j \in [K] \tag{129}$$

and let the probability that the random measure $\mathbb{P}_i$ belongs to the partition C_j be as follows

$$\nu(C_j) = P(\mathbb{P}_i \in C_j) \tag{130}$$

Then for all $t > 0$

$$P\left(\sum_{j=1}^{K} |Z_j - n\nu(C_j)| \geq 2t\right) \leq 2^{K+1} \exp\left(-\frac{2t^2}{\chi(G)n}\right), \tag{131}$$

where $\chi(G)$ denotes the chromatic number of the dependency graph G.

Note. In the settings of Theorem 13, $\{\mathbb{P}_i\}_{i=1}^n$ are the probability measures associated with nodes $\{\mathbb{P}_{\boldsymbol{\xi}_i}\}_{i=1}^n$, and $\mathcal{P}(\mathcal{X})$ is the Wasserstein space of order p, $\mathcal{X} \subseteq \mathbb{R}^{f_0}$, where f_0 is the number of features of the node.

Proof. In this proof we follow the proof of Lemma 28 from (Vasileiou et al., 2025b). Let us define the number of times $\mathbb{P}_i$ appears in some subset of partitions as

$$Y_i^{(S)} := \sum_{j \in S} \mathbf{1}\{\mathbb{P}_i \in C_j\}, \forall S \subseteq [K]. \tag{132}$$

Notice that $Y_i^{(S)} \in \{0, 1\}$ since C_j are pairwise disjoint partitions and

$$\sum_{j \in S} Z_j = \sum_{i=1}^{n} Y_i^{(S)}. \tag{133}$$

Each $Y_i^{(S)}$ is a measurable function of $\mathbb{P}_i$. It follows that the family $(Y_i^{(S)})_{i=1}^n$ is G-dependent. Moreover,

$$\mathbb{E}[Y_i^{(S)}] = \mathbb{E}[\sum_{j \in S} \mathbf{1}\{\mathbb{P}_i \in C_j\}] = \sum_{j \in S} \mathbb{E}[\mathbf{1}\{\mathbb{P}_i \in C_j\}] = \sum_{j \in S} \mathrm{P}(\mathbb{P}_i \in C_j) = \sum_{j \in S} \nu(C_j) \tag{134}$$

where the sets C_j are disjoint and the $\mathbb{P}_i$ have common probability measure ν. Let us consider

$$\begin{aligned} \sum_{j=1}^{K} |Z_j - n\nu(C_j)| &= \sum_{j: Z > n\nu(C_j)} (Z_j - n\nu(C_j)) + \sum_{j: Z < n\nu(C_j)} (n\nu(C_j) - Z_j) \\ &= \max_{S \subseteq [K]} \sum_{j \in S} (Z_j - n\nu(C_j)) + \max_{S \subseteq [K]} \sum_{j \in S} (n\nu(C_j) - Z_j) \leq 2 \max\{A, B\} \end{aligned} \tag{135}$$

where

$$\begin{aligned} A &:= \max_{S \subseteq [K]} \sum_{j \in S} (Z_j - n\nu(C_j)), \\ B &:= \max_{S \subseteq [K]} \sum_{j \in S} (n\nu(C_j) - Z_j). \end{aligned} \tag{136}$$

Then

$$\begin{aligned} \mathrm{P}\left(\sum_{j=1}^{K}|Z_j - n\nu(C_j)| \geq 2t\right) &\leq \mathrm{P}\left(\max\{A, B\} > t\right) \\ &= \mathrm{P}\left((A \geq t) \cup (B \geq t)\right) \leq \mathrm{P}(A \geq t) + \mathrm{P}(B \geq t) \end{aligned} \tag{137}$$

Let us consider

$$\begin{aligned} P(A \geq t) \\ = \mathrm{P}\left(\cup_{S\subseteq[K]}\left(\sum_{j\in S}(Z_j - n\nu(C_j)\right) \geq t\right) \\ \leq \sum_{S\subseteq[K]}\mathrm{P}\left(\sum_{j\in S}\big(Z_j - n\nu(C_j)\big) \geq t\right) \\ = \sum_{S\subseteq[K]}\mathrm{P}\left(\sum_{j\in S}Z_j - n\sum_{j\in S}\nu(C_j) \geq t\right) \\ = \sum_{S\subseteq[K]}\mathrm{P}\left(\sum_{j\in S}Z_j - \sum_{i=1}^{n}\sum_{j\in S}\nu(C_j) \geq t\right) \\ = \sum_{S\subseteq[K]}\mathrm{P}\left(\sum_{j\in S}Z_j - \sum_{i=1}^{n}\mathbb{E}\left[Y_i^{(S)}\right] \geq t\right) \\ = \sum_{S\subseteq[K]}\mathrm{P}\left(\sum_{i=1}^{n}Y_i^{(S)} - \mathbb{E}\left[\sum_{i=1}^{n}Y_i^{(S)}\right] \geq t\right) \end{aligned} \tag{138}$$

Following Theorem 12 and noticing that $Y_i^{(S)} \in \{0, 1\}$, therefore, the squared range $(b_\alpha - a_\alpha)^2$ is always 1 and $\sum_{\alpha\in\mathcal{A}}(b_\alpha - a_\alpha)^2 = n$ we can show that

$$P(A \geq t) \leq \sum_{S\subset[K]}\exp\left(-\frac{2t^2}{\chi(G)n}\right) = 2^K\exp\left(-\frac{2t^2}{\chi(G)n}\right) \tag{139}$$

where 2^K is the number of all possible subsets consisting of partitions C_j. Similarly, for $\mathrm{P}(B \geq t)$ let

$$\begin{aligned} \sum_{j\in S} -Z_j &= \sum_{i=1}^{n}\hat{Y}_i^{(S)} \\ \hat{Y}_i^{(S)} = -\sum_{j\in S}\mathbf{1}\{\mathbb{P}_i \in C_j\},\ \hat{Y}_i^{(S)} &\in \{0, -1\} \\ \mathbb{E}[\hat{Y}_i^{(S)}] &= -\sum_{j\in S}\nu(C_j) \end{aligned} \tag{140}$$

Therefore,

$$\mathrm{P}(B \geq t) \leq \sum_{S\subseteq[K]}\mathrm{P}\left(\sum_{i=1}^{n}\hat{Y}_i^{(S)} - \mathbb{E}\left[\sum_{i=1}^{n}\hat{Y}_i^{(S)}\right] \geq t\right) \leq \sum_{S\subset[K]}\exp\left(-\frac{2t^2}{\chi(G)n}\right) = 2^K\exp\left(-\frac{2t^2}{\chi(G)n}\right) \tag{141}$$

And

$$\mathrm{P}\left(\sum_{j=1}^{K}|Z_j - n\nu(C_j)| \geq 2t\right) \leq \mathrm{P}(A \geq t) + \mathrm{P}(B \geq t) \leq 2^{K+1}\exp\left(-\frac{2t^2}{\chi(G)n}\right) \tag{142}$$

□

I GENERALIZATION GUARANTEES

Theorem 13. *Let $\mathcal{A}$ be the (K, ϵ)-uniformly robust learning algorithm on $\mathcal{Z}$. Then for any $\delta > 0$, with probability at least $1-\delta$, and for all samples $S = \{(\mathbb{P}_{\boldsymbol{\xi}_i}, y_i)\}_{i=1}^{N}$ drawn from probability measure ν on $\mathcal{Z}$ with the dependency graph $G[S]$, we have:*

$$W_p\left(\widehat{\mathbb{P}}_{\ell,S}, \mathbb{P}_{\ell,S}\right) \leq \epsilon + M\sqrt{\frac{\chi(G[S])}{|S|}(2(K+1)\log 2 + 2\log(\frac{1}{\delta}))} \tag{143}$$

where

$$\widehat{\mathbb{P}}_{\ell,S} := \frac{1}{N}\sum_{i=1}^{N}\ell(\mathcal{A}_S, (\mathbb{P}_{\boldsymbol{\xi}_i}, y_i)) \tag{144}$$

is the empirical loss probability measure, and

$$\mathbb{P}_{\ell,S} := \mathbb{E}_{(\mathbb{P}_{\boldsymbol{\xi}}, y)\sim\nu}[\ell(\mathcal{A}_S, (\mathbb{P}_{\boldsymbol{\xi}}, y))] \tag{145}$$

is the population loss probability measure, M is the diameter of the space of probability measures associated with losses

$$M := \sup_{(\mathbb{P}_{\boldsymbol{\xi}_i}, y_i), (\mathbb{P}_{\boldsymbol{\xi}_k}, y_j)\in\mathcal{Z}} W_p(\ell(\mathcal{A}_{\mathcal{S}}, (\mathbb{P}_{\boldsymbol{\xi}_i}, y_i)), \ell(\mathcal{A}_S, (\mathbb{P}_{\boldsymbol{\xi}_j}, y_j))) < \infty \tag{146}$$

and K is the covering number of the sample space $\mathcal{Z}$, $\chi(G)$ is the chromatic number of dependency graph G.

Proof. Let $\{C_j\}_{j=1}^{K}$ be a measurable partition of $\mathcal{Z}$ by the uniform robustness property such that

$$\sup_{(\mathbb{P}_{\boldsymbol{\xi}_i}, y_i), (\mathbb{P}_{\boldsymbol{\xi}_k}, y_j)\in C_j} W_p(\ell(A_S, (\mathbb{P}_{\boldsymbol{\xi}_i}, y_i)), \ell(A_S, (\mathbb{P}_{\boldsymbol{\xi}_k}, y_k))) \leq \epsilon,\ \forall j \in [K] \tag{147}$$

Let $N_j := \{i : \mathbb{P}_{\boldsymbol{\xi}_i} \in C_j\}$ and

$$\bar{\mathbb{P}}_{\ell,S,j} := \mathbb{E}_{(\mathbb{P}_{\boldsymbol{\xi}}, y)\sim\nu}[\ell(A_S, (\mathbb{P}_{\boldsymbol{\xi}}, y))|(\mathbb{P}_{\boldsymbol{\xi}}, y) \in C_j] \tag{148}$$

be the expected loss probability measure conditioned on a particular partition C_j.

We can write the probability measure of the empirical loss using the partition as

$$\widehat{\mathbb{P}}_{\ell,S} = \frac{1}{N}\sum_{j=1}^{K}\sum_{i\in N_j}\ell(\mathcal{A}_S,(\mathbb{P}_{\boldsymbol{\xi}_i},y_i)) \tag{149}$$

We can also write the probability measure of the population loss using the partition and leveraging the law of total probability as

$$\mathbb{P}_{\ell,S} = \sum_{j=1}^{K}\nu(C_j)\bar{\mathbb{P}}_{\ell,S,j} \tag{150}$$

Let us now consider $\sum_j \frac{|N_j|}{N}\bar{\mathbb{P}}_{\ell,S,j}$ and use the triangle inequality:

$$\begin{aligned} &W_p\left(\widehat{\mathbb{P}}_{\ell,S},\mathbb{P}_{\ell,S}\right)\\ &= W_p\left(\frac{1}{N}\sum_{j=1}^{K}\sum_{i\in N_j}\ell(\mathcal{A}_S,(\mathbb{P}_{\boldsymbol{\xi}_i},y_i)),\sum_{j=1}^{K}\nu(C_j)\bar{\mathbb{P}}_{\ell,S,j}\right)\\ &\leq W_p\left(\frac{1}{N}\sum_{j=1}^{K}\sum_{i\in N_j}\ell(\mathcal{A}_S,(\mathbb{P}_{\boldsymbol{\xi}_i},y_i)),\sum_{j=1}^{K}\frac{|N_j|}{N}\bar{\mathbb{P}}_{\ell,S,j}\right)\\ &\quad + W_p\left(\sum_{j=1}^{K}\frac{|N_j|}{N}\bar{\mathbb{P}}_{\ell,S,j},\sum_{j=1}^{K}\nu(C_j)\bar{\mathbb{P}}_{\ell,S,j}\right) = C + D \end{aligned} \tag{151}$$

Let us now consider C which is related to the differences in probability losses inside each partition C_j

$$\begin{aligned} C &= W_p\left(\frac{1}{N}\sum_{j=1}^{K}\sum_{i\in N_j}\ell(\mathcal{A}_S,(\mathbb{P}_{\boldsymbol{\xi}_i},y_i)),\sum_{j=1}^{K}\frac{|N_j|}{N}\bar{\mathbb{P}}_{\ell,S,j}\right)\\ &= W_p\left(\frac{1}{N}\sum_{i=1}^{N}\ell(\mathcal{A}_S,(\mathbb{P}_{\boldsymbol{\xi}_i},y_i)),\frac{1}{N}\sum_{i=1}^{N}\bar{\mathbb{P}}_{\ell,S,j(i)}\right)\\ &\leq \frac{1}{N}\sum_{i=1}^{N}W_p\left(\ell(\mathcal{A}_S,(\mathbb{P}_{\boldsymbol{\xi}_i},y_i)),\bar{\mathbb{P}}_{\ell,S,j(i)}\right) \leq \frac{N}{N}\epsilon = \epsilon \end{aligned} \tag{152}$$

Now let us consider D which represents the error of placing random measures into particular cell C_j

$$\begin{aligned} D &= W_p\left(\sum_{j=1}^{K}\frac{|N_j|}{N}\bar{\mathbb{P}}_{\ell,S,j},\sum_{j=1}^{K}\nu(C_j)\bar{\mathbb{P}}_{\ell,S,j}\right)\\ &\leq \max_{i,k}W_p\left(\bar{\mathbb{P}}_{\ell,S,i},\bar{\mathbb{P}}_{\ell,S,k}\right)\sum_{j=1}^{K}\left|\frac{|N_j|}{N}-\nu(C_j)\right| \leq M\sum_{j=1}^{K}\left|\frac{|N_j|}{N}-\nu(C_j)\right| \end{aligned} \tag{153}$$

Therefore, D is upper bounded by the upper bound on the cost needed to move the unit mass between cells C_j multiplied by the total mass that gets moved.

From Lemma 1 we know that

$$\mathrm{P}\left(\sum_{j=1}^{K}|Z_j - N\nu(C_j)| \geq 2t\right) \leq 2^{K+1}\exp\left(-\frac{2t^2}{\chi(G)N}\right) \tag{154}$$

Let us consider

$$\mathrm{P}\left(\sum_{j=1}^{K}\left|\frac{|N_j|}{N} - \nu(C_j)\right| \geq s\right) \tag{155}$$

Let $s = \frac{2t}{N}$, therefore

$$\mathrm{P}\left(\sum_{j=1}^{K}\left|\frac{|N_j|}{N} - \nu(C_j)\right| \geq s\right) = \mathrm{P}\left(\sum_{j=1}^{K}|N_j - N\nu(C_j)| \geq 2t\right) \tag{156}$$

Thus

$$\mathrm{P}\left(\sum_{j=1}^{K}\left|\frac{|N_j|}{N} - \nu(C_j)\right| \geq s\right) \leq 2^{K+1}\exp\left(-\frac{s^2N}{2\chi(G)}\right) \tag{157}$$

Let us now choose s so that this probability is at most δ

$$2^{K+1}\exp\left(-\frac{s^2N}{2\chi(G)}\right) = \delta \tag{158}$$

Taking logarithm, rearranging the terms and solving for s we get:

$$s = \sqrt{\frac{2\chi(G)}{N}((K+1)\log 2 + 2\log(\frac{1}{\delta}))} \tag{159}$$

Therefore,

$$D \leq M\sqrt{\frac{\chi(G(S))}{|S|}(2(K+1)\log 2 + 2\log(\frac{1}{\delta}))} \tag{160}$$

And

$$W_p\left(\widehat{\mathbb{P}}_{\ell,S}, \mathbb{P}_{\ell,S}\right) \leq C + D \leq \epsilon + M\sqrt{\frac{\chi(G(S))}{|S|}(2(K+1)\log 2 + 2\log\frac{1}{\delta})} \tag{161}$$

□

The following theorem restates Theorem 6 from the main text, followed by its proof.

Theorem 14. *Let $\mathcal{A}$ be a learning algorithm on $\mathcal{Z}$. Then for any $\delta > 0$, with probability at least $1-\delta$, and for a sample $S = \{(\mathbb{P}_{\boldsymbol{\xi}_i}, y_i)\}_{i=1}^N$ with the dependency graph $G[S]$ and a C_ℓ-Lipschitz loss function ℓ the following holds*

$$W_p(\widehat{\mathbb{P}}_{\ell,S}, \mathbb{P}_{\ell,S}) \leq 2C\epsilon + M\sqrt{\frac{\chi(G[S])}{|S|}(2(K_\epsilon+1)\log 2 + 2\log(\frac{1}{\delta}))} \tag{162}$$

where

$$\widehat{\mathbb{P}}_{\ell,S} := \frac{1}{N}\sum_{i=1}^{N} \ell(\mathcal{A}_S, (\mathbb{P}_{\boldsymbol{\xi}_i}, y_i)) \tag{163}$$

is the empirical loss probability measure, and

$$\mathbb{P}_{\ell,S} := \mathbb{E}_{(\mathbb{P}_{\boldsymbol{\xi}}, y)\sim\nu}[\ell(\mathcal{A}_S, (\mathbb{P}_{\boldsymbol{\xi}}, y))] \tag{164}$$

is the population loss probability measure, M is the diameter of the space of probability measures associated with losses

$$M := \sup_{(\mathbb{P}_{\boldsymbol{\xi}_i}, y_i), (\mathbb{P}_{\boldsymbol{\xi}_j}, y_j)\in\mathcal{Z}} W_p(\ell(\mathcal{A}_{\mathcal{S}}, (\mathbb{P}_{\boldsymbol{\xi}_i}, y_i)), \ell(\mathcal{A}_S, (\mathbb{P}_{\boldsymbol{\xi}_j}, y_j))) < \infty \tag{165}$$

where $K_\epsilon = N(\epsilon; \mathcal{Z}; d_{\mathcal{Z}})$ is the covering number of $\mathcal{Z}$, $C = C_L C_\ell$, C_L is FCD_p *Lipschitz constant, $\chi(G)$ is the chromatic number of dependency graph G.*

Proof. We can partition $\mathcal{Z}$ into $K_\epsilon = \mathcal{C}N(\epsilon; V; \mathrm{FCD}_p)$ subsets according to uniform robustness property of $\mathcal{A}$ such that each subset C_j has a diameter less or equal to $2C\epsilon$, where ϵ is the cover radius of the space of probability measures of nodes. Constant $C = C_L C_\ell$, where C_L is the Lipschitz constant associated with FCD_p and C_ℓ is the Lipschitz constant of the loss function. Therefore, the proof is a straightforward application of the Theorem 13 bound.

□

The following corollary restates Corollary 1 from the main text, followed by its proof.

Corollary 6. *$W_1\left(\mathbb{P}_{\ell,S}, \widehat{\mathbb{P}}_{\ell,S}\right)$ is an upper bound on the risk gap $|\ell_{exp}(A_S) - \ell_{emp}(A_S)|$ as follows*

$$|\ell_{exp}(\mathcal{A}_S) - \ell_{emp}(\mathcal{A}_S)| \leq W_1(\mathbb{P}_{\ell,S}, \widehat{\mathbb{P}}_{\ell,S}) \tag{166}$$

where $\ell_{exp}(\mathcal{A}_S)$ and $\ell_{emp}(A_S)$ are the expected values of the loss drawn from the measures $\mathbb{P}_{\ell,S}$ and $\widehat{\mathbb{P}}_{\ell,S}$, respectively.

Proof. From Kantorovich-Rubinstein duality (Thickstun, 2019) the following is true

$$W_1\left(\mathbb{P}_{\ell,S}, \widehat{\mathbb{P}}_{\ell,S}\right) \geq \left|\mathbb{E}_{l\sim\mathbb{P}_{\ell,S}}[h(l)] - \mathbb{E}_{\hat{l}\sim\widehat{\mathbb{P}}_{\ell,S}}[h(\hat{l})]\right| \tag{167}$$

where $h(\cdot)$ is 1-Lipschitz function. Let $h(\cdot)$ be the identity function $h(x) = x$.[2] Therefore

[2]For the identity function $h(x) = x$, the 1-Lipschitz condition $|h(x) - h(y)| \leq 1 \cdot |x - y|$ is satisfied because $|x - y| = 1 \cdot |x - y|$.

$$
\begin{aligned}
W_1\left(\mathbb{P}_{\ell,S}, \widehat{\mathbb{P}}_{\ell,S}\right) &\geq \left|\mathbb{E}_{l\sim\mathbb{P}_{\ell,S}}[l] - \mathbb{E}_{\hat{l}\sim\widehat{\mathbb{P}}_{\ell,S}}[\hat{l}]\right| \\
&= \left|\mathbb{E}_{(\mathbb{P}_{\xi_l}, y_l)\sim\mu}[\ell(\mathcal{A}_S, (\mathbb{P}_{\xi_l}, y_l))] - \mathbb{E}_{(\mathbb{P}_{\xi_{\hat{l}}}, y_{\hat{l}})\sim\hat{\mu}}[\ell(\mathcal{A}_S, (\mathbb{P}_{\xi_{\hat{l}}}, y_{\hat{l}}))]\right| \\
&= |\ell_{exp}(A_S) - \ell_{emp}(A_S)|
\end{aligned} \tag{168}
$$

where $\mathbb{P}_{\ell,S}$ and $\widehat{\mathbb{P}}_{\ell,S}$ are pushforward measures for μ and $\hat{\mu}$ under the loss map, respectively.

□

J EXPERIMENTS

J.1 Synthetic Data

For our synthetic data we use graphs sampled from the planted partition stochastic block model (SBM) with two groups, and node features sampled from a Gaussian Markov Random Field (GMRF) with a conditional independence structure governed by the graph.

We denote the number of vertices in group $g \in \{0, 1\}$ as n_g, and the expected number of edges between groups g and g' as $k_{gg'}$.

To sample a random graph from this model, we first calculate the probability p_{ij} of an edge between vertices v_i and v_j,

$$
p_{ij} = \frac{k_{g_i g_j}}{n_{g_j}}, \tag{169}
$$

where g_i is the group assignment of node v_i. Then, we decide on presence or absence of an edge by drawing $A_{ij} \sim \text{Bern}[p_{ij}]$ for $i < j$ and setting $A_{ji} = A_{ij}$, where A is the graphs adjacency matrix and Bern$[p]$ denotes the Bernoulli distribution with bias p. In our experiments we use $n = 256$ nodes divided into groups of equal size $n_1 = n_2 = 128$ with $k_{00} = k_{11} = 8$ expected intra-group connections, and $k_{01} = k_{10} = 4$ expected inter group connections.

To define node features, we first define the distribution of node features, i.e., the distribution of the random variable $\vec{\boldsymbol{\xi}} \in \mathbb{R}^{nf}$, and then sample a particular realization $\vec{\boldsymbol{x}} \in \mathbb{R}^{nf}$.

We assume that node features, at the same node and across nodes, follow a joint multivariate Gaussian distribution $\mathcal{N}(\mu, \Lambda^{-1})$ with precision matrix, i.e., inverse covariance, $\Lambda \in \mathbb{R}^{nf\times nf}$ and mean $\mu \in \mathbb{R}^{nf}$. To enforce that the graphs adjacency matrix governs conditional independence between features we construct Λ as a Kronecker product,

$$
\Lambda = I_n \otimes \Sigma_V + L \otimes \beta I_f, \tag{170}
$$

where $I_n \in \mathbb{R}^{n\times n}, I_f \in \mathbb{R}^{f\times f}$ are the n and f-dimensional identity matrices respectively, $\Sigma_V \in \mathbb{R}^{f\times f}$ is the marginal covariance at a node in absence of edge coupling, $L = D - A \in \mathbb{R}^{n\times n}$ is the combinatorial graph Laplacian, and $\gamma \in \mathbb{R}$ is the edge coupling strength. Larger values of γ induce stronger correlation between the same feature at different nodes that are connected by an edge. We further parametrize Σ_f in terms of a single standard deviation parameter $\sigma \geq 0$, and correlation parameter $\rho \in [-1, 1]$ as

$$
\Sigma_f = \sigma((\sigma - \rho)I_f + \rho \mathbf{1}_f \mathbf{1}_f^\mathsf{T}), \tag{171}
$$

where $\mathbf{1}_f \in \mathbb{R}^f$ is the vector of all ones.

We choose the mean of node v_i's features as a function of its group membership,

$$
\mu = \mu_0(\mathbf{1}_{n_0} \otimes \mathbf{1}_f) + \mu_1(\mathbf{1}_{n_1} \otimes \mathbf{1}_f), \tag{172}
$$

where $\mu_0, \mu_1 \in \mathbb{R}$ govern the mean of node features for nodes in group $g = 0$ and $g = 1$ respectively.

For our experiments, we always set the mean parameters to $\mu_0 = 2$ and $\mu_1 = -2$ and use 3-dimensional ($f = 3$) features. We consider three different settings for the precision matrix:

- **independent nodes independent features (inif)**: Features are iid random variables $\sigma = 0.25$, $\rho = 0$ $\gamma = 0$.
- **independent nodes dependent features (indf)**: Features at the same node are correlated but there are no correlations between features at different nodes, $\sigma = 0.25$, $\rho = 0.5$, $\gamma = 0$.
- **dependent nodes dependent features (dndf)**: Features at the same node are correlated, and the same feature at adjacent nodes is correlated, $\sigma = 1$, $\rho = 0.5$, $\gamma = 1.5$. This setting ensures that the average marginal covariance satisfies, $\frac{1}{nf}\sum_{i=1}^{nf}[\Lambda^{-1}]_{ii} \approx 0.25$.

For numerical stability, we draw samples from the standard normal distribution and transform them into samples from $\mathcal{N}(\mu, \Lambda^{-1})$ using the Cholesky factorization of Λ of its inverse $\Sigma = \Lambda^{-1}$.

Architectures and Training. In all our experiments, we use the following architectures: Simple Graph Convolution (SGC) with $L = 2$ or $L = 3$ layers, and Graph Convolutional Network (GCN) with $L = 2$ and $L = 3$ layers with $f_1 = f_2 = 16$ hidden features, and nonlinearity in $\sigma \in \{\texttt{ReLU}, \texttt{GELU}, \texttt{tanh}, \texttt{sigmoid}\}$.

For training we use the empirical mean $\tilde{\mu} = \frac{1}{N}\sum_{i=1}^{n_{\mathrm{MC}}} X^{(i)}$, where $\{X^{(i)}\}_{i=1}^{n_{\mathrm{MC}}}$ are $n_{\mathrm{MC}} = 10^4$ samples obtained as described in the previous section. We randomly assign 70% of the nodes to the training set and the remaining 30% to the test set. We train using the `AdaBelief` optimizer as implemented in `optax` (DeepMind et al., 2020) with learning rate $\eta = 10^{-3}$ and perform $n_{\mathrm{epochs}} = 1000$ full-batch gradient updates.

For moment propagation and certified adversarial robustness, we used the transductive learning scenario. To illustrate generalization results, we used the inductive setting. Tables 4 and 5, respectively, show accuracy results on synthetic data for the inductive and transductive scenarios.

Table 4: Accuracy for the Inductively Trained Models on Synthetic Data When Evaluated on the Mean. Note that these are all equal because the datasets have the same mean and only differ in their correlation structure.

dataset	L	reps	α_{test}	α_{train}	α_{rand}
`inif`	2	1000	0.94	0.96	0.50
`inif`	3	1000	0.91	0.91	0.50
`indf`	2	1000	0.94	0.96	0.50
`indf`	3	1000	0.91	0.91	0.50
`dndf`	2	1000	0.94	0.96	0.50
`dndf`	3	1000	0.91	0.91	0.50

Additional Results on Moment Propagation. We use the `time` function from the `time` module of `python` to estimate the runtime of each method, as the runtimes of these methods are on the order of minutes rather than sub-seconds this method is viable. Storing all samples can be memory intensive. Hence, we employ a batched, online algorithm (Pebay, 2008) for moment estimation from samples. We use a batch size of $n_{\mathrm{batch}} = 50$.

Tables 6, 7, and 8 provide additional results on moment propagation for different methods on synthetic data. As in the main text, we find that layer-wise Taylor approximations (`1d-T1`, `1d-T2-Tr`) match or outperform their high-dimensional counterparts (`T1`,`T2`) while being more memory efficient. For `tanh` and to a lesser extent `GELU` nonlinearities, performance can be slightly improved by including Gaussian moment closure (`1d-T2-GC`). For $l = 2$ first order methods (`T1`, `1d-T1`) perform badly on `GELU` but not `ReLU`. For $l = 3$ first order methods perform poorly on both `GELU` and `ReLU` and errors are comparable in this case. `PTPE` shows best performance for nonlinearities other than `sigmoid`, where it fails. When comparing results between `inif` and `indf`, we find that they are qualitatively similar. For `dndf` the errors are considerably larger than in the `inif` and `indf` cases. In summary, this means that `PTPE` is a strong baseline for moment propagation but layer-wise second order Taylor approximations can be a viable alternative. We prefer Gaussian moment closure (`1d-T2-GC`) over truncation (`1d-T2`) as this approach can sometimes improve performance, but only marginally increases computation cost in the case of elementwise functions.

Table 5: Accuracy for the Transductively Trained Models on Synthetic Data When Evaluated on the Mean. Note that these are all equal for the same hyperparameters on different datasets because the datasets have the same mean and only differ in their correlation structure.

dataset	L	activation	reps	f_l	α_{test}	α_{train}	α_{rand}
inif	2	ReLU	1000	16	0.92	0.91	0.50
inif	3	ReLU	1000	16	0.90	0.93	0.50
inif	2	tanh	1000	16	0.92	0.92	0.50
inif	3	tanh	1000	16	0.90	0.92	0.50
inif	2	GELU	1000	16	0.90	0.91	0.50
inif	3	GELU	1000	16	0.92	0.94	0.50
inif	2	sigmoid	1000	16	0.92	0.91	0.50
inif	3	sigmoid	1000	16	0.90	0.93	0.50
indf	2	ReLU	1000	16	0.92	0.91	0.50
indf	3	ReLU	1000	16	0.90	0.93	0.50
indf	2	tanh	1000	16	0.92	0.92	0.50
indf	3	tanh	1000	16	0.90	0.92	0.50
indf	2	GELU	1000	16	0.92	0.91	0.50
indf	3	GELU	1000	16	0.92	0.94	0.50
indf	2	sigmoid	1000	16	0.92	0.91	0.50
indf	3	sigmoid	1000	16	0.90	0.93	0.50
dndf	2	ReLU	1000	16	0.92	0.91	0.50
dndf	3	ReLU	1000	16	0.90	0.93	0.50
dndf	2	tanh	1000	16	0.92	0.92	0.50
dndf	3	tanh	1000	16	0.90	0.92	0.50
dndf	2	GELU	1000	16	0.92	0.91	0.50
dndf	3	GELU	1000	16	0.92	0.94	0.50
dndf	2	sigmoid	1000	16	0.92	0.91	0.50
dndf	3	sigmoid	1000	16	0.90	0.93	0.50

Additional Results on Robustness. Tables 9, 10, and 11 show additional results on robustness for synthetic data for GCNs with `GELU`, `ReLU`, `tanh`, and `sigmoid` nonlinearities and $L = 2$ and $L = 3$ layers with $f_1 = f_2 = 16$ hidden features.

As a baseline method, we implement the method presented in Cohen et al. (Cohen et al., 2019). The algorithm has four hyperparameters, which we set as follows: A sample size $n_0 = 100$, a (ideally larger) sample size $n_1 = 9900$, a confidence $\delta = 0.05$, and a standard deviation $\sigma = \frac{1}{nf_0}\sum_{i=1}^{nf_0}\sigma_i$ of the isotropic Gaussian noise distribution. In contrast to our approach, Cohen et al. assume isotropic Gaussian noise. We can thus not use the same feature distribution as in our method. Instead, we use an isotropic Gaussian with standard deviation equal to the average empirically estimated standard deviation of our samples.

Additional Results on Generalization. The results on generalization bounds for the synthetic data are in Table 12.

J.2 Real-world Data

Datasets. In our experiments, we use seven real-world graphs from different domains including scientific collaboration, online encyclopedias, and academic web pages. All datasets are obtained from `torch_geometric.datasets` (Fey et al., 2025). Below we report the number of nodes n, the number of edges m, and the node feature dimension $\tilde{f}_0$ of each graph, as well as what nodes and edges represent.

- The `Cornell` ($n = 183$, $m = 298$, $\tilde{f}_0 = 1703$), `Texas` ($n = 183$, $m = 325$, $\tilde{f}_0 = 1703$), and `Wisconsin` ($n = 251$, $m = 515$, $\tilde{f}_0 = 1703$) datasets are part of the `WebKB` collection. Nodes represent academic webpages and edges represent hyperlinks between them. The node features are bag-of-words features created from the content of each webpage.

- The `Cora` ($n = 2708$, $m = 10,556$, $\tilde{f}_0 = 1433$) and `Citeseer` ($n = 3327$, $m = 9104$, $\tilde{f}_0 = 3703$) are part of the `Planetoid` collection. Nodes represent scientific publications and edges indicate citations. Node features are bag-of-words features created from the abstract of a given publication.

Table 6: `inif` Dataset: Moment Propagation in Terms of L_2-norm $||\cdot||_2$, Wasserstein distance W_2, and Runtime T with Varying Number of Layers L, Nonlinear Activation Function, and Method.

L	f_l	activation	method	$\|\|\hat{\mu} - \hat{\mu}_{MC}\|\|_2$	$\|\|\hat{\sum} - \hat{\sum}_{MC}\|\|_F$	$W_2(\mathcal{N}, \mathcal{N}_{MC})$	T (seconds)
2	16	ReLU	T1	0.05	0.02	0.05	5.07
2	16	ReLU	T2-Tr	0.08	0.01	0.09	2.03
2	16	ReLU	PTPE	0.0003	0.002	0.005	7.96
2	16	ReLU	1d-T1	0.05	0.02	0.05	10.21
2	16	ReLU	1d-T2-Tr	0.08	0.02	0.09	9.99
2	16	ReLU	1d-T2-GC	0.08	0.02	0.09	10.31
2	16	ReLU	sampling	0.00	0.00	0.00	1.79
2	16	GELU	T1	0.18	0.01	0.18	5.84
2	16	GELU	T2-Tr	0.001	0.03	0.05	2.02
2	16	GELU	PTPE	0.02	0.003	0.02	8.09
2	16	GELU	1d-T1	0.18	0.01	0.18	10.45
2	16	GELU	1d-T2-Tr	0.0004	0.04	0.04	10.09
2	16	GELU	1d-T2-GC	0.0005	0.01	0.02	10.28
2	16	GELU	sampling	0.00	0.00	0.00	1.99
2	16	tanh	T1	0.04	0.01	0.05	4.85
2	16	tanh	T2-Tr	0.001	0.01	0.03	1.52
2	16	tanh	PTPE	0.003	0.001	0.004	8.19
2	16	tanh	1d-T1	0.04	0.01	0.05	9.98
2	16	tanh	1d-T2-Tr	0.001	0.01	0.02	10.20
2	16	tanh	1d-T2-GC	0.001	0.01	0.01	10.34
2	16	tanh	sampling	0.00	0.00	0.00	1.81
2	16	sigmoid	T1	0.03	0.003	0.03	5.02
2	16	sigmoid	T2-Tr	0.0002	0.003	0.03	1.63
2	16	sigmoid	PTPE	13.45	0.30	13.45	8.33
2	16	sigmoid	1d-T1	0.03	0.003	0.03	9.99
2	16	sigmoid	1d-T2-Tr	0.0002	0.002	0.007	10.19
2	16	sigmoid	1d-T2-GC	0.0002	0.002	0.002	10.35
2	16	sigmoid	sampling	0.00	0.00	0.00	1.76
3	16	ReLU	T1	0.27	0.05	0.27	6.77
3	16	ReLU	T2-Tr	0.29	0.05	0.29	2.63
3	16	ReLU	PTPE	0.0003	0.004	0.007	14.38
3	16	ReLU	1d-T1	0.27	0.05	0.27	16.73
3	16	ReLU	1d-T2-Tr	0.28	0.06	0.29	16.41
3	16	ReLU	1d-T2-GC	0.29	0.06	0.29	16.86
3	16	ReLU	sampling	0.00	0.00	0.00	1.78
3	16	GELU	T1	0.27	0.02	0.27	5.89
3	16	GELU	T2-Tr	0.0006	0.06	0.04	2.71
3	16	GELU	PTPE	0.02	0.01	0.03	14.26
3	16	GELU	1d-T1	0.27	0.02	0.27	16.63
3	16	GELU	1d-T2-Tr	0.0006	0.08	0.04	16.33
3	16	GELU	1d-T2-GC	0.0006	0.04	0.02	16.74
3	16	GELU	sampling	0.00	0.00	0.00	4.03
3	16	tanh	T1	0.04	0.01	0.04	6.86
3	16	tanh	T2-Tr	0.002	0.01	0.03	2.03
3	16	tanh	PTPE	0.01	0.004	0.006	14.29
3	16	tanh	1d-T1	0.04	0.01	0.04	16.30
3	16	tanh	1d-T2-Tr	0.002	0.017	0.02	16.57
3	16	tanh	1d-T2-GC	0.002	0.017	0.01	16.94
3	16	tanh	sampling	0.00	0.00	0.00	1.64
3	16	sigmoid	T1	0.09	0.05	0.10	7.03
3	16	sigmoid	T2-Tr	0.01	0.05	0.06	2.01
3	16	sigmoid	PTPE	14.67	0.64	14.68	14.42
3	16	sigmoid	1d-T1	0.09	0.05	0.10	16.15
3	16	sigmoid	1d-T2-Tr	0.01	0.05	0.05	16.47
3	16	sigmoid	1d-T2-GC	0.01	0.05	0.04	16.81
3	16	sigmoid	sampling	0.00	0.00	0.00	1.63

- The `Chameleon` ($n = 2277$, $m = 36,101$, $\tilde{f}_0 = 2325$) and `Squirrel` ($n = 5201$, $m = 217,073$, $\tilde{f}_0 = 2089$) datasets are part of the `WikipediaNetwork` collection. Nodes represent webpages of the online encyclopedia Wikipedia, and edges indicate hyperlinks between these webpages. Features are bag-of-words features

Table 7: `indf` Dataset: Moment Propagation in Terms of L_2-norm $||\cdot||_2$, Wasserstein distance W_2, and Runtime T with Varying Number of Layers L, Nonlinear Activation Function, and Method.

L	f_l	activation	method	$\|\|\hat{\mu} - \hat{\mu}_{MC}\|\|_2$	$\|\|\hat{\sum} - \hat{\sum}_{MC}\|\|_F$	$W_2(\mathcal{N}, \mathcal{N}_{MC})$	T (seconds)
2	16	ReLU	T1	0.05	0.03	0.05	5.14
2	16	ReLU	T2-Tr	0.08	0.03	0.09	2.06
2	16	ReLU	PTPE	0.0006	0.004	0.008	8.12
2	16	ReLU	1d-T1	0.05	0.03	0.05	10.38
2	16	ReLU	1d-T2-Tr	0.08	0.03	0.09	10.13
2	16	ReLU	1d-T2-GC	0.08	0.03	0.09	10.23
2	16	ReLU	sampling	0.00	0.00	0.00	1.63
2	16	GELU	T1	0.18	0.02	0.18	5.82
2	16	GELU	T2-Tr	0.002	0.04	0.06	2.15
2	16	GELU	PTPE	0.02	0.01	0.02	7.89
2	16	GELU	1d-T1	0.18	0.02	0.18	10.45
2	16	GELU	1d-T2-Tr	0.001	0.05	0.06	10.01
2	16	GELU	1d-T2-GC	0.002	0.02	0.06	10.29
2	16	GELU	sampling	0.00	0.00	0.00	1.99
2	16	tanh	T1	0.07	0.04	0.08	4.94
2	16	tanh	T2-Tr	0.004	0.04	0.06	1.43
2	16	tanh	PTPE	0.003	0.003	0.007	8.29
2	16	tanh	1d-T1	0.07	0.04	0.08	9.93
2	16	tanh	1d-T2-Tr	0.004	0.04	0.04	10.05
2	16	tanh	1d-T2-GC	0.004	0.04	0.04	10.22
2	16	tanh	sampling	0.00	0.00	0.00	1.69
2	16	sigmoid	T1	0.06	0.01	0.06	4.91
2	16	sigmoid	T2-Tr	0.0007	0.001	0.04	1.52
2	16	sigmoid	PTPE	13.45	0.61	13.45	8.22
2	16	sigmoid	1d-T1	0.06	0.01	0.06	9.85
2	16	sigmoid	1d-T2-Tr	0.0006	0.01	0.02	9.98
2	16	sigmoid	1d-T2-GC	0.0006	0.008	0.01	10.19
2	16	sigmoid	sampling	0.00	0.00	0.00	1.78
3	16	ReLU	T1	0.28	0.08	0.29	6.81
3	16	ReLU	T2-Tr	0.28	0.08	0.29	2.59
3	16	ReLU	PTPE	0.0006	0.01	0.01	14.25
3	16	ReLU	1d-T1	0.28	0.08	0.29	16.41
3	16	ReLU	1d-T2-Tr	0.28	0.08	0.29	16.38
3	16	ReLU	1d-T2-GC	0.28	0.09	0.29	16.50
3	16	ReLU	sampling	0.00	0.00	0.00	1.72
3	16	GELU	T1	0.32	0.08	0.33	5.84
3	16	GELU	T2-Tr	0.002	0.12	0.08	2.68
3	16	GELU	PTPE	0.02	0.01	0.03	14.07
3	16	GELU	1d-T1	0.32	0.08	0.33	16.53
3	16	GELU	1d-T2-Tr	0.002	0.16	0.08	16.15
3	16	GELU	1d-T2-GC	0.002	0.12	0.05	16.56
3	16	GELU	sampling	0.00	0.00	0.00	4.07
3	16	tanh	T1	0.07	0.05	0.07	6.70
3	16	tanh	T2-Tr	0.01	0.05	0.05	2.04
3	16	tanh	PTPE	0.01	0.01	0.01	14.39
3	16	tanh	1d-T1	0.07	0.05	0.07	16.01
3	16	tanh	1d-T2-Tr	0.01	0.05	0.04	16.37
3	16	tanh	1d-T2-GC	0.01	0.05	0.03	16.57
3	16	tanh	sampling	0.00	0.00	0.00	1.77
3	16	sigmoid	T1	0.17	0.15	0.20	6.87
3	16	sigmoid	T2-Tr	0.04	0.16	0.13	1.93
3	16	sigmoid	PTPE	14.67	1.27	14.69	14.27
3	16	sigmoid	1d-T1	0.17	0.15	0.20	15.96
3	16	sigmoid	1d-T2-Tr	0.04	0.16	0.13	16.12
3	16	sigmoid	1d-T2-GC	0.04	0.14	0.10	16.32
3	16	sigmoid	sampling	0.00	0.00	0.00	1.83

created from the content of each webpage.

We treat all graphs as undirected graphs. As the full covariance matrix has size $(n\tilde{f}_0)^2$ entries, using the initial

Table 8: `dndf` Dataset: Moment Propagation in Terms of L_2-norm $||\cdot||_2$, Wasserstein distance W_2, and Runtime T with Varying Number of Layers L, Nonlinear Activation Function, and Method.

L	f_l	activation	method	$\|\|\hat{\mu} - \hat{\mu}_{MC}\|\|_2$	$\|\|\hat{\sum} - \hat{\sum}_{MC}\|\|_F$	$W_2(\mathcal{N}, \mathcal{N}_{MC})$	T (seconds)
2	16	`ReLU`	`T1`	0.18	0.97	0.23	4.93
2	16	`ReLU`	`T2-Tr`	0.17	0.97	0.27	1.95
2	16	`ReLU`	`PTPE`	0.002	0.14	0.04	7.88
2	16	`ReLU`	`1d-T1`	0.18	0.97	0.23	10.05
2	16	`ReLU`	`1d-T2-Tr`	0.16	0.99	0.26	9.76
2	16	`ReLU`	`1d-T2-GC`	0.16	1.01	0.25	10.01
2	16	`ReLU`	sampling	0.00	0.00	0.00	1.67
2	16	`GELU`	`T1`	0.31	1.13	0.40	5.69
2	16	`GELU`	`T2-Tr`	0.03	1.14	0.32	2.23
2	16	`GELU`	`PTPE`	0.02	0.17	0.07	7.85
2	16	`GELU`	`1d-T1`	0.31	1.13	0.40	10.37
2	16	`GELU`	`1d-T2-Tr`	0.02	1.18	0.32	9.82
2	16	`GELU`	`1d-T2-GC`	0.02	2.52	0.71	10.11
2	16	`GELU`	sampling	0.00	0.00	0.00	2.04
2	16	`tanh`	`T1`	0.22	0.69	0.29	4.80
2	16	`tanh`	`T2-Tr`	0.03	0.69	0.26	1.42
2	16	`tanh`	`PTPE`	0.004	0.12	0.04	8.01
2	16	`tanh`	`1d-T1`	0.22	0.69	0.29	9.71
2	16	`tanh`	`1d-T2-Tr`	0.03	0.67	0.24	9.86
2	16	`tanh`	`1d-T2-GC`	0.03	0.67	0.20	10.16
2	16	`tanh`	sampling	0.00	0.00	0.00	1.60
2	16	`sigmoid`	`T1`	0.16	0.27	0.19	4.85
2	16	`sigmoid`	`T2-Tr`	0.004	0.27	0.16	1.57
2	16	`sigmoid`	`PTPE`	13.42	7.81	13.47	8.11
2	16	`sigmoid`	`1d-T1`	0.16	0.27	0.19	9.64
2	16	`sigmoid`	`1d-T2-Tr`	0.004	0.22	0.15	9.92
2	16	`sigmoid`	`1d-T2-GC`	0.004	0.21	0.08	10.02
2	16	`sigmoid`	sampling	0.00	0.00	0.00	1.65
3	16	`ReLU`	`T1`	0.67	7.16	0.87	6.60
3	16	`ReLU`	`T2-Tr`	0.51	7.23	0.81	2.50
3	16	`ReLU`	`PTPE`	0.01	0.27	0.16	13.88
3	16	`ReLU`	`1d-T1`	0.67	7.16	0.87	16.10
3	16	`ReLU`	`1d-T2-Tr`	0.51	7.24	0.81	15.95
3	16	`ReLU`	`1d-T2-GC`	0.51	7.93	0.80	16.33
3	16	`ReLU`	sampling	0.00	0.00	0.00	1.84
3	16	`GELU`	`T1`	0.66	9.89	1.08	5.62
3	16	`GELU`	`T2-Tr`	0.16	9.92	0.93	2.53
3	16	`GELU`	`PTPE`	0.03	0.53	0.25	13.96
3	16	`GELU`	`1d-T1`	0.66	9.89	1.08	16.21
3	16	`GELU`	`1d-T2-Tr`	0.16	10.25	0.95	15.93
3	16	`GELU`	`1d-T2-GC`	0.16	14.44	0.85	16.31
3	16	`GELU`	sampling	0.00	0.00	0.00	3.90
3	16	`tanh`	`T1`	0.45	3.20	0.66	6.74
3	16	`tanh`	`T2-Tr`	0.08	3.23	0.62	1.96
3	16	`tanh`	`PTPE`	0.008	0.64	0.12	14.07
3	16	`tanh`	`1d-T1`	0.45	3.20	0.66	15.86
3	16	`tanh`	`1d-T2-Tr`	0.08	3.32	0.62	16.05
3	16	`tanh`	`1d-T2-GC`	0.08	3.30	0.47	16.28
3	16	`tanh`	sampling	0.00	0.00	0.00	1.50
3	16	`sigmoid`	`T1`	0.34	1.95	0.47	6.81
3	16	`sigmoid`	`T2-Tr`	0.13	2.00	0.40	2.01
3	16	`sigmoid`	`PTPE`	14.66	11.47	14.72	14.43
3	16	`sigmoid`	`1d-T1`	0.34	1.94	0.47	15.96
3	16	`sigmoid`	`1d-T2-Tr`	0.11	1.95	0.40	16.31
3	16	`sigmoid`	`1d-T2-GC`	0.11	1.85	0.33	16.45
3	16	`sigmoid`	sampling	0.00	0.00	0.00	1.81

high-dimensional node features leads to prohibitively large matrices. For example, the full covariance matrix of the `Citeseer` dataset has ca. 150×10^{12} entries requiring approximately 600 TB of storage using 32 bit floating point numbers. We thus resort to dimensionality reduction and use 5-dimensional ($f_0 = 5$) node features,

Table 9: `inif` Dataset: Certified Adversarial Robustness Radii. L is the number of layers, f_l is the number of hidden dimensions, p_{CERT}^{Thm1} and p_{CERT}^{Chn} are the percentage of nodes for which a positive radius could be certified with Theorem 1 (Thm1) in our manuscript and Cohen's method (Chn) respectively, ϵ_{CERT}^{Thm1} and ϵ_{CERT}^{Chn} are the average robustness radius among nodes with a positive robustness radius using Theorem 1 or Cohen's method respectively, T^{Thm1} and T^{Chn} the runtime for computing radii using Theorem 1 or Cohen's method, for the former we include the time it takes to estimate the moments. Time is reported in seconds.

L	f_l	activation	method	p_{CERT}^{Thm1}	p_{CERT}^{Chn}	ϵ_{CERT}^{Thm1}	ϵ_{CERT}^{Chn}	T^{Thm1}	T^{Chn}	$\bar{C}$
2	16	`ReLU`	sample	0.95	1.00	0.08	0.84	3.85	563.70	29.99
2	16	`GELU`	sample	0.96	1.00	0.03	0.84	3.94	575.14	100.08
2	16	`tanh`	sample	0.96	1.00	0.04	0.84	3.77	557.77	66.93
2	16	`sigmoid`	sample	0.97	1.00	0.09	0.84	3.71	553.01	26.44
2	16	`ReLU`	`PTPE`	0.95	1.00	0.09	0.84	9.59	563.70	29.99
2	16	`GELU`	`PTPE`	0.96	1.00	0.03	0.84	9.62	575.14	100.08
2	16	`tanh`	`PTPE`	0.96	1.00	0.04	0.84	9.74	557.77	66.93
2	16	`sigmoid`	`PTPE`	0.94	1.00	0.07	0.84	9.86	553.01	26.44
2	16	`ReLU`	`1d-T2-GC`	0.95	1.00	0.09	0.84	11.94	563.70	29.99
2	16	`GELU`	`1d-T2-GC`	0.96	1.00	0.03	0.84	11.81	575.14	100.08
2	16	`tanh`	`1d-T2-GC`	0.95	1.00	0.04	0.84	11.88	557.77	66.93
2	16	`sigmoid`	`1d-T2-GC`	0.97	1.00	0.09	0.84	11.88	553.01	26.44
2	16	`ReLU`	`T1`	0.95	1.00	0.09	0.84	6.70	563.70	29.99
2	16	`GELU`	`T1`	0.96	1.0	0.03	0.84	7.37	575.14	100.08
2	16	`tanh`	`T1`	0.95	1.00	0.04	0.84	6.40	557.77	66.93
2	16	`sigmoid`	`T1`	0.97	1.00	0.09	0.84	6.55	553.01	26.44
3	16	`ReLU`	sample	0.92	1.00	0.04	0.83	3.74	576.05	71.06
3	16	`GELU`	sample	0.91	1.00	0.01	0.83	5.98	578.42	266.41
3	16	`tanh`	sample	0.92	1.00	0.01	0.83	3.59	570.56	435.14
3	16	`sigmoid`	sample	0.94	1.00	0.01	0.84	3.58	569.09	308.82
3	16	`ReLU`	`PTPE`	0.92	1.00	0.04	0.83	15.92	576.05	71.06
3	16	`GELU`	`PTPE`	0.91	1.00	0.01	0.83	15.81	578.42	266.41
3	16	`tanh`	`PTPE`	0.92	1.00	0.01	0.83	15.83	570.56	435.14
3	16	`sigmoid`	`PTPE`	0.93	1.0	0.01	0.84	15.95	569.09	308.82
3	16	`ReLU`	`1d-T2-GC`	0.92	1.00	0.04	0.83	18.40	576.05	71.06
3	16	`GELU`	`1d-T2-GC`	0.91	1.00	0.01	0.83	18.28	578.42	266.41
3	16	`tanh`	`1d-T2-GC`	0.92	1.00	0.01	0.83	18.48	570.56	435.14
3	16	`sigmoid`	`1d-T2-GC`	0.95	1.00	0.01	0.84	18.34	569.09	308.82
3	16	`ReLU`	`T1`	0.92	1.00	0.04	0.83	8.31	576.05	71.06
3	16	`GELU`	`T1`	0.91	1.00	0.01	0.83	7.43	578.42	266.41
3	16	`tanh`	`T1`	0.92	1.00	0.01	0.83	8.40	570.56	435.14
3	16	`sigmoid`	`T1`	0.95	1.00	0.01	0.84	8.56	569.09	308.82

obtained via Singular Value Decomposition (SVD), as input to our models.

To control for the influence of the relative size of the training data across datasets, we create our own train-test splits. We randomly assign 25% of nodes to the training data, and 75% of nodes to the test data. We do not use a validation set.

To obtain probabilistic features, we treat the features after SVD as the mean of a multivariate Gaussian distribution with diagonal covariance matrix $\Sigma = \text{diag}(\sigma_{11}^2, \ldots, \sigma_{nf_0}^2)$ with $\sigma_{ij} = \nu|X_{ij}|$, where $\nu = 0.05$ is the coefficient of variation and governs the size of the standard deviation relative to absolute magnitude the mean.

Architectures and Training. We study GCN architectures with $L = 2$ and $L = 3$ layers in combination with four nonlinearities $\sigma \in \{$`ReLU`, `GELU`, `tanh`, `sigmoid`$\}$. We hold the embedding dimension f_l constant across layers l, setting $f_l = 4$ or $f_l = 8$ based on dataset size and test performance. For our SGC architectures, we also consider $L = 2$ and $L = 3$ layers. Note that for SGC does not have hidden embeddings. The dimension of the resulting embedding is equal to the number of classes. We use a Xavier initialization with rescaling based on the fan-in, and set bias terms in GCN initially to zero. The SGC architecture does not have bias terms. Models are implemented using `equinox` (Kidger and Garcia, 2021) and `jax` (DeepMind et al., 2020).

We train all models using the `AdaGrad` optimizer with full-batch gradients and initial learning rate of $\eta = 10^{-3}$ as implemented in `optax` (DeepMind et al., 2020) and select the number of epochs $n_{\text{epoch}} \in \{250, 500, 1000, 2000, 4000\}$ based on best test performance.

Table 10: `indf` Dataset: Certified Adversarial Robustness Radii. L is the number of layers, f_l is the number of hidden dimensions, p_{CERT}^{Thm1} and p_{CERT}^{Chn} are the percentage of nodes for which a positive radius could be certified with Theorem 1 (Thm1) in our manuscript and Cohen's method (Chn) respectively, ϵ_{CERT}^{Thm1} and ϵ_{CERT}^{Chn} are the average robustness radius among nodes with a positive robustness radius using Theorem 1 or Cohen's method respectively, T^{Thm1} and T^{Chn} the runtime for computing radii using Theorem 1 or Cohen's method, for the former we include the time it takes to estimate the moments. Time is reported in seconds.

L	f_l	activation	method	p_{CERT}^{Thm1}	p_{CERT}^{Chn}	ϵ_{CERT}^{Thm1}	ϵ_{CERT}^{Chn}	T^{Thm1}	T^{Chn}	$\bar{C}$
2	16	ReLU	sample	0.93	1.00	0.08	0.84	3.58	573.54	30.10
2	16	GELU	sample	0.94	1.00	0.02	0.84	3.95	584.82	100.73
2	16	tanh	sample	0.93	1.00	0.04	0.84	3.65	577.36	66.91
2	16	sigmoid	sample	0.94	1.00	0.09	0.84	3.72	579.67	26.44
2	16	ReLU	PTPE	0.93	1.00	0.08	0.84	9.66	573.54	30.10
2	16	GELU	PTPE	0.94	1.00	0.02	0.84	9.42	584.82	100.73
2	16	tanh	PTPE	0.93	1.00	0.04	0.84	9.83	577.36	66.91
2	16	sigmoid	PTPE	0.93	1.00	0.07	0.84	9.74	579.67	26.44
2	16	ReLU	1d-T2-GC	0.93	1.00	0.08	0.84	11.82	573.54	30.10
2	16	GELU	1d-T2-GC	0.94	1.00	0.02	0.84	11.82	584.82	100.73
2	16	tanh	1d-T2-GC	0.93	1.00	0.04	0.84	11.76	577.36	66.91
2	16	sigmoid	1d-T2-GC	0.94	1.00	0.09	0.84	11.71	579.67	26.44
2	16	ReLU	T1	0.93	1.00	0.08	0.84	6.67	573.54	30.10
2	16	GELU	T1	0.94	1.00	0.02	0.84	7.34	584.82	100.73
2	16	tanh	T1	0.93	1.00	0.04	0.84	6.49	577.36	66.91
2	16	sigmoid	T1	0.94	1.00	0.09	0.84	6.44	579.67	26.44
3	16	ReLU	sample	0.87	1.00	0.04	0.83	3.67	594.13	71.22
3	16	GELU	sample	0.83	1.00	0.01	0.82	6.02	616.99	264.55
3	16	tanh	sample	0.90	1.00	0.01	0.83	3.74	609.29	435.26
3	16	sigmoid	sample	0.91	1.00	0.01	0.84	3.78	589.20	309.03
3	16	ReLU	PTPE	0.87	1.00	0.04	0.83	15.80	594.13	71.22
3	16	GELU	PTPE	0.83	1.00	0.01	0.82	15.61	616.99	264.55
3	16	tanh	PTPE	0.90	1.00	0.01	0.83	15.93	609.29	435.26
3	16	sigmoid	PTPE	0.92	1.00	0.01	0.84	15.81	589.20	309.03
3	16	ReLU	1d-T2-GC	0.89	1.00	0.03	0.83	18.05	594.13	71.22
3	16	GELU	1d-T2-GC	0.83	1.00	0.01	0.82	18.10	616.99	264.55
3	16	tanh	1d-T2-GC	0.90	1.00	0.01	0.83	18.10	609.29	435.26
3	16	sigmoid	1d-T2-GC	0.91	1.00	0.01	0.84	17.87	589.20	309.03
3	16	ReLU	T1	0.89	1.00	0.03	0.83	8.36	594.13	71.22
3	16	GELU	T1	0.83	1.00	0.01	0.82	7.38	616.99	264.55
3	16	tanh	T1	0.90	1.00	0.01	0.83	8.23	609.29	435.26
3	16	sigmoid	T1	0.91	1.00	0.01	0.84	8.42	589.20	309.03

For real-world data, Tables 13 and 14 report the accuracy results for the inductively trained SGC models and the transductively trained GCN models, respectively.

Additional Results on Moment Propagation. Tables 15, 16, 17, 18, 19, 20, and 21 contain results on moment propagation for the `Cornell`, `Wisconsin`, `Texas`, `Cora`, `Citeseer`, `Chameleon`, and `Squirrel` datasets, respectively.

Additional Results on Robustness Radii. Tables 22, 23, 24, 25, 26, 27, and 28 list results on certified adversarial robustness radii for the `Cornell`, `Wisconsin`, `Texas`, `Cora`, `Citeseer`, `Chameleon`, and `Squirrel` datasets, respectively.

Additional Results on Generalization. Table 29 presents the generalization bounds for for the `Cornell`, `Wisconsin`, `Texas`, `Cora`, `Citeseer`, `Chameleon`, and `Squirrel` datasets.

Table 11: `dndf` Dataset: Certified adversarial robustness radii. L is the number of layers, f_l is the number of hidden dimensions, p_{CERT}^{Thm1} and p_{CERT}^{Chn} are the percentage of nodes for which a positive radius could be certified with Theorem 1 (Thm1) in our manuscript and Cohen's method (Chn) respectively, ϵ_{CERT}^{Thm1} and ϵ_{CERT}^{Chn} are the average robustness radius among nodes with a positive robustness radius using Theorem 1 or Cohen's method respectively, T^{Thm1} and T^{Chn} the runtime for computing radii using Theorem 1 or Cohen's method, for the former we include the time it takes to estimate the moments. Time is reported in seconds.

L	f_l	activation	method	p_{CERT}^{Thm1}	p_{CERT}^{Chn}	ϵ_{CERT}^{Thm1}	ϵ_{CERT}^{Chn}	T^{Thm1}	T^{Chn}	$\bar{C}$
2	16	ReLU	sample	0.72	1.00	0.06	0.85	3.62	604.63	30.08
2	16	GELU	sample	0.71	1.00	0.02	0.85	4.00	604.64	99.98
2	16	tanh	sample	0.84	1.00	0.03	0.85	3.55	594.43	66.92
2	16	sigmoid	sample	0.85	1.00	0.07	0.85	3.60	597.25	26.46
2	16	ReLU	PTPE	0.72	1.00	0.06	0.85	9.41	604.63	30.08
2	16	GELU	PTPE	0.71	1.00	0.02	0.85	9.38	604.64	99.98
2	16	tanh	PTPE	0.83	1.00	0.03	0.85	9.53	594.43	66.92
2	16	sigmoid	PTPE	0.79	1.00	0.05	0.85	9.63	597.25	26.46
2	16	ReLU	1d-T2-GC	0.72	1.00	0.06	0.85	11.54	604.63	30.08
2	16	GELU	1d-T2-GC	0.71	1.00	0.02	0.85	11.63	604.64	99.98
2	16	tanh	1d-T2-GC	0.84	1.00	0.03	0.85	11.68	594.43	66.92
2	16	sigmoid	1d-T2-GC	0.85	1.00	0.07	0.85	11.54	597.25	26.46
2	16	ReLU	T1	0.72	1.00	0.06	0.85	6.46	604.63	30.08
2	16	GELU	T1	0.71	1.00	0.02	0.85	7.22	604.64	99.98
2	16	tanh	T1	0.84	1.00	0.03	0.85	6.33	594.43	66.92
2	16	sigmoid	T1	0.85	1.00	0.07	0.85	6.38	597.25	26.46
3	16	ReLU	sample	0.47	1.00	0.02	0.84	3.81	622.51	71.36
3	16	GELU	sample	0.40	1.00	0.01	0.84	5.84	903.78	266.71
3	16	tanh	sample	0.61	1.00	0.004	0.84	3.46	601.25	435.20
3	16	sigmoid	sample	0.79	1.00	0.01	0.86	3.77	586.11	309.56
3	16	ReLU	PTPE	0.47	1.00	0.02	0.84	15.43	622.51	71.36
3	16	GELU	PTPE	0.40	1.00	0.01	0.84	15.47	903.78	266.71
3	16	tanh	PTPE	0.61	1.00	0.004	0.84	15.61	601.25	435.20
3	16	sigmoid	PTPE	0.75	1.00	0.004	0.86	15.97	586.11	309.56
3	16	ReLU	1d-T2-GC	0.47	1.00	0.02	0.84	17.89	622.51	71.36
3	16	GELU	1d-T2-GC	0.40	1.00	0.005	0.84	17.84	903.78	266.71
3	16	tanh	1d-T2-GC	0.61	1.00	0.004	0.84	17.81	601.25	435.20
3	16	sigmoid	1d-T2-GC	0.80	1.00	0.01	0.86	17.99	586.11	309.56
3	16	ReLU	T1	0.47	1.00	0.02	0.84	8.15	622.51	71.36
3	16	GELU	T1	0.41	1.00	0.01	0.84	7.15	903.78	266.71
3	16	tanh	T1	0.61	1.00	0.004	0.84	8.28	601.25	435.20
3	16	sigmoid	T1	0.79	1.00	0.01	0.86	8.34	586.11	309.56

Table 12: Generalization Bounds on Synthetic Data.

dataset	L	emp. LHS	RHS	C	M	ρ_{S}	ρ_{P}	K	χ
inif	2	4.75	164.94	75.68	4.75	0.90	0.92	1.97	185
inif	3	5.11	141.29	63.15	5.11	0.89	0.91	1.97	185
indf	2	4.76	189.09	87.85	4.76	0.90	0.92	1.97	185
indf	3	5.12	152.30	68.69	5.12	0.89	0.91	1.97	185
dndf	2	4.75	134.94	60.53	4.75	0.90	0.92	1.97	185
dndf	3	5.10	133.47	59.22	5.10	0.89	0.91	1.97	185

Table 13: Accuracy for Inductively Trained Models on Real-world Data.

dataset	L	reps	α_{test}	α_{train}	α_{rand}
`Cornell`	2	4000	0.53	0.90	0.20
`Cornell`	3	1000	0.53	0.83	0.20
`Wisconsin`	2	1000	0.54	0.93	0.20
`Wisconsin`	3	1000	0.52	0.91	0.20
`Texas`	2	500	0.58	0.85	0.20
`Texas`	3	250	0.54	0.78	0.20
`Cora`	2	2000	0.57	0.51	0.14
`Cora`	3	2000	0.55	0.51	0.14
`Citeseer`	2	1000	0.68	0.63	0.17
`Citeseer`	3	1000	0.67	0.63	0.17
`Chameleon`	2	250	0.30	0.42	0.20
`Chameleon`	3	4000	0.30	0.43	0.20
`Squirrel`	2	250	0.22	0.26	0.20
`Squirrel`	3	250	0.23	0.22	0.20

Table 14: Accuracy for Transductively Trained Models on Real-world Data.

dataset	L	f_l	reps	nonlin	α_{test}	α_{train}	α_{rand}
Cornell	2	4	500	ReLU	0.49	0.68	0.20
Cornell	2	4	250	GELU	0.41	0.61	0.20
Cornell	2	4	1000	tanh	0.45	0.93	0.20
Cornell	2	4	500	sigmoid	0.52	0.66	0.20
Cornell	3	8	1000	ReLU	0.37	1.00	0.20
Cornell	3	8	500	GELU	0.42	1.00	0.20
Cornell	3	8	250	tanh	0.49	0.80	0.20
Cornell	3	8	250	sigmoid	0.48	0.63	0.20
Wisconsin	2	4	2000	ReLU	0.45	0.63	0.20
Wisconsin	2	4	4000	tanh	0.52	0.95	0.20
Wisconsin	2	4	250	GELU	0.46	0.57	0.20
Wisconsin	2	4	4000	sigmoid	0.52	0.91	0.20
Wisconsin	3	4	2000	ReLU	0.54	0.82	0.20
Wisconsin	3	4	2000	GELU	0.49	1.00	0.20
Wisconsin	3	4	500	tanh	0.45	0.86	0.20
Wisconsin	3	4	250	sigmoid	0.46	0.52	0.20
Texas	2	4	4000	ReLU	0.56	1.00	0.20
Texas	2	4	250	GELU	0.45	0.63	0.20
Texas	2	4	250	tanh	0.52	0.76	0.20
Texas	2	4	250	sigmoid	0.51	0.59	0.20
Texas	3	4	250	ReLU	0.50	0.71	0.20
Texas	3	4	1000	GELU	0.47	1.00	0.20
Texas	3	4	500	tanh	0.46	1.00	0.20
Texas	3	4	1000	sigmoid	0.50	0.93	0.20
Cora	2	8	1000	ReLU	0.69	0.80	0.14
Cora	2	8	2000	GELU	0.70	0.84	0.14
Cora	2	8	2000	tanh	0.68	0.81	0.14
Cora	2	8	4000	sigmoid	0.71	0.86	0.14
Cora	3	8	1000	ReLU	0.73	0.82	0.14
Cora	3	8	1000	GELU	0.72	0.87	0.14
Cora	3	8	2000	tanh	0.71	0.90	0.14
Cora	3	8	4000	sigmoid	0.72	0.93	0.14
Citeseer	2	4	4000	ReLU	0.69	0.72	0.17
Citeseer	2	4	4000	GELU	0.69	0.74	0.17
Citeseer	2	4	2000	tanh	0.69	0.74	0.17
Citeseer	2	4	2000	sigmoid	0.69	0.72	0.17
Citeseer	3	4	1000	ReLU	0.65	0.69	0.17
Citeseer	3	4	500	GELU	0.69	0.72	0.17
Citeseer	3	4	1000	tanh	0.70	0.72	0.17
Citeseer	3	4	4000	sigmoid	0.70	0.74	0.17
Chameleon	2	8	4000	ReLU	0.55	0.61	0.20
Chameleon	2	8	4000	GELU	0.56	0.66	0.20
Chameleon	2	8	4000	tanh	0.58	0.68	0.20
Chameleon	2	8	4000	sigmoid	0.58	0.68	0.20
Chameleon	3	8	4000	ReLU	0.56	0.66	0.20
Chameleon	3	8	4000	GELU	0.62	0.76	0.20
Chameleon	3	8	4000	tanh	0.56	0.67	0.20
Chameleon	3	8	4000	sigmoid	0.62	0.69	0.20
Squirrel	2	4	1000	ReLU	0.35	0.37	0.20
Squirrel	2	4	2000	GELU	0.37	0.39	0.20
Squirrel	2	4	4000	tanh	0.37	0.38	0.20
Squirrel	2	4	4000	sigmoid	0.37	0.38	0.20
Squirrel	3	4	4000	ReLU	0.38	0.37	0.20
Squirrel	3	4	500	GELU	0.37	0.38	0.20
Squirrel	3	4	4000	tanh	0.38	0.42	0.20
Squirrel	3	4	4000	sigmoid	0.36	0.40	0.20

Table 15: `Cornell` Dataset: Moment Propagation in Terms of L_2-norm $||\cdot||_2$, Wasserstein distance W_2, and Runtime T with Varying Number of Layers L, Nonlinear Activation Function, and Method.

L	f_l	activation	method	$\|\|\hat{\mu} - \hat{\mu}_{MC}\|\|_2$	$\|\|\hat{\sum} - \hat{\sum}_{MC}\|\|_F$	$W_2(\mathcal{N}, \mathcal{N}_{MC})$	T (seconds)
2	4	ReLU	T1	0.58	1.10	0.66	2.61
2	4	ReLU	T2-Tr	0.51	0.98	0.59	1.67
2	4	ReLU	PTPE	NaN	NaN	NaN	2.13
2	4	ReLU	1d-T1	0.58	1.10	0.66	1.49
2	4	ReLU	1d-T2-Tr	0.51	0.98	0.59	0.46
2	4	ReLU	1d-T2-GC	0.51	1.28	0.61	0.46
2	4	ReLU	sampling	0.00	0.00	0.00	18.37
2	4	GELU	T1	0.21	0.07	0.22	2.71
2	4	GELU	T2-Tr	0.01	0.09	0.13	1.67
2	4	GELU	PTPE	0.02	0.03	0.04	2.18
2	4	GELU	1d-T1	0.21	0.07	0.23	1.55
2	4	GELU	1d-T2-Tr	0.01	0.08	0.12	0.47
2	4	GELU	1d-T2-GC	0.01	0.07	0.08	0.44
2	4	GELU	sampling	0.00	0.00	0.00	20.07
2	4	tanh	T1	2.51	9.48	2.97	2.10
2	4	tanh	T2-Tr	1.09	14.09	2.64	1.04
2	4	tanh	PTPE	0.14	1.66	0.43	2.30
2	4	tanh	1d-T1	2.51	9.48	2.97	0.66
2	4	tanh	1d-T2-Tr	0.84	11.09	2.18	0.39
2	4	tanh	1d-T2-GC	0.88	9.13	2.06	0.43
2	4	tanh	sampling	0.00	0.00	0.00	19.94
2	4	sigmoid	T1	0.09	0.04	0.10	2.04
2	4	sigmoid	T2-Tr	0.01	0.04	0.07	1.06
2	4	sigmoid	PTPE	11.36	0.22	11.37	2.30
2	4	sigmoid	1d-T1	0.09	0.04	0.10	0.69
2	4	sigmoid	1d-T2-Tr	0.01	0.04	0.05	0.41
2	4	sigmoid	1d-T2-GC	0.01	0.04	0.05	0.40
2	4	sigmoid	sampling	0.00	0.00	0.00	18.37
3	8	ReLU	T1	4.32	41.08	5.95	2.65
3	8	ReLU	T2-Tr	5.29	61.20	7.19	2.18
3	8	ReLU	PTPE	0.29	9.80	1.01	2.87
3	8	ReLU	1d-T1	4.32	41.08	5.95	2.35
3	8	ReLU	1d-T2-Tr	5.49	50.71	7.42	1.53
3	8	ReLU	1d-T2-GC	5.49	55.00	7.41	1.59
3	8	ReLU	sampling	0.00	0.00	0.00	19.73
3	8	GELU	T1	7.84	83.04	10.46	2.83
3	8	GELU	T2-Tr	18.28	675.86	21.04	2.25
3	8	GELU	PTPE	0.31	25.07	1.93	2.99
3	8	GELU	1d-T1	7.84	83.04	10.46	2.53
3	8	GELU	1d-T2-Tr	13.56	344.97	16.74	1.51
3	8	GELU	1d-T2-GC	15.61	1307.95	33.36	1.60
3	8	GELU	sampling	0.00	0.00	0.00	22.39
3	8	tanh	T1	0.34	0.26	0.40	2.30
3	8	tanh	T2-Tr	0.04	0.31	0.29	1.39
3	8	tanh	PTPE	0.04	0.09	0.11	3.09
3	8	tanh	1d-T1	0.34	0.26	0.40	1.54
3	8	tanh	1d-T2-Tr	0.04	0.26	0.22	1.43
3	8	tanh	1d-T2-GC	0.04	0.28	0.22	1.71
3	8	tanh	sampling	0.00	0.00	0.00	20.62
3	8	sigmoid	T1	0.03	0.002	0.04	2.27
3	8	sigmoid	T2-Tr	0.002	0.003	0.03	1.44
3	8	sigmoid	PTPE	9.50	0.06	9.51	3.11
3	8	sigmoid	1d-T1	0.03	0.002	0.04	1.55
3	8	sigmoid	1d-T2-Tr	0.002	0.002	0.02	1.48
3	8	sigmoid	1d-T2-GC	0.001	0.002	0.02	1.66
3	8	sigmoid	sampling	0.00	0.00	0.00	19.63

Table 16: `Wisconsin` Dataset: Moment Propagation in Terms of L_2-norm $||\cdot||_2$, Wasserstein distance W_2, and Runtime T with Varying Number of Layers L, Nonlinear Activation Function, and Method.

L	f_l	activation	method	$\|\|\hat{\mu} - \hat{\mu}_{MC}\|\|_2$	$\|\|\hat{\sum} - \hat{\sum}_{MC}\|\|_F$	$W_2(\mathcal{N}, \mathcal{N}_{MC})$	T (seconds)
2	4	`ReLU`	`T1`	1.86	8.28	2.23	2.64
2	4	`ReLU`	`T2-Tr`	2.88	17.87	3.59	1.79
2	4	`ReLU`	`PTPE`	NaN	NaN	NaN	2.42
2	4	`ReLU`	`1d-T1`	1.86	8.28	2.234	1.87
2	4	`ReLU`	`1d-T2-Tr`	2.88	16.64	3.59	0.76
2	4	`ReLU`	`1d-T2-GC`	2.88	8.04	3.06	0.80
2	4	`ReLU`	sampling	0.00	0.00	0.00	19.94
2	4	`GELU`	`T1`	0.18	0.14	0.19	2.69
2	4	`GELU`	`T2-Tr`	0.01	0.15	0.09	2.09
2	4	`GELU`	`PTPE`	0.02	0.05	0.03	2.40
2	4	`GELU`	`1d-T1`	0.18	0.14	0.19	1.92
2	4	`GELU`	`1d-T2-Tr`	0.01	0.15	0.08	0.79
2	4	`GELU`	`1d-T2-GC`	0.01	0.19	0.06	0.84
2	4	`GELU`	sampling	0.00	0.00	0.00	22.15
2	4	`tanh`	`T1`	23.28	1995.77	54.54	2.12
2	4	`tanh`	`T2-Tr`	178.20	34621.90	183.73	1.16
2	4	`tanh`	`PTPE`	3.34	171.48	11.01	2.39
2	4	`tanh`	`1d-T1`	23.28	1995.74	54.54	0.94
2	4	`tanh`	`1d-T2-Tr`	74.68	6683.33	83.13	0.76
2	4	`tanh`	`1d-T2-GC`	141.72	27030.57	210.48	0.74
2	4	`tanh`	sampling	0.00	0.00	0.00	19.94
2	4	`sigmoid`	`T1`	15.33	1025.08	33.07	2.14
2	4	`sigmoid`	`T2-Tr`	134.67	20955.21	137.83	1.33
2	4	`sigmoid`	`PTPE`	39.37	155.53	41.42	2.67
2	4	`sigmoid`	`1d-T1`	15.33	1025.07	33.07	0.91
2	4	`sigmoid`	`1d-T2-Tr`	16.08	528.86	25.21	0.74
2	4	`sigmoid`	`1d-T2-GC`	34.00	3022.77	60.05	0.75
2	4	`sigmoid`	sampling	0.00	0.00	0.00	20.38
3	4	`ReLU`	`T1`	4.69	67.06	6.32	2.61
3	4	`ReLU`	`T2-Tr`	10.25	158.71	11.62	2.27
3	4	`ReLU`	`PTPE`	NaN	NaN	NaN	2.55
3	4	`ReLU`	`1d-T1`	4.69	67.06	6.32	2.06
3	4	`ReLU`	`1d-T2-Tr`	10.21	137.66	11.79	0.96
3	4	`ReLU`	`1d-T2-GC`	10.20	130.39	12.73	1.02
3	4	`ReLU`	sampling	0.00	0.00	0.00	21.15
3	4	`GELU`	`T1`	94.79	21127.12	162.59	2.67
3	4	`GELU`	`T2-Tr`	50.35	27814.57	156.33	2.23
3	4	`GELU`	`PTPE`	25.08	6014.98	51.09	2.70
3	4	`GELU`	`1d-T1`	94.79	21126.14	162.59	2.03
3	4	`GELU`	`1d-T2-Tr`	82.92	27571.39	170.54	1.01
3	4	`GELU`	`1d-T2-GC`	95.38	28637.76	169.47	1.03
3	4	`GELU`	sampling	0.00	0.00	0.00	24.57
3	4	`tanh`	`T1`	2.40	8.77	3.57	2.25
3	4	`tanh`	`T2-Tr`	1.11	14.89	3.59	1.53
3	4	`tanh`	`PTPE`	0.43	3.58	1.24	2.63
3	4	`tanh`	`1d-T1`	2.40	8.77	3.57	1.17
3	4	`tanh`	`1d-T2-Tr`	1.50	11.29	3.45	0.90
3	4	`tanh`	`1d-T2-GC`	1.92	12.56	3.48	1.15
3	4	`tanh`	sampling	0.00	0.00	0.00	21.99
3	4	`sigmoid`	`T1`	0.02	0.001	0.02	2.22
3	4	`sigmoid`	`T2-Tr`	0.001	0.001	0.02	1.41
3	4	`sigmoid`	`PTPE`	3.63	0.02	3.63	2.82
3	4	`sigmoid`	`1d-T1`	0.02	0.001	0.02	1.08
3	4	`sigmoid`	`1d-T2-Tr`	0.001	0.001	0.003	1.00
3	4	`sigmoid`	`1d-T2-GC`	0.001	0.001	0.005	1.06
3	4	`sigmoid`	sampling	0.00	0.00	0.00	21.66

Table 17: `Texas` Dataset: Moment Propagation in Terms of L_2-norm $||\cdot||_2$, Wasserstein distance W_2, and Runtime T with Varying Number of Layers L, Nonlinear Activation Function, and Method.

L	f_l	activation	method	$\|\|\hat{\mu} - \hat{\mu}_{MC}\|\|_2$	$\|\|\hat{\sum} - \hat{\sum}_{MC}\|\|_F$	$W_2(\mathcal{N}, \mathcal{N}_{MC})$	T (seconds)
2	4	`ReLU`	`T1`	7.45	526.04	10.02	2.60
2	4	`ReLU`	`T2-Tr`	8.41	568.10	13.50	1.80
2	4	`ReLU`	`PTPE`	0.47	NaN	NaN	2.07
2	4	`ReLU`	`1d-T1`	7.45	526.04	10.02	1.53
2	4	`ReLU`	`1d-T2-Tr`	8.30	558.14	13.84	0.42
2	4	`ReLU`	`1d-T2-GC`	8.17	720.87	11.90	0.47
2	4	`ReLU`	sampling	0.00	0.00	0.00	19.04
2	4	`GELU`	`T1`	0.13	0.04	0.14	2.68
2	4	`GELU`	`T2-Tr`	0.01	0.05	0.08	1.79
2	4	`GELU`	`PTPE`	0.02	0.01	0.03	2.07
2	4	`GELU`	`1d-T1`	0.13	0.04	0.14	1.85
2	4	`GELU`	`1d-T2-Tr`	0.01	0.05	0.06	0.42
2	4	`GELU`	`1d-T2-GC`	0.01	0.04	0.04	0.47
2	4	`GELU`	sampling	0.00	0.00	0.00	20.35
2	4	`tanh`	`T1`	0.09	0.09	0.11	2.06
2	4	`tanh`	`T2-Tr`	0.004	0.09	0.07	1.11
2	4	`tanh`	`PTPE`	0.02	0.02	0.02	2.31
2	4	`tanh`	`1d-T1`	0.09	0.09	0.11	0.67
2	4	`tanh`	`1d-T2-Tr`	0.005	0.08	0.06	0.46
2	4	`tanh`	`1d-T2-GC`	0.01	0.09	0.06	0.43
2	4	`tanh`	sampling	0.00	0.00	0.00	18.71
2	4	`sigmoid`	`T1`	0.02	0.001	0.02	2.34
2	4	`sigmoid`	`T2-Tr`	0.0003	0.001	0.03	1.05
2	4	`sigmoid`	`PTPE`	5.44	0.02	5.44	2.35
2	4	`sigmoid`	`1d-T1`	0.02	0.001	0.02	0.64
2	4	`sigmoid`	`1d-T2-Tr`	0.0003	0.001	0.01	0.42
2	4	`sigmoid`	`1d-T2-GC`	0.0003	0.001	0.01	0.44
2	4	`sigmoid`	sampling	0.00	0.00	0.00	19.55
3	4	`ReLU`	`T1`	1.09	0.68	1.17	2.67
3	4	`ReLU`	`T2-Tr`	1.17	0.60	1.25	1.78
3	4	`ReLU`	`PTPE`	NaN	NaN	NaN	2.12
3	4	`ReLU`	`1d-T1`	1.09	0.68	1.17	1.66
3	4	`ReLU`	`1d-T2-Tr`	1.15	0.67	1.25	0.53
3	4	`ReLU`	`1d-T2-GC`	1.15	0.92	1.24	0.58
3	4	`ReLU`	sampling	0.00	0.00	0.00	19.42
3	4	`GELU`	`T1`	7.32	277.57	11.73	2.75
3	4	`GELU`	`T2-Tr`	5.67	302.98	11.14	1.90
3	4	`GELU`	`PTPE`	0.31	41.38	3.11	2.17
3	4	`GELU`	`1d-T1`	7.32	277.56	11.73	1.76
3	4	`GELU`	`1d-T2-Tr`	6.57	305.02	11.75	0.54
3	4	`GELU`	`1d-T2-GC`	6.83	967.78	22.53	0.61
3	4	`GELU`	sampling	0.00	0.00	0.00	21.96
3	4	`tanh`	`T1`	2.80	66.49	5.36	2.09
3	4	`tanh`	`T2-Tr`	1.38	68.48	4.90	1.20
3	4	`tanh`	`PTPE`	0.31	7.07	1.03	2.37
3	4	`tanh`	`1d-T1`	2.80	66.49	5.36	0.76
3	4	`tanh`	`1d-T2-Tr`	1.33	70.35	4.95	0.54
3	4	`tanh`	`1d-T2-GC`	1.34	68.06	5.27	0.66
3	4	`tanh`	sampling	0.00	0.00	0.00	19.60
3	4	`sigmoid`	`T1`	6.65	131.54	9.79	2.12
3	4	`sigmoid`	`T2-Tr`	14.73	303.67	17.73	1.10
3	4	`sigmoid`	`PTPE`	34.30	110.13	35.58	2.40
3	4	`sigmoid`	`1d-T1`	6.65	131.54	9.79	0.72
3	4	`sigmoid`	`1d-T2-Tr`	3.99	103.11	8.96	0.57
3	4	`sigmoid`	`1d-T2-GC`	19.33	2070.56	43.05	0.68
3	4	`sigmoid`	sampling	0.00	0.00	0.00	19.46

Table 18: `Cora` Dataset: Moment Propagation in Terms of L_2-norm $||\cdot||_2$, Wasserstein distance W_2, and Runtime T with Varying Number of Layers L, Nonlinear Activation Function, and Method.

L	f_l	activation	method	$\|\|\hat{\mu} - \hat{\mu}_{MC}\|\|_2$	$\|\|\hat{\sum} - \hat{\sum}_{MC}\|\|_F$	$W_2(\mathcal{N}, \mathcal{N}_{MC})$	T (seconds)
2	8	`ReLU`	`T1`	2.57	0.86	2.63	37.65
2	8	`ReLU`	`T2-Tr`	2.96	1.58	3.03	262.13
2	8	`ReLU`	`PTPE`	0.01	0.37	0.07	1680.84
2	8	`ReLU`	`1d-T1`	2.57	0.86	2.63	2559.09
2	8	`ReLU`	`1d-T2-Tr`	2.96	1.46	3.03	2567.68
2	8	`ReLU`	`1d-T2-GC`	2.96	0.84	3.004	2596.62
2	8	`ReLU`	sampling	0.00	0.00	0.00	950.29
2	8	`GELU`	`T1`	2.48	4.27	2.60	39.31
2	8	`GELU`	`T2-Tr`	0.23	8.45	1.17	268.13
2	8	`GELU`	`PTPE`	0.26	2.13	0.31	1718.57
2	8	`GELU`	`1d-T1`	2.48	4.26	2.60	2596.36
2	8	`GELU`	`1d-T2-Tr`	0.22	7.56	1.11	2603.03
2	8	`GELU`	`1d-T2-GC`	0.22	5.52	0.90	2618.94
2	8	`GELU`	sampling	0.00	0.00	0.00	972.65
2	8	`tanh`	`T1`	3.44	10.33	3.94	27.29
2	8	`tanh`	`T2-Tr`	0.76	21.02	2.60	234.01
2	8	`tanh`	`PTPE`	0.24	2.41	0.40	1720.36
2	8	`tanh`	`1d-T1`	3.44	10.32	3.94	2597.27
2	8	`tanh`	`1d-T2-Tr`	0.73	20.06	2.16	2602.33
2	8	`tanh`	`1d-T2-GC`	0.73	14.63	2.08	2618.38
2	8	`tanh`	sampling	0.00	0.00	0.00	968.20
2	8	`sigmoid`	`T1`	7.19	62.37	9.01	27.34
2	8	`sigmoid`	`T2-Tr`	2.14	91.70	6.10	234.22
2	8	`sigmoid`	`PTPE`	305.03	122.96	305.73	1721.21
2	8	`sigmoid`	`1d-T1`	7.19	62.34	9.01	2584.22
2	8	`sigmoid`	`1d-T2-Tr`	2.00	80.24	5.37	2562.52
2	8	`sigmoid`	`1d-T2-GC`	2.01	57.32	6.23	2576.11
2	8	`sigmoid`	sampling	0.00	0.00	0.00	967.34
3	8	`ReLU`	`T1`	5.30	3.59	5.40	47.23
3	8	`ReLU`	`T2-Tr`	5.93	4.99	6.06	297.10
3	8	`ReLU`	`PTPE`	0.04	1.30	0.17	2653.37
3	8	`ReLU`	`1d-T1`	5.30	3.59	5.40	3466.59
3	8	`ReLU`	`1d-T2-Tr`	5.93	4.35	6.07	3477.16
3	8	`ReLU`	`1d-T2-GC`	5.93	3.66	6.03	3494.04
3	8	`ReLU`	sampling	0.00	0.00	0.00	891.58
3	8	`GELU`	`T1`	4.36	56.20	5.56	39.76
3	8	`GELU`	`T2-Tr`	2.20	61.96	4.57	226.72
3	8	`GELU`	`PTPE`	0.27	7.47	0.61	1403.80
3	8	`GELU`	`1d-T1`	4.36	56.18	5.56	1825.33
3	8	`GELU`	`1d-T2-Tr`	2.09	59.30	4.48	1834.27
3	8	`GELU`	`1d-T2-GC`	2.13	67.18	4.44	1848.42
3	8	`GELU`	sampling	0.00	0.00	0.00	758.15
3	8	`tanh`	`T1`	6.77	62.25	8.88	25.30
3	8	`tanh`	`T2-Tr`	2.54	95.78	7.20	185.30
3	8	`tanh`	`PTPE`	0.61	8.76	1.30	1422.48
3	8	`tanh`	`1d-T1`	6.77	62.22	8.87	1855.49
3	8	`tanh`	`1d-T2-Tr`	2.68	88.62	6.41	1862.23
3	8	`tanh`	`1d-T2-GC`	2.67	65.06	7.01	1873.40
3	8	`tanh`	sampling	0.00	0.00	0.00	756.95
3	8	`sigmoid`	`T1`	26.58	578.80	37.01	25.32
3	8	`sigmoid`	`T2-Tr`	28.97	2358.61	37.73	185.22
3	8	`sigmoid`	`PTPE`	394.08	577.48	397.29	1419.23
3	8	`sigmoid`	`1d-T1`	26.58	578.66	37.003	1849.25
3	8	`sigmoid`	`1d-T2-Tr`	26.43	1937.33	32.88	1861.18
3	8	`sigmoid`	`1d-T2-GC`	30.17	1949.68	54.60	1873.34
3	8	`sigmoid`	sampling	0.00	0.00	0.00	756.14

Table 19: `Citeseer` Dataset: Moment Propagation in Terms of L_2-norm $||\cdot||_2$, Wasserstein distance W_2, and Runtime T with Varying Number of Layers L, Nonlinear Activation Function, and Method.

L	f_l	activation	method	$\|\|\hat{\mu} - \hat{\mu}_{MC}\|\|_2$	$\|\|\hat{\sum} - \hat{\sum}_{MC}\|\|_F$	$W_2(\mathcal{N}, \mathcal{N}_{MC})$	T (seconds)
2	4	ReLU	T1	2.61	0.43	2.67	45.89
2	4	ReLU	T2-Tr	2.92	0.49	2.98	292.71
2	4	ReLU	PTPE	0.02	NaN	0.02	886.60
2	4	ReLU	1d-T1	2.61	0.43	2.67	1061.58
2	4	ReLU	1d-T2-Tr	2.92	0.46	2.98	1061.40
2	4	ReLU	1d-T2-GC	2.92	0.43	2.97	1065.85
2	4	ReLU	sampling	0.00	0.00	0.00	984.36
2	4	GELU	T1	0.26	0.06	0.26	46.87
2	4	GELU	T2-Tr	0.003	0.09	0.04	293.31
2	4	GELU	PTPE	0.11	0.05	0.11	886.29
2	4	GELU	1d-T1	0.26	0.06	0.26	1060.91
2	4	GELU	1d-T2-Tr	0.003	0.09	0.04	1061.34
2	4	GELU	1d-T2-GC	0.003	0.07	0.04	1065.79
2	4	GELU	sampling	0.00	0.00	0.00	986.34
2	4	tanh	T1	0.33	0.09	0.34	38.57
2	4	tanh	T2-Tr	0.004	0.15	0.10	270.09
2	4	tanh	PTPE	0.15	0.06	0.15	886.40
2	4	tanh	1d-T1	0.33	0.09	0.34	1059.04
2	4	tanh	1d-T2-Tr	0.004	0.13	0.09	1062.53
2	4	tanh	1d-T2-GC	0.01	0.10	0.09	1065.66
2	4	tanh	sampling	0.00	0.00	0.00	985.40
2	4	sigmoid	T1	0.26	0.08	0.26	38.60
2	4	sigmoid	T2-Tr	0.004	0.11	0.06	272.92
2	4	sigmoid	PTPE	96.50	1.61	96.54	886.97
2	4	sigmoid	1d-T1	0.26	0.08	0.26	1059.69
2	4	sigmoid	1d-T2-Tr	0.003	0.10	0.06	1061.85
2	4	sigmoid	1d-T2-GC	0.004	0.08	0.06	1066.57
2	4	sigmoid	sampling	0.00	0.00	0.00	984.33
3	4	ReLU	T1	2.87	0.35	2.90	40.35
3	4	ReLU	T2-Tr	3.04	0.37	3.07	228.24
3	4	ReLU	PTPE	NaN	NaN	NaN	643.43
3	4	ReLU	1d-T1	2.87	0.35	2.90	729.36
3	4	ReLU	1d-T2-Tr	3.04	0.37	3.07	739.41
3	4	ReLU	1d-T2-GC	3.04	0.35	3.07	736.34
3	4	ReLU	sampling	0.00	0.00	0.00	836.34
3	4	GELU	T1	0.18	0.04	0.18	41.55
3	4	GELU	T2-Tr	0.003	0.05	0.03	232.61
3	4	GELU	PTPE	0.07	0.03	0.07	646.93
3	4	GELU	1d-T1	0.18	0.04	0.18	735.88
3	4	GELU	1d-T2-Tr	0.002	0.04	0.03	733.48
3	4	GELU	1d-T2-GC	0.003	0.04	0.03	733.47
3	4	GELU	sampling	0.00	0.00	0.00	838.56
3	4	tanh	T1	0.26	0.07	0.26	30.78
3	4	tanh	T2-Tr	0.01	0.10	0.07	203.65
3	4	tanh	PTPE	0.20	0.04	0.20	652.74
3	4	tanh	1d-T1	0.26	0.07	0.26	729.68
3	4	tanh	1d-T2-Tr	0.01	0.07	0.05	748.63
3	4	tanh	1d-T2-GC	0.01	0.06	0.05	748.57
3	4	tanh	sampling	0.00	0.00	0.00	837.24
3	4	sigmoid	T1	1.57	2.96	1.75	30.87
3	4	sigmoid	T2-Tr	0.25	4.27	0.94	200.55
3	4	sigmoid	PTPE	214.54	18.63	214.80	644.87
3	4	sigmoid	1d-T1	1.57	2.96	1.75	724.48
3	4	sigmoid	1d-T2-Tr	0.13	3.95	0.76	729.56
3	4	sigmoid	1d-T2-GC	0.15	3.07	0.82	736.92
3	4	sigmoid	sampling	0.00	0.00	0.00	840.61

Table 20: `Chameleon` Dataset: Moment Propagation in Terms of L_2-norm $||\cdot||_2$, Wasserstein distance W_2, and Runtime T with Varying Number of Layers L, Nonlinear Activation Function, and Method.

L	f_l	activation	method	$\|\|\hat{\mu} - \hat{\mu}_{MC}\|\|_2$	$\|\|\hat{\sum} - \hat{\sum}_{MC}\|\|_F$	$W_2(\mathcal{N}, \mathcal{N}_{MC})$	T (seconds)
2	8	`ReLU`	`T1`	26.97	43.60	27.06	45.44
2	8	`ReLU`	`T2-Tr`	29.09	50.18	29.20	245.65
2	8	`ReLU`	`PTPE`	0.13	11.95	0.50	947.64
2	8	`ReLU`	`1d-T1`	26.97	43.73	27.06	1346.22
2	8	`ReLU`	`1d-T2-Tr`	29.09	49.41	29.20	1350.01
2	8	`ReLU`	`1d-T2-GC`	29.09	113.70	29.24	1358.06
2	8	`ReLU`	sampling	0.00	0.00	0.00	346.94
2	8	`GELU`	`T1`	6.96	152.06	9.11	46.23
2	8	`GELU`	`T2-Tr`	2.76	171.76	5.81	245.09
2	8	`GELU`	`PTPE`	0.71	13.48	0.85	947.21
2	8	`GELU`	`1d-T1`	6.96	152.06	9.11	1345.35
2	8	`GELU`	`1d-T2-Tr`	3.04	175.68	5.95	1349.08
2	8	`GELU`	`1d-T2-GC`	3.07	184.62	6.59	1357.35
2	8	`GELU`	sampling	0.00	0.00	0.00	351.57
2	8	`tanh`	`T1`	7.16	161.07	11.34	24.84
2	8	`tanh`	`T2-Tr`	2.97	172.54	9.38	236.003
2	8	`tanh`	`PTPE`	0.44	13.24	1.12	948.91
2	8	`tanh`	`1d-T1`	7.16	160.32	11.32	1344.91
2	8	`tanh`	`1d-T2-Tr`	3.68	158.15	9.06	1349.40
2	8	`tanh`	`1d-T2-GC`	3.69	77.80	6.91	1357.62
2	8	`tanh`	sampling	0.00	0.00	0.00	347.31
2	8	`sigmoid`	`T1`	5.34	36.88	6.03	24.85
2	8	`sigmoid`	`T2-Tr`	1.83	57.16	4.30	235.99
2	8	`sigmoid`	`PTPE`	186.44	255.76	187.01	948.15
2	8	`sigmoid`	`1d-T1`	5.34	36.56	6.03	1346.12
2	8	`sigmoid`	`1d-T2-Tr`	1.59	44.87	3.86	1349.63
2	8	`sigmoid`	`1d-T2-GC`	1.94	44.78	4.22	1357.78
2	8	`sigmoid`	sampling	0.00	0.00	0.00	347.20
3	8	`ReLU`	`T1`	67.66	77.78	67.83	66.79
3	8	`ReLU`	`T2-Tr`	68.65	81.27	68.83	331.64
3	8	`ReLU`	`PTPE`	0.07	7.84	0.53	1573.37
3	8	`ReLU`	`1d-T1`	67.66	77.77	67.83	1970.13
3	8	`ReLU`	`1d-T2-Tr`	68.65	81.45	68.84	1977.55
3	8	`ReLU`	`1d-T2-GC`	68.65	77.27	68.82	1990.30
3	8	`ReLU`	sampling	0.00	0.00	0.00	353.37
3	8	`GELU`	`T1`	12.42	1259.12	18.01	67.32
3	8	`GELU`	`T2-Tr`	7.93	1293.84	16.49	331.27
3	8	`GELU`	`PTPE`	0.87	51.58	1.68	1573.54
3	8	`GELU`	`1d-T1`	12.42	1258.17	18.004	1969.13
3	8	`GELU`	`1d-T2-Tr`	7.79	1282.83	16.15	1977.88
3	8	`GELU`	`1d-T2-GC`	7.81	1414.02	14.70	1988.42
3	8	`GELU`	sampling	0.00	0.00	0.00	357.63
3	8	`tanh`	`T1`	7.07	58.94	7.65	43.14
3	8	`tanh`	`T2-Tr`	2.12	108.40	5.84	309.53
3	8	`tanh`	`PTPE`	0.79	6.27	0.93	1573.44
3	8	`tanh`	`1d-T1`	7.07	58.69	7.65	1969.62
3	8	`tanh`	`1d-T2-Tr`	2.18	105.62	5.59	1976.53
3	8	`tanh`	`1d-T2-GC`	2.18	71.38	4.10	1988.27
3	8	`tanh`	sampling	0.00	0.00	0.00	354.11
3	8	`sigmoid`	`T1`	4.27	100.33	6.68	43.18
3	8	`sigmoid`	`T2-Tr`	2.67	107.08	5.91	309.68
3	8	`sigmoid`	`PTPE`	210.05	223.86	210.82	1575.11
3	8	`sigmoid`	`1d-T1`	4.27	98.30	6.62	1968.94
3	8	`sigmoid`	`1d-T2-Tr`	2.82	65.45	3.89	1977.15
3	8	`sigmoid`	`1d-T2-GC`	4.07	136.32	5.21	1988.98
3	8	`sigmoid`	sampling	0.00	0.00	0.00	354.03

Table 21: `Squirrel` Dataset: Moment Propagation in Terms of L_2-norm $||\cdot||_2$, Wasserstein distance W_2, and Runtime T with Varying Number of Layers L, Nonlinear Activation Function, and Method.

L	f_l	activation	method	$\|\|\hat{\mu} - \hat{\mu}_{MC}\|\|_2$	$\|\|\hat{\sum} - \hat{\sum}_{MC}\|\|_F$	$W_2(\mathcal{N}, \mathcal{N}_{MC})$	T (seconds)
2	4	`ReLU`	`T1`	3.90	0.50	3.93	384.89
2	4	`ReLU`	`T2-Tr`	3.98	0.50	4.01	1969.39
2	4	`ReLU`	`PTPE`	0.02	0.05	0.04	3313.58
2	4	`ReLU`	`1d-T1`	3.90	0.50	3.93	3749.34
2	4	`ReLU`	`1d-T2-Tr`	3.98	0.51	4.01	3760.08
2	4	`ReLU`	`1d-T2-GC`	3.98	0.50	4.01	3770.32
2	4	`ReLU`	sampling	0.00	0.00	0.00	1743.86
2	4	`GELU`	`T1`	0.40	0.74	0.42	384.67
2	4	`GELU`	`T2-Tr`	0.02	0.77	0.14	1957.38
2	4	`GELU`	`PTPE`	0.24	0.15	0.24	3312.75
2	4	`GELU`	`1d-T1`	0.40	0.74	0.42	3749.33
2	4	`GELU`	`1d-T2-Tr`	0.02	0.78	0.13	3753.40
2	4	`GELU`	`1d-T2-GC`	0.02	1.10	0.17	3765.10
2	4	`GELU`	sampling	0.00	0.00	0.00	1744.38
2	4	`tanh`	`T1`	3.46	27.69	5.14	257.53
2	4	`tanh`	`T2-Tr`	5.29	65.07	6.04	2048.23
2	4	`tanh`	`PTPE`	0.37	2.82	0.64	3319.05
2	4	`tanh`	`1d-T1`	3.46	27.66	5.14	3754.48
2	4	`tanh`	`1d-T2-Tr`	5.08	63.12	5.93	3756.05
2	4	`tanh`	`1d-T2-GC`	5.08	74.24	9.17	3767.23
2	4	`tanh`	sampling	0.00	0.00	0.00	1744.30
2	4	`sigmoid`	`T1`	2.07	4.03	2.80	268.29
2	4	`sigmoid`	`T2-Tr`	0.82	7.15	2.41	2134.51
2	4	`sigmoid`	`PTPE`	43.33	11.97	43.56	3333.05
2	4	`sigmoid`	`1d-T1`	2.07	4.04	2.80	3751.95
2	4	`sigmoid`	`1d-T2-Tr`	0.77	5.32	2.08	3815.69
2	4	`sigmoid`	`1d-T2-GC`	0.95	5.23	1.96	3844.56
2	4	`sigmoid`	sampling	0.00	0.00	0.00	1743.07
3	4	`ReLU`	`T1`	20.46	24.99	20.78	507.09
3	4	`ReLU`	`T2-Tr`	20.52	21.92	20.79	2463.28
3	4	`ReLU`	`PTPE`	NaN	NaN	NaN	4283.26
3	4	`ReLU`	`1d-T1`	20.46	25.00	20.78	4715.82
3	4	`ReLU`	`1d-T2-Tr`	20.55	14.43	20.76	4722.99
3	4	`ReLU`	`1d-T2-GC`	20.55	18.23	20.81	4737.29
3	4	`ReLU`	sampling	0.00	0.00	0.00	1768.75
3	4	`GELU`	`T1`	0.36	0.17	0.36	508.64
3	4	`GELU`	`T2-Tr`	0.01	0.22	0.08	2470.05
3	4	`GELU`	`PTPE`	0.17	0.11	0.17	4279.99
3	4	`GELU`	`1d-T1`	0.36	0.17	0.36	4714.39
3	4	`GELU`	`1d-T2-Tr`	0.01	0.19	0.09	4723.64
3	4	`GELU`	`1d-T2-GC`	0.01	0.32	0.06	4739.72
3	4	`GELU`	sampling	0.00	0.00	0.00	1767.51
3	4	`tanh`	`T1`	4.78	88.01	8.42	375.59
3	4	`tanh`	`T2-Tr`	7.64	88.12	8.19	2505.17
3	4	`tanh`	`PTPE`	0.74	8.18	1.39	4285.40
3	4	`tanh`	`1d-T1`	4.78	88.00	8.42	4719.42
3	4	`tanh`	`1d-T2-Tr`	3.06	114.68	8.21	4722.58
3	4	`tanh`	`1d-T2-GC`	3.10	185.11	11.31	4738.42
3	4	`tanh`	sampling	0.00	0.00	0.00	1764.62
3	4	`sigmoid`	`T1`	4.89	79.50	8.99	524.99
3	4	`sigmoid`	`T2-Tr`	9.37	145.30	10.90	3193.72
3	4	`sigmoid`	`PTPE`	124.63	68.63	125.09	4360.26
3	4	`sigmoid`	`1d-T1`	4.89	79.47	8.99	4708.11
3	4	`sigmoid`	`1d-T2-Tr`	5.12	101.86	6.74	4720.08
3	4	`sigmoid`	`1d-T2-GC`	13.32	509.08	17.92	4733.75
3	4	`sigmoid`	sampling	0.00	0.00	0.00	1756.82

Table 22: `Cornell` Dataset: Certified Adversarial Robustness Radii. L is the number of layers, f_l is the number of hidden dimensions, p_{CERT}^{Thm1} and p_{CERT}^{Chn} are the percentage of nodes for which a positive radius could be certified with Theorem 1 (Thm1) in our manuscript and Cohen's method (Chn) respectively, ϵ_{CERT}^{Thm1} and ϵ_{CERT}^{Chn} are the average robustness radius among nodes with a positive robustness radius using Theorem 1 or Cohen's method respectively, T^{Thm1} and T^{Chn} the runtime for computing radii using Theorem 1 or Cohen's method, for the former we include the time it takes to estimate the moments. Time is reported in seconds.

L	f_l	activation	method	p_{CERT}^{Thm1}	p_{CERT}^{Chn}	ϵ_{CERT}^{Thm1}	ϵ_{CERT}^{Chn}	T^{Thm1}	T^{Chn}	$\bar{C}$
2	4	`ReLU`	sample	0.85	0.99	0.0053	0.1845	20.09	588.74	188
2	4	`GELU`	sample	0.51	1.00	0.0030	0.1677	21.73	593.24	392
2	4	`tanh`	sample	0.48	0.97	0.0005	0.1595	21.67	581.17	3091
2	4	`sigmoid`	sample	0.90	1.00	0.0042	0.1888	20.11	568.81	107
2	4	`ReLU`	`PTPE`	0.00	0.99	NaN	0.1845	3.26	588.74	188
2	4	`GELU`	`PTPE`	0.51	1.00	0.0030	0.1677	3.24	593.24	392
2	4	`tanh`	`PTPE`	0.48	0.97	0.0005	0.1595	3.45	581.17	3091
2	4	`sigmoid`	`PTPE`	0.90	1.00	0.0046	0.1888	3.45	568.81	107
2	4	`ReLU`	`1d-T2-GC`	0.77	0.99	0.0054	0.1845	1.60	588.74	188
2	4	`GELU`	`1d-T2-GC`	0.51	1.00	0.0030	0.1677	1.51	593.24	392
2	4	`tanh`	`1d-T2-GC`	0.44	0.97	0.0005	0.1595	1.59	581.17	3091
2	4	`sigmoid`	`1d-T2-GC`	0.87	1.00	0.0042	0.1888	1.57	568.81	107
2	4	`ReLU`	`T1`	0.86	0.99	0.0052	0.1845	3.74	588.74	188
2	4	`GELU`	`T1`	0.51	1.00	0.0030	0.1677	3.77	593.24	392
2	4	`tanh`	`T1`	0.48	0.97	0.0005	0.1595	3.26	581.17	3091
2	4	`sigmoid`	`T1`	0.90	1.00	0.0042	0.1888	3.18	568.81	107
3	8	`ReLU`	sample	0.28	1.00	0.0011	0.1426	21.60	565.81	7489
3	8	`GELU`	sample	0.23	0.98	0.0003	0.1428	24.12	590.97	12192
3	8	`tanh`	sample	0.48	1.00	0.0006	0.1692	22.40	563.48	1009
3	8	`sigmoid`	sample	0.93	1.00	0.0042	0.1868	21.35	564.62	120
3	8	`ReLU`	`PTPE`	0.28	1.00	0.0011	0.1426	4.02	565.81	7489
3	8	`GELU`	`PTPE`	0.24	0.98	0.0003	0.1428	4.15	590.97	12192
3	8	`tanh`	`PTPE`	0.48	1.00	0.0006	0.1692	4.25	563.48	1009
3	8	`sigmoid`	`PTPE`	0.78	1.00	0.0055	0.1868	4.24	564.62	120
3	8	`ReLU`	`1d-T2-GC`	0.28	1.00	0.0011	0.1426	2.74	565.81	7489
3	8	`GELU`	`1d-T2-GC`	0.25	0.98	0.0003	0.1428	2.74	590.97	12192
3	8	`tanh`	`1d-T2-GC`	0.48	1.00	0.0006	0.1692	2.89	563.48	1009
3	8	`sigmoid`	`1d-T2-GC`	0.90	1.00	0.0042	0.1868	2.79	564.62	120
3	8	`ReLU`	`T1`	0.28	1.00	0.0012	0.1426	3.80	565.81	7489
3	8	`GELU`	`T1`	0.24	0.98	0.0003	0.1428	3.96	590.97	12192
3	8	`tanh`	`T1`	0.48	1.00	0.0006	0.1692	3.49	563.48	1009
3	8	`sigmoid`	`T1`	0.93	1.00	0.0042	0.1868	3.39	564.62	120

Table 23: `Wisconsin` Dataset: Certified Adversarial Robustness Radii. L is the number of layers, f_l is the number of hidden dimensions, p_{CERT}^{Thm1} and p_{CERT}^{Chn} are the percentage of nodes for which a positive radius could be certified with Theorem 1 (Thm1) in our manuscript and Cohen's method (Chn) respectively, ϵ_{CERT}^{Thm1} and ϵ_{CERT}^{Chn} are the average robustness radius among nodes with a positive robustness radius using Theorem 1 or Cohen's method respectively, T^{Thm1} and T^{Chn} the runtime for computing radii using Theorem 1 or Cohen's method, for the former we include the time it takes to estimate the moments. Time is reported in seconds.

L	f_l	activation	method	p_{CERT}^{Thm1}	p_{CERT}^{Chn}	ϵ_{CERT}^{Thm1}	ϵ_{CERT}^{Chn}	T^{Thm1}	T^{Chn}	$\bar{C}$
2	4	`ReLU`	sample	0.52	1.00	0.0005	0.1917	21.81	562.23	1292
2	4	`GELU`	sample	0.84	0.99	0.0028	0.2044	24.09	580.86	169
2	4	`tanh`	sample	0.50	0.98	0.0000	0.1749	22.07	556.01	31269
2	4	`sigmoid`	sample	0.59	0.97	0.0001	0.1728	22.38	569.72	17104
2	4	`ReLU`	`PTPE`	0.00	1.00	NaN	0.1917	3.66	562.23	1292
2	4	`GELU`	`PTPE`	0.84	0.99	0.0028	0.2044	3.63	580.86	169
2	4	`tanh`	`PTPE`	0.52	0.98	0.0000	0.1749	3.85	556.01	31269
2	4	`sigmoid`	`PTPE`	0.54	0.97	0.0001	0.1728	4.06	569.72	17104
2	4	`ReLU`	`1d-T2-GC`	0.35	1.00	0.0005	0.1917	2.05	562.23	1292
2	4	`GELU`	`1d-T2-GC`	0.84	0.99	0.0028	0.2044	2.08	580.86	169
2	4	`tanh`	`1d-T2-GC`	0.42	0.98	0.0001	0.1749	2.20	556.01	31269
2	4	`sigmoid`	`1d-T2-GC`	0.45	0.97	0.0001	0.1728	2.14	569.72	17104
2	4	`ReLU`	`T1`	0.52	1.00	0.0006	0.1917	3.88	562.23	1292
2	4	`GELU`	`T1`	0.84	0.99	0.0028	0.2044	3.94	580.86	169
2	4	`tanh`	`T1`	0.60	0.98	0.0001	0.1749	3.55	556.01	31269
2	4	`sigmoid`	`T1`	0.63	0.97	0.0001	0.1728	3.55	569.72	17104
3	4	`ReLU`	sample	0.33	1.00	0.0001	0.1619	23.11	594.57	16622
3	4	`GELU`	sample	0.19	0.93	0.00001	0.1256	26.73	594.60	556421
3	4	`tanh`	sample	0.42	1.00	0.0002	0.1758	23.90	582.89	5339
3	4	`sigmoid`	sample	1.00	1.00	0.0135	0.2076	23.64	584.04	40
3	4	`ReLU`	`PTPE`	0.00	1.00	NaN	0.1619	3.95	594.57	16622
3	4	`GELU`	`PTPE`	0.19	0.93	0.00001	0.1256	4.15	594.60	556421
3	4	`tanh`	`PTPE`	0.43	1.00	0.0002	0.1758	4.07	582.89	5339
3	4	`sigmoid`	`PTPE`	1.00	1.00	0.0143	0.2076	4.09	584.04	40
3	4	`ReLU`	`1d-T2-GC`	0.34	1.00	0.0001	0.1619	2.39	594.57	16622
3	4	`GELU`	`1d-T2-GC`	0.22	0.93	0.00001	0.1256	2.49	594.60	556421
3	4	`tanh`	`1d-T2-GC`	0.44	1.00	0.0002	0.1758	2.58	582.89	5339
3	4	`sigmoid`	`1d-T2-GC`	0.94	1.00	0.0135	0.2076	2.33	584.04	40
3	4	`ReLU`	`T1`	0.36	1.00	0.0001	0.1619	3.99	594.57	16622
3	4	`GELU`	`T1`	0.25	0.93	0.00001	0.1256	4.14	594.60	556421
3	4	`tanh`	`T1`	0.47	1.00	0.0002	0.1758	3.65	582.89	5339
3	4	`sigmoid`	`T1`	1.00	1.00	0.0134	0.2076	3.55	584.04	40

Table 24: `Texas` Dataset: Certified Adversarial Robustness Radii. L is the number of layers, f_l is the number of hidden dimensions, p_{CERT}^{Thm1} and p_{CERT}^{Chn} are the percentage of nodes for which a positive radius could be certified with Theorem 1 (Thm1) in our manuscript and Cohen's method (Chn) respectively, ϵ_{CERT}^{Thm1} and ϵ_{CERT}^{Chn} are the average robustness radius among nodes with a positive robustness radius using Theorem 1 or Cohen's method respectively, T^{Thm1} and T^{Chn} the runtime for computing radii using Theorem 1 or Cohen's method, for the former we include the time it takes to estimate the moments. Time is reported in seconds.

L	f_l	activation	method	p_{CERT}^{Thm1}	p_{CERT}^{Chn}	ϵ_{CERT}^{Thm1}	ϵ_{CERT}^{Chn}	T^{Thm1}	T^{Chn}	$\bar{C}$
2	4	ReLU	sample	0.50	0.99	0.0022	0.1434	20.89	572.29	2553
2	4	GELU	sample	0.58	1.00	0.0026	0.1393	22.12	570.39	131
2	4	tanh	sample	0.69	1.00	0.0092	0.1474	20.48	553.35	84
2	4	sigmoid	sample	0.85	1.00	0.0172	0.1603	21.28	556.76	19
2	4	ReLU	PTPE	0.00	0.99	NaN	0.1434	3.33	572.29	2553
2	4	GELU	PTPE	0.58	1.00	0.0026	0.1393	3.30	570.39	131
2	4	tanh	PTPE	0.69	1.00	0.0092	0.1474	3.48	553.35	84
2	4	sigmoid	PTPE	0.86	1.00	0.0151	0.1603	3.48	556.76	19
2	4	ReLU	1d-T2-GC	0.51	0.99	0.0022	0.1434	1.70	572.29	2553
2	4	GELU	1d-T2-GC	0.58	1.00	0.0026	0.1393	1.71	570.39	131
2	4	tanh	1d-T2-GC	0.69	1.00	0.0093	0.1474	1.60	553.35	84
2	4	sigmoid	1d-T2-GC	0.85	1.00	0.0173	0.1603	1.58	556.76	19
2	4	ReLU	T1	0.51	0.99	0.0022	0.1434	3.85	572.29	2553
2	4	GELU	T1	0.58	1.00	0.0026	0.1393	3.89	570.39	131
2	4	tanh	T1	0.69	1.00	0.0093	0.1474	3.26	553.35	84
2	4	sigmoid	T1	0.85	1.00	0.0173	0.1603	3.48	556.76	19
3	4	ReLU	sample	0.73	1.00	0.0014	0.1556	21.16	566.97	692
3	4	GELU	sample	0.46	1.00	0.0003	0.1436	23.74	574.26	13731
3	4	tanh	sample	0.43	0.99	0.0005	0.1347	21.39	561.55	5755
3	4	sigmoid	sample	0.71	1.00	0.0003	0.1444	21.36	570.01	7666
3	4	ReLU	PTPE	0.00	1.00	NaN	0.1556	3.27	566.97	692
3	4	GELU	PTPE	0.45	1.00	0.0003	0.1436	3.35	574.26	13731
3	4	tanh	PTPE	0.44	0.99	0.0005	0.1347	3.56	561.55	5755
3	4	sigmoid	PTPE	0.71	1.00	0.0002	0.1444	3.57	570.01	7666
3	4	ReLU	1d-T2-GC	0.69	1.00	0.0013	0.1556	1.73	566.97	692
3	4	GELU	1d-T2-GC	0.46	1.00	0.0003	0.1436	1.81	574.26	13731
3	4	tanh	1d-T2-GC	0.43	0.99	0.0005	0.1347	1.85	561.55	5755
3	4	sigmoid	1d-T2-GC	0.65	1.00	0.0003	0.1444	1.86	570.01	7666
3	4	ReLU	T1	0.74	1.00	0.0014	0.1556	3.82	566.97	692
3	4	GELU	T1	0.46	1.00	0.0003	0.1436	3.94	574.26	13731
3	4	tanh	T1	0.43	0.99	0.0005	0.1347	3.28	561.55	5755
3	4	sigmoid	T1	0.73	1.00	0.0003	0.1444	3.30	570.01	7666

Table 25: `Cora` Dataset: Certified Adversarial Robustness Radii. L is the number of layers, f_l is the number of hidden dimensions, p^{Thm1}_{CERT} and p^{Chn}_{CERT} are the percentage of nodes for which a positive radius could be certified with Theorem 1 (Thm1) in our manuscript and Cohen's method (Chn) respectively, ϵ^{Thm1}_{CERT} and ϵ^{Chn}_{CERT} are the average robustness radius among nodes with a positive robustness radius using Theorem 1 or Cohen's method respectively, T^{Thm1} and T^{Chn} the runtime for computing radii using Theorem 1 or Cohen's method, for the former we include the time it takes to estimate the moments. Time is reported in seconds.

L	f_l	activation	method	p^{Thm1}_{CERT}	p^{Chn}_{CERT}	ϵ^{Thm1}_{CERT}	ϵ^{Chn}_{CERT}	T^{Thm1}	T^{Chn}	$\bar{C}$
2	8	`ReLU`	sample	0.76	1.00	0.0057	0.0672	959.62	581.58	314
2	8	`GELU`	sample	0.65	1.00	0.0018	0.0663	982.21	592.35	1630
2	8	`tanh`	sample	0.66	1.00	0.0015	0.0667	976.96	574.51	1402
2	8	`sigmoid`	sample	0.57	1.00	0.0005	0.0658	976.47	573.74	4388
2	8	`ReLU`	`PTPE`	0.76	1.00	0.0057	0.0672	1689.16	581.58	314
2	8	`GELU`	`PTPE`	0.65	1.00	0.0018	0.0663	1727.50	592.35	1630
2	8	`tanh`	`PTPE`	0.66	1.00	0.0015	0.0667	1728.77	574.51	1402
2	8	`sigmoid`	`PTPE`	0.56	1.00	0.0005	0.0658	1729.76	573.74	4388
2	8	`ReLU`	`1d-T2-GC`	0.76	1.00	0.0057	0.0672	2604.90	581.58	314
2	8	`GELU`	`1d-T2-GC`	0.65	1.00	0.0018	0.0663	2627.91	592.35	1630
2	8	`tanh`	`1d-T2-GC`	0.66	1.00	0.0015	0.0667	2626.78	574.51	1402
2	8	`sigmoid`	`1d-T2-GC`	0.56	1.00	0.0005	0.0658	2584.98	573.74	4388
2	8	`ReLU`	`T1`	0.76	1.00	0.0057	0.0672	45.97	581.58	314
2	8	`GELU`	`T1`	0.65	1.00	0.0018	0.0663	48.04	592.35	1630
2	8	`tanh`	`T1`	0.66	1.00	0.0015	0.0667	35.65	574.51	1402
2	8	`sigmoid`	`T1`	0.56	1.00	0.0005	0.0658	36.17	573.74	4388
3	8	`ReLU`	sample	0.72	1.00	0.0042	0.0671	901.00	607.11	507
3	8	`GELU`	sample	0.68	1.00	0.0015	0.0670	767.37	617.05	2351
3	8	`tanh`	sample	0.63	1.00	0.0004	0.0655	767.38	609.40	8258
3	8	`sigmoid`	sample	0.58	0.99	0.0001	0.0652	765.92	614.30	46994
3	8	`ReLU`	`PTPE`	0.72	1.00	0.0042	0.0671	2662.52	607.11	507
3	8	`GELU`	`PTPE`	0.68	1.00	0.0015	0.0670	1412.53	617.05	2351
3	8	`tanh`	`PTPE`	0.64	1.00	0.0004	0.0655	1431.74	609.40	8258
3	8	`sigmoid`	`PTPE`	0.59	0.99	0.0001	0.0652	1428.09	614.30	46994
3	8	`ReLU`	`1d-T2-GC`	0.72	1.00	0.0042	0.0671	3504.19	607.11	507
3	8	`GELU`	`1d-T2-GC`	0.68	1.00	0.0015	0.0670	1857.08	617.05	2351
3	8	`tanh`	`1d-T2-GC`	0.63	1.00	0.0004	0.0655	1882.71	609.40	8258
3	8	`sigmoid`	`1d-T2-GC`	0.57	0.99	0.0001	0.0652	1882.18	614.30	46994
3	8	`ReLU`	`T1`	0.72	1.00	0.0042	0.0671	56.02	607.11	507
3	8	`GELU`	`T1`	0.69	1.00	0.0015	0.0670	48.69	617.05	2351
3	8	`tanh`	`T1`	0.63	1.00	0.0004	0.0655	34.17	609.40	8258
3	8	`sigmoid`	`T1`	0.59	0.99	0.0001	0.0652	34.41	614.30	46994

Table 26: `Citeseer` Dataset: Certified Adversarial Robustness Radii. L is the number of layers, f_l is the number of hidden dimensions, p_{CERT}^{Thm1} and p_{CERT}^{Chn} are the percentage of nodes for which a positive radius could be certified with Theorem 1 (Thm1) in our manuscript and Cohen's method (Chn) respectively, ϵ_{CERT}^{Thm1} and ϵ_{CERT}^{Chn} are the average robustness radius among nodes with a positive robustness radius using Theorem 1 or Cohen's method respectively, T^{Thm1} and T^{Chn} the runtime for computing radii using Theorem 1 or Cohen's method, for the former we include the time it takes to estimate the moments. Time is reported in seconds.

L	f_l	activation	method	p_{CERT}^{Thm1}	p_{CERT}^{Chn}	ϵ_{CERT}^{Thm1}	ϵ_{CERT}^{Chn}	T^{Thm1}	T^{Chn}	$\bar{C}$
2	4	`ReLU`	sample	0.86	1.00	0.0133	0.0714	995.82	605.56	84
2	4	`GELU`	sample	0.85	1.00	0.0058	0.0718	996.88	610.76	210
2	4	`tanh`	sample	0.86	1.00	0.0149	0.0718	996.25	597.80	81
2	4	`sigmoid`	sample	0.87	1.00	0.0121	0.0716	995.11	598.43	98
2	4	`ReLU`	`PTPE`	0.00	1.00	NaN	0.0714	896.98	605.56	84
2	4	`GELU`	`PTPE`	0.85	1.00	0.0058	0.0718	896.23	610.76	210
2	4	`tanh`	`PTPE`	0.86	1.00	0.0149	0.0718	896.22	597.80	81
2	4	`sigmoid`	`PTPE`	0.83	1.00	0.0074	0.0716	896.83	598.43	98
2	4	`ReLU`	`1d-T2-GC`	0.86	1.00	0.0133	0.0714	1076.12	605.56	84
2	4	`GELU`	`1d-T2-GC`	0.85	1.00	0.0058	0.0718	1076.02	610.76	210
2	4	`tanh`	`1d-T2-GC`	0.86	1.00	0.0149	0.0718	1075.52	597.80	81
2	4	`sigmoid`	`1d-T2-GC`	0.87	1.00	0.0121	0.0716	1077.40	598.43	98
2	4	`ReLU`	`T1`	0.86	1.00	0.0133	0.0714	56.08	605.56	84
2	4	`GELU`	`T1`	0.85	1.00	0.0058	0.0718	57.21	610.76	210
2	4	`tanh`	`T1`	0.86	1.00	0.0149	0.0718	48.44	597.80	81
2	4	`sigmoid`	`T1`	0.87	1.00	0.0121	0.0716	48.49	598.43	98
3	4	`ReLU`	sample	0.84	1.00	0.0104	0.0710	846.39	622.06	90
3	4	`GELU`	sample	0.90	1.00	0.0129	0.0720	849.25	607.22	97
3	4	`tanh`	sample	0.88	1.00	0.0092	0.0719	848.17	607.46	122
3	4	`sigmoid`	sample	0.78	1.00	0.0007	0.0708	851.80	612.37	1637
3	4	`ReLU`	`PTPE`	0.00	1.00	NaN	0.0710	652.80	622.06	90
3	4	`GELU`	`PTPE`	0.90	1.00	0.0129	0.0720	657.68	607.22	97
3	4	`tanh`	`PTPE`	0.88	1.00	0.0092	0.0719	662.86	607.46	122
3	4	`sigmoid`	`PTPE`	0.63	1.00	0.0004	0.0708	655.12	612.37	1637
3	4	`ReLU`	`1d-T2-GC`	0.84	1.00	0.0104	0.0710	745.66	622.06	90
3	4	`GELU`	`1d-T2-GC`	0.90	1.00	0.0129	0.0720	743.47	607.22	97
3	4	`tanh`	`1d-T2-GC`	0.88	1.00	0.0092	0.0719	758.73	607.46	122
3	4	`sigmoid`	`1d-T2-GC`	0.78	1.00	0.0007	0.0708	747.14	612.37	1637
3	4	`ReLU`	`T1`	0.84	1.00	0.0104	0.0710	49.74	622.06	90
3	4	`GELU`	`T1`	0.90	1.00	0.0129	0.0720	52.10	607.22	97
3	4	`tanh`	`T1`	0.88	1.00	0.0092	0.0719	40.90	607.46	122
3	4	`sigmoid`	`T1`	0.78	1.00	0.0007	0.0708	41.14	612.37	1637

Table 27: `Chameleon` Dataset: Certified Adversarial Robustness Radii. L is the number of layers, f_l is the number of hidden dimensions, p^{Thm1}_{CERT} and p^{Chn}_{CERT} are the percentage of nodes for which a positive radius could be certified with Theorem 1 (Thm1) in our manuscript and Cohen's method (Chn) respectively, ϵ^{Thm1}_{CERT} and ϵ^{Chn}_{CERT} are the average robustness radius among nodes with a positive robustness radius using Theorem 1 or Cohen's method respectively, T^{Thm1} and T^{Chn} the runtime for computing radii using Theorem 1 or Cohen's method, for the former we include the time it takes to estimate the moments. Time is reported in seconds.

L	f_l	activation	method	p^{Thm1}_{CERT}	p^{Chn}_{CERT}	ϵ^{Thm1}_{CERT}	ϵ^{Chn}_{CERT}	T^{Thm1}	T^{Chn}	$\bar{C}$
2	8	ReLU	sample	0.29	0.99	7.87e-07	0.0253	353.62	691.10	681087
2	8	GELU	sample	0.08	0.99	6.23e-07	0.0217	358.33	621.52	1318627
2	8	tanh	sample	0.20	0.99	2.20e-07	0.0219	354.14	620.90	3251464
2	8	sigmoid	sample	0.19	0.98	4.06e-07	0.0243	353.89	616.34	1593721
2	8	ReLU	PTPE	0.29	0.99	8.00e-07	0.0253	953.68	691.10	681087
2	8	GELU	PTPE	0.07	0.99	6.81e-07	0.0217	953.25	621.52	1318627
2	8	tanh	PTPE	0.20	0.99	2.23e-07	0.0219	955.27	620.90	3251464
2	8	sigmoid	PTPE	0.33	0.98	6.12e-07	0.0243	954.58	616.34	1593721
2	8	ReLU	1d-T2-GC	0.26	0.99	6.48e-07	0.0253	1364.00	691.10	681087
2	8	GELU	1d-T2-GC	0.08	0.99	6.14e-07	0.0217	1363.57	621.52	1318627
2	8	tanh	1d-T2-GC	0.22	0.99	2.18e-07	0.0219	1363.82	620.90	3251464
2	8	sigmoid	1d-T2-GC	0.16	0.98	4.45e-07	0.0243	1364.03	616.34	1593721
2	8	ReLU	T1	0.26	0.99	6.58e-07	0.0253	51.47	691.10	681087
2	8	GELU	T1	0.08	0.99	6.26e-07	0.0217	52.26	621.52	1318627
2	8	tanh	T1	0.21	0.99	2.22e-07	0.0219	31.14	620.90	3251464
2	8	sigmoid	T1	0.18	0.98	4.11e-07	0.0243	31.22	616.34	1593721
3	8	ReLU	sample	0.09	0.99	5.72e-08	0.0234	360.09	641.33	14021955
3	8	GELU	sample	0.09	0.96	1.31e-08	0.0217	364.84	643.32	105156296
3	8	tanh	sample	0.09	0.98	3.15e-08	0.0217	360.80	634.00	25991014
3	8	sigmoid	sample	0.32	1.00	1.54e-08	0.0245	361.33	634.02	65930444
3	8	ReLU	PTPE	0.09	0.99	5.77e-08	0.0234	1579.50	641.33	14021955
3	8	GELU	PTPE	0.09	0.96	1.29e-08	0.0217	1579.88	643.32	105156296
3	8	tanh	PTPE	0.10	0.98	3.14e-08	0.0217	1579.72	634.00	25991014
3	8	sigmoid	PTPE	0.40	1.00	1.33e-08	0.0245	1581.50	634.02	65930444
3	8	ReLU	1d-T2-GC	0.09	0.99	6.27e-08	0.0234	1996.44	641.33	14021955
3	8	GELU	1d-T2-GC	0.09	0.96	1.31e-08	0.0217	1994.79	643.32	105156296
3	8	tanh	1d-T2-GC	0.09	0.98	3.07e-08	0.0217	1994.45	634.00	25991014
3	8	sigmoid	1d-T2-GC	0.34	1.00	1.53e-08	0.0245	1995.38	634.02	65930444
3	8	ReLU	T1	0.09	0.99	6.25e-08	0.0234	72.96	641.33	14021955
3	8	GELU	T1	0.09	0.96	1.30e-08	0.0217	73.72	643.32	105156296
3	8	tanh	T1	0.09	0.98	3.02e-08	0.0217	49.32	634.00	25991014
3	8	sigmoid	T1	0.33	1.00	1.58e-08	0.0245	49.57	634.02	65930444

Table 28: `Squirrel` Dataset: Certified Adversarial Robustness Radii. L is the number of layers, f_l is the number of hidden dimensions, p_{CERT}^{Thm1} and p_{CERT}^{Chn} are the percentage of nodes for which a positive radius could be certified with Theorem 1 (Thm1) in our manuscript and Cohen's method (Chn) respectively, ϵ_{CERT}^{Thm1} and ϵ_{CERT}^{Chn} are the average robustness radius among nodes with a positive robustness radius using Theorem 1 or Cohen's method respectively, T^{Thm1} and T^{Chn} the runtime for computing radii using Theorem 1 or Cohen's method, for the former we include the time it takes to estimate the moments. Time is reported in seconds.

L	f_l	activation	method	p_{CERT}^{Thm1}	p_{CERT}^{Chn}	ϵ_{CERT}^{Thm1}	ϵ_{CERT}^{Chn}	T^{Thm1}	T^{Chn}	$\bar{C}$
2	4	`ReLU`	sample	0.69	1.00	0.0000	0.0342	1753.84	675.65	8724
2	4	`GELU`	sample	0.58	1.00	4.44e-06	0.0339	1755.14	649.10	67495
2	4	`tanh`	sample	0.37	1.00	9.83e-07	0.0320	1754.68	645.46	258934
2	4	`sigmoid`	sample	0.50	1.00	2.04e-06	0.0323	1753.59	649.44	119309
2	4	`ReLU`	`PTPE`	0.69	1.00	0.0000	0.0342	3323.28	675.65	8724
2	4	`GELU`	`PTPE`	0.58	1.00	4.43e-06	0.0339	3321.91	649.10	67495
2	4	`tanh`	`PTPE`	0.37	1.00	9.86e-07	0.0320	3328.34	645.46	258934
2	4	`sigmoid`	`PTPE`	0.60	1.00	2.29e-06	0.0323	3342.29	649.44	119309
2	4	`ReLU`	`1d-T2-GC`	0.69	1.00	0.0000	0.0342	3779.88	675.65	8724
2	4	`GELU`	`1d-T2-GC`	0.58	1.00	4.42e-06	0.0339	3774.32	649.10	67495
2	4	`tanh`	`1d-T2-GC`	0.37	1.00	1.01e-06	0.0320	3776.52	645.46	258934
2	4	`sigmoid`	`1d-T2-GC`	0.50	1.00	2.14e-06	0.0323	3853.81	649.44	119309
2	4	`ReLU`	`T1`	0.69	1.00	0.0000	0.0342	393.72	675.65	8724
2	4	`GELU`	`T1`	0.58	1.00	4.43e-06	0.0339	393.91	649.10	67495
2	4	`tanh`	`T1`	0.37	1.00	1.01e-06	0.0320	266.82	645.46	258934
2	4	`sigmoid`	`T1`	0.51	1.00	2.13e-06	0.0323	277.53	649.44	119309
3	4	`ReLU`	sample	0.44	0.99	1.42e-06	0.0315	1779.31	744.89	217879
3	4	`GELU`	sample	0.59	1.00	4.06e-06	0.0346	1778.35	716.53	66101
3	4	`tanh`	sample	0.27	0.99	5.20e-08	0.0311	1775.09	697.63	5752412
3	4	`sigmoid`	sample	0.36	0.99	7.79e-08	0.0290	1767.01	726.68	3228206
3	4	`ReLU`	`PTPE`	0.00	0.99	NaN	0.0315	4292.48	744.89	217879
3	4	`GELU`	`PTPE`	0.59	1.00	4.06e-06	0.0346	4289.18	716.53	66101
3	4	`tanh`	`PTPE`	0.26	0.99	5.21e-08	0.0311	4294.61	697.63	5752412
3	4	`sigmoid`	`PTPE`	0.45	0.99	9.06e-08	0.0290	4369.06	726.68	3228206
3	4	`ReLU`	`1d-T2-GC`	0.42	0.99	1.27e-06	0.0315	4746.52	744.89	217879
3	4	`GELU`	`1d-T2-GC`	0.59	1.00	4.06e-06	0.0346	4748.83	716.53	66101
3	4	`tanh`	`1d-T2-GC`	0.27	0.99	5.15e-08	0.0311	4747.63	697.63	5752412
3	4	`sigmoid`	`1d-T2-GC`	0.35	0.99	7.62e-08	0.0290	4742.58	726.68	3228206
3	4	`ReLU`	`T1`	0.42	0.99	1.27e-06	0.0315	516.33	744.89	217879
3	4	`GELU`	`T1`	0.59	1.00	4.06e-06	0.0346	517.79	716.53	66101
3	4	`tanh`	`T1`	0.27	0.99	5.06e-08	0.0311	384.81	697.63	5752412
3	4	`sigmoid`	`T1`	0.36	0.99	8.00e-08	0.0290	533.83	726.68	3228206

Table 29: Generalization Bounds on Real-world Data.

dataset	L	empirical LHS	RHS	C in FCD_p	M	ρ_{S}	ρ_{P}	K	χ
`Cornell`	2	51.043	647.14	22.82	51.04	0.81	0.85	95	38
`Cornell`	3	19.38	254.44	13.09	19.38	0.78	0.82	95	38
`Wisconsin`	2	21.71	341.35	22.23	21.71	0.78	0.78	130	52
`Wisconsin`	3	23.92	364.57	18.65	23.92	0.73	0.71	130	52
`Texas`	2	17.84	238.15	14.08	17.83	0.70	0.70	95	38
`Texas`	3	12.39	170.46	12.28	12.39	0.50	0.47	95	38
`Cora`	2	6.00	436.38	44.43	5.91	0.77	0.78	2502.5	715
`Cora`	3	6.01	432.61	45.01	5.83	0.76	0.78	2502.5	715
`Citeseer`	2	6.18	423.58	22.21	6.19	0.81	0.82	2709	903
`Citeseer`	3	5.84	401.74	21.72	5.85	0.80	0.82	2709	903
`Chameleon`	2	2.40	151.95	31.75	1.93	0.86	0.82	1525	610
`Chameleon`	3	4.83	416.93	97.39	4.87	0.72	0.68	1525	610
`Squirrel`	2	2.15	155.54	18.00	1.71	0.92	0.94	3527.5	1411
`Squirrel`	3	1.98	136.03	18.69	1.42	0.85	0.89	3527.5	1411