Generalization Bounds for Spectral GNNs via Fourier Domain Analysis

Vahan A. Martirosyan[1] **Daniele Malitesta**[1] **Hugues Talbot**[1]
Jhony H. Giraldo[2] **Fragkiskos D. Malliaros**[1]
[1]Université Paris-Saclay, CentraleSupélec, Inria, France
[2]LTCI, Télécom Paris, Institut Polytechnique de Paris, France

Abstract

Spectral graph neural networks learn graph filters, but their behavior with increasing depth and polynomial order is not well understood. We analyze these models in the graph Fourier domain, where each layer becomes an element-wise frequency update, separating the fixed spectrum from trainable parameters and making depth and order explicit. In this setting, we show that Gaussian complexity is invariant under the Graph Fourier Transform, which allows us to derive data-dependent, depth, and order-aware generalization bounds together with stability estimates. In the linear case, our bounds are tighter, and on real graphs, the data-dependent term correlates with the generalization gap across polynomial bases, highlighting practical choices that avoid frequency amplification across layers.

1 INTRODUCTION

Graph Neural Networks (GNNs) have become the leading paradigm for graph machine learning, achieving state-of-the-art results on tasks ranging from node classification to link prediction (Wu et al., 2019; Zhou et al., 2020). A widely used class of these models, spectral GNNs, defines graph convolutions by applying filter operations in the graph's frequency domain (Shuman et al., 2012; Sandryhaila and Moura, 2013). Architectures such as ChebNet (Defferrard et al., 2016) and its recent advancements (Hariri et al., 2025), GCNs (Kipf and Welling, 2017), and more recent models using Bernstein (He et al., 2021) or Jacobi polynomials (Wang and Zhang, 2022) can all be unified under a common framework of learnable polynomial graph filters (Section 4).

Despite good empirical results, the theoretical principles governing the generalization of GNNs remain incomplete (Garg et al., 2020; Verma and Zhang, 2019). Clarifying the conditions under which these models generalize is important for designing robust architectures, particularly in the transductive setting where the training and test data are not i.i.d. (Esser et al., 2021; Vasileiou et al., 2025b). The core challenge comes from the complex, non-i.i.d. nature of graph data. Unlike traditional machine learning settings, the nodes are interconnected, and in the transductive setting, the labeled training set and unlabeled test set are inherently dependent. This structure makes it difficult to directly apply standard learning-theoretic tools and complicates efforts to derive interpretable generalization bounds that can guide architectural design. While significant research has focused on the expressive power of GNNs, often relating them to the Weisfeiler-Leman test (Morris et al., 2023), understanding why these models generalize well from a small set of labeled nodes to the rest of the graph remains an active area of research (Vasileiou et al., 2025b).

In this paper, we analyze the generalization of multi-layer spectral GNNs in the transductive setting. We work in the graph Fourier domain, and by Lemma 2 the full transductive Gaussian complexity is unchanged. This basis change turns graph convolution into element-wise multiplication by the frequency response, which lets us separate graph spectrum from learnable parameters and then plug a Fourier-side complexity bound into our generalization gap theorem (Theorem 2). Our main contributions are:

- We formalize spectral GNNs with arbitrary polyno-

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

mial bases (Section 4) and introduce the generalized Vandermonde matrix $\boldsymbol{V}_P$ (Definition 3) to represent frequency responses compactly (Eq. (11)), which simplifies the analysis.

- We derive a Fourier-domain, data-dependent Gaussian complexity bound for deep spectral GNNs that isolates the roles of the spectrum (via $\boldsymbol{V}_P$), basis/order, depth, and parameter norms (Section 5.2, Theorem 3). Substituting it into Theorem 2 yields an explicit bound on the transductive generalization gap.
- For linear spectral GNNs we prove a sharper bound that makes the depth effect explicit through the basis amplification profile (Theorem 4).
- We bound the network Jacobian norm (Theorem 5) and show the same factors that control the gap also control worst-case sensitivity.
- The first-layer term in our bound leads to a simple energy-weighted frequency regularizer (Eq. (17)) that reduces the measured generalization gap and improves accuracy with stable bases (see Table 2 and Table 1).

Our results provide a principle for architectural design: generalization is better when selecting polynomial bases that are spectrally stable and by ensuring that the learnable filters do not excessively amplify the dominant frequencies of the input signal.

2 RELATED WORK

Theoretical analysis of GNN generalization has largely followed three directions: analyzing algorithmic stability, using the PAC-Bayesian framework, and adapting classical tools like the Vapnik–Chervonenkis (VC) dimension and Rademacher complexity.

Stability, PAC-Bayesian, and Covering Number Bounds. Algorithmic stability offers an alternative view on generalization, connecting it to how much the model's output changes when the training set is perturbed. Verma and Zhang (2019) first studied this for GCNs, deriving stability-based bounds that depend on the spectral norm of the graph convolution filter. More recently, concurrent work (Liu and Wang, 2025) also uses stability to study spectral GNNs, deriving bounds in expectation over generative models (contextual stochastic block models) for single-layer monomial filters. While both of these works are limited to single-layer models, our approach evaluates capacity directly on any fixed graph instance, explicitly accommodating deep, multi-layer architectures and arbitrary polynomial bases. Though we rely on Gaussian complexity rather than algorithmic stability, we arrive at a similar core conclusion: spectral properties are key to generalization.

The PAC-Bayesian framework has also been successfully applied. Liao et al. (2021) established generalization bounds for GCNs and Message Passing Neural Networks (MPNNs) that depend on the spectral norms of the weight matrices and the maximum node degree. This line of work also emphasizes the role of parameter norms, a factor that appears explicitly in our bounds.

Another line of work studies generalization by placing a metric on the space of graphs and showing that MPNNs are Lipschitz (or uniformly continuous) with respect to this metric, which yields covering-number bounds (Vasileiou et al., 2025a). Unlike these graph-space results, we analyze a single fixed graph in the transductive setting.

VC Dimension and Rademacher Complexity. Initial theoretical studies focused on bounding the VC dimension of GNNs. Scarselli et al. (2018) provided early bounds for a specific class of GNNs by using Pfaffian function theory. More recently, Morris et al. (2023) connected the VC dimension directly to the expressive power of GNNs, showing it is related to the number of non-isomorphic graphs distinguishable by the Weisfeiler-Leman test. However, as noted by Esser et al. (2021), VC dimension-based bounds for GNNs can often be loose or even trivial, limiting their practical utility for explaining generalization on a fixed graph.

To obtain more informative, data-dependent bounds, some studies have turned to Rademacher complexity. Garg et al. (2020) derived Rademacher complexity bounds for graph-level prediction tasks, highlighting dependencies on model depth and feature dimension. For the transductive (node-level) setting central to our work, El-Yaniv and Pechyony (2009) developed Transductive Rademacher Complexity (TRC), which can provide meaningful bounds where VC dimension fails. Using TRC (Esser et al., 2021) demonstrates the importance of the interaction term between the graph diffusion operator and the node features, a finding that is consistent with our data-dependent results. In our work we study the same transductive node setting, but in the graph Fourier domain, which allows for a more fine-grained decomposition of this interaction.

3 PRELIMINARIES

This section provides the foundational concepts for our analysis. We first define the notation and the formal problem setup for transductive node regression. We then explain the theoretical basis for our approach to generalization, detailing the relationship between the

generalization gap and Gaussian complexity. Finally, we review the principles of graph Fourier analysis.

Notation and Problem Setup. We consider an undirected graph $G = (V, E)$, where V is the set of $n = |V|$ nodes and E is the set of edges. Each node is associated with an initial feature vector. These are collected in a feature matrix $\boldsymbol{H}^{(0)} \in \mathbb{R}^{n \times d_0}$, where d_0 is the dimensionality of the input features for each node.

For our analysis, we use the normalized graph Adjacency matrix $\hat{\boldsymbol{A}} = \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$, where $\boldsymbol{A}$ is the adjacency matrix and $\boldsymbol{D}$ is the degree matrix. As a real symmetric matrix, $\hat{\boldsymbol{A}}$ has an eigendecomposition $\hat{\boldsymbol{A}} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$. The matrix $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n]$ contains the orthonormal eigenvectors, and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ contains the corresponding real eigenvalues, $-1 \leq \lambda_1 \leq \cdots \leq \lambda_n \leq 1$. The eigenvectors $\boldsymbol{U}$ constitute the basis for the Graph Fourier Transform (GFT) (Shuman et al., 2012). The GFT of a graph signal $\boldsymbol{x} \in \mathbb{R}^n$ is $\widehat{\boldsymbol{x}} = \boldsymbol{U}^\top \boldsymbol{x}$, and its inverse is $\boldsymbol{x} = \boldsymbol{U}\widehat{\boldsymbol{x}}$.

The task is transductive node regression, where the complete graph structure and the features $\boldsymbol{H}^{(0)}$ for all n nodes are accessible during training. However, the ground-truth labels, represented by a vector $\boldsymbol{y} \in \mathbb{R}^n$, are only revealed for a subset of nodes. The node set V is partitioned into a labeled training set V_L of size m and an unlabeled test set V_U of size $u = n - m$.

A GNN is a parameterized function, denoted $f_{\boldsymbol{\theta}}$, that maps the graph structure and node features to a vector of predictions for all nodes, $\hat{\boldsymbol{y}} = f_{\boldsymbol{\theta}}(\boldsymbol{H}^{(0)}, G) \in \mathbb{R}^n$. The performance of the model is measured by a loss function $\ell(\cdot, \cdot)$. We distinguish between two key performance metrics:

- The empirical risk, $R_L(f)$, is the average loss computed on the labeled training data. It measures how well the model fits the data it was trained on:
$$R_L(f) = \frac{1}{m} \sum_{i \in V_L} \ell(\hat{y}_i, y_i). \tag{1}$$
- The generalization error, $R_U(f)$, is the average loss on the unlabeled data. In the transductive setting, this serves as the proxy for the true risk and measures how well the model performs on unseen nodes:
$$R_U(f) = \frac{1}{u} \sum_{i \in V_U} \ell(\hat{y}_i, y_i). \tag{2}$$

Notation convention. We overload f to denote both the predictor and its output vector on all n nodes; when context is clear we drop the arguments and write $f \in \mathbb{R}^n$, with f_i meaning the prediction at node i (i.e., $(f(\boldsymbol{H}^{(0)}, G))_i$). In the Fourier domain we write $\widehat{f} = \boldsymbol{U}^\top f$ and $\widehat{f}_i$ for its i-th coefficient.

The Generalization Gap and Transductive Complexity. The central objective of our theoretical analysis is to understand and bound the generalization gap, defined as the absolute difference $|R_U(f) - R_L(f)|$(Bartlett and Mendelson, 2002). The generalization gap is the critical indicator of a model's ability to learn underlying patterns versus simply memorizing the training data. A large gap signifies overfitting, where the model has learned specific patterns of the training set that do not hold for the rest of the graph. The primary driver of overfitting is the complexity of the function class $\mathcal{F} = \{f_{\boldsymbol{\theta}} | \boldsymbol{\theta} \in \Theta\}$ from which the model is selected. To analyze this in the transductive setting, we use a specialized tool: TRC, which measures a function class's ability to fit random labels on a random partition of all n nodes.

Definition 1 (Transductive Rademacher Complexity (El-Yaniv and Pechyony, 2009))**.** *Let $\mathcal{F}$ be a class of real-valued functions over a domain of n points, let m be the number of labeled points, and let $p \in [0, 0.5]$. Let $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_n)^\top$ be a vector of independent random variables, where each σ_i takes the value $+1$ or -1 with probability p, and 0 with probability $1 - 2p$. The Transductive Rademacher Complexity of $\mathcal{F}$ is defined as:*

$$\mathfrak{R}_{m,n}(\mathcal{F}) = \left(\frac{1}{m} + \frac{1}{n-m}\right) \cdot \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{n} \sigma_i f_i\right]. \tag{3}$$

The relationship between the generalization gap and TRC is fundamental. For a loss function with values in a range of width C, the gap is bounded with high probability over the random partition of labeled nodes.

Theorem 1 (Transductive Generalization Bound (El-Yaniv and Pechyony, 2009))**.** *With probability at least $1 - \delta$, for any predictor $f \in \mathcal{F}$:*

$$R_U(f) \leq R_L(f) + \mathfrak{R}_{m,n}(\mathcal{F}) + C_1 \frac{n\sqrt{\min\{m, u\}}}{mu} + C_2 \sqrt{\frac{n}{mu} \ln(\frac{1}{\delta})}, \tag{4}$$

where $u = n - m$, and C_1, C_2 are absolute constants.

This theorem shows that the generalization gap is controlled by the TRC. While TRC provides the formal framework, our analysis will derive a bound on the closely related Full Transductive Gaussian Complexity (FTGC), which is more suitable for spectral analysis due to the properties of Gaussian variables.

Definition 2 (FTGC)**.** *The Gaussian complexity of a function class $\mathcal{F}$ over the entire vertex set V is:*

$$\mathcal{G}_V(\mathcal{F}) = \mathbb{E}_{\boldsymbol{g}} \left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} g_i f_i\right], \tag{5}$$

where $g_1, \ldots, g_n \sim \mathcal{N}(0,1)$ are i.i.d. standard Gaussian variables.

The two complexity measures are linked by the following standard result, which we prove in App. A for completeness.

Lemma 1 (TRC Bound by FTGC)**.** *The TRC $\mathfrak{R}_{m,n}(\mathcal{F})$ is upper-bounded by the full transductive Gaussian complexity $\mathcal{G}_V(\mathcal{F})$ as:*

$$\mathfrak{R}_{m,n}(\mathcal{F}) \leq \frac{n^2}{mu}\sqrt{\frac{\pi}{2}} \cdot \mathcal{G}_V(\mathcal{F}), \tag{6}$$

where $u = n - m$. For simplicity in the main generalization bound, we can denote the constant $\sqrt{\pi/2}$ as C_{gc}.

By substituting the result of Lemma 1 into the main generalization bound of Theorem 1, we arrive at a final bound expressed in terms of FTGC.

Theorem 2 (Generalization Gap by FTGC)**.** *With probability at least $1-\delta$, for any predictor $f \in \mathcal{F}$:*

$$\begin{aligned} R_U(f) \leq R_L(f) + \frac{n^2 C_{gc}}{mu}\mathcal{G}_V(\mathcal{F}) \\ + C_1 \frac{n\sqrt{\min\{m,u\}}}{mu} + C_2\sqrt{\frac{n}{mu}\ln(\frac{1}{\delta})}, \end{aligned} \tag{7}$$

where $\mathcal{G}_V(\mathcal{F})$ is the FTGC, $u = n-m$, and C_{gc}, C_1, C_2 are absolute constants.

This final theorem establishes the path for our analysis. Our primary technical goal is to derive a data-dependent bound on $\mathcal{G}_V(\mathcal{F})$ for spectral GNNs. This bound can then be substituted into Theorem 2 to provide a generalization bound. To achieve this, we shift the entire analysis from the spatial domain to the graph Fourier domain. We will show that FTGC is invariant under this transformation, which allows us to analyze the model in a domain where the complex graph convolution operator simplifies to an element-wise product. This change of basis is the key that enables us to separate the contributions of the graph structure, the chosen filter basis, and the learnable network parameters, leading to a more interpretable bound.

Node classification. Our bounds extend to multi-class node classification. Let the predictor output logits $F \in \mathbb{R}^{n\times C}$ and train with a bounded, L_ℓ-Lipschitz surrogate (e.g., multi-class hinge, or cross-entropy with logits clipped to $[-B, B]$ so $\ell \in [0, C]$). By the vector-contraction inequality for Gaussian/Rademacher complexities, composing with ℓ only scales the FTGC term by L_ℓ (Bartlett and Mendelson, 2002). Hence, our lemmas and theorems hold after replacing the regression loss by ℓ and R_L, R_U by their classification counterparts.

4 A UNIFIED FRAMEWORK FOR SPECTRAL GNNS

To analyze generalization for spectral GNNs, we adopt a unified formulation. Popular models such as ChebNet (Defferrard et al., 2016), GCN (Kipf and Welling, 2017), and BernNet (He et al., 2021) appear as special cases of a learnable polynomial filter. This formulation treats the polynomial basis as a choice of parametrization, which allows us to study all these models simultaneously and separate graph-dependent terms from learnable parameters.

We define a spectral GNN with L layers for node regression. The input is the feature matrix $\boldsymbol{H}^{(0)} \in \mathbb{R}^{n\times d_0}$. The propagation rule for layer $l \in \{0, \ldots, L-1\}$ is:

$$\boldsymbol{H}^{(l+1)} = \sigma\left(g_{\boldsymbol{\theta}^{(l)}}(\hat{\boldsymbol{A}})\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}\right), \tag{8}$$

where $\boldsymbol{W}^{(l)} \in \mathbb{R}^{d_l \times d_{l+1}}$ is a learnable weight matrix, σ is a non-linear activation function, and $g_{\boldsymbol{\theta}^{(l)}}(\hat{\boldsymbol{A}})$ is a graph filter. For efficiency and localization, the filter is a K-order polynomial of the normalized adjacency matrix:

$$g_{\boldsymbol{\theta}^{(l)}}(\hat{\boldsymbol{A}}) = \sum_{k=0}^{K} \theta_k^{(l)} P_k(\hat{\boldsymbol{A}}), \tag{9}$$

where $\{P_k\}_{k=0}^K$ is a chosen polynomial basis and $\boldsymbol{\theta}^{(l)} \in \mathbb{R}^{K+1}$ are the learnable filter coefficients. The action of this filter is best understood in the Fourier domain, where it becomes a multiplier on the graph frequencies:

$$g_{\boldsymbol{\theta}^{(l)}}(\hat{\boldsymbol{A}}) = \boldsymbol{U}\left(\sum_{k=0}^{K}\theta_k^{(l)}P_k(\boldsymbol{\Lambda})\right)\boldsymbol{U}^\top = \boldsymbol{U}g_{\boldsymbol{\theta}^{(l)}}(\boldsymbol{\Lambda})\boldsymbol{U}^\top. \tag{10}$$

The filter's frequency response can be expressed compactly using a matrix derived from the polynomial basis and the graph spectrum.

Definition 3 (Generalized Vandermonde Matrix)**.** *For a polynomial basis $\{P_k\}_{k=0}^K$ and normalized adjacency eigenvalues $\{\lambda_i\}_{i=1}^n$, the generalized Vandermonde matrix $\boldsymbol{V}_P \in \mathbb{R}^{n\times(K+1)}$ has entries: $(\boldsymbol{V}_P)_{ik} = P_k(\lambda_i)$.*

The vector of filter responses is $\boldsymbol{V}_P\boldsymbol{\theta}^{(l)}$, so $g_{\boldsymbol{\theta}^{(l)}}(\boldsymbol{\Lambda}) = \mathrm{diag}(\boldsymbol{V}_P\boldsymbol{\theta}^{(l)})$. The GNN layer from (8) can then be written as:

$$\boldsymbol{H}^{(l+1)} = \sigma\left(\boldsymbol{U}\,\mathrm{diag}(\boldsymbol{V}_P\boldsymbol{\theta}^{(l)})\,\boldsymbol{U}^\top\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}\right). \tag{11}$$

This formulation separates the graph structure $(\boldsymbol{U}, \boldsymbol{V}_P)$ from the learnable parameters $(\boldsymbol{\theta}^{(l)}, \boldsymbol{W}^{(l)})$.

App. J lists the exact polynomial bases (Chebyshev, Bernstein, Legendre, Monomial) and their properties.

5 MAIN RESULTS: GENERALIZATION AND STABILITY OF SPECTRAL GNNS

5.1 Analysis in the Graph Fourier Domain

Our goal is to bound the generalization gap by analyzing the complexity of the function class $\mathcal{F} = \{f_{\boldsymbol{\theta}}(\boldsymbol{H}^{(0)}, G) | \boldsymbol{\theta} \in \Theta\}$ that the GNN represents. We use the FTGC (Definition 2). A key insight of our analysis is that this complexity is invariant to the GFT. This allows us to move the analysis from the spatial (node) domain to the simpler spectral domain. Let $\widehat{\mathcal{F}} = \{\widehat{f} | f \in \mathcal{F}\}$ be the function class in the Fourier domain, where $\widehat{f} = \boldsymbol{U}^\top f$.

Lemma 2 (FTGC Invariance)**.** *The FTGC of the GNN function class is the same in the spatial and Fourier domains. That is,* $\mathcal{G}_V(\mathcal{F}) = \mathcal{G}_V(\widehat{\mathcal{F}})$.

A detailed proof is in the App. B. This lemma allows us to analyze the network's behavior in the Fourier domain, where the complex graph convolution becomes a simple element-wise multiplication by the filter's frequency response. To formalize this, let $\widehat{\boldsymbol{H}}^{(l)} = \boldsymbol{U}^\top \boldsymbol{H}^{(l)}$ be the node features at layer l in the Fourier domain. We define a transformed activation function, $\tau(\boldsymbol{Z}) = \boldsymbol{U}^\top \sigma(\boldsymbol{U}\boldsymbol{Z})$, which encapsulates the change of basis. Using this notation, the propagation rule from (11) simplifies to:

$$\widehat{\boldsymbol{H}}^{(l+1)} = \tau\left(\operatorname{diag}(\boldsymbol{V}_P \boldsymbol{\theta}^{(l)}) \widehat{\boldsymbol{H}}^{(l)} \boldsymbol{W}^{(l)}\right). \tag{12}$$

For our analysis to proceed, we must ensure that this transformed activation function preserves the properties of the original activation σ. The following lemma confirms this.

Lemma 3 (Lipschitz Preservation of Transformed Activation)**.** *Let* $\sigma : \mathbb{R} \to \mathbb{R}$ *be an* α*-Lipschitz function applied element-wise. Then, the transformed activation* $\tau(\boldsymbol{Z}) = \boldsymbol{U}^\top \sigma(\boldsymbol{U}\boldsymbol{Z})$ *is also* α*-Lipschitz with respect to the Frobenius norm* $\|\cdot\|_F$.

A detailed proof can be found in App. C. With these tools, we have established a simpler, equivalent representation of the GNN in the Fourier domain. This forms the foundation for deriving our main generalization bound in the next section.

5.2 A Data-Dependent Generalization Bound

Building on our Fourier domain framework, we now derive our main result: a data-dependent generalization bound. This bound connects the GNN's generalization error to the spectral properties of the input data, offering a more detailed view than data-independent analyses (Morris et al., 2023; Scarselli et al., 2018). We begin by defining the spectral energy of the input features.

Definition 4 (Input Signal Spectral Energy)**.** *The spectral energy of the input feature matrix* $\boldsymbol{H}^{(0)} \in \mathbb{R}^{n \times d_0}$ *at the graph frequency* λ_i *is the squared Frobenius norm of the* i*-th row of its GFT,* $\widehat{\boldsymbol{H}}^{(0)} = \boldsymbol{U}^\top \boldsymbol{H}^{(0)}$*:*

$$\mathcal{E}_0(\lambda_i) := \|(\widehat{\boldsymbol{H}}^{(0)})_i\|_2^2 = \|(\boldsymbol{U}^\top \boldsymbol{H}^{(0)})_i\|_2^2. \tag{13}$$

This quantity measures how much of the input signal's "energy" is concentrated at each graph frequency. Using this, we can state our main theorem.

Theorem 3 (Data-Dependent FTGC Bound)**.** *Let* $f_{\boldsymbol{\theta}}$ *be an* L*-layer spectral GNN with an* α*-Lipschitz activation* σ *satisfying* $\sigma(0) = 0$*. Assume the model parameters are constrained such that for each layer* $l \in \{0, \ldots, L-1\}$*, the weight matrix norm* $\|\boldsymbol{W}^{(l)}\|_2 \le C_{W,l}$ *and the filter coefficient norm* $\|\boldsymbol{\theta}^{(l)}\|_2 \le C_{\theta,l}$*. The FTGC of the function class* $\mathcal{F}$ *is bounded by:*

$$\mathcal{G}_V(\mathcal{F}) = \mathcal{G}_V(\widehat{\mathcal{F}}) \le \frac{1}{\sqrt{n}} \|\boldsymbol{V}_P\|_{2,\infty}^{L-1} \left(\prod_{l=0}^{L-1} \alpha C_{W,l} C_{\theta,l}\right) \cdot \left(\sum_{i=1}^{n} \|\boldsymbol{v}_i\|_2^2 \mathcal{E}_0(\lambda_i)\right)^{1/2}, \tag{14}$$

where $\boldsymbol{v}_i$ *is the* i*-th row of the generalized Vandermonde matrix* $\boldsymbol{V}_P$*, and* $\|\boldsymbol{V}_P\|_{2,\infty} = \max_i \|\boldsymbol{v}_i\|_2$ *is the maximum row-norm of* $\boldsymbol{V}_P$.

The full proof is provided in App. D. This bound reveals how different factors contribute to the model complexity. The term $\prod_{l=1}^{L-1}(\cdot)$ shows an exponential dependency on the network depth for layers $L-1$ down to 1. Crucially, the final term, $\left(\sum_{i=1}^{n} \|\boldsymbol{v}_i\|_2^2 \mathcal{E}_0(\lambda_i)\right)^{1/2}$, captures the interaction between the data and the model at the first layer. Here, $\|\boldsymbol{v}_i\|_2^2$ measures the potential amplification of the filter basis at frequency λ_i, while $\mathcal{E}_0(\lambda_i)$ is the signal's energy at that frequency. The bound is minimized when the signal's energy is concentrated at frequencies where the filter basis has a small response magnitude. This provides a key insight: good generalization is associated with spectral filters that avoid excessively amplifying the dominant frequencies of the input signal.

Next, we bound the FTGC for the case when we have no nonlinearities. By avoiding the initial bounding of the complexity by the output norm, we can derive a result that is tighter than the general non-linear bound from Theorem 3 and which more clearly reveals the influence of network depth on the model's complexity.

We consider an L-layer GNN with the non-linear activation σ removed, so the propagation rule becomes $\boldsymbol{H}^{(l+1)} = g_{\boldsymbol{\theta}^{(l)}}(\hat{\boldsymbol{A}})\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)}$.

Theorem 4 (Tighter Complexity Bound for Deep Linear GNNs)**.** *Let $f_{\boldsymbol{\theta}}$ be an L-layer linear spectral GNN with a single output feature. Assume the model parameters are constrained such that $\|\boldsymbol{W}^{(l)}\|_2 \leq C_{W,l}$, and $\|\boldsymbol{\theta}^{(l)}\|_2 \leq C_{\theta,l}$ for all layers l. The FTGC of the function class $\mathcal{F}$ is bounded by:*

$$\mathcal{G}_V(\mathcal{F}) \leq \frac{1}{n}\left(\prod_{l=0}^{L-1} C_{W,l}C_{\theta,l}\right)\left(\sum_{i=1}^{n}\|\boldsymbol{v}_i\|_2^{2L}\mathcal{E}_0(\lambda_i)\right)^{1/2}, \tag{15}$$

where $\boldsymbol{v}_i$ is the i-th row of $\boldsymbol{V}_P$ and $\mathcal{E}_0(\lambda_i)$ is the input signal spectral energy.

The full proof is in App. E. This bound provides a more precise characterization of complexity for deep linear models. Compared to the general non-linear bound in (14), it is tighter by a factor of $1/\sqrt{n}$. Most importantly, it clearly shows how the potential for complexity grows with depth. The data-dependent term now contains the factor $\|\boldsymbol{v}_i\|_2^{2L}$, indicating that any spectral instability in the polynomial basis (large $\|\boldsymbol{v}_i\|_2$ for some frequency λ_i) is amplified exponentially with the number of layers L. This provides a good theoretical justification for choosing spectrally stable polynomial bases (*e.g.*, Chebyshev or Bernstein) when designing deep spectral GNNs, even in the absence of non-linearities. In App. H we plot our FTGC-based bound from Theorem 3 against the measured generalization gap across polynomial order, bases, and datasets.

5.3 Worst-Case Stability via Jacobian Norm

Our analysis reveals that the same core principles governing the generalization error of a spectral GNN also control its stability. This property, which measures the sensitivity of a model's output to small perturbations in its input, is important for ensuring robustness and preventing issues like exploding gradients during training (Bousquet and Elisseeff, 2002). In this section, we formalize this connection by analyzing the network's Jacobian from a worst-case perspective.

A standard way to quantify stability is by bounding the spectral norm of the network's Jacobian (Hariri et al., 2025; Yoshida and Miyato, 2017; Novak et al., 2018). A large Jacobian norm implies that small input changes can be amplified into large output changes. We define the network Jacobian as $\mathcal{J} = \frac{\partial \mathrm{vec}(\boldsymbol{H}^{(L)})}{\partial \mathrm{vec}(\boldsymbol{H}^{(0)})}$, where $\mathrm{vec}(\cdot)$ vectorizes the feature matrices.

Theorem 5 (Jacobian Norm Bound)**.** *Let the GNN be an L-layer spectral GNN with an α-Lipschitz and continuously differentiable activation σ satisfying $\sigma(0) = 0$. Under the same parameter constraints as Theorem 3, the spectral norm of the network Jacobian $\mathcal{J}$ is bounded by:*

$$\|\mathcal{J}\|_2 \leq \prod_{l=0}^{L-1}\left(\alpha C_{W,l}C_{\theta,l}\|\boldsymbol{V}_P\|_{2,\infty}\right), \tag{16}$$

where $\|\boldsymbol{V}_P\|_{2,\infty} = \max_i \|\boldsymbol{v}_i\|_2$ is the maximum row-norm of the generalized Vandermonde matrix.

The full proof is in App. F. This bound quantifies the worst-case sensitivity of the GNN. It reveals an exponential dependency on depth (L) and highlights the central role of $\|\boldsymbol{V}_P\|_{2,\infty}$, the maximum amplification potential of the polynomial basis. We empirically validate this bound in App. G, demonstrating that it bounds the true spectral norm of the network Jacobian within a small constant factor across varying polynomial orders.

5.4 A Unified Principle for Architectural Design

Comparing our generalization bound (Theorem 3) and our stability bound (Theorem 5) reveals a unified principle: the same architectural factors govern both properties. The core components—the activation's Lipschitz constant (α), the norms of the weights ($C_{W,l}, C_{\theta,l}$), and the characteristics of the polynomial basis appear in both bounds.

The worst-case stability is controlled by the maximum amplification factor of the basis, $\|\boldsymbol{V}_P\|_{2,\infty}$. This term also appears in the generalization bound for the deeper layers. However, the generalization bound is more nuanced. For the first layer, it depends on the fine-grained interaction term $\left(\sum_{i=1}^{n}\|\boldsymbol{v}_i\|_2^2\mathcal{E}_0(\lambda_i)\right)^{1/2}$, which measures how the filter basis amplifies the input signal's actual spectral energy distribution.

This provides a clear design strategy. To ensure a model is both stable and generalizable, one should start by selecting a polynomial basis with a low maximum amplification, $\|\boldsymbol{V}_P\|_{2,\infty}$. A large $\|\boldsymbol{v}_i\|_2$ means the chosen polynomial basis is highly sensitive to that frequency, so any signal energy present there will contribute more significantly to the model's complexity. Good generalization is therefore achieved when the signal's energy is concentrated at frequencies the filter basis considers less important.

This spectral perspective offers a distinct advantage over analyses in the spatial domain (Esser et al., 2021), which often rely on the diffusion operator's norm $\|\hat{\boldsymbol{A}}\|_\infty$ (bounded by $O(\sqrt{degree_{max}/degree_{min}})$) or on the maximum degree (Liao et al., 2021). As we empirically demonstrate in App. I, this reliance causes

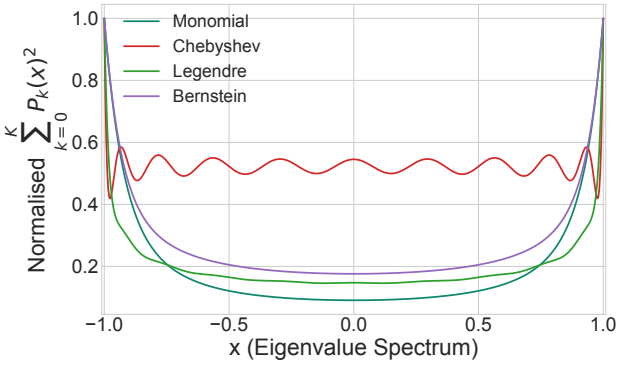


Figure 1: Basis stability across the spectrum ($K = 10$).

spatial bounds to grow exponentially with network depth. Our approach effectively trades a dependency on the graph's structural irregularity for a dependency on the mathematical stability of the filter basis. For many common architectures like GCN, this trade-off is highly favorable, as a simple polynomial basis can yield a small, constant $\|\boldsymbol{V}_P\|_{2,\infty}$ (*e.g.*, 1 for GCN) regardless of the graph's degree disparity, leading to a much tighter guarantee. By selecting a spectrally stable polynomial basis (low $\|\boldsymbol{V}_P\|_{2,\infty}$) and designing filters that do not amplify the dominant frequencies of the input signal, we can simultaneously build models that are less prone to overfitting and exploding gradients.

6 DESIGN IMPLICATIONS: BASIS STABILITY AND OVERSMOOTHING

Our theoretical bounds in Section 5 show that the generalization and stability of a spectral GNN depend on the properties of the chosen polynomial basis $\{P_k\}$. This dependency appears through the term $\|\boldsymbol{v}_i\|_2^2$, which is the squared row norm of the generalized Vandermonde matrix $\boldsymbol{V}_P$.

We refer to the function $x \mapsto \sum_{k=0}^{K} P_k(x)^2$ as the amplification profile of the basis, as it shows how sensitive the basis is to different frequencies. When this function is evaluated at the graph eigenvalues λ_i, its value is exactly equal to $\|\boldsymbol{v}_i\|_2^2$. In Figure 1, we plot the amplification profiles for different bases. From the plot, we can group these bases, which are normalized to a maximum value of 1, into three classes based on their uniformity:

1. The *Chebyshev* basis is the most uniform. Its profile is relatively flat across the middle of the spectrum, with its minimum value (around 0.5) being comparatively close to its maximum value of 1.0.
2. The *Bernstein* and *Legendre* bases have a clear U-shape, making them non-uniform. Their sensitivity is much higher at the spectral edges than in the center.
3. The *Monomial* basis is the least uniform. It exhibits the most extreme U-shape, with the largest comparative difference between its sensitivity at the edges versus the center.

This uniformity has direct consequences for generalization. As shown in our main bound (Theorem 3), model complexity depends on the interaction term $\sum_i \|\boldsymbol{v}_i\|_2^2 \mathcal{E}_0(\lambda_i)$. For a non-uniform basis, the model's complexity becomes highly sensitive to the signal's energy distribution ($\mathcal{E}_0(\lambda_i)$), leading to unpredictable performance. The uniform profile of Chebyshev makes the complexity less dependent on the signal, which should lead to more consistent performance across different graphs.

Furthermore, this analysis provides insight into the problem of *oversmoothing* (Li et al., 2018) in deep GNNs. Our bound for deep linear models (Theorem 4) includes the term $\|\boldsymbol{v}_i\|_2^{2L}$. As depth L increases, the model's complexity becomes exponentially dominated by the largest values of the amplification profile. For non-uniform bases, this forces the model to focus only on the spectral edges, which is a potential cause of oversmoothing (NT and Maehara, 2019; Oono and Suzuki, 2020).

Practical regularization. The first-layer, data-dependent term of our nonlinear bound (Eq. 14) suggests penalizing energy-weighted (EW) amplification. To discourage large gains at frequencies where the input has energy, we use the simple ratio:

$$\mathcal{R}_{\text{EW}} = \frac{\|\text{diag}(\boldsymbol{V}_P\boldsymbol{\theta})\,\widehat{\boldsymbol{H}}^{(0)}\|_F^2}{\|\widehat{\boldsymbol{H}}^{(0)}\|_F^2} = \frac{\|g_{\boldsymbol{\theta}}(\hat{\boldsymbol{A}})\boldsymbol{H}^{(0)}\|_F^2}{\|\boldsymbol{H}^{(0)}\|_F^2}. \tag{17}$$

In practice, we detach $\boldsymbol{H}^{(0)}$ in this penalty term so that the gradient only updates the filter parameters. Training details are provided in Section 7.

7 EMPIRICAL VALIDATION

7.1 Experimental Setup

To test the hypothesis from Section 6, we conduct node classification experiments on several benchmark datasets using the different polynomial bases.

Model. The core of our experimental model consists of a single linear spectral graph layer, as defined in Section 4. A residual connection is added to the graph

Table 1: **Test accuracy improvement depends on basis stability.** The column Δ uses empirically calculated ρ for paired significance ($^{*}p < 0.05, ^{**}p < 0.01, ^{***}p < 0.001$). While other bases occasionally degrade performance (negative Δ), a consistent positive improvement is observed across all datasets only for the Chebyshev basis.

	Chameleon			Squirrel			Cora			Citeseer		
Basis	**Base**	**+Reg**	**Δ**	**Base**	**+Reg**	**Δ**	**Base**	**+Reg**	**Δ**	**Base**	**+Reg**	**Δ**
Monomial	43.68 ± 2.0	44.28 ± 1.2	$+0.60 \uparrow$	27.87 ± 1.2	26.76 ± 0.8	$-1.11^{*} \downarrow$	78.33 ± 1.4	75.34 ± 1.5	$-2.99^{***} \downarrow$	66.15 ± 1.9	64.30 ± 2.5	$-1.85 \downarrow$
Legendre	41.22 ± 1.8	42.74 ± 1.9	$+1.52 \uparrow$	26.32 ± 1.7	29.15 ± 1.0	$+2.83^{***} \uparrow$	79.53 ± 1.4	79.36 ± 1.4	$-0.17 \downarrow$	65.35 ± 1.7	66.77 ± 2.0	$+1.42 \uparrow$
Bernstein	40.58 ± 1.8	42.23 ± 1.9	$+1.65^{*} \uparrow$	26.44 ± 0.6	31.23 ± 0.9	$+4.79^{***} \uparrow$	78.45 ± 2.0	79.08 ± 1.7	$+0.63 \uparrow$	65.47 ± 2.0	65.30 ± 2.0	$-0.17 \downarrow$
Chebyshev	41.76 ± 1.7	43.29 ± 1.9	$+1.53 \uparrow$	27.89 ± 0.8	31.20 ± 0.7	$+3.31^{***} \uparrow$	77.64 ± 2.0	78.23 ± 2.1	$+0.59 \uparrow$	65.57 ± 1.6	65.88 ± 1.6	$+0.31 \uparrow$

Table 2: **The regularizer's effect on the generalization gap.** Column Δ shows the change, where a negative value indicates a reduction in the gap ($^{*}p < 0.05, ^{**}p < 0.01, ^{***}p < 0.001$). The gap reduction is most consistent and significant on the Chebyshev basis.

	Chameleon			Squirrel			Cora			Citeseer		
Basis	**Base**	**+Reg**	**Δ**	**Base**	**+Reg**	**Δ**	**Base**	**+Reg**	**Δ**	**Base**	**+Reg**	**Δ**
Monomial	4.93 ± 0.97	1.99 ± 0.17	$-2.94^{***} \downarrow$	4.21 ± 1.43	0.89 ± 0.3	$-3.32^{***} \downarrow$	0.75 ± 0.06	0.63 ± 0.04	$-0.12^{***} \downarrow$	1.02 ± 0.04	1.02 ± 0.05	$+0.00$
Legendre	4.72 ± 1.1	2.23 ± 0.35	$-2.49^{***} \downarrow$	2.04 ± 0.66	1.79 ± 0.07	$-0.25 \downarrow$	0.65 ± 0.06	0.49 ± 0.04	$-0.16^{***} \downarrow$	0.83 ± 0.06	0.90 ± 0.03	$+0.07^{**} \uparrow$
Bernstein	2.14 ± 0.3	3.46 ± 1.38	$+1.32^{*} \uparrow$	2.44 ± 0.55	0.29 ± 0.18	$-2.15^{***} \downarrow$	0.64 ± 0.06	0.54 ± 0.04	$-0.10^{***} \downarrow$	1.00 ± 0.05	0.99 ± 0.03	$-0.01 \downarrow$
Chebyshev	3.06 ± 1.0	2.20 ± 0.17	$-0.86 \downarrow$	2.77 ± 0.31	0.09 ± 0.09	$-2.68^{***} \downarrow$	0.79 ± 0.07	0.69 ± 0.06	$-0.10^{**} \downarrow$	0.99 ± 0.02	0.92 ± 0.06	$-0.07^{*} \downarrow$

layer to improve training stability. To enhance expressive power, the input node features are first processed by an MLP before being passed to the graph layer. The output features are then passed to a linear classifier for node classification. Even with these components, our central graph convolution layer remains simple and linear. This allows us to isolate and observe the effects of the different polynomial bases and our spectral regularizer, which is the primary goal of this experiment.

Regularization. We apply standard dropout and add the regularization term $\lambda_{\text{EW}}\mathcal{R}_{\text{EW}}$ from Eq. (17), with λ_{EW} tuned on the validation set.

Evaluation. We evaluate all models on four representative datasets. For each dataset we use 10 random sparsified splits: 10 labeled nodes per class for training; from the remaining nodes we use 35% for validation and 65% for test. Results are reported as mean $\pm$ 95% confidence interval (CI) over the 10 splits. We report two metrics: test accuracy (%) and the generalization gap. We define the generalization gap as testing loss minus training loss. For both tables we do two hyperparameter searches before and after adding the regularisation. The full details are in App. K.

7.2 Results and Analysis

The results of our experiments are presented in Table 1 for test accuracy and Table 2 for the generalization gap. They confirm our analysis from Section 6.

First, for the *highly non-uniform Monomial basis*, the regularizer's effect is poor. As shown in Table 1, it fails to improve test accuracy and often hurts performance significantly (*e.g.*, on Cora and Citeseer). Second, for the *moderately non-uniform Bernstein and Legendre bases*, the results are mixed. The regularizer sometimes improves accuracy (*e.g.*, Bernstein on Squirrel), but the effect is inconsistent across datasets. For example, Bernstein's performance decreases on Citeseer, and Legendre's effect on Cora is negligible. The gap reduction for these bases, shown in Table 2, is also unreliable. Finally, for the *uniform Chebyshev basis*, the results are consistently positive. Table 2 shows that the regularizer reliably reduces the generalization gap, and Table 1 shows that this translates directly into a consistent improvement in test accuracy across all datasets.

These results provide clear evidence that the uniformity of a basis's amplification profile is a key predictor of its response to spectrally-aware regularization. The predictable behavior of the uniform Chebyshev basis allows the regularizer to function effectively and consistently.

8 LIMITATIONS

Our analysis targets the transductive, fixed-graph setting and does not cover inductive generalization to new nodes or graphs, or distribution shift. We study polynomial spectral filters on the normalized adjacency and assume α-Lipschitz activations with bounded parameter norms. Attention layers, non-polynomial operators, and other normalizations are outside our scope. Furthermore, while we provide a sharper bound for deep linear networks, our non-linear bound relies on sequential Lipschitz contractions. Applying covering number arguments and Dudley's entropy integral

could yield tighter capacity bounds for the non-linear case. Finally, our derived bounds help us understand spectral GNNs, but they are not direct predictors of test accuracy. They explain trends rather than provide point estimates.

9 CONCLUSIONS

We presented a unified Fourier-domain analysis of spectral GNNs that leads to new, depth and order-aware generalization bounds. By showing that FTGC is invariant under the GFT, we derived data-dependent bounds that explicitly capture the interaction between polynomial bases, filter amplification, and the spectral energy of the input. Our analysis also established tighter results in the linear setting and revealed that the same factors governing generalization also control network stability. These insights yield a simple regularizer and a clear design principle: spectrally stable bases and filters that avoid amplifying dominant frequencies improve both generalization and robustness. Together, our theoretical and empirical findings provide a principled foundation for understanding and guiding the design of spectral GNN architectures.

ACKNOWLEDGEMENTS

Vahan Martirosyan is the recipient of a PhD scholarship from the STIC Doctoral School of Université Paris-Saclay.

References

Abramowitz, M. and Stegun, I. A. (1964). *Handbook of Mathematical Functions*. Dover.

Bartlett, P. L. and Mendelson, S. (2002). *Rademacher and Gaussian Complexities: Risk Bounds and Structural Results*. JMLR.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Bousquet, O. and Elisseeff, A. (2002). Stability and generalization. *Journal of Machine Learning Research*, 2.

Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems (NeurIPS)*.

El-Yaniv, R. and Pechyony, D. (2009). Transductive rademacher complexity and its applications. *Journal of Artificial Intelligence Research*.

Esser, P. M., Vankadara, L. C., and Ghoshdastidar, D. (2021). Learning theory can (sometimes) explain generalisation in graph neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Garg, V. K., Jegelka, S., and Jaakkola, T. (2020). Generalization and representational limits of graph neural networks. In *International Conference on Machine Learning (ICML)*.

Hariri, A., Álvaro Arroyo, Gravina, A., Eliasof, M., Schönlieb, C.-B., Bacciu, D., Azizzadenesheli, K., Dong, X., and Vandergheynst, P. (2025). Return of chebnet: Understanding and improving an overlooked gnn on long range tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

He, M., Wei, Z., Huang, Z., and Xu, H. (2021). Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Higham, N. J. (2002). *Accuracy and Stability of Numerical Algorithms*. SIAM.

Horn, R. A. and Johnson, C. R. (2012). *Matrix Analysis*. Cambridge University Press, second edition.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. (2020). Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Review*.

Ledoux, M. and Talagrand, M. (1991). *Probability in Banach Spaces: Isoperimetry and Processes*. Springer-Verlag.

Li, Q., Han, Z., and Wu, X.-M. (2018). Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Liao, R., Urtasun, R., and Zemel, R. (2021). A pac-bayesian approach to generalization bounds for graph neural networks. In *International Conference on Learning Representations (ICLR)*.

Liu, F. and Wang, Q. (2025). Generalization of spectral graph neural networks.

Lorentz, G. G. (2012). *Bernstein Polynomials*. AMS, 2nd edition.

Mason, J. C. and Handscomb, D. C. (2002). *Chebyshev Polynomials*. CRC Press.

Morris, C., Geerts, F., Tönshoff, J., and Grohe, M. (2023). Wl meet vc. In *International Conference on Machine Learning (ICML)*.

Novak, R., Bahri, Y., Abolafia, D. A., Pennington, J., and Sohl-Dickstein, J. (2018). Sensitivity and generalization in neural networks: An empirical study. In *International Conference on Learning Representations (ICLR)*.

NT, H. and Maehara, T. (2019). Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550*.

Oono, K. and Suzuki, T. (2020). Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations (ICLR)*.

Pei, H., Wei, B., Chang, K. C.-C., Lei, Y., and Yang, B. (2020). Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Sandryhaila, A. and Moura, J. M. (2013). Discrete signal processing on graphs. *IEEE Transactions on Signal Processing*.

Scarselli, F., Tsoi, A. C., and Hagenbuchner, M. (2018). The vapnik-chervonenkis dimension of graph and recursive neural networks. *Neural Networks*.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. (2008). Collective classification in network data. *AI magazine*.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.

Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and Vandergheynst, P. (2012). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*.

Vasileiou, A., Finkelshtein, B., Geerts, F., Levie, R., and Morris, C. (2025a). Covered forest: Fine-grained generalization analysis of graph neural networks. In *International Conference on Machine Learning (ICML)*.

Vasileiou, A., Jegelka, S., Levie, R., and Morris, C. (2025b). Survey on generalization theory for graph neural networks. *arXiv preprint arXiv:2503.15650*.

Verma, S. and Zhang, Z.-L. (2019). Stability and generalization of graph convolutional neural networks. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*.

Wang, X. and Zhang, M. (2022). How powerful are spectral graph neural networks. In *International Conference on Machine Learning (ICML)*.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. (2019). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Yoshida, Y. and Miyato, T. (2017). Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*.

Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*.

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 — Mathematical setting, assumptions, and model are in Section 4 and Section 5, with notation in Section 3.]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable — We do not introduce a new algorithm.]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes — The link to the anonymous repository for the code is provided in the Appendix.]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes — Assumptions (α-Lipschitz σ, bounded norms) are stated before each result (Section 5).]
   (b) Complete proofs of all theoretical results. [Yes — Complete proofs are provided in the Appendix.]
   (c) Clear explanations of any assumptions. [Yes — We explain assumptions before theorems and lemmas (Section 5).]

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 — The anonymous repository

hosting the code to reproduce the results of this paper is provided in the Appendix.]

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes — We provide full details regarding training in the Appendix.]

(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). [Yes — All experiments are repeated across 10 random splits, with further details reported in Section 7.]

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes — We report complete details about the computing infrastructure in the Appendix.]

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. [Yes — All the needed references and credits have been explicitly mentioned in our code.]

(b) The license information of the assets, if applicable. [Yes — All license information of the existing assets were properly included.]

(c) New assets either in the supplemental material or as a URL, if applicable. [Yes — The only new assets of our work are the implemented codes for which we report a detailed Readme file in the anonymous repository.]

(d) Information about consent from data providers/curators. [Not Applicable — Since all used datasets are open-access, no permission was needed.]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable — Since all used datasets are open-access, no permission was needed.]

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 — Our paper does not involve crowdsourcing nor research with human subjects.]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable — Our paper does not involve crowdsourcing nor research with human subjects.]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable — Our paper does not involve crowdsourcing nor research with human subjects.]

Supplementary Materials

APPENDIX ORGANIZATION AND NOTATION

This appendix provides proofs for all theoretical claims made in the main paper, presents further empirical analyses that validate our theory, and includes additional details to ensure reproducibility.

- **Section Norms and Products:** We define the vector and matrix norms used throughout the analysis.
- **Section A:** Proof of Lemma 1, relating TRC and FTGC.
- **Section B:** Proof of Lemma 2, showing FTGC invariance under the GFT.
- **Section C:** Proof of Lemma 3, on the Lipschitz preservation of the transformed activation.
- **Section D:** Proof of Theorem 3, our main data-dependent FTGC bound.
- **Section E:** Proof of Theorem 4, the tighter FTGC bound for linear models.
- **Section F:** Proof of Theorem 5, the network Jacobian norm bound.
- **Section G:** Empirical validation demonstrating the tightness of our Jacobian norm bound.
- **Section H:** Empirical validation of our theoretical bound against the measured generalization gap, including sensitivity analysis and large-scale validation on ogbn-arxiv dataset.
- **Section I:** Quantitative comparison of our FTGC bound against prior spatial transductive bounds.
- **Section J:** Formal definitions and properties of the polynomial bases discussed.
- **Section K:** Full details on experimental setup, including hyperparameters, datasets, and splits.

Norms and Products

We use the following standard norms and matrix products in our analysis:

- **Vector l_2-norm:** For a vector $\boldsymbol{x} \in \mathbb{R}^n$, $\|\boldsymbol{x}\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$.
- **Matrix Frobenius norm:** For a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, the Frobenius norm is $\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n A_{ij}^2} = \sqrt{\mathrm{Tr}(\boldsymbol{A}^\top \boldsymbol{A})}$.
- **Matrix spectral norm:** For a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, the spectral norm (or operator norm) is induced by the vector l_2-norm: $\|\boldsymbol{A}\|_2 = \sup_{\|\boldsymbol{x}\|_2=1} \|\boldsymbol{A}\boldsymbol{x}\|_2 = \sigma_{\max}(\boldsymbol{A})$, where $\sigma_{\max}(\boldsymbol{A})$ is the largest singular value of $\boldsymbol{A}$.
- **Matrix $(2,\infty)$-norm:** For a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ with rows $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m$, this norm is the maximum l_2-norm among all its rows: $\|\boldsymbol{A}\|_{2,\infty} = \max_{i=1,\ldots,m} \|\boldsymbol{a}_i\|_2$.
- **Kronecker product:** For matrices $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{B} \in \mathbb{R}^{p\times q}$, their Kronecker product $\boldsymbol{A} \otimes \boldsymbol{B}$ is an $mp \times nq$ block matrix where the (i,j)-th block is the $p \times q$ matrix $A_{ij}\boldsymbol{B}$.
- **Khatri-Rao product:** For matrices $\boldsymbol{A} \in \mathbb{R}^{m\times k}$ and $\boldsymbol{B} \in \mathbb{R}^{n\times k}$ with the same number of columns, their Khatri-Rao (or column-wise Kronecker) product $\boldsymbol{A} \odot \boldsymbol{B}$ is an $mn \times k$ matrix where the j-th column is the Kronecker product of the j-th columns of $\boldsymbol{A}$ and $\boldsymbol{B}$, i.e., $(\boldsymbol{A} \odot \boldsymbol{B})_{:,j} = \boldsymbol{A}_{:,j} \otimes \boldsymbol{B}_{:,j}$. In our proofs, we use a row-wise variant, denoted $\odot_r$.

A PROOF OF LEMMA 1 (RELATION BETWEEN TRC AND FTGC)

Lemma 1 (TRC Bound by FTGC)**.** *The TRC $\mathfrak{R}_{m,n}(\mathcal{F})$ is upper-bounded by the full transductive Gaussian complexity $\mathcal{G}_V(\mathcal{F})$ as:*

$$\mathfrak{R}_{m,n}(\mathcal{F}) \leq \frac{n^2}{mu}\sqrt{\frac{\pi}{2}} \cdot \mathcal{G}_V(\mathcal{F}), \tag{6}$$

where $u = n - m$. For simplicity in the main generalization bound, we can denote the constant $\sqrt{\pi/2}$ as C_{gc}.

Proof. The proof connects the two complexity measures by showing that the expectation over the TRC random variables (σ_i) is bounded by the expectation over standard Gaussian variables (g_i). This is done by using standard Rademacher variables ($\epsilon_i \in \{-1, +1\}$) as an intermediate step.

We begin by defining the unnormalized core expectation terms:

$$E_{TRC} = \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{n} \sigma_i f_i \right],$$
$$E_{GC} = \mathbb{E}_{\boldsymbol{g}} \left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{n} g_i f_i \right],$$

where $\boldsymbol{\sigma}$ is the TRC random vector and $\boldsymbol{g}$ is a vector of i.i.d. $\mathcal{N}(0, 1)$ variables.

A random variable σ_i from the TRC definition (value $+1$ or -1 with probability p, and 0 with probability $1-2p$) can be constructed as the product $\sigma_i = \delta_i \epsilon_i$. Here, ϵ_i is a standard Rademacher variable (taking values $+1$ or -1 with probability 0.5), and δ_i is an independent Bernoulli variable (taking value 1 with probability $2p$ and 0 otherwise).

Using this construction, we can rewrite E_{TRC} and apply the law of total expectation:

$$E_{TRC} = \mathbb{E}_{\boldsymbol{\delta}, \boldsymbol{\epsilon}} \left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{n} \delta_i \epsilon_i f_i \right] = \mathbb{E}_{\boldsymbol{\delta}} \left[\mathbb{E}_{\boldsymbol{\epsilon}} \left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{n} \epsilon_i (\delta_i f_i) \right] \right]. \tag{18}$$

We now apply the contraction principle of Ledoux and Talagrand (1991). For any fixed realization of the Bernoulli vector $\boldsymbol{\delta}$, we have $|\delta_i| \leq 1$. The principle states that for such contractions, the expected supremum does not increase:

$$\mathbb{E}_{\boldsymbol{\epsilon}} \left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{n} \epsilon_i (\delta_i f_i) \right] \leq \mathbb{E}_{\boldsymbol{\epsilon}} \left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{n} \epsilon_i f_i \right]. \tag{19}$$

Since this inequality holds for any specific outcome of $\boldsymbol{\delta}$, it also holds when we take the expectation over $\boldsymbol{\delta}$. This gives us:

$$E_{TRC} \leq \mathbb{E}_{\boldsymbol{\epsilon}} \left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{n} \epsilon_i f_i \right]. \tag{20}$$

Let $\boldsymbol{g}$ be a vector of i.i.d. standard Gaussian variables, and let $\boldsymbol{\epsilon}$ be an independent Rademacher vector. The distribution of $\boldsymbol{g}$ is identical to the distribution of the vector $(|g_1|\epsilon_1, \ldots, |g_n|\epsilon_n)$. Using this property:

$$E_{GC} = \mathbb{E}_{\boldsymbol{g}} \left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{n} g_i f_i \right] = \mathbb{E}_{|\boldsymbol{g}|, \boldsymbol{\epsilon}} \left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{n} |g_i| \epsilon_i f_i \right]. \tag{21}$$

The function $\phi(z_1, \ldots, z_n) = \sup_f \sum_i z_i f_i$ is a convex function of its arguments z_i. We can therefore apply Jensen's inequality to the expectation over the magnitudes $|\boldsymbol{g}|$:

$$E_{GC} \geq \mathbb{E}_{\boldsymbol{\epsilon}} \left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{n} \mathbb{E}[|g_i|] \epsilon_i f_i \right]. \tag{22}$$

The expected absolute value of a standard Gaussian variable is a constant: $\mathbb{E}[|g_i|] = \sqrt{2/\pi}$. Substituting this in gives:

$$E_{GC} \geq \mathbb{E}_{\boldsymbol{\epsilon}} \left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{n} \sqrt{\frac{2}{\pi}} \epsilon_i f_i \right] = \sqrt{\frac{2}{\pi}} \cdot \mathbb{E}_{\boldsymbol{\epsilon}} \left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{n} \epsilon_i f_i \right]. \tag{23}$$

Rearranging this inequality and combining it with the result from Eq. (20), we get:

$$E_{TRC} \leq \mathbb{E}_{\boldsymbol{\epsilon}} \left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{n} \epsilon_i f_i \right] \leq \sqrt{\frac{\pi}{2}} \cdot E_{GC}. \tag{24}$$

We now have the relationship $E_{TRC} \leq \sqrt{\pi/2} \cdot E_{GC}$. We relate these unnormalized expectations back to the full complexity measures:

$$\mathfrak{R}_{m,n}(\mathcal{F}) = \left(\frac{1}{m} + \frac{1}{u}\right) \cdot E_{TRC} = \frac{n}{mu} \cdot E_{TRC},$$
$$\mathcal{G}_V(\mathcal{F}) = \frac{1}{n} \cdot E_{GC} \implies E_{GC} = n \cdot \mathcal{G}_V(\mathcal{F}).$$

Substituting these into our combined inequality:

$$\frac{mu}{n}\mathfrak{R}_{m,n}(\mathcal{F}) \leq \sqrt{\frac{\pi}{2}} \cdot (n \cdot \mathcal{G}_V(\mathcal{F})). \tag{25}$$

Finally, solving for $\mathfrak{R}_{m,n}(\mathcal{F})$ yields the desired result:

$$\mathfrak{R}_{m,n}(\mathcal{F}) \leq \frac{n^2}{mu}\sqrt{\frac{\pi}{2}} \cdot \mathcal{G}_V(\mathcal{F}). \tag{26}$$

This completes the proof. □

B PROOF OF LEMMA 2 (FTGC INVARIANCE)

Lemma 2 (FTGC Invariance)**.** *The FTGC of the GNN function class is the same in the spatial and Fourier domains. That is,* $\mathcal{G}_V(\mathcal{F}) = \mathcal{G}_V(\widehat{\mathcal{F}})$.

Proof. The FTGC over the entire vertex set V of size n is defined as:

$$\mathcal{G}_V(\mathcal{F}) = \mathbb{E}_{\boldsymbol{g}}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} g_i(f(\boldsymbol{H}^{(0)}))_i\right], \tag{27}$$

where $g_1, \ldots, g_n \sim \mathcal{N}(0,1)$ are i.i.d. standard Gaussian variables. Let $\boldsymbol{g} \in \mathbb{R}^n$ be the vector of these variables. Let $\hat{\boldsymbol{y}} = f(\boldsymbol{H}^{(0)})$ be an output from the function class $\mathcal{F}$, and let $\widehat{\hat{\boldsymbol{y}}} = \boldsymbol{U}^\top \hat{\boldsymbol{y}}$ be its representation in the Fourier domain, which belongs to the class $\widehat{\mathcal{F}}$. The proof relies on the rotational invariance of the standard multivariate Gaussian distribution.

$$\begin{aligned}
n \cdot \mathcal{G}_V(\mathcal{F}) &= \mathbb{E}_{\boldsymbol{g}}\left[\sup_{\hat{\boldsymbol{y}}\in\mathcal{F}} \langle \boldsymbol{g}, \hat{\boldsymbol{y}}\rangle\right] \\
&= \mathbb{E}_{\boldsymbol{g}}\left[\sup_{\widehat{\hat{\boldsymbol{y}}}\in\widehat{\mathcal{F}}} \langle \boldsymbol{g}, \boldsymbol{U}\widehat{\hat{\boldsymbol{y}}}\rangle\right] \\
&= \mathbb{E}_{\boldsymbol{g}}\left[\sup_{\widehat{\hat{\boldsymbol{y}}}\in\widehat{\mathcal{F}}} \langle \boldsymbol{U}^\top\boldsymbol{g}, \widehat{\hat{\boldsymbol{y}}}\rangle\right] \\
&= \mathbb{E}_{\boldsymbol{g}'}\left[\sup_{\widehat{\hat{\boldsymbol{y}}}\in\widehat{\mathcal{F}}} \langle \boldsymbol{g}', \widehat{\hat{\boldsymbol{y}}}\rangle\right] && \text{Let } \boldsymbol{g}' = \boldsymbol{U}^\top\boldsymbol{g}. \text{ Since } \boldsymbol{U} \text{ is orthonormal, } \boldsymbol{g}' \overset{d}{=} \boldsymbol{g}. \\
&= n \cdot \mathcal{G}_V(\widehat{\mathcal{F}}).
\end{aligned}$$

Therefore, it follows that $\mathcal{G}_V(\mathcal{F}) = \mathcal{G}_V(\widehat{\mathcal{F}})$. □

C PROOF OF LEMMA 3 (LIPSCHITZ PRESERVATION OF TRANSFORMED ACTIVATION)

Lemma 3 (Lipschitz Preservation of Transformed Activation)**.** *Let* $\sigma : \mathbb{R} \to \mathbb{R}$ *be an* α*-Lipschitz function applied element-wise. Then, the transformed activation* $\tau(\boldsymbol{Z}) = \boldsymbol{U}^\top\sigma(\boldsymbol{U}\boldsymbol{Z})$ *is also* α*-Lipschitz with respect to the Frobenius norm* $\|\cdot\|_F$.

Proof. We show that for any matrices $\boldsymbol{X}, \boldsymbol{Y}$ of the same dimensions, the function τ satisfies $\|\tau(\boldsymbol{X}) - \tau(\boldsymbol{Y})\|_F \leq \alpha\|\boldsymbol{X} - \boldsymbol{Y}\|_F$.

$$\begin{aligned}
\|\tau(\boldsymbol{X}) - \tau(\boldsymbol{Y})\|_F &= \|\boldsymbol{U}^\top\sigma(\boldsymbol{U}\boldsymbol{X}) - \boldsymbol{U}^\top\sigma(\boldsymbol{U}\boldsymbol{Y})\|_F \\
&= \|\boldsymbol{U}^\top(\sigma(\boldsymbol{U}\boldsymbol{X}) - \sigma(\boldsymbol{U}\boldsymbol{Y}))\|_F \\
&= \|\sigma(\boldsymbol{U}\boldsymbol{X}) - \sigma(\boldsymbol{U}\boldsymbol{Y})\|_F && \text{Unitary invariance of } \|\cdot\|_F \\
&\leq \alpha\|\boldsymbol{U}\boldsymbol{X} - \boldsymbol{U}\boldsymbol{Y}\|_F && \sigma \text{ is } \alpha\text{-Lipschitz element-wise} \\
&= \alpha\|\boldsymbol{U}(\boldsymbol{X} - \boldsymbol{Y})\|_F \\
&= \alpha\|\boldsymbol{X} - \boldsymbol{Y}\|_F. && \text{Unitary invariance of } \|\cdot\|_F
\end{aligned}$$

This completes the proof. □

D PROOF OF THEOREM 3

Theorem 3 (Data-Dependent FTGC Bound)**.** *Let $f_{\boldsymbol{\theta}}$ be an L-layer spectral GNN with an α-Lipschitz activation σ satisfying $\sigma(0) = 0$. Assume the model parameters are constrained such that for each layer $l \in \{0, \ldots, L-1\}$, the weight matrix norm $\|\boldsymbol{W}^{(l)}\|_2 \leq C_{W,l}$ and the filter coefficient norm $\|\boldsymbol{\theta}^{(l)}\|_2 \leq C_{\theta,l}$. The FTGC of the function class $\mathcal{F}$ is bounded by:*

$$\begin{aligned}
\mathcal{G}_V(\mathcal{F}) = \mathcal{G}_V(\widehat{\mathcal{F}}) \leq \frac{1}{\sqrt{n}}\|\boldsymbol{V}_P\|_{2,\infty}^{L-1}\left(\prod_{l=0}^{L-1}\alpha C_{W,l}C_{\theta,l}\right) \\
\cdot\left(\sum_{i=1}^{n}\|\boldsymbol{v}_i\|_2^2\mathcal{E}_0(\lambda_i)\right)^{1/2},
\end{aligned} \tag{14}$$

where $\boldsymbol{v}_i$ is the i-th row of the generalized Vandermonde matrix $\boldsymbol{V}_P$, and $\|\boldsymbol{V}_P\|_{2,\infty} = \max_i\|\boldsymbol{v}_i\|_2$ is the maximum row-norm of $\boldsymbol{V}_P$.

Proof. The proof proceeds by bounding the Frobenius norm of the final layer's features in the Fourier domain, $\|\widehat{\boldsymbol{h}}^{(L)}\|_2$, and then relating this quantity to the Gaussian complexity.

First, we connect the Gaussian complexity to the norm of the output features. For a function class producing matrices, a standard result from statistical learning theory bounds the complexity by the expected correlation with a random Gaussian matrix $\boldsymbol{g} \in \mathbb{R}^n$ (Shalev-Shwartz and Ben-David, 2014). Using the Cauchy-Schwarz inequality and the fact that $\mathbb{E}[\|\boldsymbol{g}\|_2] \leq \sqrt{n}$, we get:

$$\mathcal{G}_V(\mathcal{F}) \leq \frac{1}{n}\mathbb{E}[\|\boldsymbol{g}\|_2]\sup_{\boldsymbol{\theta}}\|\widehat{\boldsymbol{h}}^{(L)}\|_2 \leq \frac{1}{\sqrt{n}}\sup_{\boldsymbol{\theta}}\|\widehat{\boldsymbol{h}}^{(L)}\|_2. \tag{28}$$

Our task now is to bound $\sup_{\boldsymbol{\theta}}\|\widehat{\boldsymbol{h}}^{(L)}\|_2$ by unrolling the network's recursion from Eq. (12). For any intermediate layer $l \in \{1, \ldots, L-1\}$, we can derive a general, data-independent bound. Starting with the propagation rule and using the α-Lipschitz property of τ (Lemma 2), we have:

$$\|\widehat{\boldsymbol{H}}^{(l+1)}\|_F = \|\tau(\mathrm{diag}(\boldsymbol{V}_P\boldsymbol{\theta}^{(l)})\widehat{\boldsymbol{H}}^{(l)}\boldsymbol{W}^{(l)})\|_F \leq \alpha\|\mathrm{diag}(\boldsymbol{V}_P\boldsymbol{\theta}^{(l)})\widehat{\boldsymbol{H}}^{(l)}\boldsymbol{W}^{(l)}\|_F. \tag{29}$$

Using the submultiplicative property of matrix norms ($\|ABC\|_F \leq \|A\|_2\|B\|_F\|C\|_2$) (Horn and Johnson, 2012), this becomes:

$$\|\widehat{\boldsymbol{H}}^{(l+1)}\|_F \leq \alpha\|\mathrm{diag}(\boldsymbol{V}_P\boldsymbol{\theta}^{(l)})\|_2\|\widehat{\boldsymbol{H}}^{(l)}\|_F\|\boldsymbol{W}^{(l)}\|_2. \tag{30}$$

The spectral norm of the diagonal matrix is its largest absolute entry. We can bound this using the Cauchy-Schwarz inequality: $\|\mathrm{diag}(\boldsymbol{V}_P\boldsymbol{\theta}^{(l)})\|_2 = \max_i|\langle\boldsymbol{v}_i, \boldsymbol{\theta}^{(l)}\rangle| \leq (\max_i\|\boldsymbol{v}_i\|_2)\|\boldsymbol{\theta}^{(l)}\|_2 = \|\boldsymbol{V}_P\|_{2,\infty}\|\boldsymbol{\theta}^{(l)}\|_2$. This gives the recursive bound for intermediate layers:

$$\|\widehat{\boldsymbol{H}}^{(l+1)}\|_F \leq \alpha\|\boldsymbol{W}^{(l)}\|_2\|\boldsymbol{\theta}^{(l)}\|_2\|\boldsymbol{V}_P\|_{2,\infty}\|\widehat{\boldsymbol{H}}^{(l)}\|_F. \tag{31}$$

Unrolling this recursion from $l = L-1$ down to $l = 1$ gives:

$$\|\widehat{\boldsymbol{H}}^{(L)}\|_F \leq \left(\prod_{l=1}^{L-1} \alpha \|\boldsymbol{W}^{(l)}\|_2 \|\boldsymbol{\theta}^{(l)}\|_2 \|\boldsymbol{V}_P\|_{2,\infty}\right) \|\widehat{\boldsymbol{H}}^{(1)}\|_F. \tag{32}$$

For the first layer ($l = 0$), we derive a tighter, data-dependent bound for $\|\widehat{\boldsymbol{H}}^{(1)}\|_F$. Following the same initial steps:

$$\|\widehat{\boldsymbol{H}}^{(1)}\|_F \leq \alpha \|\mathrm{diag}(\boldsymbol{V}_P \boldsymbol{\theta}^{(0)}) \widehat{\boldsymbol{H}}^{(0)} \boldsymbol{W}^{(0)}\|_F \leq \alpha \|\boldsymbol{W}^{(0)}\|_2 \|\mathrm{diag}(\boldsymbol{V}_P \boldsymbol{\theta}^{(0)}) \widehat{\boldsymbol{H}}^{(0)}\|_F. \tag{33}$$

We now analyze the squared norm of the filtered signal term by expanding it:

$$\begin{aligned}
\|\mathrm{diag}(\boldsymbol{V}_P \boldsymbol{\theta}^{(0)}) \widehat{\boldsymbol{H}}^{(0)}\|_F^2 &= \sum_{i=1}^{n} \sum_{j=1}^{d_0} |(\mathrm{diag}(\boldsymbol{V}_P \boldsymbol{\theta}^{(0)}))_{ii} (\widehat{\boldsymbol{H}}^{(0)})_{ij}|^2 \\
&= \sum_{i=1}^{n} |(\boldsymbol{V}_P \boldsymbol{\theta}^{(0)})_i|^2 \sum_{j=1}^{d_0} |(\widehat{\boldsymbol{H}}^{(0)})_{ij}|^2 \\
&= \sum_{i=1}^{n} |\langle \boldsymbol{v}_i, \boldsymbol{\theta}^{(0)} \rangle|^2 \|(\widehat{\boldsymbol{H}}^{(0)})_i\|_2^2 \\
&\leq \sum_{i=1}^{n} (\|\boldsymbol{v}_i\|_2^2 \|\boldsymbol{\theta}^{(0)}\|_2^2) \mathcal{E}_0(\lambda_i) \qquad \text{(by Cauchy-Schwarz)} \\
&= \|\boldsymbol{\theta}^{(0)}\|_2^2 \sum_{i=1}^{n} \|\boldsymbol{v}_i\|_2^2 \mathcal{E}_0(\lambda_i).
\end{aligned}$$

Taking the square root provides the bound on the norm of the feature matrix after the first layer:

$$\|\widehat{\boldsymbol{H}}^{(1)}\|_F \leq \alpha \|\boldsymbol{W}^{(0)}\|_2 \|\boldsymbol{\theta}^{(0)}\|_2 \left(\sum_{i=1}^{n} \|\boldsymbol{v}_i\|_2^2 \mathcal{E}_0(\lambda_i)\right)^{1/2}. \tag{34}$$

Finally, substituting this data-dependent bound into the recursive inequality for $\|\widehat{\boldsymbol{H}}^{(L)}\|_F$, applying the parameter constraints $(C_{W,l}, C_{\theta,l})$ and plugging the result into the initial inequality for $\mathcal{G}_V(\mathcal{F})$ completes the proof. □

E PROOF OF THEOREM 4

Theorem 4 (Tighter Complexity Bound for Deep Linear GNNs)**.** *Let $f_{\boldsymbol{\theta}}$ be an L-layer linear spectral GNN with a single output feature. Assume the model parameters are constrained such that $\|\boldsymbol{W}^{(l)}\|_2 \leq C_{W,l}$, and $\|\boldsymbol{\theta}^{(l)}\|_2 \leq C_{\theta,l}$ for all layers l. The FTGC of the function class $\mathcal{F}$ is bounded by:*

$$\mathcal{G}_V(\mathcal{F}) \leq \frac{1}{n} \left(\prod_{l=0}^{L-1} C_{W,l} C_{\theta,l}\right) \left(\sum_{i=1}^{n} \|\boldsymbol{v}_i\|_2^{2L} \mathcal{E}_0(\lambda_i)\right)^{1/2}, \tag{15}$$

where $\boldsymbol{v}_i$ is the i-th row of $\boldsymbol{V}_P$ and $\mathcal{E}_0(\lambda_i)$ is the input signal spectral energy.

Proof. We work in the graph Fourier domain throughout. Recall that the (linear) L-layer model is

$$\widehat{\boldsymbol{H}}^{(l+1)} \;=\; \boldsymbol{D}_l \, \widehat{\boldsymbol{H}}^{(l)} \, \boldsymbol{W}^{(l)}, \qquad \boldsymbol{D}_l \;=\; \mathrm{diag}\big(\boldsymbol{V}_P \, \boldsymbol{\theta}^{(l)}\big),$$

with $\widehat{\boldsymbol{H}}^{(0)} \in \mathbb{R}^{n \times d_0}$, $\boldsymbol{W}^{(l)} \in \mathbb{R}^{d_l \times d_{l+1}}$, and $d_L = 1$. Let $\boldsymbol{v}_i^\top$ denote the i-th row of $\boldsymbol{V}_P$ and $\mathcal{E}_0(\lambda_i) := \|(\widehat{\boldsymbol{H}}^{(0)})_i\|_2^2$ the input spectral energy at frequency λ_i.

Unrolling gives $\widehat{\boldsymbol{h}}^{(L)} = \big(\prod_{l=0}^{L-1} \boldsymbol{D}_l\big) \widehat{\boldsymbol{H}}^{(0)} \big(\prod_{l=0}^{L-1} \boldsymbol{W}^{(l)}\big)$. By spectral-norm duality (applied layer-by-layer),

$$\sup_{\{\boldsymbol{W}^{(l)} : \|\boldsymbol{W}^{(l)}\|_2 \leq C_{W,l}\}} \langle \boldsymbol{g}, \widehat{\boldsymbol{h}}^{(L)} \rangle \;=\; \Big(\prod_{l=0}^{L-1} C_{W,l}\Big) \, \big\| \widehat{\boldsymbol{H}}^{(0)\top} \big(\prod_{l=0}^{L-1} \boldsymbol{D}_l\big) \boldsymbol{g} \big\|_2. \tag{35}$$

Hence, by the FTGC definition,

$$\mathcal{G}_V(\mathcal{F}) \;\le\; \frac{1}{n}\Big(\prod_{l=0}^{L-1} C_{W,l}\Big)\,\mathbb{E}_{\boldsymbol{g}}\Big[\sup_{\{\boldsymbol{\theta}^{(l)}:\,\|\boldsymbol{\theta}^{(l)}\|_2\le C_{\theta,l}\}} \big\|\widehat{\boldsymbol{H}}^{(0)\top}\big(\textstyle\prod_{l=0}^{L-1}\boldsymbol{D}_l\big)\boldsymbol{g}\big\|_2\Big]. \tag{36}$$

Next, we write the product of diagonals as a single diagonal via row-wise Kronecker lifting. For each frequency i, we have

$$\big(\textstyle\prod_{l=0}^{L-1}\boldsymbol{D}_l\big)_{ii} \;=\; \prod_{l=0}^{L-1}\langle \boldsymbol{v}_i, \boldsymbol{\theta}^{(l)}\rangle \;=\; \big\langle \boldsymbol{v}_i^{\otimes L},\, \boldsymbol{\theta}^{(L-1)}\otimes\cdots\otimes\boldsymbol{\theta}^{(0)}\big\rangle.$$

Let

$$\Theta \;:=\; \boldsymbol{\theta}^{(L-1)}\otimes\cdots\otimes\boldsymbol{\theta}^{(0)} \in \mathbb{R}^{(K+1)^L}, \qquad \boldsymbol{V}_{\otimes} \;:=\; \underbrace{\boldsymbol{V}_P \odot_r \boldsymbol{V}_P \odot_r \cdots \odot_r \boldsymbol{V}_P}_{L \text{ times}} \in \mathbb{R}^{n\times(K+1)^L},$$

where $\odot_r$ denotes the row-wise Khatri–Rao/Kronecker (Kolda and Bader, 2009) product so that the i-th row of $\boldsymbol{V}_{\otimes}$ equals $\boldsymbol{v}_i^{\otimes L}$ and $\|\boldsymbol{v}_i^{\otimes L}\|_2 = \|\boldsymbol{v}_i\|_2^L$. Then

$$\prod_{l=0}^{L-1}\boldsymbol{D}_l \;=\; \operatorname{diag}(\boldsymbol{V}_{\otimes}\Theta).$$

Thus, for each fixed $\boldsymbol{g}$,

$$\sup_{\{\boldsymbol{\theta}^{(l)}\}} \big\|\widehat{\boldsymbol{H}}^{(0)\top}\big(\prod_{l=0}^{L-1}\boldsymbol{D}_l\big)\boldsymbol{g}\big\|_2 \;=\; \sup_{\{\boldsymbol{\theta}^{(l)}\}} \big\|\widehat{\boldsymbol{H}}^{(0)\top}\operatorname{diag}(\boldsymbol{g})\,\boldsymbol{V}_{\otimes}\Theta\big\|_2 \;\le\; \Big(\prod_{l=0}^{L-1} C_{\theta,l}\Big)\,\big\|\widehat{\boldsymbol{H}}^{(0)\top}\operatorname{diag}(\boldsymbol{g})\,\boldsymbol{V}_{\otimes}\big\|_2,$$

since $\|\Theta\|_2 = \prod_{l=0}^{L-1}\|\boldsymbol{\theta}^{(l)}\|_2 \le \prod_l C_{\theta,l}$.

Taking expectation in Eq. (36) gives

$$\mathcal{G}_V(\mathcal{F}) \;\le\; \frac{1}{n}\Big(\prod_{l=0}^{L-1} C_{W,l}C_{\theta,l}\Big)\,\mathbb{E}_{\boldsymbol{g}}\Big[\big\|\widehat{\boldsymbol{H}}^{(0)\top}\operatorname{diag}(\boldsymbol{g})\,\boldsymbol{V}_{\otimes}\big\|_2\Big]. \tag{37}$$

By using $\|\boldsymbol{X}\|_2 \le \|\boldsymbol{X}\|_F$ and Jensen/Cauchy–Schwarz:

$$\mathbb{E}_{\boldsymbol{g}}\big[\|\widehat{\boldsymbol{H}}^{(0)\top}\operatorname{diag}(\boldsymbol{g})\,\boldsymbol{V}_{\otimes}\|_2\big] \;\le\; \sqrt{\mathbb{E}_{\boldsymbol{g}}\big[\|\widehat{\boldsymbol{H}}^{(0)\top}\operatorname{diag}(\boldsymbol{g})\,\boldsymbol{V}_{\otimes}\|_F^2\big]}\,.$$

Let $\boldsymbol{Q} := \widehat{\boldsymbol{H}}^{(0)}\widehat{\boldsymbol{H}}^{(0)\top}$. Then

$$\|\widehat{\boldsymbol{H}}^{(0)\top}\operatorname{diag}(\boldsymbol{g})\,\boldsymbol{V}_{\otimes}\|_F^2 = \operatorname{Tr}\Big(\boldsymbol{V}_{\otimes}^{\top}\operatorname{diag}(\boldsymbol{g})\,\boldsymbol{Q}\,\operatorname{diag}(\boldsymbol{g})\,\boldsymbol{V}_{\otimes}\Big).$$

Since $\mathbb{E}[g_i g_j] = \delta_{ij}$,

$$\mathbb{E}\big[\operatorname{diag}(\boldsymbol{g})\,\boldsymbol{Q}\,\operatorname{diag}(\boldsymbol{g})\big] = \operatorname{diag}(\boldsymbol{Q}).$$

Therefore

$$\mathbb{E}_{\boldsymbol{g}}\|\widehat{\boldsymbol{H}}^{(0)\top}\operatorname{diag}(\boldsymbol{g})\,\boldsymbol{V}_{\otimes}\|_F^2 = \operatorname{Tr}\Big(\boldsymbol{V}_{\otimes}^{\top}\operatorname{diag}(\boldsymbol{Q})\,\boldsymbol{V}_{\otimes}\Big) = \sum_{i=1}^{n}\boldsymbol{Q}_{ii}\,\big\|(\boldsymbol{V}_{\otimes})_{i,:}\big\|_2^2 \;=\; \sum_{i=1}^{n}\|(\widehat{\boldsymbol{H}}^{(0)})_i\|_2^2\,\|\boldsymbol{v}_i^{\otimes L}\|_2^2. \tag{38}$$

We have $\|(\widehat{\boldsymbol{H}}^{(0)})_i\|_2^2 = \mathcal{E}_0(\lambda_i)$ and $\|\boldsymbol{v}_i^{\otimes L}\|_2^2 = \|\boldsymbol{v}_i\|_2^{2L}$, so

$$\mathbb{E}_{\boldsymbol{g}}\|\widehat{\boldsymbol{H}}^{(0)\top}\operatorname{diag}(\boldsymbol{g})\,\boldsymbol{V}_{\otimes}\|_F^2 \;=\; \sum_{i=1}^{n}\|\boldsymbol{v}_i\|_2^{2L}\,\mathcal{E}_0(\lambda_i).$$

Combining with Eq. (37) yields

$$\mathcal{G}_V(\mathcal{F}) \;\le\; \frac{1}{n}\Big(\prod_{l=0}^{L-1} C_{W,l}C_{\theta,l}\Big)\,\left(\sum_{i=1}^{n}\|\boldsymbol{v}_i\|_2^{2L}\,\mathcal{E}_0(\lambda_i)\right)^{1/2}.$$

□

F PROOF OF THEOREM 5

Theorem 5 (Jacobian Norm Bound)**.** *Let the GNN be an L-layer spectral GNN with an α-Lipschitz and continuously differentiable activation σ satisfying $\sigma(0) = 0$. Under the same parameter constraints as Theorem 3, the spectral norm of the network Jacobian $\mathcal{J}$ is bounded by:*

$$\|\mathcal{J}\|_2 \leq \prod_{l=0}^{L-1} \left(\alpha C_{W,l} C_{\theta,l} \|\boldsymbol{V}_P\|_{2,\infty}\right), \tag{16}$$

where $\|\boldsymbol{V}_P\|_{2,\infty} = \max_i \|\boldsymbol{v}_i\|_2$ is the maximum row-norm of the generalized Vandermonde matrix.

Proof. The proof proceeds by analyzing the Jacobian in the graph Fourier domain, where the graph convolution operation simplifies to an element-wise product, and then combining bounds for each layer using the chain rule.

Our first step is to move the analysis to the graph Fourier domain. The Jacobian in the Fourier domain is defined as $\widehat{\mathcal{J}} = \frac{\partial \mathrm{vec}(\widehat{\boldsymbol{H}}^{(L)})}{\partial \mathrm{vec}(\widehat{\boldsymbol{H}}^{(0)})}$. The vectorized features in the two domains are related by the linear transformation $\mathrm{vec}(\widehat{\boldsymbol{H}}^{(l)}) = (\boldsymbol{I} \otimes \boldsymbol{U}^\top)\mathrm{vec}(\boldsymbol{H}^{(l)})$, where $\boldsymbol{U}$ is the orthogonal matrix of normalized Adjacency eigenvectors. By the multivariate chain rule, the Jacobians are related by a similarity transformation:

$$\widehat{\mathcal{J}} = (\boldsymbol{I} \otimes \boldsymbol{U}^\top)\mathcal{J}(\boldsymbol{I} \otimes \boldsymbol{U}). \tag{39}$$

Since $\boldsymbol{U}$ is an orthogonal (and thus unitary) matrix, the Kronecker product $(\boldsymbol{I} \otimes \boldsymbol{U})$ is also unitary. A fundamental property of the spectral norm is its unitary invariance, meaning $\|\boldsymbol{V}^H \boldsymbol{A} \boldsymbol{V}\|_2 = \|\boldsymbol{A}\|_2$. for any unitary matrix $\boldsymbol{V}$. Therefore, the spectral norms of the Jacobians in the two domains are identical:

$$\|\mathcal{J}\|_2 = \|\widehat{\mathcal{J}}\|_2 \tag{40}$$

This equivalence allows us to derive the bound in the Fourier domain without loss of generality.

By the chain rule, the full Jacobian can be expressed as the product of the Jacobians of individual layers:

$$\widehat{\mathcal{J}} = \prod_{l=0}^{L-1} \widehat{\mathcal{J}}^{(l)} \quad \text{where} \quad \widehat{\mathcal{J}}^{(l)} = \frac{\partial \mathrm{vec}(\widehat{\boldsymbol{H}}^{(l+1)})}{\partial \mathrm{vec}(\widehat{\boldsymbol{H}}^{(l)})}. \tag{41}$$

Using the submultiplicative property of the spectral norm, we can bound the total norm by the product of the individual layer norms: $\|\widehat{\mathcal{J}}\|_2 \leq \prod_{l=0}^{L-1} \|\widehat{\mathcal{J}}^{(l)}\|_2$.

We now bound the norm $\|\widehat{\mathcal{J}}^{(l)}\|_2$ for a single layer l. Let $\widehat{S}^{(l)} = \mathrm{diag}(\boldsymbol{V}_P \boldsymbol{\theta}^{(l)})\widehat{\boldsymbol{H}}^{(l)}\boldsymbol{W}^{(l)}$ be the pre-activation matrix in the Fourier domain. The propagation rule is $\widehat{\boldsymbol{H}}^{(l+1)} = \tau(\widehat{S}^{(l)}) = \boldsymbol{U}^\top \sigma(\boldsymbol{U}\widehat{S}^{(l)})$. The layer-wise Jacobian is given by:

$$\widehat{\mathcal{J}}^{(l)} = \underbrace{\frac{\partial \mathrm{vec}(\tau(\widehat{S}^{(l)}))}{\partial \mathrm{vec}(\widehat{S}^{(l)})}}_{J_\tau} \underbrace{\frac{\partial \mathrm{vec}(\widehat{S}^{(l)})}{\partial \mathrm{vec}(\widehat{\boldsymbol{H}}^{(l)})}}_{J_S}. \tag{42}$$

The second term, J_S, is the Jacobian of a linear transformation, which evaluates to the Kronecker product $J_S = (\boldsymbol{W}^{(l)T} \otimes \mathrm{diag}(\boldsymbol{V}_P \boldsymbol{\theta}^{(l)}))$, since $\mathrm{vec}(\widehat{S}^{(l)}) = \left((\boldsymbol{W}^{(l)})^\top \otimes \mathrm{diag}(\boldsymbol{V}_P \boldsymbol{\theta}^{(l)})\right)\mathrm{vec}(\widehat{\boldsymbol{H}}^{(l)})$. The first term, J_τ, is the Jacobian of the transformed activation, which is $J_\tau = (\boldsymbol{I} \otimes \boldsymbol{U}^\top)\mathrm{diag}(\mathrm{vec}(\sigma'(\boldsymbol{U}\widehat{S}^{(l)})))(\boldsymbol{I} \otimes \boldsymbol{U})$. We can now bound the norm of the product:

$$\begin{aligned}
\|\widehat{\mathcal{J}}^{(l)}\|_2 &\leq \|J_\tau\|_2 \cdot \|J_S\|_2 \\
&= \|\mathrm{diag}(\mathrm{vec}(\sigma'(\boldsymbol{U}\widehat{S}^{(l)})))\|_2 \cdot \|\boldsymbol{W}^{(l)T} \otimes \mathrm{diag}(\boldsymbol{V}_P \boldsymbol{\theta}^{(l)})\|_2 && \text{(Unitary invariance of } J_\tau) \\
&= \left(\max_{i,j} |\sigma'((\boldsymbol{U}\widehat{S}^{(l)})_{ij})|\right) \cdot \|\boldsymbol{W}^{(l)}\|_2 \cdot \|\mathrm{diag}(\boldsymbol{V}_P \boldsymbol{\theta}^{(l)})\|_2. && \text{(Norm of diag and Kronecker)}
\end{aligned}$$

Since σ is α-Lipschitz, its derivative is bounded by α. The norm of the diagonal filter matrix is bounded by its largest entry, which via Cauchy-Schwarz is $|\langle \boldsymbol{v}_i, \boldsymbol{\theta}^{(l)} \rangle| \leq \|\boldsymbol{v}_i\|_2 \|\boldsymbol{\theta}^{(l)}\|_2$. This gives $\|\mathrm{diag}(\boldsymbol{V}_P \boldsymbol{\theta}^{(l)})\|_2 \leq \|\boldsymbol{\theta}^{(l)}\|_2 \|\boldsymbol{V}_P\|_{2,\infty}$. Substituting these bounds yields:

$$\|\widehat{\mathcal{J}}^{(l)}\|_2 \leq \alpha \|\boldsymbol{W}^{(l)}\|_2 \|\boldsymbol{\theta}^{(l)}\|_2 \|\boldsymbol{V}_P\|_{2,\infty}. \tag{43}$$

Finally, we substitute the per-layer bound into the product from Step 2:

$$\|\mathcal{J}\|_2 = \|\widehat{\mathcal{J}}\|_2 \leq \prod_{l=0}^{L-1} \|\widehat{\mathcal{J}}^{(l)}\|_2 \leq \prod_{l=0}^{L-1} \left(\alpha \|\boldsymbol{W}^{(l)}\|_2 \|\boldsymbol{\theta}^{(l)}\|_2 \|\boldsymbol{V}_P\|_{2,\infty}\right). \tag{44}$$

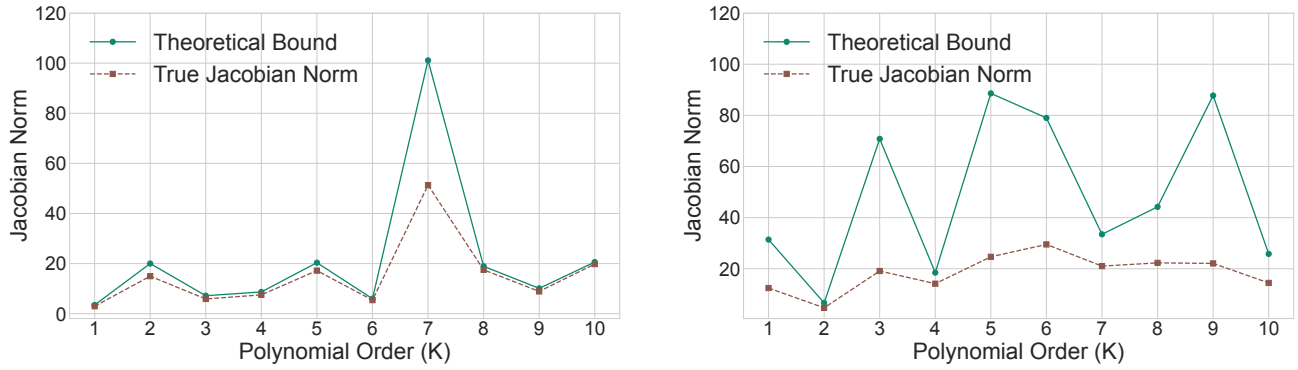
Applying the parameter constraints $\|\boldsymbol{W}^{(l)}\|_2 \leq C_{W,l}$ and $\|\boldsymbol{\theta}^{(l)}\|_2 \leq C_{\theta,l}$ completes the proof. □

G EMPIRICAL TIGHTNESS OF THE JACOBIAN BOUND

To empirically evaluate the tightness of the stability bound derived in Theorem 5, we computed both our theoretical upper bound and the true spectral norm of the network Jacobian.

We evaluated a 2-layer nonlinear model utilizing the Monomial basis across varying polynomial orders ($K \in [1, 10]$) on both the Cora and Chameleon datasets. The true spectral norm of the network Jacobian was computed exactly using power iteration via automatic differentiation (to avoid materializing the full matrix in memory). This empirical true norm was then compared directly against our theoretical bound given by $\prod_{l=0}^{L-1} (\alpha C_{W,l} C_{\theta,l} \|\boldsymbol{V}_P\|_{2,\infty})$.

As illustrated in Figure 2, our theoretical bound tracks the true Jacobian norm reliably. On the Cora dataset, the bound is exceptionally tight, remaining within a small factor (1.0x to 2.0x) of the true norm, and it correctly identifies the location of the instability spike at $K = 7$. On the Chameleon dataset, it successfully bounds the true norm within a factor of 1.3x to 4.0x. While the bound slightly overestimates the magnitude of instability at certain orders on Chameleon, it provides a tight measure of the model's worst-case sensitivity.



(a) Cora (b) Chameleon

Figure 2: Jacobian bound tightness.

H CORRELATION BETWEEN THE THEORETICAL BOUND AND THE GENERALIZATION GAP

To compare theory and measurements under controlled conditions, we keep the architecture minimal and the training setup fixed across all plots. Unless noted, we use the spectral layer in Eq. (8) *without* activations for the linear setting (Section 5) and the standard model *with* activations for the nonlinear setting. The hyperparameters are the following: learning rate 0.01, weight decay 10^{-5}, maximum 500 epochs with early stopping after 100 epochs without validation improvement. Results are averaged over 30 random splits (compared to 10 random splits in Section 7). The generalization gap is defined as Gap = test_loss − train_loss. No extra regularizers (beyond weight decay) are used in these plots. Error bars in the accuracy panels are standard deviations across splits.

For the linear experiments (Cora and Chameleon, $L \in \{1, 2\}$), we compare directly against the tighter linear FTGC bound (Theorem 4), which isolates depth, basis amplification, and the input spectrum (see Figures 3 and

5). The bound is evaluated via

$$\mathcal{G}_V(\mathcal{F}) \;\le\; \frac{1}{n}\left(\prod_{l=0}^{L-1} C_{W,l}C_{\theta,l}\right)\left(\sum_{i=1}^{n}\|\boldsymbol{v}_i\|_2^{2L}\,\mathcal{E}_0(\lambda_i)\right)^{1/2}.$$

For the nonlinear experiments ($L = 2$ only), we use the nonlinear bound from Theorem 3 (see Figures 4 and 6). We only show $L = 2$ for the nonlinear case because with $L = 1$ the neural network is effectively linear and does not add anything new to this comparison. Across all runs, the models fit the labeled nodes well. On Chameleon, the minimum training accuracy is 76%, and on Cora, it is 94%. Since these are much larger than their corresponding test accuracies, the positive gaps are not due to underfitting.

Sensitivity Analysis. To verify that our Fourier Transductive Gaussian Complexity (FTGC) bound truly captures the generalization gap via spectral properties rather than just parameter norms, we performed a sensitivity analysis by decomposing the bound into the Weight Term ($\prod \alpha C_{W,l}C_{\theta,l}$) and the Spectral Interaction Term ($\|\boldsymbol{V}_P\|_{2,\infty}\sqrt{\sum\|\boldsymbol{v}_i\|_2^2\mathcal{E}_0(\lambda_i)}$). For a 2-layer nonlinear model on Cora, the Pearson correlation drops drastically from 0.86 (Full Bound, Figure 4a) to -0.19 when using only the Weight Term. Similarly, on Chameleon, the correlation drops from 0.36 (Figure 6a) to -0.53. This confirms that the data-dependent spectral term is the crucial driver for predicting the generalization gap, validating our Fourier-domain analysis.

Large-Scale Validation on ogbn-arxiv. To verify that our theoretical framework scales to larger networks, we extended our empirical validation to the ogbn-arxiv dataset (approximately 170,000 nodes). We maintained the same highly sparse label setting used for the smaller datasets (only 10 training nodes per class) to rigorously stress-test our bound. Using the practical $L = 2$ nonlinear model, we computed the FTGC bound across different polynomial bases. As shown in Figure 7, we observed a strong positive correlation, achieving a Pearson coefficient of 0.85 and a Spearman rank coefficient of 0.71. This confirms that our bound robustly tracks generalization behavior even on massive graphs.

Main observations. The results shown in Figures 3–7 suggest the following main observations:

1. In the bound–gap panels, we observe a clear positive correlation between our bound and the empirical gap on all evaluated datasets. We report the overall Pearson r and Spearman ρ (with 95% CIs by Fisher transform) inside panels (a) and (c) of each figure.

2. The correlation is visibly stronger for $L = 2$ than for $L = 1$ (Figures 3 and 5). This matches Theorem 4, as the basis term scales as $\|\boldsymbol{v}_i\|_2^{2L}$ and magnifies differences across bases and spectra with depth.

3. Points cluster by polynomial basis in the bound–gap plane, reflecting their amplification profiles $x \mapsto \sum_{k=0}^{K} P_k(x)^2$ (Figure 1 of the main paper). Furthermore, stable bases like Chebyshev exhibit tighter correlations within their own group. This predictable behavior occurs because the spectral amplification profile of Chebyshev is relatively uniform (nearly constant) across the spectrum, preventing extreme fluctuations in the theoretical bound.

4. Chebyshev's relatively uniform profile does not downweight any part of the spectrum. Because it amplifies all frequencies nearly equally (including the middle spectrum), the term $\sum_i \|\boldsymbol{v}_i\|_2^{2L}\mathcal{E}_0(\lambda_i)$ becomes strictly larger when $\mathcal{E}_0(\lambda_i)$ spreads broadly. This perfectly explains why it tends to yield both higher theoretical bounds and higher empirical generalization gaps (as seen in the top-right clusters of the plots).

5. Cora is homophilous, meaning low-frequency content is more predictive. Bases with stronger low-pass behavior (more U-shaped amplification in Figure 1 of the main paper) tend to achieve higher test accuracy.

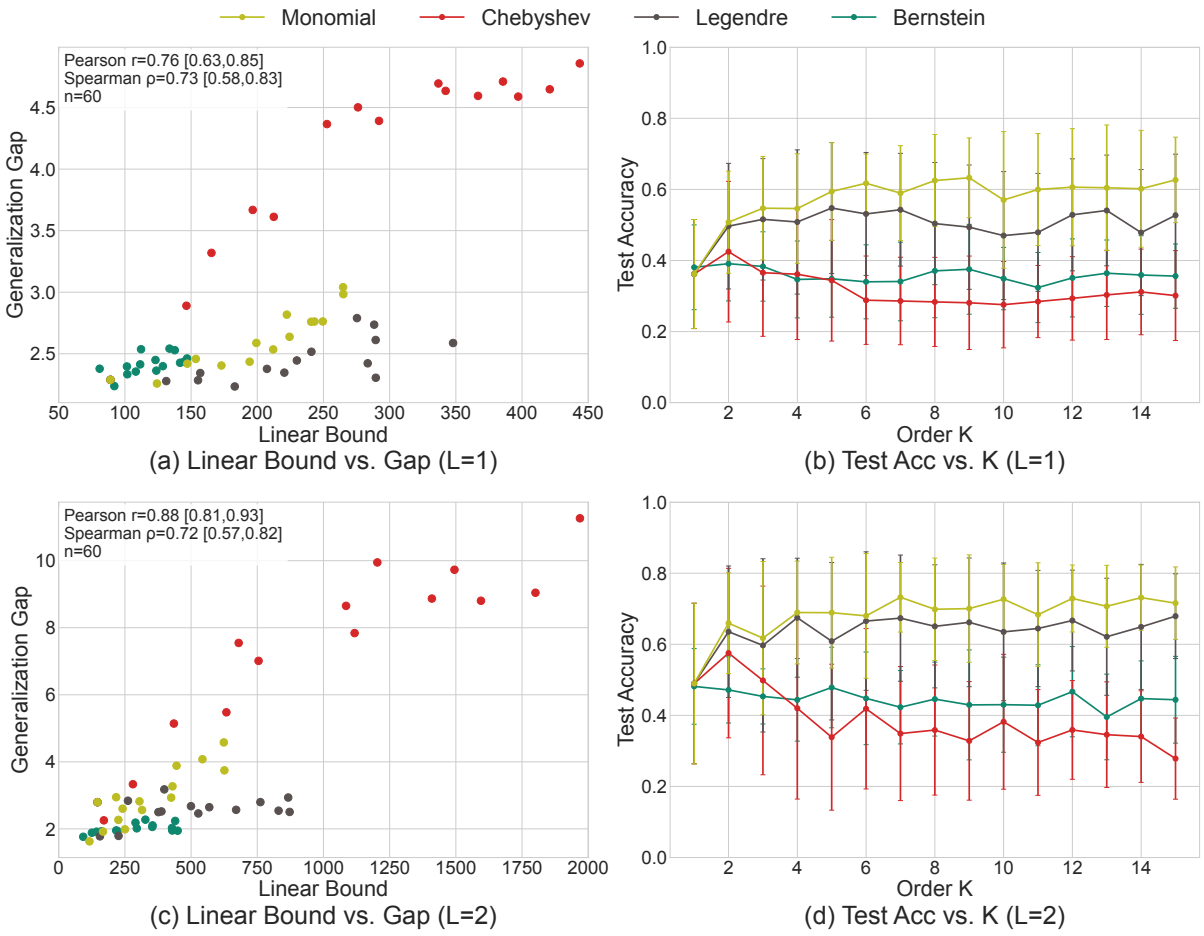


Figure 3: Cora linear models.

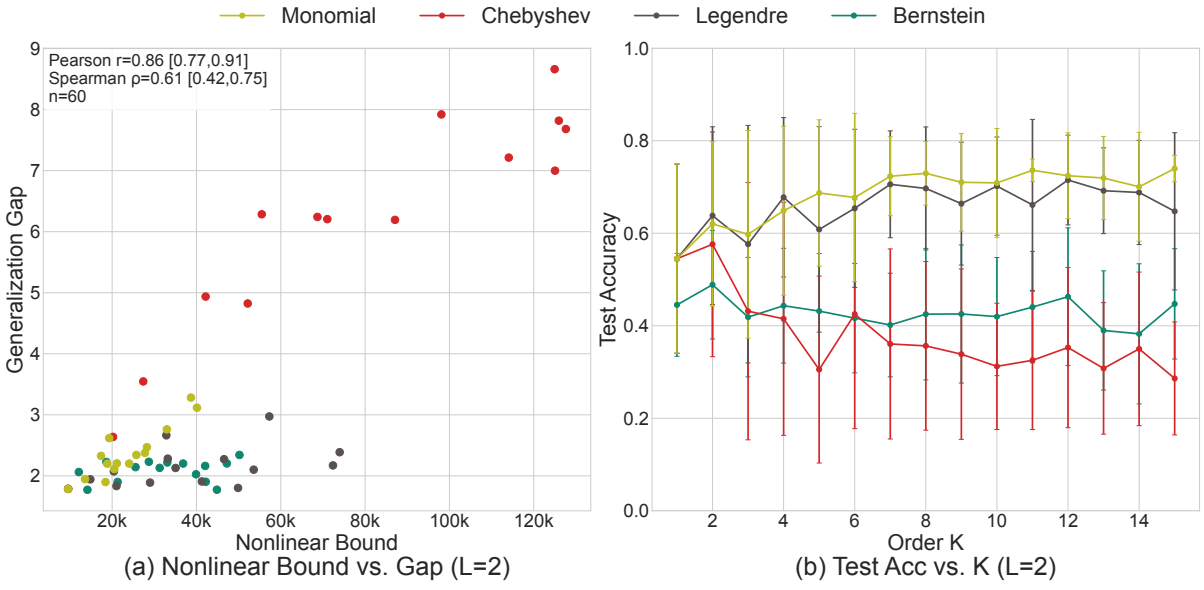


Figure 4: Cora nonlinear model.

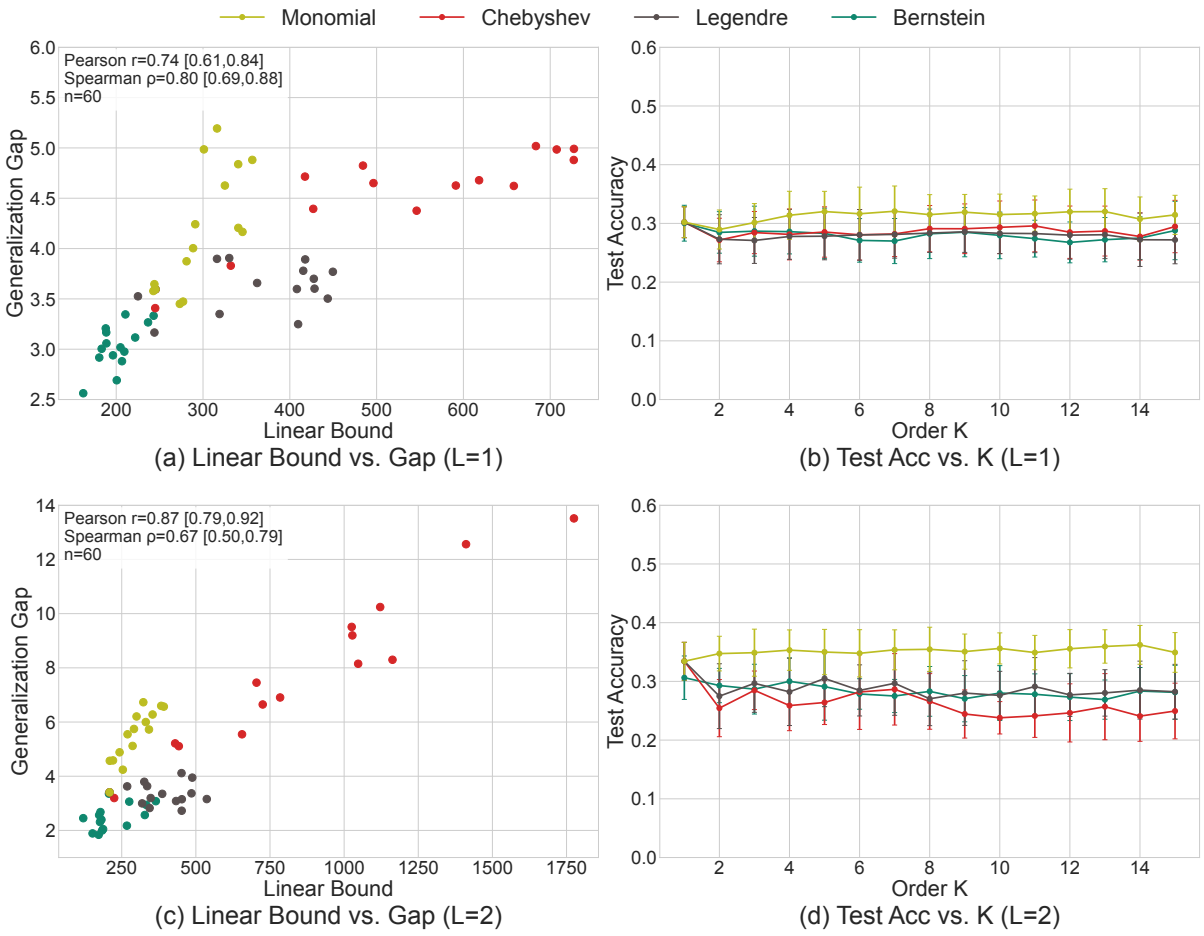


Figure 5: Chameleon linear models.

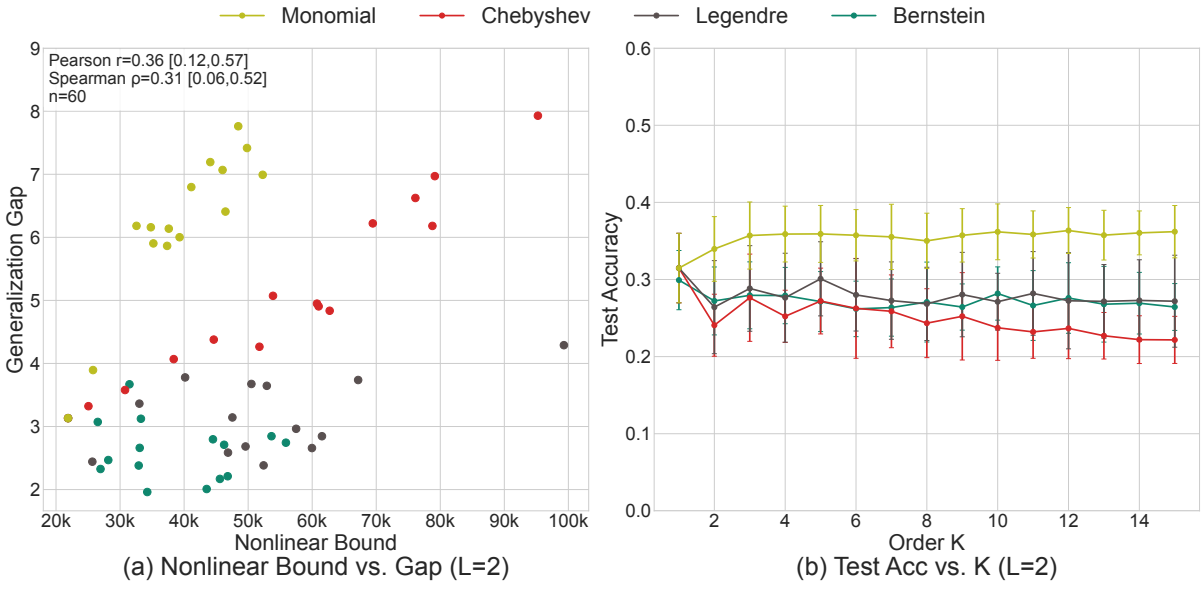


Figure 6: Chameleon nonlinear model.

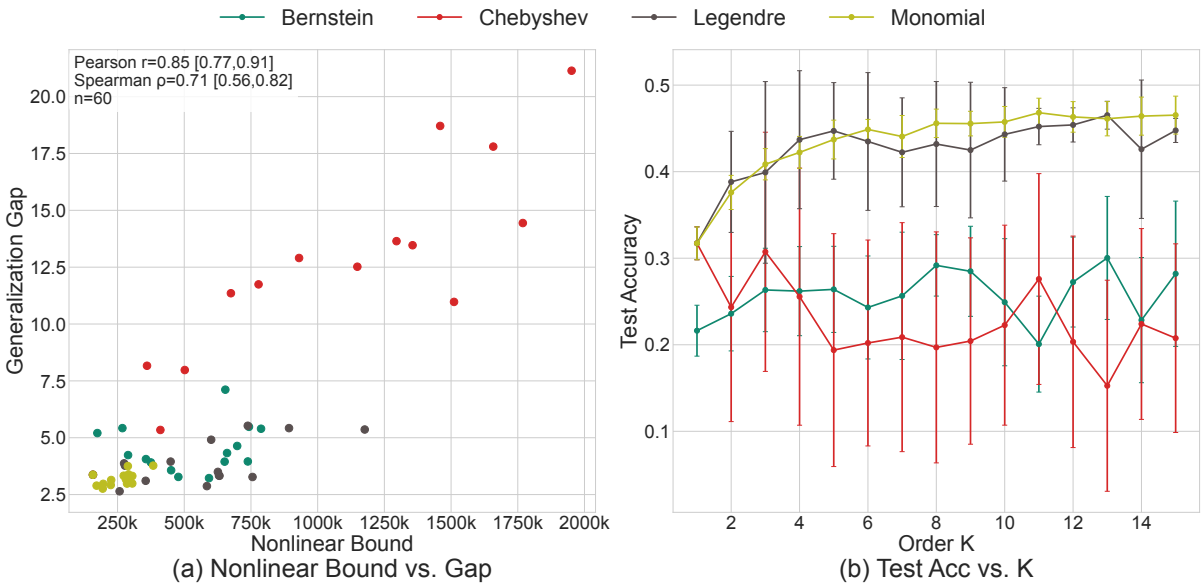
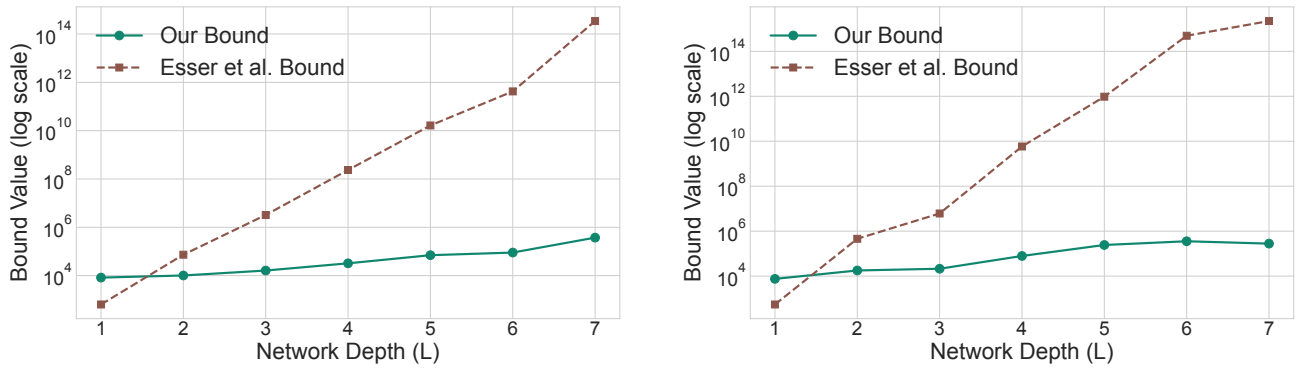


Figure 7: Ogbn-arxiv nonlinear model.

I COMPARISON WITH PRIOR TRANSDUCTIVE BOUNDS

To quantitatively evaluate the tightness of our proposed Fourier-Transductive Gaussian Complexity (FTGC) bound, we compare it against the generalization bound derived specifically for GNNs by Esser et al. (2021). For a fair comparison, we employ a standard GCN architecture, as the theoretical analysis by Esser et al. (2021) is limited to GCNs. We calculate both bounds across varying network depths ($L \in [1, 7]$) on the Cora and Chameleon datasets.

As shown in Figure 8, as network depth increases, the spatial bound from Esser et al. (2021) suffers from exponential amplification due to its dependency on the spatial graph shift operator ($\|\hat{\boldsymbol{A}}\|_\infty$). By depth 7, this bound becomes completely vacuous ($\sim O(10^{14})$). In contrast, our FTGC bound depends on the spectral norm of the generalized Vandermonde matrix ($||\boldsymbol{V}_P||_{2,\infty}$), which is 1 for GCNs. Consequently, our bound remains significantly tighter ($\sim O(10^5)$) and avoids exponential explosion, while still successfully capturing the generalization trends demonstrated earlier in App. H.



(a) Cora (b) Chameleon

Figure 8: Comparison of generalization bounds.

J FORMAL DEFINITIONS AND PROPERTIES OF THE POLYNOMIAL BASES

This section provides the formal definitions for the polynomial bases used in our analysis. The key property for our bounds is the filter's amplification profile, which measures the total response of the basis functions at a given frequency $x \in [-1, 1]$:

$$M_K(x) = \sum_{k=0}^{K} P_k(x)^2.$$

This quantity directly relates to the maximum row norm of the Vandermonde matrix $\boldsymbol{V}_P$, which appears in our generalization and stability bounds:

$$\|\boldsymbol{V}_P\|_{2,\infty} = \max_i \|\boldsymbol{v}(\lambda_i)\|_2 = \max_{\lambda \in \Lambda} \sqrt{M_K(\lambda)}.$$

For all four bases we consider, the amplification profile $M_K(x)$ reaches its maximum on $[-1, 1]$ at the endpoints $x = \pm 1$.

Monomial Basis. This is the simplest polynomial basis, also known as the power basis. While straightforward, it is known to be poorly conditioned for high orders K, leading to numerical instability (Higham, 2002). The basis functions are defined as:

$$P_k(x) = x^k, \quad \text{for } k = 0, 1, \ldots, K. \tag{45}$$

Its amplification profile is a geometric series, $M_K(x) = \sum_{k=0}^{K} x^{2k}$, which reaches its maximum value of $M_K(\pm 1) = K + 1$.

Chebyshev Basis. The Chebyshev polynomials $T_k(x)$ are a sequence of orthogonal polynomials known for their numerical stability and role in approximation theory (Mason and Handscomb, 2002). They are defined on the interval $[-1, 1]$ by the recurrence relation:

$$T_0(x) = 1, \tag{46}$$
$$T_1(x) = x, \tag{47}$$
$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x), \quad \text{for } k \geq 1. \tag{48}$$

Since $T_k(\pm 1)^2 = 1$ and $|T_k(x)| \leq 1$ for all k, the amplification profile reaches its maximum of $M_K(\pm 1) = K + 1$ at the endpoints.

Legendre Basis. Legendre polynomials $P_k(x)$ are another sequence of orthogonal polynomials on $[-1, 1]$, notable for being solutions to Legendre's differential equation (Abramowitz and Stegun, 1964). They are defined by Bonnet's recurrence relation:

$$P_0(x) = 1, \tag{49}$$
$$P_1(x) = x, \tag{50}$$
$$(k+1)P_{k+1}(x) = (2k+1)xP_k(x) - kP_{k-1}(x), \quad \text{for } k \geq 1. \tag{51}$$

Just like the Chebyshev polynomials, $P_k(\pm 1)^2 = 1$ and $|P_k(x)| \leq 1$, so the amplification profile has a maximum of $M_K(\pm 1) = K + 1$ at the endpoints.

Bernstein Basis. The Bernstein polynomials are known for their shape-preserving properties and numerical stability, particularly near interval endpoints (Lorentz, 2012). For the spectral domain $[-1, 1]$, they are defined using an affine transformation $t = (x + 1)/2$:

$$P_k(x) = \binom{K}{k} t^k (1-t)^{K-k}, \quad \text{for } k = 0, 1, \ldots, K. \tag{52}$$

Because these polynomials are non-negative and sum to 1, their sum of squares is bounded by 1. The maximum amplification is $M_K(\pm 1) = 1$, which occurs at the endpoints where only one basis polynomial is non-zero.

Summary of Maximum Amplification. These maximum values directly determine the worst-case amplification term in our bounds. For a filter of order K, we get:

$$\|\boldsymbol{V}_P\|_{2,\infty} = \max_{x\in\Lambda} \sqrt{M_K(x)} \leq \max_{x\in[-1,1]} \sqrt{M_K(x)} = \begin{cases} \sqrt{K+1} & \text{for Monomial, Chebyshev, Legendre} \\ 1 & \text{for Bernstein.} \end{cases}$$

This shows that the depth-dependent terms in our bounds grow with $\sqrt{K+1}$ for the first three bases, but remain constant for the Bernstein basis.

To match worst-case amplification across bases, we can rescale

$$\tilde{P}_k(x) = \frac{P_k(x)}{\sqrt{\max_{x\in[-1,1]} M_K(x)}},$$

so that $\max_x \sum_{k=0}^{K} \tilde{P}_k(x)^2 = 1$ for all four bases and $\|\boldsymbol{V}_P\|_{2,\infty} \leq 1$ (for fixed K). This keeps parameter-norm constraints comparable across bases.

K HYPERPARAMETERS, DATASETS, AND SPLITS

K.1 Datasets and Splits

We conduct experiments on four public node classification datasets: the homophilic citation networks Cora and Citeseer (Sen et al., 2008), and the heterophilic Wikipedia networks Chameleon and Squirrel (Pei et al., 2020). For the large-scale bound validation in App. H, we also use the OGBN-Arxiv dataset (Hu et al., 2020), which represents a much larger citation network. For all datasets, we use 10 random stratified splits where each split contains 10 training nodes per class, with the remaining nodes divided into a validation set (35%) and a test set (65%). We report the mean and 95% confidence interval over these 10 splits.

K.2 Model and Training

Our model architecture implements a spectral filter layer with a residual connection, inserted between the two linear layers of an MLP. Specifically, input features are first processed by a linear layer and a ReLU activation to produce hidden feature. These features are then filtered in the spectral domain, and the result is added back to the original hidden features. A final linear layer then acts as a readout to produce the class logits. We use two distinct dropout rates, which are tuned separately. The first is applied to the input features and to the features after the spectral filtering block. The second is applied to the hidden features after the ReLU activation.

We use the Adam optimizer (Kingma and Ba, 2015) and train for a maximum of 2000 epochs, with early stopping triggered if the validation accuracy does not improve for 200 epochs.

K.3 Hyperparameter Optimization

For each dataset and polynomial basis, we conduct two separate hyperparameter searches using the Hyperopt library's Tree-structured Parzen Estimator (TPE) algorithm (Bergstra et al., 2011), optimizing for the highest mean validation accuracy. The first search optimizes a baseline model where the regularizer strength λ_{EW} is fixed at 0. The second search optimizes the fully regularized model over the complete hyperparameter space listed below.

- **Learning Rate**: $\{0.0005, 0.001, 0.005, 0.01, 0.05, 0.1\}$
- **Weight Decay**: $\{0, 10^{-6}, 5\cdot10^{-6}, 10^{-5}, 5\cdot10^{-5}, 10^{-4}, 5\cdot10^{-4}, 10^{-3}\}$
- **Hidden Dimension**: $\{8, 16, 32, 64, 128\}$
- **Polynomial Degree**: $\{10\}$
- **Dropout 1 (Input/Post-Filter)**: $\{0.0, 0.2, 0.4, 0.6, 0.8\}$
- **Dropout 2 (Hidden)**: $\{0.0, 0.2, 0.4, 0.6, 0.8\}$
- **Regularization Strength** (λ_{EW}): $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10\}$

K.4 Code and Infrastructure

All experiments were run on a single NVIDIA A100 GPU. The complete code to reproduce our results is available on this link: `https://github.com/vmart20/spectral-GNN-regularization`