# Long-Range Graph Wavelet Networks

**Filippo Guerranti**[1,2,3] **Fabrizio Forte**[1] **Simon Geisler**[1,2,*] **Stephan Günnemann**[1,2,3]

[1] School of Computation, Information and Technology, Technical University of Munich
[2] Munich Data Science Institute [3] Munich Center for Machine Learning (MCML)

{f.guerranti,f.forte,s.geisler,s.guennemann}@tum.de

## Abstract

Modeling long-range interactions, the propagation of information across distant parts of a graph, is a central challenge in graph machine learning. Graph wavelets, inspired by multi-resolution signal processing, provide a principled way to capture both local and global structures. However, existing wavelet-based graph neural networks rely on finite-order polynomial approximations, which limit their receptive fields and hinder long-range propagation. We propose Long-Range Graph Wavelet Networks (LR-GWN), which decompose wavelet filters into complementary local and global components. Local aggregation is handled with efficient low-order polynomials, while long-range interactions are captured through a flexible spectral-domain parameterization. This hybrid design unifies short- and long-distance information flow within a principled wavelet framework. Experiments show that LR-GWN achieves state-of-the-art performance among wavelet-based methods on long-range benchmarks, while remaining competitive on short-range datasets.

## 1 Introduction

Long-range interactions are central to complex systems, from electron correlations in quantum chemistry [2, 25] to allosteric effects in biology [12, 48]. In graph neural networks (GNNs) [21, 40, 19, 8], capturing these interactions requires models that can propagate information beyond local neighborhoods without the computational overhead of dense global interactions [1, 14].

Wavelet-based graph neural networks (WGNNs) [22], inspired by wavelet theory [33], provide a principled framework for this challenge, defining spectral filters that can in principle capture different scales of information propagation through their filtering characteristics. However, designing exact wavelet filters face a fundamental computational bottleneck, as it would require computing the full eigenvalue decomposition of the graph Laplacian, which is prohibitive for large graphs.

To circumvent this cost, current WGNNs [22, 47, 28] approximate wavelet filters using low-order polynomials. While computationally efficient, this approach suffers from two critical limitations. First, polynomial filters aggregate information only within a finite number of hops [4], inherently restricting their spatial reach. More fundamentally, polynomial approximations face an inherent trade-off between computational efficiency and functional expressiveness: while high-degree polynomials can theoretically approximate any function defined on the real interval (Weierstrass theorem), practically feasible orders cannot accurately represent the discontinuous or steep characteristics needed for selective frequency filtering, such as wavelets [18].

Achieving both computational efficiency and sufficient functional expressiveness requires rethinking wavelet filter parametrization. Current approaches fail to resolve this trade-off: low-order polynomials

[*] Now at Google Research.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: New Perspectives in Advancing Graph Machine Learning.

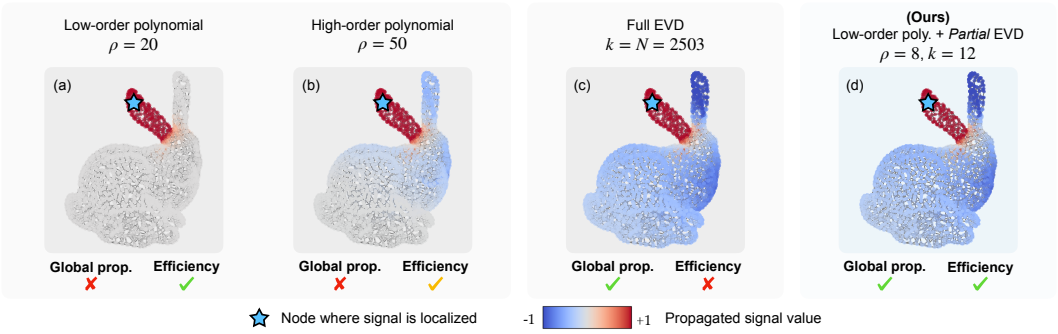


Figure 1: *Signal propagation under different Mexican hat wavelet filter approximations.* (a) Low-order polynomial ($\rho = 20$) restricts propagation to local neighborhoods. (b) Higher-order polynomial ($\rho = 50$) extends reach but remains spatially bounded. (c) Full EVD ($k = N = 2503$) enables global propagation at prohibitive cost. (d) LR-GWN ($\rho = 8, k = 12$) achieves global propagation with minimal computational overhead.

confine propagation to local neighborhoods (Fig. 1a), while higher orders extend the radius at additional computational cost, but still fail to achieve global propagation (Fig. 1b). In contrast, the exact solution via full EVD enables global information flow but at prohibitive cost (Fig. 1c).

We propose Long-Range Graph Wavelet Networks (LR-GWN), which overcome the limitations of polynomial-based WGNNs through a hybrid filter design. Our method decomposes each wavelet filter into two components: a low-order polynomial for efficient local aggregation and a spectral filter operating on the low-frequency eigenspace to enable long-range propagation (Fig. 1d). This approach requires only a partial eigendecomposition at preprocessing time, making it computationally practical while achieving the global reach that purely polynomial methods cannot provide.

LR-GWN can operate under strict wavelet theory or with relaxed constraints for enhanced performance, providing a principled yet flexible framework for wavelet-based long-range graph learning.

**Our contributions** are: (i) a **hybrid parametrization** combining polynomial spatial filters with learnable spectral filters operating on truncated eigenspaces; (ii) a **principled wavelet framework** that maintains theoretical admissibility constraints while optionally allowing relaxation when needed; (iii) an **efficient implementation** requiring only partial EVD, adding minimal preprocessing overhead; and (iv) **state-of-the-art performance** among wavelet-based GNNs on long-range benchmarks.

## 2 Background

We recall the main tools from spectral graph theory and wavelet analysis; we refer the reader to Appendix A for a detailed exposition on these topics.

**Graphs.** We consider undirected graphs $\mathcal{G} = (\boldsymbol{A}, \boldsymbol{X})$ with adjacency matrix $\boldsymbol{A} \in \{0,1\}^{n\times n}$ and node features $\boldsymbol{X} \in \mathbb{R}^{n\times d}$. Throughout we use the symmetrically normalized Laplacian $\mathcal{L} = \boldsymbol{I} - \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$, with $\boldsymbol{D} = \mathrm{diag}(\boldsymbol{A}\mathbf{1})$, which admits eigendecomposition $\mathcal{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$ with orthogonal eigenvectors $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ and eigenvalues $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^{n\times n}$, $0 = \lambda_1 \leq \cdots \leq \lambda_n \leq 2$ [10]. The eigenvectors of $\mathcal{L}$ form an orthogonal basis, often called the graph Fourier basis. Any graph signal $\boldsymbol{x} \in \mathbb{R}^n$ can be decomposed in this basis via the graph Fourier transform (GFT) as $\widehat{\boldsymbol{x}} = \boldsymbol{U}^\top\boldsymbol{x}$, with inverse GFT $\boldsymbol{x} = \boldsymbol{U}\widehat{\boldsymbol{x}}$. Spectral filtering applies a kernel $g : \mathbb{R} \to \mathbb{R}$ by modulating its Fourier coefficients: $g *_{\mathcal{G}} \boldsymbol{x} = \boldsymbol{U}\,\widehat{g}(\boldsymbol{\Lambda})\,\boldsymbol{U}^\top\boldsymbol{x}$, where $\widehat{g}(\boldsymbol{\Lambda}) = \mathrm{diag}(\widehat{g}(\lambda_1), \ldots, \widehat{g}(\lambda_n))$. Small eigenvalues correspond to smooth/global variations, large ones to oscillatory/local patterns [26, see Fig. 3 for an intuitive visualization].

**Wavelets.** Wavelet transforms extend classic Fourier analysis by providing joint localization in both frequency and space [33, 34]. A mother wavelet $\psi$ generates dilated and translated functions $\psi_{s,l}(t) = s^{-1/2}\psi\big(\frac{t-l}{s}\big)$, whose coefficients $W_f(s,l) = \int_{\mathbb{R}} f(t)\psi^*_{s,l}(t)dt$ decompose a signal $f$ across scales $s$ and locations $l$. Reconstruction requires the admissibility condition, $\widehat{\psi}(0) = 0$, ensuring $\widehat{\psi}$ acts as a band-pass filter. A complementary scaling function $\phi$ with $\widehat{\phi}(0) > 0$ provides low-pass coverage, yielding a multiresolution analysis where wavelets capture localized variations and the scaling function captures global trends.

**Wavelets on graphs.** The spectral graph wavelet transform (SGWT) [22] adapts wavelet analysis to graphs using the spectrum of $\mathcal{L}$. Since graphs lack translation and scaling, wavelets are defined spectrally: a band-pass kernel $\widehat{\psi}$ yields the wavelet operator $\Psi = \boldsymbol{U}\,\widehat{\psi}(\boldsymbol{\Lambda})\boldsymbol{U}^\top$, and dilation to scale $s$ is obtained by $\widehat{\psi}(s\boldsymbol{\Lambda})$. A complementary scaling function defines $\Phi = \boldsymbol{U}\,\widehat{\phi}(\boldsymbol{\Lambda})\boldsymbol{U}^\top$. For a signal $\boldsymbol{x}$, the wavelet coefficients at scale $s$ are $W_{\boldsymbol{x}}(s) = \boldsymbol{U}\,\widehat{\psi}(s\boldsymbol{\Lambda})\boldsymbol{U}^\top\boldsymbol{x}$, while scaling coefficients are $H_{\boldsymbol{x}} = \boldsymbol{U}\,\widehat{\phi}(\boldsymbol{\Lambda})\boldsymbol{U}^\top\boldsymbol{x}$. Together, $W_{\boldsymbol{x}}(s)$ and $H_{\boldsymbol{x}}$ provide a multiscale representation, where $\widehat{\psi}$ captures localized variations (band-pass filter) and $\widehat{\phi}$ global structure (low-pass filter).

**Polynomial approximations.** Exact spectral filtering requires full eigendecomposition, which is impractical for large graphs. A common workaround is to approximate filters with low-order polynomials of $\mathcal{L}$. Chebyshev expansions are particularly effective [11]: $\widehat{g}_{\boldsymbol{\omega}}(\boldsymbol{\Lambda}) = \sum_{i=0}^{\rho}\omega_i T_i(\tilde{\boldsymbol{\Lambda}})$, where $\boldsymbol{\omega} \in \mathbb{R}^{\rho+1}$, $\tilde{\boldsymbol{\Lambda}} = (2/\lambda_{\max})\boldsymbol{\Lambda} - \boldsymbol{I}$, and $T_i$ are Chebyshev polynomials defined recursively by $T_0(x) = 1,\ T_1(x) = x,\ T_i(x) = 2xT_{i-1}(x) - T_{i-2}(x)$. Such filters require no eigendecomposition and correspond to message passing within at most $\rho$ hops on the graph [4].

## 3 Limitations of Polynomial Graph Filters

Approximating spectral filters with polynomials avoids explicit eigendecomposition, since powers of $\mathcal{L}$ act directly in the vertex domain. For example, a monomial expansion $\widehat{g}_{\boldsymbol{\omega}}(\boldsymbol{\Lambda}) = \sum_{i=0}^{\rho}\omega_i\boldsymbol{\Lambda}^i$ yields the spatial operator $g_{\boldsymbol{\omega}}(\mathcal{L}) = \sum_{i=0}^{\rho}\omega_i\mathcal{L}^i$. The Chebyshev basis follows an analogous reasoning (see Appendix F.1). While computationally attractive, such filters face inherent expressivity barriers that explain the limitations of many GNN architectures built upon them. We outline these issues from three complementary perspectives: (*i*) message propagation, (*ii*) function approximation, and (*iii*) convergence to discontinuous filters.

**Perspective I: Propagation.** Balcilar et al. [4] show that an $l$-layer message-passing GNN corresponds exactly to an $l$-order polynomial filter. In message-passing networks, each layer typically uses a first-order update of the form $\boldsymbol{H}^{(l)} = (\omega_0\boldsymbol{I} + \omega_1\mathcal{L})\boldsymbol{H}^{(l-1)}$, where each application of $\mathcal{L}$ aggregates information from one-hop neighbors [24]. Stacking $l$ such layers yields a polynomial filter of order $l$, enabling information flow across at most $l$ hops. Beyond this range, information is inaccessible, and increasing depth triggers *over-squashing* [1], where distant signals collapse into compressed representations. Thus polynomial filters cannot achieve effective long-range propagation without prohibitive computational costs or architectural limitations.

**Perspective II: Approximation.** Polynomials of low degree are also weak function approximators. By Markov's inequality [35], the slope of a degree-$\rho$ polynomial on $[-1, 1]$ is bounded as $\max_{x\in[-1,1]}\left|P'(x)\right| \leq \rho^2$, which restricts the ability of the polynomial to approximate functions with sharp and steep transitions. Smooth functions can be represented with modest degree, but scaling functions and low-scale wavelets, which exhibit steep cutoffs, require prohibitively high $\rho$. This confines polynomial filters to overly smooth spectral responses.

**Perspective III: Convergence.** The most fundamental limitation arises with discontinuous filters such as ideal band-pass or virtual-node mechanisms. Geisler et al. [18] show that polynomial approximations converge arbitrarily slowly in operator norm, regardless of degree. In other words, there exists no sequence of polynomials that can efficiently approximate such filters across graphs. This establishes a structural gap between polynomial parametrizations and the expressive, frequency-selective operators required for modeling long-range dependencies.

**Summary:** Polynomial filters are efficient but inherently local, spectrally smooth, and cannot represent discontinuous frequency-selective behaviors needed for long-range modeling.

## 4 Long-Range Graph Wavelet Networks

We propose the *Long-Range Graph Wavelet Network (LR-GWN)*, a parametrization of graph wavelet filters that enables efficient, principled, and expressive filtering of graph signals, specifically tailored to capture long-range dependencies. Our design addresses three core limitations of prior approaches: it resolves the locality bottleneck of polynomial filters, preserves the interpretability of pure wavelet propagation, and achieves linear complexity with respect to the number of edges.

### 4.1 Hybrid Parametrization of Wavelet Filters

Effective graph filters must balance *efficiency*, *expressivity*, and *theoretical grounding*. Polynomial filters are attractive because they can be implemented recursively with linear cost in the number of edges. However, their inherent smoothness makes them poorly suited to approximate the sharp spectral transitions characteristic of wavelets. In contrast, purely spectral filters can realize principled wavelet propagation, but computing the full eigendecomposition of the Laplacian is prohibitive.

To reconcile these trade-offs, we introduce a *hybrid parametrization* that combines a polynomial backbone with a spectral correction. We recall that our goal is to parametrize wavelet filters, namely the scaling function $\widehat{\phi}$ and wavelet function $\widehat{\psi}$. At each layer $l$, both the wavelet kernel $\widehat{\psi}^{(l)}$ and the scaling function $\widehat{\phi}^{(l)}$ are modeled as the sum of local and global contributions:

$$\widehat{\psi}^{(l)}(\boldsymbol{\Lambda}) = P_{\boldsymbol{\omega}_{\psi}^{(l)}}(\boldsymbol{\Lambda}) + S_{\boldsymbol{\theta}_{\psi}^{(l)}}(\boldsymbol{\Lambda}), \qquad \widehat{\phi}^{(l)}(\boldsymbol{\Lambda}) = P_{\boldsymbol{\omega}_{\phi}^{(l)}}(\boldsymbol{\Lambda}) + S_{\boldsymbol{\theta}_{\phi}^{(l)}}(\boldsymbol{\Lambda}), \tag{1}$$

where $P_{\boldsymbol{\omega}_{\bullet}^{(l)}}$ is a finite-order polynomial (spatial/local component) and $S_{\boldsymbol{\theta}_{\bullet}^{(l)}}$ is a learnable spectral (global) component. The polynomial term efficiently captures localized interactions, while the spectral term affords precise control over selected frequency ranges, crucial for long-range propagation.

**Lemma 1.** *For a graph signal $\boldsymbol{x} \in \mathbb{R}^n$ and a filter kernel $\kappa$ parametrized as in Eq. (1), the filtering operation can be expressed as*

$$\kappa * \boldsymbol{x} \;=\; \boldsymbol{U}[P(\boldsymbol{\Lambda}) + S(\boldsymbol{\Lambda})]\,\boldsymbol{U}^\top \boldsymbol{x} \;=\; P(\mathcal{L})\boldsymbol{x} \;+\; \boldsymbol{U}S(\boldsymbol{\Lambda})\boldsymbol{U}^\top \boldsymbol{x}.$$

This decomposition highlights the dual nature of our filtering mechanism: the polynomial term $P(\mathcal{L})\boldsymbol{x}$ acts directly on $\mathcal{L}$ (i.e., the vertex domain), while the spectral correction $\boldsymbol{U}S(\boldsymbol{\Lambda})\boldsymbol{U}^\top\boldsymbol{x}$ acts on the eigenvalues (i.e., the spectral domain) and introduces frequency-selective adjustments. A proof is provided in Appendix F. Accordingly, at layer $l$ we define

$$\psi^{(l)}(\boldsymbol{U}, \boldsymbol{\Lambda}, \mathcal{L})\boldsymbol{x} = \left[P_{\boldsymbol{\omega}_{\psi}^{(l)}}(\mathcal{L}) \;+\; \boldsymbol{U}S_{\boldsymbol{\theta}_{\psi}^{(l)}}(\boldsymbol{\Lambda})\boldsymbol{U}^\top\right]\boldsymbol{x}, \tag{2}$$

$$\phi^{(l)}(\boldsymbol{U}, \boldsymbol{\Lambda}, \mathcal{L})\boldsymbol{x} = \left[P_{\boldsymbol{\omega}_{\phi}^{(l)}}(\mathcal{L}) \;+\; \boldsymbol{U}S_{\boldsymbol{\theta}_{\phi}^{(l)}}(\boldsymbol{\Lambda})\boldsymbol{U}^\top\right]\boldsymbol{x}. \tag{3}$$

**Truncated spectrum.** In practice, we compute a partial eigendecomposition $(\boldsymbol{U}, \boldsymbol{\Lambda}) = \mathrm{EVD}(\mathcal{L}, k)$ of lowest $k$ eigenpairs with $k \ll n$, i.e., $\boldsymbol{U} \in \mathbb{R}^{n\times k}$, $\boldsymbol{\Lambda} \in \mathbb{R}^{k\times k}$. This focuses the spectral component on informative frequencies (e.g., low $\lambda$ for long-range effects) while maintaining scalability. In practice, truncating to the lowest $k$ eigenpairs preserves most of the signal power, since long-range interactions are dominated by low-frequency modes. The decomposition costs $\mathcal{O}(km)$, where $m \ll n^2$ is the number of edges, is computed once at preprocessing time and reused across layers.

**Filter parametrization.** We implement the spatial filter $P(\mathcal{L})$ using a Chebyshev basis for stable, recursive evaluation, and the spectral filter $S(\boldsymbol{\Lambda})$ via Gaussian smearing [41], enabling fine-grained, band-specific control. Implementation details are given in Appendix D.1.

### 4.2 Principled Wavelet Propagation

A central feature of LR-GWN is that propagation is realized entirely through wavelet operators. Each layer applies a scaling function $\widehat{\phi}$ and a family of wavelet functions $\{\widehat{\psi}_j\}_{j=1}^{J}$, followed by pointwise nonlinearities and aggregation. The result is a propagation process that consists *solely* of wavelet-based transformations, preserving an interpretable, wavelet-based filtering operation.

Formally, given input features $\boldsymbol{H}^{(l-1)}$, one LR-GWN layer computes

$$\boldsymbol{H}^{(l)} = \sigma\Big[\phi^{(l)}(\boldsymbol{U}, \boldsymbol{\Lambda}, \mathcal{L})\, f_{\boldsymbol{\vartheta}}^{(l)}(\boldsymbol{H}^{(l-1)})\Big] \;\oplus\; \bigoplus_{j=1}^{J} \sigma\Big[\psi_j^{(l)}(\boldsymbol{U}, \boldsymbol{\Lambda}, \mathcal{L})\, f_{\boldsymbol{\vartheta}}^{(l)}(\boldsymbol{H}^{(l-1)})\Big], \tag{4}$$

where $\sigma$ is a nonlinearity, $\oplus$ denotes aggregation, and $f_{\boldsymbol{\vartheta}}^{(l)}$ is a learnable feature transformation. In practice, we incorporate $f_{\boldsymbol{\vartheta}}^{(l)}(\boldsymbol{H}^{(l-1)})$ into each filter and compute, for the scaling function,

$$\phi^{(l)}(\boldsymbol{U}, \boldsymbol{\Lambda}, \mathcal{L})\, f_{\boldsymbol{\vartheta}}^{(l)}(\boldsymbol{H}^{(l-1)}) = \boldsymbol{U}\Big(S_{\boldsymbol{\theta}_{\phi}^{(l)}}(\boldsymbol{\Lambda}) \odot \big[\boldsymbol{U}^\top f_{\boldsymbol{\vartheta}}^{(l)}(\boldsymbol{H}^{(l-1)})\big]\Big) + P_{\boldsymbol{\omega}_{\phi}^{(l)}}(\mathcal{L})\, f_{\boldsymbol{\vartheta}}^{(l)}(\boldsymbol{H}^{(l-1)}), \tag{5}$$

with the wavelet functions defined analogously via $(\boldsymbol{\omega}_{\psi}^{(l)}, \boldsymbol{\theta}_{\psi}^{(l)})$.

We can ensure the band-pass behavior of the wavelet family, hence its wavelet theoretical validity, by enforcing the admissibility condition $\widehat{\psi}(0) = 0$, which guarantees invertibility of the wavelet transform. In practice, strict admissibility is not always necessary and relaxing it can yield slight empirical gains. A distinctive feature of LR-GWN is that it supports both regimes: admissibility can be enforced when theoretical guarantees are required, or relaxed when flexibility and performance are prioritized. In this context, the theoretical guarantees refer to the invertibility of the transform and the preservation of band-pass behavior under the admissibility condition $\widehat{\psi}(0) = 0$. Enforcing admissibility ensures that LR-GWN defines a valid wavelet transform with full reconstruction capability, while the relaxed variant trades strict invertibility for small empirical gains. We show how to enforce the constraint in Appendix D.2.2.

### 4.3 Computational Efficiency

LR-GWN achieves per layer linear complexity in the number of edges, $\mathcal{O}(m)$, by combining efficient polynomial filters ($\mathcal{O}(\rho m)$) with low-rank spectral projections ($\mathcal{O}(kdn)$). The key insight is that partial eigendecomposition with $k \ll n$ enables global spectral control at $\mathcal{O}(km)$ preprocessing cost, amortized across all layers. This contrasts with purely spectral methods requiring $\mathcal{O}(n^3)$ full EVD or polynomial methods limited to $\rho$-hop neighborhoods. Detailed analysis is provided in Appendix D.3.

## 5 Empirical Results

We evaluate LR-GWN on graph learning tasks requiring both long-range and short-range modeling capabilities. Our evaluation follows two axes: (*i*) comparing methods based on their adherence to pure wavelet propagation, distinguishing theory-compliant from theory-relaxed approaches, and (*ii*) demonstrating performance across interaction scales to validate the generality of our hybrid design.

**Experimental setup.** We conduct between 5 and 10 independent runs per experiment using different random seeds and report the mean performance and standard error of the mean. Our experiments are optimized for resource efficiency, requiring less than 10 GB of GPU memory, making them feasible on widely available hardware such as Nvidia GTX 1080Ti and 2080Ti GPUs. We use the AdamW optimizer [32] with cosine annealing scheduler [31] and linear warmup for stable training. Early stopping based on validation metrics prevents overfitting. All results use the *independent filter parametrization* (see Appendix D.1). We follow standard practice by augmenting node features with positional encodings (PEs) [45], noting that our partial EVD provides PEs for free. Additional implementation details are provided in Appendix D.

**Wavelet methods categorization.** To fairly assess wavelet-based approaches, we distinguish between methods based on their theoretical adherence. *Theory-compliant* (TC) methods satisfy admissibility constraints and propagate entirely through wavelet operations. Only SGWT [22] and LR-GWN (with admissibility) qualify as theory-compliant. *Theory-relaxed* (TR) methods either incorporate auxiliary non-wavelet components (WaveGC [28]) or relax theoretical constraints (LR-GWN (without admissibility), GWNN [47], DEFT [5], ASWT-SGNN [30]).

### 5.1 Long-Range Tasks

**Datasets.** We evaluate on PEPTIDES-FUNC (multi-label classification of 10 functional classes) and PEPTIDES-STRUCT (node-level regression for 11 3D structural properties) from the Long-Range Graph Benchmark [15, 45]. These tasks specifically test long-range propagation in biological structures where distant interactions determine biological functions.

**Results.** LR-GWN achieves state-of-the-art performance on both datasets, as reported in Table 1a. Most notably, it significantly outperforms all existing theory-compliant methods (70.52 vs 60.23 on PEPTIDES-FUNC) while maintaining theoretical guarantees. When constraints are relaxed, it surpasses all theory-relaxed baselines. It is worth noting that even the theory-compliant version of LR-GWN, achieves comparable performance to the theory-relaxed baselines, while maintaining pure wavelet interpretability and propagation.

**Training Time.** On PEPTIDES-FUNC, training LR-GWN takes approximately 20 seconds per epoch, totaling under 2.5 hours for full training (400 epochs). The partial EVD preprocessing costs

Table 1: **Benchmark results by theoretical adherence.** Theory-compliant methods (TC) satisfy wavelet admissibility and use pure wavelet propagation. Theory-relaxed methods (TR) either use additional non-wavelet modules, or relax theory constraints. Bold/underline represent first and second best, respectively.

(a) **Long-range benchmarks.**

| Method | PEPTIDES-FUNC (↑) | | PEPTIDES-STRUCT (↓) | |
|---|---|---|---|---|
| | TC | TR | TC | TR |
| SGWT [22] | <u>60.23 ± 0.27</u> | - | <u>25.39 ± 0.21</u> | - |
| GWNN [47] | - | 65.47 ± 0.48 | - | 27.34 ± 0.04 |
| DEFT [5] | - | 66.95 ± 0.63 | - | 25.06 ± 0.13 |
| WaveGC [28] | - | <u>69.73 ± 0.43</u> | - | <u>24.95 ± 0.07</u> |
| **LR-GWN (ours)** | **70.52 ± 0.29** | **72.16 ± 0.41** | **24.63 ± 0.07** | **24.62 ± 0.06** |

(b) **Short-range benchmarks.**

| Method | PHOTO (↑) | | COMPUTERS (↑) | |
|---|---|---|---|---|
| | TC | TR | TC | TR |
| SGWT [22] | <u>92.45 ± 0.15</u> | - | <u>85.19 ± 0.22</u> | - |
| GWNN [47] | - | 94.45 ± 0.18 | - | 90.75 ± 0.16 |
| ASWT-SGNN [29] | - | 93.80 ± 0.12 | - | 89.40 ± 0.19 |
| WaveGC [28] | - | <u>95.37 ± 0.44</u> | - | **92.26 ± 0.18** |
| **LR-GWN (ours)** | **95.22 ± 0.20** | **95.69 ± 0.23** | **91.15 ± 0.08** | <u>91.03 ± 0.20</u> |

roughly 1 minute, accounting for approximately 0.7% of total training time, highlighting that spectral augmentation via partial EVD remains computationally practical.

### 5.2 Short-Range Tasks

**Datasets.** We evaluate on PHOTO and COMPUTERS from Amazon co-purchase graphs [36, 42]. These datasets focus on local neighborhood interactions, validating that our long-range optimization does not compromise local modeling capabilities.

**Results.** As reported in Table 1b, LR-GWN achieves state-of-the-art performance on PHOTO and competitive results on COMPUTERS, demonstrating our proposed hybrid parametrization offers the flexibility to model both global and local propagation. Contrary to long-range tasks, relaxing admissibility constraints does not consistently improve performance on local interaction datasets.

**Training Time.** On PHOTO, training LR-GWN takes approximately 3s per epoch with total training time around 20 minutes for the full run (400 epochs). The partial EVD preprocessing with $k = 50$ costs roughly 1.3s using sparse solvers, accounting for approximately 0.1% of total training time.

### 5.3 Ablation Study

We assess the contribution of each component in LR-GWN for long-range tasks by performing an ablation study on PEPTIDES-FUNC.

**Component Analysis.** The spectral component proves essential for long-range modeling (3.34% performance drop when removed), while the polynomial component contributes primarily to computational efficiency with minimal performance impact (0.24% drop). This validates our design rationale: spectral filtering enables long-range propagation, while polynomial filtering provides efficient local aggregation.

Table 2: **Component ablation study on PEPTIDES-FUNC.** We report absolute and relative drops in average precision ($\Delta_{\text{AP}}$) when removing individual components from LR-GWN.

| **Ablation** | $\Delta_{\text{AP}}$ (abs) | $\Delta_{\text{AP}}$ (%) |
|---|---|---|
| No Spatial Component | $-0.0017$ | $-0.24$ |
| No Spectral Component | $-0.0239$ | $-3.34$ |

We report additional results in Appendix E, and discuss related work in Appendix C.

## 6 Discussion and Conclusion

We revisited the limitations of wavelet GNNs constrained by polynomial approximations and introduced Long-Range Graph Wavelet Networks (LR-GWN), a hybrid design that couples polynomial aggregation with spectral corrections on truncated eigenspaces. This construction unifies local and global information flow within wavelet operators, while supporting both strict admissibility for theoretical guarantees and relaxed variants for empirical gains.

Empirically, LR-GWN achieves state-of-the-art performance on long-range benchmarks and remains competitive on short-range tasks. More importantly, LR-GWN establishes wavelet-based filters as interpretable and theoretically grounded tools for graph learning, laying the foundation for adaptive and more expressive architectures. LR-GWN currently assumes static graphs and fixed truncation levels. Extending it to dynamic graphs or adapting $k$ during training remains an open avenue.

With LR-GWN, we aim to advance both the theoretical foundations and practical applications of wavelet-based long-range graph representation learning.

## Acknowledgments

The authors thank Marcel Kollovieh for his input on the complexity analysis; Marten Lienen, Lukas Gosch, Yan Scholten, Tim Beyer, Alessandro Palma, and Armando Bellante for their careful feedback and proofreading; F.G. thanks Denise Cocchiarella for their helpful support and insightful discussions. This research was supported by the German Federal Ministry of Education and Research (BMBF) through grant number 031L0289C. The authors of this work take full responsibility for its content.

## References

[1] Uri Alon and Eran Yahav. On the Bottleneck of Graph Neural Networks and its Practical Implications. In *International Conference on Learning Representations*, 2021.

[2] Alberto Ambrosetti, Anthony M. Reilly, Robert A. DiStasio, Jr., and Alexandre Tkatchenko. Long-range correlation energy calculated from coupled atomic response functions. *The Journal of Chemical Physics*, 140(18):18A508, February 2014.

[3] Adrián Arnaiz-Rodríguez, Ahmed Begga, Francisco Escolano, and Nuria M. Oliver. DiffWire: Inductive Graph Rewiring via the Lovász Bound. In *Learning on Graphs Conference*, November 2022.

[4] Muhammet Balcilar, Guillaume Renton, Pierre Héroux, Benoit Gaüzère, Sébastien Adam, and Paul Honeine. Analyzing the Expressive Power of Graph Neural Networks in a Spectral Perspective. In *International Conference on Learning Representations*, October 2020.

[5] Anson Bastos, Abhishek Nadgeri, Kuldeep Singh, Toyotaro Suzumura, and Manish Singh. Learnable Spectral Wavelets on Dynamic Graphs to Capture Global Interactions. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence*, volume 37 of *AAAI'23/IAAI'23/EAAI'23*, pp. 6779–6787. AAAI Press, February 2023.

[6] Deyu Bo, Chuan Shi, Lele Wang, and Renjie Liao. Specformer: Spectral Graph Neural Networks Meet Transformers. In *The Eleventh International Conference on Learning Representations*, February 2023.

[7] Ronald N. Bracewell. *The Fourier transform and its applications*. McGraw-Hill series in electrical and computer engineering Circuits and systems. McGraw-Hill, Boston, 3. ed edition, 2000.

[8] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric Deep Learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, July 2017. Conference Name: IEEE Signal Processing Magazine.

[9] Hyuna Cho, Jaeyoon Sim, Guorong Wu, and Won Hwa Kim. Neurodegenerative Brain Network Classification via Adaptive Diffusion with Temporal Regularization. In *Forty-first International Conference on Machine Learning*, June 2024.

[10] Fan R. K. Chung. *Spectral Graph Theory*. American Mathematical Soc., 1997.

[11] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in Neural Information Processing Systems*, volume 29, pp. 3844–3852. Curran Associates, Inc., 2016.

[12] Nikolay V. Dokholyan. Controlling Allosteric Networks in Proteins. *Chemical Reviews*, 116 (11):6463–6487, June 2016. Publisher: American Chemical Society.

[13] Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph Neural Networks with Learnable Structural and Positional Representations. In *International Conference on Learning Representations*, October 2021.

[14] Vijay Prakash Dwivedi, Ladislav Rampášek, Michael Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long Range Graph Benchmark. *Advances in Neural Information Processing Systems*, 35:22326–22340, December 2022.

[15] Vijay Prakash Dwivedi, Chaitanya K. Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking Graph Neural Networks. *Journal of Machine Learning Research*, 24(43):1–48, 2023.

[16] Johannes Gasteiger, Stefan Weiß enberger, and Stephan Günnemann. Diffusion Improves Graph Learning. In *Advances in Neural Information Processing Systems*, volume 32, pp. 13354–13366. Curran Associates, Inc., 2019.

[17] Simon Geisler, Yujia Li, Daniel J. Mankowitz, Ali Taylan Cemgil, Stephan Günnemann, and Cosmin Paduraru. Transformers Meet Directed Graphs. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 11144–11172. PMLR, July 2023. ISSN: 2640-3498.

[18] Simon Geisler, Arthur Kosmala, Daniel Herbst, and Stephan Günnemann. Spatio-Spectral Graph Neural Networks, June 2024. arXiv:2405.19121 [cs].

[19] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural Message Passing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1263–1272. PMLR, July 2017.

[20] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, March 2010. ISSN: 1938-7228.

[21] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A New Model for Learning in Graph Domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks*, volume 2, pp. 729–734, July 2005.

[22] David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, March 2011.

[23] Mingguo He, Zhewei Wei, Zengfeng Huang, and Hongteng Xu. BernNet: Learning Arbitrary Graph Spectral Filters via Bernstein Approximation. In *Advances in Neural Information Processing Systems*, November 2021.

[24] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*, November 2016.

[25] J. Knörzer, C. J. van Diepen, T.-K. Hsiao, G. Giedke, U. Mukhopadhyay, C. Reichl, W. Wegscheider, J. I. Cirac, and L. M. K. Vandersypen. Long-range electron-electron interactions in quantum dot systems and applications in quantum chemistry. *Physical Review Research*, 4(3):033043, July 2022. Publisher: American Physical Society.

[26] Devin Kreuzer, Dominique Beaini, William L. Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking Graph Transformers with Spectral Attention. In *Advances in Neural Information Processing Systems*, volume 34. Curran Associates, Inc., November 2021.

[27] Renjie Liao, Zhizhen Zhao, Raquel Urtasun, and Richard Zemel. LanczosNet: Multi-Scale Deep Graph Convolutional Networks. In *International Conference on Learning Representations*, September 2018.

[28] Nian Liu, Xiaoxin He, Thomas Laurent, Francesco Di Giovanni, Michael M. Bronstein, and Xavier Bresson. Advancing Graph Convolutional Networks via General Spectral Wavelets, May 2024. arXiv:2405.13806 [cs].

[29] Nian Liu, Xiaoxin He, Thomas Laurent, Francesco Di Giovanni, Michael M. Bronstein, and Xavier Bresson. A General Graph Spectral Wavelet Convolution via Chebyshev Order Decomposition, May 2025. arXiv:2405.13806 [cs].

[30] Ruyue Liu, Rong Yin, Yong Liu, and Weiping Wang. ASWT-SGNN: Adaptive Spectral Wavelet Transform-Based Self-Supervised Graph Neural Network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(12):13990–13998, March 2024. Number: 12.

[31] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. In *International Conference on Learning Representations*, February 2017.

[32] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, September 2018.

[33] Stephane G. Mallat. Multiresolution Approximations and Wavelet Orthonormal Bases of L 2 (R). *Transactions of the American Mathematical Society*, 315(1):69, September 1989.

[34] Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, Boston, 3rd edition, 2009.

[35] Andrei Andreyevich Markov. On a question by D.I. Mendeleev. *Zapiski Imperatorskoi Akademii Nauk*, 62:1–24, October 1889.

[36] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-Based Recommendations on Styles and Substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pp. 43–52, New York, NY, USA, August 2015. Association for Computing Machinery.

[37] Erxue Min, Runfa Chen, Yatao Bian, Tingyang Xu, Kangfei Zhao, Wenbing Huang, Peilin Zhao, Junzhou Huang, Sophia Ananiadou, and Yu Rong. Transformer for Graphs: An Overview from Architecture Perspective. *arXiv preprint arXiv:2202.08455*, February 2022. arXiv:2202.08455 [cs].

[38] Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a General, Powerful, Scalable Graph Transformer. *arXiv preprint arXiv:2205.12454*, January 2023. arXiv:2205.12454 [cs].

[39] Aliaksei Sandryhaila and José M. F. Moura. Discrete Signal Processing on Graphs. *IEEE Transactions on Signal Processing*, 61(7):1644–1656, April 2013. Conference Name: IEEE Transactions on Signal Processing.

[40] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1):61–80, January 2009. Conference Name: IEEE Transactions on Neural Networks.

[41] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[42] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of Graph Neural Network Evaluation. In *Relational Representation Learning Workshop, Advances in Neural Information Processing Systems*, volume 32, November 2018.

[43] David I Shuman, Pierre Vandergheynst, and Pascal Frossard. Chebyshev polynomial approximation for distributed signal processing. In *2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, pp. 1–8, June 2011.

[44] Kimberly Stachenfeld, Jonathan Godwin, and Peter Battaglia. Graph Networks with Spectral Message Passing, December 2020. arXiv:2101.00079 [stat].

[45] Jan Tönshoff, Martin Ritzert, Eran Rosenbluth, and Martin Grohe. Where Did the Gap Go? Reassessing the Long-Range Graph Benchmark. In *The Second Learning on Graphs Conference*, November 2023.

[46] Xiyuan Wang and Muhan Zhang. How Powerful are Spectral Graph Neural Networks. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 23341–23362. PMLR, June 2022.

[47] Bingbing Xu, Huawei Shen, Qi Cao, Yunqi Qiu, and Xueqi Cheng. Graph Wavelet Neural Network. In *International Conference on Learning Representations*, 2019.

[48] Jingxuan Zhu, Juexin Wang, Weiwei Han, and Dong Xu. Neural relational inference to learn long-range allosteric interactions in proteins from molecular dynamics simulations. *Nature Communications*, 13(1):1661, March 2022. Publisher: Nature Publishing Group.

# A Extended Background

This appendix provides a more detailed overview of the mathematical foundations underlying our method. We first recall core concepts from spectral graph theory, then review wavelet analysis in both Euclidean and graph domains, and finally introduce polynomial approximations that enable scalable implementations.

## A.1 Graphs and Laplacians

A graph $\mathcal{G} = (\boldsymbol{A}, \boldsymbol{X})$ consists of $n$ nodes, $m$ edges, an adjacency matrix $\boldsymbol{A} \in \{0,1\}^{n\times n}$, and node features $\boldsymbol{X} \in \mathbb{R}^{n\times d}$. The degree matrix is defined as $\boldsymbol{D} := \mathrm{diag}(\boldsymbol{A}\mathbf{1}) \in \mathbb{R}^{n\times n}$, where $\mathbf{1} \in \mathbb{R}^n$ denotes the all-ones vector. From this, the combinatorial Laplacian is $\boldsymbol{L} := \boldsymbol{D} - \boldsymbol{A}$, while the symmetrically normalized Laplacian is $\mathcal{L} := \boldsymbol{I} - \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$. In this work we primarily use $\mathcal{L}$, though other variants (e.g., random-walk Laplacian) are also common.

For undirected graphs, $\mathcal{L}$ is symmetric and positive semidefinite, which guarantees an eigenvalue decomposition $\mathcal{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$. Here $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ contains the eigenvectors as columns, forming an orthogonal basis for functions defined on the graph ($\boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}$), and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ collects the eigenvalues. The spectrum is bounded as $0 = \lambda_1 \leq \cdots \leq \lambda_n \leq 2$. Small eigenvalues correspond to smooth, slowly varying components, while large eigenvalues capture oscillatory, high-frequency behavior [10].

## A.2 Graph Fourier Transform

The graph Fourier transform (GFT) [43, 39] generalizes the classical discrete Fourier transform (DFT) [7] to signals supported on graphs. In the classical Euclidean setting, a signal is decomposed into complex exponentials, whose frequencies correspond to sinusoidal oscillations of different scales. On a graph, there is no notion of translation or frequency in the usual sense. Instead, the eigenvectors of the Laplacian play the role of Fourier modes, with the associated eigenvalues serving as the notion of frequencies.

Given a graph signal $\boldsymbol{x} \in \mathbb{R}^n$ (often corresponding to a column of the feature matrix $\boldsymbol{X}$), the GFT is $\widehat{\boldsymbol{x}} = \boldsymbol{U}^\top \boldsymbol{x}$, which projects the signal onto the eigenbasis of $\mathcal{L}$. The inverse GFT reconstructs the signal as $\boldsymbol{x} = \boldsymbol{U}\widehat{\boldsymbol{x}}$. Filtering with a spectral kernel $g : \mathbb{R} \to \mathbb{R}$ amounts to rescaling each Fourier coefficient according to the eigenvalue, yielding

$$g *_{\mathcal{G}} \boldsymbol{x} = \boldsymbol{U}\, \widehat{g}(\boldsymbol{\Lambda})\, \boldsymbol{U}^\top \boldsymbol{x},$$

where $\widehat{g}(\boldsymbol{\Lambda}) = \mathrm{diag}(\widehat{g}(\lambda_1), \ldots, \widehat{g}(\lambda_n))$, and $\widehat{g} = \boldsymbol{U}^\top g$. In this sense, spectral filters are functions of the Laplacian spectrum, directly analogous to frequency filters in classical Fourier analysis.

## A.3 Wavelet Analysis

Wavelet theory extends Fourier analysis by enabling localized, multiresolution representations of signals [33, 34]. Whereas Fourier bases are global and capture only frequency information, wavelets provide both spectral and spatial localization by combining scaling and translation operations.

The continuous wavelet transform (CWT) of a function $f : \mathbb{R} \to \mathbb{R}$ at scale $s > 0$ and location $l \in \mathbb{R}$ is defined as

$$W_f(s, l) = \int_{\mathbb{R}} f(t)\, \psi^*_{s,l}(t)\, dt,$$

where the wavelet family is generated from a mother wavelet $\psi$ by scaling and translation, $\psi_{s,l}(t) = \frac{1}{\sqrt{s}}\psi\big(\frac{t-l}{s}\big)$, and $^*$ denotes complex conjugation. Exact reconstruction requires the mother wavelet to satisfy the admissibility condition, ensuring that no frequency component is lost.

**Proposition 1** (Wavelet Admissibility)**.** *A wavelet $\psi$ with Fourier transform $\widehat{\psi}$ is admissible if*

$$C_\psi = \int_0^\infty \frac{|\widehat{\psi}(\zeta)|^2}{\zeta}\, d\zeta < \infty,$$

*which holds when $\widehat{\psi}(0) = 0$ and $\lim_{\zeta\to\infty} \widehat{\psi}(\zeta) = 0$.*

This condition ensures that $\widehat{\psi}$ behaves as a band-pass filter, emphasizing intermediate frequency bands while attenuating very low and high frequencies. To retain the low-frequency components, wavelet systems are typically complemented by a scaling function $\phi$ with Fourier transform $\widehat{\phi}(0) > 0$, which acts as a low-pass filter. Together, $\phi$ and the wavelet family provide a multiresolution decomposition of signals.

### A.4 Wavelets on Graphs

Hammond et al. [22] extend wavelet analysis to graphs by defining filters in the Laplacian spectral domain. Let $\mathcal{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$ be the eigendecomposition of the (normalized) Laplacian, and let $\widehat{\psi}$ and $\widehat{\phi}$ denote a band-pass wavelet kernel and a low-pass scaling kernel, respectively (both applied elementwise to eigenvalues).

The *wavelet operator* at scale $s > 0$ and the *scaling operator* are

$$\Psi_s := \boldsymbol{U}\,\widehat{\psi}(s\boldsymbol{\Lambda})\,\boldsymbol{U}^\top, \qquad \Phi := \boldsymbol{U}\,\widehat{\phi}(\boldsymbol{\Lambda})\,\boldsymbol{U}^\top.$$

Given a graph signal $\boldsymbol{x} \in \mathbb{R}^n$, the wavelet coefficients at scale $s$ are $W_{\boldsymbol{x}}(s) = \Psi_s \boldsymbol{x}$, while the scaling coefficients are $H_{\boldsymbol{x}} = \Phi \boldsymbol{x}$. This parallels the classical construction: wavelets capture localized variations, whereas the scaling function captures smooth, global structure. A multiscale analysis is obtained by a family of dilations $\{s_j\}$, i.e., $\{\widehat{\psi}(s_j\cdot)\}_j$. On graphs, admissibility is typically enforced by $\widehat{\psi}(0) = 0$, with low frequencies covered by $\widehat{\phi}$ satisfying $\widehat{\phi}(0) > 0$.

### A.5 Chebyshev Polynomial Approximations

Although spectral constructions are mathematically elegant, direct computation is expensive due to the eigenvalue decomposition, which scales cubically in the number of nodes. To make spectral filters scalable, they are commonly approximated with low-order polynomials in the Laplacian, which can be applied directly in the vertex domain.

A widely used choice is the Chebyshev polynomial basis [22, 11]. For a filter $\widehat{g}$, one approximates

$$\widehat{g}_{\boldsymbol{\omega}}(\boldsymbol{\Lambda}) = \sum_{i=0}^{\rho} \omega_i T_i(\tilde{\boldsymbol{\Lambda}}),$$

where $\boldsymbol{\omega} = [\omega_0, \ldots, \omega_\rho]$ are learnable coefficients, $\rho$ is the polynomial order, and $\tilde{\boldsymbol{\Lambda}} = \frac{2}{\lambda_{\max}}\boldsymbol{\Lambda} - \boldsymbol{I}$ rescales the spectrum to $[-1, 1]$, the domain of the Chebyshev basis.

**Definition 1** (Chebyshev Polynomials of the First Kind)**.** *The Chebyshev polynomials $T_i$ are defined recursively by*

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_i(x) = 2xT_{i-1}(x) - T_{i-2}(x) \quad (i \geq 2).$$

Chebyshev expansions are particularly effective because they minimize the maximum approximation error on $[-1, 1]$ (in the minimax sense), and can be evaluated efficiently via recurrence. In practice, this enables fast, localized, and scalable approximations of spectral filters, including graph wavelets, without explicit eigendecomposition.

## B Extended Discussion on Polynomial Filter Limitations

Polynomial filters have become the standard choice for approximating spectral graph operators because they can be applied without eigendecomposition. In this section, we expand on the three perspectives introduced in Section 3, providing formal statements and illustrative examples.

### B.1 Propagation and Hop-Limits

The interpretation of polynomial filters as message-passing operators is immediate from the fact that $\mathcal{L}^k$ encodes walks of length $k$ [4]. A degree-$\rho$ polynomial filter can therefore only aggregate information from nodes within $\rho$ hops of each target node. This locality is beneficial for efficiency, but also imposes a hard ceiling on the effective receptive field. Deep message-passing GNNs, which correspond to higher-degree polynomials, in principle extend this range but suffer from *over-squashing* [1]: as the neighborhood grows exponentially with hop count, long-range signals are

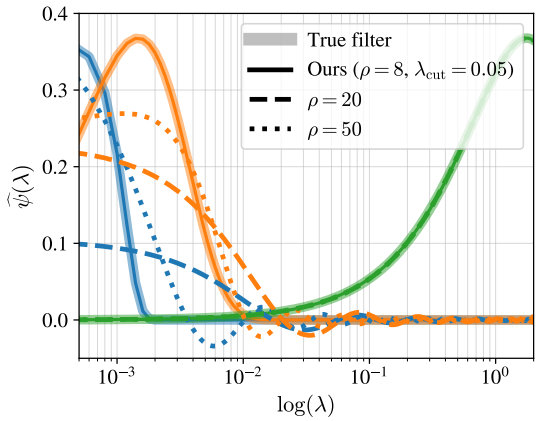


Figure 2: Different polynomial approx. vary in their ability to model low-frequency wavelets. Both low-order ($\rho = 20$) and high-order ($\rho = 50$) variants struggle with sharp transitions. Our method ($\rho = 8, \lambda_{\text{cut}} = 0.05$) captures both smooth and sharp irregularities.

compressed into fixed-dimensional node embeddings, leading to severe information loss. Thus, regardless of how coefficients are chosen, polynomial filters cannot escape their fundamentally local nature.

### B.2 Function Approximation and Smoothness

From a spectral perspective, the limitations of polynomial filters manifest in their inability to approximate functions with steep slopes. This is formally captured by Markov's inequality:

**Theorem 1** (Markov's Inequality [35])**.** *Let $P$ be a polynomial of degree at most $\rho$ with $|P(x)| \leq 1$ for all $x \in [-1, 1]$. Then*

$$\max_{x \in [-1,1]} \left| \frac{dP(x)}{dx} \right| \leq \rho^2.$$

The theorem shows that the slope of any polynomial is globally bounded, preventing sharp transitions. Consequently, low-order polynomials can approximate smooth filters, but require very high degree to approximate narrow band-pass filters or scaling functions with steep low-frequency cutoffs. In Fig. 2, we visualize this phenomenon: wavelets at medium scales are well approximated even with low-order expansions, while scaling functions and fine-scale wavelets are captured poorly unless the degree is significantly increased.

### B.3 Convergence to Discontinuous Filters

A more severe limitation emerges when considering discontinuous spectral filters, such as ideal low-pass or band-pass functions. While continuous functions can, in principle, be approximated uniformly by polynomials (by the Weierstrass theorem), discontinuous functions break this guarantee. Recent results make this precise in the context of graph filters:

**Theorem 2** (Slow Polynomial Convergence [18])**.** *Let $\widehat{g}$ be a discontinuous spectral filter. For any polynomial sequence $(g_{\gamma_\rho})_\rho$, there exists a graph sequence $(\mathcal{G}_\rho)_\rho$ such that*

$$\nexists \alpha > 0 : \sup_{0 \neq \boldsymbol{X} \in \mathbb{R}^{|\mathcal{G}_\rho| \times d}} \frac{\|(g_{\gamma_\rho} - g) *_{\mathcal{G}_\rho} \boldsymbol{X}\|_F}{\|\boldsymbol{X}\|_F} = \mathcal{O}(\rho^{-\alpha}).$$

This result formalizes the intuition that discontinuous filters cannot be efficiently represented by polynomials: convergence is arbitrarily slow in operator norm, regardless of polynomial degree. This barrier persists across graphs of increasing size, implying that polynomial parametrizations are structurally mismatched with frequency-selective or discontinuous behaviors.

### B.4 Takeaways

Together, these results reveal the inherent trade-off in polynomial filters: they offer computational simplicity but at the cost of locality, smoothness restrictions, and poor convergence on discontinuous operators. These limitations highlight the need for alternative parametrizations—such as wavelet-based designs—that can preserve efficiency while enabling non-local, multi-scale, and spectrally expressive behavior.

## C Related Work

**Wavelets on Graphs.** The Spectral Graph Wavelet Transform (SGWT) by Hammond et al. [22] introduced cubic spline wavelets and an exponential scaling function at fixed scales, ensuring admissibility but lacking learnable parameters. SGWT approximates filters using finite-order Chebyshev polynomials. Graph Wavelet Neural Networks (GWNN) [47] extend SGWT with fixed exponential wavelets, i.e., $g(s\lambda_i) = e^{s\lambda_i}$. While building on wavelet theory, this construction does not enforce the admissibility condition, since $g(0) = 1$. DEFT [5] follows SGWT and parametrizes wavelet filters using free-parameters Chebyshev polynomials via a combination of GNNs and MLPs. The coefficients, however, are not bound and, by construction, do not enforce the admissibility condition. ASWT-SGNN [30] approximates wavelet filter coefficients using Jackson–Chebyshev polynomials. Similarly to the discussion above about GWNN and DEFT, the coefficients are not bound to satisfy $g(0) = 0$. AGT [9] extends GWNN, and learns node-wise scales, either exactly or via approximation. As with the methods discussed so far, this approach does not enforce the admissibility condition, and thus does not guarantee that the resulting filters behave as band-pass wavelets. WaveGC [29] satisfies the admissibility condition by parametrizing the scaling function with odd-terms of a Chebyshev polynomial, and the wavelet function with even-terms. However, the separation of basis terms limits functional expressiveness, and the use of non-wavelet-interpretable message passing diminishes the benefit of using wavelets.

**Combining Spatial and Spectral Filters for Long-Range Interactions.** Hybrid GNNs combine local spatial and global spectral information to enhance long-range modeling. Stachenfeld et al. [44] propose a hybrid message-passing framework that sacrifices permutation equivariance, while Liao et al. [27] incorporate spectral information via the Lanczos algorithm. More recently, Geisler et al. [18] introduced a spatial-spectral parametrization to improve efficiency. Beyond the explicit combination of spatial and spectral filters, graph rewiring [16, 3] enhance connectivity and mitigate over-squashing by modifying connectivity. Graph transformers (GTs) [37, 13, 17] use global attention mechanisms to enable interactions between distant nodes. GTs often rely on spectral positional encodings paired with message passing [38, 17]. Rampášek et al. [38] introduce a framework for general, powerful and scalable graph transformers (GPS), which allow for the definition of a broad family of architectures that combine local message passing with global attention mechanisms. Similarly to these models, LR-GWN integrates local and global components. In our case, local aggregation is achieved through polynomial filters, while global propagation is handled via a spectral wavelet filter. However, the core methodology and motivation differ. LR-GWN is rooted in wavelet theory, with the aim of modeling low-pass and band-pass filters in a principled and interpretable manner. Graph transformers, by contrast, cover a broad family of architectures with varying attention mechanisms and are generally not designed with the same frequency-selective properties in mind.

Our mention of graph transformers is intended purely for conceptual contrast; they were not included in experimental comparisons, as our focus is restricted to wavelet-based operators.

## D Implementation Details

### D.1 Filter Parametrization

**Spatial Filter.** We model the spatial filter $P_{\boldsymbol{\omega}^{(l)}} : [0, \lambda_{\text{max}}]^k \to \mathbb{R}^k$ using a finite-order Chebyshev polynomial expansion with learnable coefficients $\boldsymbol{\omega}^{(l)} = [\omega_0^{(l)}, \ldots, \omega_\rho^{(l)}] \in \mathbb{R}^{\rho+1}$:

$$P_{\boldsymbol{\omega}^{(l)}}(\mathcal{L}) = \sum_{i=0}^{\rho} \omega_i^{(l)} T_i(\mathcal{L}), \tag{6}$$

where $T_i(\cdot)$ is the $i$-th Chebyshev polynomial. This formulation enables spatially localized filtering with computational efficiency. Alternative polynomial bases, such as Bernstein [23] or Jacobi polynomials [46], could also be employed.

**Spectral Filter.** The spectral component $S_{\boldsymbol{\theta}^{(l)}} : [0, \lambda_{\text{max}}]^k \to \mathbb{R}^k$ operates directly on the eigenvalues of $\mathcal{L}$, offering fine-grained frequency control. It is defined as

$$S_{\boldsymbol{\theta}^{(l)}}(\lambda) = \text{GaussianSmearing}(\lambda)\boldsymbol{W}_{\theta^{(l)}}, \tag{7}$$

where $\boldsymbol{W}_{\theta^{(l)}} \in \mathbb{R}^{z \times d^{(l)}}$ are learnable weights, and $d^{(l)}$ is the output dimensionality of layer $l$. The basis $\text{GaussianSmearing}(\lambda) : [0, 2]^k \to \mathbb{R}^{k \times z}$ projects eigenvalues onto $z$ Gaussian radial functions, introducing a frequency cutoff $\lambda_{\text{cut}} < \lambda_{\text{max}}$ [41]. We follow Geisler et al. [18] and apply a frequency-domain windowing function to regularize the spectral response. Element-wise, the filter is defined as $S_{\boldsymbol{\theta}^{(l)}}(\boldsymbol{\Lambda})_{u,v} := S_v(\lambda_u; \boldsymbol{\theta}^{(l)})$. Alternative spectral parametrizations, such as SpecFormer [6], could be used depending on application-specific needs.

#### D.1.1 Shared and Independent Filter Configurations

We define two configurations for our wavelet-based graph neural network: the *shared filter* and the *independent filter* approaches. In the shared filter configuration, we parametrize a single mother wavelet and generate a family of self-similar wavelet functions by applying predefined or learnable scales. Conversely, in the independent filter approach, each wavelet function is parametrized independently, allowing greater flexibility and expressivity.

**Shared Filter.** In this configuration, the $j$-th wavelet filter is constructed as

$$\psi(\boldsymbol{U}, \boldsymbol{\Lambda}, \mathcal{L}; s_j) = \boldsymbol{U} S_{\theta_\psi}(s_j \boldsymbol{\Lambda}) \boldsymbol{U}^\top + P_{\omega_\psi}(s_j \mathcal{L}), \tag{8}$$

where a single function $\psi$ is parametrized following Eq. (3) and then scaled using predefined or learnable scales (Appendix D.1.2). This method is computationally efficient, as it requires only a single set of parameters $\theta_\psi$ and $\omega_\psi$, reducing the overall complexity of the model.

**Independent Filters.** To increase model expressivity, we introduce an alternative approach in which each wavelet function is parametrized independently instead of being derived from a single mother wavelet. This configuration allows each wavelet to adjust its spectral and polynomial components separately, adapting to different levels of smoothness or sharpness. Larger-scale wavelets tend to be smoother and benefit more from polynomial components, while smaller-scale wavelets are sharper and require a stronger contribution from the spectral component. In the independent filters configuration, the $j$-th wavelet filter is given by

$$\psi_j(\boldsymbol{U}, \boldsymbol{\Lambda}, \mathcal{L}) = \boldsymbol{U} S_{\theta_{\psi_j}}(\boldsymbol{\Lambda}) \boldsymbol{U}^\top + P_{\omega_{\psi_j}}(\mathcal{L}), \tag{9}$$

where each $\psi_j$ has its own set of learnable spectral parameters $\theta_{\psi_j}$ and polynomial parameters $\omega_{\psi_j}$.

#### D.1.2 Scale Parametrization.

Wavelet-based graph filters rely on a set of scales $\{s_j\}_{j=1}^J$ to capture multi-resolution information. Instead of using fixed scales as in Hammond et al. [22], we introduce a learnable interpolation scheme in the shared filter configuration (Appendix D.1.1). The scale bounds are defined as $\log s_{\min} = \log(t_{\text{L}}/\lambda_{\text{max}})$ and $\log s_{\max} = \log(t_{\text{U}}\lambda_{\text{LP}}/\lambda_{\text{max}})$, where $t_{\text{L}}$ and $t_{\text{U}}$ are learnable lower and upper limits, and $\lambda_{\text{LP}}$ emphasizes low-frequency components. Assuming the maximum eigenvalue of the graph to be $\lambda_{\max} = 2$ as it is usually the case in real-world graphs, we have $\log s_{\min} = \log t_{\text{L}}, \log s_{\max} = \log t_{\text{U}}\lambda_{\text{LP}}$. The intermediate scales are uniformly spaced in log-space,

$$s_i = \exp\left(\log s_{\max} + \frac{i(\log s_{\min} - \log s_{\max})}{N-1}\right). \tag{10}$$

To ensure positive scales, we apply the $\text{softplus}$ function.

### D.2 Modeling Practices

#### D.2.1 Initialization

The wavelet filter parameters are initialized using standard Xavier initialization [20] for neural network weights and by setting $\omega_0 = 1$ and $\omega_i = 0$ for all $i = 1, \ldots, \rho$ in the Chebyshev polynomials.

While the spectral and spatial components could theoretically be designed to match a predefined wavelet, such as the Mexican hat wavelet, we observed no significant improvements from this approach. Consequently, we adopt conventional parameter initialization without enforcing wavelet-based constraints in our experiments.

#### D.2.2 Admissibility Condition

To ensure the theoretical validity of our graph wavelet filter, we must guarantee the invertibility of the wavelet transform. This requires satisfying the admissibility condition at zero frequency, as stated in Proposition 1. Since the graph spectrum is bounded (i.e., $\lambda \in [0, \lambda_{\max}]$ with $\lambda_{\max} \leq 2$), we only need to enforce the condition $\widehat{\psi}(0) = 0$, while disregarding $\lim_{\lambda \to \infty} \widehat{\psi}(\lambda) = 0$.

Enforcing the admissibility condition is not always a necessity. Rather, it is a theoretical property that ensures the wavelet behaves as a band-pass filter. In practice, many applications do not strictly require this condition, and relaxing it can lead to slight empirical gains in some cases. However, with LR-GWN, we aim to provide a theoretically grounded framework that can allow to enforce the admissibility condition if needed, while also permitting for flexibility in applications where this condition is not strictly required.

**Primary Approach.** In practice, we can enforce this constraint by subtracting the zero-frequency response from the spectral and polynomial filters:

$$\widetilde{P}_\omega(\lambda) = P_\omega(\lambda) - P_\omega(0), \quad \widetilde{S}_\theta(\lambda) = S_\theta(\lambda) - S_\theta(0). \tag{11}$$

This transformation ensures $\widehat{\psi}(0) = \widetilde{P}_\omega(0) + \widetilde{S}_\theta(0) = 0$ by construction. Thus, the wavelet filter in the spectral domain becomes

$$\widehat{\psi}(\mathbf{\Lambda}) = \widetilde{P}_\omega(\mathbf{\Lambda}), + \widetilde{S}_\theta(\mathbf{\Lambda}) \tag{12}$$

and the filtering operation in the vertex domain is given by

$$\psi\left(\boldsymbol{U}, \mathbf{\Lambda}, \mathcal{L}\right) = \widetilde{P}_\omega(\mathcal{L}), + \boldsymbol{U}\widetilde{S}_\theta(\mathbf{\Lambda})\boldsymbol{U}^\top \tag{13}$$

where both $\widetilde{P}_\omega(\mathcal{L})$ and $\widetilde{S}_\theta(\mathbf{\Lambda})$ are zero-frequency corrected versions of their respective filters.

**Alternative Approach.** For the wavelet filter $\psi(\mathbf{\Lambda}) = S_\theta(\mathbf{\Lambda}) + P_\omega(\mathbf{\Lambda})$, where $P_\omega(\mathbf{\Lambda})$ is expressed as a Chebyshev polynomial, the admissibility condition at zero frequency is satisfied if

$$\omega_0 = \sum_{i=1}^{\rho} (-1)^{i+1} \omega_i - S_\theta(0).$$

We show the steps that lead to the above formulation. The wavelet filter $\psi(\lambda)$ consists of two components:

$$\psi(\lambda) = S_\theta(\lambda) + P_\omega(\lambda),$$

where $S_\theta(\lambda)$ and $P_\omega(\lambda)$ are the spectral and polynomial filter responses, respectively, at frequency $\lambda$. The admissibility condition requires that $\psi(0) = 0$, i.e.,

$$S_\theta(0) + P_\omega(0) = 0.$$

For the admissibility condition to hold, we either enforce both components to be zero ($S_\theta(0) = P_\omega(0) = 0$) or allow $P_\omega(0) = -S_\theta(0)$. We express $P_\omega(\lambda)$ as a Chebyshev polynomial. After rescaling the input from $\lambda \in [0, 2]$ to $\widetilde{\lambda} \in [-1, 1]$, the condition $P_\omega(0) = \widetilde{P}_\omega(-1)$ leads to the equation

$$\omega_0 = \sum_{i=1}^{\rho} (-1)^{i+1} \omega_i - S_\theta(0),$$

which satisfies the admissibility condition at zero frequency.

However, this approach is more challenging to implement, as it requires manually updating the learnable parameters. While this is feasible, we found that the primary approach worked better overall. It is more flexible and streamlined, making it easier to integrate into our model while still satisfying the necessary conditions.

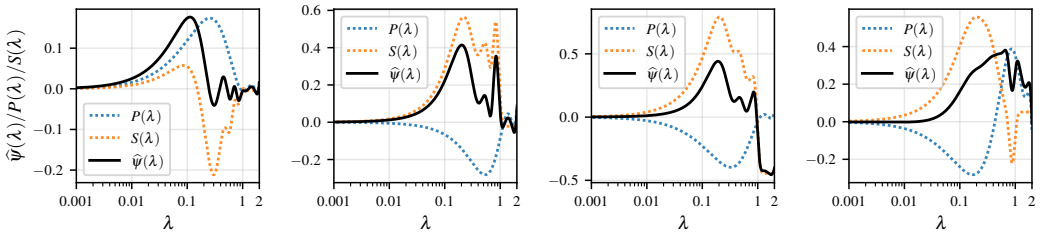


Figure 3: Examples of filters learned by our model during training. The polynomial component $P(\lambda)$ captures the smooth part of the filter, while the spectral component $S(\lambda)$ introduces flexibility, enabling sharper variations. Our filters inherently satisfy the admissibility condition $\widehat{\psi}(0) = 0$.

### D.2.3 Wavelet Residuals

We introduce residual connections within the wavelet filter (*wavelet residuals*) to improve model generalization. We write the forward step of our model with included wavelet residual connections as

$$\hat{\boldsymbol{H}}_{\psi,j}^{(l-1)} = \sigma\left[\left(\psi^{(l)}\left(\boldsymbol{U},\boldsymbol{\Lambda},\mathcal{L}\right) + \boldsymbol{I}\right)\hat{\boldsymbol{H}}^{(l-1)}\right] = \sigma\left[\psi_{\boldsymbol{I}}{}^{(l)}\left(\boldsymbol{U},\boldsymbol{\Lambda},\mathcal{L}\right)\hat{\boldsymbol{H}}^{(l-1)}\right]. \tag{14}$$

In the graph frequency domain, this corresponds to the graph wavelet filter $\widehat{\psi}_{\boldsymbol{I}}(\boldsymbol{\Lambda}) = S_\theta(\boldsymbol{\Lambda}) + P_\omega(\boldsymbol{\Lambda}) + \boldsymbol{I}$. While including wavelet residual connections brings desirable properties in terms of gradient flow, preserving the theoretical properties of the wavelet transform remains important to theoretically guarantee the existence of the inverse wavelet transform. Similarly to how we enforced the admissibility condition in Appendix D.2.2, we can now let the spectral and polynomial filters transform as $S_\theta(\lambda) \to S_\theta(\lambda) - S_\theta(0) - 1/2$, and $P_\omega(\lambda) \to P_\omega(\lambda) - P_\omega(0) - 1/2$. In fact, this allows us to write $\widehat{\psi}_{\boldsymbol{I}}(\boldsymbol{\Lambda}) = S_\theta(\boldsymbol{\Lambda}) - S_\theta(\boldsymbol{0}) - \boldsymbol{I}/2 + P_\omega(\boldsymbol{\Lambda}) - P_\omega(\boldsymbol{0}) - \boldsymbol{I}/2 + \boldsymbol{I}$. The wavelet filter then becomes $\psi_{\boldsymbol{I}}\left(\boldsymbol{U},\boldsymbol{\Lambda},\mathcal{L}\right) = U\widetilde{S}_\theta(\boldsymbol{\Lambda})U^\top + \widetilde{P}_\omega(\mathcal{L}) + \boldsymbol{I}$, where $\widetilde{S}_\theta(\boldsymbol{\Lambda}) = S_\theta(\boldsymbol{\Lambda}) - S_\theta(\boldsymbol{0}) - \boldsymbol{I}/2$ and $\widetilde{P}_\omega(\mathcal{L}) = P_\omega(\mathcal{L}) - P_\omega(\boldsymbol{0}) - \boldsymbol{I}/2$, thus ensuring the condition $\widehat{\psi}(0) = 0$ by construction.

**Residual Connections.** While the inclusion of wavelet residuals theoretically ensures the preservation of the admissibility condition, in practice, we observe that normal residual connections, where the residual is added after the filtering operation, further improve the model's generalization. Therefore, we opt to use standard residual connections.

### D.3 Computational Complexity.

The LR-GWN layer integrates multiple structural components whose computational costs can be precisely characterized. Let $n$ and $m$ denote the number of nodes and edges in the input graph, and let $d$ represent the feature dimensionality, assumed constant across all layers of the network. We denote by $L$ the depth of the MLP $f_\vartheta^{(l)}$, by $\rho$ the polynomial order used in the Chebyshev approximation, by $k$ the number of retained eigenvectors for the spectral component, and by $J+1$ the total number of filters, including the scaling function.

Each layer begins with a node-wise transformation via an $L$-layer MLP of hidden size $d$, shared across both filtering components. This projection incurs a cost of $\mathcal{O}(Ld^2n)$. The polynomial component uses recursive Chebyshev polynomials over the sparse Laplacian. Recall that multiplication between an $n \times n$ sparse matrix with $m$ non-zero entries, and a dense $n \times d$ matrix has cost $\mathcal{O}(md)$. For a $\rho$-order polynomial, and across all filters, this results in a total cost of $\mathcal{O}((J+1)\rho md)$. The spectral component computes a projection via $\boldsymbol{U}$ and back, for each filter. This involves two matrix multiplications of cost $\mathcal{O}(kdn)$ and an element-wise scaling of cost $\mathcal{O}(kd)$, which is asymptotically negligible. Across filters, the spectral component contributes a total cost of $\mathcal{O}((J+1)kdn)$.

Crucially, the partial eigendecomposition required for the spectral component is computed once per graph at cost $\mathcal{O}(km)$ as a preprocessing step, and can be amortized across the network.

In summary, the per-layer complexity is $\mathcal{O}\left(Ld^2n + (J+1)\rho dm + (J+1)kdn\right)$. Since $J$, $L$, $d$, $\rho$, and $k$ are typically small constants in practice, the dominant terms grow with the number of nodes $n$, and edges $m$. The expression therefore simplifies to $\mathcal{O}(n+m)$. For sparse graphs we can assume constant node degree deg, hence $m = \text{deg} \cdot n = \Omega(n)$, yielding an asymptotical complexity of $\mathcal{O}(m)$.

This decomposition reveals the efficiency of LR-GWN: despite leveraging both spatial and spectral graph filters, the model remains scalable to large, sparse graphs.

# E Additional results

We report here additional results to those presented in Section 5.

## E.1 Ablation Study

**Admissibility Trade-off.** The performance difference between theory-compliant (70.52) and theory-relaxed (72.16) variants on PEPTIDES-FUNC demonstrates LR-GWN's flexibility in balancing theoretical guarantees with empirical performance, allowing adaptation to application-specific requirements.

**Considerations on $k$ and $\rho$.** We conducted preliminary ablation studies on both the number of retained eigenvalues $k$ and polynomial order $\rho$. While performance varies with both hyperparameters, no consistent pattern emerges across datasets. For smaller graphs like PEPTIDES-FUNC, we retain a large number of eigenvalues ($k = 150$) while keeping computational costs manageable, exploring the full expressive capacity. On larger graphs, smaller $k$ values typically suffice to balance efficiency and performance. Similarly, optimal $\rho$ is dataset-specific, but the model remains robust within reasonable ranges (typically $\rho = 3$-15).

# F Theoretical Results

## F.1 Polynomial filters in spectral and spatial domain

For a graph with Laplacian $\mathcal{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$ and a polynomial filter of degree $\rho$, the spectral filtering operation $\boldsymbol{U}\widehat{g}_\omega(\boldsymbol{\Lambda})\boldsymbol{U}^\top x$ is equivalent to the spatial operation $g_\omega(\mathcal{L})x$, where $\widehat{g}_\omega(\boldsymbol{\Lambda}) = \sum_{i=0}^{\rho} \omega_i \boldsymbol{\Lambda}^i$.

*Proof.* We prove this by showing that powers of the Laplacian in the spatial domain correspond to powers of eigenvalues in the spectral domain. Starting with the spectral filtering operation:

$$\boldsymbol{U} g_\omega(\boldsymbol{\Lambda})\boldsymbol{U}^\top x = \boldsymbol{U}\left(\sum_{i=0}^{\rho} \omega_i \boldsymbol{\Lambda}^i\right)\boldsymbol{U}^\top x \tag{15}$$

$$= \sum_{i=0}^{\rho} \omega_i \boldsymbol{U}\boldsymbol{\Lambda}^i\boldsymbol{U}^\top x \tag{16}$$

Now we use the key insight that for the eigendecomposition $\mathcal{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$, we have:

$$\boldsymbol{U}\boldsymbol{\Lambda}^i\boldsymbol{U}^\top = (\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top)^i = \mathcal{L}^i.$$

This holds by induction:

- Base case $i = 1$: $\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top = \mathcal{L}$ by definition
- Inductive step: Assume $\boldsymbol{U}\boldsymbol{\Lambda}^k\boldsymbol{U}^\top = \mathcal{L}^k$. Then:

$$\boldsymbol{U}\boldsymbol{\Lambda}^{k+1}\boldsymbol{U}^\top = \boldsymbol{U}\boldsymbol{\Lambda}^k\boldsymbol{\Lambda}\boldsymbol{U}^\top \tag{17}$$

$$= \boldsymbol{U}\boldsymbol{\Lambda}^k(\boldsymbol{U}^\top\boldsymbol{U})\boldsymbol{\Lambda}\boldsymbol{U}^\top \quad (\text{since } \boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}) \tag{18}$$

$$= (\boldsymbol{U}\boldsymbol{\Lambda}^k\boldsymbol{U}^\top)(\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top) \tag{19}$$

$$= \mathcal{L}^k\mathcal{L} = \mathcal{L}^{k+1} \tag{20}$$

Therefore:

$$\boldsymbol{U} g_\omega(\boldsymbol{\Lambda})\boldsymbol{U}^\top x = \sum_{i=0}^{\rho} \omega_i \mathcal{L}^i x = g_\omega(\mathcal{L})x$$

This equivalence explains why polynomial spectral filters can be implemented efficiently without eigendecomposition—they correspond exactly to polynomial operations on the Laplacian matrix itself. □

**Remark.** This result extends immediately to any polynomial basis (Chebyshev, Bernstein, etc.) since they are all linear combinations of monomials.

### F.2 Proof of Lemma 1

*Proof.* We begin by expanding the left-hand side of the expression. Distributing the product over the sum gives

$$\boldsymbol{U}S(\boldsymbol{\Lambda})\boldsymbol{U}^\top + \boldsymbol{U}P(\boldsymbol{\Lambda})\boldsymbol{U}^\top. \tag{21}$$

Next, we focus on the term $P(\boldsymbol{\Lambda})$, which is parameterized as a polynomial function of the eigenvalues of the graph Laplacian $\boldsymbol{\Lambda}$. Specifically, $P(\boldsymbol{\Lambda})$ can be written as

$$P(\boldsymbol{\Lambda}) = \sum_{i=0}^{\rho} \gamma_i \boldsymbol{\Lambda}^i, \tag{22}$$

where $\gamma_i$ are learnable coefficients and $\rho$ is the polynomial degree. Substituting this expansion for $P(\boldsymbol{\Lambda})$ into the expression, we get:

$$\boldsymbol{U}P(\boldsymbol{\Lambda})\boldsymbol{U}^\top = \boldsymbol{U}\left(\sum_{i=0}^{\rho} \gamma_i \boldsymbol{\Lambda}^i\right)\boldsymbol{U}^\top \tag{23}$$

$$= \sum_{i=0}^{\rho} \gamma_i \boldsymbol{U}\boldsymbol{\Lambda}^i\boldsymbol{U}^\top \tag{24}$$

The key insight is that for the eigendecomposition $\mathcal{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$, we have:

$$\boldsymbol{U}\boldsymbol{\Lambda}^i\boldsymbol{U}^\top = \mathcal{L}^i. \tag{25}$$

This holds by induction as shown in Appendix F.1. Substituting Eq. (25) into Eq. (24), we obtain

$$\boldsymbol{U}P(\boldsymbol{\Lambda})\boldsymbol{U}^\top = \sum_{i=0}^{\rho} \gamma_i \mathcal{L}^i = P(\mathcal{L}). \tag{26}$$

Finally, combining the two terms from Eq. (21), we get the desired decomposition

$$\boldsymbol{U}\left[S(\boldsymbol{\Lambda}) + P(\boldsymbol{\Lambda})\right]\boldsymbol{U}^\top = \boldsymbol{U}S(\boldsymbol{\Lambda})\boldsymbol{U}^\top + P(\mathcal{L}). \tag{27}$$

This completes the proof. □