Proceedings of Machine Learning Research **318** (2026)
The 39th Canadian Conference on Artificial Intelligence

Poly-WaveGC: A Generalized Spectral Wavelet Graph Convolution Network with Adaptive Orthogonal Polynomials

Wei Cao[†,*], Rachid Hedjam[†, △], Bessam Abdulrazak[‡], Jun-Peng Zhu[◇,*]
[†] Bishop's University, Sherbrooke, Quebec, Canada
[‡] Université de Sherbrooke, Sherbrooke, Quebec, Canada
[◇] Northwest A&F University, Xi'an, Shaanxi, China
[△] Université du Québec en Outaouais, Gatineau, Quebec, Canada

Abstract

Spectral Graph Neural Networks (GNNs) typically rely on fixed polynomial bases, such as Chebyshev polynomials, to approximate graph filters. While efficient, these bases enforce a rigid weight function that implicitly assumes a specific prior on the graph signal density, often leading to suboptimal fitting on graphs with diverse spectral distributions. In this paper, we propose Poly-WaveGC, a generalized spectral graph wavelet framework. We introduce a novel zero-point shift mechanism to adapt general Jacobi polynomials for wavelet construction. This approach allows the basis parameters (α, β) to flexibly learn the graph's specific spectral density while strictly enforcing the wavelet admissibility condition ($g(0) = 0$) structurally. To address the loss of orthogonality inherent in adaptive bases, we introduce an explicit frame-bound regularization that constrains the filter bank to approximate a tight frame, thereby guaranteeing numerical stability. Extensive experiments on 10 benchmarks demonstrate that Poly-WaveGC significantly outperforms fixed-basis baselines on diverse graph structures and tasks, while maintaining robustness in deep networks. The code is available at https://github.com/weicaocw/Poly-WaveGC-public.

Keywords: Spectral Graph Neural Networks, Graph Wavelets, Jacobi Polynomials

1. Introduction

Among various methods for graph representation learning, Graph Neural Networks (GNNs) excel consistently in nearly all tasks [1–3]. Spectral GNNs operate by designing filters for graph signals within the spectral domain. Through the adjustment of polynomial coefficients, spectral GNNs are capable of approximating various filter types. This flexibility extends their applicability to both homophilic and heterophilic graph structures [4].

Nowadays, spectral methods have progressed from early polynomial approximations to sophisticated learnable filter banks. Pioneering works such as ChebNet [5] utilized Chebyshev polynomials to circumvent expensive eigendecompositions, while GCN [6] further simplified this to a first-order approximation. Recently, the focus has shifted to improving the basis functions themselves. For example, BernNet [7], JacobiConv [8], and LegendreNet [9] explored diverse polynomial families to search for a balance between expressivity, computational efficiency, and numerical stability.

A more recent milestone is WaveGC [10], which utilized Chebyshev polynomials to generalize the graph wavelet transform. By separating the polynomial basis into odd and even terms, WaveGC elegantly satisfies the wavelet admissibility condition ($\hat{\psi}(0) = 0$), which previous works failed to achieve.

However, despite its innovations, WaveGC remains constrained by its dependency on the fixed Chebyshev basis. This reliance on a non-adaptive basis inevitably leads to a mismatch with the actual graph signal distribution [8]. Chebyshev polynomials (T_n) are orthogonal with respect to the weight function $(1 - x^2)^{-1/2}$, placing maximal emphasis on spectral boundaries. Yet, real-world graphs often exhibit diverse spectral densities that

[*] wcao23@ubishops.ca, zjp.dase@nwafu.edu.cn

differ significantly from this prior. Sticking to a fixed basis to satisfy admissibility limits the model's ability to adapt to varying distributions.

To address this limitation, we propose Poly-WaveGC, a generalized framework that integrates fully adaptive polynomial bases. Our approach moves beyond the "one-size-fits-all" constraint of fixed bases by exploring general Jacobi polynomials ($P_n^{(\alpha,\beta)}$), where α and β are learnable hyperparameters that dynamically adjust to the graph's empirical signal density. Crucially, general Jacobi polynomials lack the symmetry relied on by WaveGC's odd-even terms separation strategy. To overcome this, we introduce a novel zero-point shift mechanism, which strictly enforces the wavelet admissibility condition for arbitrary Jacobi bases without sacrificing their adaptivity.

Furthermore, the introduction of shifted, non-uniform bases inevitably compromises the orthogonality of the filter bank. We therefore introduce an explicit frame-bound regularization term in the loss function. This empirically improves numerical stability by constraining the learned filters to approximate a tight frame during training. Extensive experiments on 10 benchmark datasets demonstrate that aligning the mathematical properties of the basis with the graph's signal density has superior performance.

In summary, our primary contributions are:

(1) We propose Poly-WaveGC, which generalizes spectral graph wavelets to adaptive Jacobi polynomials. We resolve the conflict between basis adaptivity and wavelet admissibility via a novel zero-point shift mechanism.
(2) We address the loss of orthogonality inherent in adaptive bases by introducing explicit frame-bound regularization. This technique empirically improves the numerical stability of the learned filter banks.
(3) We achieve superior performance on 10 benchmark datasets, ranging from long-range graphs to heterophilic graphs and large-scale homophilic graphs.

2. Related Work

Spectral Graph Convolution. Spectral Graph Theory (SGT) [11] provides a mathematical framework for graph analysis, though explicit eigen-decomposition for the Graph Fourier Transform (GFT) [12] is computationally expensive ($O(N^3)$). To address this, research shifted toward approximating spectral filters via polynomials of the Laplacian [13], balancing efficiency with representation power. These models are generally categorized into fixed or learnable filters [7]. Early learnable approaches, such as ChebNet [5] and GPRGNN [4], optimized polynomial coefficients for adaptivity. Driven by the efficiency-expressivity trade-off, initial works like ChebNet [5] and GCN [6] utilized truncated Chebyshev polynomials or first-order approximations for rapid training.

Recent research focuses on basis diversity to enhance expressivity. BernNet [7] and LegendreNet [9] introduced Bernstein and Legendre polynomials to model arbitrary spectral shapes and mitigate boundary overfitting. However, these methods rely on fixed polynomial representations [8]. JacobiConv [8] generalized this to the Jacobi family, demonstrating that aligning the polynomial weight function with the graph's signal density significantly improves convergence, covering Chebyshev, Legendre, and Bernstein bases as special cases.

Graph Wavelets. To capture multi-scale structural information, graph wavelets extend classical wavelet theory [14] to the graph domain using filter banks [10]. SGWT [13] established the computational framework via spectral theory and Chebyshev approximation. However, methods like GWNN [15] often fail to strictly satisfy the wavelet admissibility condition [10, 14]. Other approaches, including UFGCONV [16], WaveNet [17], and WGGP [18], rely on rigid, predefined forms (e.g., Haar or Mexican Hat), limiting their adaptability across diverse datasets.

Effective graph wavelets require bases that are both mathematically valid and flexible [10]. While some prior works fail theoretical conditions [19, 20] or lack adaptability [16, 21], WaveGC [10] advanced the field with a learnable framework. It leverages Chebyshev parity decomposition to theoretically satisfy admissibility while approximating scaling and wavelet functions. Nevertheless, WaveGC remains constrained by the fixed Chebyshev basis. Since orthogonal bases aligned with the graph's spectral density achieve optimal convergence [8], we bridge this gap by incorporating the adaptive bases of JacobiConv [8] while strictly enforcing wavelet admissibility [14] through novel stability constraints.

3. Preliminaries

Graph Laplacian and Spectral Filters. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X)$ be an undirected graph with N nodes, adjacency matrix A, and degree matrix D. We utilize the normalized graph Laplacian $\hat{L} = I_N - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$. Being symmetric positive semi-definite, $\hat{L}$ admits an eigendecomposition $\hat{L} = U\Lambda U^\top$, with eigenvalues $0 \leq \lambda_i \leq 2$ and orthonormal eigenvectors U.

Spectral filtering modulates graph signals in the frequency domain. For a signal X and a filter $g : [0, 2] \to \mathbb{R}$, the operation is defined as $Z = g(\hat{L})X = Ug(\Lambda)U^\top X$. To avoid expensive eigendecomposition, spectral GNNs typically parameterize $g(\hat{L})$ as a polynomial of order K, i.e., $g(\hat{L}) = \sum_{k=0}^{K} \alpha_k \hat{L}^k$, which also guarantees spatial localization.

Wavelet Admissibility and Tight Frames. To act as a valid wavelet, $g(\lambda)$ must satisfy the admissibility condition to ensure the convergence of the inverse transform:

$$C_\Psi = \int_0^\infty \frac{|g(\lambda)|^2}{\lambda} d\lambda < \infty. \tag{3.1}$$

Crucially, this condition implies that the wavelet must have a zero mean, i.e., $g(0) = 0$. Multi-scale analysis is achieved by introducing scales s_j, yielding operators $\Psi_{s_j} = Ug(s_j\Lambda)U^\top$.

To ensure numerical stability and energy conservation during reconstruction, the filter bank $\{h, g_{s_j}\}$ should theoretically form a *tight frame*. This requires the sum of squared spectral responses to equal the identity (Parseval's identity):

$$G(\lambda) = h(\lambda)^2 + \sum_j g(s_j\lambda)^2 \equiv 1, \quad \forall \lambda \in [0, 2]. \tag{3.2}$$

Satisfying this condition implies that the transform is isometric, allowing for stable signal recovery.

4. Methodology

We propose Poly-WaveGC, a generalized spectral graph wavelet framework that extends WaveGC [10] from fixed symmetric bases to adaptive Jacobi polynomial bases while strictly preserving wavelet admissibility. The key change is that we no longer rely on an odd-even terms split which is available only for symmetric polynomials. Instead, we enforce the admissibility constraint $g(0) = 0$ via a DC-endpoint boundary shift, and stabilize the resulting non-orthogonal filter bank through an explicit frame-bound regularization.

4.1. Overall Architecture

Given node features $X \in \mathbb{R}^{N\times d}$, a Poly-WaveGC block applies a learnable scaling filter $h(\hat{L})$ and J learnable wavelet filters $\{g_j(\hat{L})\}_{j=1}^{J}$ to produce a multi-resolution representation:

$$Z = \mathrm{Fuse}\Big(h(\hat{L})X,\, g_1(\hat{L})X,\, \ldots,\, g_J(\hat{L})X \Big). \tag{4.1}$$

All filters are implemented as truncated Jacobi polynomial expansions computed via three-term recurrence, keeping the complexity linear in the number of edges. Besides, the coefficients of the polynomials and the scales are obtained from the encoding of the graph's eigenvalues, followed by an MLP and a pooling mechanism, as shown in Figure 1. For efficient spectral decomposition of large-scale graphs, such as ogbn-arxiv, we can employ randomized singular value decomposition [22]. This approach reduces computational complexity to $\mathcal{O}(N^2 \log K)$ by retaining only the top K eigenvectors.

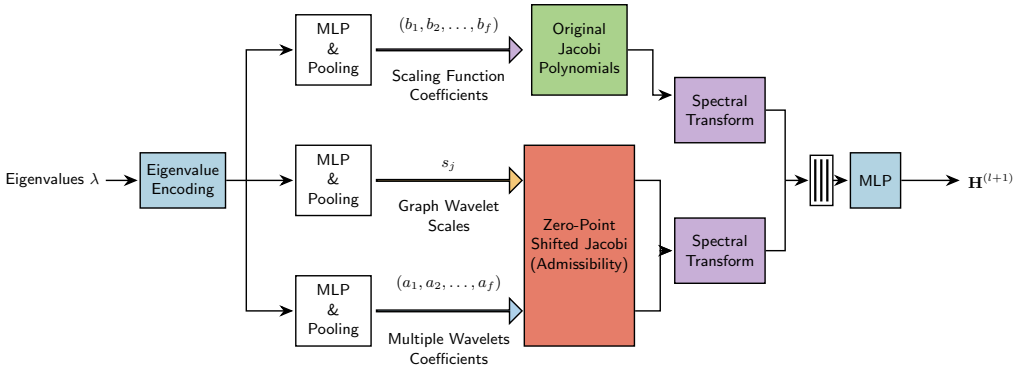

Figure 1. The architecture of the proposed Poly-WaveGC Block. By employing the zero-point shift scheme, we ensure wavelet admissibility without relying on symmetric bases.

4.2. **Jacobi Polynomial Bases on the Rescaled Spectrum**

Fixed symmetric bases (e.g., Chebyshev in WaveGC) may mismatch the graph's signal density, especially on graphs whose informative content concentrates at non-Chebyshev-friendly bands (e.g., heterophily with stronger high-frequency components). Inspired by the analysis in JacobiConv [8], we therefore adopt general Jacobi polynomials $P_k^{\alpha,\beta}$ with learnable (α, β) to adapt the basis shape to the signal density.

We rescale the spectrum to $[-1, 1]$:

$$\tilde{L} = \frac{2}{\lambda_{\max}} \hat{L} - I_N, \quad \tilde{\lambda} = \frac{2}{\lambda_{\max}} \lambda - 1, \tag{4.2}$$

The Jacobi basis $\{P_k^{\alpha,\beta}(\tilde{L})\}_{k=0}^{K}$ is generated using the following three-term recurrence. For $k = 0$ and $k = 1$:

$$\begin{aligned} P_0^{\alpha,\beta}(z) &= 1, \\ P_1^{\alpha,\beta}(z) &= \frac{\alpha - \beta}{2} + \frac{\alpha + \beta + 2}{2} z. \end{aligned} \tag{4.3}$$

For $k \geq 2$, the polynomials follow the recurrence relation:

$$P_k^{\alpha,\beta}(z) = (\theta_k z + \theta'_k) P_{k-1}^{\alpha,\beta}(z) - \theta''_k P_{k-2}^{\alpha,\beta}(z), \tag{4.4}$$

where the coefficients $\theta_k, \theta'_k, \theta''_k$ are defined as:

$$\begin{aligned} \theta_k &= \frac{(2k + \alpha + \beta)(2k + \alpha + \beta - 1)}{2k(k + \alpha + \beta)}, \\ \theta'_k &= \frac{(2k + \alpha + \beta - 1)(\alpha^2 - \beta^2)}{2k(k + \alpha + \beta)(2k + \alpha + \beta - 2)}, \\ \theta''_k &= \frac{(k + \alpha - 1)(k + \beta - 1)(2k + \alpha + \beta)}{k(k + \alpha + \beta)(2k + \alpha + \beta - 2)}. \end{aligned} \tag{4.5}$$

4.3. **Wavelet-Admissible Construction via DC Boundary Shift**

As reviewed in Preliminaries, wavelet admissibility requires $g(0) = 0$; otherwise the wavelet channel leaks DC (the constant eigenvector at $\lambda = 0$) into detail bands and the admissibility integral diverges.

General Jacobi polynomials do not admit a clean odd/even decomposition unless $\alpha = \beta$, hence odd-even-split based constructions are not applicable. We enforce $g(0) = 0$ by shifting each non-constant basis function at the DC endpoint $\tilde{\lambda} = -1$:

$$\psi_k^{(\alpha,\beta)}(\tilde{\lambda}) = P_k^{(\alpha,\beta)}(\tilde{\lambda}) - P_k^{(\alpha,\beta)}(-1)\, P_0^{(\alpha,\beta)}(\tilde{\lambda}), \qquad k \geq 1. \tag{4.6}$$

Since $P_0 \equiv 1$, this guarantees $\psi_k^{(\alpha,\beta)}(-1) = 0$, and thus any filter expanded on $\{\psi_k\}_{k\geq 1}$ satisfies $g(0) = 0$ structurally.

Under the standard Jacobi normalization, one has $P_k^{(\alpha,\beta)}(-1) = (-1)^k \binom{k+\beta}{k}$. If a different normalization is used in implementation, we compute $P_k^{(\alpha,\beta)}(-1)$ by running the same recurrence at $z = -1$.

4.4. **Learnable Multi-Resolution Filter Bank**

As illustrated in Figure 1, the eigenvalue encoding module dynamically generates scaling coefficients $\mathbf{b} = \{b_k\}_{k=1}^K$, shared wavelet coefficients $\mathbf{a} = \{a_k\}_{k=1}^K$, and wavelet scales $\mathbf{s} = \{s_j\}_{j=1}^J$.

By dilating the spectrum via $\tilde{L}_{s_j} = \frac{2}{\lambda_{\max}}(s_j \hat{L}) - I_N$, the scaling and multi-resolution wavelet filters are respectively parameterized as:

$$h(\hat{L}) = \sum_{k=1}^{K} b_k\, P_k^{(\alpha,\beta)}(\tilde{L}), \qquad g_j(\hat{L}) = \sum_{k=1}^{K} a_k\, \psi_k^{(\alpha,\beta)}(\tilde{L}_{s_j}), \quad j = 1, \ldots, J. \tag{4.7}$$

Since $\psi_k^{(\alpha,\beta)}(-1) = 0$, this construction structurally guarantees $g_j(0) = 0$ across all scales, inherently preserving wavelet admissibility.

4.5. **Explicit Frame-Bound Regularization**

The DC boundary shift is a linear recombination with the constant basis. While it enforces admissibility, it increases correlation among polynomial atoms, which may widen frame bounds and harm stability in deep stacks. We therefore explicitly regularize the frame bounds.

Let $\mathcal{W}X = (h(\hat{L})X,\ g_1(\hat{L})X, \ldots, g_J(\hat{L})X)$ and define the frame operator

$$\mathcal{S} = h(\hat{L})^2 + \sum_{j=1}^{J} g_j(\hat{L})^2, \tag{4.8}$$

with frame bounds $A = \lambda_{\min}(\mathcal{S})$ and $B = \lambda_{\max}(\mathcal{S})$ (tight when $A = B = 1$; see Preliminaries).

Since our framework relies on the eigendecomposition of the graph Laplacian to evaluate the spectral filters, we have direct access to the graph spectrum $\{\lambda_i\}_{i=1}^N$. This allows us to exactly evaluate the frame operator's spectral response at each valid eigenvalue:

$$S(\lambda_i) = h(\lambda_i)^2 + \sum_{j=1}^{J} g_j(\lambda_i)^2. \tag{4.9}$$

Instead of relying on approximations, the exact frame bounds over the active spectrum of the graph can be directly obtained by:

$$A = \min_{\lambda_i} S(\lambda_i), \qquad B = \max_{\lambda_i} S(\lambda_i). \tag{4.10}$$

To encourage the filter bank to form a tight frame, we penalize the deviation of these exact bounds from the Parseval target:

$$\mathcal{L}_{fb} = (B-1)^2 + (1-A)^2. \tag{4.11}$$

We optimize:

$$\mathcal{J} = \mathcal{L}_{task} + \gamma\, \mathcal{L}_{fb} \tag{4.12}$$

5. Experiments

We design our experiments to answer the following research questions:

(1) How effectively does the Poly-WaveGC framework capture and leverage long-range dependencies within complex graph structures?
(2) To what extent does the proposed framework maintain robust performance when navigating heterophilic graph structures compared to traditional GNN benchmarks?
(3) Does the integration of polynomial wavelets ensure numerical stability and prevent vanishing/exploding gradients as the network depth increases?

5.1. Experimental Setup

We run all experiments on a single NVIDIA RTX 3080 Ti GPU (12 GB) with a 12-core Intel Xeon Silver 4214R CPU and 90 GB of RAM, using Python 3.12, PyTorch 2.5.1, and CUDA 12.4 on Ubuntu 22.04.

Datasets. We selected 10 benchmark datasets, grouped into three categories, to test the model's performance rigorously. The selected benchmarks are from the PyTorch Geometric (PyG) and OGB libraries.

(1) Long-Range Graph Benchmark (LRGB): We utilized PascalVOC-SP (node classification), Peptides-func (graph classification), and Peptides-struct (graph regression) [23]. These datasets are critical for verifying if our Jacobi adaptive bases improve upon WaveGC's ability to capture distant node interactions, which are typically located in the low-frequency spectrum.
(2) Heterophilic graphs: We employed Chameleon, Squirrel [24], and Actor [25]. Unlike the original WaveGC study, which focused less on this area, these datasets test our model's capacity to handle high-frequency spectral components common in disassortative mixing.
(3) Homophilic & large-scale graphs: We included standard citation networks (Cora, Citeseer, PubMed) [26] and ogbn-arxiv [27] to demonstrate scalability on larger graphs.

The summary of the datasets is shown in Table 1.

Baselines. We compared Poly-WaveGC against three categories of methods: 1) Wavelet-based GNNs: WaveGC [10] (our primary competitor and backbone); 2) Polynomial Spectral GNNs: BernNet [7] (for non-negative filtering), JacobiConv [8] (for spectral density matching), LegendreNet [9], and ChebNet [5]; and 3) Other Basic GNNs: GCN [6] and GAT [28].

5.2. Results

In this section, we present a comprehensive comparison of Poly-WaveGC against baselines on three different graph categories: long-range dependency benchmarks, heterophilic graphs, and standard homophilic datasets. The quantitative results show that our proposed framework not only generalizes well across different graph topologies but also significantly outperforms fixed-basis spectral GNNs.

Dataset	Type	Size	Task	Description & Role in Experiments
PascalVOC-SP	LRGB	∼11k graphs	Node Class.	Core benchmark for validating long-range dependency capture.
Peptides-func	LRGB	∼15k graphs	Graph Class.	Tests ability to model long-range molecular interactions.
Peptides-struct	LRGB	∼15k graphs	Graph Reg.	Validates regression stability on complex structures.
Chameleon	Heterophilic	2,277 nodes	Node Class.	Critical for testing high-frequency filtering capabilities.
Squirrel	Heterophilic	5,201 nodes	Node Class.	Benchmark for graphs with disassortative mixing.
Actor	Heterophilic	7,600 nodes	Node Class.	Evaluates robustness on diverse spectral densities.
Cora	Homophilic	<3k nodes	Node Class.	Evaluate model on small scale homophilic graphs.
Citeseer	Homophilic	<4k nodes	Node Class.	Evaluate model on small-medium scale homophilic graphs.
PubMed	Homophilic	<20k nodes	Node Class.	Evaluate model on large scale homophilic graphs.
ogbn-arxiv	Large-scale	∼169k nodes	Node Class.	Demonstrates model scalability on larger citation networks.

Table 1. Summary of benchmark datasets used in experiments. The selection covers long-range dependencies (LRGB), heterophilic structures, and standard homophilic baselines to evaluate model versatility.

Spectral Preprocessing. The time cost and memory usage of the spectral preprocessing are reported in Table 2. For small-scale single-graph and multi-graph datasets in which each graph has a small number of nodes, we use full eigendecomposition; the former saves only the bottom 30% of the eigenpairs, while the latter saves all of them. For large-scale graphs, randomized singular value decomposition (RSVD) reserves 5000 eigenpairs for further computation. The time cost and memory usage of the spectral preprocessing are directly related to the number of nodes and graphs. Among all, ogbn-arxiv is the largest graph costing 1531.51 seconds and 34227 MiB of memory.

Dataset	Type	Method	N / $N_{\max}$	k saved	Time (s)	Peak RAM (MiB)
Cora	node	full	2,708	812	1.36	856
Citeseer	node	full	3,327	998	2.18	1,010
Chameleon	node	full	2,277	683	0.75	777
Squirrel	node	full	5,201	1,560	7.52	1,732
Actor	node	full	7,600	2,280	22.42	2,918
PubMed	node	RSVD (k=5000)	19,717	5,000	147.99	4,731
ogbn-arxiv	node	RSVD (k=5000)	169,343	5,000	1,531.51	34,227
VOCSuperpixels	graph	per-graph full	500	5,443,545	255.70	1,222
Peptides-*	graph	per-graph full	444	2,344,859	64.66	848

Table 2. Time cost and peak memory usage of the spectral preprocessing.

Performance on Long-Range Dependencies. As shown in Table 3, the results on the Long-Range Graph Benchmark (LRGB) highlight the significant advantage of the decoupled wavelet architecture. Standard polynomial methods (e.g., ChebNet, JacobiConv) struggle to capture distant interactions with F1 scores around 19% on PascalVOC-SP. In contrast, both WaveGC and Poly-WaveGC achieve a significant improvement in performance. Specifically, Poly-WaveGC further slightly outperforms the original WaveGC (e.g., +2.8% F1 score on PascalVOC-SP and +1.63% AP on Peptides-func).

Generalization to Heterophilic Graphs. As shown in Table 4, the experiments on heterophilic datasets (Chameleon, Squirrel, Actor) provide strong evidence for the significance of our adaptive basis. The original WaveGC falls behind baselines like BernNet and JacobiConv on Chameleon (63.77% vs. 66.86%), suggesting that the fixed Chebyshev basis struggles to filter the complex high-frequency components characteristic of heterophilic graphs (where connected nodes tend to have different labels). In contrast, Poly-WaveGC outperforms the second-best method (JacobiConv) by approximately 4.16% on Chameleon and 5.31% on Squirrel.

Robustness on Homophilic and Large-Scale Graphs. As shown in Table 5, on standard citation networks (Cora, Citeseer, PubMed), Poly-WaveGC consistently achieves the best performance, slightly outperforming JacobiConv. This confirms that the added expressivity does not lead to overfitting on simple tasks. In addition, on the large-scale ogbn-arxiv dataset, Poly-WaveGC surpasses WaveGC (73.26% vs. 73.01%), validating the scalability of our model. The wavelet-admissible Jacobi recurrence relation maintains the computational efficiency of the original framework while offering better spectral adaptivity, proving its effectiveness for large-scale applications.

Model	**PascalVOC-SP**	**Peptides-func**	**Peptides-struct**
	F1 score ↑	AP ↑	MAE ↓
GCN	12.35 ± 0.40	58.97 ± 0.21	32.73 ± 0.12
GAT	12.40 ± 0.67	53.35 ± 0.71	34.34 ± 0.39
ChebNet	18.98 ± 0.46	59.99 ± 0.33	27.45 ± 0.12
BernNet	19.42 ± 0.31	60.47 ± 0.26	27.06 ± 0.08
JacobiConv	19.36 ± 0.34	59.81 ± 0.29	27.18 ± 0.09
LegendreNet	19.39 ± 0.33	60.10 ± 0.28	27.09 ± 0.09
WaveGC	$\underline{39.87 \pm 0.25}$	$\underline{69.73 \pm 0.43}$	$\underline{24.83 \pm 0.11}$
Poly-WaveGC (Ours)	$\mathbf{42.67 \pm 0.56}$	$\mathbf{71.36 \pm 0.38}$	$\mathbf{24.69 \pm 0.10}$

Table 3. Performance comparison on Long-Range Graph Benchmark (LRGB). Metrics: F1 Score (↑) for VOC, Average Precision (AP, ↑) for Peptides-func, and MAE (↓) for Peptides-struct. Best results are **bolded**, second best are underlined.

Model	**Chameleon**	**Squirrel**	**Actor**
GCN	55.97 ± 1.78	44.23 ± 0.69	32.13 ± 1.28
GAT	61.26 ± 1.83	43.39 ± 1.01	33.01 ± 2.66
ChebNet	57.14 ± 1.13	38.99 ± 0.58	38.07 ± 0.79
BernNet	$\underline{66.15 \pm 1.46}$	49.87 ± 0.65	$\underline{39.43 \pm 0.91}$
JacobiConv	$\underline{66.86 \pm 1.77}$	$\underline{49.88 \pm 0.63}$	39.27 ± 0.90
LegendreNet	59.61 ± 1.91	41.55 ± 0.74	39.05 ± 0.78
WaveGC	63.77 ± 1.33	47.49 ± 0.52	37.05 ± 0.79
Poly-WaveGC (Ours)	$\mathbf{71.02 \pm 1.73}$	$\mathbf{55.19 \pm 0.81}$	$\mathbf{41.27 \pm 1.11}$

Table 4. Classification Accuracy (↑) on Heterophilic Graphs. Comparisons with BernNet highlight the importance of high-frequency spectral filtering. Best results are **bolded**, second best are underlined.

5.3. **Ablation Study**

Frame Potential Regularization and Deep Stability. A theoretical prerequisite for stable wavelet transforms is the tight frame property (i.e., frame bounds $A \approx B \approx 1$)

Model	Cora	Citeseer	PubMed	ogbn-arxiv
GCN	85.75 ± 1.20	76.42 ± 0.89	82.94 ± 0.38	70.33 ± 0.32
GAT	87.27 ± 0.96	78.29 ± 0.77	85.96 ± 0.36	70.98 ± 0.31
ChebNet	86.95 ± 0.81	78.69 ± 0.76	86.50 ± 0.36	70.62 ± 0.31
BernNet	88.36 ± 0.94	79.94 ± 0.78	88.21 ± 0.40	70.77 ± 0.30
JacobiConv	$\underline{88.76 \pm 0.45}$	$\underline{80.57 \pm 0.78}$	$\underline{89.37 \pm 0.40}$	71.20 ± 0.30
LegendreNet	88.19 ± 0.93	79.82 ± 0.77	88.03 ± 0.40	70.95 ± 0.30
WaveGC	87.51 ± 0.88	79.26 ± 0.76	87.29 ± 0.37	$\underline{73.01 \pm 0.18}$
Poly-WaveGC (Ours)	$\mathbf{88.96 \pm 0.44}$	$\mathbf{80.78 \pm 0.79}$	$\mathbf{89.62 \pm 0.41}$	$\mathbf{73.26 \pm 0.17}$

Table 5. Classification Accuracy (↑) on Standard Homophilic and Large-Scale datasets. Comparisons on ogbn-arxiv demonstrate scalability. Best results are **bolded**, second best are underlined.

[14], which ensures energy conservation and numerical stability (Parseval's identity). While WaveGC attempts to satisfy this implicitly via initialization, the free parameters of the neural network will probably deviate during stochastic gradient descent, damaging the frame structure. This worsens the condition number of the filter bank, causing gradient explosion or vanishing in deep architectures.

To solve this problem, we introduced an explicit frame-bound regularization term in the loss function. This term explicitly penalizes the deviation of the learned filters from a tight frame.

To validate its efficacy, we conducted a stability stress test by increasing the network depth from 2 to 32 layers on the Cora and Amazon Computers datasets. We compared our full model (Poly-WaveGC w/ Reg) against a variant without regularization (w/o Reg) and the original WaveGC.

As shown in Table 6 and Figure 2, standard spectral GNNs suffer from a severe performance collapse as depth increases. The original WaveGC and the unregularized Poly-WaveGC exhibit a sharp drop in accuracy beyond 8 layers. At 32 layers, they degenerate to random guessing or suffer from gradient explosion (marked as OOM/NaN). In contrast, Poly-WaveGC equipped with $\mathcal{L}_{reg}$ maintains robust performance even at 32 layers. The accuracy drop is minor ($< 2\%$), and the training remains numerically stable. This confirms that explicitly enforcing the tight frame property acts as a vital stabilizer.

Depth	Cora (Accuracy %)				Computers (Accuracy %)			
	2 L	8 L	16 Layers	32 L	2 L	8 L	16 L	32 L
GCN	87.14	75.20	21.50	13.20	83.32	78.10	35.40	30.10
WaveGC	87.51	84.20	65.40	*Fail*	87.54	86.10	72.30	*Fail*
Poly-WaveGC (w/o Reg)	**88.98**	86.50	68.10	42.50	**88.65**	87.20	75.40	55.30
Poly-WaveGC (w/ Reg)	88.96	**88.10**	**87.40**	**86.80**	88.58	**88.20**	**87.90**	**87.10**

Table 6. Deep GNN Stability Test. We report classification accuracy (↑) across varying depths. "*Fail*" indicates training divergence (NaN loss) or severe performance collapse ($< 20\%$). Poly-WaveGC with Frame Regularization demonstrates superior robustness, maintaining high accuracy even at 32 layers (32 L).

6. Discussion

We analyze Poly-WaveGC's performance through three theoretical lenses: basis adaptivity, structural admissibility, and frame stability.

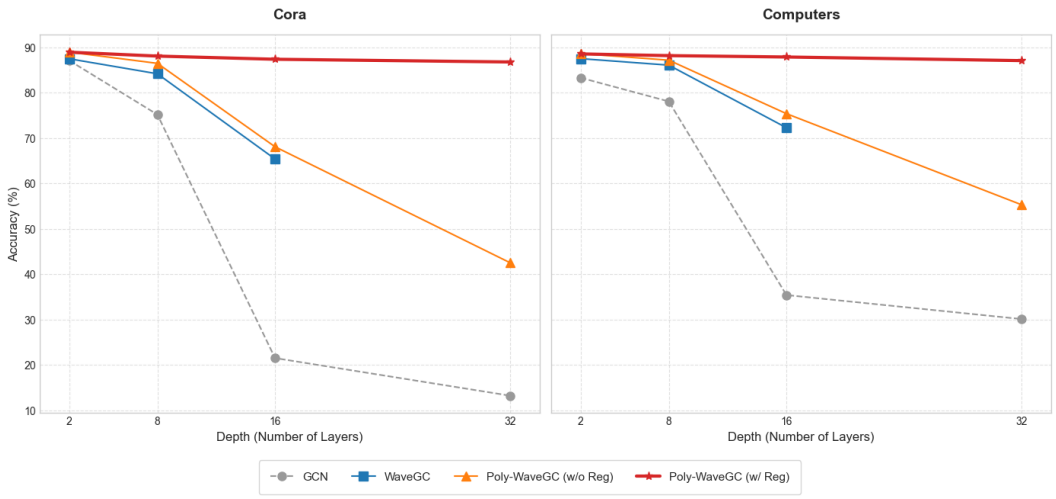

Figure 2. Deep GNN Stability Test. We report classification accuracy (↑) across varying depths. Poly-WaveGC with Frame Regularization demonstrates superior robustness, maintaining high accuracy even at 32 layers.

Basis Adaptivity vs. Fixed Priors. Poly-WaveGC's superiority on heterophilic graphs comes from overcoming the limitations of fixed bases. Chebyshev polynomials impose a rigid prior that concentrates weight on spectral endpoints, which misaligns with the irregular energy profiles of heterophilic data. By treating Jacobi parameters (α, β) as learnable, Poly-WaveGC dynamically aligns the basis orthogonality with the graph's signal density. This alignment minimizes the approximation error for complex filters at the same polynomial order K, enhancing expressivity without increasing computational cost.

Structural Admissibility via Zero-Point Shift. Constructing valid wavelets from asymmetric Jacobi bases ($\alpha \neq \beta$) makes WaveGC's odd-even separation inapplicable. Instead, we enforce admissibility via a zero-point shift. By defining shifted atoms as $\psi_k^{(\alpha,\beta)}(\tilde{\lambda}) = P_k^{(\alpha,\beta)}(\tilde{\lambda}) - P_k^{(\alpha,\beta)}(-1)$, we structurally guarantee $\psi_k(-1) = 0$. This ensures that all wavelet filters strictly satisfy $g(0) = 0$, preventing DC leakage into detail channels. This exact separation of low-pass and band-pass components explains the model's robustness on long-range tasks.

Depth Stability via Frame Regularization. While the zero-point shift ensures admissibility, it inevitably increases basis correlation, pushing the filter bank away from a Parseval tight frame. This results in a frame operator $\mathcal{S}$ with loose bounds and a large condition number $\kappa = B/A$, which amplifies gradients in deep networks. Our explicit frame-bound regularization directly penalizes deviations from the identity target $A = B = 1$. By keeping $\mathcal{S}$ well-conditioned, Poly-WaveGC prevents energy distortion, maintaining performance at depths (e.g., 32 layers) where unregularized baselines collapse.

7. Conclusion and Future Work

Conclusion. We presented Poly-WaveGC, a framework that overcomes fixed polynomial settings in spectral GNNs using Adaptive General Jacobi Polynomials. Matching diverse signal densities yields significant performance improvements on heterophilic and long-range benchmarks. Crucially, we reconciled basis flexibility with mathematical stability. Through a structural zero-point shift and explicit frame-bound regularization, Poly-WaveGC enforces strict wavelet admissibility and restores the tight frame property. Combined with linearly

scalable three-term recurrences, it establishes an efficient paradigm for adaptive spectral GNNs.

Future Work. Although Poly-WaveGC improves spectral adaptivity, it remains limited by polynomial approximation limits; as FIR filters, polynomials inherently suffer from the Runge Phenomenon when approximating sharp cutoffs [14]. Future research will therefore transition to Rational Spectral Filters (i.e., IIR filters, $r(\lambda) = p(\lambda)/q(\lambda)$). By introducing poles, rational functions can approximate singular spectral functions with exponential convergence and fewer parameters [29]. Specifically, we aim to develop a differentiable variant of the AAA algorithm [30] for GNNs. This Rational-AAA paradigm will efficiently model complex singularities in large-scale datasets, ultimately transforming spectral graph theory from polynomial to rational.

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC # DDG-2025-00036) Fund.

The authors used Gemini for editing and polishing the manuscript and take full responsibility for the final manuscript's content.

References

[1] L. Yao, C. Mao, and Y. Luo. *Graph Convolutional Networks for Text Classification*. 2018. arXiv: 1809.05679 [cs.CL]. URL: https://arxiv.org/abs/1809.05679.

[2] A. Fout, J. Byrd, B. Shariat, and A. Ben-Hur. "Protein interface prediction using graph convolutional networks". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, 6533–6542. ISBN: 9781510860964.

[3] J. Chen, T. Ma, and C. Xiao. *FastGCN: Fast Learning with Graph Convolutional Networks via Importance Sampling*. 2018. arXiv: 1801.10247 [cs.LG]. URL: https://arxiv.org/abs/1801.10247.

[4] E. Chien, J. Peng, P. Li, and O. Milenkovic. *Adaptive Universal Generalized PageRank Graph Neural Network*. 2021. arXiv: 2006.07988 [cs.LG]. URL: https://arxiv.org/abs/2006.07988.

[5] M. Defferrard, X. Bresson, and P. Vandergheynst. *Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering*. 2017. arXiv: 1606.09375 [cs.LG]. URL: https://arxiv.org/abs/1606.09375.

[6] T. N. Kipf and M. Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. 2017. arXiv: 1609.02907 [cs.LG]. URL: https://arxiv.org/abs/1609.02907.

[7] M. He, Z. Wei, Z. Huang, and H. Xu. *BernNet: Learning Arbitrary Graph Spectral Filters via Bernstein Approximation*. 2022. arXiv: 2106.10994 [cs.LG]. URL: https://arxiv.org/abs/2106.10994.

[8] X. Wang and M. Zhang. *How Powerful are Spectral Graph Neural Networks*. 2022. arXiv: 2205.11172 [cs.LG]. URL: https://arxiv.org/abs/2205.11172.

[9] J. Chen and L. Xu. "Improved Modeling and Generalization Capabilities of Graph Neural Networks With Legendre Polynomials". In: *IEEE Access* 11 (2023), pp. 63442–63450. DOI: 10.1109/ACCESS.2023.3289002.

[10] N. Liu, X. He, T. Laurent, F. D. Giovanni, M. M. Bronstein, and X. Bresson. *A General Graph Spectral Wavelet Convolution via Chebyshev Order Decomposition*. 2025. arXiv: 2405.13806 [cs.LG]. URL: https://arxiv.org/abs/2405.13806.

[11] F. R. K. Chung. *Spectral graph theory*. Vol. 92. American Mathematical Society, 1997.

[12] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains". In: *IEEE Signal Processing Magazine* 30.3 (2013), pp. 83–98.

[13] D. K. Hammond, P. Vandergheynst, and R. Gribonval. "Wavelets on graphs via spectral graph theory". In: *Applied and Computational Harmonic Analysis* 30.2 (2011), pp. 129–150. ISSN: 1063-5203. DOI: https://doi.org/10.1016/j.acha.2010.04.005. URL: https://www.sciencedirect.com/science/article/pii/S1063520310000552.

[14] S. Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. Third. Burlington, MA: Academic Press, 2009. ISBN: 978-0-12-374370-1.

[15] B. Xu, H. Shen, Q. Cao, Y. Qiu, and X. Cheng. *Graph Wavelet Neural Network*. 2019. arXiv: 1904.07785 [cs.LG]. URL: https://arxiv.org/abs/1904.07785.

[16] X. Zheng, B. Zhou, J. Gao, Y. G. Wang, P. Lio, M. Li, and G. Montufar. *How Framelets Enhance Graph Neural Networks*. 2021. arXiv: 2102.06986 [cs.LG]. URL: https://arxiv.org/abs/2102.06986.

[17] Z. Yang, Y. Hu, S. Ouyang, J. Liu, S. Wang, X. Ma, W. Wang, H. Su, and Y. Liu. "WaveNet: Tackling Non-stationary Graph Signals via Graph Spectral Wavelets". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 38.8 (2024), pp. 9287–9295.

[18] F. L. Opolka, Y.-C. Zhi, P. Liò, and X. Dong. *Adaptive Gaussian Processes on Graphs via Spectral Graph Wavelets*. 2022. arXiv: 2110.12752 [cs.LG]. URL: https://arxiv.org/abs/2110.12752.

[19] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. *How Powerful are Graph Neural Networks?* 2019. arXiv: 1810.00826 [cs.LG]. URL: https://arxiv.org/abs/1810.00826.

[20] M. Xu, W. Dai, C. Li, J. Zou, H. Xiong, and P. Frossard. *Graph Neural Networks With Lifting-based Adaptive Graph Wavelets*. 2022. arXiv: 2108.01660 [cs.LG]. URL: https://arxiv.org/abs/2108.01660.

[21] H. Cho, M. Jeong, S. Jeon, S. Ahn, and W. H. Kim. "Multi-resolution Spectral Coherence for Graph Generation with Score-based Diffusion". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023. URL: https://openreview.net/forum?id=qUlpDjYnsp.

[22] N. Halko, P.-G. Martinsson, and J. A. Tropp. *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*. 2010. arXiv: 0909.4061 [math.NA]. URL: https://arxiv.org/abs/0909.4061.

[23] V. P. Dwivedi, L. Rampášek, M. Galkin, A. Parviz, G. Wolf, A. T. Luu, and D. Beaini. *Long Range Graph Benchmark*. 2023. arXiv: 2206.08164 [cs.LG]. URL: https://arxiv.org/abs/2206.08164.

[24] B. Rozemberczki, C. Allen, and R. Sarkar. *Multi-scale Attributed Node Embedding*. 2021. arXiv: 1909.13021 [cs.LG]. URL: https://arxiv.org/abs/1909.13021.

[25] H. Pei, B. Wei, K. C.-C. Chang, Y. Lei, and B. Yang. *Geom-GCN: Geometric Graph Convolutional Networks*. 2020. arXiv: 2002.05287 [cs.LG]. URL: https://arxiv.org/abs/2002.05287.

[26] Z. Yang, W. W. Cohen, and R. Salakhutdinov. *Revisiting Semi-Supervised Learning with Graph Embeddings*. 2016. arXiv: 1603.08861 [cs.LG]. URL: https://arxiv.org/abs/1603.08861.

[27] W. Hu, M. Fey, H. Ren, M. Nakata, Y. Dong, and J. Leskovec. *OGB-LSC: A Large-Scale Challenge for Machine Learning on Graphs*. 2021. arXiv: 2103.09430 [cs.LG]. URL: https://arxiv.org/abs/2103.09430.

[28] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. *Graph Attention Networks*. 2018. arXiv: 1710.10903 [stat.ML]. URL: https://arxiv.org/abs/1710.10903.

[29] L. N. Trefethen. *Approximation Theory and Approximation Practice, Extended Edition*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2019. DOI: 10.1137/1.9781611975949. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611975949. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611975949.

[30] Y. Nakatsukasa, O. Sète, and L. N. Trefethen. "The AAA Algorithm for Rational Approximation". In: *SIAM Journal on Scientific Computing* 40.3 (Jan. 2018), A1494–A1522. ISSN: 1095-7197. DOI: 10.1137/16m1106122. URL: http://dx.doi.org/10.1137/16M1106122.