Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# SpectralMLP: Message-Passing-Free Dual-Frequency Spectral Filtering on Hypergraphs

**Rong Qian**
Beijing Electronic Science and Technology Institute
**Yu Cheng**
chengyu_cs@zjut.edu.cn

Beijing Electronic Science and Technology Institute
**Hongbo Zhao**
Beijing Electronic Science and Technology Institute

**Research Article**

**Keywords:** Hypergraph learning, Spectral filtering, Message-passing-free, Dual-frequency, Jacobi polynomial

**Posted Date:** July 31st, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-10337318/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** No competing interests reported.

# SpectralMLP: Message-Passing-Free Dual-Frequency Spectral Filtering on Hypergraphs

Rong Qian[1], Yu Cheng[1*], Hongbo Zhao[1*]

[1]School of Cybersecurity, Beijing Electronic Science and Technology Institute, Beijing, China.

*Corresponding author(s). E-mail(s): chengyu_cs@zjut.edu.cn; 202203150831@zjut.edu.cn;
Contributing authors: rqian@besti.edu.cn;

**Abstract**

Whether through local message passing, global attention, or spectral projection, hypergraph neural networks almost uniformly rely on *aggregation* to enrich node representations. Spectral filtering, however, is theoretically equivalent to spatial aggregation and can match its representational power without aggregation's practical overhead. We introduce **SpectralMLP**, a framework that replaces aggregation with **dual-frequency spectral filtering** via learnable Jacobi polynomial bases. Its central module, the **Adaptive Dual-Frequency Spectral Filter (ADF-SF)**, decomposes hidden representations into low- and high-frequency branches, each governed by its own Jacobi parameters, and recombines them through a learned gate. The entire operation reduces to sparse matrix-vector products, requiring neither eigendecomposition nor neighborhood expansion. Across 10 hypergraph node classification benchmarks and 9 baselines, SpectralMLP ranks first or second on 7 datasets when configured with the best filter mode per dataset (dual-frequency as default, single-branch modes as task-specific options). The largest gains appear where existing methods struggle: **+27.9%** over the MLP baseline on Congress (under uniform settings; severe spectral degeneration) and **+3.2%** on ModelNet40 (40 classes). SpectralMLP adds a median of roughly **2.4%** additional parameters over the MLP backbone, and its total parameter count is lower than most message-passing baselines.

**Keywords:** Hypergraph learning, Spectral filtering, Message-passing-free, Dual-frequency, Jacobi polynomial

# 1 Introduction

Social networks, biological systems, and computer vision all involve interactions among groups of entities, not just pairs. Hypergraphs capture such multi-way relationships through hyperedges, enabling richer modeling than ordinary graphs Kim et al. (2024).

Despite their diversity, virtually all existing hypergraph neural networks (HGNNs) share a common structural pattern: they *aggregate* information from other nodes to enrich each node's representation. Message-passing methods Feng et al. (2019); Yadati et al. (2019); Chien et al. (2022) aggregate from $k$-hop neighbors; global attention methods Liu et al. (2024) aggregate from all nodes via quadratic-complexity self-attention Vaswani et al. (2017); and eigendecomposition-based spectral methods Ai et al. (2025) aggregate via spectral projections. Each aggregation strategy carries a fundamental cost. Local message passing suffers from *over-squashing* Alon and Yahav (2020) and noise sensitivity. Global attention incurs $\mathcal{O}(N^2)$ complexity, limiting scalability to large hypergraphs. Exact eigendecomposition costs $\mathcal{O}(N^3)$, making it infeasible beyond a few thousand nodes.

A natural question arises: *can we bypass aggregation entirely and instead **filter**?* Spectral filtering selectively enhances or suppresses specific frequency components in node representations, offering a fundamentally different mechanism for leveraging graph structure. Crucially, spectral filtering is *theoretically equivalent* to spatial aggregation: any finite-step neighbor aggregation can be expressed as a spectral filter, and vice versa. This equivalence means that spectral filtering can achieve the same representational power as message passing, but without its practical drawbacks. Polynomial spectral filters (Chebyshev Defferrard et al. (2016), Jacobi Wang and Zhang (2022)) further avoid eigendecomposition by operating directly on the sparse Laplacian through recurrence relations, achieving $\mathcal{O}(K \cdot \text{nnz} \cdot H)$ complexity per layer.

Yet applying spectral filtering to hypergraphs raises a critical question: *which frequencies matter?* Different hypergraph tasks exhibit fundamentally different spectral characteristics. Homophilic graphs benefit from low-frequency signals that smooth node features across communities; heterophilic and multi-class graphs require high-frequency signals that sharpen inter-class boundaries. A particularly challenging case is *spectral degeneration*: when Laplacian eigenvalues cluster tightly (*e.g.*$\lambda_{\max} - \lambda_{\min} < 10^{-4}$ in Congress), standard spectral filtering collapses to a constant response, losing all frequency discrimination. No single spectral response can handle this diversity.

**SpectralMLP** is our answer: a message-passing-free framework that answers "which frequencies matter" through **dual-frequency spectral filtering** with learnable Jacobi polynomials. Its core module, the **Adaptive Dual-Frequency Spectral Filter (ADF-SF)**, decomposes node representations into independently parameterized low- and high-frequency branches, each with its own learnable Jacobi polynomial basis, and adaptively fuses them via a data-driven gate. The entire operation reduces to sparse matrix-vector products, requiring neither eigendecomposition nor neighbor expansion.

SpectralMLP demonstrates that spectral filtering alone—without any learnable aggregation operators—suffices for hypergraph node classification; ADF-SF replaces the message-passing paradigm with a dual-frequency Jacobi polynomial filter whose only graph operation is sparse matrix multiplication with the fixed Laplacian $\hat{\mathbf{L}}$, at

$\mathcal{O}(K \cdot \text{nnz} \cdot H)$ per layer, yet achieves competitive or superior accuracy to message-passing methods with fewer parameters. Building on this result, we bring learnable Jacobi polynomial filtering to hypergraphs for the first time and pair it with a dual-frequency architecture that captures both homophily and heterophily signals; the learnable $(\alpha, \beta)$ parameters adapt the polynomial basis to each dataset's spectral distribution at the cost of only 2 extra parameters per filter, and an adaptive gate automatically discovers the optimal frequency balance. Experiments on 10 datasets against 9 baselines validate these choices: SpectralMLP ranks first or second on 7 datasets, with a median of roughly 2.4% additional parameters over the MLP backbone (see Section 4.2.5). The largest gains appear where existing methods struggle: +27.9% over the MLP baseline on Congress (under uniform settings) and +3.2% on ModelNet40 (40 classes), all achieved with fewer parameters than most message-passing baselines.

Section 2 positions our work within the hypergraph learning and spectral filtering literature. Section 3 introduces the necessary background on hypergraph Laplacians and spectral filtering. We describe the SpectralMLP framework in Section 4 and present experimental results in Section 5, closing with a discussion of limitations and future directions in Section 6.

# 2 Related Work

## 2.1 Hypergraph Neural Networks

The landscape of HGNN design is dominated by *aggregation-based* methods that enrich node representations by collecting information from other nodes Kim et al. (2024); Yang and Xu (2025). On the message-passing side, HGNN Feng et al. (2019) and HyperGCN Yadati et al. (2019) work on clique-expanded graphs; UniGNN Huang and Yang (2021) offers a unified convolution framework spanning several GCN variants; AllSet Chien et al. (2022) replaces fixed aggregation with learned set functions; and HNHN Dong et al. (2020) adds hyperedge-level updates. These methods work well in practice, but they all require $k$-hop neighborhood expansion, a cost that grows with both graph size and depth. Their iterative aggregation also makes it hard to disentangle what individual frequency bands contribute. ED-HNN Wang et al. (2023) introduces diffusion-based operators on hypergraphs, while SheafHyperGCN Duta et al. (2023) leverages sheaf-theoretic constructions, but both remain within the aggregation paradigm.

Global attention methods take a different approach to aggregation: HyperGT Liu et al. (2024) applies transformer-style self-attention across all node pairs, capturing long-range dependencies but at $\mathcal{O}(N^2)$ cost, which limits scalability. CEGAT Veličković et al. (2018a) combines graph attention Veličković et al. (2018b) with hypergraph-specific encodings. While these methods differ in aggregation scope (local neighborhoods versus global node pairs), they share a common reliance on collecting information *from other nodes*.

A notable exception is the class of MLP-based methods that avoid neighbor aggregation entirely. Graph-MLP Hu et al. (2021) demonstrates that MLPs with structure-aware regularization can be effective on graphs; Hypergraph-MLP Yi et al. (2023) extends this idea to hypergraphs. TF-HNN Tang et al. (2025) uses temporal

features without aggregation. However, these methods lack an explicit mechanism for incorporating spectral information from the hypergraph structure.

A critical observation across all aggregation-based methods is that message-passing implicitly implements low-pass filtering: each aggregation step smooths node features, attenuating high-frequency components. This connection motivates our approach. Rather than aggregating as a proxy for spectral smoothing, we *filter* directly in the spectral domain, gaining explicit frequency control while avoiding aggregation's practical costs.

## 2.2 Spectral Graph and Hypergraph Methods

Spectral methods perform filtering in the eigen-domain of the graph Laplacian Chung (1997); Bo et al. (2023), a paradigm first established for neural networks by Bruna et al. Bruna et al. (2013). A key theoretical insight is that spectral filtering is equivalent to finite-step spatial aggregation Zhu et al. (2021): any spectral filter can be expressed as a localized aggregation, and vice versa. This equivalence suggests that aggregation is not *necessary* for leveraging graph structure; filtering suffices. Several works further show that spectral filtering can avoid eigendecomposition entirely He et al. (2022). On ordinary graphs, BernNet He et al. (2021) learns arbitrary spectral responses via Bernstein polynomials, JacobiConv Wang and Zhang (2022) introduces learnable Jacobi polynomial bases, and GPRGNN Chien et al. (2021) uses generalized polynomial residuals. These works reveal that the polynomial basis itself matters: JacobiConv demonstrates that learnable $(\alpha, \beta)$ parameters adapt to each dataset's spectral distribution, outperforming fixed Chebyshev bases. Yet none of these methods were built for hypergraphs. They ignore the spectral quirks that clique expansion introduces and offer no mechanism for spectral degeneration.

On hypergraphs, HyperAim Li et al. (2026) employs adaptive multi-frequency filtering via polynomial and framelet transforms for contrastive learning, but relies on message passing with substantial parameter overhead. ACF-HNN Huang et al. (2026) adaptively captures both low- and high-frequency signals through a channel-wise mechanism, but similarly depends on message passing. GrokFormer Ai et al. (2025) achieves strong performance on ordinary graphs via learnable Fourier-series spectral filters, but requires precomputed eigendecomposition, limiting scalability to large hypergraphs. SpectralMLP's polynomial approach avoids this bottleneck by operating directly on the sparse Laplacian through recurrence relations. It also applies frequency decomposition directly to learned node representations, rather than within the message-passing loop.

A shared challenge is spectral degeneration: when eigenvalues cluster tightly, standard spectral filtering degenerates to an approximately constant response, losing frequency discrimination. ACF-HNN Huang et al. (2026) addresses spectral diversity through multi-band filtering (low-pass, high-pass, band-pass) with learnable bandwidth, but operates within the message-passing paradigm. SpectralMLP instead applies dual-frequency decomposition with independently parameterized *learnable Jacobi polynomial bases* directly to learned representations, without any aggregation operators. This decoupling from message passing allows each branch to specialize purely in frequency space. We construct the Laplacian via clique expansion because it retains

the well-understood spectral interpretation from graph theory (eigenvalues in $[0, 2]$, Cheeger inequalities), which simplifies polynomial filter design.

### 2.3 Polynomial Spectral Filters

Polynomial approximations to spectral filters He et al. (2022) avoid the $\mathcal{O}(N^3)$ cost of eigendecomposition by operating directly on the Laplacian. Different polynomial families carry different inductive biases: Chebyshev polynomials emphasize the ends of the eigenvalue interval, whereas Jacobi polynomials with learnable $(\alpha, \beta)$ can shift spectral weight adaptively Wang and Zhang (2022), a useful property under spectral degeneration. However, standard polynomial filters apply a single spectral response $g(\lambda)$ across the entire eigenvalue range and cannot simultaneously extract low-frequency homophily and high-frequency heterophily signals. We extend learnable Jacobi polynomial filtering to hypergraphs and introduce a dual-frequency design that decomposes filtering into two independently parameterized branches.

Table 1 summarizes the key distinctions between SpectralMLP and related methods.

**Table 1:** Qualitative comparison with representative methods. MP-Free: no explicit neighbor aggregation (message passing); note that global attention methods (HyperGT) avoid message passing but still aggregate information across node pairs. Explicit SF: explicitly designed spectral filtering; LB: learnable spectral basis; HG: hypergraph-native.

| Method | MP-Free | Explicit SF | LB | HG |
|---|---|---|---|---|
| GCN Kipf and Welling (2017) | × | × | × | × |
| BernNet He et al. (2021) | × | ✓ | × | × |
| JacobiConv Wang and Zhang (2022) | × | ✓ | ✓ | × |
| HGNN Feng et al. (2019) | × | × | × | ✓ |
| GrokFormer Ai et al. (2025) | ✓ | ✓ | ✓ | × |
| HyperAim Li et al. (2026) | × | ✓ | × | ✓ |
| ACF-HNN Huang et al. (2026) | × | ✓ | × | ✓ |
| Hypergraph-MLP Yi et al. (2023) | ✓ | × | × | ✓ |
| **SpectralMLP (ours)** | ✓ | ✓ | ✓ | ✓ |

## 3 Preliminary

### 3.1 Hypergraph and Laplacian

A hypergraph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{H})$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ is the node set, $\mathcal{E} = \{e_1, \ldots, e_{|\mathcal{E}|}\}$ is the hyperedge set, and $\mathbf{H} \in \{0, 1\}^{N \times |\mathcal{E}|}$ is the incidence matrix with $H_{ve} = 1$ if node $v \in e$ and 0 otherwise Zhou et al. (2006).

The *clique expansion* transforms the hypergraph into an ordinary graph. The normalized Laplacian is:

$$\mathbf{L} = \mathbf{I} - \mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{D}_v^{-1/2} \tag{1}$$

where $\mathbf{D}_v$ and $\mathbf{D}_e$ are the diagonal degree matrices for nodes and hyperedges, respectively. To ensure eigenvalues lie in $[-1, 1]$, we use the scaled Laplacian:

$$\hat{\mathbf{L}} = \frac{2}{\lambda_{\max}}\mathbf{L} - \mathbf{I} \tag{2}$$

where $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{L}$, computed efficiently via power iteration.

***Remark on Clique Expansion.***

We choose clique expansion because it preserves well-understood spectral properties (eigenvalues in $[0, 2]$, Cheeger inequalities) and enables efficient polynomial filtering via sparse matrix operations. Alternative constructions (*e.g.*star expansion, tensor-based Laplacians) are left for future work.

## 3.2 Spectral Filtering on Hypergraphs

By the spectral theorem, the Laplacian admits eigendecomposition $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$. Spectral filtering is expressed as $\mathbf{h}' = \mathbf{U} \cdot g(\mathbf{\Lambda}) \cdot \mathbf{U}^\top\mathbf{h}$, where $g(\cdot)$ is the spectral response function. Defining the low-pass projection $\mathbf{P} = \mathbf{U}\, g(\mathbf{\Lambda})\, \mathbf{U}^\top$, the signal decomposes as $\mathbf{h} = \mathbf{P}\mathbf{h} + (\mathbf{I} - \mathbf{P})\mathbf{h}$, yielding a lossless decomposition into low- and high-frequency components.

***Spectral Degeneration.***

As noted in Section 1, when $\lambda_{\max} - \lambda_{\min} < \epsilon$ for a small threshold $\epsilon$, spectral filtering degenerates to an approximately constant response, losing frequency discrimination. On Congress, $\lambda_{\max} - \lambda_{\min} < 10^{-4}$, making this a particularly challenging scenario.

*Remark 1* (Completeness of Complementary Spectral Decomposition) Defining $\mathbf{h}_{\text{low}} = \mathbf{U}\, g(\mathbf{\Lambda})\, \mathbf{U}^\top\mathbf{h}$ and $\mathbf{h}_{\text{high}} = \mathbf{U}\,(\mathbf{I} - g(\mathbf{\Lambda}))\, \mathbf{U}^\top\mathbf{h}$, we have $\mathbf{h}_{\text{low}} + \mathbf{h}_{\text{high}} = \mathbf{h}$ since $\mathbf{U}$ is orthonormal, motivating the dual-frequency design. However, this requires $\mathcal{O}(N^3)$ eigendecomposition; our polynomial-based ADF-SF avoids this via independently parameterized Jacobi filters, with signal preservation ensured by the residual connection (Section 4.1).

**Proposition 1** (Universal Approximation of Jacobi Filters) *For any continuous spectral response $g : [-1, 1] \to \mathbb{R}$ and $\varepsilon > 0$, there exist polynomial order $K$ and coefficients $\{\theta_k\}_{k=0}^K$ such that $\sup_{\lambda\in[-1,1]} |g(\lambda) - \sum_{k=0}^K \theta_k P_k^{(\alpha,\beta)}(\lambda)| < \varepsilon$.*

This follows because Jacobi polynomials form a complete orthogonal basis for $L_w^2([-1, 1])$ Szegő (1975), and learnable $(\alpha, \beta)$ enable the model to adapt the basis to the target spectral response.

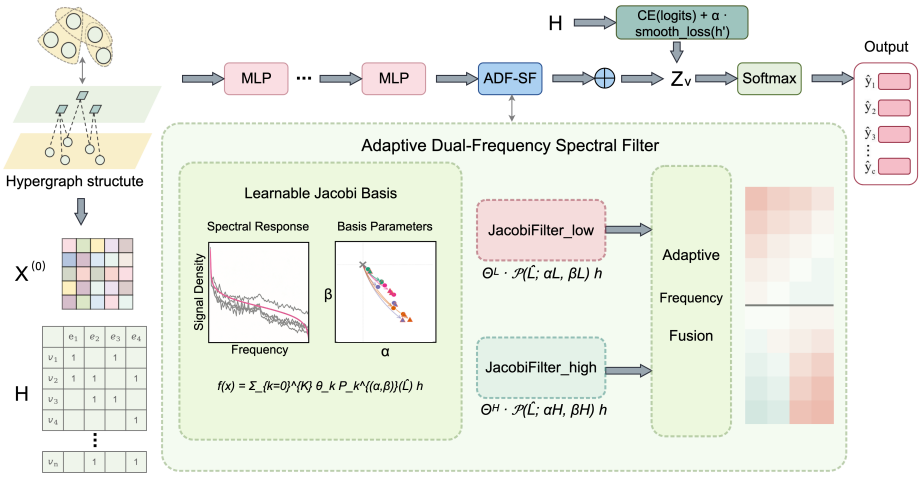



**Fig. 1:** Overview of SpectralMLP. An MLP backbone produces hidden representations that are refined by the Adaptive Dual-Frequency Spectral Filter (ADF-SF) before classification; ADF-SF combines low- and high-frequency Jacobi branches through a learned gate.

# 4 Method

## 4.1 Overview: SpectralMLP Framework

Figure 1 illustrates the overall architecture of SpectralMLP. The framework consists of three components: (1) an MLP backbone that transforms input features into hidden representations; (2) the proposed Adaptive Dual-Frequency Spectral Filter (ADF-SF) that performs spectral filtering without message passing; and (3) a classifier for node classification.

The processing pipeline is:

$$\mathbf{X} \xrightarrow{\text{MLP}} \mathbf{h} \xrightarrow{\text{ADF-SF}} \mathbf{h}' \xrightarrow{\text{Classifier}} \text{logits} \tag{3}$$

Structural simplicity is a defining property: the only graph operations are MLP layers and sparse matrix-vector multiplication with the fixed Laplacian, with no learnable aggregation operators. Frequency balance is handled adaptively by the gating mechanism, and the parameter overhead stays modest: a median of roughly 2.4% additional parameters at $\mathcal{O}(K \cdot \text{nnz} \cdot H)$ per-layer complexity.

***Training Objective.***

SpectralMLP is trained by combining cross-entropy loss and a structure-aware smoothness regulariser:

$$\mathcal{L} = \mathcal{L}_{\text{CE}}(\hat{y}, y) + \alpha \cdot \mathcal{L}_{\text{smooth}}(\mathbf{h}') \tag{4}$$

where $\mathcal{L}_{\text{CE}}$ is the standard cross-entropy loss on labeled nodes, and $\mathcal{L}_{\text{smooth}}$ penalizes the maximum pairwise Euclidean distance between representations of nodes within the same hyperedge (computed over a random sample of at most $S$ hyperedges for efficiency):

$$\mathcal{L}_{\text{smooth}} = \frac{1}{S} \sum_{e \in \mathcal{S}} \max_{u,v \in e} \|\mathbf{h}'_u - \mathbf{h}'_v\|_2 \tag{5}$$

where $\mathcal{S} \subseteq \mathcal{E}$ is a random subset of at most $S$ hyperedges. The hyperparameter $\alpha$ regulates the weight of smoothness regularization, encouraging nodes within the same hyperedge to have similar representations. This loss requires no additional message passing and is computed purely from the hidden representations.

***Complexity of $\mathcal{L}_{smooth}$.***

In practice, we set $S = 2000$, pad each sampled hyperedge to size $s_{\max} = 100$, and compute pairwise distances in batches via `torch.cdist`, yielding $O(S \cdot s_{\max}^2 \cdot H)$ complexity, negligible relative to the forward pass.

***MLP Backbone.***

The MLP backbone has the same structure as Hypergraph-MLP Yi et al. (2023) and is thus fair to compare. Each of its $L$ layers consists of linear transformation, LayerNorm Ba et al. (2016), ReLU activation, and dropout Srivastava et al. (2014), transforming input features $\mathbf{X} \in \mathbb{R}^{N \times F}$ into hidden representations $\mathbf{h} \in \mathbb{R}^{N \times H}$.

## 4.2 Adaptive Dual-Frequency Spectral Filter (ADF-SF)

As discussed in Sections 1 and 3, standard spectral filters apply a single response $g(\lambda)$ to all frequencies, which fails under spectral degeneration or frequency heterogeneity. ADF-SF instead uses independently parameterized low- and high-frequency branches, each with its own Jacobi polynomial basis $(\alpha, \beta)$ and coefficients $\theta$. Signal preservation is achieved through the residual connection $\mathbf{h}' = \mathbf{h} + \beta_{\text{res}} \cdot \mathbf{h}_{\text{sp}}$, and each branch can be independently optimised without constraining others.

### 4.2.1 Jacobi Polynomial Spectral Basis.

Jacobi polynomials $P_k^{(\alpha,\beta)}(x)$ are a type of orthogonal polynomials defined on $[-1, 1]$ with parameters $\alpha, \beta > -1$. They satisfy the three-term recurrence relation:

$$P_0^{(\alpha,\beta)}(x) = 1 \tag{6}$$

$$P_1^{(\alpha,\beta)}(x) = \frac{\alpha - \beta}{2} + \frac{\alpha + \beta + 2}{2} x \tag{7}$$

$$P_k^{(\alpha,\beta)}(x) = (a_k x + b_k) P_{k-1}^{(\alpha,\beta)}(x) - c_k P_{k-2}^{(\alpha,\beta)}(x) \tag{8}$$

where the recurrence coefficients $a_k$, $b_k$, $c_k$ depend on $\alpha$, $\beta$, and $k$ (see Wang and Zhang (2022); Szegő (1975) for closed-form expressions).

The Jacobi polynomial filter is:

$$\mathbf{h}_{\text{filtered}} = \sum_{k=0}^{K} \theta_k P_k^{(\alpha,\beta)}(\hat{\mathbf{L}})\mathbf{h} \tag{9}$$

where $\theta_k$ are learnable polynomial coefficients and $K$ is the polynomial order. Since $P_k^{(\alpha,\beta)}(\hat{\mathbf{L}})$ is computed via sparse matrix-vector products through the recurrence, eigendecomposition is not necessary.

Our key innovation is making $\alpha$ and $\beta$ learnable. When $\alpha = \beta = -0.5$, Jacobi polynomials reduce to Chebyshev polynomials. When $\alpha = \beta = 0$, they reduce to Legendre polynomials. Learnable $(\alpha, \beta)$ enable the model to discover the optimal polynomial basis for each dataset, adding only 2 parameters per filter.

To increase expressiveness, we use channel-wise polynomial coefficients $\theta_k^d \in \mathbb{R}^{(K+1)\times H}$. Each hidden dimension $d$ has its own polynomial coefficients, enabling different channels to learn distinct frequency responses.

### 4.2.2 Dual-Frequency Decomposition.

ADF-SF decomposes the hidden representation into independently parameterized low-frequency and high-frequency components:

**Low-pass branch:**

$$\mathbf{h}_{\text{low}} = \text{JacobiFilter}_{\text{low}}(\mathbf{h}, \hat{\mathbf{L}};\, \theta_{\text{low}}, \alpha_l, \beta_l) \tag{10}$$

**High-pass branch:**

$$\mathbf{h}_{\text{high}} = \text{JacobiFilter}_{\text{high}}(\mathbf{h}, \hat{\mathbf{L}};\, \theta_{\text{high}}, \alpha_h, \beta_h) \tag{11}$$

The two branches use independent parameters $(\theta_{\text{low}}, \alpha_l, \beta_l)$ and $(\theta_{\text{high}}, \alpha_h, \beta_h)$, allowing each to focus on different frequency components. By Proposition 1, each branch can approximate any continuous spectral response for large $K$. Unlike the complementary decomposition above (which constrains the high-pass branch to $1-g(\lambda)$), our design decouples the two branches. Each independently learns its own polynomial basis $(\alpha, \beta)$ and coefficients $\theta$ for good spectral specialization (see Figure 2).

### 4.2.3 Adaptive Gating and Fusion.

The dual-frequency components are fused with a learned gating mechanism:

$$\mathbf{g} = \sigma(\mathbf{W}_g[\mathbf{h}_{\text{low}}; \mathbf{h}_{\text{high}}] + \mathbf{b}_g) \in \mathbb{R}^{N\times 1} \tag{12}$$

$$\mathbf{h}_{\text{sp}} = \mathbf{g} \odot \mathbf{h}_{\text{low}} + (1 - \mathbf{g}) \odot \mathbf{h}_{\text{high}} \tag{13}$$

where $\sigma$ is the sigmoid function, $[\cdot;\cdot]$ is concatenation, and $\odot$ is element-wise multiplication. The gate $\mathbf{g} \in \mathbb{R}^{N\times 1}$ sets a single scalar weight per node, distributing low- and high-frequency contributions equally across all channels. A scalar gate is sufficient

**Algorithm 1** SpectralMLP Forward Pass

**Require:** Input features $\mathbf{X} \in \mathbb{R}^{N \times F}$, incidence matrix $\mathbf{H}$, polynomial order $K$
**Ensure:** Node classification logits

Compute scaled Laplacian: $\hat{\mathbf{L}} = \frac{2}{\lambda_{\max}}\mathbf{L} - \mathbf{I}$ ▷ Preprocessing
$\mathbf{h} \leftarrow \text{MLP}(\mathbf{X})$ ▷ MLP backbone
Compute $\mathbf{h}_{\text{low}} \leftarrow \sum_{k=0}^{K} \theta_k^l P_k^{(\alpha_l, \beta_l)}(\hat{\mathbf{L}})\,\mathbf{h}$ ▷ Low-pass Jacobi filter
Compute $\mathbf{h}_{\text{high}} \leftarrow \sum_{k=0}^{K} \theta_k^h P_k^{(\alpha_h, \beta_h)}(\hat{\mathbf{L}})\,\mathbf{h}$ ▷ High-pass Jacobi filter
$\mathbf{g} \leftarrow \sigma(\mathbf{W}_g[\mathbf{h}_{\text{low}}; \mathbf{h}_{\text{high}}] + \mathbf{b}_g)$ ▷ Adaptive gate
$\mathbf{h}_{\text{sp}} \leftarrow \mathbf{g} \odot \mathbf{h}_{\text{low}} + (1 - \mathbf{g}) \odot \mathbf{h}_{\text{high}}$ ▷ Dual-frequency fusion
$\mathbf{h}' \leftarrow \mathbf{h} + \beta_{\text{res}} \cdot \mathbf{h}_{\text{sp}}$ ▷ Residual connection
**return** Classifier($\mathbf{h}'$)

because the channel-wise polynomial coefficients $\theta_k^d$ already provide per-channel frequency specialization: the gate only needs to set a global low-vs-high ratio per node. A channel-wise gate ($O(H^2)$ parameters) would introduce redundancy with $\theta_k^d$ and be prone to overfitting on small datasets (e.g., Zoo with 101 nodes).

The gate can flexibly learn to balance frequency contributions: in spectral degeneration scenarios (*e.g.*Congress), it suppresses noisy high-frequency components; in multi-class settings (*e.g.*ModelNet40), it enhances class-discriminative high-frequency components.

#### 4.2.4 Residual Connection.

To ensure the stability of training and keep the original signal:

$$\mathbf{h}' = \mathbf{h} + \beta_{\text{res}} \cdot \mathbf{h}_{\text{sp}} \tag{14}$$

where $\beta_{\text{res}}$ is a learnable residual scale initialized to 0.1, and it regulates the size of the spectral filtering term.

#### 4.2.5 Filter Mode Configuration.

The ADF-SF module supports four filter modes: **Dual** (both branches with adaptive gate), **Low-pass** or **High-pass** (single branch), and **MLP Only** (no spectral filtering, only the smoothness regularizer). The dual-frequency mode is recommended as the default because the adaptive gate can learn to emulate any single-branch behavior (*e.g.*$g \to 0$ degenerates to pure high-pass). However, when a validation set is available for hyperparameter tuning, the filter mode can be used as an additional configuration option alongside polynomial order $K$ and learning rate. The search over five candidates has a very low cost. We report results under both the best-performing per-dataset configuration (Table 3) and a controlled ablation under uniform settings (Table 4).

### 4.3 Computational Complexity Analysis

The complexity of exact eigendecomposition is $\mathcal{O}(N^3)$ and it is too slow for large hypergraphs (*e.g.*DBLP with 41K nodes). Our Jacobi polynomial filter does not require

eigendecomposition and can be applied to the sparse Laplacian $\hat{\mathbf{L}}$ via the three-term recurrence, yielding $\mathcal{O}(K \cdot \text{nnz} \cdot H)$ per-layer complexity. Preprocessing (computing $\hat{\mathbf{L}}$ and $\lambda_{\max}$ via power iteration) costs $\mathcal{O}(\text{nnz})$ and only needs to be done once.

The additional parameters amount to $2(K+1)H+2H+6 \approx \mathcal{O}(KH)$, with a median overhead of about 2.4% across datasets (see Section 5.4 for per-dataset breakdown).

***Expressiveness of the Dual-Frequency Design.***

Two independently parameterized Jacobi filters of order $K$ are strictly more expressive than a single filter of order $2K$ in terms of *basis adaptability*. The dual design uses two independent sets $(\alpha_l, \beta_l)$ and $(\alpha_h, \beta_h)$, enabling each branch to specialize in a different spectral region. This yields a form of *piecewise basis adaptation* that a single globally defined polynomial basis cannot replicate, and explains why the dual-frequency design outperforms single-branch variants in our ablation (Section 5.3).

# 5 Experiments

## 5.1 Experimental Setup

### 5.1.1 Datasets.

We have selected 10 hypergraph node classification datasets of different scales, from small (Zoo, 101 nodes) to large (DBLP, 41K nodes), binary to 67 classes, and homophilic to heterophilic or degenerate spectral conditions. These include cocitation networks (Cora, Citeseer) Sen et al. (2008), a coauthorship network (DBLP) Yadati et al. (2019), a congressional network (Congress) Benson et al. (2016) exhibiting severe spectral degeneration ($\lambda_{\max} - \lambda_{\min} < 10^{-4}$), and six LE benchmarks Feng et al. (2019) (Zoo, Mushroom, 20News, ModelNet40, NTU2012, House). Table 2 summarizes the statistics.

**Table 2:** Dataset statistics. $N$: #nodes, $|\mathcal{E}|$: #hyperedges, $C$: #classes, $F$: #features, Avg.$|e|$: avg. hyperedge size, Homophily: node homophily.

| Dataset | $N$ | $\lvert\mathcal{E}\rvert$ | $F$ | $C$ | Avg.$\lvert e\rvert$ | Homophily |
|---|---|---|---|---|---|---|
| Cora | 2,708 | 1,579 | 1,433 | 7 | 3.03 | 0.86 |
| Citeseer | 3,312 | 1,079 | 3,703 | 6 | 3.20 | 0.83 |
| DBLP | 41,302 | 22,363 | 1,425 | 6 | 4.45 | 0.87 |
| Zoo | 101 | 43 | 16 | 7 | 39.93 | 0.66 |
| 20News | 16,242 | 100 | 100 | 4 | 654.51 | 0.49 |
| Mushroom | 8,124 | 298 | 22 | 2 | 136.31 | 0.95 |
| NTU2012 | 2,012 | 2,012 | 100 | 67 | 5.00 | 0.86 |
| ModelNet40 | 12,311 | 12,311 | 100 | 40 | 5.00 | 0.92 |
| House | 1,290 | 341 | 100 | 2 | 34.79 | 0.58 |
| Congress | 1,718 | 83,105 | 100 | 2 | 8.66 | 0.56 |

### 5.1.2 Baselines.

We compare with 8 representative methods and a vanilla MLP baseline (9 baselines in total):

- **Message-passing HGNNs:** AllSet Chien et al. (2022), HGNN Feng et al. (2019), UniGCNII Huang and Yang (2021)
- **Transformer-based:** HyperGT Liu et al. (2024), CEGAT Veličković et al. (2018a)
- **Diffusion-based:** ED-HNN Wang et al. (2023)
- **Sheaf-based:** SheafHyperGCN Duta et al. (2023)
- **Message-passing-free (MLP-based):** Hypergraph-MLP Yi et al. (2023) and vanilla MLP (without hypergraph structure)

### 5.1.3 Implementation Details.

Following Hypergraph-MLP Yi et al. (2023), we randomly partition each dataset into 50%/25%/25% training/validation/test splits. All experiments are carried out on an NVIDIA RTX 4080 SUPER using PyTorch. We use the Adam optimizer Kingma and Ba (2015) with dataset-specific learning rates ($10^{-3}$, $10^{-4}$, $10^{-2}$) and weight decay (0 or $5 \times 10^{-4}$), early stopping with patience 100 on validation accuracy, polynomial order $K \in \{10, 15, 20\}$ per dataset, and filter mode $\in$ {Dual, Low-pass, High-pass, MLP Only}. The smoothness regulariser weight $\alpha$ is selected via grid search on the validation set. Results are presented as mean $\pm$ standard deviation after 20 independent repetitions.

## 5.2 Main Results

Table 3 presents the node classification accuracy of SpectralMLP against all 9 baselines across 10 datasets. SpectralMLP results use the best-performing dataset-specific configuration (see Section 5.3 for a controlled ablation under uniform settings).

The Congress dataset provides the sharpest test: despite eigenvalue clustering below $10^{-4}$, SpectralMLP reaches 95.88%, ahead of AllSet (95.26%) and HyperGT (95.23%), while Hypergraph-MLP collapses to 70.47%, a 25-point gap confirming the value of dual-frequency filtering under degeneration. SpectralMLP also tops ModelNet40 (98.08%) and NTU2012 (89.34%), two multi-class benchmarks where the high-frequency branch helps separate fine-grained categories. On the remaining datasets, SpectralMLP remains competitive but does not always lead (ED-HNN and SheafHyperGCN outperform it on Cora, ED-HNN on DBLP, HyperGT on House), trading some peak performance on well-behaved graphs for robustness under challenging spectral conditions. Across all 10 datasets, SpectralMLP ranks first or second on 7, at a median parameter overhead of about 2.4%.

***On the role of dual-frequency filtering.***

The main-table results use the best-performing filter mode per dataset. Single-branch modes lead on four datasets (DBLP, House: high-pass; 20News: low-pass; Mushroom: MLP Only). Dual-frequency, however, is the *only* variant that excels on the most challenging case, Congress, surpassing both single branches by over 6.2%. It also remains

**Table 3:** Node classification accuracy (%) on 10 hypergraph datasets. **Bold**: best, underline: second best. OOM: out-of-memory. Mean ± std over 20 runs.

| Method | Cora | Citeseer | DBLP | Zoo | 20News |
|---|---|---|---|---|---|
| AllSet | 79.53±1.62 | 72.48±1.69 | 91.44±0.21 | 89.81±9.53 | 81.43±0.49 |
| MLP | 67.50±1.53 | 68.81±1.65 | 83.32±0.29 | 88.27±7.84 | 78.88±0.61 |
| CEGAT | 76.02±1.56 | 71.10±1.54 | 88.16±0.31 | 44.62±12.81 | OOM |
| HGNN | 79.37±1.36 | 72.53±1.87 | 90.94±0.30 | 89.62±6.61 | 80.80±0.41 |
| UniGCNII | 79.34±1.31 | 71.67±1.86 | 91.71±0.24 | 86.73±7.84 | 80.89±0.46 |
| Hypergraph-MLP | 79.45±1.33 | **73.72**±1.72 | 90.29±0.26 | 90.58±6.89 | 81.46±0.45 |
| HyperGT | 77.10±1.18 | 70.05±1.41 | 90.34±0.25 | 90.00±6.99 | 81.08±0.33 |
| SheafHyperGCN | 80.06±1.12 | 73.27±0.50 | 90.83±0.23 | 91.15±8.29 | OOM |
| ED-HNN | **80.31**±1.35 | 73.70±1.38 | **91.90**±0.19 | 90.36±1.61 | 80.19±1.43 |
| **SpectralMLP** | 79.85±1.51 | **73.72**±1.57 | 90.88±0.23 | **91.73**±7.72 | **81.56**±0.51 |
| Method | Mushroom | NTU2012 | ModelNet40 | House | Congress |
| AllSet | **99.95**±0.07 | 88.39±1.26 | 97.97±0.27 | 66.73±2.09 | 95.26±0.84 |
| MLP | 89.71±8.77 | 83.02±1.78 | 94.87±0.36 | 67.24±4.03 | 68.00±2.86 |
| CEGAT | 96.60±1.67 | 81.81±2.09 | 92.05±0.42 | 68.93±3.64 | 72.30±5.87 |
| HGNN | 97.86±0.39 | 87.64±1.60 | 95.34±0.29 | 59.88±3.50 | 90.83±1.36 |
| UniGCNII | 99.81±0.12 | 87.42±1.58 | 96.88±0.20 | 68.73±2.55 | 94.81±0.81 |
| Hypergraph-MLP | 99.94±0.11 | 88.42±1.32 | 95.16±0.49 | 69.78±2.75 | 70.47±3.41 |
| HyperGT | 99.92±0.06 | 80.95±2.46 | 96.89±0.26 | **73.56**±0.47 | 95.23±0.73 |
| SheafHyperGCN | **99.95**±0.08 | 89.20±1.42 | 97.93±0.29 | 72.66±2.26 | 90.37±1.52 |
| ED-HNN | 99.16±1.62 | 79.91±1.69 | 97.21±0.34 | 68.87±2.98 | 95.00±0.99 |
| **SpectralMLP** | 99.94±0.09 | **89.34**±1.72 | **98.08**±0.19 | 70.29±2.40 | **95.88**±1.03 |

within 1% of the best variant on *all* 10 datasets, making it a robust zero-configuration default.

## 5.3 Ablation Study

Table 4 presents an ablation study comparing different filter variants of SpectralMLP across all 10 datasets under uniform settings. The main-table results (Table 3) use dataset-specific configurations and may differ slightly.

All SpectralMLP variants outperform the plain MLP baseline, with gains from +2.7% to +27.9%. Dual-frequency is best on 6 datasets (Cora, Citeseer, Zoo, NTU2012, ModelNet40, Congress), high-pass on 2 (DBLP, House), and single-branch variants on the remaining 2, confirming that different datasets benefit from different frequency strategies. The dual-frequency advantage is most pronounced on Congress (95.88% vs. 89.60%/89.12% for single branches). Even where Dual is not the top variant, it stays within 1% of the best across all 10 datasets, confirming effective branch interpolation by the adaptive gate.

***Why does dual-frequency filtering help under spectral degeneration?***

When eigenvalues cluster, both branches degenerate to approximately constant responses, yet Dual still achieves 95.88% on Congress, far above both single branches (∼89%). We attribute this to three mechanisms: channel-wise polynomial coefficients $\theta_k^d$ that allow per-dimension adaptation; the residual connection that safeguards the

**Table 4:** Ablation on filter modes (Jacobi polynomial variants): accuracy (%) on 10 hypergraph datasets. Best in **bold**. Mean ± std over 20 runs. Cite.=Citeseer, 20New=20News, Mush.=Mushroom, NTU=NTU2012, MN40=ModelNet40, Cong.=Congress.

| Method | Cora | Cite. | DBLP | Zoo | 20New |
|---|---|---|---|---|---|
| MLP (raw) | 67.50±1.53 | 68.81±1.65 | 83.32±0.29 | 88.27±7.84 | 78.88±0.61 |
| MLP Only | 79.59±1.54 | 72.83±1.47 | 90.19±0.08 | 90.58±8.51 | 81.48±0.51 |
| Low-pass | 79.82±2.03 | 73.30±1.55 | 90.86±0.06 | 91.58±8.95 | **81.56**±0.51 |
| High-pass | 79.38±1.44 | 72.52±1.35 | **90.88**±0.23 | 91.35±7.37 | 81.55±0.48 |
| Dual (ours) | **79.85**±1.51 | **73.72**±1.57 | 90.86±0.11 | **91.73**±7.92 | 81.53±0.50 |
| Method | Mush. | NTU | MN40 | House | Cong. |
| MLP (raw) | 89.71±8.77 | 83.02±1.78 | 94.87±0.36 | 67.24±4.03 | 68.00±2.86 |
| MLP Only | **99.94**±0.09 | 87.30±1.49 | 94.88±0.42 | 70.11±2.25 | 68.37±1.88 |
| Low-pass | 99.93±0.11 | 87.18±1.72 | 97.98±0.22 | 70.09±1.91 | 89.60±0.96 |
| High-pass | 99.90±0.08 | 88.87±1.87 | 97.14±0.27 | **70.29**±2.40 | 89.12±1.47 |
| Dual (ours) | 99.93±0.06 | **89.34**±1.72 | **98.08**±0.19 | 70.28±2.10 | **95.88**±1.03 |

MLP backbone; and the gate that learns to treat the spectral component conservatively, functioning as a regularizer rather than a primary feature extractor, which is consistent with the 27.5-point gap between Dual (95.88%) and MLP Only (68.37%).

### 5.4 Parameter Efficiency

A key advantage of spectral filtering is parameter efficiency. Table 5 compares parameter counts across 10 datasets. SpectralMLP has the second-lowest median count (393K), surpassed only by the shallow HGNN (323K), yet uses far fewer parameters than AllSet (1,446K), HyperGT (1,877K), and SheafHyperGCN (1,160K), confirming that spectral filtering provides an efficient alternative to learnable aggregation operators, attention projections, and diffusion/sheaf parameters.

Internally, ADF-SF adds a median of about 2.4% overhead over the MLP-only backbone (below 2.5% on six datasets; the higher relative ratios on Zoo, NTU2012, and ModelNet40 reflect their small base models).

### 5.5 Visualization and Analysis

Figures 2 and 3 confirm that SpectralMLP adapts its frequency behavior to each dataset. On Congress, it achieves the sharpest binary separation, consistent with 95.88% accuracy; on Citeseer, NTU2012, and ModelNet40, it produces tighter intra-class clusters with clearer inter-class boundaries, confirming that dual-frequency filtering benefits a broad range of tasks beyond spectral degeneration.

## 6 Conclusion

SpectralMLP demonstrates that spectral filtering alone, without learnable aggregation operators, suffices for hypergraph node classification. By replacing message passing with dual-frequency Jacobi polynomial filtering whose only graph operation is sparse matrix

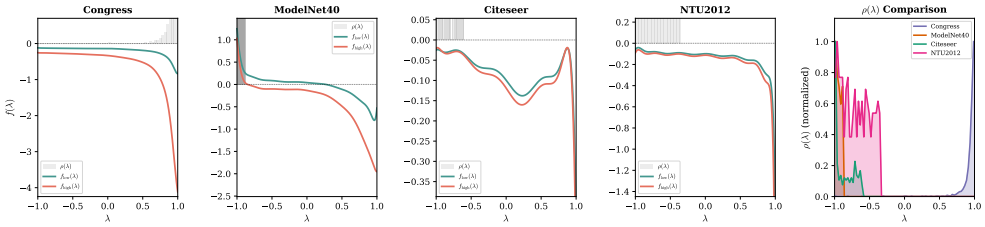



**Fig. 2:** Learned spectral filter responses $f_{\text{low}}(\lambda)$ (blue) and $f_{\text{high}}(\lambda)$ (red) overlaid on eigenvalue density $\rho(\lambda)$ (gray), shown for four datasets plus a cross-dataset density comparison. On Congress (spectral degeneration) the model strongly suppresses high-frequency components, while on multi-class datasets both branches remain active.

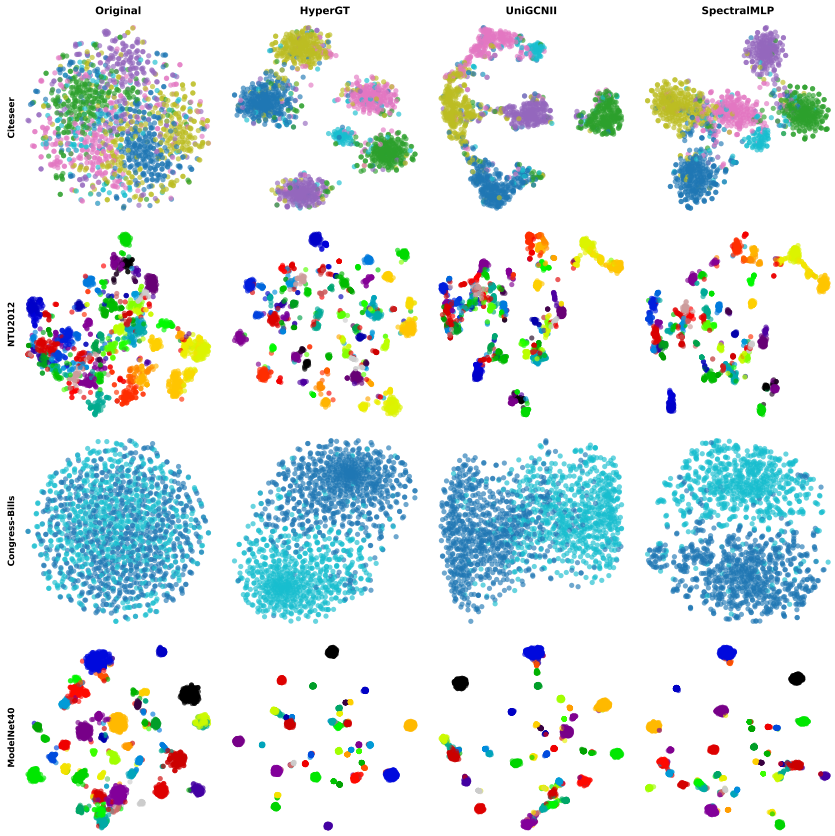



**Fig. 3:** t-SNE visualization of learned node embeddings on four datasets. Each row is one dataset; columns show Original features, HyperGT, UniGCNII, and SpectralMLP. Points are colored by ground-truth class. SpectralMLP produces tighter intra-class clusters with clearer inter-class boundaries.

**Table 5:** Parameter count comparison across models on 10 datasets. Cite.=Citeseer, 20New=20News, Mush.=Mushroom, NTU=NTU2012, MN40=ModelNet40, Cong.=Congress, Med.=Median.

| | Cora | Cite. | DBLP | Zoo | 20New |
|---|---|---|---|---|---|
| AllSet | 1,134,343 | 5,379,078 | 3,046,406 | 28,552 | 451,076 |
| CEGAT | 739,107 | 3,801,630 | 1,468,958 | 56,360 | 110,100 |
| HGNN | 369,166 | 1,900,044 | 733,708 | 1,680 | 27,144 |
| UniGCNII | 499,975 | 2,423,814 | 1,257,478 | 9,800 | 578,052 |
| HyperGT | 400,462 | 552,332 | 8,465,676 | 44,688 | 1,086,472 |
| ED-HNN | 434,951 | 1,015,814 | 212,326 | 19,848 | 521,604 |
| SheafHGCN | 2,359,507 | 5,835,528 | 2,336,520 | 15,814 | 243,462 |
| **SpectralMLP** | 504,611 | 1,905,698 | 1,266,742 | 9,251 | 29,994 |
| MLP Only | 501,357 | 1,899,372 | 1,244,208 | 6,643 | 29,290 |
| Overhead | 0.65% | 0.33% | 1.81% | 39.26% | 2.40% |
| | Mush. | NTU | MN40 | House | Cong. |
| AllSet | 107,784 | 467,267 | 1,707,561 | 1,688,067 | 450,819 |
| CEGAT | 4,328 | 22,159 | 148,173 | 109,071 | 54,543 |
| HGNN | 4,112 | 43,398 | 73,298 | 53,766 | 26,886 |
| UniGCNII | 10,184 | 19,011 | 597,033 | 577,539 | 157,699 |
| HyperGT | 574,864 | 302,278 | 1,618,834 | 144,902 | 5,577,606 |
| ED-HNN | 502,152 | 529,731 | 526,377 | 488,707 | 488,707 |
| SheafHGCN | 51,078 | 121,222 | 45,766 | 296,712 | 296,712 |
| **SpectralMLP** | 8,980 | 111,957 | 6,098 | 58,910 | 29,490 |
| MLP Only | 270 | 99,663 | 4,773 | 58,627 | 28,786 |
| Overhead | – | 12.33% | 27.76% | 0.48% | 2.44% |

Median parameter count: SpectralMLP **393,173**, MLP Only 387,299, HGNN 323,320, UniGCNII 613,059, ED-HNN 474,022, CEGAT 651,443, SheafHGCN 1,160,232, AllSet 1,446,095, HyperGT 1,876,811. Overhead median: 2.4%.

multiplication with the fixed Laplacian, the framework achieves competitive or superior accuracy to message-passing methods while using fewer parameters and avoiding over-squashing, $\mathcal{O}(N^2)$ attention, and $\mathcal{O}(N^3)$ eigendecomposition. The *filtering-versus-aggregation* perspective carries practical weight: spectral filtering provides explicit frequency control that aggregation lacks, and the dual-frequency architecture demonstrates that two lightweight branches specializing independently in different frequency regimes outperform a single filter, with the adaptive gate automatically discovering the right balance.

Experiments on 10 datasets against 9 baselines confirm these findings. SpectralMLP ranks first or second on 7 datasets with about 2.4% additional parameters over the MLP backbone. The dual-frequency mode serves as a robust zero-configuration default (within 1% of the best variant on all datasets), with single-branch modes as task-specific options. The strongest result, +27.9% on Congress under uniform settings, shows the value of dual-frequency filtering for spectrally degenerate hypergraphs.

***Limitations and Future Work.***

The ADF-SF module could benefit from adaptive polynomial order $K$ (determined through hypergradient optimization or eigenvalue-spread heuristics) and from a special gating mechanism to model per-channel homophily/heterophily patterns. The MLP backbone also sets an upper bound on the overall quality; high-capacity residual connections or attention-augmented MLPs can increase performance further. Spectral filters that operate directly on the hypergraph incidence structure (rather than through clique expansion) would be more conceptually intuitive, but it is currently unknown whether polynomial recurrence relations can be extended to this setting.

## Declarations


- **Funding:** The authors did not receive support from any organization for the submitted work.
- **Conflict of interest:** The authors have no competing interests to declare that are relevant to the content of this article.
- **Data availability:** The datasets used in this study are publicly available from their respective sources cited in the references.
- **Code availability:** The code will be made available upon publication.
- **Author contribution:** All authors contributed to the study conception and design. All authors read and approved the final manuscript.


## References


Ai, G., Pang, G., Qiao, H., Gao, Y., Yan, H.: Grokformer: Graph fourier kolmogorov-arnold transformers. In: Proceedings of the 42nd International Conference on Machine Learning (ICML) (2025). https://doi.org/10.48550/arXiv.2411.17296

Alon, U., Yahav, E.: On the bottleneck of graph neural networks and its practical implications. arXiv preprint arXiv:2006.05205 (2020) https://doi.org/10.48550/arXiv.2006.05205 . Published at ICLR 2021

Benson, A.R., Gleich, D.F., Leskovec, J.: Higher-order organization of complex networks. Science **353**(6295), 163–166 (2016) https://doi.org/10.1126/science.aad9029

Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016) https://doi.org/10.48550/arXiv.1607.06450

Bo, D., Wang, X., Liu, Y., Fang, Y., Li, Y., Shi, C.: A survey on spectral graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. (2023) https://doi.org/10.48550/arXiv.2302.05631

Bruna, J., Zaremba, W., LeCun, Y.: Spectral networks and locally connected networks on graphs. In: Proceedings of the International Conference on Learning Representations (ICLR) (2013). https://doi.org/10.48550/arXiv.1312.6203

Chung, F.R.: Spectral Graph Theory. American Mathematical Society, ??? (1997). https://doi.org/10.1090/cbms/092

Chien, E., Peng, J., Li, P., Milenkovic, O.: Adaptive universal generalized pagerank graph neural network. In: International Conference on Learning Representations (ICLR) (2021). https://doi.org/10.48550/arXiv.2006.07988

Chien, E., Pan, C., Peng, J., Milenkovic, O.: You are allset: A multiset function framework for hypergraph neural networks. In: Proceedings of the International Conference on Learning Representations (ICLR) (2022). https://doi.org/10.48550/arXiv.2106.13264

Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in Neural Information Processing Systems (NeurIPS), pp. 3844–3852 (2016). https://doi.org/10.48550/arXiv.1606.09375

Duta, I., Cassarà, G., Silvestri, F., Liò, P.: Sheaf hypergraph networks. In: Advances in Neural Information Processing Systems (NeurIPS) (2023). https://doi.org/10.48550/arXiv.2309.17116

Dong, Y., Sawin, W., Bengio, Y.: Hnhn: Hypergraph networks with hyperedge neurons. arXiv preprint arXiv:2006.12278 (2020) https://doi.org/10.48550/arXiv.2006.12278

Feng, Y., You, H., Zhang, Z., Ji, R., Gao, Y.: Hypergraph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 3558–3565 (2019). https://doi.org/10.1609/aaai.v33i01.33013558

Huang, C., Sun, J., Zha, L., Wang, Y., Huang, X.: Beyond homophily: Adaptive cross-frequency convolution for hypergraph learning. Knowl.-Based Syst. **335**, 115064 (2026) https://doi.org/10.1016/j.knosys.2025.115064

He, M., Wei, Z., Huang, Z., Xu, H.: Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 34, pp. 17429–17440 (2021). https://doi.org/10.48550/arXiv.2106.10994

He, M., Wei, Z., Wen, J.-R.: Convolutional neural networks on graphs with chebyshev approximation, revisited. In: Advances in Neural Information Processing Systems (NeurIPS) (2022). https://doi.org/10.48550/arXiv.2202.03580

Huang, J., Yang, J.: Unignn: A unified framework for graph and hypergraph neural networks. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pp. 2391–2397 (2021). https://doi.org/10.24963/ijcai.2021/353

Hu, Y., You, H., Wang, Z., Wang, Z., Zhou, E., Gao, Y.: Graph-mlp: Node classification without message passing in graph. arXiv preprint arXiv:2106.04051 (2021) https:

//doi.org/10.48550/arXiv.2106.04051

Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the International Conference on Learning Representations (ICLR) (2015). https://doi.org/10.48550/arXiv.1412.6980

Kim, S., Lee, S.Y., Gao, Y., Antelmi, A., Polato, M., Shin, K.: A survey on hypergraph neural networks: An in-depth and step-by-step guide. In: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) (2024). https://doi.org/10.48550/arXiv.2404.01039

Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: Proceedings of the International Conference on Learning Representations (ICLR) (2017). https://doi.org/10.48550/arXiv.1609.02907

Liu, Z., Tang, B., Ye, Z., Dong, X., Chen, S., Wang, Y.: Hypergraph transformer for semi-supervised classification. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2024). https://doi.org/10.1109/ICASSP48485.2024.10446248

Li, M., Zhao, R., Yan, Z., Bai, L., Cui, L., Cao, F.: Hyperaim: Hypergraph contrastive learning with adaptive multi-frequency filters. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 40, pp. 23063–23070 (2026). https://doi.org/10.1609/aaai.v40i27.39472

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(56), 1929–1958 (2014) https://doi.org/10.5555/2627435.2670313

Sen, P., Namata, G., Bilgić, M., Getoor, L., Gallagher, B., Eliassi-Rad, T.: Collective classification in network data. AI Mag. **29**(3), 93–106 (2008) https://doi.org/10.1609/aimag.v29i3.2157

Szegő, G.: Orthogonal Polynomials, 4th edn. American Mathematical Society, Providence, RI (1975). https://doi.org/10.1090/coll/023

Tang, B., Liu, Z., Jiang, K., Chen, S., Dong, X.: Training-free message passing for learning on hypergraphs. In: International Conference on Learning Representations (ICLR) (2025). https://doi.org/10.48550/arXiv.2312.09778

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: Proceedings of the International Conference on Learning Representations (ICLR) (2018). https://doi.org/10.48550/arXiv.1710.10903

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: Proceedings of the International Conference on Learning Representations (ICLR) (2018). https://doi.org/10.48550/arXiv.1710.10903

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (NeurIPS) (2017). https://doi.org/10.48550/arXiv.1706.03762

Wang, P., Yang, S., Liu, Y., Wang, Z., Li, P.: Equivariant hypergraph diffusion neural operators. In: International Conference on Learning Representations (ICLR) (2023). https://doi.org/10.48550/arXiv.2207.06680

Wang, X., Zhang, M.: How powerful are spectral graph neural networks. In: International Conference on Machine Learning (ICML), pp. 23610–23622 (2022). https://doi.org/10.48550/arXiv.2205.11172

Yi, J., Li, J., Chen, M.: Hypergraph-mlp: A simple yet effective hypergraph neural network. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pp. 2401–2407 (2023)

Yadati, N., Nimishakavi, M., Yadav, P., Nitin, V., Louis, M., Talukdar, P.: Hypergcn: Hypergraph convolutional networks. In: Advances in Neural Information Processing Systems (NeurIPS) (2019). https://doi.org/10.48550/arXiv.1809.02589

Yang, M., Xu, X.-J.: Recent advances in hypergraph neural networks. arXiv preprint arXiv:2503.07959 (2025) https://doi.org/10.48550/arXiv.2503.07959

Zhou, D., Huang, J., Schölkopf, B.: Learning with hypergraphs: Clustering, classification, and embedding. In: Advances in Neural Information Processing Systems (NeurIPS) (2006). https://doi.org/10.7551/mitpress/7503.003.0205

Zhu, M., Wang, X., Zhang, C.S., *et al.*: Interpreting and unifying graph neural networks with an optimization perspective. In: Proceedings of the Web Conference (WWW) (2021). https://doi.org/10.1145/3442381.3449953