SPECVALLEY: AGGREGATION AS SPECTRAL VALLEY IN DENSE HETEROPHILIC GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Node classification in heterophilic graphs remains an open problem: aggregating over neighbourhoods can damp precisely the part of the signal that carries the labels. The field has answered with ever more expressive propagation operators, leaving the aggregation step that precedes them fixed by degree normalisation. However, degree normalisation acts as a spectral valley in its own right, attenuating a fixed middle band whose position at $\lambda = 1$ is set by the normalisation rather than by the data, exactly where the eigenvalues of dense graphs accumulate. We therefore begin with the representations rather than the propagation, in two stages: a learnable aggregation followed by an ordinary spectral filter. The first, SpecValley learns the position, sharpness and flank gains of the valley for every graph. We show formally how the aggregation operator and downstream filter interact. We establish that the response $1 - \lambda$ of degree normalisation vanishes exactly at that accumulation point. A spectral energy budget shows that what aggregation removes from a band, no linear spectral filter can recover. Crowding alone says nothing about the benefit. That is decided by how much label energy the valley removes, which we measure for every graph. SpecValley achieves strong results across heterophilic benchmarks, and a controlled ablation over aggregation operators and downstream filters attributes the gain to aggregation rather than the filter.

1 INTRODUCTION

Graphs are a fundamental data structure for relational information: nodes stand for entities, edges for the relations between them. Node classification is a central task, with applications in social networks, biological systems and recommendation settings (Kipf and Welling, 2017; Veličković et al., 2018). Graph neural networks (GNNs) shape this field by propagating and aggregating information across local neighbourhoods.

Most GNNs assume homophily, that is, that neighbouring nodes share similar features or classes (Zhu et al., 2020; Pei et al., 2020). In heterophilic graphs this does not hold (Yan et al., 2022): repeated aggregation amplifies mismatched signals and reduces class separability, particularly in dense graphs (Oono and Suzuki, 2020). The field has answered with feature-driven models (Abu-El-Haija et al., 2019; Lim et al., 2021), contrastive learning (Veličković et al., 2019; Hassani and Khasahmadi, 2020; Qiu et al., 2020), structure learning (Franceschi et al., 2019; Wu et al., 2022) and increasingly elaborate spectral and transformer architectures (Chien et al., 2021; Lin et al., 2024; Koke and Cremers, 2024; Deng et al., 2024), each tied to a regime: transformers require large datasets, spectral models excel on small or dense heterophilic graphs. All of them design the convolution operator or the structure of the graph and leave the step before it untouched: the filter is learned, the material it operates on is not.

That step is not neutral. Degree normalisation acts spectrally as a fixed valley with its zero at $\lambda = 1$, a position set by the normalisation rather than by the data, and it is exactly there that the eigenvalues of dense graphs accumulate. Whether this helps depends not on the crowding itself, which every dense graph shows, but on how much label energy lies in the attenuated band. We therefore treat aggregation as a stage of its own, upstream of an ordinary spectral filter and make it learnable: SpecValley learns the position, sharpness and flank gains of the valley per graph, staying close to degree normalisation where the band carries little label energy and moving its centre away from $\lambda = 1$ where it carries much.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

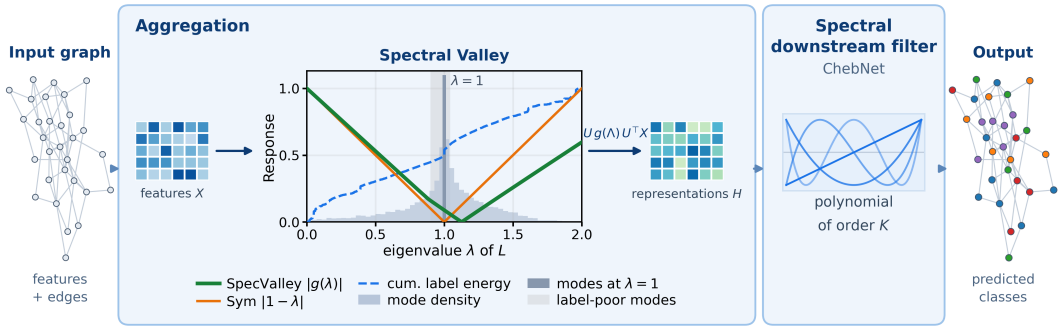

Figure 1: Overview of SpecValley. Top: node features first pass an aggregation stage and then an ordinary spectral filter (ChebNet, Eq. (5)*), which predicts the node labels. Bottom: A more detailed visualization of the aggregation stage. On dense graphs the eigenvalues of the normalised Laplacian accumulate at* $\lambda = 1$ *(grey histogram) and degree normalisation acts there as the fixed response* $1 - \lambda$*, a valley whose zero is pinned by the normalisation rather than by the data (Proposition 4.1). SpecValley replaces it by a response whose centre, sharpness and flank gains are learned per graph (Eq.* (3)*), so that the attenuated band can be placed away from the modes that carry the label energy.*

Contributions. We contribute a learnable aggregation stage that generalises degree normalisation and fixes the spectral energy budget available to every downstream filter. A formal account of the interaction shows that the response $1 - \lambda$ vanishes at the accumulation point of dense spectra, and no linear spectral filter can recover what aggregation has removed from a band. We perform a controlled study on stored splits, crossing five aggregators with four filters on six graphs, together with the label-energy spectrum of each graph and the response SpecValley learns on it. SpecValley achieves strong results throughout. The ablation attributes the gain to the aggregation stage rather than to the filter and learned attention adds little beyond a fixed aggregation structure once a spectral filter follows it (Section 5.2).

2 Related Work

Standard neighbourhood aggregation amplifies mismatched signals in heterophilic graphs, causing oversmoothing and reduced class separability (Li et al., 2018; Yan et al., 2022; Oono and Suzuki, 2020). Feature-driven methods emphasise multi-hop interactions or decouple features from topology (Abu-El-Haija et al., 2019; Lim et al., 2021), structure-oriented methods reweight or denoise edges (Franceschi et al., 2019; Wu et al., 2022). Both operate on fixed representations. Spectral methods such as GPR-GNN (Chien et al., 2021) and FAGCN (Bo et al., 2021) learn adaptive frequency weights, BernNet (He et al., 2021) approximates arbitrary filters, Specformer (Bo et al., 2023) learns the filter with a transformer over the eigenvalues, JacobiConv (Wang and Zhang, 2022) uses Jacobi polynomial bases, NLSF (Lin et al., 2024) introduces equivariant nonlinear filtering. These design expressive operators but apply them to a frequency profile that the preceding aggregation has already fixed. Few works address that step itself, and they do so spatially: separating ego from neighbourhood information (Zhu et al., 2020; Platonov et al., 2023), projecting along and against the edge direction (Rossi et al., 2023) or learning edge weights by attention (Veličković et al., 2018). Read spectrally, ego-neighbour separation yields the affine family $a + b\,(1 - \lambda)$, so the position of the response is learnable but its shape is not: sharpness and asymmetry remain those of degree normalisation. Attention weights are not a function of L, and our ablation measures how they compare with degree normalisation (Section 5.2). Graph transformers inject global structure at the feature level via Laplacian or random-walk encodings (Dwivedi et al., 2022; Rampášek et al., 2022; Kreuzer et al., 2021) and Polynormer (Deng et al., 2024) combines polynomial expressivity with local and global attention.

Our Positioning. We treat the aggregation stage itself as a spectral operator and make it learnable, with degree normalisation as the special case of a fixed valley whose position, sharpness and flanks we instead learn. Since aggregation fixes how much spectral energy reaches the downstream filters at all, our approach does not replace filter design but precedes it.

3 SPECTRAL REPRESENTATIONS

3.1 PROBLEM FORMULATION

Let $G = (V, E)$ be a graph on n nodes with features $x_i \in \mathbb{R}^d$, collected in $X \in \mathbb{R}^{n\times d}$, and adjacency $A \in \mathbb{R}^{n\times n}$. We work in the transductive setting, where A is fixed and the task is to learn node representations $Z \in \mathbb{R}^{n\times d'}$ for (semi-)supervised node classification. Message-passing GNNs obtain them by repeatedly aggregating each node's neighbourhood and combining the result with the node's own state (Gilmer et al., 2017). This paper concerns the aggregation operator, not the propagation that follows.

3.2 LINEAR AGGREGATORS WITH AND WITHOUT PARAMETERS

In heterophilic graphs adjacent nodes are often only weakly aligned semantically, which makes uniform aggregation ineffective (Dai et al., 2021). We therefore view the encoder as an aggregation operator $P \in \mathbb{R}^{n\times n}$ supported on E that refines node representations before graph convolution, $H' = P\,V$, with learnable value projection $V = XW_V$.

Parameter-free aggregators. As references we use two degree-based operators, the symmetric normalisation of GCN (Kipf and Welling, 2017; Wu et al., 2019) and mean aggregation as in GraphSAGE (Hamilton et al., 2017),

$$P_{\text{sym}} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}\colon\ h'_i = \textstyle\sum_{j\in\mathcal{N}(i)} \frac{V_j}{\sqrt{d_i d_j}},\quad P_{\text{mean}} = D^{-1}A\colon\ h'_i = \frac{1}{d_i}\sum_{j\in\mathcal{N}(i)} V_j. \tag{1}$$

Both are linear and fixed by the graph; only W_V is learned. For undirected A, $P_{\text{sym}} = I - L$ shares the eigenbasis of the normalised Laplacian, whereas the row-stochastic $P_{\text{mean}} = D^{-1/2}P_{\text{sym}}D^{1/2}$ is similar to it but not symmetric (Chung, 1997).

Learned aggregation via self-attention. Instead of fixing edge weights by degree, we learn them from the features via scaled dot-product self-attention, which has shown advantages over uniform aggregation (Veličković et al., 2018; Zhao et al., 2021). With query (Q), key (K), and value (V) projections of the node features and hidden dimensionality F_{hidden},

$$e_{ij} = \frac{Q_i^\top K_j}{\sqrt{F_{\text{hidden}}}} \qquad \alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k\in\mathcal{N}(i)}\exp(e_{ik})} \qquad h'_i = \sum_{j\in\mathcal{N}(i)} \alpha_{ij}V_j \tag{2}$$

Here, e_{ij} measures feature compatibility and α_{ij} reweights the neighbours in $\mathcal{N}(i)$, so that h'_i is the input to the subsequent spectral layer. In matrix form, $H' = A(X)\,V$ with $A(X)_{ij} = \alpha_{ij}$: attention keeps the support and, like P_{mean}, the row-stochasticity of Eq. (1), but its weights depend on the features, so $A(X)$ is not linear in X. With frozen query and key projections (*attn*$_{\text{init}}$, Section 5.2), $A(X)$ is a fixed matrix on the transductive graph and acts linearly on V. Although Eq. (2) is a local 1-hop message-passing step, it differs from Graph Attention Networks (GATs) (Veličković et al., 2018) in being applied once as an input encoder rather than interleaved with multi-hop propagation.

3.3 ADAPTIVE SPECTRAL AGGREGATION

The fixed aggregators of Eq. (1) act spectrally through the response $1 - \lambda$, a valley at $\lambda = 1$ whose position and shape follow from the normalisation rather than from the data (Section 4). In combination with ChebNet, learned attention behaves similarly on several graphs (Section 5.2). We therefore propose an aggregator that learns the valley per graph:

$$H' \;=\; \underbrace{XW_I}_{\text{ego}} \;+\; \underbrace{SXW_S + S^\top XW_T}_{\text{neighbours}} \;+\; \underbrace{R(L)\,XW_R}_{\text{spectral valley}}, \qquad R(L) = U\,r(\Lambda)\,U^\top. \tag{3}$$

Aggregation does not only apply the response $1 - \lambda$, it also dilutes each node's own features. The ego term preserves them, which matters where labels are node-level, and the neighbour terms aggregate with $S = D^{-1/2}AD^{-1/2}$ along and $S^\top$ against the edge direction (Rossi et al., 2023); on undirected graphs both reduce to P_{sym}. Separate projections $W_\bullet \in \mathbb{R}^{d\times F_{\text{hidden}}}$ let the model read ego and neighbour information differently, as recommended for heterophilic graphs (Zhu et al., 2020; Lim et al., 2021; Platonov et al., 2023). The spectral term rescales each Laplacian eigenmode λ by

$$r(\lambda) = a(\lambda)\,|\lambda - c|^p, \qquad a(\lambda) = a_L \text{ for } \lambda < c, \quad a(\lambda) = a_R \text{ for } \lambda \geq c. \tag{4}$$

The response vanishes at the valley centre $c \in (0, \lambda_{\max})$, the exponent $p > 0$ sets its sharpness, narrow for $p < 1$ and broad for $p > 1$, and the side gains $a_L, a_R \in \mathbb{R}$ set slope and sign of either flank. With $c = p = 1$, $a_L = a_R = 1$ gives $|1 - \lambda|$ and $a_L = -a_R = 1$ the signed response $1 - \lambda$ of P_{sym}, so the fixed aggregators are special cases; we initialise at $|1 - \lambda|$ so that training learns departures from them. Each term is scaled by a scalar gate $\tau_\bullet$ (initialised to 1), which adds no expressivity but is fitted separately: all projections $W_\bullet$ and the operator parameters $\theta = (c, p, a_L, a_R, \tau)$ are learned on the training loss, the latter by a separate optimiser without weight decay, and the epoch is selected on the validation set. Test labels are never used.

3.4 Spectral Propagation

Building on Sections 3.1 to 3.3, we employ a spectral graph convolution to propagate information via the graph Laplacian, whose eigenbasis captures global frequency structure beyond local neighborhoods (Bruna et al., 2014; Defferrard et al., 2016). Let $L \in \mathbb{R}^{n \times n}$ denote the normalized Laplacian and $\tilde{L} := 2L/\lambda_{\max} - I = L - I$ its version shifted to $[-1, 1]$, using the bound $\lambda_{\max} = 2$. For node features $H^{(l)} \in \mathbb{R}^{n \times d'}$, spectral propagation is defined as

$$H^{(l+1)} = \mathrm{LN}\left(H^{(l)} + \mathrm{ReLU}\left(\sum_{k=0}^{K-1} T_k(\tilde{L})\, H^{(l)} W_k \right) \right), \tag{5}$$

with learnable parameters W_k, layer normalization LN, and Chebyshev polynomials $T_k(\tilde{L})$, which follow the recursion

$$T_0(\tilde{L}) = I, \qquad T_1(\tilde{L}) = \tilde{L}, \qquad T_k(\tilde{L}) = 2\tilde{L}\, T_{k-1}(\tilde{L}) - T_{k-2}(\tilde{L}), \quad k \geq 2. \tag{6}$$

This Chebyshev convolution, in the style of ChebNet (Defferrard et al., 2016), approximates spectral filters without eigendecomposition. Since $|T_k| \leq 1$ on $[-1, 1]$, the recursion is numerically stable. We use a standard Chebyshev filter rather than adaptive spectral operators such as NLSF or HeroFilter (Lin et al., 2024; Zhang et al., 2025), since spectral filters are effective under heterophily, where discriminative signal often sits at high frequencies (Chien et al., 2021; Bo et al., 2021), and a fixed downstream filter isolates the effect of the aggregation stage.

SpecValley combines $\ell = 3$ such layers of order $K = 5$ with the adaptive valley aggregator of Eq. (3), which supplies $H^{(0)}$; the ablation in Section 5.2 substitutes the other aggregators of Section 3.2 in the same position. Propagation thus operates on aggregated rather than raw inputs.

4 Theoretical Properties of Aggregation Representations

4.1 Computational Complexity

Let n be the number of nodes, $s = |E|$ the number of edges, and d and d' the input and hidden dimension. Every model projects the features to d' and applies ℓ polynomial filter layers of order K (ChebNet in SpecValley), at a common cost $C_0 := O(ndd') + \ell\, C_K$ with $C_K = O\big(K(nd'^2 + sd')\big)$ per layer (Table 1). P_{sym} adds one sparse product. The ego and neighbour terms of SpecValley add $O(ndd' + sd')$; its remaining cost lies in the spectral term $Ur(\Lambda)U^\top XW_R$. The eigendecomposition of L takes $O(n^3)$ time once per graph, U occupies $O(n^2)$ memory during training, and each pass multiplies by $U^\top$ and U at $O(n^2 d')$. Learning the valley only re-evaluates r at the n eigenvalues, at $O(n)$ per step. For $n > 14{,}000$, a degree-M Chebyshev expansion of r in $L - I$ ($M = 40$) replaces this path at $O(Msd')$ time and $O(Mnd')$ memory. Bilevel training adds one forward and backward pass per epoch, less than doubling the training cost.

Model	**Cost per forward pass**
ChebNet	C_0
P_{sym} + ChebNet	$C_0 + O(sd')$
SpecValley	$C_0 + O(n^2 d') + O(n^3)^*$

Table 1: Complexity; C_0 covers the input projection and the ChebNet layers. *Once per graph.

4.2 Uniform Aggregation as a Spectral Valley

Analyses of the contextual stochastic block model (CSBM) delimit when aggregation helps: one graph convolution widens linear separability by a factor of order $1/\sqrt{\bar{d}}$ in the expected degree

$\bar{d}$ (Baranwal et al., 2021) and attention helps only when class means are well separated or structural noise dominates, collapsing onto uniform convolution otherwise (Fountoulakis et al., 2023; Javaloy et al., 2023; Ma et al., 2025). Our datasets are not CSBM graphs, so these results frame rather than predict our observations; on dense graphs with weak features, they point to uniform aggregation.

For this, let A be symmetric and nonnegative, without self-loops and isolated nodes, with degree matrix D, and let $L = I - P_{\mathrm{sym}} = U\Lambda U^\top$ with $P_{\mathrm{sym}} = D^{-1/2}AD^{-1/2}$. We call $d_{\mathrm{eff}} := n/\|P_{\mathrm{sym}}\|_F^2$ the *effective degree*; for unweighted A it lies between the minimum and maximum degree and equals r on r-regular graphs (Appendix A.1).

Proposition 4.1. *In the setting above, $\frac{1}{n}\sum_k \lambda_k = 1$ and $\frac{1}{n}\sum_k(\lambda_k-1)^2 = 1/d_{\mathrm{eff}}$, and $P_{\mathrm{sym}} = U(I-\Lambda)U^\top$ has the spectral response $1-\lambda$, which vanishes at this mean. If moreover $d_{\mathrm{eff}} \geq 1/(\beta^2\eta)$ for some $\beta > 0$ and $\eta \in (0,1)$, then at least $(1-\eta)\,n$ of the eigenvalues $\lambda_1, \ldots, \lambda_n$ satisfy $|\lambda_k - 1| < \beta$, and P_{sym} scales each of the corresponding modes by a factor of absolute value less than β.*

The second statement is Chebyshev's inequality applied to the spectrum and holds for every graph; for random graphs with minimum expected degree well above $\log^2 n$, all non-trivial eigenvalues concentrate at 1 (Chung et al., 2003). Uniform aggregation thus places its zero where dense spectra accumulate: it is a fixed spectral valley rather than a low-pass filter, attenuating the bulk and keeping both ends of the spectrum, where community ($\lambda \approx 0$) and alternating ($\lambda \approx 2$) structure reside.

Interpretation. Proposition 4.1 explains why aggregation helps on dense graphs and where it acts. Isotropic noise spreads evenly over the modes, so on dense graphs most of it lies in the band that P_{sym} attenuates, which retains exactly a fraction $1/d_{\mathrm{eff}}$ of the expected noise energy. Whether the labels survive is a separate question: label energy in the same band is attenuated along with the noise. Since no fixed response suits every graph, this motivates learning the valley (Section 3.3).

4.3 The Spectral Energy Budget of Aggregation

The downstream filter works with what aggregation leaves. For this, fix *a priori* a partition $\{\mathcal{I}_b\}_{b=1}^m$ of the frequency indices into bands with projections $P_b := \sum_{k\in\mathcal{I}_b} u_k u_k^\top$ and band energies $E_b(Y) := \|P_bY\|_F^2$. Let $\mathcal{G}_\Gamma := \{\, g : \sigma(L) \to \mathbb{R} \,:\, \max_k |g(\lambda_k)| \leq \Gamma \,\}$ be the admissible filters, $g(L) = Ug(\Lambda)U^\top$, with extremal gains $\mu_b^-(g) := \min_{k\in\mathcal{I}_b}|g(\lambda_k)|$ and $\mu_b^+(g) := \max_{k\in\mathcal{I}_b}|g(\lambda_k)|$; g is *band-constant* if $\mu_b^-(g) = \mu_b^+(g)$ for all b.

Corollary 4.2. *In the setting above, let Y be a graph signal, such as the aggregated features $Y = A(x)X$, and $Z := g(L)\,Y$ with $g \in \mathcal{G}_\Gamma$. Then:*

(i) Band-wise budget. $\|P_bZ\|_F^2 = \sum_{k\in\mathcal{I}_b} g(\lambda_k)^2\|u_k^\top Y\|_2^2$, *hence* $\big(\mu_b^-(g)\big)^2 E_b(Y) \leq \|P_bZ\|_F^2 \leq \big(\mu_b^+(g)\big)^2 E_b(Y)$, *with equality for band-constant g.*

(ii) Non-creation. *If $E_b(Y) = 0$, then $P_bZ = 0$ for every $g \in \mathcal{G}_\Gamma$.*

With Proposition 4.1, (i) quantifies the valley: for $Y = P_{\mathrm{sym}}X$ and the central band $\mathcal{I}_\beta := \{k : |\lambda_k - 1| < \beta\}$ with projection P_β and band energy E_β, $u_k^\top Y = (1-\lambda_k)\,u_k^\top X$ gives $\|P_\beta Z\|_F^2 \leq \Gamma^2\beta^2 E_\beta(X)$ for every admissible g. Spectral aggregators move no energy between bands; non-spectral ones can.

Interpretation. The corollary makes the band-energy profile the interface between aggregation and filter: a linear spectral filter can only reweight what aggregation supplies (i) and cannot restore what it removed (ii), and on dense graphs this concerns the band that holds almost all modes. The ego term of Eq. (3) keeps every band of X available to the filter, consistent with the adaptive valley avoiding the collapse of the other aggregators on Actor and Tolokers. And where the label energy is spread over the modes like noise, as on Actor, no spectral aggregator can improve the ratio of label energy to noise. Proofs are given in Appendices A.1 and A.2.

4.4 Compatibility of Aggregation and Filter

In the model, aggregation and filter act in sequence. For spectral functions $q(L) = Uq(\Lambda)U^\top$ and $h(L) = Uh(\Lambda)U^\top$, diagonal matrices commute, so $h(L)\,q(L) = U\,h(\Lambda)\,q(\Lambda)\,U^\top = (hq)(L) =$

$q(L)\,h(L)$, and weight matrices, acting from the right, do not interfere. Each Chebyshev term of the first ChebNet layer (Eq. (5)) applied to $q(L)XW$ is thus one spectral filter with response $T_k(\lambda - 1)\,q(\lambda)$ (Shuman et al., 2013). For degree normalisation, all these responses vanish at $\lambda = 1$: the pair is a ChebNet of one order higher with a fixed root (Wu et al., 2019). On undirected graphs all branches of SpecValley are spectral as well, with responses 1, $1 - \lambda$ and $r(\lambda)$; the ego branch carries no root, and since r is in general not a polynomial, neither are the products. Attention and P_{mean} on non-regular graphs are not spectral functions of L and do not commute with $h(L)$ in general, since a matrix commutes with $h(L)$ only if it maps each of its eigenspaces into itself (Horn and Johnson, 2012); their output enters the filter as an input signal, to which Corollary 4.2 applies. All statements concern the linear step: ReLU and layer normalisation can move energy between modes.

5 Experimental Evaluation

5.1 Experimental Settings

We evaluate our method on node classification in dense relational graphs, using six heterophilic benchmarks: Squirrel-f, Chameleon-f, Actor, Crocodile (Rozemberczki et al., 2021), Tolokers (Platonov et al., 2023) and HM-categories (Bazhenov et al., 2025) (Table 2). Squirrel-f and Chameleon-f are the node-deduplicated variants of Platonov et al. (2023), which remove the train–test leakage caused by duplicated pages, and are used in place of the originals for comparison with prior work (Lin et al., 2024). Crocodile is a denser Wikipedia network from the same source, whose published target is continuous and which we discretise into five equally sized classes, as done for Chameleon and Squirrel by Pei et al. (2020) and on this graph by Xiao et al. (2023). The selection spans more than an order of magnitude in size and degree and deliberately covers two spectral regimes. On Squirrel-f, Chameleon-f, Crocodile and HM-categories, neighbourhood aggregation clearly improves over the feature-only baseline (Table 3), so label information is recoverable by spectral denoising. On Actor and Tolokers it is not: Actor is sparse with a label spectrum close to that of noise (Section 4.3), while Tolokers, and in part HM-categories, carry label information at the level of individual nodes, where models that separate ego from neighbour information and attention-based architectures perform best (Platonov et al., 2023; Bazhenov et al., 2025). Covering both regimes lets us attribute gains and failures to the spectral structure of a graph rather than to the choice of benchmarks.

Dataset	Nodes	Edges	Features	Classes	Degree
Squirrel-f*	2.2K	94K	2.1K	5	42
Chameleon-f*	0.9K	14K	2.3K	5	31
Actor*	7.6K	30K	932	5	8
Crocodile*	11.6K	360K	13.2K	5	62
Tolokers	11.8K	519K	10	2	88
HM-categories	46.6K	21.5M	120	21	923

*Table 2: Dataset statistics. *: 48/32/20 train/val/test split; Tolokers: 50/25/25; HM-categories: 10/10/80 (GraphLand RL split). Squirrel-f and Chameleon-f denote the node-deduplicated filtered variants from (Platonov et al., 2023).*

For Squirrel-f, Chameleon-f, Actor and Crocodile we use 10 random 48%/32%/20% train/validation/test splits (Liu et al., 2023; Yuan et al., 2023); for Tolokers the 10 official 50%/25%/25% splits (Platonov et al., 2023); and for HM-categories the official GraphLand RL split (10%/10%/80%) with three seeds (Bazhenov et al., 2025). We report 95% confidence intervals (Lin et al., 2024). All splits are stored and reused across models. Detailed configurations are provided in Appendix C.1.

5.2 Interplay of Aggregation Operator and Downstream Filter

To isolate the contribution of the aggregation operator, we cross five aggregators with four downstream filters (Table 3). As parameter-free references we use symmetric normalisation (*sym*) and mean aggregation (*mean*); as parameterised operators, learned attention (*attn*), initialised attention ($\mathit{attn}_{\mathrm{init}}$), whose query and key projections stay fixed after initialisation so that only the value projection is learned, and the adaptive valley (*valley*, Section 3.3). The filters cover the rational (APPNP (Gasteiger et al., 2019), ARMA (Bianchi et al., 2022)) and polynomial (Gegenbauer (Castro-Correa et al., 2024; Göksu, 2025), ChebNet (Defferrard et al., 2016)) families. A row without aggregation (*none*) and a block without a filter serve as references. All runs follow Section 5.1, train for 150 epochs, and report 95% confidence intervals.

Filt.	Aggr.	Squirrel*	Chameleon*	Actor	Crocodile	Tolokers**	HM-cat.
APPNP	none	36.2±1.4	37.6±2.2	36.0±0.6	69.7±0.7	76.9±1.1	37.7±0.7
	sym	33.4±1.1	42.5±2.5	24.9±1.0	75.7±0.4	75.4±0.7	53.1±0.4
	mean	33.5±0.8	42.4±3.3	25.1±0.7	76.5±0.5	73.4±0.7	51.4±1.4
	$\text{attn}_{\text{init}}$	33.9±1.7	43.8±2.8	24.9±0.6	76.0±0.6	73.2±0.8	53.5±0.7
	attn	34.3±0.8	42.3±2.2	24.9±0.6	75.8±0.6	75.4±0.8	54.4±0.1
	valley	39.9±2.0	40.4±3.0	34.8±0.8	74.0±0.8	80.6±0.8	59.3±0.9
ARMA	none	36.4±1.3	37.2±3.5	**36.4±0.7**	69.8±0.5	74.8±0.9	36.5±0.3
	sym	33.2±1.3	42.6±2.8	25.2±0.9	75.8±0.5	75.8±0.7	52.2±0.5
	mean	32.9±1.6	42.3±2.1	25.2±0.7	76.3±0.6	73.2±0.6	51.0±0.5
	$\text{attn}_{\text{init}}$	34.2±1.4	43.0±2.7	25.6±0.9	76.2±0.8	73.1±0.8	52.6±0.7
	attn	33.6±1.4	41.5±2.0	25.3±0.7	76.3±0.7	75.9±0.7	53.6±0.1
	valley	35.8±1.2	39.7±2.3	35.0±1.2	73.7±1.0	80.3±0.9	57.4±3.0
Gegen.	none	37.8±1.2	34.2±2.3	33.6±1.1	68.1±0.6	81.9±0.7	47.2±1.4
	sym	40.5±1.8	41.6±2.8	26.7±1.1	76.1±0.6	80.8±0.8	55.3±0.3
	mean	40.7±1.2	42.6±2.2	26.4±1.2	76.3±0.7	79.1±0.6	52.2±0.8
	$\text{attn}_{\text{init}}$	39.5±1.8	42.9±2.7	27.2±0.7	76.4±0.8	79.1±0.6	54.3±1.4
	attn	39.6±2.0	41.4±1.5	26.8±1.0	76.4±0.6	79.1±0.8	55.6±0.4
	valley	42.5±1.4	44.0±3.9	33.3±0.9	75.0±1.0	81.9±0.6	59.4±1.1
ChebNet	none	37.5±1.6	37.8±2.5	35.4±0.6	72.4±0.6	**85.2±0.5**	57.1±0.6
	sym	42.3±1.8	43.1±1.8	29.4±0.8	76.7±0.6	82.6±0.8	60.3±1.2
	mean	41.7±1.4	43.0±2.5	27.7±0.9	77.1±0.6	81.7±0.8	57.7±0.3
	$\text{attn}_{\text{init}}$	41.5±1.9	44.3±2.2	27.0±0.8	**77.3±0.6**	81.5±0.7	60.0±0.3
	attn	41.7±2.1	42.4±3.4	27.4±1.0	**77.3±0.5**	81.7±0.7	61.0±0.3
	valley	**42.7±2.6**	**44.5±1.5**	35.6±0.8	76.5±0.8	**85.2±0.5**	**63.2±0.9**
No filter	none	33.5±1.9	35.5±2.0	36.2±0.8	68.4±0.7	71.5±0.5	35.8±0.4
	sym	33.2±1.1	42.4±2.7	25.3±0.8	58.0±0.8	72.3±0.9	42.4±1.6
	mean	33.0±1.8	42.3±3.2	25.1±0.8	66.3±0.7	67.2±1.1	38.4±0.2
	$\text{attn}_{\text{init}}$	33.4±1.3	43.8±2.5	25.3±0.8	66.8±0.7	67.3±1.1	37.0±0.7
	attn	32.9±1.2	40.5±2.7	25.3±0.8	66.8±0.7	73.0±0.8	44.6±0.3
	valley	37.3±1.4	40.6±2.0	36.1±0.6	76.2±0.7	77.3±0.7	48.1±0.2

*Table 3: All combinations of aggregation operator and downstream filter. Accuracy (%), mean ± 95% CI over 10 splits, HM-categories over 3 seeds. *node-deduplicated variant of Platonov et al. (2023); **ROC-AUC. All runs 150 epochs. **Bold** marks the best value per dataset, underline the best per filter; colour compares each operator with* none *under the same filter (better, worse).*

The benefit depends on the spectral regime of the graph. On Chameleon-f, Crocodile and HM-categories every aggregator improves over *none* under each of the four filters (without a filter, only the valley does so on Crocodile). On Squirrel-f the picture is split: with the polynomial filters all of them improve, with APPNP and without a filter only the valley does, and with ARMA none does. On Actor, and for 16 of 20 combinations on Tolokers, the parameter-free and attention-based aggregators instead fall below *none*, consistent with label energy that is spread like noise or carried by individual nodes (Sections 4.2 and 4.3). The adaptive valley avoids this collapse: on Actor it stays within 1.5 points of *none*, where the others lose up to 11, and on Tolokers it matches *none* under the polynomial filters and exceeds it by 3.7 to 5.8 points otherwise. Comparing *attn* with $\textit{attn}_{\text{init}}$ without a downstream filter tests whether the learned weights add a dataset-specific alignment beyond the fixed aggregation structure. On Tolokers they gain 5.7 points there, but at most 2.8 once a filter is applied and essentially nothing for the polynomial filters, so the learned reweighting is not aligned with what the filter does downstream. HM-categories is the complementary case: attention helps with and without a filter, consistent with it exploiting node-specific as well as spectral information.

5.3 Benchmark Results

SpecValley (valley aggregator + ChebNet, Section 3.3) is level with or above the best published result on four of six benchmarks, within the confidence interval of the best on Tolokers and behind on Actor (Table 4). The benchmark runs use the protocol of Section 5.1 with 1000 epochs and a polynomial order K chosen per dataset on the validation score (Appendix C.1); Table 3 uses 150 epochs and $K = 5$ throughout, which explains the differences between the two tables. On Squirrel-filtered and Chameleon-filtered, spectral and frequency-aware methods constitute the strongest prior work. SpecValley improves over the next best (NLSFs (Lin et al., 2024)) by 2.4 and 1.3 points, with FAGCN (Bo et al., 2021) further below. Both margins lie within the confidence intervals. On Crocodile the listed prior work is self-supervised (GraphACL (Xiao et al., 2023), DSSL (Xiao et al., 2022), HGRL (Chen et al., 2022)), so we do not read the difference as a margin. On Tolokers and HM-categories, attention-based transformers (Polynormer (Deng et al., 2024), GAT-sep (Platonov et al., 2023), Exphormer (Shirzad et al., 2023), GT and GAT (Bazhenov et al., 2025)) are the strongest baselines. SpecValley leads on HM-categories by 0.5 points and trails Polynormer-r on Tolokers by 0.7 points, in both cases within the confidence intervals. Actor is the dataset where SpecValley is clearly behind: contrastive and graph-structure-

Rank	Squirrel*		Chameleon*		Actor	
1	**SpecValley**	**45.1** ±2.1	**SpecValley**	**43.8** ±3.1	MUSE	38.6 ±1.3
2	NLSFs	42.7 ±1.7	NLSFs	42.5 ±1.5	GloGNN	37.7 ±1.4
3	FAGCN	41.1 ±2.3	FAGCN	41.9 ±2.7	GGCN	37.5 ±1.6
4	GCN	39.5 ±1.5	ResNet+SGC	41.0 ±4.5	**SpecValley**	**35.7** ±0.9
Rank	Crocodile		Tolokers**		HM-cat.	
1	**SpecValley**	**77.2** ±0.5	Polynorm.-r	85.9 ±0.7	**SpecValley**	**69.7** ±1.2
2	GraphACL	66.2 ±0.2	**SpecValley**	**85.2** ±0.6	GT	69.2 ±0.5
3	DSSL	63.0 ±0.5	GAT-sep	83.8 ±0.4	GAT	68.0 ±0.3
4	HGRL	61.9 ±0.5	Exphormer	83.8 ±0.8	GCN	61.7 ±0.4

*Table 4: SpecValley vs. the three strongest reported baselines per dataset as published. Accuracy (%), mean ± 95% CI over 10 splits, HM-categories over 3 seeds. **ROC-AUC, *node-deduplicated (Platonov et al., 2023), over 1k epochs. On Crocodile the baselines are self-supervised, so the results are indicative rather than strictly comparable.*

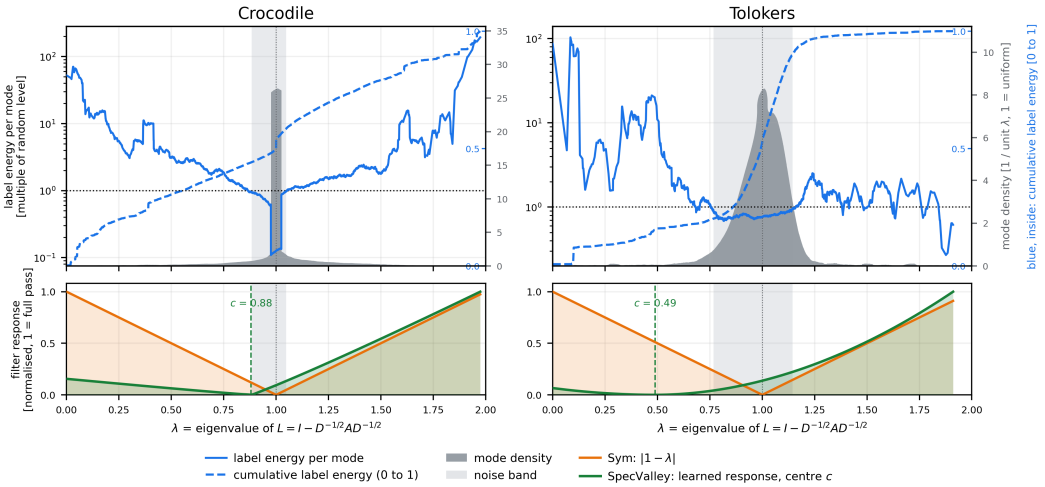

Figure 2: Label energy across the Laplacian spectrum (top) and the filter magnitude $|g(\lambda)|$ of degree normalisation and of SpecValley (bottom), for two dense graphs. Quantities are defined in the text; the remaining four graphs are in Appendix C.2.

aware methods (MUSE (Yuan et al., 2023), GloGNN (Li et al., 2022), GGCN (Yan et al., 2022)) dominate, consistent with the analysis showing that no spectral aggregator can separate signal from noise when the label energy is spread uniformly across the spectrum (Section 4.2).

6 Understanding Spectral Representations

6.1 Spectral Valley Aggregation

To see where aggregation acts, we diagonalise $L = I - D^{-1/2}AD^{-1/2} = U\Lambda U^\top$ and measure the share of label energy each mode carries, $\hat{e}_k = \|u_k^\top Y\|_2^2 / \|Y\|_F^2$ with $\sum_k \hat{e}_k = 1$ (Appendix C.2). Figure 2 plots $n\,\hat{e}_k$, the multiple of the level a random labelling would produce, together with the cumulative share, the eigenvalue density and a band marking modes significantly below the random level. The lower panel contrasts the fixed response $1 - \lambda$ of degree normalisation (Proposition 4.1) with the learned response $r(\lambda)$ of Eq. (4), which recovers it for $c = p = 1$ and $a_L = -a_R = 1$.

The two graphs have nearly the same spectrum: as Proposition 4.1 predicts for dense graphs, the eigenvalues pile up at $\lambda = 1$, with 76% and 66% of the modes in $|\lambda - 1| < 0.1$. The label energy tells them apart. On Crocodile this core carries only 17% of the label energy, so attenuating it removes mostly noise, and degree normalisation helps under the polynomial filters (Table 3). On Tolokers it carries 52%, so attenuation removes signal as well; there degree normalisation hurts under the same filters, and the learned valley moves its centre away from $\lambda = 1$. Mode crowding alone therefore does not decide whether aggregation helps; the label energy in the attenuated band matters. It is not a complete predictor, however: on HM-categories, whose core also carries most of the label energy, degree normalisation helps under every filter.

6.2 Generalisation, Robustness and Filter Order

We probe SpecValley along three axes that leave the protocol of Section 5.1 otherwise unchanged (Figure 3): the graph is thinned by halving the edges in six steps from 100% to 0%; the features are reduced in ten equal steps over the same range; and the order K of the downstream ChebNet layer (Eq. (5)) is swept over $K \in \{0, 1, 2, 3, 4, 5, 8, 10, 12, 15, 20, 25, 40, 60\}$.

Sparsification. Panel (a) shows that the effectiveness of SpecValley rests on the density of the edges: every halving costs performance, but the loss stays small down to about half the edges and grows only below that, so the scores remain relatively stable over a wide range. Panel (b) shows the converse: apart from HM-categories and from very strong reductions, the scores stay almost constant across the whole range, so the label information resides in the structure rather than in the feature dimension.

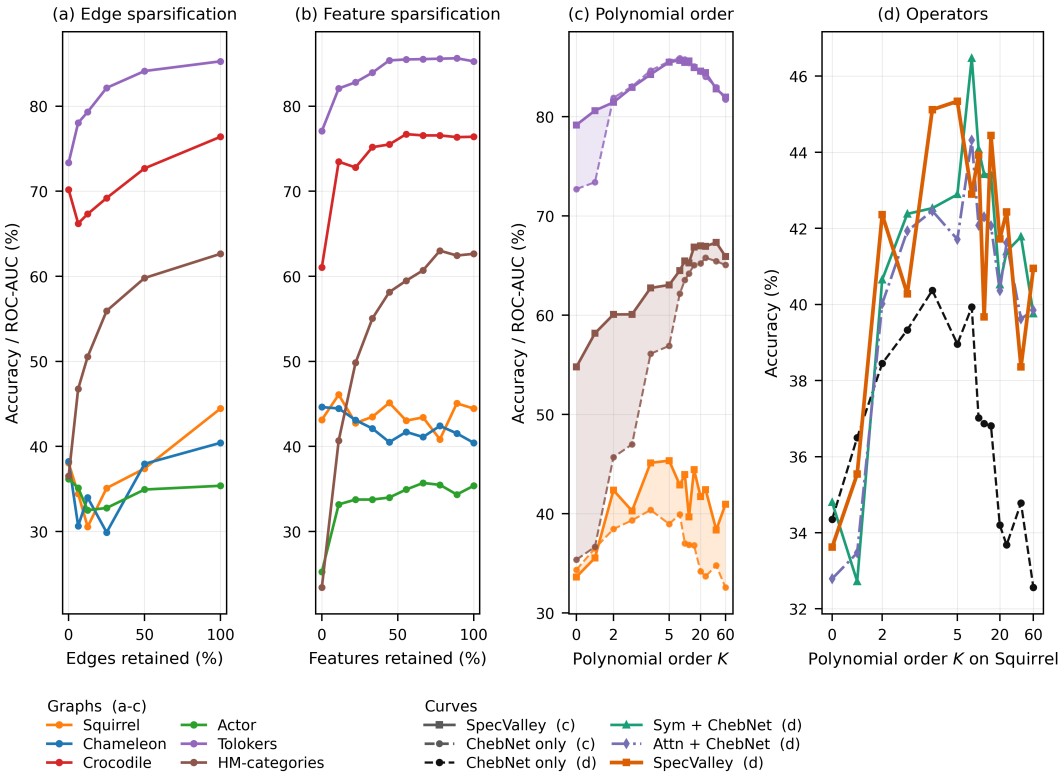

Figure 3: Robustness and filter order. (a,b) Edges and features removed progressively, SpecValley with ChebNet. (c) Polynomial order K of the ChebNet layer with and without the valley aggregator, all six graphs. (d) The same sweep on Squirrel-f for four aggregation operators. Three splits (one on HM-categories), 150 epochs.

Filter order. Panel (c) shows that the filter does not absorb the aggregation step. On the graphs where the valley is effective (Squirrel-f, Chameleon-f, Crocodile and HM-categories) the gap to ChebNet alone does not close with K, even at orders far beyond those used in the benchmark. The composition of aggregator and filter is a polynomial of only one order higher (Section 4.4), so this is not a question of expressive capacity but of what the two parameterisations reach in training. On Tolokers and Actor the two curves coincide from moderate orders onward, consistent with spectra in which no aggregator finds a band to attenuate (Section 6.1). Panel (d) repeats the sweep on Squirrel-f with four operators. All three aggregators stay above ChebNet alone across the whole range, and the spread between them is small next to that gap, so at fixed K it is the presence of an aggregation step, more than its particular form, that accounts for the difference. Beyond $K \approx 20$ every curve declines, including ChebNet alone. Further results are reported in Appendix C.3.

7 Conclusion

The aggregation step that precedes every graph filter is a spectral operator in its own right, and on dense graphs its fixed zero falls exactly where the eigenvalues accumulate. Making this step learnable is enough to change the outcome: SpecValley, a learned valley in front of an unchanged ChebNet, is the best on four of six heterophilic benchmarks and within the confidence interval of the best on a fifth, and the ablation over aggregators and filters shows that no downstream filter closes the gap on its own. The label-energy spectrum explains where the valley helps and where, as on Actor, no spectral aggregator can.

Limitations and Future Work. Our analysis is confined to dense heterophilic graphs with informative spectral structure and to the linear part of the pipeline; ReLU and layer normalisation can move energy between modes, and the operator parameters are fitted on half of the validation set. Two questions remain open: when attention before propagation aligns with the spectrum and when it learns node-specific weights, and why a Chebyshev filter does not reproduce the gain of the valley by itself, although the composition of valley and filter is a polynomial of only one order higher.

AI USE STATEMENT

We used generative AI tools for research ideation, method development, for writing experimental code, for writing and editing the text and for generating the figures. We have reviewed the AI-assisted work and the final content of this work.

REPRODUCIBILITY STATEMENT

All datasets are public (Section 5.1). The stored splits, the training protocol and all hyperparameters are given in Appendix C.1 and complete proofs in Appendix A. Code and splits are included with the submission.

REFERENCES

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://arxiv.org/abs/1609.02907.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://arxiv.org/abs/1710.10903.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://arxiv.org/abs/2006.11468.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=S1e2agRcFS.

Yujun Yan, Milad Hashemi, Kevin Swersky, Yaoqing Yang, and Danai Koutra. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. In *IEEE International Conference on Data Mining (ICDM)*, 2022. URL https://arxiv.org/abs/2102.06462.

Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=S1ldO2EFPr.

Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *International Conference on Machine Learning (ICML)*, 2019. URL https://arxiv.org/abs/1905.00067.

Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser-Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://arxiv.org/abs/2110.14446.

Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep graph infomax. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://arxiv.org/abs/1809.10341.

Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning (ICML)*, 2020. URL https://arxiv.org/abs/2006.05582.

Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2020. URL https://arxiv.org/abs/2006.09963.

Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. Learning discrete structures for graph neural networks. In *International Conference on Machine Learning (ICML)*, 2019. URL https://arxiv.org/abs/1903.11960.

Lirong Wu, Haitao Lin, Yufei Huang, and Stan Z. Li. Knowledge distillation improves graph structure augmentation for graph neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://openreview.net/forum?id=7yHte3tH8Xh.

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=n6jl7fLxrP.

Ya-Wei Eileen Lin, Ronen Talmon, and Ron Levie. Equivariant machine learning on graphs with nonlinear spectral filters. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2406.01249.

Christian Koke and Daniel Cremers. Holonets: Spectral convolutions do extend to directed graphs. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=EhmEwfavOW.

Chenhui Deng, Zichao Yue, and Zhiru Zhang. Polynormer: Polynomial-expressive graph transformer in linear time. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=hmv1LpNfXa.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018. URL https://arxiv.org/abs/1801.07606.

Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021. URL https://arxiv.org/abs/2101.00797.

Mingguo He, Zhewei Wei, Zengfeng Huang, and Hongteng Xu. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://arxiv.org/abs/2106.10994.

Deyu Bo, Chuan Shi, Lele Wang, and Renjie Liao. Specformer: Spectral graph neural networks meet transformers. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=0pdSt3oyJa1.

Xiyuan Wang and Muhan Zhang. How powerful are spectral graph neural networks. In *International Conference on Machine Learning (ICML)*, 2022. URL https://arxiv.org/abs/2205.11172.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of GNNs under heterophily: Are we really making progress? In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=tJbbQfw-5wv.

Emanuele Rossi, Bertrand Charpentier, Francesco Di Giovanni, Fabrizio Frasca, Stephan Günnemann, and Michael M. Bronstein. Edge directionality improves learning on heterophilic graphs. In *Proceedings of the Second Learning on Graphs Conference (LoG)*, volume 231 of *Proceedings of Machine Learning Research*, pages 25:1–25:27. PMLR, 2023. URL https://proceedings.mlr.press/v231/rossi24a.html.

Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=wTTjnvGphYj.

Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2205.12454.

Devin Kreuzer, Dominique Beaini, William L. Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://arxiv.org/abs/2106.03893.

Enyan Dai, Shijie Zhou, Zhimeng Guo, and Suhang Wang. Label-wise graph convolutional network for heterophilic graphs, 2021. URL https://arxiv.org/abs/2110.08128.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 6861–6871, 2019.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. URL https://arxiv.org/abs/1706.02216.

Fan R. K. Chung. *Spectral Graph Theory*. Number 92 in CBMS Regional Conference Series in Mathematics. American Mathematical Society, 1997.

Jianan Zhao, Chaozhuo Li, Qianlong Wen, Yiqi Wang, Yuming Liu, Hao Sun, Xing Xie, and Yanfang Ye. Gophormer: Ego-graph transformer for node classification. *CoRR*, abs/2110.13094, 2021. URL https://arxiv.org/abs/2110.13094.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR)*, 2014. URL https://arxiv.org/abs/1312.6203.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. URL https://arxiv.org/abs/1606.09375.

Shuaicheng Zhang, Haohui Wang, Junhong Lin, Xiaojie Guo, Yada Zhu, Si Zhang, Dongqi Fu, and Dawei Zhou. Herofilter: Adaptive spectral graph filter for varying heterophilic relations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2510.10864.

Aseem Baranwal, Kimon Fountoulakis, and Aukosh Jagannath. Graph convolution for semi-supervised classification: Improved linear separability and out-of-distribution generalization. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 684–693. PMLR, 2021. URL https://proceedings.mlr.press/v139/baranwal21a.html.

Kimon Fountoulakis, Amit Levi, Shenghao Yang, Aseem Baranwal, and Aukosh Jagannath. Graph attention retrospective. *Journal of Machine Learning Research*, 24(246):1–52, 2023. URL http://jmlr.org/papers/v24/22-125.html.

Adrián Javaloy, Pablo Sanchez-Martin, Amit Levi, and Isabel Valera. Learnable graph convolutional attention networks. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=WsUMeHPo-2.

Zhongtian Ma, Qiaosheng Zhang, Bocheng Zhou, Yexin Zhang, Shuyue Hu, and Zhen Wang. Graph attention is not always beneficial: A theoretical analysis of graph attention mechanisms via contextual stochastic block models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 42252–42292. PMLR, 2025. URL https://proceedings.mlr.press/v267/ma25w.html.

Fan Chung, Linyuan Lu, and Van Vu. Spectra of random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 100(11):6313–6318, 2003.

David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*, 30(3):83–98, 2013.

Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2 edition, 2012.

Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021. doi: 10.1093/comnet/cnab014.

Gleb Bazhenov, Oleg Platonov, and Liudmila Prokhorenkova. GraphLand: Evaluating graph machine learning models on diverse industrial data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2409.14500.

Teng Xiao, Huaisheng Zhu, Zhengyu Chen, and Suhang Wang. Simple and asymmetric graph contrastive learning without augmentations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Yixin Liu, Yizhen Zheng, Daokun Zhang, Vincent C. S. Lee, and Shirui Pan. Beyond smoothing: Unsupervised graph representation learning with edge heterophily discriminating. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023. doi: 10.1609/aaai.v37i4.25573. URL https://ojs.aaai.org/index.php/AAAI/article/view/25573.

Mengyi Yuan, Minjie Chen, and Xiang Li. MUSE: Multi-view contrastive learning for heterophilic graphs. *arXiv preprint arXiv:2307.16026*, 2023. URL https://arxiv.org/abs/2307.16026.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://arxiv.org/abs/1810.05997.

Filippo Maria Bianchi, Daniele Grattarola, Lorenzo Livi, and Cesare Alippi. Graph neural networks with convolutional ARMA filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3496–3507, 2022. doi: 10.1109/TPAMI.2021.3054830. URL https://arxiv.org/abs/1901.01343.

Jhon A. Castro-Correa, Jhony H. Giraldo, Mohsen Badiey, and Fragkiskos D. Malliaros. Gegenbauer graph neural networks for time-varying signal reconstruction. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9):11734–11745, 2024. doi: 10.1109/TNNLS.2024.3381069. URL https://arxiv.org/abs/2403.19800.

Hüseyin Göksu. Gegenbauernet: Finding the optimal compromise in the gnn flexibility-stability trade-off, 2025. URL https://arxiv.org/abs/2511.13730. arXiv preprint.

Teng Xiao, Zhengyu Chen, Zhimeng Guo, Zeyang Zhuang, and Suhang Wang. Decoupled self-supervised learning for non-homophilous graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Jingfan Chen, Guanghui Zhu, Yifan Qi, Chunfeng Yuan, and Yihua Huang. Towards self-supervised learning on graphs with heterophily. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM)*, pages 201–211, 2022. doi: 10.1145/3511808.3557478.

Hamed Shirzad, Ameya Velingker, Balaji Venkatachalam, Danica J. Sutherland, and Ali Kemal Sinop. Exphormer: Sparse transformers for graphs. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *PMLR*, 2023.

Xiang Li, Renyu Zhu, Yao Cheng, Caihua Shan, Siqiang Luo, Dongsheng Li, and Weining Qian. Finding global homophily in graph neural networks when meeting heterophily. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pages 13242–13256. PMLR, 2022. URL https://proceedings.mlr.press/v162/li22ad.html.

A Theoretical Analysis

A.1 Proof of Proposition 4.1 (Uniform Aggregation as a Spectral Valley)

Let $A \in \mathbb{R}^{n\times n}$ be symmetric with nonnegative entries, without self-loops ($A_{ii} = 0$) and without isolated nodes, i.e. with degrees $d_i := \sum_j A_{ij} > 0$ for all i; let $D := \mathrm{diag}(d_1, \dots, d_n)$, $d_{\min} := \min_i d_i$ and $d_{\max} := \max_i d_i$. We call the graph r-regular if $d_i = r$ for all i, writing r for the common degree since D denotes the degree matrix. Let

$$P_{\mathrm{sym}} := D^{-1/2} A D^{-1/2}, \qquad L := I - P_{\mathrm{sym}} = U \Lambda U^\top, \qquad d_{\mathrm{eff}} := \frac{n}{\|P_{\mathrm{sym}}\|_F^2},$$

where $U = [u_1, \dots, u_n]$ is orthogonal and $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_n)$ lists the eigenvalues of L with multiplicity (this decomposition exists by Step 0 below). We call u_k the k-th mode, write e_k for the k-th standard basis vector, and for $q : \sigma(L) \to \mathbb{R}$ define the spectral function $q(L) := U\,\mathrm{diag}\big(q(\lambda_1), \dots, q(\lambda_n)\big)\,U^\top$.

Proposition 4.1. In the setting above, $\frac{1}{n}\sum_k \lambda_k = 1$ and $\frac{1}{n}\sum_k (\lambda_k - 1)^2 = 1/d_{\mathrm{eff}}$, and $P_{\mathrm{sym}} = U(I-\Lambda)U^\top$ has the spectral response $1-\lambda$, which vanishes at this mean. If moreover $d_{\mathrm{eff}} \geq 1/(\beta^2\eta)$ for some $\beta > 0$ and $\eta \in (0,1)$, then at least $(1-\eta)\,n$ of the eigenvalues $\lambda_1, \dots, \lambda_n$ satisfy $|\lambda_k - 1| < \beta$, and P_{sym} scales each of the corresponding modes by a factor of absolute value less than β. For unweighted A, $d_{\min} \leq d_{\mathrm{eff}} \leq d_{\max}$, with $d_{\mathrm{eff}} = r$ on r-regular graphs.

Proof. Step 0 checks that all objects are well defined, Steps 1–3 establish the first two moments of the spectrum and the response of P_{sym}, Steps 4–5 the concentration and the attenuation, and Step 6 the degree bounds.

Step 0 (well-posedness). Since every $d_i > 0$, $D^{-1/2}$ exists. As A is symmetric and D diagonal, $P_{\mathrm{sym}}^\top = D^{-1/2} A^\top D^{-1/2} = P_{\mathrm{sym}}$, so P_{sym} and L are real symmetric, and by the spectral theorem $L = U\Lambda U^\top$ with orthogonal U and real λ_k. Since $d_1 = \sum_j A_{1j} > 0$ and $A_{1j} \geq 0$, some entry of A is positive, so $\|P_{\mathrm{sym}}\|_F^2 = \sum_{i,j} A_{ij}^2/(d_i d_j) > 0$ and $d_{\mathrm{eff}} \in (0, \infty)$.

Step 1 (mean). Since $A_{ii} = 0$, the diagonal of L is $L_{ii} = 1 - (P_{\mathrm{sym}})_{ii} = 1 - A_{ii}/d_i = 1$, so $\mathrm{tr}(L) = n$. By cyclicity of the trace and $U^\top U = I$, $\mathrm{tr}(L) = \mathrm{tr}(U\Lambda U^\top) = \mathrm{tr}(\Lambda U^\top U) = \sum_k \lambda_k$. Hence $\frac{1}{n}\sum_k \lambda_k = 1$.

Step 2 (spectral form and mode-wise action). Since U is square and orthogonal, $I = UU^\top$, and

$$P_{\mathrm{sym}} \;=\; I - L \;=\; UU^\top - U\Lambda U^\top \;=\; U(I-\Lambda)U^\top, \tag{7}$$

so $P_{\mathrm{sym}} = q(L)$ with $q(\lambda) = 1 - \lambda$. With $U^\top u_k = e_k$, this gives $P_{\mathrm{sym}} u_k = U(I-\Lambda)e_k = (1-\lambda_k)\,u_k$. By symmetry of P_{sym} (Step 0), for every $Y \in \mathbb{R}^{n\times F}$,

$$u_k^\top P_{\mathrm{sym}} Y \;=\; (P_{\mathrm{sym}} u_k)^\top Y \;=\; (1-\lambda_k)\,u_k^\top Y, \tag{8}$$

i.e. P_{sym} multiplies the component of every signal along u_k by $1-\lambda_k$. The response $1-\lambda$ vanishes if and only if $\lambda = 1$, the mean of Step 1.

Step 3 (mean-square deviation). For every $M \in \mathbb{R}^{n\times F}$, orthogonality of U gives

$$\|M\|_F^2 \;=\; \mathrm{tr}(M^\top M) \;=\; \mathrm{tr}(M^\top UU^\top M) \;=\; \|U^\top M\|_F^2 \;=\; \sum_{k=1}^{n} \|u_k^\top M\|_2^2, \tag{9}$$

since the k-th row of $U^\top M$ is $u_k^\top M$; this is Parseval's identity in the graph Fourier basis. For $M = P_{\mathrm{sym}}$, (8) with $Y = I$ gives $u_k^\top P_{\mathrm{sym}} = (1-\lambda_k)\,u_k^\top$, hence $\|u_k^\top P_{\mathrm{sym}}\|_2^2 = (1-\lambda_k)^2\|u_k\|_2^2 = (\lambda_k - 1)^2$, and (9) yields

$$\|P_{\mathrm{sym}}\|_F^2 \;=\; \sum_{k=1}^{n} (\lambda_k - 1)^2, \qquad \text{i.e.} \qquad \frac{1}{n}\sum_k (\lambda_k - 1)^2 \;=\; \frac{\|P_{\mathrm{sym}}\|_F^2}{n} \;=\; \frac{1}{d_{\mathrm{eff}}}.$$

Step 4 (concentration). Let $K_\beta := \{k : |\lambda_k - 1| \geq \beta\}$. Every $k \in K_\beta$ contributes at least β^2 to the sum of Step 3, and all other terms are nonnegative, so

$$\beta^2 \, |K_\beta| \;\leq\; \sum_{k \in K_\beta} (\lambda_k - 1)^2 \;\leq\; \sum_{k=1}^{n} (\lambda_k - 1)^2 \;=\; \frac{n}{d_{\text{eff}}}.$$

This is Chebyshev's inequality for the uniform distribution over $\lambda_1, \ldots, \lambda_n$, whose mean is 1 and whose variance is $1/d_{\text{eff}}$ by Steps 1 and 3. Since $\beta > 0$, $d_{\text{eff}} > 0$ and $d_{\text{eff}} \geq 1/(\beta^2 \eta)$, it follows that $|K_\beta| \leq n/(\beta^2 d_{\text{eff}}) \leq \eta \, n$, so at least $n - |K_\beta| \geq (1 - \eta) \, n$ indices satisfy $|\lambda_k - 1| < \beta$.

Step 5 (attenuation). For every $k \notin K_\beta$, $|1 - \lambda_k| = |\lambda_k - 1| < \beta$, so by (8) P_{sym} scales the corresponding mode by a factor of absolute value less than β.

Step 6 (degree bounds). For unweighted A, $A_{ij} \in \{0, 1\}$ gives $A_{ij}^2 = A_{ij}$, and with $\mathcal{N}(i) := \{j : A_{ij} = 1\}$, so that $|\mathcal{N}(i)| = d_i$,

$$\|P_{\text{sym}}\|_F^2 \;=\; \sum_{i,j} \frac{A_{ij}}{d_i d_j} \;=\; \sum_{i=1}^{n} \frac{1}{d_i} \sum_{j \in \mathcal{N}(i)} \frac{1}{d_j}.$$

The inner sum has d_i terms, each in $[1/d_{\max}, \, 1/d_{\min}]$, so it lies in $[d_i/d_{\max}, \, d_i/d_{\min}]$, and each outer summand lies in $[1/d_{\max}, \, 1/d_{\min}]$. Summing over the n nodes, $n/d_{\max} \leq \|P_{\text{sym}}\|_F^2 \leq n/d_{\min}$, i.e. $d_{\min} \leq d_{\text{eff}} \leq d_{\max}$. On an r-regular graph every outer summand equals $\frac{1}{r} \cdot r \cdot \frac{1}{r} = \frac{1}{r}$, hence $\|P_{\text{sym}}\|_F^2 = n/r$ and $d_{\text{eff}} = r$. □

Remark (noise). As in Step 2, every spectral function satisfies $u_k^\top q(L) \, Y = q(\lambda_k) \, u_k^\top Y$, so $q(L)$ scales the energy of mode k by $q(\lambda_k)^2$, for signal and noise alike. Let $N \in \mathbb{R}^{n \times F}$ have i.i.d. zero-mean entries of variance $\sigma^2 > 0$. Since $\mathbb{E}[N_{:j} N_{:j}^\top] = \sigma^2 I$ for every column,

$$\mathbb{E}\|PN\|_F^2 \;=\; F\sigma^2 \, \|P\|_F^2 \qquad \text{for every } P \in \mathbb{R}^{n \times n}. \tag{10}$$

With $P = P_{\text{sym}}$ and Step 3, $\mathbb{E}\|P_{\text{sym}} N\|_F^2 = \mathbb{E}\|N\|_F^2 / d_{\text{eff}}$: uniform aggregation retains exactly a fraction $1/d_{\text{eff}}$ of the expected noise energy. With $P = u_k^\top$, $\mathbb{E}\|u_k^\top N\|_2^2 = F\sigma^2$ for every k: isotropic noise spreads evenly over the modes. If a signal S spreads evenly as well, $\|u_k^\top S\|_2^2 = \|S\|_F^2 / n$ for all k, then by (9) every $q(L)$ with $\sum_k q(\lambda_k)^2 > 0$ scales the signal energy and the expected noise energy by the same factor $\frac{1}{n} \sum_k q(\lambda_k)^2$, so no spectral aggregator changes their ratio.

Remark (mean aggregation). $P_{\text{mean}} := D^{-1} A = D^{-1/2} P_{\text{sym}} D^{1/2}$ is similar to P_{sym} and has the same eigenvalues $1 - \lambda_k$, but it is symmetric, and then equal to P_{sym}, only if the degree is constant on every connected component. Otherwise it is not a spectral function of L, and the proposition applies to it only through its eigenvalues.

Remark (provenance). Steps 1–3 are standard facts about the normalised Laplacian (Chung, 1997), and Step 4 is Chebyshev's inequality. What the proposition adds is the reading of uniform aggregation as a *fixed spectral valley* whose zero coincides with the mean of the spectrum, together with the effective degree d_{eff} as the quantity that controls how many modes lie in that valley. For random graphs with minimum expected degree well above $\log^2 n$, all non-trivial eigenvalues concentrate at 1 (Chung et al., 2003); the proposition gives a weaker but deterministic statement for every graph. The $1/d_{\text{eff}}$ noise reduction is the energy counterpart of the $1/\sqrt{\bar{d}}$ gain in separability in the expected degree $\bar{d}$ of Baranwal et al. (2021).

Remark (what the proposition does not claim). The bound in Step 4 uses only the second moment of the spectrum and is therefore conservative; it is informative only for $d_{\text{eff}} > 1/\beta^2$, since otherwise $n/(\beta^2 d_{\text{eff}}) \geq n$. The effective degree is not the mean degree: on a star with $n \geq 3$ nodes the mean degree is $2(n-1)/n < 2$, but $\|P_{\text{sym}}\|_F^2 = 2$ and hence $d_{\text{eff}} = n/2$, and indeed $n - 2$ eigenvalues equal 1 exactly. The proposition locates the modes, not the labels: whether attenuating the core removes more noise than label information depends on the task, and no statement about accuracy follows. The noise remark assumes isotropic noise; noise correlated with the graph does not spread evenly over the modes.

A.2 Proof of Corollary 4.2 (The Spectral Energy Budget of Aggregation)

Let $L = U\Lambda U^\top$ be as in Appendix A.1, with orthogonal $U = [u_1, \dots, u_n]$ and $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_n)$. Fix *a priori* a partition $\{\mathcal{I}_b\}_{b=1}^m$ of $\{1, \dots, n\}$ into nonempty bands, and set

$$P_b := \sum_{k \in \mathcal{I}_b} u_k u_k^\top, \qquad E_b(Y) := \|P_b Y\|_F^2 \quad \text{for } Y \in \mathbb{R}^{n \times F}.$$

Let $\mathcal{G}_\Gamma := \{\, g : \sigma(L) \to \mathbb{R} \,:\, \max_k |g(\lambda_k)| \le \Gamma \,\}$, and for $g \in \mathcal{G}_\Gamma$ put $g(L) := U g(\Lambda) U^\top$ with $g(\Lambda) := \mathrm{diag}\big(g(\lambda_1), \dots, g(\lambda_n)\big)$,

$$\mu_b^-(g) := \min_{k \in \mathcal{I}_b} |g(\lambda_k)|, \qquad \mu_b^+(g) := \max_{k \in \mathcal{I}_b} |g(\lambda_k)|;$$

g is *band-constant* if $\mu_b^-(g) = \mu_b^+(g) =: \mu_b(g)$ for every b.

Corollary 4.2. In the setting above, let $Y \in \mathbb{R}^{n \times F}$ be a graph signal, such as the aggregated features $Y = A(x)X$, and $Z := g(L)\, Y$ with $g \in \mathcal{G}_\Gamma$. Then:

(i) Band-wise budget. $\|P_b Z\|_F^2 = \sum_{k \in \mathcal{I}_b} g(\lambda_k)^2 \|u_k^\top Y\|_2^2$, hence $\big(\mu_b^-(g)\big)^2 E_b(Y) \le \|P_b Z\|_F^2 \le \big(\mu_b^+(g)\big)^2 E_b(Y)$, with equality for band-constant g.

(ii) Non-creation. If $E_b(Y) = 0$, then $P_b Z = 0$ for every $g \in \mathcal{G}_\Gamma$.

Proof. For an index set $\mathcal{I} \subseteq \{1, \dots, n\}$ write $\Pi_{\mathcal{I}} := \sum_{k \in \mathcal{I}} e_k e_k^\top$ for its coordinate selector, and $\Pi_b := \Pi_{\mathcal{I}_b}$. Steps 0–3 collect the properties of the projections, Steps 4–6 prove (i) and (ii), and Steps 7–8 prove the two statements of the main text that follow the corollary: the central-band bound and the absence of cross-band transfer for spectral aggregators.

Step 0 (well-posedness). U is orthogonal by Step 0 of Appendix A.1. Every $g \in \mathcal{G}_\Gamma$ is defined on $\sigma(L)$, so $g(\Lambda)$ and $g(L)$ are well defined, and since every band is nonempty, $\mu_b^\pm(g)$ are minima and maxima over nonempty finite sets.

Step 1 (projections in the eigenbasis). Since $u_k = U e_k$,

$$P_b \;=\; \sum_{k \in \mathcal{I}_b} U e_k e_k^\top U^\top \;=\; U \Pi_b U^\top. \tag{11}$$

As $\{\mathcal{I}_b\}$ partitions $\{1, \dots, n\}$, $\sum_b \Pi_b = I$ and $\Pi_b \Pi_{b'} = \delta_{bb'} \Pi_b$. With $U^\top U = U U^\top = I$ this gives $\sum_b P_b = I$, $P_b P_{b'} = \delta_{bb'} P_b$, and $P_b^\top = P_b$.

Step 2 (commutation with spectral functions). Let $h : \sigma(L) \to \mathbb{R}$ and $h(L) := U h(\Lambda) U^\top$. By (11) and $U^\top U = I$, $P_b\, h(L) = U \Pi_b h(\Lambda) U^\top$ and $h(L)\, P_b = U h(\Lambda) \Pi_b U^\top$. As Π_b and $h(\Lambda)$ are diagonal, they commute, hence

$$P_b\, h(L) \;=\; h(L)\, P_b \qquad \text{for every } b \text{ and every } h. \tag{12}$$

Step 3 (band energy in the modes). For every $M \in \mathbb{R}^{n \times F}$, (11) gives $U^\top P_b M = \Pi_b U^\top M$, whose k-th row is $u_k^\top M$ for $k \in \mathcal{I}_b$ and zero otherwise. With the Frobenius isometry $\|P_b M\|_F = \|U^\top P_b M\|_F$ of (9),

$$E_b(M) \;=\; \|P_b M\|_F^2 \;=\; \sum_{k \in \mathcal{I}_b} \|u_k^\top M\|_2^2. \tag{13}$$

Step 4 (identity of part (i)). By (11), $U^\top U = I$ and $g(L) = U g(\Lambda) U^\top$,

$$U^\top P_b Z \;=\; U^\top U \Pi_b U^\top U g(\Lambda) U^\top Y \;=\; \Pi_b\, g(\Lambda)\, U^\top Y,$$

whose k-th row is $g(\lambda_k)\, u_k^\top Y$ for $k \in \mathcal{I}_b$ and zero otherwise. Since $\|P_b Z\|_F = \|U^\top P_b Z\|_F$ and each $g(\lambda_k)$ is a scalar,

$$\|P_b Z\|_F^2 \;=\; \sum_{k \in \mathcal{I}_b} \big\| g(\lambda_k)\, u_k^\top Y \big\|_2^2 \;=\; \sum_{k \in \mathcal{I}_b} g(\lambda_k)^2\, \|u_k^\top Y\|_2^2. \tag{14}$$

Steps 3 and 4 use neither the bound Γ nor the partition property: they hold verbatim for every function g on $\sigma(L)$ and for every index set $\mathcal{I}$ in place of $\mathcal{I}_b$, with $P_{\mathcal{I}} := U\Pi_{\mathcal{I}}U^\top$ and $E_{\mathcal{I}}(M) := \|P_{\mathcal{I}}M\|_F^2$.

Step 5 (two-sided bound and equality case). For $k \in \mathcal{I}_b$, the definition of $\mu_b^{\pm}(g)$ gives $\mu_b^-(g) \le |g(\lambda_k)| \le \mu_b^+(g)$, and squaring preserves these inequalities between nonnegative numbers. Bounding every summand of (14) on both sides and using $\sum_{k\in\mathcal{I}_b}\|u_k^\top Y\|_2^2 = E_b(Y)$ from (13),

$$\big(\mu_b^-(g)\big)^2 E_b(Y) \;\le\; \|P_bZ\|_F^2 \;\le\; \big(\mu_b^+(g)\big)^2 E_b(Y).$$

If g is band-constant, both bounds equal $\mu_b(g)^2E_b(Y)$, so $\|P_bZ\|_F^2 = \mu_b(g)^2E_b(Y)$. This proves (i).

Step 6 (part (ii)). If $E_b(Y) = 0$, the upper bound of Step 5 gives $\|P_bZ\|_F^2 \le 0$, hence $P_bZ = 0$. Only the finiteness of $\mu_b^+(g)$ was used, so the conclusion holds for every $g \in \mathcal{G}_\Gamma$ simultaneously. This proves (ii).

Step 7 (central band). Let $\beta > 0$, $\mathcal{I}_\beta := \{k : |\lambda_k - 1| < \beta\}$, $P_\beta := P_{\mathcal{I}_\beta}$ and $E_\beta := E_{\mathcal{I}_\beta}$. For $Y = P_{\mathrm{sym}}X$, (8) gives $u_k^\top Y = (1-\lambda_k)\,u_k^\top X$. By (14) for the index set $\mathcal{I}_\beta$ (Step 4), $g(\lambda_k)^2 \le \Gamma^2$, and $(1-\lambda_k)^2 < \beta^2$ for $k \in \mathcal{I}_\beta$,

$$\|P_\beta Z\|_F^2 \;=\; \sum_{k\in\mathcal{I}_\beta} g(\lambda_k)^2(1-\lambda_k)^2\|u_k^\top X\|_2^2 \;\le\; \Gamma^2\beta^2 \sum_{k\in\mathcal{I}_\beta}\|u_k^\top X\|_2^2 \;=\; \Gamma^2\beta^2\,E_\beta(X)$$

for every $g \in \mathcal{G}_\Gamma$, using (13) for $\mathcal{I}_\beta$ in the last step. If $\mathcal{I}_\beta = \emptyset$, both sides are zero.

Step 8 (cross-band transfer). Let $q : \sigma(L) \to \mathbb{R}$. For $b \ne b'$, (12) with $h = q$ and Step 1 give $P_b\,q(L)\,P_{b'} = q(L)\,P_bP_{b'} = 0$: a spectral aggregator maps no energy from band b' into band b, and $P_b\,q(L)X = q(L)\,P_bX$ depends on X only through P_bX. Since Step 4 does not use Γ, it applies with g replaced by q and $Y = X$, giving $E_b\big(q(L)X\big) = \sum_{k\in\mathcal{I}_b} q(\lambda_k)^2\,\|u_k^\top X\|_2^2$. For a general $A(x) \in \mathbb{R}^{n\times n}$, the cross-band blocks $P_bA(x)P_{b'}$, $b \ne b'$, need not vanish, and energy can move between bands. □

Remark (provenance). Parts (i) and (ii) are elementary consequences of Parseval's identity and of (12); we state them as a corollary because Step 7 combines them with Proposition 4.1. Their use lies in the quantifier: (ii) holds for *every* admissible filter, which makes it an impossibility statement about the composition of aggregation and filter.

Remark (multi-branch aggregators). Since (i) and (ii) hold for every signal Y, they apply to the output $H' = XW_I + SXW_S + S^\top XW_T + R(L)XW_R$ of Eq. (3), with the scalar gates absorbed into the projections. On undirected graphs $S = S^\top = P_{\mathrm{sym}}$, and $R(L) = Ur(\Lambda)U^\top$ is a spectral function of L, as is its Chebyshev approximation, which is a polynomial in L. Every branch operator is therefore spectral, and Step 8 gives $P_bH' = (P_bX)W_I + P_{\mathrm{sym}}(P_bX)(W_S+W_T) + R(L)(P_bX)W_R$: the band-$b$ component of H' depends on X only through P_bX, so the adaptive valley moves no energy between bands. Its ego branch keeps every band of X available to the filter: if $E_b(X) > 0$, then $P_bX \ne 0$, so $P_bXv \ne 0$ for some $v \in \mathbb{R}^d$, and with $W_I := v\,e_1^\top$ and $W_S = W_T = W_R = 0$ one obtains $E_b(H') = \|P_bXv\|_2^2 > 0$. Hence the architecture never forces $E_b(H') = 0$ on a band where $E_b(X) > 0$, whereas P_{sym} alone does so on every band consisting only of eigenvalues equal to 1.

Remark (scope). The corollary bounds the energy that reaches each band, not accuracy: energy in a band may be present without being linearly decodable. The partition $\{\mathcal{I}_b\}$ must be fixed before the band energies are inspected; chosen post hoc, the statement is void. The equality case of (i) is an idealisation: a ChebNet response of degree at most $K-1$ has constant magnitude on a band with more than $2(K-1)$ distinct eigenvalues only if it is constant, since $p^2 - \mu^2$ would have more roots than its degree; Step 5 bounds the deviation.

B GCN LAYER DESCRIPTIONS

In this part of the Appendix, we present the linear spectral downstream GCN layers that follow the aggregation stage introduced in Section 3 and are evaluated in Section 5.2. We adopt simplified implementations of the layers rather than detailed one-to-one full reproductions. This choice is made to avoid unnecessary architectural complexity and to emphasize the essential structural principles of the respective models. Consequently, not all implementations correspond exactly to their original paper versions in every technical detail.

B.1 APPNP (GASTEIGER, BOJCHEVSKI & GÜNNEMANN, 2019)

Mathematical Description:

APPNP combines a learnable feature transformation with a personalized PageRank-type diffusion process. First, an initial representation is computed as

$$\mathbf{H}^{(0)} = \mathbf{X}\mathbf{W}, \tag{15}$$

where $\mathbf{W}$ is a learnable weight matrix. Propagation is then performed iteratively as

$$\mathbf{H}^{(k+1)} = (1-\alpha)\mathbf{H}^{(0)} + \alpha\tilde{\mathbf{A}}\mathbf{H}^{(k)}, \qquad k = 0, \ldots, K-1, \tag{16}$$

with final output

$$\mathbf{H}^{(out)} = \mathrm{ReLU}(\mathbf{H}^{(K)}). \tag{17}$$

Here, $\tilde{\mathbf{A}}$ denotes the symmetrically normalized adjacency matrix

$$\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}. \tag{18}$$

This corresponds to a truncated personalized PageRank (PPR) diffusion. In a PageRank-style process, each node's representation is updated as a weighted combination of its own features and the features of its neighbors. The parameter α controls the relative weight of the propagated neighbor information versus the node's original features, analogous to the teleport probability in classical PageRank. Gasteiger et al. (2019)

Legend:

- $\mathbf{X} \in \mathbb{R}^{N\times d}$: Input node feature matrix
- $\mathbf{W} \in \mathbb{R}^{d\times d'}$: Learnable linear transformation
- $\mathbf{H}^{(k)} \in \mathbb{R}^{N\times d'}$: Representation after k propagation steps
- $\mathbf{A} \in \mathbb{R}^{N\times N}$: Adjacency matrix
- $\mathbf{D}$: Diagonal degree matrix
- $\tilde{\mathbf{A}}$: Symmetrically normalized adjacency matrix
- $\alpha \in (0, 1)$: Teleport (diffusion) parameter, controlling the weight of the initial node features
- K: Number of propagation steps

B.2 ARMA (BIANCHI, GRATTAROLA, LIVI & ALIPPI, 2022)

Mathematical Description:

The ARMA spectral layer performs a frequency-selective, edge-aware graph convolution by iteratively combining Laplacian-propagated features with transformed input features. First, a linear feature transformation is applied Bianchi et al. (2022):

$$\mathbf{H}^{(0)} = \mathbf{X}\mathbf{W}, \tag{19}$$

where $\mathbf{W}$ is a learnable weight matrix. Propagation is then performed iteratively over K steps:

$$\mathbf{H}^{(k+1)} = \alpha\mathbf{L}\mathbf{H}^{(k)} + (1-\alpha)\mathbf{H}^{(0)}, \qquad k = 0, \ldots, K-1, \tag{20}$$

with final output

$$\mathbf{H}^{(out)} = \mathrm{ReLU}(\mathbf{H}^{(K)}). \tag{21}$$

Here, $\mathbf{L}$ denotes the normalized Laplacian matrix of the graph.

Legend:

- $\mathbf{X} \in \mathbb{R}^{N\times d}$: Input node feature matrix
- $\mathbf{W} \in \mathbb{R}^{d\times d'}$: Learnable linear transformation
- $\mathbf{H}^{(k)} \in \mathbb{R}^{N\times d'}$: Representation after k propagation steps
- $\mathbf{L} \in \mathbb{R}^{N\times N}$: Normalized Laplacian matrix
- $\alpha \in (0, 1)$: Weighting between Laplacian propagation and original features
- K: Number of propagation iterations
- $\mathrm{ReLU}(\cdot)$: Rectified Linear Unit activation function

B.3 Gegenbauer Spectral Layer

Mathematical Description:

Our Gegenbauer layer constructs a spectral graph filter based on Gegenbauer polynomials of the graph Laplacian. It implements a polynomial spectral filter, where each polynomial order is associated with learnable feature mixing, allowing flexible aggregation across orders. This design is inspired by recent work on Gegenbauer-based spectral GNNs and GegenConv operators, which generalize Chebyshev filters using orthogonal Gegenbauer polynomials Castro-Correa et al. (2024); Göksu (2025). Let $\mathbf{L} \in \mathbb{R}^{N\times N}$ denote the (normalized) graph Laplacian and $\mathbf{H} \in \mathbb{R}^{N\times d}$ the input features.

The filter is defined as a truncated Gegenbauer polynomial expansion

$$g(\mathbf{L})\mathbf{H} = \sum_{k=0}^{K-1} \mathbf{C}_k^{(\alpha)}(\mathbf{L})\mathbf{H}\mathbf{W}_k, \tag{22}$$

where $\mathbf{W}_k \in \mathbb{R}^{d\times d}$ are learnable weight matrices and $\mathbf{C}_k^{(\alpha)}(\cdot)$ denotes the k-th order Gegenbauer polynomial with parameter $\alpha > 0$.

The polynomials are defined recursively as

$$\mathbf{C}_0^{(\alpha)}(\mathbf{L})\mathbf{H} = \mathbf{H}, \tag{23}$$

$$\mathbf{C}_1^{(\alpha)}(\mathbf{L})\mathbf{H} = -2\alpha\mathbf{L}\mathbf{H}, \tag{24}$$

$$\mathbf{C}_k^{(\alpha)}(\mathbf{L})\mathbf{H} = \frac{2(k+\alpha-1)}{k}\mathbf{L}\mathbf{C}_{k-1}^{(\alpha)}(\mathbf{L})\mathbf{H} - \frac{k+2\alpha-2}{k}\mathbf{C}_{k-2}^{(\alpha)}(\mathbf{L})\mathbf{H}, \tag{25}$$

where the sign on the $\mathbf{C}_1$ term reflects the implementation's use of $-\mathbf{L}$ (equivalently, $\hat{\mathbf{A}} - \mathbf{I}$) as the polynomial argument so that the first-order term combines with $\mathbf{C}_0$ as $(\mathbf{I} - 2\alpha\mathbf{L})\mathbf{H}$, which recovers the propagation operator $\hat{\mathbf{A}}\mathbf{H}$ in the restricted form below.

The final layer output is then

$$\mathbf{H}^{(out)} = \mathrm{ReLU}\left(\sum_{k=0}^{K-1} \mathbf{C}_k^{(\alpha)}(\mathbf{L})\mathbf{H}\mathbf{W}_k\right). \tag{26}$$

Thus, the model learns a polynomial spectral filter whose shape is controlled by the order K, the learnable matrices $\mathbf{W}_k$, and the Gegenbauer parameter α. Different choices of α interpolate between known polynomial bases (e.g., Chebyshev-type cases).

Legend:

- $\mathbf{H} \in \mathbb{R}^{N\times d}$: Input node feature matrix
- $\mathbf{L} \in \mathbb{R}^{N\times N}$: Graph Laplacian
- $\mathbf{W}_k \in \mathbb{R}^{d\times d}$: Learnable weight matrix for order k
- $\alpha > 0$: Gegenbauer parameter controlling spectral shape
- K: Polynomial order (number of spectral components)

C EXPERIMENTAL RESULTS

C.1 IMPLEMENTATION DETAILS

We provide additional implementation details as a complement to the Experimental Settings section in the main paper 5.1.

TRAINING PROTOCOL

All models are trained from scratch on each split, with hidden dimension 64, input dimension equal to the number of node features and output dimension equal to the number of classes. We use Adam (learning rate 3×10^{-3}, weight decay 5×10^{-4}) on the cross-entropy loss, with dropout 0.5 in the aggregation stage and in every downstream layer; on Tolokers, the only binary task, the loss is class-weighted by the inverse class ratio. Nodes without a label are excluded from all masks. All six graphs are treated as undirected; where the source provides directed edges, we symmetrise them before computing L, so that $S = S^\top = P_{\mathrm{sym}}$ in Eq. (3). We evaluate after every epoch and report the test score of the epoch with the best validation score; the test labels enter no selection at any point. Table 5 summarises the protocol of the three measurements.

Dataset	Split	Repeats	Metric	Selection
Squirrel-f	48/32/20, random	10	Accuracy	validation epoch
Chameleon-f	48/32/20, random	10	Accuracy	validation epoch
Crocodile	48/32/20, random	10	Accuracy	validation epoch
Actor	48/32/20, random	10	Accuracy	validation epoch
Tolokers	50/25/25, official	10	ROC-AUC	validation epoch
HM-categories	10/10/80, official	1 split, 3 seeds	Accuracy	validation epoch
Measurements				
Section 5.2 (ablation)	150 epochs	$K = 5$ on every graph, all repeats above		
Section 5.3 (benchmark)	1000 epochs	K per dataset (Table 6), all repeats above		
Section 6.2 (robustness, order K)	150 epochs	$K = 5$ or swept; 3 repeats, 1 on HM-categories		

Table 5: Evaluation protocol. Splits are stored once and reused across all models and measurements. The epoch is always selected on the validation set.

For SpecValley the validation set is split in half once per run: the operator parameters $\theta = (c, p, a_L, a_R, \tau)$ are trained on half A by a second Adam without weight decay, while the epoch is selected on half B, which never enters the spectral fit. All other parameters follow the training loss. The spectral branch $R(L)$ uses a cached eigendecomposition of L for graphs with at most $14{,}000$ nodes and a Chebyshev expansion of degree 40 in $L - I$ above, which applies to HM-categories only. All experiments run on a single NVIDIA RTX 3060 (12 GB) with 32 GB system RAM.

	Squirrel-f	Chameleon-f	Crocodile	Actor	Tolokers	HM-cat.
ChebNet						
Selected K	4	2	40	2	8	25
Score	40.37	40.05	75.72	36.86	85.88	65.78
Score at K=5	38.96	35.63	72.56	35.66	85.57	56.90
Δ	+1.41	+4.42	+3.16	+1.20	+0.31	+8.88
SpecValley						
Selected K	5	20	25	4	8	40
Score	45.34	43.42	77.13	36.16	85.67	67.32
Score at K=5	45.34	42.56	76.29	35.92	85.49	63.04
Δ	0.00	+0.86	+0.84	+0.24	+0.18	+4.28

Table 6: Polynomial order K of the ChebNet layer in the benchmark, for ChebNet alone and for SpecValley, from a 150-epoch sweep. Scores are test accuracy (ROC-AUC on Tolokers) at K selected on the validation score and at $K = 5$, taken at the epoch with the best validation score and averaged over the sweep splits; Δ is the difference between the two.

Choice of the Polynomial Order K for the Benchmark

For the benchmark in Section 5.3, we set the order K of the downstream ChebNet layer per dataset, since performance varies strongly with K across graphs (Section 6.2, Figure 3). K is selected in a preliminary sweep over $K \in \{0, 1, 2, 3, 4, 5, 8, 10, 12, 15, 20, 25, 40, 60\}$, training ChebNet alone and SpecValley with ChebNet for 150 epochs per order on three splits (one on HM-categories). $K = 0$ applies no filter and $K = 1$ does not use the graph. For each model and graph we take the order with the highest mean validation score. The test scores in Table 6 are reported for information only and enter no selection.

Runtime

Table 7 reports the cost of SpecValley under the protocol of Appendix C.1, once with $K = 5$ on every graph as in the ablation of Section 5.2, and once with the per-dataset order of Table 6 as in the benchmark of Section 5.3. The eigendecomposition of L is computed once per graph and cached; it is not part of the per-epoch times.

Two regimes are visible, and they follow the complexity of Section 4.1. On the five graphs with at most $14{,}000$ nodes the spectral branch runs on the cached eigenbasis at $O(n^2 d')$ per pass, and an epoch costs a fraction of a second. HM-categories falls on the other side of the threshold and uses the degree-40 Chebyshev expansion at $O(Msd')$, where its 21.5 million edges dominate: one epoch takes 15.8 seconds, about 90 times Tolokers at four times the node count. Raising K is comparatively cheap, because the input projection and the aggregation stage do not depend on it: five times the order costs $2.3\times$ the epoch on Crocodile and $1.7\times$ on HM-categories. A full split takes 41 minutes for one 150-epoch ablation split and 7.7 hours for one 1000-epoch benchmark split, of which HM-categories accounts for more than 95%.

Dataset	Nodes	K	Time/epoch (s)	Split @ 150 ep.	Split @ 1000 ep.
Ablation setting: $K = 5$ on every graph					
Chameleon-f	890	5	0.073	11.1 s	73.4 s
Squirrel-f	2,223	5	0.084	12.6 s	84.1 s
Actor	7,600	5	0.073	10.9 s	72.7 s
Crocodile	11,631	5	0.216	32.6 s	3.6 min
Tolokers	11,758	5	0.176	26.4 s	2.9 min
HM-categories	46,563	5	15.76	39.4 min	4.4 h
All six, one split				41.0 min	4.6 h
Benchmark setting: K per dataset					
Chameleon-f	890	20	0.199	29.9 s	3.3 min
Squirrel-f	2,223	5	0.074	11.2 s	74.3 s
Actor	7,600	4	0.064	9.7 s	64.3 s
Crocodile	11,631	25	0.497	74.7 s	8.3 min
Tolokers	11,758	8	0.240	36.0 s	4.0 min
HM-categories	46,563	40	26.53	66.3 min	7.4 h
All six, one split				69.0 min	7.7 h

Table 7: Training time of SpecValley on a single NVIDIA RTX 3060 (12 GB). The ablation trains for 150 epochs with $K = 5$, the benchmark for 1000 epochs with the per-dataset order of Table 6; the eigendecomposition is computed once per graph and excluded.

C.2 Spectral Valley Aggregation for all Datasets

Figure 4 extends Figure 2 to all six graphs, with $\hat{e}_k$ as defined in Section 6.1 and $L = U\Lambda U^\top$ as in Appendix A.1. The solid curve reports $n\,\hat{e}_k$, averaged over a sliding window of m modes, so that 1 is the level of a labelling uncorrelated with the graph; the dashed curve is the cumulative share $\sum_{\lambda_j \le \lambda} \hat{e}_j$. The grey area is the eigenvalue density, normalised so that 1 corresponds to the uniform distribution on $[0, 2]$, and the light band marks modes with $n\,\hat{e}_k < 1 - z\sqrt{2/((C-1)m)}$, a one-sided test at $z = 1.645$ with C the number of classes.

The lower panel shows what the two aggregators do with this spectrum. Degree normalisation acts

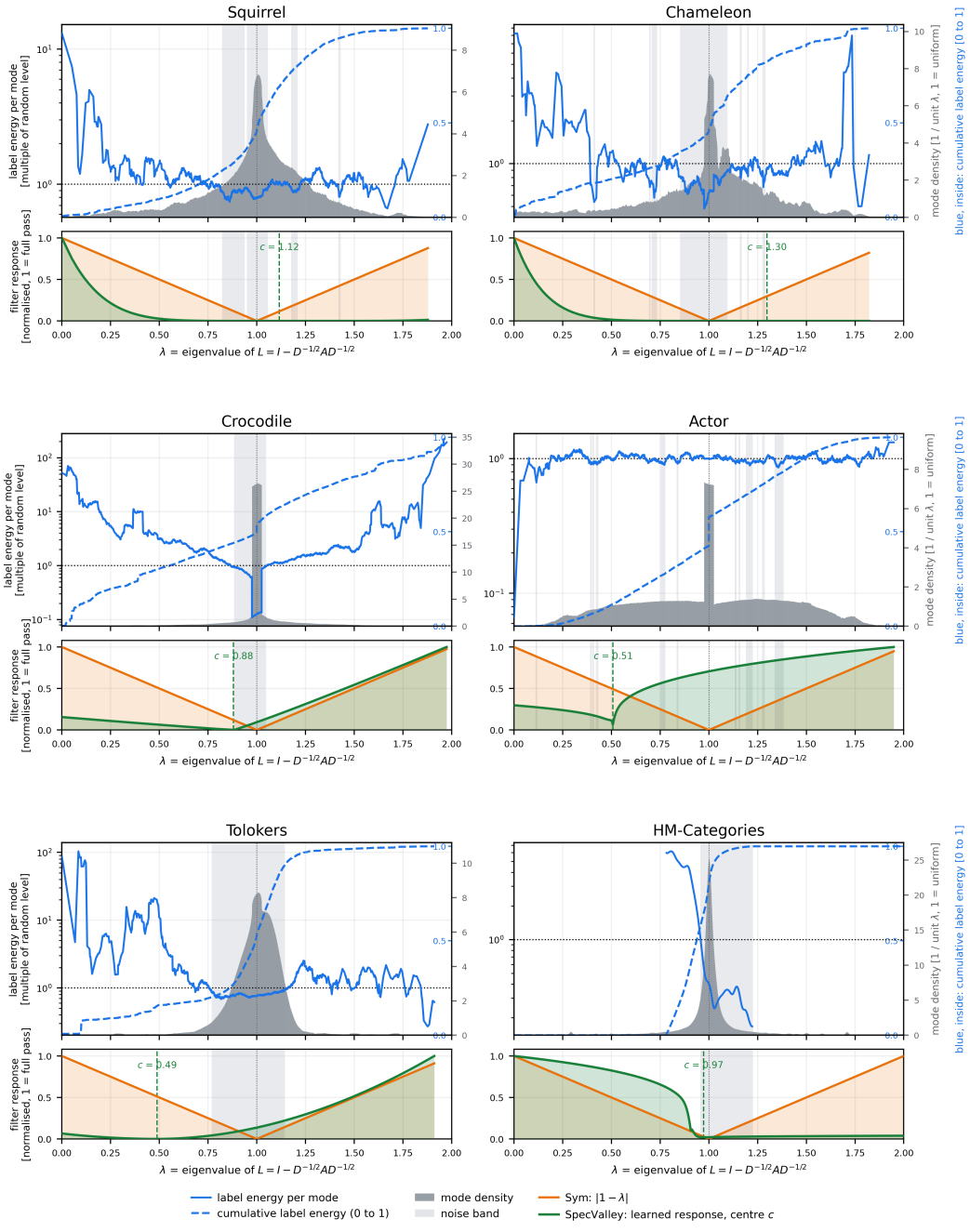

Figure 4: Label energy across the Laplacian spectrum (top) and magnitude of the fixed and learned aggregation responses (bottom) for all six graphs.

as the spectral function $1 - \lambda$ (Proposition 4.1), a valley whose zero is pinned at $\lambda = 1$ by the normalisation, not by the data. SpecValley replaces it by the learned response $r(\lambda)$ of Eq. (4), with centre c, sharpness $p > 0$ and flank gains a_L, a_R; training starts at $|1 - \lambda|$ ($c = p = 1$, $a_L = a_R = 1$). We plot the magnitude of each response normalised to its maximum, which turns the zero into a visible valley and $|1 - \lambda|$ into a V with its vertex at $\lambda = 1$. Two consequences of the normalisation should be kept in mind: a large p makes the response hug the axis over a wide range even though it is not zero there, and a strong asymmetry $a_L \gg |a_R|$ flattens the weaker flank relative to the stronger one.

Reading the six graphs. The density panels show the accumulation of Proposition 4.1: on Crocodile, Tolokers and HM-categories 76%, 66% and 79% of the modes lie in $|\lambda - 1| < 0.1$, on Squirrel and Chameleon the density forms a broader hill (41%, 40%), and on the sparser Actor graph (27%) it is weakest. The graphs differ in where the label energy lies relative to this core. On Crocodile, Squirrel and Chameleon the core carries little of it (17% on Crocodile), so $1 - \lambda$ removes mainly modes at or below the random level. On Tolokers the core carries 52% of the label energy, so $1 - \lambda$ removes label information together with the noise. On HM-categories the label energy concentrates in a narrow band just below $\lambda = 1$ and falls below the random level above it; on Actor it is flat at the random level. The learned responses follow these differences: on Crocodile the valley stays close to $|1 - \lambda|$, on Tolokers its centre moves away from $\lambda = 1$, and on HM-categories it passes this band just below $\lambda = 1$ and strongly attenuates everything above it. The figure is descriptive: it shows where the label energy lies and how the learned response places itself relative to it, but no single spectral statistic predicts the benefit of aggregation on all six graphs (Section 6.1).

C.3 Polynomial Order of the Downstream Filter

As discussed in Section 6.2, we sweep the order $K \in \{0, \ldots, 5\} \cup \{8, 10, 12, 15, 20, 25, 40, 60\}$ of the downstream ChebNet layer (Eq. (5)), once with the valley aggregator of Eq. (3) in front of it and once without, under the protocol of Section 5.1 with 150 epochs per order. Figure 5 shows the two curves per graph, grouped into two panels for legibility.

On the graphs where the valley is effective (Squirrel-f, Chameleon-f, Crocodile and HM-categories) the gap does not close with K: the filter alone does not reproduce what the aggregator supplies, even at orders far beyond those used in the benchmark. The composition of aggregator and filter is a polynomial of only one order higher (Section 4.4), so this is not a question of expressive capacity but of what the two parameterisations reach in training. On Tolokers and Actor the two curves coincide from moderate orders onward, consistent with a spectrum in which no aggregator finds a band to attenuate (Section 6.1); there the aggregator contributes only at very low orders, where the filter is too constrained to use the graph at all.

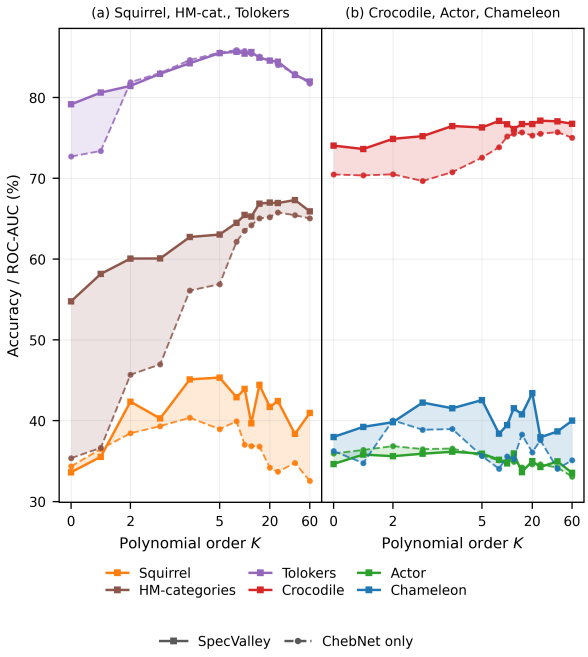

Figure 5: Polynomial order K of the ChebNet layer, with and without the valley aggregator. Test score at the epoch with the best validation score, averaged over three splits (one on HM-categories), 150 epochs per order.