A Details on Spectral Filters

A.1 Fixed Filter

Linear. A layer of **GCN** [51] propagation $f(\tilde{\mathbf{A}}) = \mathbf{I} + \tilde{\mathbf{A}}$ is equivalent to a single-hop of spectral convolution. Recall that $\tilde{\mathbf{L}} = \mathbf{I} - \tilde{\mathbf{A}}$, the filter can be expressed as:

$$g(\tilde{\mathbf{L}}) = 2\mathbf{I} - \tilde{\mathbf{L}}.$$

Impulse. **SGC** [108] and **gfNN** [82] adopt a pre-propagation decoupled architecture, while **GZoom** [22] applies post-propagation to achieve expansion of the closed-form filter $(\mathbf{I} + \tilde{\mathbf{L}})^{-1}$. All these models result in a J-hop spatial diffusion operation as $f(\mathbf{A}) = \tilde{\mathbf{A}}^J$. By respectively examining bases $T^{(k)}(\tilde{\mathbf{L}}) = (\mathbf{I} - \tilde{\mathbf{L}})^k$ and $T^{(k)}(\tilde{\mathbf{L}}) = \tilde{\mathbf{L}}^k$, we have two equivalent formulations of the filter:

$$g(\tilde{\mathbf{L}}) = (\mathbf{I} - \tilde{\mathbf{L}})^K, \quad T^{(k)}(\tilde{\mathbf{L}}) = (\mathbf{I} - \tilde{\mathbf{L}})^k, \quad \theta_0 = \theta_1 = \cdots = \theta_{K-1} = 0,\ \theta_K = 1; \text{ or}$$

$$g(\tilde{\mathbf{L}}) = \sum_{k=0}^{K} \theta_k \tilde{\mathbf{L}}^k, \quad T^{(k)}(\tilde{\mathbf{L}}) = \tilde{\mathbf{L}}^k, \quad \theta_k = \binom{K}{k}(-1)^k,$$

where $\binom{K}{k}$ is the binomial coefficient.

Monomial. $\mathbf{S}^2$**GC** [130] summarizes K-hop propagation results with uniform weights in decouple precomputation, which is classifies as the monomial propagation $f(\mathbf{A}) = \sum_{j=1}^{J} \xi_j \tilde{\mathbf{A}}^j$, where $\xi_j = 1/(J+1)$. **GRAND+** [32] examines an approximate propagation to acquire the filter. Similarly, there are two commonly used spectral interpretations based on two bases:

$$g(\tilde{\mathbf{L}}) = \sum_{k=0}^{K} \theta_k (\mathbf{I} - \tilde{\mathbf{L}})^k, \quad T^{(k)}(\tilde{\mathbf{L}}) = (\mathbf{I} - \tilde{\mathbf{L}})^k, \quad \theta_0 = \cdots = \theta_K = \frac{1}{K+1}; \text{ or}$$

$$g(\tilde{\mathbf{L}}) = \sum_{k=0}^{K} \theta_k \tilde{\mathbf{L}}^k, \quad T^{(k)}(\tilde{\mathbf{L}}) = \tilde{\mathbf{L}}^k, \quad \theta_k = \frac{1}{K+1} \sum_{j=k}^{J} \binom{j}{k}(-1)^k,$$

Personalized PageRank (PPR). **GLP** [58] derives a closed-form $\hat{f}(\mathbf{A}) = (\mathbf{I} + \alpha\mathbf{L})^{-1}$ from the auto regressive (AR) filter [97], while **PPNP** [52] solves PPR [84] as $\hat{f}(\tilde{\mathbf{A}}) = \alpha\big(\mathbf{I} + (1-\alpha)\tilde{\mathbf{A}}\big)^{-1}$. These two graph processing techniques are equivalent in essence. $\alpha \in [0, 1]$ is the coefficient for balancing the strength of neighbor propagation, that a larger α results in stronger node identity and weaker neighboring impact, and vice versa. In both works, the filter is approximated by a recursive calculation $\mathbf{H}^{(j+1)} = \varphi((1-\alpha)\tilde{\mathbf{A}}\mathbf{H}^{(j)} + \alpha\mathbf{H}^{(0)})$, which is widely accepted in later studies such as **GCNII** [81]. The explicit spatial and spectral interpretations of the polynomial approximation are respectively:

$$f(\tilde{\mathbf{A}}) = \sum_{j=0}^{J} \alpha(1-\alpha)^j \tilde{\mathbf{A}}^j; \quad g(\tilde{\mathbf{L}}) = \sum_{k=0}^{K} \theta_k (\mathbf{I} - \tilde{\mathbf{L}})^k, \quad \theta_k = \alpha(1-\alpha)^k.$$

In addition, approximate computations have been proposed by representative works including GDC [37], PPRGo [9], GRAND+ [32]. GBP [14] and **AGP** [100] explored the adjacency aggregation under graph normalization.

Heat Kernel (HK). **GDC** [37] inspects the heat kernel PageRank (HKPR) [20] replacing the PPR calculation by an exponential parameter. Equivalent HK filters are also studied in DGC [105] and AGP [100]. Let $\alpha > 0$ be the temperature coefficient, the filter $\hat{f}(\tilde{\mathbf{A}}) = e^{-\alpha\tilde{\mathbf{L}}}$ is expanded in spatial and spectral forms as:

$$f(\tilde{\mathbf{A}}) = \sum_{j=0}^{J} \frac{e^{-\alpha}\alpha^j}{j!} \tilde{\mathbf{A}}^j; \quad g(\tilde{\mathbf{L}}) = \sum_{k=0}^{K} \theta_k (\mathbf{I} - \tilde{\mathbf{L}})^k, \quad \theta_k = \frac{e^{-\alpha}\alpha^k}{k!}.$$

Gaussian. $\mathbf{G}^2$**CN** [55] uses the Gaussian filter [11] for better flexibility on capturing local information. When concentrating on low frequency, the closed-form propagation is $\hat{f}(\tilde{\mathbf{A}}) = e^{-\alpha(2\mathbf{I} - \tilde{\mathbf{L}})}$. Invoking Taylor expansion for the filter leads to:

$$f^{(j)}(\tilde{\mathbf{A}}) = \mathbf{I} - \frac{\alpha}{J}(2\mathbf{I} - \tilde{\mathbf{L}}); \quad g(\tilde{\mathbf{L}}) = \sum_{k=0}^{K} \theta_k (2\mathbf{I} - \tilde{\mathbf{L}})^k, \quad \theta_k = \frac{\alpha^k}{k!}.$$

A.2 Variable Filter

Linear. **GIN** [110] alters the iterative ridged adjacency propagation with a learnable scaling parameter $\theta > 0$ controlling the strength of self loops, i.e., skip connections. It proves that its propagation $f(\tilde{\mathbf{A}}) = (1 + \xi)\mathbf{I} + \tilde{\mathbf{A}}$ is more expressive with regard to the Weisfeiler-Lehman (WL) test. **AKGNN** [50] elaborates the expression in spectral domain, that the scaling parameter adaptively balances the threshold between high and low frequency. We thereby obtain the linear spectral function for each layer as:

$$g(\tilde{\mathbf{L}}; \theta) = (1 + \theta)\mathbf{I} - \tilde{\mathbf{L}}.$$

Monomial. **DAGNN** [71] studies the scheme of assigning learnable parameters to each hop of the propagation, but implements a costly concatenation-based scheme. **GPRGNN** [19] considers the iterative generalized PageRank computation [56] under heterophily, which also produces the same variable monomial spectral filter formulated as $f(\mathbf{A}) = \sum_{j=0}^{J} \xi_j \tilde{\mathbf{A}}^j$. We present two bases and corresponding relationship between spatial and spectral parameters when $J = K$:

$$g(\tilde{\mathbf{L}}; \theta) = \sum_{k=0}^{K} \theta_k (\mathbf{I} - \tilde{\mathbf{L}})^k, \quad T^{(k)}(\tilde{\mathbf{L}}) = (\mathbf{I} - \tilde{\mathbf{L}})^k, \quad \theta_k = \xi_j; \text{ or}$$

$$g(\tilde{\mathbf{L}}; \theta) = \sum_{k=0}^{K} \theta_k \tilde{\mathbf{L}}^k, \quad T^{(k)}(\tilde{\mathbf{L}}) = \tilde{\mathbf{L}}^k, \quad \theta_k = \sum_{j=k}^{J} \binom{j}{k}(-1)^k \xi_j.$$

[19] also inspects the effect of initialization of the parameters ξ_j, θ_k on fitting heterophilous graph signals.

Horner. Horner's method is a recursive algorithm to compute the summation of monomial bases with residual connections. Consider adding a residual term in the layer-wise propagation corresponding to the Monomial filter $\mathbf{H}^{(j+1)} = \varphi(\tilde{\mathbf{A}}\mathbf{H}^{(j)} + \xi_j\mathbf{H}^{(0)})$. When the

balancing parameter is fixed $\xi_j = \alpha/(1-\alpha)$, it is equivalent to the PPR computation in Appendix A.1. When ξ_j is variable, the model is regarded as **HornerGCN** which is introduced by [42]. Alternatively, **ARMAGNN** [6] utilizes the Auto Regressive Moving Average (ARMA) filter $\hat{f}(\mathbf{A}) = \beta(\mathbf{I} - \alpha\tilde{\mathbf{A}})^{-1}$ [97] as an approach to describe the residual connection learned by respective weights in iterative architecture. Their spectral filter is:

$$g(\tilde{\mathbf{L}};\theta) = \sum_{k=0}^{K} \theta_k \, (\mathbf{I} - \tilde{\mathbf{L}})^k, \quad \theta_k = \xi_{K-k}.$$

Although it shares an identical spectral interpretation with the Monomial filter, the explicit residual connection proves beneficial in guiding the learnable parameters to recognize node identity and alleviate over-smoothing throughout propagation.

Chebyshev. The Chebyshev basis is widely accepted for graph signal processing, which is powerful in producing a minimax polynomial approximation for the analytic functions [44, 93]. **ChebNet** [21] utilizes it to replace the adjacency propagation in iterative network architecture so that $f^{(j)}(\tilde{\mathbf{A}}) = T^{(j)}(\mathbf{I} - \tilde{\mathbf{A}})$. To adapt the basis to decoupled propagation with explicit variable parameters, [46] proposes **ChebBase**. The spectral expressiveness of these two models are the same, and the filter is:

$$g(\tilde{\mathbf{L}};\theta) = \sum_{k=0}^{K} \theta_k T^{(k)}(\tilde{\mathbf{L}}), \quad T^{(k)}(\tilde{\mathbf{L}}) = 2\tilde{\mathbf{L}}T^{(k-1)}(\tilde{\mathbf{L}}) - T^{(k-2)}(\tilde{\mathbf{L}}),$$
$$T^{(1)}(\tilde{\mathbf{L}}) = \tilde{\mathbf{L}}, \; T^{(0)}(\tilde{\mathbf{L}}) = \mathbf{I}.$$

The Chebyshev polynomial is expressed in the three-term recurrence relation, which is favorable for the GNN iterative propagation. One can also write the Chebyshev basis of the first kind as the closed-form expression $T^{(k)}(\lambda) = \cos(k \arccos \lambda)$.

Chebyshev Interpolation (ChebInterp). **ChebNetII** [46] utilizes Chebyshev interpolation [39] to modify the Chebyshev filter parameter for better approximation with generally decaying weights. For each Chebyshev basis $T^{(k)}(\tilde{\mathbf{L}})$, it appends the basis with K-order interpolation:

$$g(\tilde{\mathbf{L}};\theta) = \frac{2}{K+1} \sum_{k=0}^{K} \sum_{\kappa=0}^{K} \theta_\kappa T^{(k)}(x_\kappa) \, T^{(k)}(\tilde{\mathbf{L}}), \quad x_\kappa = \cos\left(\frac{\kappa + 1/2}{K+1}\pi\right),$$

where $T^{(k)}(x_\kappa), T^{(k)}(\tilde{\mathbf{L}})$ follow the Chebyshev basis, and x_κ are the Chebyshev nodes of $T^{(K+1)}$.

Clenshaw. Similar to Horner's method, **ClenshawGCN** [42] applies Clenshaw algorithm on top of the Chebyshev filter to incorporate explicit residual connections. Its spatial convolution is obtained as $\mathbf{H}^{(j+1)} = \varphi\big(2\tilde{\mathbf{A}}\mathbf{H}^{(j)} - \mathbf{H}^{(j-1)} + \xi_j \mathbf{H}^{(0)}\big), \mathbf{H}^{(-1)} = \mathbf{H}^{(-2)} = \mathbf{O}$. The form of spectral filter is related with Chebyshev polynomials of the second kind:

$$g(\tilde{\mathbf{L}};\theta) = \sum_{k=0}^{K} \theta_k T^{(k)}(\tilde{\mathbf{L}}), \quad T^{(k)}(\tilde{\mathbf{L}}) = 2\tilde{\mathbf{L}}T^{(k-1)}(\tilde{\mathbf{L}}) - T^{(k-2)}(\tilde{\mathbf{L}}),$$
$$T^{(1)}(\tilde{\mathbf{L}}) = 2\tilde{\mathbf{L}}, \; T^{(0)}(\tilde{\mathbf{L}}) = \mathbf{I}.$$

Alternatively, the closed-form definition of the Chebyshev basis of the second kind is $T^{(k)}(\cos\lambda) = \frac{\sin((k+1)\lambda)}{\sin\lambda}$. The relation to spatial parameters is $\theta_k = \xi_{K-k}$.

Bernstein. BernNet [45] pursues more interpretable spectral filters by by Bernstein polynomial approximation [29] and invokes constraints form prior knowledge to avoid ill-posed variable parameters. The spatial propagation is special as it applies two graph matrices instead of one. The filter with regard to Bernstein basis $T^{(k)}$ is:

$$g(\tilde{\mathbf{L}};\theta) = \sum_{k=0}^{K} \frac{\theta_k}{2^K} T^{(k)}(\tilde{\mathbf{L}}), \quad T^{(k)}(\tilde{\mathbf{L}}) = \binom{K}{k} (2\mathbf{I} - \tilde{\mathbf{L}})^{K-k} \, \tilde{\mathbf{L}}^k,$$

where learnable parameters are initialized as $\theta_k = T^{(k)}(k/K)$.

Legendre. LegendreNet [13] exploits the Legendre polynomials in an accumulation form of calculation similar to BernNet:

$$g(\tilde{\mathbf{L}};\theta) = \sum_{k=0}^{K} \theta_k T^{(k)}(\tilde{\mathbf{L}}), \quad T^{(k)}(\tilde{\mathbf{L}}) = \frac{(-1)^k}{k!\binom{2k}{k}} \, (2\mathbf{I} - \tilde{\mathbf{L}})^k \, \tilde{\mathbf{L}}^k.$$

Based on the relation of Bernstein bases and Legendre polynomials [28], we can also express it in the recurrence form:

$$g(\tilde{\mathbf{L}};\theta) = \sum_{k=0}^{K} \theta_k T^{(k)}(\tilde{\mathbf{L}}), \quad T^{(0)}(\tilde{\mathbf{L}}) = \mathbf{I}, \; T^{(1)}(\tilde{\mathbf{L}}) = \tilde{\mathbf{L}},$$
$$T^{(k)}(\tilde{\mathbf{L}}) = \frac{2k-1}{k}\tilde{\mathbf{L}}T^{(k-1)}(\tilde{\mathbf{L}}) - \frac{k-1}{k}T^{(k-2)}(\tilde{\mathbf{L}}).$$

Jacobi. JacobiConv [103] utilizes the more general Jacobi basis, whereas Chebyshev and Legendre polynomials can be regarded as special cases. Intuitively, it provides more flexible weight functions with two hyperparameters α, β to adapt to different signals of the spectral graphs. Each polynomial term of JacobiConv can be formulated by the three-term recurrence as following:

$$g(\tilde{\mathbf{L}};\theta) = \sum_{k=0}^{K} \theta_k T^{(k)}(\tilde{\mathbf{L}}), \quad T^{(0)}(\tilde{\mathbf{L}}) = \mathbf{I},$$
$$T^{(1)}(\tilde{\mathbf{L}}) = \frac{\alpha-\beta}{2}\mathbf{I} + \frac{\alpha+\beta+2}{2}(\mathbf{I} - \tilde{\mathbf{L}}),$$
$$T^{(k)}(\tilde{\mathbf{L}}) = \delta_k(\mathbf{I} - \tilde{\mathbf{L}})T^{(k-1)}(\tilde{\mathbf{L}}) + \delta'_k \, T^{(k-1)}(\tilde{\mathbf{L}}) - \delta''_k \, T^{(k-2)}(\tilde{\mathbf{L}}),$$

where

$$\delta_k = \frac{(2k+\alpha+\beta)(2k+\alpha+\beta-1)}{2k(k+\alpha+\beta)}, \; \delta'_k = \frac{(2k+\alpha+\beta-1)(\alpha^2-\beta^2)}{2k(k+\alpha+\beta)(2k+\alpha+\beta-2)},$$
$$\delta''_k = \frac{(k+\alpha-1)(k+\beta-1)(2k+\alpha+\beta)}{k(k+\alpha+\beta)(2k+\alpha+\beta-2)} \text{ for } k \geq 2.$$

Favard. FavardGNN [43] exploits the Favard's Theorem [38] to learn the polynomial basis from available space and ensure orthonormality. It is achieved by a three-term recurrence form with multiple series of hop-dependent variable parameters θ, α, β:

$$g(\tilde{\mathbf{L}};\theta) = \sum_{k=0}^{K} \theta_k T^{(k)}(\tilde{\mathbf{L}}), \; T^{(-1)}(\tilde{\mathbf{L}}) = \mathbf{O}, \; T^{(0)}(\tilde{\mathbf{L}}) = \frac{1}{\sqrt{\alpha_0}}\mathbf{I},$$
$$T^{(k)}(\tilde{\mathbf{L}}) = \frac{1}{\sqrt{\alpha_k}}\left((\mathbf{I} - \tilde{\mathbf{L}})T^{(k-1)}(\tilde{\mathbf{L}}) - \beta_k T^{(k-1)}(\tilde{\mathbf{L}}) - \sqrt{\alpha_{k-1}}\, T^{(k-2)}(\tilde{\mathbf{L}})\right).$$

OptBasis. OptBasisGNN [43] considers a basis to be optimal with regard to convergence rate in the graph signal denoising problem. By replacing the learnable parameters in FavardGNN with

parameters derived from the current input signal $\mathbf{h}^{(k)}$, it can approach the optimal basis without occurring additional overhead.

$$g(\tilde{\mathbf{L}};\theta)=\sum_{k=0}^{K}\theta_k T^{(k)}(\tilde{\mathbf{L}}),\quad T^{(-1)}(\tilde{\mathbf{L}})=\mathbf{O},\ T^{(0)}(\tilde{\mathbf{L}})=\frac{1}{\|\mathbf{h}^{(0)}\|}\mathbf{I},$$

$$T^{(k)}(\tilde{\mathbf{L}})=\frac{1}{\|\mathbf{h}^{(k)}\|}\Big((\mathbf{I}-\tilde{\mathbf{L}})T^{(k-1)}(\tilde{\mathbf{L}})\ -\beta_{k-1}T^{(k-1)}(\tilde{\mathbf{L}})\ -\|\mathbf{h}^{(k-1)}\|T^{(k-2)}(\tilde{\mathbf{L}})\Big),$$

$$\beta_{k-1}=\langle(\mathbf{I}-\tilde{\mathbf{L}})\mathbf{h}^{(k-1)},\mathbf{h}^{(k-1)}\rangle.$$

A.3 Filter Bank

AdaGNN. AdaGNN [24] designs $Q=F$ adaptive filters by assigning feature-specific parameters to the Linear filter $g_q(\tilde{\mathbf{L}})=\mathbf{I}-\gamma_q\tilde{\mathbf{L}}, 1\le q\le F$. The representation update is performed in an iterative manner as $\mathbf{H}^{(j+1)}=\mathbf{H}^{(j)}-\tilde{\mathbf{L}}\mathbf{H}^{(j)}\mathbf{\Gamma}^{(j)}$, where $\mathbf{\Gamma}^{(j)}=\mathrm{diag}(\gamma_1^{(j)},...,\gamma_F^{(j)})$ containing the learnable feature-wise parameter for the j-th layer. Hence, considering a single layer, the corresponding aggregated filter in spectral domain is:

$$\mathscr{g}(\tilde{\mathbf{L}};\gamma)=\Big\|_{q=1}^{F}(\mathbf{I}-\gamma_q\tilde{\mathbf{L}}),$$

where each filter $g_q(\tilde{\mathbf{L}})$ is only applied to the q-th feature, and $\|$ is the concatenation operator that combines filtering result tensors among all features.

FBGNN. FBGNN [76] introduces the concept of filter bank for combining multiple filters [104] in spectral GNN under the context of graph heterophily. It designs a two-channel scheme using graph adjacency $T_1=\tilde{\mathbf{A}}$ as Linear low-pass filter (LP) and Laplacian $T_2=\tilde{\mathbf{L}}$ for Linear high-pass filter (HP) to learn the smooth and non-smooth components together. FBGNN adopts an iterative form with scalar parameters $\gamma_1,\gamma_2\in[0,1]$ for weighted sum. By omitting components including transformation weights and non-linear activation functions, we give the equivalent spectral function for each layer as:

$$\mathscr{g}(\tilde{\mathbf{L}};\gamma)=\gamma_1(\mathbf{I}-\tilde{\mathbf{L}})+\gamma_2\tilde{\mathbf{L}}.$$

ACMGNN. ACMGNN [75] extends FBGNN to three filters with the additional identity (ID) diffusion matrix $T_3=\mathbf{I}$, which corresponds to an all-pass filter maintaining node identity throughout propagation. It has two variants of applying different relative order between transformation and propagation, which are denoted as ACMGNN-I and ACMGNN-II. Both of their single-layer spectral expressions can be simplified as:

$$\mathscr{g}(\tilde{\mathbf{L}};\gamma)=\gamma_1(\mathbf{I}-\tilde{\mathbf{L}})+\gamma_2\tilde{\mathbf{L}}+\gamma_3\mathbf{I}.$$

FAGCN. FAGCN [8] combines two Linear filters with bias for capturing low- and high-frequency signals as $T_1=(\beta+1)\mathbf{I}-\tilde{\mathbf{L}}$ and $T_2=(\beta-1)\mathbf{I}+\tilde{\mathbf{L}}$, where $\beta\in[0,1]$ is the scaling coefficient. Since both filters are linear to $\tilde{\mathbf{L}}$, the fused representation of each layer can be computed using only one propagation by employing attention mechanism. We generally write the channel-wise parameters as $\gamma_1,\gamma_2\in[0,1]$ and $\gamma_1+\gamma_2=1$. Then the spectral expression for one layer is:

$$\mathscr{g}(\tilde{\mathbf{L}};\gamma)=\gamma_1((\beta+1)\mathbf{I}-\tilde{\mathbf{L}})+\gamma_2((\beta-1)\mathbf{I}+\tilde{\mathbf{L}}).$$

G^{2}CN. G^{2}CN [55] derives the Gaussian filters for high- and low-frequency concentration centers. Specifically, it adopts 2-hop propagation in each layer. Utilizing the decay coefficient $\alpha\in[0,1]$ and the scaling coefficient $\beta\in[0,1]$, we rewrite the layer-wise propagation as $f^{(j)}(\tilde{\mathbf{A}})=\mathbf{I}-\alpha((1\pm\beta)\mathbf{I}-\tilde{\mathbf{L}})^2/J$. The model integrates the $Q=2$ channels after the decoupled propagation, hence:

$$\mathscr{g}(\tilde{\mathbf{L}};\gamma)=\gamma_1\sum_{k=0}^{\lfloor K/2\rfloor}\theta_{1,k}T_1^{(k)}+\gamma_2\sum_{k=0}^{\lfloor K/2\rfloor}\theta_{2,k}T_2^{(k)},$$

$$T_1^{(k)}=((1+\beta_1)\mathbf{I}-\tilde{\mathbf{L}})^{2k},\quad \theta_{1,k}=\frac{\alpha_1^k}{k!},$$

$$T_2^{(k)}=((1-\beta_2)\mathbf{I}-\tilde{\mathbf{L}})^{2k},\quad \theta_{2,k}=\frac{\alpha_2^k}{k!}.$$

GNN-LF/HF. GNN-LF/HF [131] proposes a pair of generalized GNN propagations based on low- and high-passing filtering (LF/HF) on top of its unified optimization framework depicting a range of GNN designs. Intuitively, its filter formulation is similar to the PPR scheme, except a $(\mathbf{I}\pm\beta\tilde{\mathbf{L}})$ factor applying to the input signal for distinguishing low- and high-frequency components. We transform the original dual filters into one model with $Q=2$ channels by learning the balancing coefficients $\gamma_1,\gamma_2\in[0,1]$ which adjust the relative strength between node identity and low/high frequency features. Consequently, the channels can be implemented in a shared adjacency-based propagation. We formulate the filter bank version of GNN-LF/HF as:

$$\mathscr{g}(\tilde{\mathbf{L}};\gamma)=\gamma_1\sum_{k=0}^{K}\theta_{1,k}T_1^{(k)}+\gamma_2\sum_{k=0}^{K}\theta_{2,k}T_2^{(k)},$$

$$T_1^{(k)}=(\mathbf{I}-\beta_1\tilde{\mathbf{L}})(\mathbf{I}-\tilde{\mathbf{L}})^k,\quad \theta_{1,k}=\alpha_1(1-\alpha_1)^k,$$

$$T_2^{(k)}=(\mathbf{I}+\beta_2\tilde{\mathbf{L}})(\mathbf{I}-\tilde{\mathbf{L}})^k,\quad \theta_{2,k}=\alpha_2(1-\alpha_2)^k,$$

where there are $\alpha_1,\alpha_2\in[0,1],\beta_1\in[0,1/2],\beta_2\in(0,+\infty)$ according to the optimization objective.

FiGURe. FiGURe [27] suggests using filter bank to adapt the unsupervised settings where the graph information can assist filter formation. It considers up to $Q=4$ filters, i.e., Identity $\mathbf{I}$, Monomial $\mathbf{I}-\tilde{\mathbf{L}}$, Chebyshev, and Bernstein bases for the filter bank. In the first unsupervised stage, an embedding function $\gamma_q:\mathbb{R}^{n\times F}\to\mathbb{R}^{n\times F}$ is learned for each filter by maximizing the mutual information across all channels. The second stage of supervised graph representation learning fine-tunes another scalar weight γ_q' to tailor for the downstream task, along with other trainable model parameters. Formulation of the eventual FiGURe filter can be written as:

$$\mathscr{g}(\tilde{\mathbf{L}};\gamma,\theta)=\sum_{q=1}^{Q}\gamma_q'\gamma_q\cdot g_q(\tilde{\mathbf{L}};\theta),\quad g_q(\tilde{\mathbf{L}};\theta)=\sum_{k=0}^{K}\theta_{q,k}T_q^{(k)}(\tilde{\mathbf{L}}),$$

B Implementation Design

Our framework design is based on the popular graph learning library PyTorch Geometric (PyG) [33] with the same arrangement of components. We list the following highlights of our framework compared to PyTorch Geometric and similar works [25, 77]:

- Plug-and-play modules: The model and layer implementation in our framework follows the same design as PyTorch Geometric,

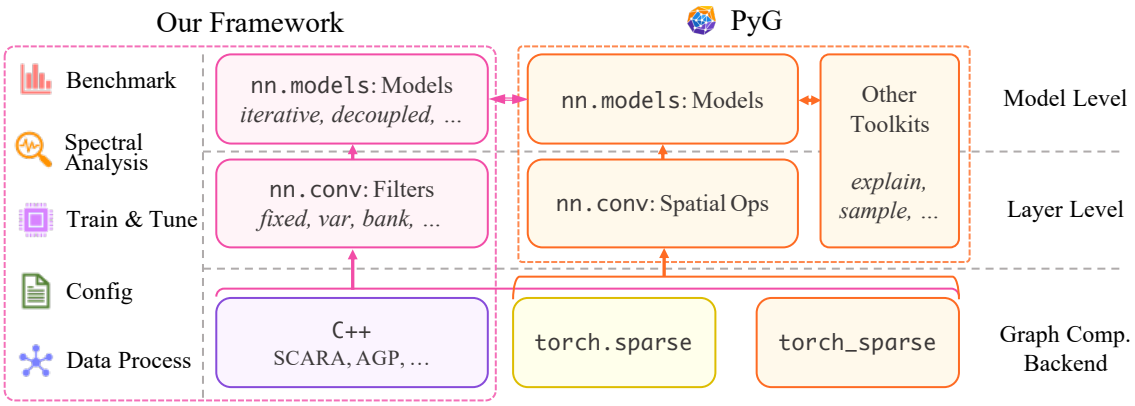


Figure 7: Code structure of our framework and relation to PyG.

implying that they can be seamlessly integrated into PyG-based programs. Our rich collection of spectral models and filters greatly extends the PyG model zoo.

- Separated spectral kernels: We decouple non-spectral designs and feature the pivotal spectral filters being consistent throughout different settings. Most of the filters are thence applicable to be combined with various network architectures, learning schemes, and other toolkits, including those provided by PyG and PyG-based frameworks.
- High scalability: As spectral GNNs are inherently suitable for large-scale learning, our framework is feasible to common scalable learning schemes and acceleration techniques. Several spectral-oriented approximation algorithms are also supported.

We further elaborate on our code structure in Figure 7 by comparing corresponding components in PyG:

- We implement the spectral filters as `nn.conv` modules, which are pivotal and basic blocks for building spectral GNNs. This is similar to the implementation of spatial layers in PyG.
- On the model level, common architectures formulated in Section 2.1 are included in `nn.models`. Since these models are built in PyG style, they can seamlessly access other utility interfaces offered by PyG and PyG-based toolkits.
- Regarding the underlying backend for sparse matrix multiplication in graph propagation, we support PyTorch-based computations inherited from PyG, as well as extendable interfaces for scalable C++-based algorithms specialized for spectral filtering.
- On top of the framework, we also offer utilities for benchmarking, efficiency evaluation, and spectral analysis.

C Full Experiment Results

C.1 Full-batch Learning

Efficiency evaluation of all full-batch models is shown in Table 9. Note that OBGN-ARXIV and ARXIV-YEAR share the same underlying graph structure, so we only present the performance on OBGN-ARXIV.

C.2 Mini-batch Learning

Table 10 displays the efficacy results of applicable spectral filters under mini-batch training across all datasets. Mini-batch efficiency is shown in Table 11 with the separated precomputation process.

C.3 Architecture Comparison

As comparison with other implementations, we additionally evaluate available spatial and spectral models in the PyG framework [33] in Table 8. The models are evaluated with iterative architecture $K = J = 4$ under available PyG graph backends and full-batch training, as PyG does not provide the mini-batch scheme. Especially, the EI backend is the most common backend used in PyG by default with $O(mF)$ space footprint for graph propagation. The SP backend is more efficient and used as the default backend in our main experiments, while only a handful of PyG models are compatible with it.

It can be observed that the accuracy and efficiency of the SP backend are comparable to our results considering the architectural difference, while the most common EI backend encounters the out-of-memory error on large datasets.

Table 8: Effectiveness and efficiency of PyG models with full-batch training on representative datasets.

Model (Backend*)	ARXIV					MAG				
	Acc	Train	Infer	RAM	GPU	Acc	Train	Infer	RAM	GPU
GCN (SP)	53.2	50.6	30.0	1.5	1.1	27.4	311.8	112.8	2.9	7.8
GraphSAGE (SP)	50.3	40.9	4.5	1.5	1.2	24.8	247.6	9.8	2.9	7.7
GCN (EI)	53.0	64.1	56.7	1.5	3.2			(OOM)		
GraphSAGE (EI)	54.1	87.2	28.7	1.6	2.7	28.2	325.5	176.1	2.4	10.8
ChebNet (EI)	53.4	112.3	51.1	1.3	3.0			(OOM)		
Model (Backend)	PENN94					GENIUS				
GCN (SP)	73.1	41.7	32.5	3.5	1.3	86.4	63.2	40.2	1.5	1.8
GraphSAGE (SP)	74.2	77.4	2.6	3.4	2.3	87.0	63.2	4.8	1.5	2.5
GCN (EI)	67.6	60.6	57.0	3.4	3.7	87.2	161.6	153.8	1.7	7.0
GraphSAGE (EI)			(OOM)			87.7	160.7	135.4	1.8	6.9
ChebNet (EI)			(OOM)			82.0	85.3	22.9	1.4	3.6
Model (Backend)	POKEC					TWITCH				
GCN (SP)	60.4	663.3	349.9	6.6	10.2	93.5	165.7	96.2	2.4	2.8
GraphSAGE (SP)	63.4	448.8	8.7	6.6	9.6	96.0	85.5	5.9	2.4	1.7
GCN (EI)			(OOM)			96.9	266.6	605.6	2.3	14.6
GraphSAGE (EI)			(OOM)			96.9	202.6	4.1	2.3	7.6
ChebNet (EI)			(OOM)			86.1	162.6	175.3	1.5	7.2

* **Backend**: SP = `torch.sparse`, EI = `torch_geometric.EdgeIndex` [33].

Table 9: Time and memory efficiency of *full-batch* training on medium and large datasets. "Train" and "Infer" respectively refer to average training time per epoch and inference time (ms), while "RAM", and "GPU" respectively refer to peak RAM and GPU memory (GB). For each column, highlighted ones are results ranked first, second, and third within the error interval.

Filter	FLICKR				PENN94				ARXIV				TWITCH			
	Train	Infer	RAM	GPU	Train	Infer	RAM	GPU	Train	Infer	RAM	GPU	Train	Infer	RAM	GPU
Identity	6.3±1.0	2.1±0.2	1.7±0.4	0.4±0.0	7.9±0.8	2.0±0.5	3.4±0.3	0.9±0.0	8.9±0.9	1.7±0.1	1.5±0.3	0.5±0.0	6.9±0.8	2.3±0.0	1.8±0.3	0.4±0.0
Linear	46.3±1.0	7.8±0.5	1.7±0.4	0.6±0.0	66.0±1.1	4.7±0.2	3.5±0.4	1.1±0.0	95.8±0.8	7.0±0.4	1.6±0.3	0.9±0.0	165.4±0.7	7.4±0.2	1.8±0.4	0.9±0.0
Impulse	38.5±1.0	6.7±0.1	1.7±0.4	0.6±0.0	59.0±1.1	4.8±0.1	3.4±0.4	1.1±0.0	79.6±0.9	7.1±0.2	1.6±0.3	0.9±0.0	141.8±1.1	6.8±0.2	1.8±0.4	0.9±0.0
Monomial	39.3±1.1	6.8±0.2	1.7±0.4	0.6±0.0	59.1±0.7	4.8±0.3	3.4±0.4	1.1±0.0	79.6±0.5	7.3±0.1	1.6±0.3	0.9±0.0	143.2±1.0	7.5±0.1	1.8±0.4	0.9±0.0
PPR	38.3±1.1	6.9±0.0	1.7±0.4	0.6±0.0	63.4±1.9	4.8±0.3	3.5±0.4	1.1±0.0	79.1±0.6	6.8±0.3	1.6±0.3	0.9±0.0	142.4±1.0	6.7±0.0	1.8±0.4	0.9±0.0
HK	38.1±1.0	7.1±0.6	1.7±0.4	0.6±0.0	59.3±1.1	9.5±7.8	3.4±0.4	1.1±0.0	79.0±1.0	6.8±0.0	1.6±0.3	0.7±0.0	142.4±1.0	7.0±0.8	1.7±0.5	0.9±0.0
Gaussian	38.7±1.3	7.1±0.5	1.7±0.4	0.6±0.0	68.5±2.7	4.8±0.2	3.5±0.4	1.1±0.0	80.2±0.7	7.1±0.1	1.6±0.3	0.9±0.0	140.7±0.9	7.1±0.6	1.8±0.4	0.9±0.0
Linear	52.8±1.2	7.9±0.6	1.7±0.4	1.4±0.0	71.6±1.4	7.5±0.4	3.4±0.4	1.5±0.0	106.7±1.0	7.4±0.4	1.6±0.3	2.4±0.0	176.1±0.2	7.1±0.1	1.8±0.3	2.5±0.0
Monomial	41.4±1.1	7.1±0.6	1.7±0.4	0.9±0.0	61.8±1.0	7.3±0.3	3.5±0.4	1.3±0.0	84.3±1.0	7.2±0.3	1.6±0.3	1.5±0.0	147.2±0.6	7.2±0.2	1.8±0.4	1.7±0.0
Horner	54.6±1.5	10.5±0.7	1.7±0.4	1.1±0.0	68.7±1.1	10.0±0.4	3.4±0.4	1.3±0.0	109.9±1.0	9.2±0.2	1.6±0.3	1.9±0.0	170.6±0.6	9.6±0.4	1.8±0.4	1.7±0.0
Chebyshev	46.0±1.2	9.2±0.5	1.7±0.4	1.0±0.0	63.2±0.9	8.8±0.1	3.4±0.3	1.3±0.0	94.0±1.0	10.6±4.9	1.6±0.3	1.6±0.0	153.5±0.3	8.5±0.0	1.8±0.4	1.8±0.0
Clenshaw	61.9±1.1	10.9±0.1	1.7±0.4	1.2±0.0	72.2±0.9	11.7±0.3	3.4±0.4	1.3±0.0	123.7±1.0	10.8±0.6	1.6±0.3	1.8±0.0	185.4±0.8	10.4±0.1	1.8±0.3	1.9±0.0
ChebInterp	67.3±1.9	14.1±0.1	1.7±0.4	1.0±0.0	82.9±1.1	11.2±0.4	3.5±0.4	1.3±0.0	126.0±0.5	71.6±99.9	1.6±0.3	1.6±0.0	175.8±2.5	15.6±4.3	1.8±0.4	1.8±0.0
Bernstein	80.3±0.7	9.5±0.6	1.7±0.4	1.0±0.0	122.1±1.1	10.2±0.4	3.5±0.4	1.3±0.0	204.2±3.0	9.7±0.4	1.6±0.3	1.8±0.0	308.3±0.6	9.8±0.1	1.8±0.4	1.8±0.0
Legendre	49.2±1.3	8.9±0.1	1.7±0.4	1.0±0.0	66.0±1.4	9.8±0.4	3.3±0.5	1.3±0.0	117.9±0.6	14.0±7.5	1.6±0.3	1.6±0.0	162.7±1.3	9.4±0.2	1.8±0.4	1.8±0.0
Jacobi	55.7±1.4	9.2±0.3	1.9±0.0	1.0±0.0	68.5±1.6	10.1±0.0	3.3±0.4	1.3±0.0	113.1±0.9	9.5±0.2	1.6±0.3	1.7±0.0	173.2±1.0	9.5±0.6	1.8±0.3	1.8±0.0
Favard	71.3±1.3	10.4±0.5	1.7±0.4	1.5±0.0	75.9±1.7	11.3±0.1	3.4±0.3	1.5±0.0	135.6±1.1	9.5±0.1	1.6±0.3	2.6±0.0	197.1±2.4	9.8±0.5	1.8±0.4	2.7±0.0
OptBasis	154.9±1.1	12.8±0.6	1.7±0.4	2.3±0.0	113.6±1.1	13.3±0.8	3.5±0.4	1.9±0.0	349.1±0.8	13.2±0.1	1.6±0.3	4.2±0.0	413.3±0.8	12.7±0.2	1.9±0.4	4.1±0.0
AdaGNN	43.2±1.6	6.8±0.3	1.7±0.4	1.0±0.0	79.9±1.1	6.6±0.1	3.5±0.4	1.3±0.0	87.8±1.4	6.1±0.2	1.6±0.3	1.6±0.0	148.6±1.5	6.2±0.6	1.9±0.3	1.7±0.0
FBGNNI	147.3±1.7	13.0±0.4	1.7±0.4	2.9±0.0	185.6±1.6	12.4±2.1	3.5±0.4	2.2±0.0	290.4±0.9	10.8±0.4	1.6±0.3	5.2±0.0	434.6±2.8	10.0±0.1	1.9±0.4	5.3±0.0
FBGNNII	149.1±0.7	12.4±0.3	1.7±0.4	3.8±0.0	176.9±24.2	10.2±1.1	3.4±0.4	2.6±0.0	289.9±0.6	13.1±1.1	1.6±0.3	7.0±0.0	435.6±2.1	10.8±1.3	1.9±0.4	7.0±0.0
ACMGNNI	188.1±1.5	14.2±1.0	1.7±0.4	3.9±0.0	181.9±3.0	13.2±1.9	3.5±0.4	2.6±0.0	364.3±1.1	15.7±2.8	1.6±0.3	7.1±0.0	508.0±1.1	13.0±1.0	1.8±0.4	7.1±0.0
ACMGNNII	189.7±1.6	14.8±0.4	1.7±0.4	4.8±0.0	230.4±1.3	13.1±0.3	3.5±0.4	3.1±0.0	366.1±0.7	14.9±1.5	1.6±0.3	8.9±0.0	510.7±1.2	13.7±0.6	1.8±0.4	8.9±0.0
FAGNN	89.5±0.8	8.5±0.2	1.7±0.4	0.6±0.0	128.4±1.1	5.8±0.2	3.5±0.3	1.1±0.0	186.6±0.7	9.0±0.1	1.6±0.3	1.0±0.0	334.9±1.4	8.8±1.1	1.8±0.4	1.0±0.0
G^2CN	44.8±1.4	7.0±0.3	1.7±0.4	0.5±0.0	70.7±0.8	5.6±0.1	3.5±0.4	1.1±0.0	94.3±1.1	8.0±0.2	1.6±0.3	0.9±0.0	170.1±1.1	7.6±0.3	1.8±0.4	1.0±0.0
GNN-LF/HF	80.7±1.1	8.9±0.4	1.7±0.4	0.7±0.0	123.9±0.8	6.0±0.5	3.5±0.4	1.1±0.0	170.0±0.9	8.6±1.0	1.6±0.3	1.0±0.0	318.8±0.7	7.8±0.0	1.8±0.4	1.1±0.0
FiGURe	161.4±1.6	14.3±0.1	1.7±0.4	2.1±0.0	260.9±22.7	12.8±0.1	3.4±0.4	1.8±0.0	346.3±0.7	13.3±0.9	1.6±0.3	3.6±0.0	618.7±1.0	16.2±2.7	1.9±0.4	3.8±0.0
Filter	GENIUS				MAG				POKEC				SNAP			
Identity	11.0±0.9	3.1±0.1	1.4±0.1	0.9±0.0	60.3±0.9	4.5±0.7	2.9±0.3	7.3±0.0	46.3±0.9	2.9±0.2	5.3±0.3	4.1±0.0	108.1±1.9	3.1±1.5	10.3±0.4	8.8±0.0
Linear	114.2±0.9	7.1±0.2	1.4±0.1	2.0±0.0	508.1±2.8	10.9±0.7	2.9±0.3	7.2±0.0	1071.8±0.7	8.7±0.0	5.3±0.4	8.2±0.0	1386.8±1.4	71.9±2.4	10.3±0.4	15.6±0.0
Impulse	84.0±0.9	7.4±0.1	1.4±0.1	1.7±0.0	431.8±5.0	10.7±0.5	2.9±0.3	7.1±0.1	896.7±1.3	8.3±0.1	5.3±0.4	8.2±0.0	1133.8±10.4	68.4±10.9	10.3±0.4	15.6±0.0
Monomial	84.3±1.0	6.9±0.5	1.4±0.1	2.0±0.0	434.7±0.8	10.4±0.1	3.0±0.3	7.2±0.0	897.0±0.7	8.4±0.4	5.3±0.4	8.2±0.0	1134.7±4.9	10.9±4.1	10.3±0.4	14.0±0.0
PPR	84.6±1.0	7.3±0.1	1.4±0.1	2.0±0.0	434.7±1.8	10.9±0.9	2.9±0.3	7.1±0.1	897.3±1.1	8.9±0.2	5.3±0.4	8.2±0.0	1142.7±1.6	65.3±1.7	10.3±0.4	15.6±0.0
HK	84.6±1.1	7.2±0.1	1.4±0.1	2.0±0.0	432.5±2.3	10.6±0.3	2.9±0.3	7.2±0.0	892.5±0.9	8.5±0.1	5.3±0.4	6.3±0.0	1141.8±7.2	64.9±3.5	10.3±0.4	15.6±0.0
Gaussian	83.8±1.0	6.9±0.5	1.4±0.1	1.7±0.0	435.3±1.9	10.7±0.5	2.9±0.3	7.0±0.2	897.0±1.1	8.7±0.3	5.3±0.4	7.2±0.0	1140.5±9.8	50.0±33.7	10.3±0.4	15.6±0.0
Linear	138.3±0.8	7.2±0.5	1.4±0.2	5.5±0.0	551.3±0.5	10.7±0.4	2.9±0.3	14.5±0.0	(OOM)				(OOM)			
Monomial	96.3±1.1	6.6±0.3	1.4±0.2	3.6±0.0	447.4±1.5	10.7±0.3	3.0±0.3	10.3±0.0	944.0±0.9	8.8±0.5	5.3±0.4	14.4±0.0	(OOM)			
Horner	157.3±1.8	9.6±0.4	1.4±0.1	4.4±0.0	563.7±2.9	13.5±0.8	3.0±0.3	10.9±0.0	1178.4±0.7	20.1±17.6	5.3±0.4	16.6±0.0	(OOM)			
Chebyshev	120.2±12.6	8.3±0.6	1.4±0.1	3.6±0.0	489.2±6.0	13.2±2.7	2.9±0.3	10.7±0.0	1013.2±1.0	9.3±0.4	5.3±0.4	15.2±0.0	(OOM)			
Clenshaw	200.6±4.9	10.8±0.1	1.5±0.0	4.6±0.0	631.4±6.9	14.2±0.9	2.9±0.3	10.9±0.0	1318.3±0.6	89.7±10.7	5.3±0.4	18.3±0.0	(OOM)			
ChebInterp	152.6±3.2	14.1±0.6	1.4±0.2	3.6±0.0	522.2±6.4	17.8±1.7	2.9±0.3	10.8±0.0	1081.4±0.7	13.2±0.1	5.2±0.4	14.2±0.0	(OOM)			
Bernstein	244.1±0.5	11.0±0.4	1.4±0.1	4.0±0.0	972.1±0.7	13.5±0.8	2.9±0.3	10.9±0.0	2238.9±0.8	44.7±0.7	5.3±0.4	17.5±0.0	(OOM)			
Legendre	168.2±1.3	9.5±0.3	1.4±0.1	3.8±0.0	519.7±1.1	12.8±0.4	2.9±0.3	10.5±0.0	1090.7±1.1	11.2±0.6	5.3±0.4	15.2±0.0	(OOM)			
Jacobi	160.9±2.3	10.7±0.0	1.4±0.1	3.8±0.0	566.0±1.6	12.8±0.5	2.9±0.3	10.6±0.0	1191.6±1.1	53.9±3.5	5.3±0.4	15.9±0.0	(OOM)			
Favard	230.7±3.7	24.3±13.4	1.4±0.1	6.1±0.0	673.1±0.5	12.8±0.6	2.9±0.3	14.9±0.0	(OOM)				(OOM)			
OptBasis	941.5±2.3	12.8±0.9	1.5±0.1	10.0±0.0	(OOM)				(OOM)				(OOM)			
AdaGNN	100.9±1.0	6.2±0.6	1.4±0.1	3.6±0.0	461.3±2.1	9.6±0.6	2.9±0.3	10.4±0.0	959.4±0.9	7.2±0.5	5.3±0.4	13.9±0.0	(OOM)			
FBGNNI	471.8±1.3	11.2±0.2	1.4±0.1	12.5±0.0	(OOM)				(OOM)				(OOM)			
FBGNNII	471.0±0.6	11.1±1.4	1.4±0.1	17.0±0.0	(OOM)				(OOM)				(OOM)			
ACMGNNI	649.1±0.7	14.4±0.1	1.4±0.1	17.2±0.0	(OOM)				(OOM)				(OOM)			
ACMGNNII	649.8±1.0	12.6±0.7	1.4±0.1	21.7±0.0	(OOM)				(OOM)				(OOM)			
FAGNN	221.2±0.8	9.3±0.1	1.4±0.1	1.9±0.0	933.1±0.9	11.7±0.3	2.9±0.3	7.2±0.0	2163.3±0.2	9.4±0.1	5.3±0.4	9.0±0.0	2744.4±6.4	80.1±14.0	10.3±0.4	15.3±0.0
G^2CN	98.5±0.8	7.8±0.1	1.4±0.1	2.2±0.0	502.0±1.0	10.4±0.2	2.9±0.3	7.2±0.0	1072.9±0.8	9.1±0.1	5.3±0.3	9.0±0.0	1351.4±9.5	135.3±4.7	10.3±0.4	16.9±0.0
GNN-LF/HF	181.4±1.4	8.9±1.1	1.4±0.2	2.2±0.0	860.1±2.8	11.4±0.3	2.9±0.3	7.4±0.1	2007.4±1.4	9.8±1.8	5.3±0.4	9.7±0.0	2480.8±14.4	141.1±7.9	10.3±0.4	16.0±0.0
FiGURe	394.0±1.1	14.2±0.1	1.4±0.1	8.4±0.0	1763.1±2.5	105.9±17.1	2.9±0.3	19.1±0.0	(OOM)				(OOM)			

Table 10: Effectiveness results (%) and standard deviations of spectral filters with *mini-batch* training on all datasets. For each dataset, highlighted results are filters with average scores ranked first , second , and third .

Filter	CORA	CITESEER	PUBMED	MINE	QUESTIONS	TOLOKERS	ARXIV	MAG	PRODUCTS
Identity	66.54±2.35	67.10±1.79	87.48±0.56	50.18±2.13	66.39±1.49	74.39±0.31	55.28±0.31	10.61±nan	26.62±0.04
Linear	85.29±1.27	74.66±1.11	84.96±0.50	65.64±2.16	63.13±2.83	78.84±1.44	68.24±0.21	13.06±0.92	26.59±0.01
Impulse	86.04±1.76	74.14±1.04	84.03±0.51	62.99±0.79	67.53±1.31	66.54±5.00	70.75±0.15	22.42±1.69	26.16±0.75
Monomial	86.30±1.54	74.66±1.88	89.46±0.47	58.46±5.64	73.70±2.08	81.61±1.30	70.66±0.11	32.79±0.84	26.59±0.00
PPR	87.19±1.72	75.53±1.50	88.73±0.46	81.53±2.86	65.36±1.91	80.90±1.19	70.02±0.40	17.02±3.99	26.91±0.44
HK	86.60±1.39	74.76±0.96	89.39±0.41	68.48±1.02	71.99±1.90	76.79±0.89	58.91±0.43	25.72±0.67	26.61±0.76
Gaussian	67.86±1.81	74.44±1.23	88.70±0.55	82.16±1.65	57.80±1.95	67.78±4.30	68.61±0.29	25.70±0.70	26.59±0.00
Linear	76.36±0.79	71.79±1.40	87.60±0.47	50.57±1.89	65.34±4.59	71.50±0.35	54.85±0.32	26.44±0.84	26.49±0.23
Monomial	87.41±1.24	76.26±1.35	89.83±0.53	67.66±3.43	63.67±4.81	75.35±2.85	67.05±0.47	33.11±0.37	26.82±0.42
Horner	86.64±1.63	75.33±0.80	84.17±0.46	62.14±1.32	63.35±2.36	71.22±1.21	65.53±0.63	30.04±0.47	26.59±0.00
Chebyshev	87.89±0.87	75.71±1.63	90.06±0.33	88.36±4.03	68.64±4.79	80.33±1.35	71.05±0.21	19.54±5.50	26.58±0.01
Clenshaw	80.83±2.22	72.60±0.96	75.75±1.07	54.17±2.07	63.99±1.80	75.03±1.64	54.34±0.15	29.74±0.94	26.59±0.00
ChebInterp	84.70±3.59	74.73±1.89	89.28±0.43	89.48±0.95	58.05±1.94	68.83±0.97	65.73±0.89	28.26±0.84	28.11±1.17
Bernstein	82.05±1.46	69.18±1.83	85.46±0.45	77.98±1.77	62.56±3.79	74.01±1.19	39.92±0.71	16.07±0.32	26.59±0.17
Legendre	87.43±1.00	76.17±1.02	89.84±0.29	63.32±1.81	64.14±2.24	75.78±1.02	71.65±0.13	34.74±0.45	27.10±0.36
Jacobi	87.17±1.50	75.34±1.26	89.75±0.33	89.84±0.91	71.42±3.01	76.80±0.72	71.41±0.09	32.96±0.23	24.37±3.14
OptBasis	82.20±2.46	73.11±1.23	88.23±1.10	88.56±1.13	57.70±5.70	79.62±2.38	70.46±0.24	31.29±1.68	26.56±0.06
FAGNN	86.95±1.13	75.07±1.05	85.25±0.28	73.82±0.93	62.01±2.39	77.91±0.59	70.11±0.33	35.55±0.37	26.59±0.01
G^{2}CN	86.51±1.79	75.05±1.11	84.42±0.51	73.11±1.09	72.65±2.74	69.70±6.19	67.34±0.41	29.68±0.27	26.65±0.13
GNN-LF/HF	87.36±1.02	75.95±1.35	89.76±0.39	76.77±7.55	73.70±1.52	79.35±0.95	70.51±0.06	29.41±0.22	25.80±0.91
FiGURe	87.21±1.13	76.67±1.33	89.73±0.49	57.93±3.89	70.88±1.87	77.36±3.03	71.87±0.18	34.43±1.13	26.24±1.03

Filter	CHAMELEON	SQUIRREL	ACTOR	ROMAN	RATINGS	FLICKR	YEAR	PENN94	GENIUS	TWITCH	POKEC	SNAP	WIKI
Identity	29.10±5.13	24.98±2.41	34.86±1.46	63.64±0.47	44.49±1.27	47.05±0.12	35.69±0.25	72.27±0.49	85.31±1.29	99.98±0.01	61.71±0.08	30.39±0.03	35.21±0.13
Linear	38.09±3.13	38.33±2.46	25.99±1.07	31.28±1.14	34.69±2.49	52.58±0.63	49.07±0.26	72.38±0.34	88.37±0.15	72.77±2.62	54.98±0.38	45.33±0.19	37.69±0.42
Impulse	39.83±2.21	38.94±1.78	24.32±1.60	28.63±0.87	35.81±2.37	50.55±0.62	45.23±0.15	60.52±0.49	88.38±0.34	77.50±0.67	57.84±0.11	53.56±0.18	32.88±1.47
Monomial	37.70±2.16	34.91±0.89	32.64±1.41	64.14±0.51	43.83±0.89	50.80±0.23	48.93±0.31	75.03±0.32	88.15±0.25	94.36±6.64	63.82±0.10	42.83±1.12	23.07±2.46
PPR	40.73±2.78	34.89±1.29	32.49±1.22	69.81±0.59	39.72±1.00	50.31±0.11	48.56±0.31	75.03±0.36	89.06±0.49	97.34±0.32	62.18±0.04	47.83±3.67	22.40±2.00
HK	41.63±3.14	36.37±1.94	32.88±1.39	67.82±0.49	36.66±1.16	50.69±0.27	47.50±0.30	72.77±0.76	89.10±0.14	97.79±1.10	62.04±0.08	50.99±0.16	43.06±0.30
Gaussian	39.08±2.93	35.32±1.60	32.84±1.28	65.38±1.55	34.76±2.36	51.45±0.24	48.03±0.21	73.93±0.45	88.44±0.40	99.32±0.10	62.20±0.05	52.58±0.04	42.81±0.45
Linear	37.36±3.17	34.93±1.94	35.72±0.88	63.34±0.79	36.27±0.83	46.63±0.18	35.03±0.50	74.63±0.67	86.10±0.15	89.90±5.42	62.21±0.16	30.47±0.34	38.98±0.51
Monomial	33.99±3.79	35.07±2.04	34.81±1.34	70.63±1.80	30.84±4.82	53.62±0.41	47.92±0.41	82.63±0.54	89.51±0.47	81.87±6.42	78.11±0.70	50.35±0.07	31.34±0.48
Horner	36.40±2.74	36.62±1.69	25.49±1.47	28.40±1.69	30.91±4.71	52.47±0.30	44.84±0.19	47.09±0.65	88.38±0.14	77.82±0.67	62.87±0.25	38.73±1.71	31.42±0.37
Chebyshev	34.27±3.39	34.23±1.21	29.89±1.82	35.06±2.65	37.15±2.22	47.99±1.14	45.54±0.49	83.40±0.40	86.70±0.19	99.92±0.05	78.95±0.13	42.83±0.96	24.26±2.39
Clenshaw	35.06±2.57	35.18±0.97	23.93±0.55	21.42±0.60	45.63±1.05	41.35±0.99	44.64±2.77	48.39±2.59	86.98±0.07	74.34±4.28	52.79±2.66	47.18±0.20	31.25±0.10
ChebInterp	28.37±3.09	34.62±1.98	30.36±1.22	70.95±0.69	35.58±2.22	53.79±0.64	44.28±0.29	63.59±0.70	87.65±0.38	99.71±0.45	78.60±0.68	45.77±0.56	23.85±1.84
Bernstein	39.61±3.21	35.68±2.35	30.68±1.05	53.87±1.09	31.24±2.14	44.58±2.06	37.34±1.11	79.72±0.34	88.12±0.47	97.43±0.55	73.61±0.02	48.67±0.28	24.54±1.95
Legendre	38.70±2.49	36.15±2.02	33.05±1.45	61.08±2.63	39.94±1.27	50.53±0.20	47.73±0.30	76.88±0.41	89.15±0.42	97.32±4.52	72.60±1.49	42.94±0.54	31.16±0.40
Jacobi	38.54±3.73	35.86±1.80	33.32±2.14	71.05±0.67	37.06±1.47	53.29±0.17	50.29±0.23	84.06±0.43	89.05±0.52	98.61±2.03	73.05±2.02	53.37±2.23	33.25±3.10
OptBasis	38.09±2.54	35.00±1.62	29.34±1.61	56.85±2.90	42.15±1.08	50.81±0.68	40.87±0.89	81.81±0.99	89.06±0.36	86.13±0.50	78.67±0.11	57.49±2.42	30.87±3.11
FAGNN	39.66±2.47	37.91±2.05	28.23±1.78	46.59±1.97	41.44±0.75	53.94±0.21	45.85±0.96	68.49±0.88	88.14±0.17	79.92±1.70	62.60±0.12	39.29±0.66	31.23±0.55
G^{2}CN	40.79±3.47	39.08±2.93	21.47±0.88	65.53±0.74	33.84±1.18	51.04±0.24	49.85±0.36	80.55±0.45	89.50±0.46	78.72±1.21	50.69±0.05	25.89±0.16	38.66±0.10
GNN-LF/HF	40.66±2.14	34.75±1.98	33.91±1.61	45.11±5.00	38.04±1.84	47.65±0.45	49.47±0.38	75.72±0.44	88.79±0.15	99.91±0.07	62.95±0.65	44.82±1.81	35.97±1.65
FiGURe	39.33±4.26	38.21±2.33	33.45±2.07	63.48±6.27	39.09±1.17	52.51±0.36	50.26±0.33	83.70±0.38	83.21±0.18	71.26±3.64	73.58±2.53	41.76±0.66	38.57±0.76

Table 11: Time and memory efficiency of *mini-batch* training on medium- and large-scale datasets. "Pre." refers to precomputation time (s), "Train" and "Infer" respectively refer to average training time per epoch and inference time (ms), while "RAM", and "GPU" respectively refer to peak RAM and GPU memory (GB). For each column, highlighted ones are results ranked first , second , and third within the error interval. .

Filter	FLICKR Pre.	FLICKR Train	FLICKR Infer	FLICKR RAM	FLICKR GPU	PENN94 Pre.	PENN94 Train	PENN94 Infer	PENN94 RAM	PENN94 GPU
Identity	0.0±0.0	21.3±2.7	6.0±0.9	1.8±0.2	0.0±0.0	0.0±0.0	14.8±0.8	4.1±0.6	3.6±0.2	0.2±0.0
Linear	1.8±0.0	22.1±2.3	6.8±0.2	2.4±0.3	0.0±0.0	16.0±0.7	17.1±0.9	5.3±0.7	6.6±0.2	0.2±0.0
Impulse	1.3±0.1	24.1±1.3	7.9±1.7	2.3±0.3	0.0±0.0	12.0±0.4	16.0±0.5	5.0±0.9	5.9±0.2	0.2±0.0
Monomial	1.2±0.0	22.6±3.8	6.9±0.6	2.3±0.3	0.0±0.0	12.9±0.5	16.5±0.9	5.3±0.5	5.9±0.2	0.2±0.0
PPR	1.2±0.0	20.2±2.1	7.0±0.6	2.3±0.3	0.0±0.0	12.4±0.4	16.6±0.8	4.7±0.8	5.8±0.2	0.2±0.0
HK	1.2±0.0	24.3±4.4	7.7±1.8	2.3±0.3	0.0±0.0	12.3±0.6	16.3±1.1	5.3±0.4	5.9±0.2	0.2±0.0
Gaussian	1.3±0.1	24.8±1.3	8.5±2.0	2.4±0.3	0.0±0.0	12.3±0.6	16.5±0.5	5.4±0.8	5.9±0.2	0.2±0.0
Var-Linear	1.4±0.1	55.5±0.7	27.0±5.8	5.4±0.3	0.2±0.0	13.2±1.1	26.1±2.3	15.0±8.1	19.4±0.2	1.8±0.0
Var-Monomial	1.2±0.2	60.7±2.3	23.0±5.2	5.4±0.3	0.2±0.0	12.4±1.3	32.5±0.6	11.0±1.3	19.4±0.2	1.8±0.0
Horner	1.2±0.2	51.0±0.6	30.4±0.1	5.4±0.3	0.2±0.0	13.7±0.7	31.6±0.8	18.2±3.7	19.8±0.3	1.8±0.0
Chebyshev	1.6±0.1	55.4±4.0	21.6±3.1	5.4±0.3	0.2±0.0	15.1±0.8	32.4±0.6	11.5±0.5	19.3±0.4	1.8±0.0
Clenshaw	1.9±0.2	60.1±1.8	32.7±3.6	5.4±0.3	0.2±0.0	15.5±0.6	29.0±1.0	16.1±2.9	19.8±0.4	1.8±0.0
ChebInterp	1.7±0.1	606.5±13.2	180.7±5.8	5.4±0.3	0.2±0.0	13.9±0.8	296.5±1.6	67.0±0.1	19.3±0.3	1.8±0.0
Bernstein	5.0±0.4	60.3±5.1	25.2±0.4	5.5±0.3	0.2±0.0	67.7±1.2	39.5±1.8	17.5±1.9	19.3±0.4	1.9±0.0
Legendre	1.8±0.1	51.9±1.0	21.1±0.7	5.4±0.3	0.2±0.0	18.6±2.3	36.5±3.4	13.1±4.9	19.3±0.4	1.8±0.0
Jacobi	2.3±0.1	54.5±4.0	20.6±0.6	5.4±0.3	0.2±0.0	20.8±2.8	33.3±1.6	15.1±4.9	19.3±0.4	1.8±0.0
OptBasis	11.4±0.1	50.4±1.5	20.4±0.1	5.4±0.3	0.2±0.0	108.2±6.1	30.8±0.7	10.1±1.0	19.8±0.3	1.8±0.0
FAGNN	3.8±0.2	31.6±0.4	8.5±0.0	2.7±0.3	0.1±0.0	29.5±2.6	19.7±1.2	5.5±0.6	7.3±0.2	0.3±0.0
G^{2}CN	1.4±0.1	30.1±1.6	8.5±0.1	2.7±0.3	0.1±0.0	49.6±1.7	18.0±1.3	5.4±0.7	7.4±0.2	0.3±0.0
GNN-LF/HF	2.8±0.2	30.2±5.0	8.1±0.1	2.5±0.3	0.1±0.0	24.7±1.3	16.8±0.4	4.8±0.0	6.9±0.3	0.3±0.0
FiGURe	8.6±0.4	143.1±1.6	49.3±7.3	15.1±0.3	0.6±0.0	85.4±1.9	61.8±1.5	20.4±1.0	63.3±0.2	5.3±0.0

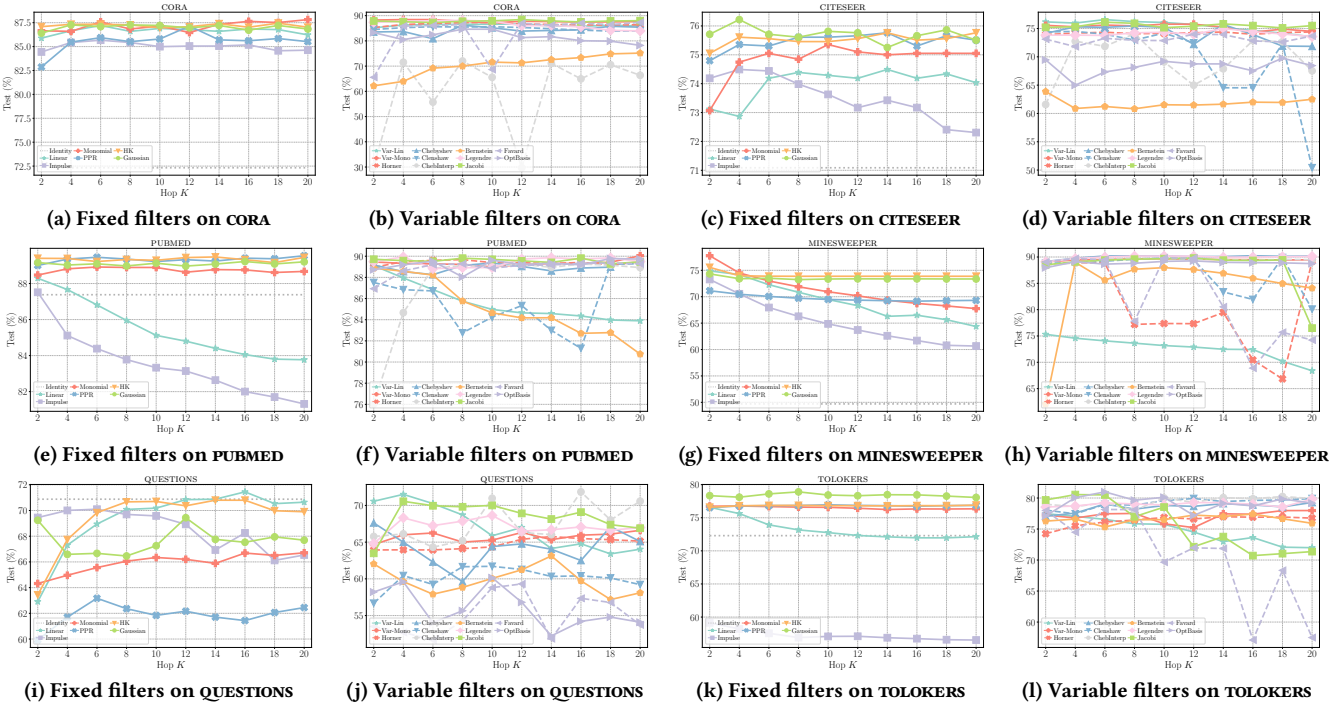
Filter	ARXIV Pre.	ARXIV Train	ARXIV Infer	ARXIV RAM	ARXIV GPU	TWITCH Pre.	TWITCH Train	TWITCH Infer	TWITCH RAM	TWITCH GPU	GENIUS Pre.	GENIUS Train	GENIUS Infer	GENIUS RAM	GENIUS GPU
Identity	0.0±0.0	33.1±1.0	9.8±2.3	1.7±0.2	0.1±0.0	0.0±0.0	32.6±1.8	8.8±0.9	2.5±0.2	0.0±0.0	0.0±0.0	96.8±13.8	40.1±13.6	1.4±0.1	0.0±0.0
Linear	0.9±0.0	41.7±2.4	13.1±2.5	1.9±0.3	0.1±0.0	0.3±0.4	40.1±5.6	14.3±4.8	2.5±0.2	0.0±0.0	0.2±0.3	87.2±10.2	33.1±9.1	1.5±0.1	0.0±0.0
Impulse	0.6±0.1	40.2±0.7	13.4±0.8	1.8±0.3	0.1±0.0	0.4±0.4	37.4±2.1	12.5±1.9	2.5±0.2	0.0±0.0	1.0±0.6	109.9±31.4	46.5±14.1	1.5±0.1	0.0±0.0
Monomial	0.6±0.1	39.1±1.5	14.8±4.1	1.8±0.3	0.1±0.0	0.3±0.3	35.5±0.7	11.9±0.8	2.5±0.2	0.0±0.0	0.4±0.5	102.5±16.1	40.0±15.0	1.5±0.1	0.0±0.0
PPR	0.6±0.0	40.0±0.8	12.8±1.6	1.8±0.3	0.1±0.0	0.2±0.0	51.9±6.8	16.9±4.4	2.5±0.2	0.0±0.0	0.3±0.4	92.5±10.0	34.3±7.1	1.5±0.1	0.0±0.0
HK	0.6±0.0	40.2±0.6	12.7±0.4	1.8±0.3	0.1±0.0	0.4±0.3	35.0±2.0	12.0±1.2	2.5±0.2	0.0±0.0	0.6±0.7	113.4±15.6	50.6±16.8	1.5±0.1	0.0±0.0
Gaussian	0.6±0.0	37.7±1.0	15.4±4.2	1.8±0.3	0.1±0.0	0.2±0.2	42.8±6.5	12.7±2.3	2.5±0.2	0.0±0.0	0.2±0.2	111.3±18.0	55.9±28.9	1.5±0.1	0.0±0.0
Var-Linear	0.6±0.0	67.8±1.0	43.4±3.8	3.5±0.3	0.1±0.0	1.4±0.0	152.7±29.8	67.4±8.7	2.4±0.3	0.0±0.0	0.3±0.3	183.7±23.5	147.2±31.6	1.9±0.2	0.0±0.0
Var-Monomial	0.6±0.1	76.7±1.0	41.2±8.9	3.4±0.3	0.1±0.0	1.2±0.3	143.6±3.5	47.7±4.4	2.4±0.3	0.0±0.0	0.3±0.3	190.3±7.2	100.0±34.0	1.9±0.2	0.0±0.0
Horner	0.7±0.1	79.6±3.1	61.0±11.4	3.5±0.4	0.1±0.0	1.9±0.2	164.6±18.2	84.0±23.7	2.4±0.3	0.0±0.0	0.1±0.0	207.2±28.8	169.3±37.6	1.9±0.2	0.0±0.0
Chebyshev	0.8±0.1	74.6±2.0	35.1±8.3	3.5±0.4	0.1±0.0	1.6±0.1	139.9±9.8	52.2±13.2	2.4±0.3	0.0±0.0	0.6±0.3	157.7±14.0	81.2±10.9	1.9±0.2	0.0±0.0
Clenshaw	0.9±0.0	76.1±1.3	53.5±5.3	3.4±0.3	0.1±0.0	1.9±0.2	155.4±10.4	71.1±1.9	2.4±0.4	0.0±0.0	0.6±0.3	183.0±17.9	168.8±35.1	1.9±0.2	0.0±0.0
ChebInterp	0.8±0.1	917.0±68.0	295.4±32.9	3.4±0.3	0.1±0.0	1.6±0.1	758.9±4.8	393.8±15.9	2.4±0.3	0.0±0.0	0.1±0.0	1632.7±99.1	867.5±62.5	1.9±0.2	0.0±0.0
Bernstein	2.5±0.3	90.8±3.6	46.5±6.7	3.5±0.3	0.1±0.0	7.2±0.7	171.6±26.1	79.8±40.6	2.5±0.4	0.0±0.0	1.4±1.9	180.9±16.4	117.8±41.7	1.9±0.2	0.0±0.0
Legendre	0.9±0.1	76.2±2.6	33.2±2.9	3.5±0.4	0.1±0.0	2.6±0.6	170.0±33.3	47.4±3.2	2.4±0.3	0.0±0.0	0.4±0.3	164.0±11.1	80.6±10.3	2.0±0.3	0.0±0.0
Jacobi	1.2±0.1	84.1±7.9	36.9±9.2	3.5±0.4	0.1±0.0	1.8±1.0	136.9±8.0	52.4±2.5	2.4±0.3	0.0±0.0	0.1±0.0	126.8±3.0	66.6±7.3	2.1±0.3	0.0±0.0
OptBasis	6.8±0.7	79.0±2.3	38.1±6.6	3.4±0.3	0.1±0.0	10.5±8.5	164.4±23.0	45.4±8.1	2.4±0.4	0.0±0.0	0.9±0.0	130.1±6.1	66.7±5.8	1.9±0.3	0.0±0.0
FAGNN	1.3±0.3	47.1±1.4	14.5±1.2	2.1±0.3	0.1±0.0	0.3±0.1	44.1±3.9	13.2±1.2	2.5±0.2	0.0±0.0	0.3±0.3	125.9±11.8	40.0±4.5	1.6±0.2	0.0±0.0
G^{2}CN	0.6±0.1	50.4±1.5	16.2±2.6	2.0±0.3	0.1±0.0	0.6±0.2	42.7±3.2	13.4±2.0	2.5±0.2	0.0±0.0	0.6±0.6	135.6±30.2	64.6±18.8	1.5±0.1	0.0±0.0
GNN-LF/HF	1.3±0.0	47.3±3.7	15.1±1.1	2.1±0.3	0.1±0.0	0.3±0.0	43.9±2.6	13.7±1.2	3.4±0.7	0.0±0.0	0.1±0.1	100.0±8.1	35.6±1.2	1.6±0.2	0.0±0.0
FiGURe	5.5±2.1	164.8±11.4	86.6±16.0	8.2±0.3	0.2±0.0	1.2±0.3	173.4±25.1	91.5±12.4	2.9±0.5	0.0±0.0	0.9±0.5	401.1±35.7	248.0±49.9	3.0±0.3	0.0±0.0

Filter	MAG Pre.	MAG Train	MAG Infer	MAG RAM	MAG GPU	POKEC Pre.	POKEC Train	POKEC Infer	POKEC RAM	POKEC GPU
Identity	0.0±0.0	9.5±0.9	3.2±0.5	2.9±0.2	5.0±0.0	0.0±0.0	14.2±0.4	5.8±0.6	6.7±0.3	0.5±0.0
Linear	4.7±0.6	11.5±1.1	5.5±1.0	3.8±0.4	5.0±0.0	8.5±0.8	13.0±0.9	6.2±0.9	6.6±0.3	0.5±0.0
Impulse	3.9±0.2	11.4±0.5	5.3±0.5	3.4±0.3	4.9±0.0	5.4±0.2	14.6±0.3	6.4±0.3	6.6±0.3	0.5±0.0
Monomial	4.1±0.0	12.1±0.2	4.7±0.3	3.4±0.3	5.0±0.0	5.2±0.0	14.5±0.6	6.5±0.6	6.6±0.3	0.5±0.0
PPR	4.0±0.1	11.4±0.6	5.4±1.1	3.4±0.3	5.0±0.0	5.7±0.3	13.7±0.6	6.3±0.7	6.6±0.3	0.5±0.0
HK	3.9±0.2	12.1±1.1	4.8±0.1	3.4±0.3	5.0±0.0	5.2±0.1	14.2±1.2	6.3±0.3	6.6±0.4	0.5±0.0
Gaussian	3.9±0.3	12.1±1.4	5.7±1.8	3.4±0.3	5.0±0.0	5.2±0.2	13.6±1.4	7.7±0.6	6.6±0.3	0.5±0.0
Linear	4.0±0.2	20.4±2.6	9.4±0.4	10.1±0.3	5.0±0.0	5.7±0.2	25.0±1.2	15.6±0.8	11.8±0.4	1.2±0.0
Monomial	3.8±0.4	22.8±1.2	7.6±0.7	10.1±0.3	5.2±0.0	6.0±0.5	30.6±5.1	14.6±0.9	11.8±0.4	1.2±0.0
Horner	3.9±0.2	20.8±0.8	12.4±3.0	10.1±0.3	5.1±0.0	6.2±0.8	26.8±1.9	17.8±0.6	11.8±0.4	1.2±0.0
Chebyshev	5.7±1.1	27.0±1.5	9.5±2.7	10.2±0.2	5.1±0.0	7.1±0.1	27.3±0.9	12.0±0.1	12.3±0.4	1.2±0.0
Clenshaw	6.0±1.9	20.1±2.7	13.2±5.0	10.1±0.2	5.1±0.0	7.7±0.3	28.5±3.0	19.1±1.5	11.8±0.4	1.2±0.0
ChebInterp	5.7±1.3	319.8±2.6	47.0±5.4	10.2±0.3	3.7±1.0	7.3±0.5	251.2±1.0	77.0±3.0	12.3±0.4	1.2±0.0
Bernstein	23.0±2.2	31.1±5.1	9.9±1.3	10.4±0.4	4.9±0.0	26.4±0.8	30.9±0.4	15.2±1.2	13.4±0.6	1.6±0.0
Legendre	6.0±0.5	23.0±0.7	10.8±4.0	9.9±0.4	5.1±0.0	7.5±0.5	31.2±3.5	12.9±1.1	11.8±0.4	1.2±0.0
Jacobi	7.1±0.1	21.2±1.2	7.0±1.1	9.9±0.4	5.1±0.0	9.0±0.0	30.7±3.7	14.4±0.4	11.8±0.4	1.2±0.0
OptBasis	30.4±0.9	24.4±3.6	10.8±6.9	10.0±0.4	3.2±0.0	35.1±1.7	28.2±0.7	13.9±1.5	11.9±0.4	1.2±0.0
FAGNN	10.1±1.2	345.0±19.1	121.3±27.7	4.4±0.4	5.0±0.0	15.3±0.8	18.2±1.8	7.7±1.0	6.6±0.4	0.6±0.0
G^{2}CN	17.7±0.6	15.1±0.6	5.5±0.3	4.3±0.5	2.2±0.0	28.2±3.4	17.3±2.0	6.6±0.1	6.6±0.4	0.6±0.0
GNN-LF/HF	9.0±0.3	15.3±0.7	5.9±0.5	4.4±0.5	4.8±0.0	12.5±0.4	16.7±0.5	6.8±0.1	7.4±1.1	0.6±0.0
FiGURe	26.5±3.0	1220.1±118.7	420.6±45.3	31.5±0.4	5.1±0.0	46.6±8.5	52.7±0.7	26.6±1.5	37.1±0.7	3.5±0.0

Filter	SNAP Pre.	SNAP Train	SNAP Infer	SNAP RAM	SNAP GPU	PRODUCTS Pre.	PRODUCTS Train	PRODUCTS Infer	PRODUCTS RAM	PRODUCTS GPU	WIKI Pre.	WIKI Train	WIKI Infer	WIKI RAM	WIKI GPU
Identity	0.0±0.0	26.8±2.3	9.0±0.6	10.1±0.3	0.6±0.0	0.0±0.0	2.7±0.6	11.5±3.8	21.6±0.3	2.4±0.0	0.0±0.0	22.9±1.4	6.9±1.7	57.4±0.3	1.0±0.0
Linear	27.3±1.0	21.0±0.6	8.3±0.4	20.8±0.3	0.6±0.0	19.5±1.2	3.1±0.8	10.3±0.4	21.6±0.4	2.3±0.0	206.5±11.4	22.8±2.6	6.8±0.9	57.5±0.4	1.0±0.0
Impulse	21.0±1.4	21.5±1.0	8.4±0.2	18.1±0.4	0.6±0.0	15.0±0.5	3.1±0.6	10.3±1.2	21.6±0.4	2.3±0.0	164.1±23.0	21.8±0.6	7.4±1.1	57.5±0.4	1.0±0.0
Monomial	21.9±4.7	22.2±0.6	8.7±0.6	18.0±0.4	0.6±0.0	15.5±1.2	3.1±0.5	10.4±0.3	21.6±0.4	2.3±0.0	172.6±18.0	23.0±1.6	6.4±0.9	57.4±0.3	0.9±0.0
PPR	20.3±0.9	20.8±0.6	8.4±0.5	18.1±0.4	0.6±0.0	15.8±1.5	3.1±0.8	10.6±0.3	21.6±0.4	1.7±0.0	166.6±16.9	22.2±0.5	6.0±0.5	57.4±0.4	1.0±0.0
HK	18.8±0.7	22.2±0.6	8.4±0.3	18.1±0.4	0.6±0.0	14.8±0.4	3.1±0.6	11.1±0.6	21.6±0.4	1.9±0.0	165.6±9.6	20.9±0.6	5.8±0.2	57.4±0.3	1.0±0.0
Gaussian	19.7±0.1	22.6±0.1	10.0±0.7	18.1±0.4	0.6±0.0	14.7±0.3	3.0±0.7	9.8±0.8	21.6±0.4	2.1±0.0	156.1±25.3	20.4±0.7	6.1±0.3	57.5±0.4	1.0±0.0
Linear	26.5±3.4	33.9±0.7	22.4±2.6	70.3±0.3	4.9±0.0	16.6±0.7	4.6±0.6	16.2±0.4	25.9±0.3	2.4±0.2	160.2±28.8	32.8±2.7	15.9±2.1	109.0±0.3	10.8±0.0
Monomial	25.0±2.0	36.3±0.7	17.0±1.8	70.3±0.3	4.9±0.0	15.3±0.9	5.3±0.7	15.4±1.4	25.9±0.3	2.1±0.0	168.1±7.6	34.4±2.5	15.6±6.7	109.0±0.3	10.8±0.0
Horner	23.5±2.7	35.5±2.5	24.7±2.9	70.3±0.3	4.9±0.0	20.9±6.2	6.8±1.2	25.5±2.6	25.9±0.3	2.5±0.0	183.1±39.9	34.4±5.0	22.4±4.7	109.0±0.3	10.8±0.0
Chebyshev	26.3±1.6	32.2±0.8	15.4±2.4	70.5±0.3	4.9±0.0	24.1±3.9	6.4±0.9	17.6±3.3	27.3±0.3	2.5±0.0	186.1±21.6	33.8±1.0	15.6±2.5	114.4±0.3	10.8±0.0
Clenshaw	35.1±3.1	43.8±3.3	31.6±7.3	70.3±0.3	4.9±0.0	28.4±3.9	6.3±0.8	23.9±0.5	25.9±0.3	2.7±0.0	204.9±51.8	36.8±2.1	21.6±4.2	109.0±0.3	10.8±0.0
ChebInterp	35.2±1.9	922.4±86.8	108.6±7.3	70.5±0.3	4.9±0.0	23.5±5.1	70.4±1.6	102.1±8.4	27.3±0.3	2.6±0.0	202.1±74.7	1243.4±23.5	69.5±17.9	114.4±0.3	10.8±0.0
Bernstein	100.0±5.5	44.7±1.6	44.3±37.5	71.0±0.4	5.2±0.0	99.7±1.0	6.5±0.6	20.6±1.1	30.7±1.2	2.5±0.0	1069.5±263.2	37.8±3.2	14.2±1.0	124.4±2.1	11.7±0.0
Legendre	32.1±2.4	38.2±1.2	15.9±2.0	70.3±0.3	4.9±0.0	22.5±2.8	5.9±0.7	15.0±0.6	25.9±0.3	2.3±0.0	176.6±36.5	34.6±3.2	18.5±15.6	108.9±0.3	10.8±0.0
Jacobi	42.8±2.8	37.4±0.4	19.8±6.1	70.3±0.4	4.9±0.0	23.3±0.5	6.1±0.7	15.7±0.3	25.9±0.3	2.5±0.0	184.4±29.8	35.0±4.0	12.1±1.7	108.9±0.3	10.8±0.0
OptBasis	269.1±74.1	42.6±2.7	18.7±2.9	70.5±0.3	4.9±0.0	87.2±8.7	6.3±0.4	17.9±4.7	25.9±0.3	2.9±0.0	448.3±81.9	32.9±2.7	10.7±1.3	109.0±0.3	10.8±0.0
FAGNN	68.9±3.8	32.0±1.8	13.8±4.7	23.6±0.3	1.0±0.0	46.9±1.2	4.2±0.8	11.8±2.1	21.6±0.3	1.5±0.5	438.8±66.9	26.4±4.1	7.2±0.3	57.2±0.3	2.0±0.0
G^{2}CN	91.0±6.0	28.1±1.9	10.1±0.9	23.6±0.3	1.0±0.0	81.9±3.8	4.2±1.3	11.0±2.0	21.5±0.3	1.0±0.0	827.7±141.0	29.2±1.4	8.0±1.2	57.2±0.3	2.0±0.0
GNN-LF/HF	52.7±2.8	29.5±1.3	11.0±0.5	21.8±0.5	1.0±0.0	38.0±2.9	4.0±0.8	11.3±1.8	23.8±2.1	1.7±0.1	385.5±46.7	26.5±0.7	7.3±0.5	59.7±4.5	2.0±0.0
FiGURe	175.1±7.0	79.4±7.6	40.4±4.7	230.8±0.5	14.5±0.0	145.2±9.1	11.4±0.9	35.6±1.7	84.6±0.7	5.7±0.0	1405.6±231.2	107.8±9.3	45.1±7.2	359.8±2.4	20.4±0.9

D Specific Evaluation

D.1 Effect of Propagation Hop

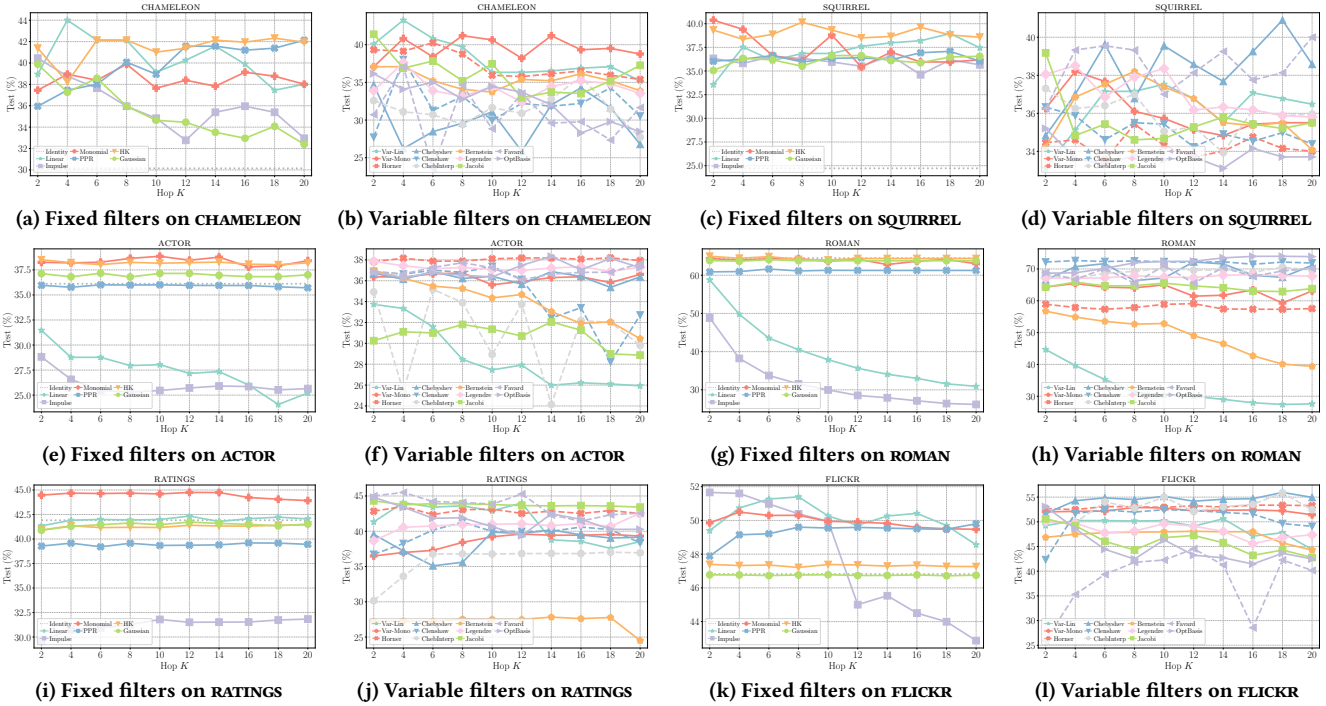


(a) Fixed filters on CORA (b) Variable filters on CORA (c) Fixed filters on CITESEER (d) Variable filters on CITESEER

(e) Fixed filters on PUBMED (f) Variable filters on PUBMED (g) Fixed filters on MINESWEEPER (h) Variable filters on MINESWEEPER

(i) Fixed filters on QUESTIONS (j) Variable filters on QUESTIONS (k) Fixed filters on TOLOKERS (l) Variable filters on TOLOKERS

Figure 8: Effect of *propagation hops* K of full-batch fixed and variable filters on homophilous datasets.



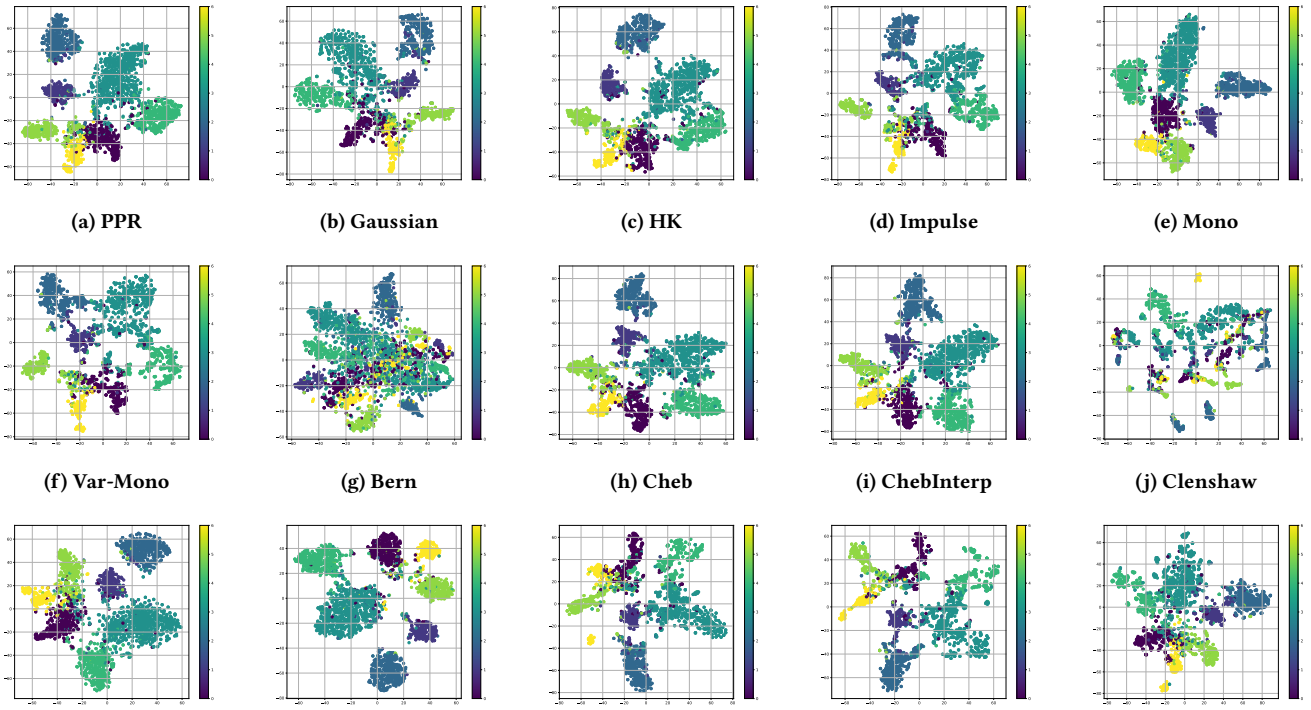
(a) Fixed filters on CHAMELEON (b) Variable filters on CHAMELEON (c) Fixed filters on SQUIRREL (d) Variable filters on SQUIRREL

(e) Fixed filters on ACTOR (f) Variable filters on ACTOR (g) Fixed filters on ROMAN (h) Variable filters on ROMAN

(i) Fixed filters on RATINGS (j) Variable filters on RATINGS (k) Fixed filters on FLICKR (l) Variable filters on FLICKR

Figure 9: Effect of *propagation hops* K of full-batch fixed and variable filters on heterophilous datasets.

D.2 Clustering Visualization

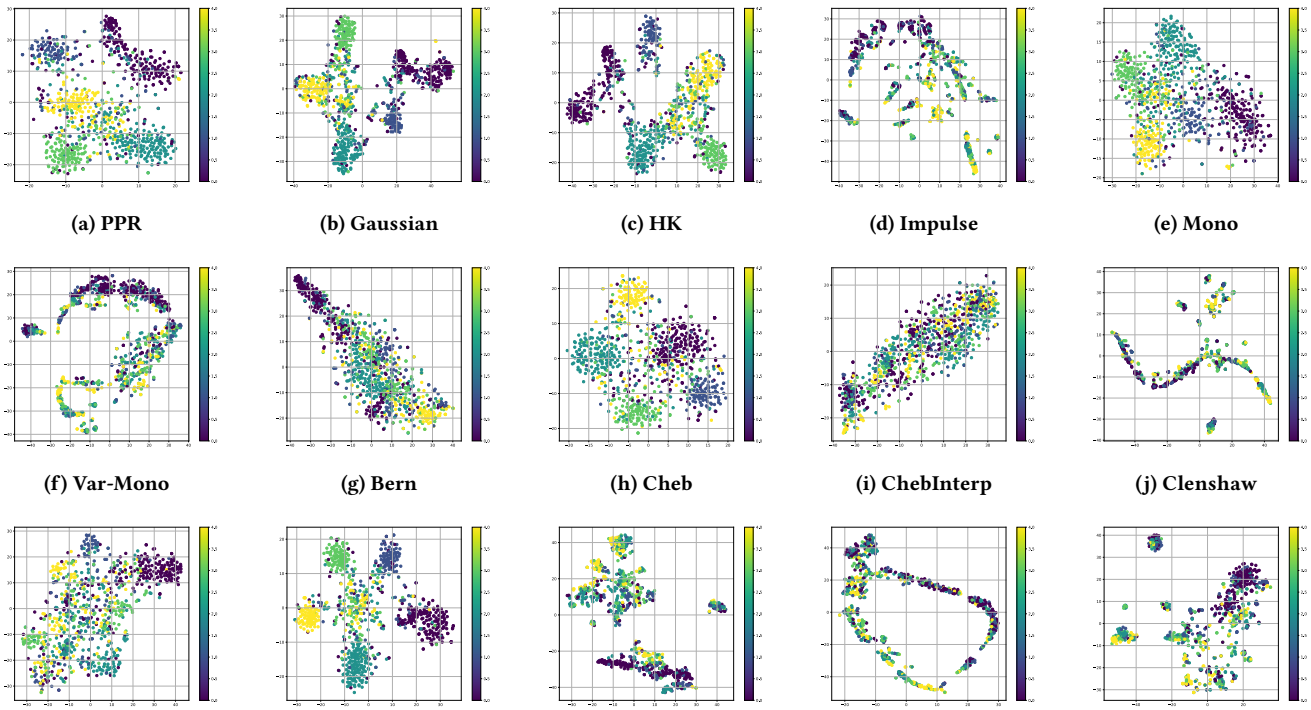


(a) PPR (b) Gaussian (c) HK (d) Impulse (e) Mono

(f) Var-Mono (g) Bern (h) Cheb (i) ChebInterp (j) Clenshaw

(k) Favard (l) Horner (m) Jacobi (n) Legendre (o) OptBasis

Figure 10: Clusters of different filters on CORA.



(a) PPR (b) Gaussian (c) HK (d) Impulse (e) Mono

(f) Var-Mono (g) Bern (h) Cheb (i) ChebInterp (j) Clenshaw

(k) Favard (l) Horner (m) Jacobi (n) Legendre (o) OptBasis

Figure 11: Clusters of different filters on CHAMELEON.

D.3 Degree-wise Accuracy Gap

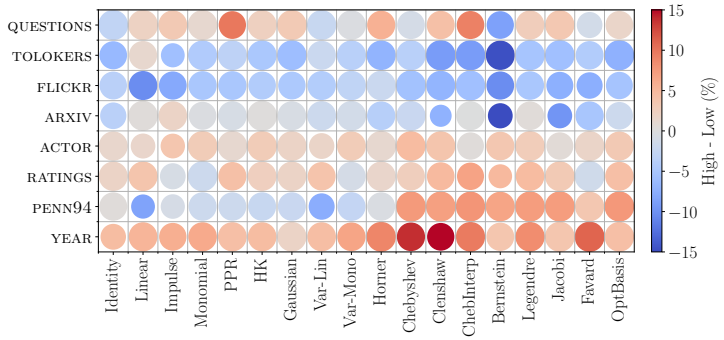
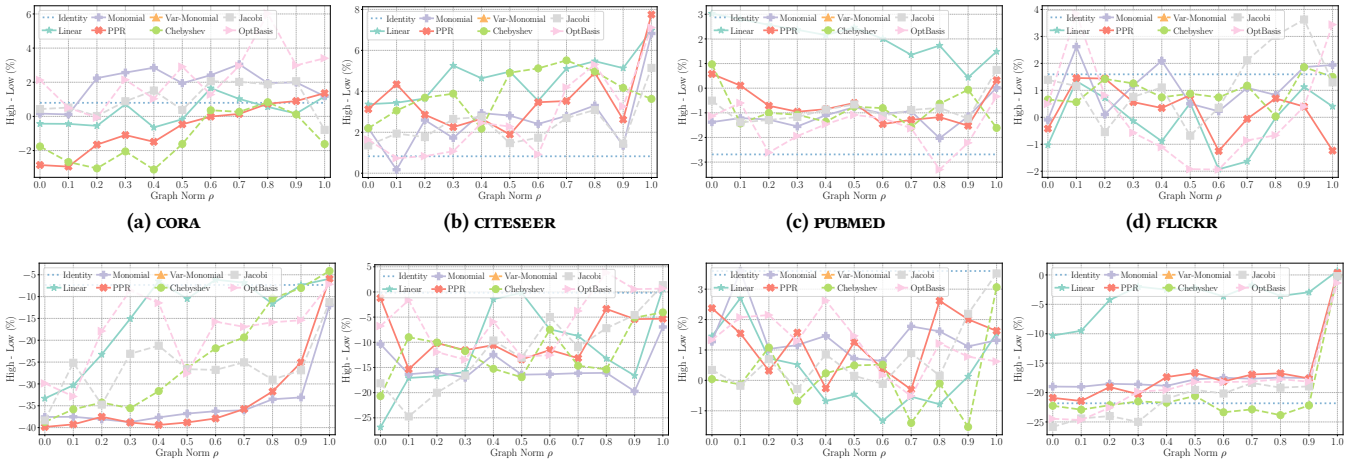


Figure 12: Accuracy gap between high- and low-degree nodes on remaining datasets.

D.4 Effect of Graph Normalization on Degree-wise Performance



(a) CORA (b) CITESEER (c) PUBMED (d) FLICKR

(e) SQUIRREL (f) CHAMELEON (g) ACTOR (h) ROMAN

Figure 13: Effect of *graph normalization* ρ on the accuracy difference between high- and low-degree nodes of selected full-batch fixed and variable filters on 4 homophilous and 4 heterophilous datasets.