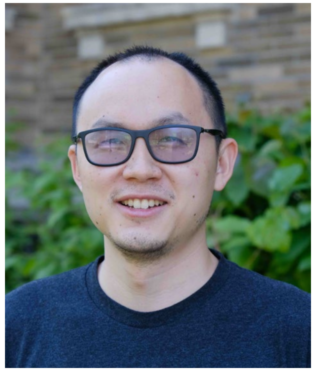
Specformer: Spectral Graph Neural Networks Meet Transformers

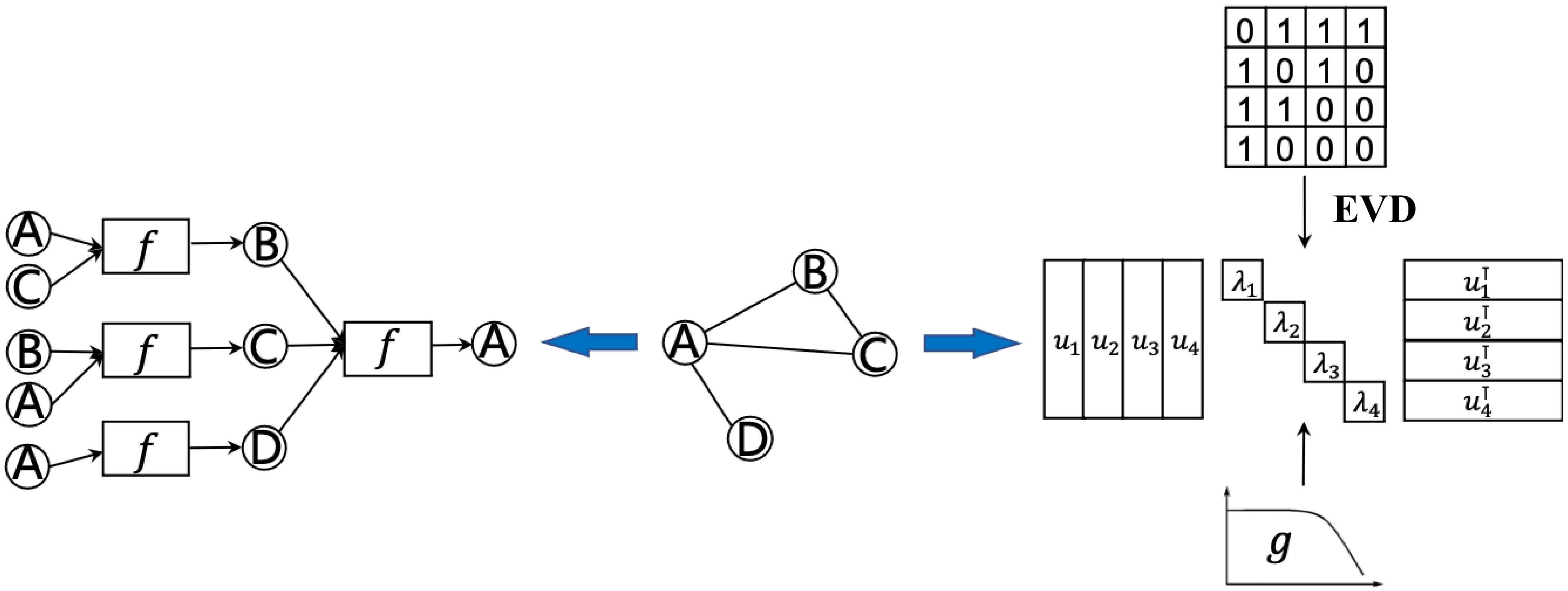
Deyu Bo[1], Chuan Shi[1], Lele Wang[2], Renjie Liao[2]

Beijing University of Posts and Telecommunications[1]
University of British Columbia[2]



Spatial & Spectral GNNs



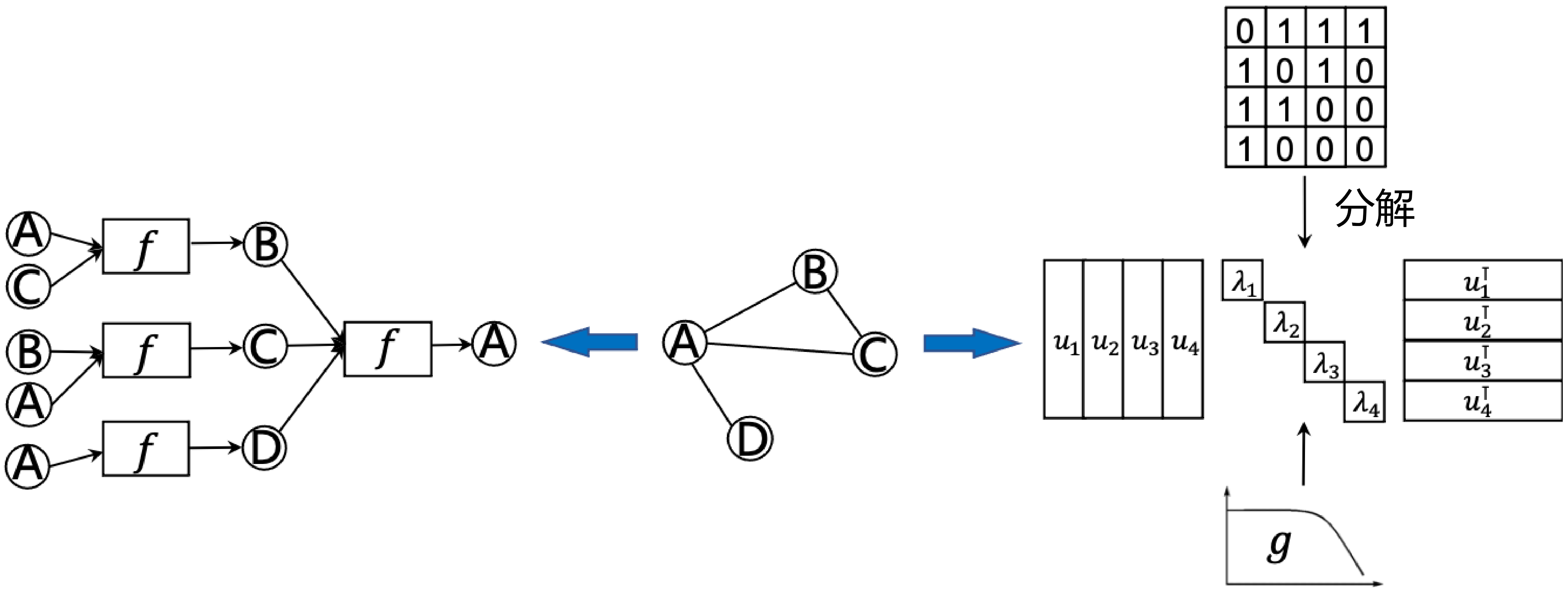
f: Aggregation

Spatial GNNs

g: Filtering

Spectral GNNs

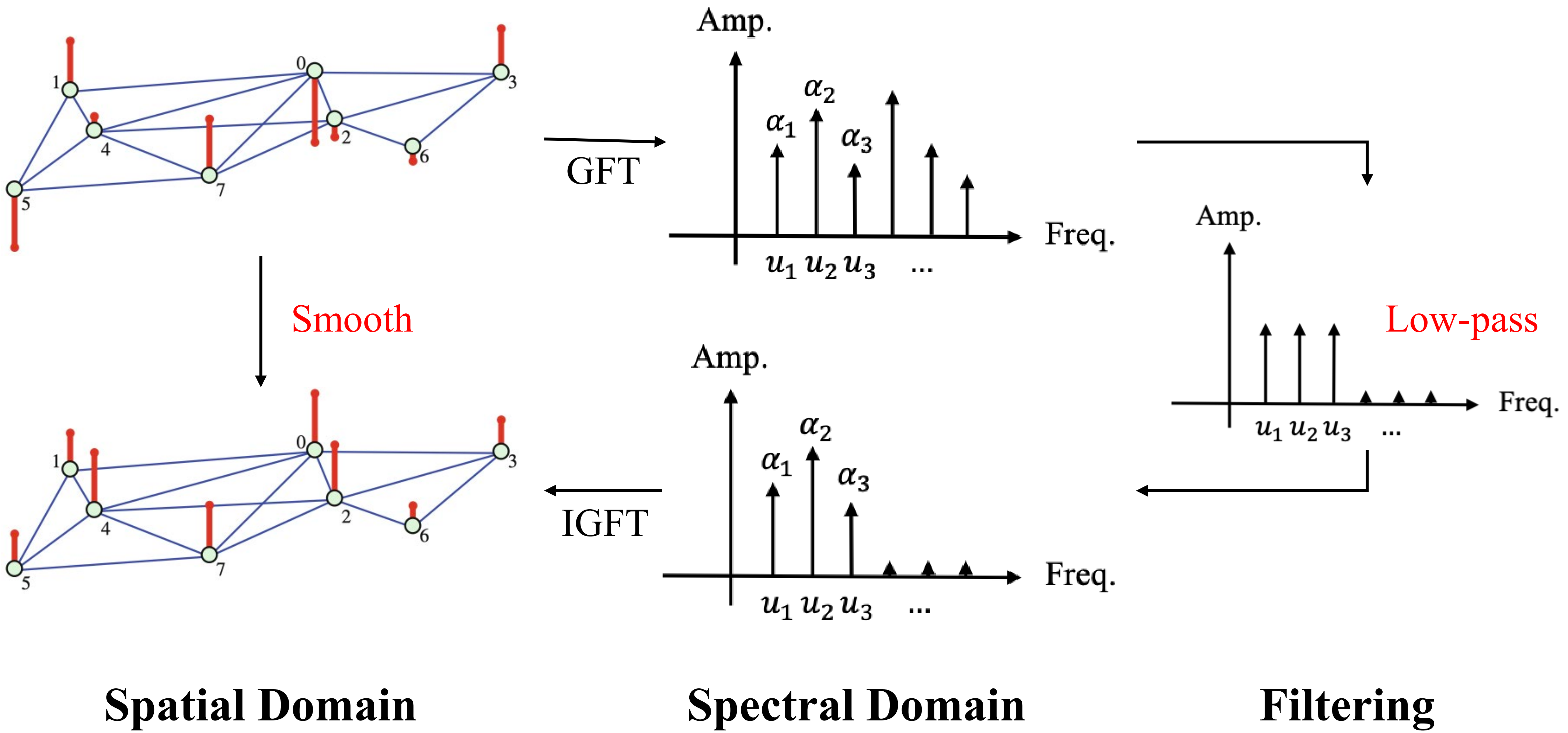
Spatial & Spectral GNNs



空域图神经网络 **图数据** **谱域图神经网络**

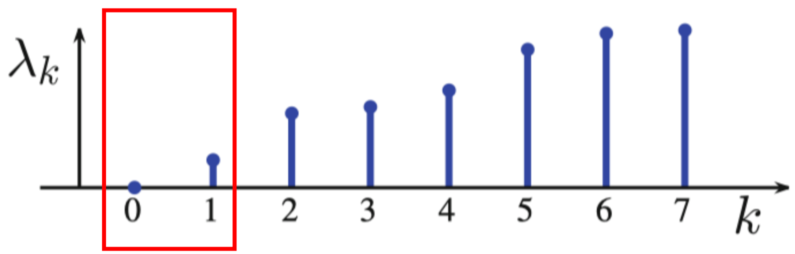
f: 聚合函数 g: 滤波函数

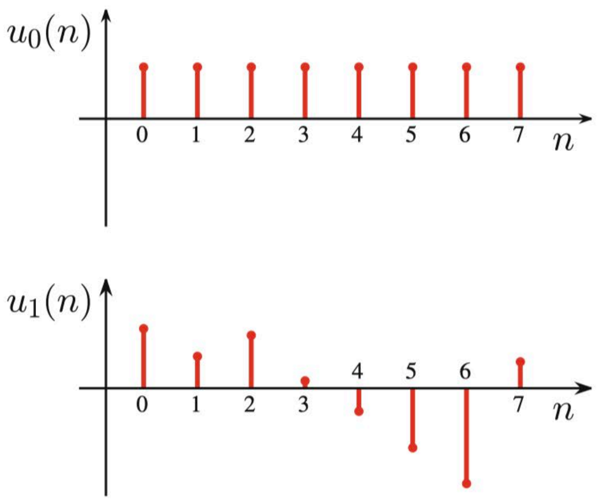
Graph Signal Processing

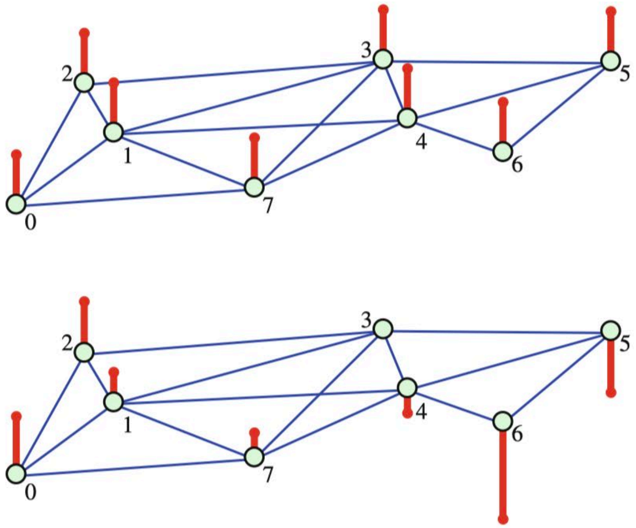


Eigenvalue Decomposition (EVD)

- $L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ is the normalized graph Laplacian matrix
- $L = U \Lambda U^{\intercal},\ \lambda_i \in [0, 2],\ \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$
- $u_i \in \mathbb{R}^{n*1},\ u_i^{\intercal} u_j = \begin{cases} 0, i \ne j \\ 1, i = j \end{cases}$ **Orthogonal & normalized**
- $\lambda_i = u_i^{\intercal} L u_i = \sum_{(p,q \in E)} (u_i(p) - u_i(q))^2$ **Total variation (Frequency)**





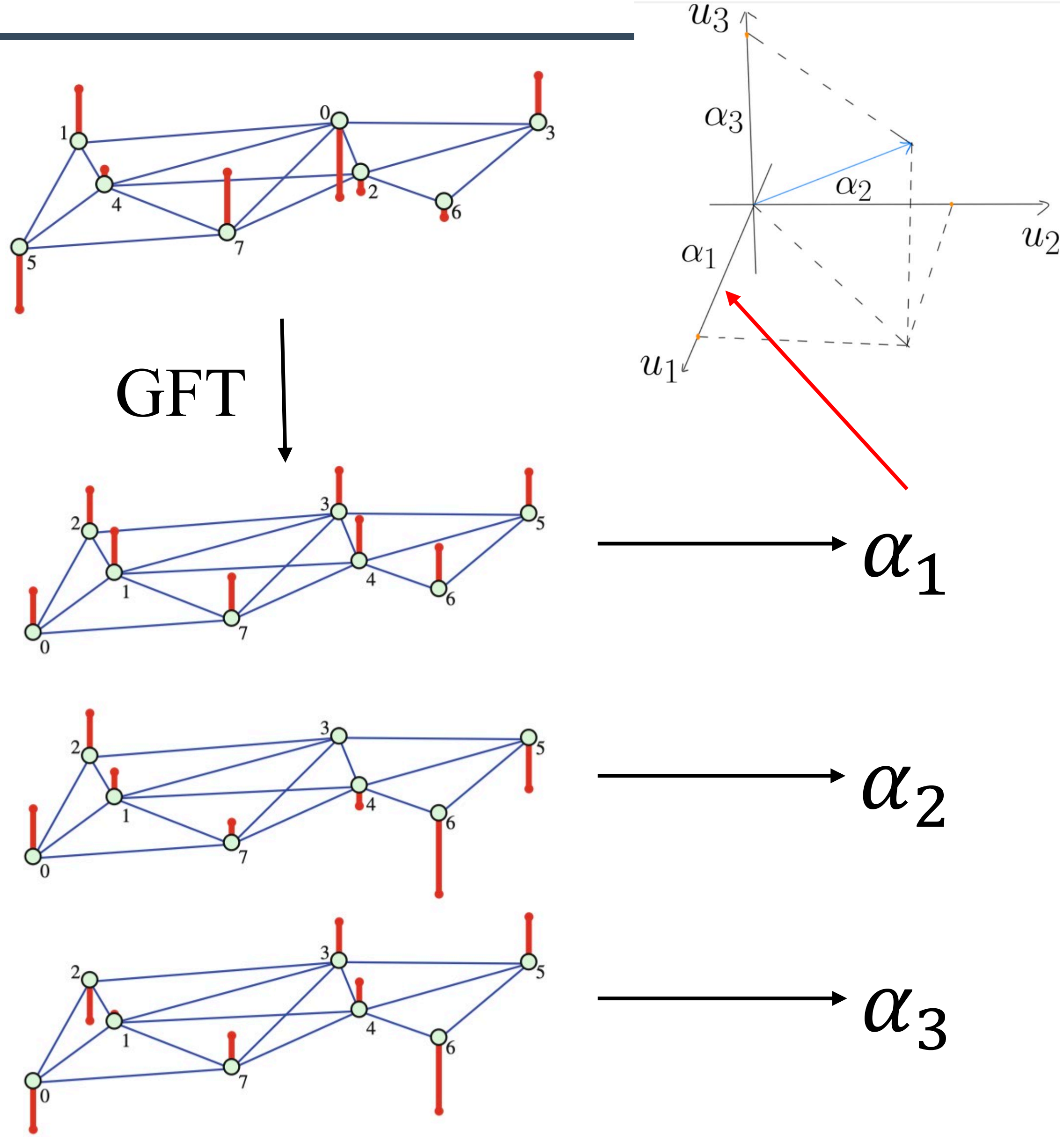


Graph Signal

- Given a graph $G = (V, E, X)$
  - $|V| = n$ nodes
  - Each node i has a scalar signal $x(i)$
  - Graph signal $x = [x(1), \dots, x(n)]^{\top}$

- Eigenvectors are special graph signals
  - $u_i \in \mathbb{R}^{n*1}$
  - u_i is orthogonal and normalized
  - u_i has different frequencies (TV)

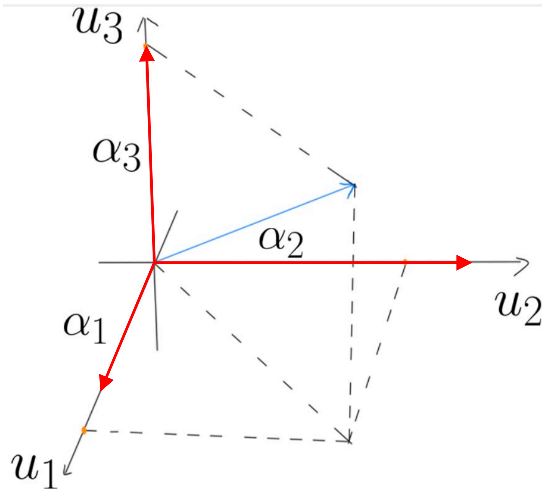
- Graph signal can be represented by U
  - $x = \sum_{i=1}^{n} \alpha_i * u_i$
  - $\alpha_i = < x, u_i >$
  - Graph Fourier Transform (GFT)

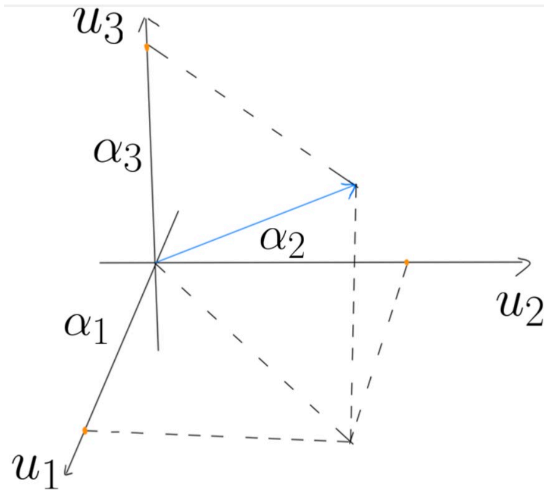


Graph Signal Processing

$$\begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix} * \begin{bmatrix} \alpha_1' \\ \alpha_2' \\ \alpha_3' \\ \alpha_4' \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix} \odot \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} u_1^\intercal \\ u_2^\intercal \\ u_3^\intercal \\ u_4^\intercal \end{bmatrix} * \begin{bmatrix} x \end{bmatrix}$$

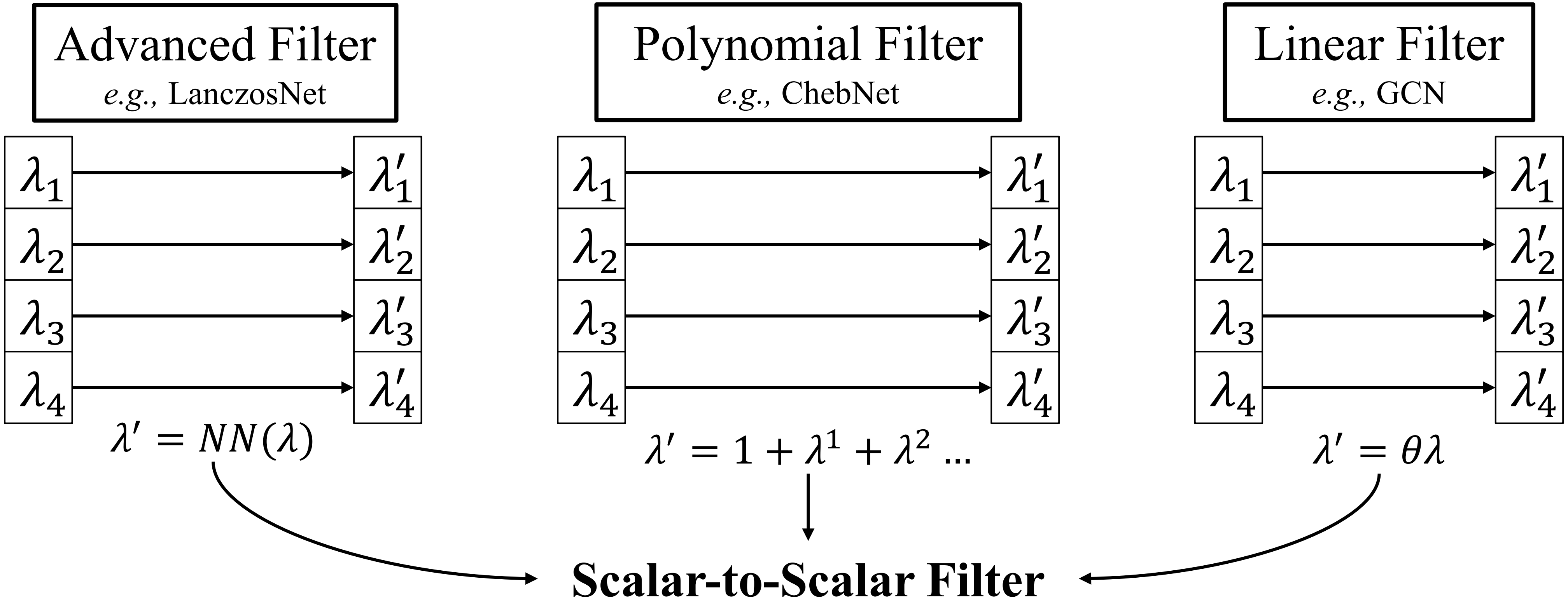
IGFT — **Filtering: $g(\lambda)$** — **Graph Fourier Transform**





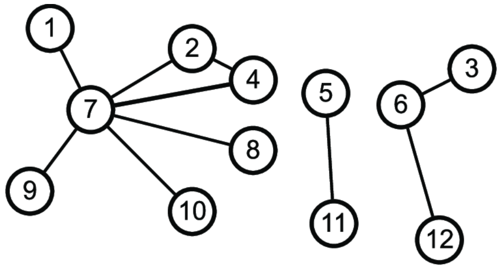
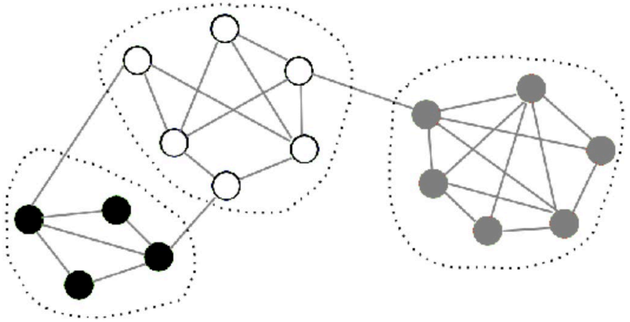
Motivation

- Taxonomy of Spectral GNNs



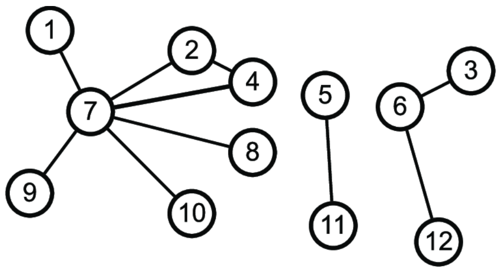
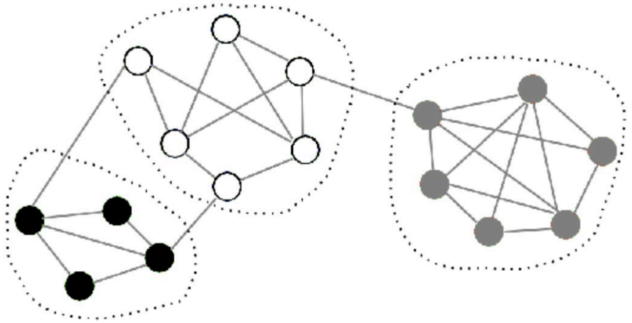
A Survey on Spectral Graph Neural Networks. Bo et al.

Motivation

Spectrum Information	Example	Definition	Scalar Input	Set Input
Algebraic Connectivity		$\mathrm{Count}(\lambda = 0)$	✗	✓
Diameter		$[\frac{4}{n\lambda_2}, \frac{1}{2m\lambda_1}]$	✗	✓
Clusterability		$\lambda_2 - \lambda_1$ $(\lambda_1 \neq \lambda_2 \neq 0)$	✗	✓

Spectral graph theory. Fan RK Chung and Fan Chung Graham

Motivation

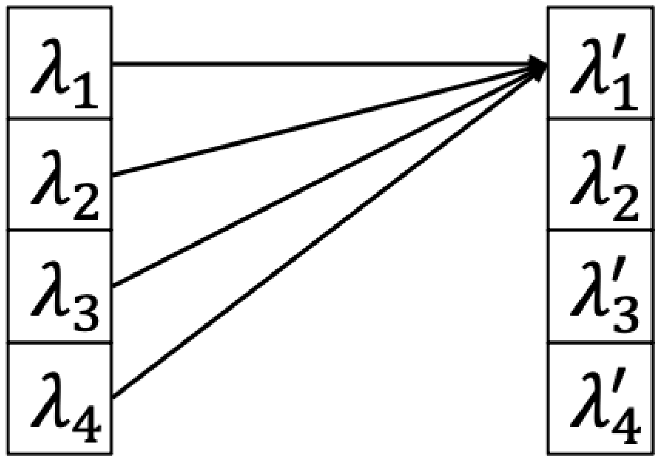
Spectrum Information	Example	Definition	Scalar Input	Set Input
Algebraic Connectivity		$\text{Count}(\lambda = 0)$	✗	✓
Clusterability		$\lambda_2 - \lambda_1$ $(\lambda_1 \neq \lambda_2 \neq 0)$	✗	✓

Spectral graph theory. Fan RK Chung and Fan Chung Graham

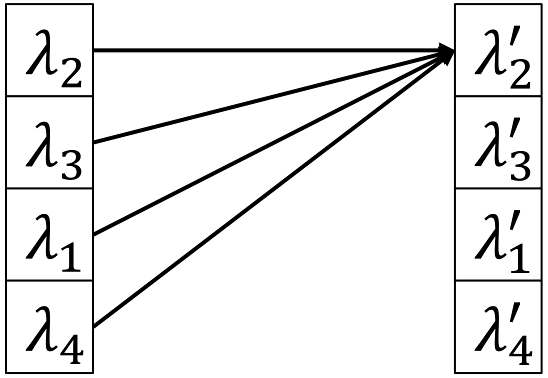
Motivation

- ☐ Set-to-Set Graph Filter

$$\lambda' = Set\ Encoder(\boldsymbol{\Lambda})$$



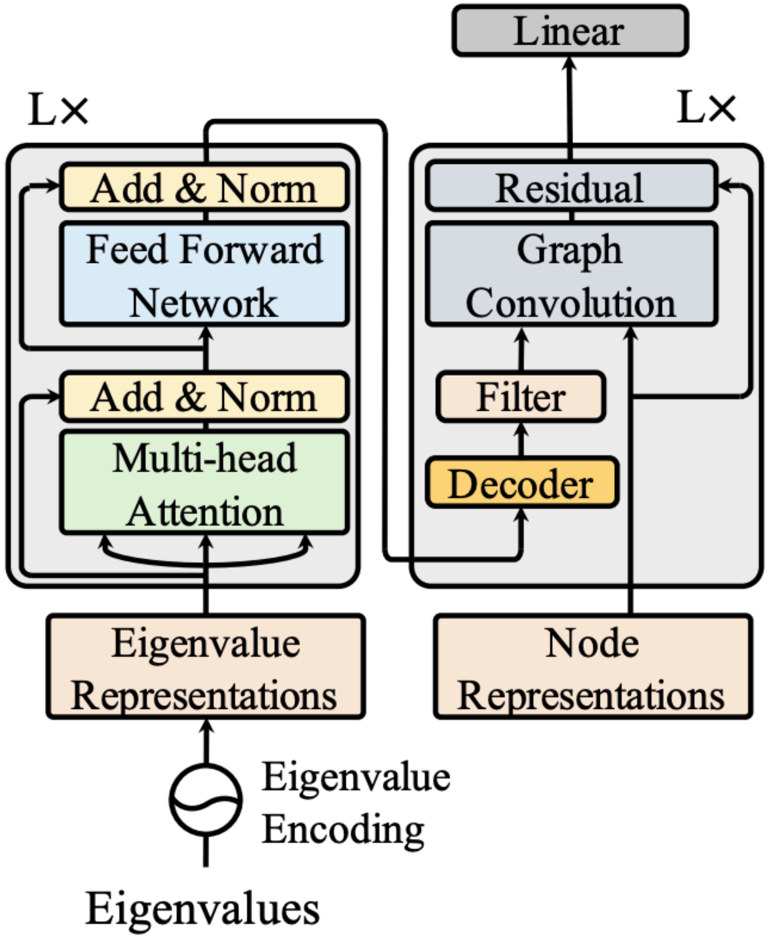
Permutation-invariant



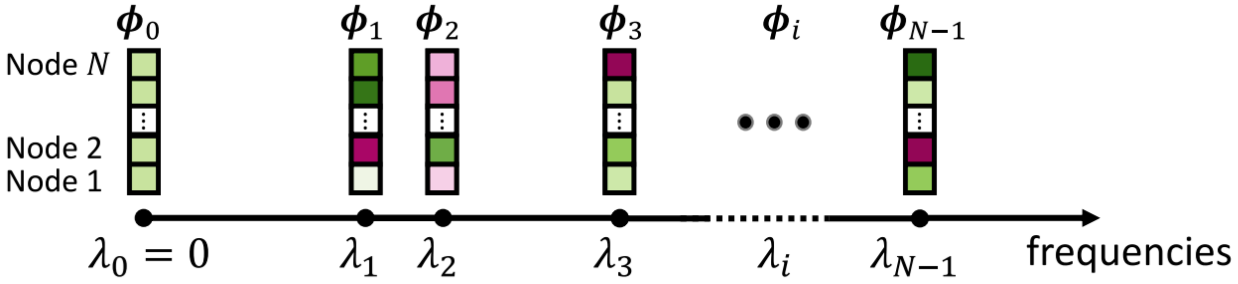
Relative Information

$$f(\lambda_2) - f(\lambda_1) = f(\lambda_2 - \lambda_1)$$

Specformer



- Eigenvalues are the coordinates on the frequency axis



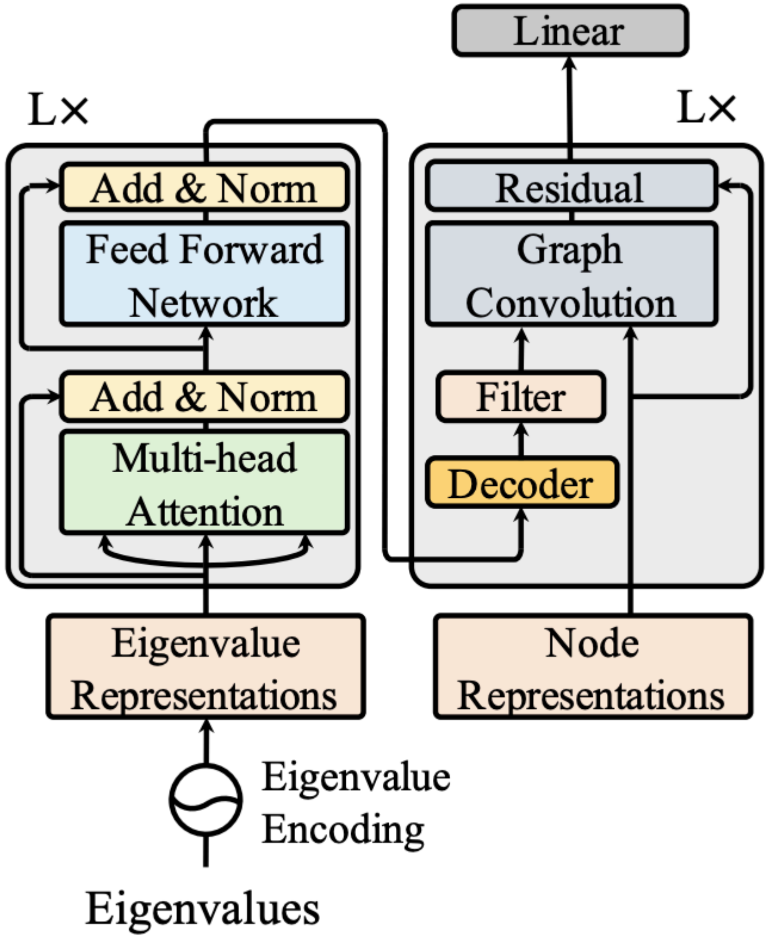
- Eigenvalue Encoding (Relative Information)

$$\rho(\lambda, 2i) = \sin\left(\epsilon\lambda/10000^{2i/d}\right)$$

$$\rho(\lambda, 2i+1) = \cos\left(\epsilon\lambda/10000^{2i/d}\right)$$

Rethinking graph transformers with spectral attention. Kreuzer et al.

Specformer

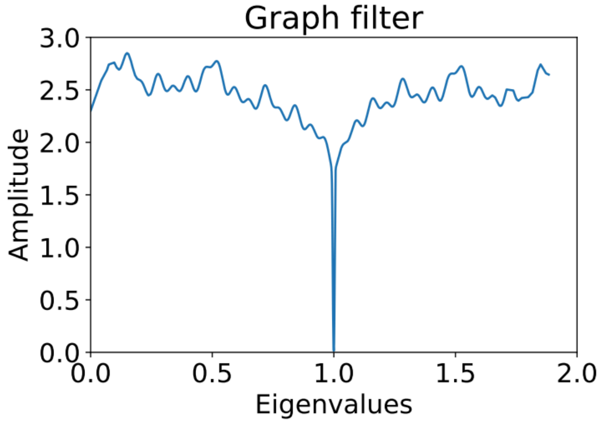


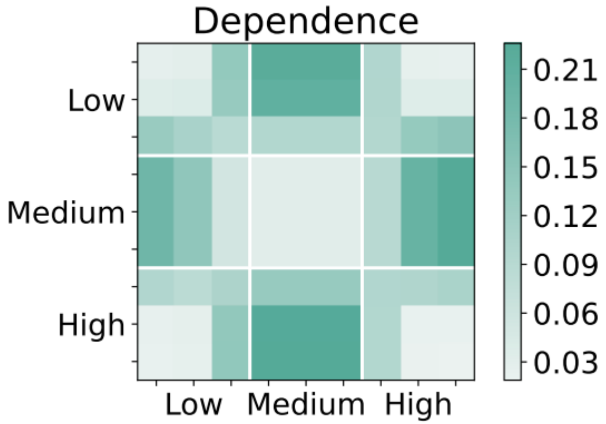
- Transformer Encoder (Permutation-invariant)

$$\tilde{\boldsymbol{Z}} = \mathrm{MHA}(\mathrm{LN}(\boldsymbol{Z})) + \boldsymbol{Z},$$
$$\hat{\boldsymbol{Z}} = \mathrm{FFN}(\mathrm{LN}(\tilde{\boldsymbol{Z}})) + \tilde{\boldsymbol{Z}}$$

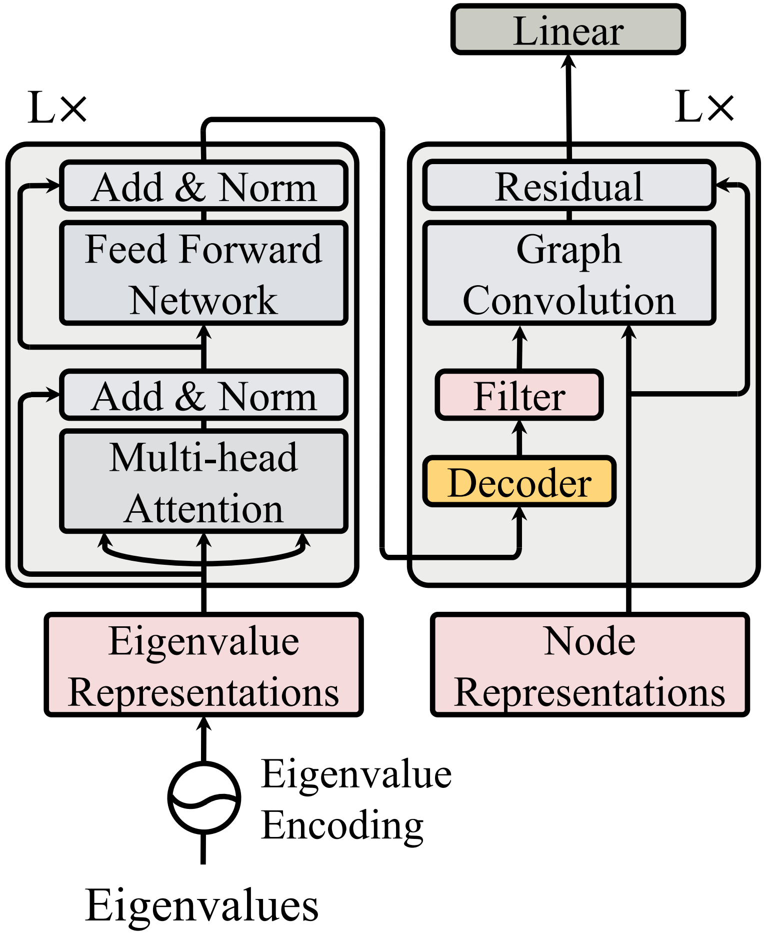
- Decoder (Non-local)

$$\boldsymbol{Z}_m = \mathrm{Attn}(\boldsymbol{Q}_m, \boldsymbol{K}_m, \boldsymbol{V}_m), \boldsymbol{\lambda}_m = \phi(\boldsymbol{Z}_m \boldsymbol{W}_\lambda)$$

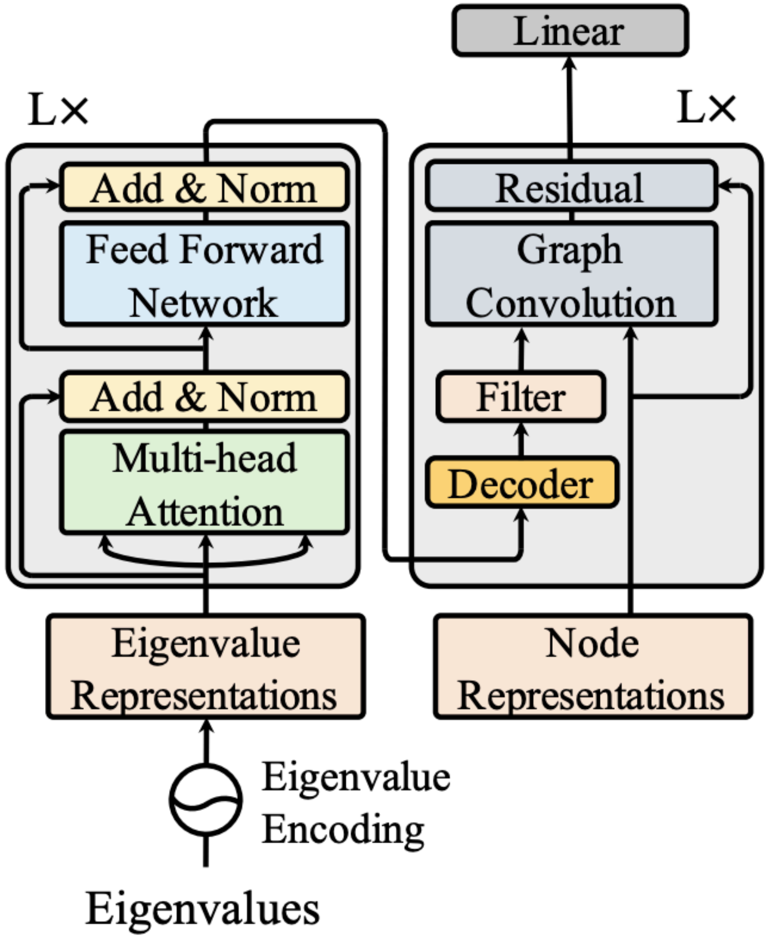




Specformer


Linear
L×
L×
Add & Norm
Feed Forward Network
Add & Norm
Multi-head Attention
Residual
Graph Convolution
Filter
Decoder
Eigenvalue Representations
Node Representations
Eigenvalue Encoding
Eigenvalues

Specformer



- Generalizing univariate functions

$$\rho(\lambda)\boldsymbol{w} = w_0\lambda + \sum_{i=1}^{d/2} w_{2i}\sin\left(\frac{\epsilon\lambda}{10000^{2i/d}}\right) + \sum_{i=1}^{d/2} w_{2i-1}\cos\left(\frac{\epsilon\lambda}{10000^{2i/d}}\right)$$

Fourier Series

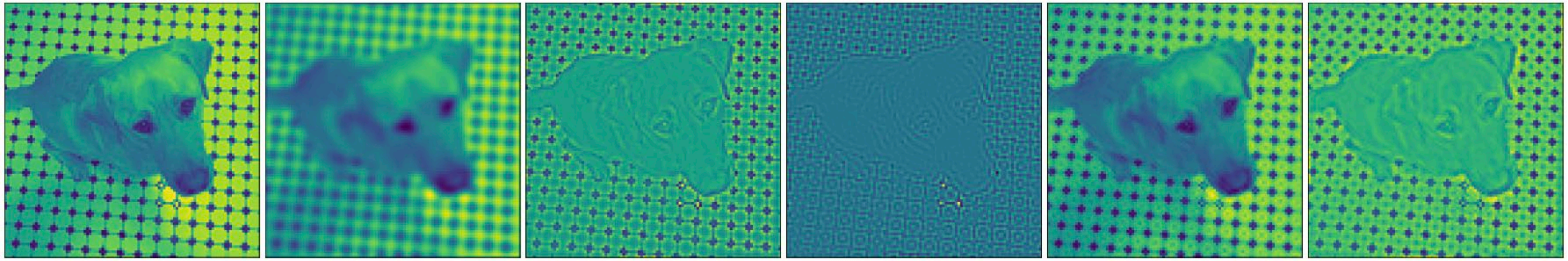
- Approximating multivariate functions

$$f(x_1,\ldots,x_M) = \rho\left(\sum_{m=1}^{M} \lambda_m \phi(x_m)\right)$$

Kolmogorov–Arnold Theorem

Deep sets. Zaheer et al.

Experiment



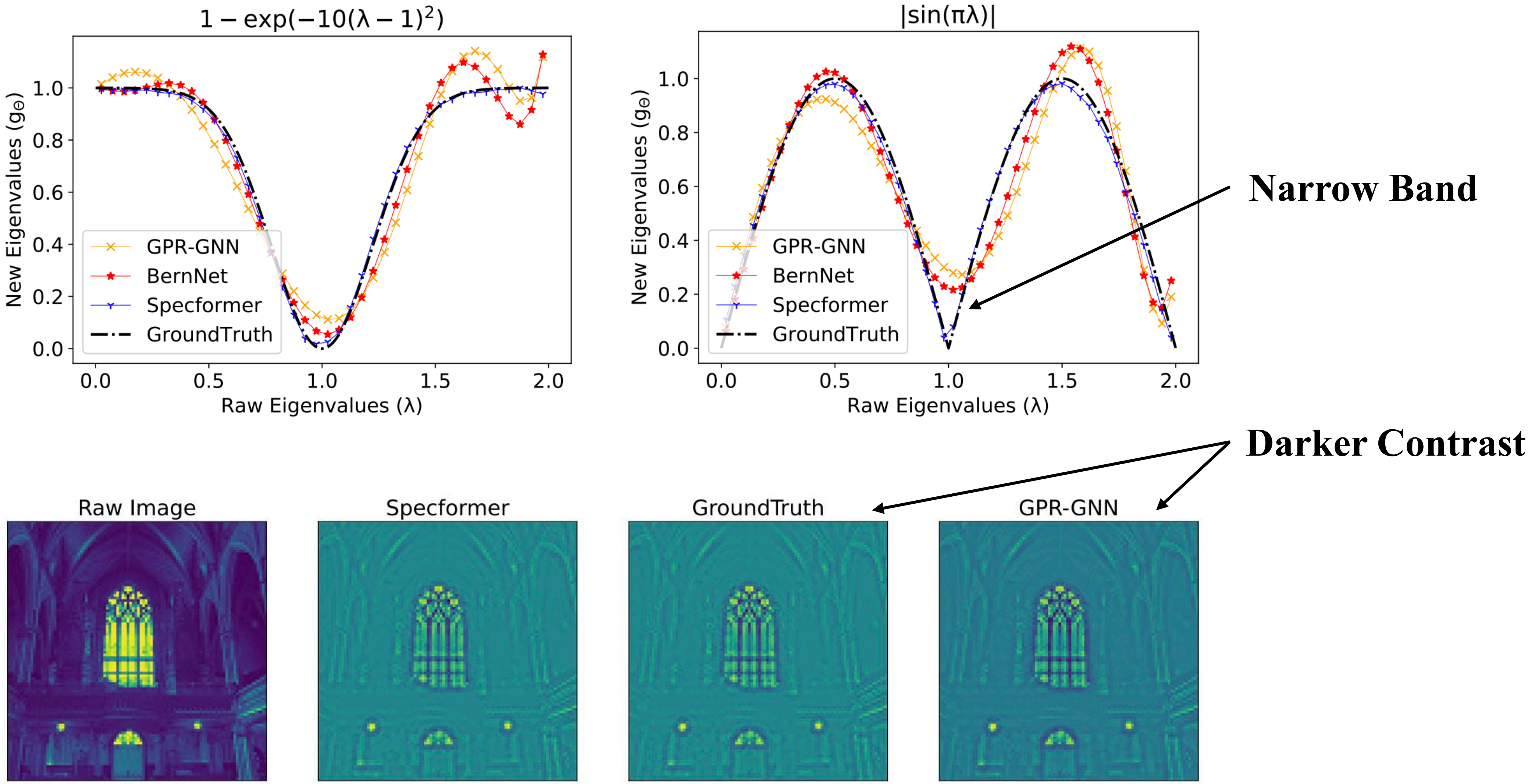
(a) Original (b) Low-pass (c) High-pass (d) Band-pass (e) Band-rejection (f) Comb

Table 1: Node regression results, mean of the sum of squared error & R^2 score, on synthetic data.

Model (~2k param.)	Low-pass	High-pass	Band-pass	Band-rejection	Comb
	$\exp(-10\lambda^2)$	$1-\exp(-10\lambda^2)$	$\exp(-10(\lambda-1)^2)$	$1-\exp(-10(\lambda-1)^2)$	$\lvert\sin(\pi\lambda)\rvert$
GCN	3.4799(.9872)	67.6635(.2364)	25.8755(.1148)	21.0747(.9438)	50.5120(.2977)
GAT	2.3574(.9905)	21.9618(.7529)	14.4326(.4823)	12.6384(.9652)	23.1813(.6957)
ChebyNet	0.8220(.9973)	0.7867(.9903)	2.2722(.9104)	2.5296(.9934)	4.0735(.9447)
GPR-GNN	0.4169(.9984)	0.0943(.9986)	3.5121(.8551)	3.7917(.9905)	4.6549(.9311)
BernNet	0.0314(.9999)	0.0113(.9999)	0.0411(.9984)	0.9313(.9973)	0.9982(.9868)
JacobiConv	0.0003(.9999)	0.0064(.9999)	0.0213(.9999)	0.0156(.9999)	0.2933(.9995)
Specformer	**0.0002(.9999)**	**0.0026(.9999)**	**0.0017(.9999)**	**0.0014(.9999)**	**0.0057(.9999)**

BernNet: Learning Arbitrary Graph Spectral Filters via Bernstein Approximation. He et al.

Experiment



(a) Image #19

Experiment

Table 2: Results on real-world node classification tasks. Mean accuracy (%) ± 95% confidence interval. * means re-implemented baselines. "OOM" means out of GPU memory.

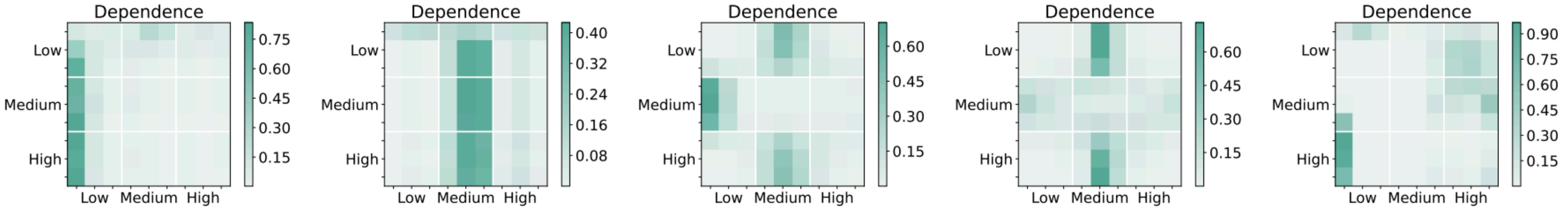
	Param. on Photo	Heterophilic				Homophilic			
		Chameleon	Squirrel	Actor	Penn94	Cora	Citeseer	Photo	arXiv
Spatial-based GNNs									
GCN	48K	59.61±2.21	46.78±0.87	33.23±1.16	82.47±0.27	87.14±1.01	79.86±0.67	88.26±0.73	71.74±0.29
GAT	49K	63.13±1.93	44.49±0.88	33.93±2.47	81.53±0.55	88.03±0.79	80.52±0.71	90.94±0.68	71.82±0.23
H_2GCN	60K	57.11±1.58	36.42±1.89	35.86±1.03	OOM	86.92±1.37	77.07±1.64	93.02±0.91	OOM
GCNII	49K	63.44±0.85	41.96±1.02	36.89±0.95	82.92±0.59	88.46±0.82	79.97±0.65	89.94±0.31	72.04±0.19
Spectral-based GNNs									
LanczosNet*	50K	64.81±1.56	48.64±1.77	38.16±0.91	81.55±0.26	87.77±1.45	80.05±1.65	93.21±0.85	71.46±0.39
ChebyNet	48K	59.28±1.25	40.55±0.42	37.61±0.89	81.09±0.33	86.67±0.82	79.11±0.75	93.77±0.32	71.12±0.22
GPR-GNN	48K	67.28±1.09	50.15±1.92	39.92±0.67	81.38±0.16	88.57±0.69	80.12±0.83	93.85±0.28	71.78±0.18
BernNet	48K	68.29±1.58	51.35±0.73	41.79±1.01	82.47±0.21	88.52±0.95	80.09±0.79	93.63±0.35	71.96±0.27
ChebNetII	48K	71.37±1.01	57.72±0.59	41.75±1.07	83.12±0.22	88.71±0.93	80.53±0.79	94.92±0.33	72.32±0.23
JacobiConv	48K	74.20±1.03	57.38±1.25	41.17±0.64	83.35±0.11	**88.98±0.46**	80.78±0.79	95.43±0.23	72.14±0.17
Graph Transformers									
Transformer*	37K	46.39±1.97	31.90±3.16	39.95±1.64	OOM	71.83±1.68	70.55±1.20	90.05±1.50	OOM
Graphormer*	139K	54.49±3.11	36.96±1.75	38.45±1.38	OOM	67.71±0.78	73.30±1.21	85.20±4.12	OOM
Specformer	32K	**74.72±1.29**	**64.64±0.81**	**41.93±1.04**	**84.32±0.32**	88.57±1.01	**81.49±0.94**	**95.48±0.32**	**72.37±0.18**

Experiment

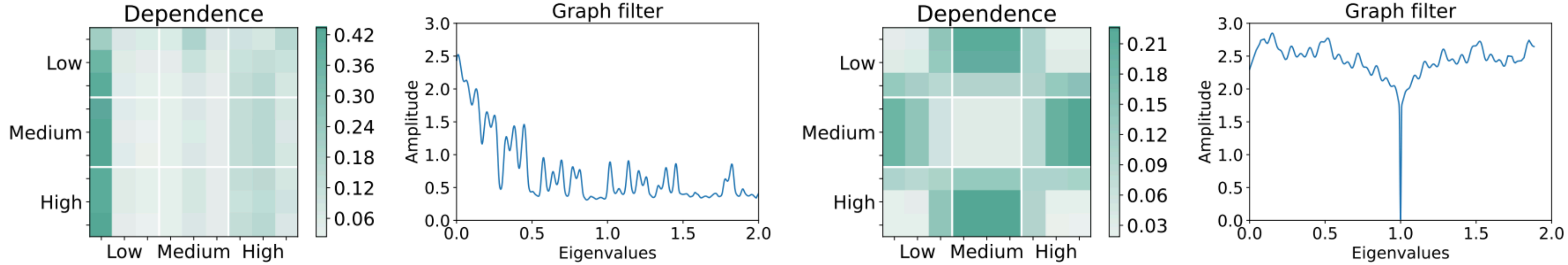
Table 3: Results on graph-level datasets. ↓ means lower the better, and ↑ means higher the better.

Model	ZINC(↓)	MolHIV(↑)	MolPCBA(↑)
GCN	0.367 ± 0.011	0.7599 ± 0.0119	0.2424 ± 0.0034
GIN	0.526 ± 0.051	0.7707 ± 0.0149	0.2703 ± 0.0023
GatedGCN	0.090 ± 0.001	-	0.267 ± 0.002
CIN	0.079 ± 0.006	**0.8094 ± 0.0057**	-
GIN-AK+	0.080 ± 0.001	0.7961 ± 0.0119	0.2930 ± 0.0044
GSN	0.101 ± 0.010	0.7799 ± 0.0100	-
DGN	0.168 ± 0.003	0.7970 ± 0.0097	0.2885 ± 0.0030
PNA	0.188 ± 0.004	0.7905 ± 0.0132	0.2838 ± 0.0035
Spec-GN	0.070 ± 0.002	-	0.2965 ± 0.0028
SAN	0.139 ± 0.006	0.7785 ± 0.0025	0.2765 ± 0.0042
Graphormer[2]	0.122 ± 0.006	0.7640 ± 0.0022	0.2643 ± 0.0017
GPS	0.070 ± 0.004	0.7880 ± 0.0101	0.2907 ± 0.0028
Specformer	**0.066 ± 0.003**	0.7889 ± 0.0124	**0.2972 ± 0.0023**

Experiment



(a) Low-pass (b) High-pass (c) Band-pass (d) Band-rejection (e) Comb



(a) Dependency of Citeseer (b) Filter of Citeseer (c) Dependency of Squirrel (d) Filter of Squirrel