FULLY SPECTRAL GRAPH SCATTERING NETWORKS WITH TRANSFORMER-BASED COMPONENT-ADAPTIVE GRAPH FILTER BANKS AND SPECTRAL NONLINEARITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph scattering networks (GSNs) learn multi-scale node representation via graph filter banks along multiple scattering paths of different lengths. However, existing GSNs are still challenged by two problems: **i)** spectral filtering via single graph filter or filter bank with uniform frequency response to different graph components, and **ii)** lack of spectral nonlinearity to ensure the invariance to node permutation. To tackle the issues, we propose a novel fully spectral graph scattering network name SpecGSN to allow invariant node representation learning with learnable spectral filtering and nonlinearity. Specifically, for spectral filtering, we design Transformer-based graph filter banks on the orthonormal subspaces induced by graph shift operators to achieve diverse frequency response to different graphs components. Moreover, we develop spectral nonlinear activation via subspace folding to guarantee the commutativity between the nonlinear operator and transform with subspace basis for invariant learnable filtering in the spectral domain. We theoretically demonstrate that SpecGSN is invariant to node permutations and subspace basis rotations. Experimental results show that SpecGSN achieves competitive or state-of-the-art node classification performance on both homophilic and heterophilic graphs with enhanced computation efficiency.

1 INTRODUCTION

Graph convolutional networks (GCNs) are prevailing graph neural networks (GNN) architectures for representation learning of non-Euclidean data such as graphs. They can be categorized into two paradigms: spatial GCNs that resort to message passing for representation (Veli$v$ković et al., 2018; Abu-El-Haija et al., 2019; Chami et al., 2019; Lei et al., 2022) and spectral GCNs that learn representations by spectral convolutional filtering with well designed spectral filters (Defferrard et al., 2016; Levie et al., 2019; He et al., 2021).

Graph scattering network (GSN) is a widely studied paradigm of spectral GCNs, which was first proposed in (Gama et al., 2019a) as an extension of group invariant scattering (Mallat, 2012; Bruna & Mallat, 2013) for Euclidean signal processing. Graph scattering networks with wavelet frames simultaneously inherit the theoretical properties of energy preservation and stability of scattering convolutional networks (Gama et al., 2019b), and satisfy permutation covariant covariance (Zou & Lerman, 2020). Compare with spectral GCNs relying on a single filter, GSNs can enhance feature diversity and model expressiveness. GSNs introduce the modulus operation in each layer to shift frequency components and overcome the limitation of missing frequency issue claimed for linear spectral GNNs (Wang & Zhang, 2022). Their hierarchical tree structure also enables various cascading arrangements of filters to realize a set of band-pass filters for distinct frequency bands.

Similar to spectral GCNs, graph filter bank design is highly related to the performance of GSNs. Recent works on GSNs explore more generalized filter banks (Tong et al., 2021; Koke & Kutyniok, 2022; Liu et al., 2026) to substitute the wavelet framework, but are challenged by two problems. **First,** existing methods based on Chebyshev polynomials (Defferrard et al., 2016), fractional polynomials (Levie et al., 2019; Bianchi et al., 2022), Bernstein polynomials (He et al., 2021), and

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.

Transformers (Bo et al., 2023) resort to a single filter bank for graphs potentially containing multiple unconnected graph components, and obtain the same frequency response to different graph components with numerically identical eigenvalues. **Second,** spectral nonlinear activations cannot ensure the invariance to node permutation due to the lack of commutative property. Graph signals are frequently transformed between the spatial and spectral domains to achieve spectral filtering and nonlinear activation in GSNs. This causes computation overhead and could be prohibitive for large-scale graphs.

To address the problems, in this paper, we propose a novel fully spectral graph scattering network named SpecGSN that leverages Transformer-based graph filter bank on orthonormal subspaces induced by graph shift operators to distinguish different graph components and achieves spectral nonlinear activation with subspace folding. SpecGSN simultaneously guarantees the invariance to node permutation and subspace basis rotations in theory, and achieves competitive or state-of-the-art performance in node classification on both homophilic and heterophilic graphs. Our contributions are summarized below.

- We propose a novel fully spectral graph scattering network with Transformer-based graph filter banks and spectral nonlinear activations to allow node representation learning with spectral filtering and nonlinearity.
- We design Transformer-based graph filter banks on orthonormal subspaces of graph shift operators to achieve diverse frequency response to different graphs components.
- We develop spectral nonlinear activation via subspace folding to guarantee the commutativity between the nonlinear operator and transform with subspace basis for invariant learnable filtering in the spectral domain.
- We demonstrate theoretically that the proposed SpecGSN satisfies the invariance to node permutation and subspace basis rotations.

2 Related Work

Scattering convolutional networks is first developed via Morlet wavelets for image analysis (Mallat, 2012). They can capture multi-directional and multi-scale features with multi-scale directional filter banks and is highly suitable for representation learning of texture. Moreover, scattering convolutional networks also enjoy the properties such as energy convergence and stability to perturbations. Based on graph wavelets (Hammond et al., 2011), Gama et al. (2019a) extend scattering transforms to graphs with diffusion wavelets (Coifman & Maggioni, 2006). Gao et al. (2019) adopt random walk matrix as graph shift operator to achieve geometric scattering. Gama et al. (2019b) and Zou & Lerman (2020) prove that graph scattering transform with wavelet frame is stable to signal and topology perturbation and the node-wise representation is permutation covariant. The frame theorem and Lipschitz continuity of representation of graph scattering transform is further improved in (Chew et al., 2022). Since GSNs can generate abundant band-pass features, Min et al. (2020) combine GSNs with GCNs to overcome the over-smoothing issue in node representation learning. Min et al. (2021) leverage the attention mechanism (Veli$v$ković et al., 2018) to achieve multi-scale representation by aggregating the band-pass features from GSNs and low-pass features from diffusion-based GCNs. Pan et al. (2021) further realize spatio-temporal GSNs by considering diffusion wavelets on spatial graph and temporal graph.

Later works on GSNs aims to extend the wavelet frame to more generalized semi-discrete frames (Wiatowski & Bölcskei, 2015). Koke & Kutyniok (2022) introduce cosine and sine functions to divide the signal into high-frequency and low-frequency components, and utilize identity function as the readout filter. Cosine and sine function with higher frequency provides a spectral refinement of sine-cosine filters with lower frequency. Inspired by the scattering networks with learnable scale parameters (Gauthier et al., 2022), Tong et al. (2021; 2024) develop the first learnable graph scattering network with adaptive filter banks via the mixture of graph shift operators at multiple diffusion scales. Liu et al. (2026) introduce more generalized semi-discrete frame including diffusion wavelets and quadratic B-spline functions (Liu et al., 2025) as kernel functions for adaptive filter bank design, and further develop the stability guarantees for GSNs with generalized learnable filters via semi-discrete frames.

However, existing GSNs, along with spectral GNNs, usually adopt a single filter or filter bank to achieve representation learning for graphs of multiple graph components. Moreover, GSNs require modulus activation for each spectral convolution and induce frequent transformation between the spectral and spatial domains. Although some filter approximation methods (Defferrard et al., 2016; Zhang & Sun, 2024) have been developed to avoid GFT and its inverse transform, such transformations remain inevitable for generalized GSNs via semi-discrete frame that consider the dependencies among multiple eigenvalues (Bo et al., 2023).

3 PRELIMINARIES

3.1 GRAPH SIGNAL PROCESSING

Denote $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{A})$ as an undirected graph where $\mathcal{V}$ is the node set, $\mathcal{E}$ is the edge set, and $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ is the adjacency matrix whose element $\boldsymbol{A}_{ij}$ indicates the connectivity of vertices v_i and v_j in $\mathcal{V}$. Let $\boldsymbol{D}$ be the degree matrix such that its i-th diagonal element $\boldsymbol{D}_{ii} = d_i = \sum_j \boldsymbol{A}_{ij}$. The symmetric normalized graph Laplacian $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$ has eigenvalue decomposition $\boldsymbol{L} = \boldsymbol{U}\Lambda\boldsymbol{U}^T$, where $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_N]$ is the matrix of eigenvectors, and $\Lambda = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_N)$ is the diagonal matrix with the eigenvalue $\lambda_i \in [0, 2]$ corresponding to $\boldsymbol{u}_i$. Given graph signal $\boldsymbol{x}$ on $\mathcal{G}$, the Graph Fourier Transform (GFT) $\mathcal{F}\boldsymbol{x} = \hat{\boldsymbol{x}} = \boldsymbol{U}^T\boldsymbol{x}$ and the inverse graph Fourier Transform (iGFT) $\mathcal{F}^{-1}\hat{\boldsymbol{x}} = \boldsymbol{x} = \sum_{i=1}^{N} \hat{\boldsymbol{x}}[i]\boldsymbol{u}_i$, where $\hat{\boldsymbol{x}}[i]$ denotes the i-th element of $\hat{\boldsymbol{x}}$. The spectral convolution of graph signal $\boldsymbol{x}$ is achieved using a graph filter h defined on the spectral domain with frequency response $h(\lambda)$ for $\lambda \in [0, 2]$.

$$h \star_{\mathcal{G}} \boldsymbol{x} = \mathcal{F}^{-1}\{h(\lambda_i) \cdot \hat{\boldsymbol{x}}[i]\}_{i=1}^{N} \tag{1}$$

3.2 GRAPH SCATTERING NETWORKS

Frequency response of graph wavelet ψ (Hammond et al. (2011)) at scale t is defined as $\hat{\psi}_t(\lambda) = \hat{\psi}(2^t\lambda)$, and wavelet coefficient of graph signal $\boldsymbol{x}$ is $\psi_t \star \boldsymbol{x}$. Given a wavelet frame $\Phi = \{\psi_j, \phi_J\}_{j=0}^{J-1}$ with maximum scale J where ψ and ϕ are mother wavelet and scaling function, graph scattering transform with wavelet frame (Zou & Lerman (2020)) generates representation on scattering path $p = [p_1, p_2, \cdots, p_l]$ of length l by

$$\text{Scatter}[\mathcal{G}][p]\boldsymbol{x} = \phi_J \star |\psi_{p_l} \star \cdots |\psi_{p_2} \star |\psi_{p_1} \star \boldsymbol{x}|| \cdots |. \tag{2}$$

Particularly, scattering with path of zero length is defined as $\text{Scatter}[\mathcal{G}][\emptyset]\boldsymbol{x} = \phi_J \star \boldsymbol{x}$. The scattering paths of an M-layer graph scattering network form a set $\Omega = \bigcup_{l=0}^{M-1} \mathcal{P}_l$ where $\mathcal{P}_l = \{0, 1, \cdots, J-1\}^l$ is the set of all scattering paths of length l. GSN achieves multi-scale feature representation of the graph signal $\boldsymbol{x}$ by aggregating the features along all scattering paths. Generalized graph scattering networks (Koke & Kutyniok (2022); Tong et al. (2021); Liu et al. (2026)) with semi-discrete frame substitute the wavelet frame Φ to more generalized frame $\Gamma = \{\boldsymbol{h}_j, \boldsymbol{g}\}_{j=1}^{J}$ such that the scattering representation of path p is

$$\text{Scatter}[\mathcal{G}][p]\boldsymbol{x} = \boldsymbol{g} \star |\boldsymbol{h}_l \star \cdots |\boldsymbol{h}_2 \star |\boldsymbol{h}_1 \star \boldsymbol{x}|| \cdots | \tag{3}$$

with $\text{Scatter}[\mathcal{G}][\emptyset]\boldsymbol{x} = \boldsymbol{g} \star \boldsymbol{x}$. $\hat{\boldsymbol{h}}_j = [h_j(\lambda_1), h_j(\lambda_2), \cdots, h_j(\lambda_N)]$ and $\hat{\boldsymbol{g}} = [g(\lambda_1), g(\lambda_2), \cdots, g(\lambda_N)]$ where h_i and g can be parameterized to form adaptive graph filter banks.

4 SPECGSN

In this section, we present the proposed spectral graph scattering network named SpecGSN, as illustrated in Figure 1. SpecGSN leverages Transformer based graph filter bank with nonlinear spectral activation to achieve graph scattering transform in the spectral domain for learning multi-scale representation on multiple graph components. The C-channel input graph signal $\boldsymbol{X} \in \mathbb{R}^{N\times C}$ on $\mathcal{G}$ is first transformed with GFT to $\hat{\boldsymbol{X}}$ for representation learning with spectral graph scattering network. The spectral graph scattering transform with scattering path $p = [p_1, p_2, \cdots, p_l]$ is

$$\text{SpecScatter}[\mathcal{G}][p]\hat{\boldsymbol{X}} = \hat{\mathbf{g}} \odot \sigma_{spec}\left(\hat{\mathbf{h}}_{p_l} \odot \cdots \sigma_{spec}\left(\hat{\mathbf{h}}_{p_2} \odot \sigma_{spec}\left(\hat{\mathbf{h}}_{p_1} \odot \hat{\boldsymbol{X}}\right)\right)\cdots\right). \tag{4}$$

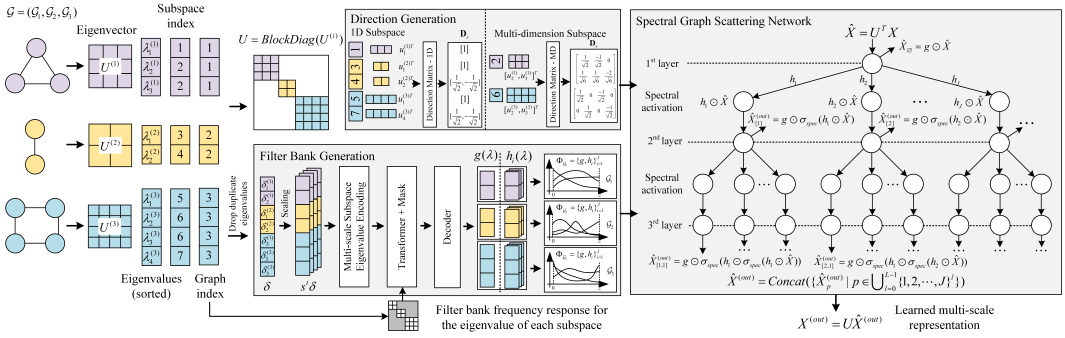


Figure 1: Overview of SpecGSN. Spectral decomposition of GSO is applied to each graph component. The eigenvalues and eigenvectors are further utilized in subspace division and direction matrix generation. Eigenvalues are aggregated within each subspace to generate component-adaptive filter banks. By adopting graph scattering, the spectrum of representation is formed by gathering the output spectrum of different scattering path p.

Here, $\odot$ is the element-wise product. Distinguishing from existing GSNs formulated in (3), we achieve Transformer based graph filter bank with learned frequency responses $\{\hat{\mathbf{h}}_j\}_{j=1}^{J}$ and $\hat{\mathbf{g}}$ for multiple graph components and spectral nonlinear activation σ_{spec} for well defined nonlinearity with commutativity in the spectral domain in (4). The spectrum of all scattering paths $p \in \Omega$ are collected to form the $\hat{\boldsymbol{X}}^{(out)} \in \mathbb{R}^{N\times C\times(1+J+\cdots+J^{L-1})}$ and converted to the node domain for the multi-scale representation $\boldsymbol{X}^{(out)} = \boldsymbol{U}\hat{\boldsymbol{X}}^{(out)}$.

Below, we first present graph shift operator decomposition and subspace division in Section 4.1. Subsequently, we design Transformer based graph filter bank for multiple graph components in Section 4.2 and develop spectral nonlinear activation in Section 4.3.

4.1 Graph Shift Operator Decomposition and Subspace Division

Spectral Decomposition of Graph Shift Operator: Consider $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ that contains n graph components $\{\mathcal{G}_i = \{\mathcal{V}^{(i)}, \mathcal{E}^{(i)}\}, i = 1, \cdots, n\}$, where $\mathcal{G}_i$ is the i-th graph component with vertex set $\mathcal{V}^{(i)}$ of N_i nodes and edge set $\mathcal{E}^{(i)}$. As illustrated in Figure 3 in Appendix A.6, different from default spectral decomposition of GSO that couples graph components, we propose spectral decomposition that separates subspaces of graph components to enable distinct graph filter banks for each graph component. We adopt the normalized symmetric graph Laplacian $\boldsymbol{L}^{(i)}$ as graph shift operator (GSO) with spectral decomposition $\boldsymbol{L}^{(i)} = \boldsymbol{U}^{(i)}\boldsymbol{\Lambda}^{(i)}\boldsymbol{U}^{(i)T}$. We sort the eigenvalues of $\mathcal{G}_i$ in the ascending order as $0 = \lambda_1^{(i)} \leq \cdots \leq \lambda_{N_i}^{(i)} \leq 2$, and rearrange the diagonal matrix of eigenvalues of $\mathcal{G}$ as $\boldsymbol{\Lambda} = \text{Diag}(\lambda_1^{(1)}, \cdots, \lambda_{N_1}^{(1)}, \cdots, \lambda_1^{(n)}, \cdots, \lambda_{N_n}^{(n)})$ for $\mathcal{G}_i$, $i = 1, \cdots, n$ with $\sum_{i=1}^{n} N_i = N$. The corresponding block-diagonal eigenvector matrix $\boldsymbol{U} = \text{BlockDiag}(\boldsymbol{U}^{(1)}, \cdots, \boldsymbol{U}^{(n)}) = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_N] \in \mathbb{R}^{N\times N}$.

Eigenspace Partition: Eigenspace partition is considered within each graph component $\mathcal{G}_i$ to group repeating eigenvalues that are numerically indistinguishable. We define the tolerance tol as the conservative upper bound on the numerical error of eigenvalue decomposition (*e.g.*, $tol = 10^{-5}$ for `float32`). For $\mathcal{G}_i$ with eigenvalues $0 = \lambda_1^{(i)} \leq \cdots \leq \lambda_{N_i}^{(i)} \leq 2$, we assign a local subspace index $s_k^{(i)}$ to $\lambda_k^{(i)}$ according to the gap between $\lambda_k^{(i)}$ and $\lambda_{k-1}^{(i)}$. Specifically, $s_1^{(i)} = 1$ for $\lambda_1^{(i)}$, and $s_k^{(i)} = 1$ with $k = 2, \cdots, N_i$ is recursively determined by $s_k^{(i)} = s_{k-1}^{(i)} + \mathbb{I}(|\lambda_k^{(i)} - \lambda_{k-1}^{(i)}| \geq tol)$, where $\mathbb{I}(\cdot)$ is the indicator function. Thus, $\mathcal{G}_i$ is decomposed into $M^{(i)} = \max_k s_k^{(i)}$ eigespaces.

For $\mathcal{G}$, the global index of k-th eigenspace in $\mathcal{G}_i$ is $\text{index}^{(i)}[k] = o^{(i)} + s_k^{(i)}$, where $o^{(i)} = \sum_{j=1}^{i-1} M^{(j)}$ is the cumulative index offset for $\mathcal{G}_i$, and the global vector $\text{index} = \text{concat}(\{\text{index}^{(i)}\}_{i=1}^{n})$ collects all the eigenspace indices. Note that eigenvalues of different graph components are assigned with

distinct eigenspaces indices, even if they are numerically the same. Thus, the r-th subspace of $\mathcal{G}$ is defined as span$(\{\boldsymbol{u}_k|k \in \mathcal{S}_r\})$ with the eigenvectors identified by the index set $\mathcal{S}_r = \{k|\text{index}[k] = r\}$, and its eigenvalue $\delta_r = \text{Mean}(\{\boldsymbol{\Lambda}_{k,k}|k \in \mathcal{S}_r\})$ by averaging the corresponding eigenvalues.

4.2 Graph Filter Bank Design for Multiple Graph Components

A single spectral filter bank ignores the spectral context and make the same frequency response to numerically identical eigenvalues for different graph components $\{\mathcal{G}_i\}_{i=1}^n$. To address the problem, we learn component-adaptive graph filter bank by generating the frequency response jointly over all subspaces while restricting spectral interactions within each graph component.

Let $\delta_k^{(i)}$ be the aggregated eigenvalue of k-th eigenspace of $\mathcal{G}_i$ and $\boldsymbol{\delta} = (\delta_1^{(1)}, \cdots, \delta_{M^{(1)}}^{(1)}, \cdots, \delta_1^{(n)}, \cdots, \delta_{M^{(n)}}^{(n)}) \in \mathbb{R}^M$ is the vector of dimension $M = \sum_{i=1}^n M^{(i)}$ containing all $\delta_k^{(i)}$ for $\mathcal{G}$. We rescale $\boldsymbol{\delta}$ with a learnable scale parameter s to $\Delta = [s^{-1}\boldsymbol{\delta}; s^0\boldsymbol{\delta}; s^1\boldsymbol{\delta}; \cdots; s^{J-1}\boldsymbol{\delta}] \in \mathbb{R}^{(J+1)\times M}$, and leverage Δ for PE encoding (Bo et al. (2023)) to obtain the high dimensional embedding $\Delta_{emb} \in \mathbb{R}^{(J+1)\times M\times \text{dim}_{hd}}$ of hidden dimension dim_{hd}. For $t = 1, \cdots, \text{dim}_{hd}/2$,

$$\Delta_{emb}^{(2t)} = \sin\left(\epsilon\Delta/10000^{2t/\text{dim}_{hd}}\right), \quad \Delta_{emb}^{(2t+1)} = \cos\left(\epsilon\Delta/10000^{2t/\text{dim}_{hd}}\right), \tag{5}$$

where ϵ is a constant. Δ_{emb} is then fed into Transformers for frequency response generation.

We introduce a mask Mask to reflect the component membership of any two subspaces with aggregated eigenvalue $\delta_k^{(i)}$ and $\delta_{k'}^{(i')}$ and global subspace index $a = o^{(i)} + k$ and $b = o^{(i')} + k'$. $\text{Mask}(a, b)$ is set to 1, only when $i = i'$. The encoded subspace representations at each scale are then processed by a shared Transformer with multi-head attention (MHA),

$$\Delta_{out} = \text{Transformer}(\Delta_{emb}, \text{ Mask}) \in \mathbb{R}^{(J+1)\times M\times \text{dim}_{hd}}, \tag{6}$$

where self-attention is performed using Δ_{emb} as the input to produce key, query and value. The attention mask allows graph component-wise filtering by restricting the interactions of eigenspaces within each graph component. Multi-scale representation for subspaces are used to form filter response of each subspace on each graph components with a decoder and construct the graph filter banks $\mathbf{F} \in \mathbb{R}^{(J+1)\times N}$. We obtain $\hat{\mathbf{g}} = \mathbf{F}[1, :]$ as the frequency response of the first filter and $\{\hat{\mathbf{h}}_j\}_{j=1}^J$ with $\hat{\mathbf{h}}_j = \mathbf{F}[j, :]$ normalized with $\sum_{j=1}^J \hat{\mathbf{h}}_j^2[k] \leq 1$ $(1 \leq k \leq N)$ for the remaining filters.

In Proposition 4.1, we demonstrate that, compared with a single spectral graph filter bank, the proposed component-adaptive graph filter bank is strictly superior for two or more graph components and is not worse for one graph component in the sense of expressiveness.

Proposition 4.1. *The proposed component-adaptive graph filter bank is at least as expressive as a single spectral graph filter bank, and is strictly more expressive when there exists at least two graph components with different sets of distinct eigenvalues.*

Proof. Please refer to Appendix A.3. □

4.3 Spectral Nonlinear Activation via Subspace Folding

Element-wise nonlinear activation is not well defined for spectral coefficients, due to its non-commutative property with respect to the subspace eigenbasis in the spectral domain. The spectral representation of graph signal $\boldsymbol{x}$ within the r-th subspace $\hat{\boldsymbol{x}}_r = \boldsymbol{U}_r^T\boldsymbol{x}$ can be represented with the orthonormal eigenbasis $\boldsymbol{U}_r = \text{Concat}(\{\boldsymbol{u}_k|k \in d_r\}) \in \mathbb{R}^{N\times|\mathcal{S}_r|}$, where $d_r = |\mathcal{S}_r|$ is the dimension of r-th subspace. Despite that $\boldsymbol{U}_r' = \boldsymbol{U}_r\boldsymbol{Q}_r$ is an orthonormal eigenbasis to satisfy $\hat{\boldsymbol{x}}_r' = \boldsymbol{U}_r'^T\boldsymbol{x} = \boldsymbol{Q}_r^T\hat{\boldsymbol{x}}_r$ for any orthonormal matrix $\boldsymbol{Q}_r \in \mathbb{R}^{d_r\times d_r}$, element-wise nonlinear activations cannot guarantee $\boldsymbol{Q}_r^T\sigma(\hat{\boldsymbol{x}}_r) = \sigma(\boldsymbol{Q}_r^T\hat{\boldsymbol{x}}_r)$. Thus, well-defined spectral nonlinear activation is required for learnable graph filter banks for multiple graph components.

Definition 4.1. *A nonlinear activation in GNN is said to be well-defined if its output in spatial domain is independent of choice of eigenbasis, and is equivariant to node permutation.*

***Example**: The conventional node-domain nonlinear activation $\sigma(\cdot)$ is well-defined. For a node permutation matrix $\boldsymbol{P}$, the element-wise activation gives $\sigma(\boldsymbol{P}\boldsymbol{x}) = \boldsymbol{P}\sigma(\boldsymbol{x})$. For orthonormal eigenbasis $\boldsymbol{U}$ and $\boldsymbol{U}'$, $\sigma(\boldsymbol{x}) = \sigma(\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{x}) = \sigma(\boldsymbol{U}'\boldsymbol{U}'^T\boldsymbol{x})$. Hence, $\sigma(\cdot)$ in node-domain is well-defined.*

The proposed spectral nonlinear activation requires a basis-consistent direction matrix $\boldsymbol{D}_r$ for each subspace. The direction matrix defines the canonical directions onto which the spectral coefficients are projected before applying the nonlinear activation. The direction matrix $\boldsymbol{D}_r$ is constructed separately in the cases of one-dimensional (1D) and multi-dimensional (MD) subspaces.

Direction matrix for 1D subspace: For $d_r = 1$, let $\boldsymbol{u}_m \in \mathbb{R}^N$ be the eigenvector. The direction matrix $\boldsymbol{D}_r$ is defined as $\boldsymbol{D}_r = [\text{sign}(\mathbf{1}^T\boldsymbol{u}_m)]$ when $|\mathbf{1}^T\boldsymbol{u}_m| \geq tol$ where $\mathbf{1}$ is all-one vector. When $|\mathbf{1}^T\boldsymbol{u}_m| < tol$, the direction cannot be numerically determined and $\boldsymbol{D}_r = \frac{1}{\sqrt{2}}[1, -1]$ to keep both directions. The coefficient $\frac{1}{\sqrt{2}}$ is adopted to guarantee $\boldsymbol{D}_r\boldsymbol{D}_r^T = \boldsymbol{I}$.

Direction matrix for MD subspace: For $d_r > 1$, the activation directions are obtained from the node-wise coordinates of the eigenspace. We decompose the candidate directions $\boldsymbol{v}_{r,k} = \boldsymbol{U}_r[k, :]^T \in \mathbb{R}^{d_r}$ into clusters according to their norms with minimum gap tol, considering that the norm $\|\boldsymbol{v}_{r,k}\| = \|\boldsymbol{Q}_r^T\boldsymbol{v}_{r,k}\|$ is invariant to basis rotation $\boldsymbol{Q}_r$, and select κ_r directions to form the matrix $\boldsymbol{Z}_r \in \mathbb{R}^{d_r \times \kappa_r}$ for the r-th subspace. Starting from the cluster with the largest norm, we iteratively select a candidate direction with the norm of residual outside the subspace spanned by all previously selected directions not larger than tol and update the subspace with the selected candidate direction. The procedure is repeated until the selected directions reach a predefined $\kappa_{\max}$ or span the whole space. Cluster partition and direction selection are elaborated in Algorithms 1 and 2 in Appendix A. The direction matrix $\boldsymbol{D}_r = (\tilde{\boldsymbol{Z}}_r\tilde{\boldsymbol{Z}}_r^T)^{\dagger 1/2}\tilde{\boldsymbol{Z}}_r$, where $\tilde{\boldsymbol{Z}}_r = [\boldsymbol{z}_1/\|[\boldsymbol{z}_1\|, \cdots, \boldsymbol{z}_{\kappa_r}/\|\boldsymbol{z}_{\boldsymbol{\kappa_r}}\|]$ is the normalized directions of $\boldsymbol{Z}_r$ and $(\cdot)^\dagger$ denotes the Moore-Penrose pseudoinverse. Note that $\boldsymbol{D}_r\boldsymbol{D}_r^T = \boldsymbol{I}_{d_r \times d_r}$, when $\text{rank}(\tilde{\boldsymbol{Z}}_r) = d_r$.

Spectral activation via subspace folding: Given the spectrum of graph signal $\boldsymbol{X} \in \mathbb{R}^{N \times C}$ of C channels, the spectral coefficient is partitioned into the M eigenspaces as $\hat{\boldsymbol{X}} = \{\hat{\boldsymbol{X}}_r\}_{r=1}^M$, where $\hat{\boldsymbol{X}}_r = \hat{\boldsymbol{X}}[\mathcal{S}_r, :] \in \mathbb{R}^{d_r \times C}$. Spectral nonlinear activation via subspaces $\sigma_{spec}(\cdot)$ is developed by folding with normalized direction matrix $\boldsymbol{D}_r$.

$$\sigma_{spec}(\hat{\boldsymbol{X}}) = \text{MERGE}_{r=1}^M \left[\sigma_{spec}^{(r)}(\hat{\boldsymbol{X}}_r)\right], \ \sigma_{spec}^{(r)}(\hat{\boldsymbol{X}}_r) = \hat{\boldsymbol{X}}_r + \boldsymbol{D}_r\left(\sigma(\boldsymbol{D}_r^T\hat{\boldsymbol{X}}_r) - \boldsymbol{D}_r^T\hat{\boldsymbol{X}}_r\right) \quad (7)$$

where $\sigma(\cdot)$ is arbitrary nonlinear activation function. When $\text{rank}(\boldsymbol{D}_r) = d_r$, we obtain the simplified formulation $\sigma_{spec}^{(r)}(\hat{\boldsymbol{X}}_r) = \boldsymbol{D}_r\sigma(\boldsymbol{D}_r^T\hat{\boldsymbol{X}}_r)$. In Appendix A.6, Figure 4 provides an illustrative exampled on the spectral nonlinear activation of a 2-dimensional subspace.

Proposition 4.2. *The proposed spectral nonlinear activation $\sigma_{spec}(\cdot)$ satisfies Definition 4.1.*

Proof. Please refer to Appendix A.2. □

It is worth mentioning that node-domain nonlinear activation with $\sigma(\hat{\boldsymbol{X}}) = \boldsymbol{U}^T\sigma(\boldsymbol{U}\hat{\boldsymbol{X}})$ in the spectral domain can be interpreted as a special case of the proposed spectral activation $\sigma_{spec}(\hat{\boldsymbol{X}})$ by considering the whole space as only one subspace and taking $\boldsymbol{D} = \boldsymbol{U}^T$ in (7).

The overall complexity of the proposed spectral activation is $\mathcal{O}(\sum_{r=1}^M d_r\kappa_r C)$. In practice, κ_r is bounded by $\mathcal{O}(d_r)$ and the overall complexity becomes $\mathcal{O}(\sum_{r=1}^M d_r^2 C)$. In contrast, performing nonlinear activation in the node domain requires an IGFT with time complexity $\mathcal{O}(N^2C)$. Since $\sum_{r=1}^M d_r = N$ and $\sum_{r=1}^M d_r^2 \ll N^2$ in the majority of graph structures, the proposed spectral nonlinear activation dramatically reduces the computational cost, especially on large-scale graphs.

5 EXPERIMENTS

We evaluate the proposed SpecGSN and its variant SpecGSN† that substitutes the proposed spectral non-linear activation with node-wise modulus for node classification. Methods for comparison include spatial GNNs (*i.e.*, GAT Veli*v*cković et al. (2018), GraphSAGE Hamilton et al. (2017), MixHop Abu-El-Haija et al. (2019), LightGCN He et al. (2020), DAGNN Liu et al. (2020), HLGNN Sun et al. (2023), DeloopSGNN Li et al. (2026)), spectral GNNs (*i.e.*, GCN Kipf & Welling (2017), ChebNet Defferrard et al. (2016), SGWF Shen et al. (2021), ARMA Bianchi et al. (2022), BernNet He et al. (2021), Specformer Bo et al. (2023), GeomGNN Wang et al. (2024)), and GSNs (*i.e.*,

Table 1: Node classification performance of homophilic graph datasets (The best result is highlighted in bold, and the second-best result is underlined.)

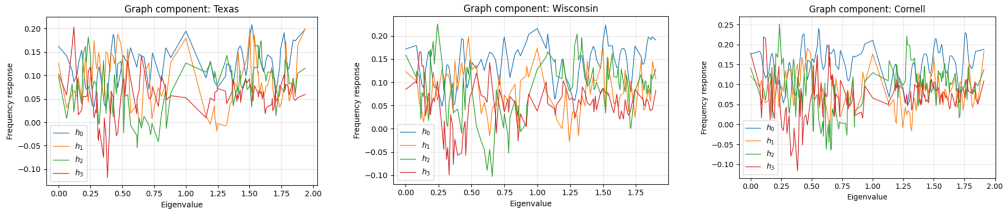
Methods		Cora	Citeseer	Pubmed	DBLP	Computers	Photo
Spatial	GAT	88.20±0.27	76.21±0.51	87.18±1.15	85.74±0.07	91.19±0.06	94.13±0.21
	GrapSAGE	88.19±0.30	76.30±0.35	88.59±0.08	84.91±0.07	90.68±0.06	95.26±0.10
	MixHop	87.30±0.38	75.35±0.36	89.04±0.14	84.71±0.13	87.76±0.70	90.80±0.12
	LightGCN	87.26±0.37	77.82±0.10	88.02±0.03	80.58±0.08	89.74±0.10	95.07±0.14
	HLGNN	88.99±0.24	**77.95±0.49**	89.30±0.10	85.30±0.13	90.57±0.06	95.02±0.12
	DAGNN	**89.31±0.19**	77.04±0.33	87.06±0.04	85.81±0.09	91.41±0.10	95.33±0.06
	DeloopSGNN	89.06±0.33	76.78±0.46	89.08±0.04	85.76±0.05	91.94±0.06	95.55±0.05
Spectral	GCN	88.56±0.23	76.81±0.52	86.80±0.07	85.72±0.13	89.95±0.08	93.65±0.08
	ChebNet	87.46±0.36	74.82±0.40	89.11±0.09	85.10±0.10	91.33±0.03	95.10±0.19
	SGWF	86.98±0.30	72.88±1.56	89.74±0.13	81.54±0.89	91.03±0.16	95.18±0.11
	ARMA	88.02±0.54	76.78±0.19	89.17±0.11	84.73±0.11	90.49±0.15	94.97±0.08
	BernNet	88.96±0.25	77.11±0.24	89.42±0.05	86.23±0.11	90.91±0.11	95.15±0.15
	Specformer	88.35±0.46	75.55±0.60	89.79±0.03	85.56±0.16	92.18±0.07	95.39±0.10
	GeomGNN	87.92±0.20	76.48±0.54	89.83±0.09	86.08±0.12	92.18±0.08	95.74±0.09
Scattering	GSN	88.06±0.21	75.42±0.36	84.65±0.06	80.71±0.10	87.50±0.19	91.28±0.15
	ScatterGCN	87.86±0.37	76.03±0.34	88.00±0.09	**86.31±0.09**	89.08±0.43	93.52±0.08
	GSAN	87.97±0.15	74.84±0.35	86.01±0.16	85.10±0.38	88.72±0.53	93.72±0.27
	GGSN	85.34±0.47	76.14±0.32	89.56±0.06	83.95±0.67	91.96±0.09	95.43±0.15
	LEGS	88.34±0.21	75.00±0.32	85.37±0.10	86.06±0.09	90.10±0.08	94.99±0.12
	ad-GSN	88.18±0.31	77.73±0.04	87.41±0.13	84.78±0.10	90.98±0.16	94.00±0.40
Proposed	SpecGSN	89.27±0.24	77.59±0.47	**90.16±0.10**	86.06±0.09	92.61±0.10	95.63±0.14
	SpecGSN+relu	89.05±0.28	77.44±0.29	90.11±0.17	86.02±0.07	92.65±0.07	95.62±0.08
	SpecGSN†	89.20±0.25	77.51±0.37	90.04±0.09	85.97±0.10	**92.78±0.08**	**95.81±0.06**

Table 2: Node classification performance of heterophilic graph datasets (The best result is highlighted in bold, and the second-best result is underlined.)

Methods		Texas	Wisconsin	Cornell	Merged	Roman-empire	Chameleon	Actor
Spatial	GAT	58.25±1.13	48.55±1.41	42.38±2.99	49.66±1.14	60.41±0.27	71.80±0.43	27.19±0.50
	GrapSAGE	82.71±1.05	77.41±1.72	70.33±0.71	72.48±0.51	76.58±0.26	52.10±0.86	35.83±0.49
	MixHop	66.01±1.28	59.94±0.60	47.84±1.66	58.00±0.38	79.11±0.26	64.67±1.33	32.20±0.44
	LightGCN	64.50±2.61	69.46±1.22	59.31±2.89	65.15±1.19	64.64±0.09	54.98±0.62	35.65±0.22
	HLGNN	79.37±1.38	81.10±1.67	70.41±1.06	70.78±0.19	74.63±0.20	52.16±0.48	35.79±0.33
	DAGNN	55.10±0.51	51.25±1.72	47.33±2.37	52.14±1.10	67.05±0.10	57.61±0.55	32.92±0.27
	DeloopSGNN	79.91±1.14	77.28±1.79	68.56±1.80	71.62±0.51	72.59±0.26	64.31±0.44	36.32±0.44
Spectral	GCN	53.17±1.00	47.27±1.43	42.29±0.81	47.61±1.70	34.22±0.12	63.78±0.65	28.82±0.27
	ChebNet	71.57±2.09	73.52±1.69	59.13±1.74	67.10±1.09	77.25±0.13	73.69±0.42	34.47±0.48
	SGWF	79.08±2.82	79.11±1.17	70.73±1.06	69.36±1.51	76.40±0.10	69.15±0.39	**37.43±0.24**
	ARMA	83.58±0.61	**84.95±0.72**	72.51±1.41	72.86±1.02	78.81±0.18	52.56±0.79	35.84±0.35
	BernNet	82.63±0.78	84.33±0.86	76.51±0.79	74.64±0.77	75.83±0.10	70.15±0.46	37.23±0.46
	Specformer	82.94±1.17	80.94±1.90	74.70±2.11	74.46±0.85	73.00±0.30	76.31±0.36	35.92±0.16
	GeomGNN	67.82±1.81	73.39±1.66	57.70±2.18	67.13±1.27	78.08±0.27	68.48±1.04	36.42±0.45
Scattering	GSN	53.97±1.40	47.50±2.10	42.73±1.21	48.57±0.70	30.43±0.11	56.88±0.79	26.68±0.25
	ScatterGCN	56.92±2.51	49.68±2.44	43.61±1.94	45.95±0.82	78.39±0.20	60.62±1.01	34.31±0.34
	GSAN	53.49±1.03	47.51±0.98	42.14±2.02	45.49±0.89	67.97±0.30	53.41±0.70	30.31±0.55
	GGSN	74.16±1.05	73.57±0.61	63.49±0.63	69.36±0.45	81.19±0.16	65.98±0.57	36.41±0.46
	LEGS	57.72±2.38	64.68±1.37	51.14±2.80	58.21±0.78	73.75±0.19	61.04±0.89	34.55±0.24
	ad-GSN	76.80±1.26	78.53±1.52	66.73±0.86	70.22±0.86	74.21±0.10	57.89±0.62	34.64±0.40
Proposed	SpecGSN	**85.92±2.00**	84.62±1.06	**78.21±1.25**	**75.78±0.63**	80.38±0.17	**76.89±0.46**	36.53±0.20
	SpecGSN+relu	85.43±1.85	83.48±1.33	77.67±1.20	75.28±0.39	77.55±0.23	76.33±0.42	36.10±0.54
	SpecGSN†	85.90±1.90	84.32±1.13	78.01±1.19	75.32±0.47	**82.31±0.28**	76.08±0.40	36.30±0.44

GSN Gama et al. (2019a), ScatterGCN Min et al. (2020), GSAN Min et al. (2021), GGSN Koke & Kutyniok (2022), LEGS Tong et al. (2024), ad-GSN Liu et al. (2026)).

Experiment Setup. We evaluate each method using 10-fold cross-validation under 5 random seeds for 50 runs in total. Data partitions remain the same to ensure a fair comparison. For model selection within each fold, we first identify the checkpoints with the highest validation accuracy, and then select at most five of them with the lowest validation losses to further reduce randomness. The test accuracies of the selected checkpoints are averaged as the result of one fold. The performance of each 10-fold cross-validation is obtained by averaging the test accuracies over the ten folds. Finally, we report the mean and standard deviation over the five independent 10-fold cross-validations. The hidden dimension for SpecGSN is set as 64 with scale parameter s initialized as 2. Scattering-based



(a) Filter banks for Texas (b) Filter banks for Wisconsin (c) Filter banks for Cornell

Figure 2: The component-adaptive filter banks generated for three graph components in Merged dataset. The same eigenvalues (eg. $\lambda = 0, 1$) are assigned with different frequency response for each graph component.

methods including GSN, GGSN, LEGS, ad-GSN, and SpecGSN are implemented with 3 layers and a filter bank containing 4 filters, while spatial and spectral methods are implemented with 2 layers.

Nonlinear Function. The nonlinear function used in spectral nonlinear activation is

$$\sigma(x) = \left(\theta_0 + \xi_{sin} \sum_{i=0}^{\gamma-1} \theta_{1,i} \sin(\omega^i |\varepsilon x|) + \xi_{cos} \sum_{i=0}^{\gamma-1} \theta_{2,i} \cos(\omega^i |\varepsilon x|)\right) x \tag{8}$$

where γ is the number of basis, ω and $\theta_\star$ are learnable parameters, $\xi_{sin}, \xi_{cos} \in \{0, 1\}$ are hyperparameters to determine the usage of cosine and sine basis. Parameter ε is fixed as 1. All the detail hyper-parameter settings for SpecGSN are released in Appendix A.5.

5.1 Node Classification

We adopt 6 homophilic datasets, including Cora, Citeseer, Pubmed, DBLP, Amazon Computers and Amazon Photo, and 6 heterophilic datasets, including Texas, Wisconsin, Cornell, Roman-empire, Chameleon and Actor. Additionally, we construct a dataset called "Merged" by combining Texas, Wisconsin and Cornell. These three datasets share the same feature dimensionality, contain 5 node classes, and all belong to WiKiCS, making them suitable for joint learning. However, the graph structure and spectral context of its three graph components (Texas, Wisconsin and Cornell) are different, and the optimal frequency responses for node classification are not identical. Therefore, the Merged dataset serves a tougher challenge for GNNs that rely on a unified filter or filter bank. The detail information of used datasets are provided in Appendix A.4.

Node classification performances of homophilic and heterophilic graph datasets are presented in table 1 and table 2 respectively. The proposed SpecGSN achieves competitive performance on most homophilic and heterophilic graph datasets with half of them reach state-of-the-art. Clearly, the advantage of SpecGSN in representation learning is more pronounced on heterophilic graphs. We attribute the effectiveness of SpecGSN not only to the proposed component-adaptive filter bank learning and the spectral non-linear activation via subspace folding, but also to the carefully designed learnable odd activation function in equation 8. In fact, the design of activation function is of crucial importance in SpecGSN. While adopting fixed activation functions (as shown in SpecGSN+relu) can lead to a performance degradation of at most 2 percentage points in heterophilic graph datasets, a properly parameterized learnable activation function enables the proposed spectral nonlinear activation to achieve performance comparable to conventional node-domain nonlinear activation. Meanwhile, we notice that SpecGSN achieves the best performance on the Merged dataset. The component-adaptive filter bank allows the learn of distinct frequency responses for Texas, Wisconsin, and Cornell, leading to better node-classification performance. The filter banks generated for the Merged dataset are shown in Figure 2.

5.2 Ablation Studies

In this section, we evaluate the effectiveness of component-adaptive filter bank by comparing it with a unified filter bank. The unified filter bank adopts the same generation process with mask in 6 removed. Cora, Citeseer, DBLP and Merged included in comparison. Computers and Photo are

Table 3: Comparison of node classification performance for the component-adaptive filter bank and the unified filter bank

	Cora	Citeseer	DBLP	Merged	Cora-m	Citeseer-m	DBLP-m
SpecGSN	89.27±0.24	77.59±0.47	86.06±0.09	75.78±0.63	90.04±0.85	73.32±0.55	86.90±0.53
SpecGSN(no mask)	89.08±0.29	77.51±0.12	86.05±0.08	74.97±0.48	89.90±1.75	73.18±0.41	86.87±0.34

Table 4: Computational time in seconds of graph scattering transform. (#MD, #dim and #dir are the total number of MD subspaces, their total dimensions and the total number of directions selected across all MD subspaces. Ex-time is the extra time in constructing direction matrix for subspaces, $\mathcal{B} = \gamma(\xi_{sin} + \xi_{cos})$) is the number of sine and cosine basis used in nonlinear function.)

Dataset	#MD	#dim	#dir	CPU (13th Gen Intel Core i5-13400)					GPU (NVIDIA GeForce RTX 3090)				
				Ex-time	Relu	$\mathcal{B}$=4	$\mathcal{B}$=16	SpecGSN†	Ex-time	Relu	$\mathcal{B}$=4	$\mathcal{B}$=16	SpecGSN†
Cora	12	364	451	0.328	0.030	0.105	0.150	0.144	1.194	0.014	0.056	0.057	0.006
Citeseer	49	508	738	0.568	0.040	0.135	0.192	0.211	1.266	0.016	0.061	0.060	0.007
Pubmed	22	12167	12216	1270.03	2.539	2.890	3.636	11.446	138.35	0.050	0.085	0.085	0.177
DBLP	149	2384	2872	8.418	0.234	0.518	1.056	9.142	5.613	0.012	0.054	0.053	0.153
Computers	633	1345	1428	4.185	0.116	0.320	0.770	5.227	4.540	0.011	0.052	0.053	0.090
Photo	122	272	296	1.346	0.054	0.166	0.393	1.320	2.211	0.011	0.053	0.053	0.027
Roman-empire	2327	5932	6040	15.726	0.651	0.955	1.676	16.519	12.425	0.016	0.052	0.056	0.261
Chameleon	17	1185	1207	0.843	0.052	0.113	0.136	0.097	1.234	0.016	0.060	0.061	0.005
Actor	20	1225	1277	2.017	0.083	0.189	0.390	1.313	2.614	0.011	0.045	0.053	0.027

excluded due to a large number of isolated single-node graph components, while others are excluded because they consist of only one graph component. Meanwhile, we construct three modified datasets named Cora-m, Citeseer-m, DBLP-m by removing the largest graph component and all isolated nodes from the original datasets. The remaining graph components are more balanced in size, which prevents the classification accuracy dominated by a single graph component. Table 3 shows the node classification performance. In summary, the gain of component-adaptive filter bank is more obvious on datasets with graph components of comparable size.

5.3 Time Consumption Analysis

We now compare the time consumption of the SpecGSN and SpecGSN† to investigate the computational efficiency of the proposed spectral nonlinear activation compared with node-domain nonlinear activation in graph scattering networks. Table 4 reports the average computational time over the first 20 epochs of the graph scattering transform on both CPU and GPU. With only a one-time cost for computing the direction matrices for subspaces (shown in Ex-time), graph scattering transform with the proposed spectral nonlinear activation exhibits far less computational time compared with graph scattering transform with conventional nonlinearity (shown in SpecGSN†). Scattering with spectral nonlinear activation is faster on CPU/GPU for graphs with more than 3000/10,000 nodes respectively. In fact, the GPU time for spectral nonlinear activation is dominated by tensor operations such as split and concatenate in most cases so that the time remains stable on datasets of different size. Notably, the computation of direction matrix for Pubmed is expensive due to a 12121 dimensional subspace with $\lambda = 1$. Nevertheless, the spectral nonlinear activation leads to over 50% time reduction in the subsequent computation of graph scattering transform on both CPU and GPU.

6 Conclusion

In this paper, we propose SpecGSN that leverages Transformer for component-adaptive filter bank learning, and introduce a novel spectral nonlinear activation paradigm with basis-rotation invariance and permutation equivalence guaranteed. We demonstrate the effectiveness of SpecGSN in node classification with homophilic and heterophilic graph datasets, and prove the expressive capability of the proposed component-adaptive filter bank. Meanwhile, we also demonstrate that the proposed spectral nonlinear activation achieves comparable performance with conventional node-domain nonlinear activation while dramatically decreases the computational time of graph scattering transform for relatively large datasets. In fact, the spectral nonlinear activation paradigm is suitable for any spectral-based GNNs for computational time reduction.

7 AI USE STATEMENT

In this work, we used AI tools to assist with and polish the writing, including correcting language errors and improving the clarity and conciseness of the text. We also use AI tools to assist the code implementation, including the update of maintained subspace in Algorithm 2, and checking the correctness of eigenspace index arrangement in SpecGSN. All AI-assisted contents were carefully reviewed, checked, and validated by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

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 Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pp. 21–29. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/abu-el-haija19a.html.

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.

Deyu Bo, Chuan Shi, Lele Wang, and Renjie Liao. Specformer: Spectral graph neural networks meet transformers. In The Eleventh International Conference on Learning Representations, may 2023.

Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8):1872–1886, aug 2013. doi: 10.1109/TPAMI.2012.230.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché Buc, Edward A. Fox, and Roman Garnett (eds.), Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada, pp. 4869–4880, Vancouver, BC, Canada, dec 2019. Curran Associates, Inc.

Joyce A. Chew, Matthew J. Hirn, Smita Krishnaswamy, Deanna Needell, Michael Perlmutter, Holly R. Steach, Siddharth Viswanath, and Hau-Tieng Wu. Geometric scattering on measure spaces. arXiv preprint arXiv:2208.08561, aug 2022.

Ronald R. Coifman and Mauro Maggioni. Diffusion wavelets. Applied and Computational Harmonic Analysis, 21(1):53–94, jul 2006. doi: 10.1016/j.acha.2006.04.004.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In Neural Information Processing Systems, pp. 3844–3852, 2016. URL https://mlanthology.org/neurips/2016/defferrard2016neurips-convolutional/.

Fernando Gama, Alejandro Ribeiro, and Joan Bruna. Diffusion scattering transforms on graphs. In International Conference on Learning Representations (ICLR), may 2019a.

Fernando Gama, Alejandro Ribeiro, and Joan Bruna. Stability of graph scattering transforms. In Neural Information Processing Systems, pp. 8038–8048, 2019b. URL https://mlanthology.org/neurips/2019/gama2019neurips-stability/.

Feng Gao, Guy Wolf, and Matthew J. Hirn. Geometric scattering for graph data analysis. In Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pp. 2122–2131, Long Beach, California, USA, jun 2019. PMLR. doi: 10.5555/3454287.3454476.

Shanel Gauthier, Benjamin Thérien, Laurent Alsène-Racicot, Muawiz Chaudhary, Irina Rish, Eugene Belilovsky, Michael Eickenberg, and Guy Wolf. Parametric scattering networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5749–5758, June 2022.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), Advances in Neural Information Processing Systems 30 (NIPS 2017), volume 30, pp. 1024–1034, Long Beach, CA, USA, dec 2017. Curran Associates, Inc.

David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. Applied and Computational Harmonic Analysis, 30(2):129–150, 2011. doi: 10.1016/j.acha.2010.04.005.

Mingguo He, Zhewei Wei, and Huan Xu. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. In Advances in Neural Information Processing Systems, volume 34, pp. 14239–14251, dec 2021.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, pp. 639–648, Virtual Event, China, jul 2020. Association for Computing Machinery. ISBN 9781450380164. doi: 10.1145/3397271.3401063.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In International Conference on Learning Representations (ICLR), 2017.

Christian Koke and Gitta Kutyniok. Graph Scattering Beyond Wavelet Shackles. In Neural Information Processing Systems, 2022. URL https://mlanthology.org/neurips/2022/koke2022neurips-graph/.

Runlin Lei, Zhen Wang, Yaliang Li, Bolin Ding, and Zhewei Wei. Evennet: Ignoring odd-hop neighbors improves robustness of graph neural networks. In Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, nov 2022.

Ron Levie, Federico Monti, Xavier Bresson, and Michael M. Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. IEEE Transactions on Signal Processing, 67(1):97–109, jan 2019. doi: 10.1109/TSP.2018.2879624.

Duanyu Li, Huijun Wu, Min Xie, Kai Lu, Wenzhe Zhang, Zhenwei Wu, Yong Dong, and Ruibo Wang. Deloopsgnn: Revisiting spectral gnns through the lens of spatial aggregation. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 40, pp. 22878–22886, Vancouver, Canada, feb 2026. AAAI Press. doi: 10.1609/aaai.v40i27.39451.

Meng Liu, Hongyang Gao, and Shuiwang Ji. Towards deeper graph neural networks. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 338–348. ACM, 2020. doi: 10.1145/3394486.3403076.

Shiyu Liu, Wenrui Dai, Shaohui Li, Ziyang Zheng, Junni Zou, and Hongkai Xiong. Stable representation learning via generalized learnable graph scattering transform. IEEE Transactions on Signal Processing, 74:701–716, 2026. doi: 10.1109/TSP.2026.3660885.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacvić, Thomas Y. Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. In The Thirteenth International Conference on Learning Representations, 2025. URL https://openreview.net/forum?id=0zo7qJ5vZi.

Stéphane Mallat. Group invariant scattering. Communications on Pure and Applied Mathematics, 65(10):1331–1398, 2012. doi: 10.1002/cpa.21413.

Yimeng Min, Frederik Wenkel, and Guy Wolf. Scattering gcn: Overcoming oversmoothness in graph convolutional networks. arXiv preprint arXiv:2003.08414, mar 2020.

Yimeng Min, Frederik Wenkel, and Guy Wolf. Geometric scattering attention networks. In ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8518–8522, Toronto, Ontario, Canada, jun 2021. IEEE. doi: 10.1109/ICASSP39728.2021.9414937.

Chao Pan, Siheng Chen, and Antonio Ortega. Spatio-temporal graph scattering transform. In International Conference on Learning Representations (ICLR), may 2021. URL https://www.merl.com/publications/TR2021-044.

Yangmei Shen, Wenrui Dai, Chenglin Li, Junni Zou, and Hongkai Xiong. Multi-scale graph convolutional network with spectral graph wavelet frame. IEEE Transactions on Signal and Information Processing over Networks, 7:595–610, 2021. doi: 10.1109/TSIPN.2021.3109820.

Yiyang Sun, Yuncong Zhao, Longjie Li, and Hu Dong. Heterogeneous line graph neural network for link prediction. In Advanced Data Mining and Applications - 19th International Conference, ADMA 2023, Shenyang, China, August 21-23, 2023, Proceedings, Part V, volume 14180 of Lecture Notes in Computer Science, pp. 3–14. Springer, 2023. doi: 10.1007/978-3-031-46677-9_1.

Alexander Tong, Frederick Wenkel, Kincaid MacDonald, Smita Krishnaswamy, and Guy Wolf. Data-driven learning of geometric scattering modules for gnns. In 2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1–6, Gold Coast, Australia, oct 2021. IEEE. doi: 10.1109/MLSP52302.2021.9596169.

Alexander Tong, Frederik Wenkel, Dhananjay Bhaskar, Kincaid Macdonald, Jackson Grady, Michael Perlmutter, Smita Krishnaswamy, and Guy Wolf. Learnable filters for geometric scattering modules. IEEE Transactions on Signal Processing, 72:2939–2952, 2024. doi: 10.1109/TSP.2024.3378001.

Petar Veli$v$cković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In International Conference on Learning Representations, 2018. URL https://mlanthology.org/iclr/2018/velickovic2018iclr-graph/.

Xiyuan Wang and Muhan Zhang. How powerful are spectral graph neural networks. In International Conference on Machine Learning, volume 162, pp. 23341–23362. PMLR, 2022. URL https://proceedings.mlr.press/v162/wang22am.html.

Zhiyang Wang, Luana Ruiz, and Alejandro Ribeiro. Geometric graph filters and neural networks: Limit properties and discriminability trade-offs. IEEE Transactions on Signal Processing, 72: 2244–2259, April 2024. doi: 10.1109/TSP.2024.3392360.

Thomas Wiatowski and Helmut Bölcskei. Deep convolutional neural networks based on semi-discrete frames. In 2015 IEEE International Symposium on Information Theory (ISIT), pp. 1212–1216, Hong Kong, China, jun 2015. IEEE. doi: 10.1109/ISIT.2015.7282648.

Jie Zhang and Min-Te Sun. From chebnet to chebgibbsnet. 2024.

Dongmian Zou and Gilad Lerman. Graph convolutional neural networks via scattering. Applied and Computational Harmonic Analysis, 49(3):1046–1074, November 2020. doi: 10.1016/j.acha.2019.06.003.

A Appendix

A.1 Algorithms for Norm-Cluster Partition and Direction Selection

Norm-cluster partition is given in algorithm 1 while direction selection is shown in algorithm 2.

Algorithm 1 Norm-Cluster Partition of Candidate Directions

Require: Candidate directions $\{\mathbf{v}_n\}_{n=1}^{N_i}$, tolerance tol
Ensure: Norm clusters $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$
Compute the norm of each candidate direction: $\rho_n \leftarrow \|\mathbf{v}_n\|_2$
Remove candidates satisfying $\rho_n < tol$
Sort candidates in descending order of ρ_n
$K \leftarrow 1$
for each candidate $\mathbf{v}_n$ in the sorted order **do**
    if $K = 1$ **then**
        $\mathcal{C}_K \leftarrow \{n\}$, $\bar{\rho}_K \leftarrow \rho_n$
    else if $|\rho_n - \bar{\rho}_K| \leq \epsilon_\rho$ **then**
        $\mathcal{C}_K \leftarrow \mathcal{C}_K \cup \{n\}$
    else
        $K \leftarrow K + 1$
        $\mathcal{C}_K \leftarrow \{n\}$, $\bar{\rho}_K \leftarrow \rho_n$
    end if
end for
return $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$

Algorithm 2 Direction Selection for an MD Eigenspace

Require: Eigenspace basis $\mathbf{U}_r \in \mathbb{R}^{N \times d_r}$, norm clusters $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$, rank tolerance tol, optional maximum number of directions $\kappa_{\max}$
Ensure: Selected direction matrix $\mathbf{Z}_r$
if $\kappa_{\max}$ is not specified **then**
    $\kappa_{\max} \leftarrow N$
end if
$\mathbf{v}_n \leftarrow \mathbf{U}_r[n, :]^\top,\ n = \{1, \ldots, N\}$ ▷ Construct candidate directions
$\mathbf{Z}_r \leftarrow [\,], \mathbf{B} \leftarrow [\,]$ ▷ $\mathbf{B}$ is an orthonormal basis of the maintained subspace
for $k = 1, \ldots, K$ **do**
    $\mathbf{V}_k = [\mathbf{v}_n]_{n \in \mathcal{C}_k}$, $\mathbf{mask}_k \leftarrow \mathbf{0}$
    if $\mathbf{B}$ is empty **then**
        $\mathbf{R}_k \leftarrow \mathbf{V}_k$
    else
        $\mathbf{R}_k \leftarrow (\mathbf{I} - \mathbf{B}\mathbf{B}^\top)\mathbf{V}_k$ ▷ Compute residuals with respect to the maintained subspace
    end if
    $\mathbf{mask}_k[j] \leftarrow \mathbb{I}\left(\|\mathbf{R}_k[:, j]\|_2 \geq tol\right)$ ▷ Generate the selection mask
    $\widetilde{\mathbf{V}}_k \leftarrow \mathbf{V}_k[:, \mathbf{mask}_k]$ ▷ Select directions in cluster
    if $\widetilde{\mathbf{V}}_k$ is not empty **then**
        $\mathbf{Z}_r \leftarrow [\mathbf{Z}_r, \widetilde{\mathbf{V}}_k]$
        Update $\mathbf{B}$ as an orthonormal basis of $\mathrm{span}(\mathbf{Z}_r)$
    end if
    if $\mathrm{rank}(\mathbf{B}) = d_r$ or $\mathrm{ncol}(\mathbf{Z}_r) \geq \kappa_{\max}$ **then**
        break
    end if
end for
return $\mathbf{Z}_r$

A.2 Proof of Proposition 4.2

For the r-th subspace, directions in $\boldsymbol{Z}_r$ are selected according to clusters separated by norm which is invariant to node permutation. Denote $\boldsymbol{Z}_r'$ as the direction matrix of $\mathcal{G}$ under node permutation,

there exist a permutation matrix $\boldsymbol{P}_r$ such that $\boldsymbol{Z}'_r = \boldsymbol{Z}_r\boldsymbol{P}_r$. Therefore, $\boldsymbol{D}'_r = \boldsymbol{D}_r\boldsymbol{P}_r$ and

$$\begin{aligned}\sigma^{(r)}_{spec}(\hat{\boldsymbol{X}}_r) &= \hat{\boldsymbol{X}}_r + \boldsymbol{D}_r\left(\sigma(\boldsymbol{D}_r^T\hat{\boldsymbol{X}}_r) - \boldsymbol{D}_r^T\hat{\boldsymbol{X}}_r\right)\\ &= \hat{\boldsymbol{X}}_r + \boldsymbol{D}_r\boldsymbol{P}_r\left(\boldsymbol{P}_r^T\sigma(\boldsymbol{D}_r^T\hat{\boldsymbol{X}}_r) - (\boldsymbol{D}_r\boldsymbol{P}_r)^T\hat{\boldsymbol{X}}_r\right)\\ &= \hat{\boldsymbol{X}}_r + \boldsymbol{D}'_r\left(\sigma(\boldsymbol{D}_r'^T\hat{\boldsymbol{X}}_r) - \boldsymbol{D}_r'^T\hat{\boldsymbol{X}}_r\right)\end{aligned}$$

since $\boldsymbol{P}^T\sigma(\boldsymbol{x}) = \sigma(\boldsymbol{P}^T\boldsymbol{x})$ for any permutation matrix.

Meanwhile, given arbitrary orthonormal matrix $\boldsymbol{Q}_r \in \mathbb{R}^{d_r\times d_r}$ with rotated basis $\boldsymbol{U}'_r = \boldsymbol{U}_r\boldsymbol{Q}_r$, the direction matrix under rotated basis is $\boldsymbol{Z}'_r = \boldsymbol{Q}_r^T\boldsymbol{Z}_r \in \mathbb{R}^{d_r\times\kappa_r}$. $\|\boldsymbol{Q}_r^T\boldsymbol{z}_h\| = \|\boldsymbol{z}_h\|$ $(h = 1, 2, \cdots, \kappa_r)$ gives $\tilde{\boldsymbol{Z}}'_r = \boldsymbol{Q}_r^T\tilde{\boldsymbol{Z}}_r$. Since the Moore–Penrose pseudoinverse is equivariant under orthogonal similarity transformations,

$$(\tilde{\boldsymbol{Z}}'_r\tilde{\boldsymbol{Z}}_r'^T)^\dagger = (\boldsymbol{Q}_r^T\tilde{\boldsymbol{Z}}_r\tilde{\boldsymbol{Z}}_r^T\boldsymbol{Q}_r)^\dagger = \boldsymbol{Q}_r^T\tilde{(}\boldsymbol{Z}_r\tilde{\boldsymbol{Z}}_r^T)^\dagger\boldsymbol{Q}_r, \tag{9}$$

the normalized direction matrix $\boldsymbol{D}'_r = \boldsymbol{Q}_r^T\boldsymbol{D}_r$. Denote $\hat{\boldsymbol{X}}'_r = \boldsymbol{U}_r'^T\boldsymbol{X} = \boldsymbol{Q}_r^T\hat{\boldsymbol{X}}_r$,

$$\begin{aligned}\boldsymbol{U}'_r\sigma^{(r)}_{spec}(\hat{\boldsymbol{X}}'_r) &= \boldsymbol{U}'_r\left(\hat{\boldsymbol{X}}'_r + \boldsymbol{D}'_r\left(\sigma(\boldsymbol{D}_r'^T\hat{\boldsymbol{X}}'_r) - \boldsymbol{D}_r'^T\hat{\boldsymbol{X}}'_r\right)\right)\\ &= \boldsymbol{U}_r\boldsymbol{Q}_r\left(\boldsymbol{Q}_r^T\hat{\boldsymbol{X}}_r + \boldsymbol{Q}_r^T\boldsymbol{D}_r\left(\sigma(\boldsymbol{D}_r^T\boldsymbol{Q}_r\boldsymbol{Q}_r^T\hat{\boldsymbol{X}}_r) - \boldsymbol{D}_r^T\boldsymbol{Q}_r\boldsymbol{Q}_r^T\hat{\boldsymbol{X}}_r\right)\right)\\ &= \boldsymbol{U}_r\left(\hat{\boldsymbol{X}}_r + \boldsymbol{D}_r\left(\sigma(\boldsymbol{D}_r^T\hat{\boldsymbol{X}}_r) + \boldsymbol{D}_r^T\hat{\boldsymbol{X}}_r\right)\right) = \boldsymbol{U}_r\sigma^{(r)}_{spec}(\hat{\boldsymbol{X}}_r).\end{aligned}$$

Therefore, the proposed spectral nonlinear activation is well-defined.

A.3 Proof of Proposition 4.1

Let $\mathcal{S}_r^{(i)}$ denotes the indices associated with the r-th subspace with numerically distinct eigenvalue $\delta_r^{(i)}$ for $\mathcal{G}_i$ with $0 = \delta_1^{(i)} \le \cdots \le \delta_{M_i}^{(i)}$. The spectral convolution of proposed spectral filter gives

$$h(\boldsymbol{L})\boldsymbol{x} = \sum_{i=1}^{n}\sum_{r=1}^{M_i} h^{(r,i)}\sum_{k\in\mathcal{S}_r^{(i)}}\boldsymbol{u}\boldsymbol{u}^T\boldsymbol{x} \tag{10}$$

where $h^{(r,i)}$ is the frequency response of $\delta_r^{(i)}$ for $\mathcal{G}_i$. Let f be the continuous filter defined in spectral domain. By assigning $h^{(r,i)} = f(\delta_r^{(i)})$, equation 10 reduced to

$$f(\boldsymbol{L})\boldsymbol{x} = \sum_{i=1}^{n}\sum_{r=1}^{M_i} f(\delta_r^{(i)})\sum_{k\in\mathcal{S}_r^{(i)}}\boldsymbol{u}\boldsymbol{u}^T\boldsymbol{x} = \sum_{\tau\in\mathcal{T}} f(\tau)\boldsymbol{u}\boldsymbol{u}^T\boldsymbol{x} \tag{11}$$

which is the form of spectral convolution with a unified filter for all graph components. Here, $\mathcal{T}$ is the set of numerically distinct eigenvalues of $\mathcal{G}$.

Let $\hat{\mathbf{h}}^{(i)} = [h^{(1,i)}, h^{(2,i)}, \cdots, h^{(M_i,i)}]$ denotes the filter response sequence for $\mathcal{G}_i$. Notice that the proposed component-wise filter bank generation gives identical filter response for $\mathcal{G}_i$ and $\mathcal{G}_j$ iff $\hat{\mathbf{h}}^{(i)} = \hat{\mathbf{h}}^{(j)}$ when $\mathcal{G}_i$ and $\mathcal{G}_j$ share the same distinct eigenvalues. When $\hat{\mathbf{h}}^{(i)} \neq \hat{\mathbf{h}}^{(j)}$, the proposed filter bank is strictly more expressive since $f(\cdot)$ does not exist that satisfy $h_0^{(i)} = f(0) = h_0^{(j)}$ by taking $\delta_0^{(i)} = \delta_0^{(j)} = 0$ as an example.

A.4 Datasets

Table 5 provides the detail information of datasets, including the number of nodes, number of edges, number of graph components, and the dimension of the largest subspace.

A.5 Experiment Settings for SpecGSN

Table 6 shows the hyperparameter settings for SpecGSN, where $drop_{in}$, $drop_{attn}$, $drop_{feat}$ and $drop_{fc}$ are the dropout settings for the input feature, multi-head attention mechanism in Transformer, the feed forward feature of Transformer, and the features of the classifier (a 2-layer fully connected network) respectively.

Table 5: Datasets used in experiments ($\dim_{max}$ is the dimension of the largest subspace)

Datasets	#nodes	#edges	#graph	$\dim_{max}$	Datasets	#nodes	#edges	#graph	$\dim_{max}$
Cora	2708	10556	78	292	Citeseer	3327	9104	438	335
Pubmed	19717	88648	1	12121	DBLP	17716	105734	589	1984
Computers	13752	491722	314	49	Photo	7650	238162	136	20
Texas	183	325	1	74	Wisconsin	251	515	1	76
Cornell	183	298	1	64	Merged	627	1138	3	76
Roman-empire	22662	65854	1	516	Chameleon	2277	36101	1	1137
Actor	7600	30019	1	1160	Cora-m	223	418	77	3
Citeseer-m	1159	1746	389	6	DBLP-m	1525	1908	588	6

Table 6: Experiment settings

	epoch	lr	decay	weight decay	drop_{in}	drop_{attn}	drop_{feat}	drop_{fc}	num head	ω_{init}	ξ_{sin}	ξ_{cos}	γ
Cora	300	1e-3	0.99	1e-3	0.6	0	0.6	0.5	4	2	1	1	2
Citeseer	300	1e-3	0.99	1e-3	0.7	0	0.7	0.6	1	2	1	1	2
Pubmed	300	1e-3	0.99	1e-3	0.2	0	0.2	0.2	1	2	1	1	2
DBLP	300	1e-3	0.99	1e-3	0.5	0	0.5	0.5	1	2	1	1	2
Computers	1000	1e-3	0.998	1e-5	0.4	0	0.4	0.2	1	2	1	1	2
Photo	1000	1e-3	0.996	1e-3	0.5	0	0.5	0.2	1	2	1	1	2
Texas	300	3e-3	0.99	1e-3	0.7	0	0.7	0.6	1	2	1	1	2
Wisconsin	300	1e-3	0.99	1e-2	0.7	0	0.7	0.6	1	2	1	1	2
Cornell	300	1e-3	0.99	1e-2	0.7	0	0.7	0.6	1	2	1	1	2
Merged	300	1e-3	0.99	3e-3	0.7	0	0.7	0.6	1	2	1	1	2
Roman-empire	1000	1e-2	0.995	1e-4	0.3	0	0.3	0	4	$10^{1/4}$	1	1	8
Chameleon	2000	3e-3	0.998	1e-3	0.8	0.2	0.8	0.7	1	$10^{1/8}$	0	1	16
Actor	1000	1e-3	0.998	5e-4	0.6	0.2	0.6	0.6	1	$10^{1/4}$	1	1	8

A.6 Additional Illustrations

An example of default and component-wise spectral decomposition of $\boldsymbol{L}$ is illustrated in figure 3.

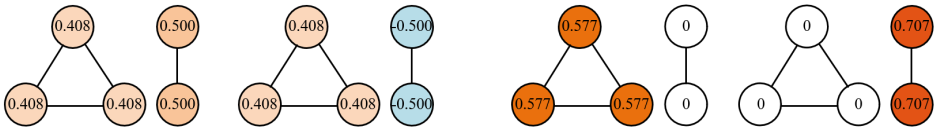


Figure 3: Eigenvectors of $\lambda = 0$ of multi-graph eigenvalue decomposition. Default eigenvalue decomposition may produce basis vectors that couples different graph components (shown in the left two figures), thereby merging their subspaces. Performing eigenvalue decomposition separately on each graph component can avoid this issue.

An illustration of spectra nonlinear activation of a 2-dimensional subspace.

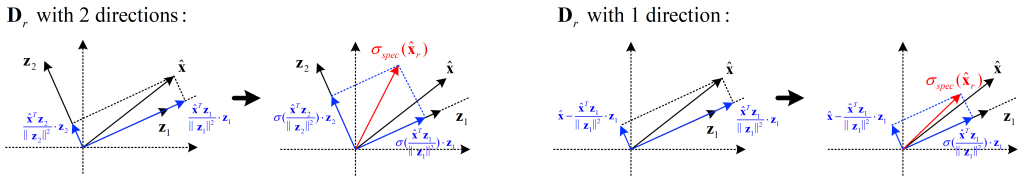


Figure 4: Illustration of spectral nonlinear activation on a 2-dimensional subspace under $\boldsymbol{D}_r$ with different number of directions. $\hat{\boldsymbol{x}}$ is the spectral coefficients, $\boldsymbol{z}_1, \boldsymbol{z}_2$ are selected directions, and $\sigma_{spec}(\hat{\boldsymbol{x}})$ is the spectral coefficients after spectral nonlinear activation.