High-Variance Graph Framelets for Heterophilous Graph Learning

Ruigang Zheng
Department of Mathematics
City University of Hong Kong
Hong Kong, SAR China
ruigzheng2-c@my.cityu.edu.hk

Ming Li
Key Lab of Education Blockchain and Intelligent Technology
Ministry of Education, Guangxi Normal University
Guilin, China
mingli@zjnu.edu.cn

Xiaosheng Zhuang
Department of Mathmetics
City University of Hong Kong
Hong Kong, SAR China
xzhuang7@cityu.edu.hk

***Abstract*—Heterophilous graphs are characterized by connections predominantly occurring between nodes of differing classes, resulting in highly non-smooth or "highly varying" label distributions across the graph. This structural property challenges conventional graph learning methods, which often rely on the assumption of label and feature smoothness. Motivated by the need to better align with the intrinsic heterophily in such graphs, we propose a general parameterized system of high-variance graph framelets. These framelets are designed to generate feature representations that are themselves highly varying, thereby enhancing the expressiveness and discriminative power of node features in heterophilous settings. The proposed high-variance framelets can be flexibly constructed without requiring data leakage or task-specific training, making them a lightweight yet effective addition to existing models. Experimental results on two representative heterophilous graph datasets demonstrate that our method consistently improves node classification accuracy, highlighting the potential of high-variance representations for addressing the challenges of heterophilous graph learning. This work opens up promising avenues for developing more adaptive and theoretically grounded spectral methods, particularly in settings where smoothness assumptions fail to hold.**

***Index Terms*—Graph neural networks, Graph learning, Heterophilous graphs, Node classification, Graph framelets.**

I. Introduction

Due to its flexibility, a variety of complex systems in real life, such as social networks, traffic networks, etc., can be described using graphs. A typical task on graphs is to identify the classes to which each node belongs. Such classification corresponds to determining the group an account belongs to in a social network or the type of traffic conjunction in a traffic network. There are abundant works in the literature that deal with node classification, where graph neural networks (GNNs) approaches are currently one of the most active topics [1], [2]. However, graphs are often assumed to connect similar entities. In node classification, such assumptions mean that connected nodes are more likely to be of the same classes, which is described as being *homophilous*. The datasets adopted in the early works are primarily homophilous and that these GNNs implicitly perform graph smoothing [3], [4]. In recent years, an ongoing active topic is to deal with the counterpart of homophilous graphs, the so-called *heterophilous* graphs [5], [6]. Intuitively, heterophily means that connected nodes are more likely to be of different classes. As a result, such a reversing nature of heterophilous graphs brings challenges to the GNNs that perform smoothing. Other types of GNNs were therefore proposed to handle node classification on heterophilous graphs.

The adaptions for heterophilous graphs in current GNNs are based on two observations: 1) Spatial aspect: For a fixed node, other nodes from the same class are distributed more outside the 1-hop neighborhood. 2) Spectral aspect: Target signals on heterophilous graphs have larger oscillation with respect to the graph Laplacian. Thus, its frequency distribution can hardly be confined to the low-frequency part. GNNs for heterophilous graphs mainly utilize graph-induced aggregations in multiple layers, which resembles the idea in deep convolution neural networks. These GNNs can be further categorized into two types according to whether the aggregations are spatially [7], [8] or spectrally defined [9]–[11]. Recently, there has been a third perspective in GNNs, in which the multiple layers of neural networks are simply linear, and all graph-induced aggregations are only involved in forming inputs for the networks [12], [13]. These methods aim at jointly generating new features for all nodes. Then, with these new features as inputs, the following training is completely supervised. Such approaches have demonstrated superior performances in heterophilous node classification despite not using complicated neural network architectures. In our view, these methods can be categorized as feature engineering for heterophilous graphs.

Inspired by this new perspective, we aim to provide new features that are rich, sparse, and, above all, highly varying. Rich features mean that we can choose among various candidate features. Features being sparse means that when regarded as a column vector for all nodes, the feature vector is sparse and thus alleviates storage and computational burden. Finally, features that are highly varying suit the need for heterophilous graphs. The idea of obtaining highly varying features comes from the mathematical intuition that neural networks are continuous maps, and therefore, similar outputs imply similar inputs. As for heterophilous graphs of which the ground truth outputs are by definition highly varying, this further implies that the inputs

Ming Li acknowledged the support from the Research Fund of Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education (EBME25-F-07). X. Zhuang was supported in part by the Research Grants Council of Hong Kong (Project no. CityU 11309122, CityU 11302023, CityU 11301224, and CityU 11300825) and a grant from the Innovation and Technology Commission of Hong Kong (Project no. MHP/054/22).

should also be highly varying.

To achieve our goal, we propose a general parameterized system of graph framelets based on finite frame theory [14] and multi-resolution analysis (MRA) on graphs [6], [15], [16]. Frames are redundant sets that span the whole vector space. Different from orthogonal bases, the redundancy in frames is robust to the loss of vectors for representing signals. On the other hand, combined with multi-resolution analysis on graphs, vectors in frames will be supported on a portion of the nodes. Consequently, these vectors will have lots of zero terms. Such sparse vectors are thus called framelets. Moreover, the supports of the framelets are induced by a modified graph that only connects each node with its 2-hop neighbors. Assigning values to the node in such supports will intuitively result in vectors that are highly variated with respect to the original graph. Finally, the parameterized system allows us to adjust the number of framelets. The framelets are then sorted according to variance on the graph, and the top ones are selected as the new features. Experiments will demonstrate the effectiveness of our approach.

In summary, the contribution of this paper is as follows: 1) We propose a general parametrized system of graph framelets for generating new features with high variance. 2) We demonstrate the effectiveness of such generated features for node classification on heterophilous graphs.

II. Preliminaries

We first introduce some necessary notation, definitions, and preliminary results. We denote an undirected weighted graph with n vertices as $\mathcal{G} := (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ (or simply $\mathcal{G} := (\mathcal{V}, \mathcal{E})$), where $\mathcal{V} := \{v_1, v_2, \dots, v_n\}$, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, $\boldsymbol{W} = (w_{ij})_{1 \le i,j \le n} \in \mathbb{R}^{n \times n}$ denote the set of vertices, the set of edges, and the weight (adjacency) matrix, respectively. The space $L^2(\mathcal{G}) := \{f \mid \mathcal{V} \to \mathbb{R}\}$ is the collection of graph functions on $\mathcal{G}$ and can be regarded as $\mathbb{R}^n$ with the usual Euclidean inner-product $\langle \cdot, \cdot \rangle$ and induced norm $\|\cdot\| := \sqrt{\langle \cdot, \cdot \rangle}$. The cardinality of a set is denoted by $|\cdot|$.

Let $\boldsymbol{L} := \boldsymbol{D}^{-1/2}(\boldsymbol{I} - \boldsymbol{W})\boldsymbol{D}^{-1/2}$ denote the *normalized graph Laplacian* matrix, where the diagonal matrix $\boldsymbol{D} := \operatorname{diag}(d_1, d_2, \dots, d_n)$ with $d_i := \sum_{j=1}^n w_{ij}, 1 \le i \le n$ being the degree matrix. $\boldsymbol{L}$ is positive semidefinite and has n eigenvalues $0 \le \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ associated with real orthonormal eigenvectors $\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_n$. The quadratic form $\boldsymbol{f}\boldsymbol{L}\boldsymbol{f}^\top, \boldsymbol{f} \in \mathbb{R}^{1 \times n}$ equals

$$2\sum_{e \in \mathcal{E}} \frac{(\boldsymbol{f}_i - \boldsymbol{f}_j)^2}{\sqrt{\boldsymbol{D}_{ii}\boldsymbol{D}_{jj}}}, \quad v_i, v_j \text{ are connected by } e,$$

which measures the variance of a function $\boldsymbol{f}$ on graph by summing (normalized) value differences on all edges. For unit-norm $\boldsymbol{f}$, $\boldsymbol{f}\boldsymbol{L}\boldsymbol{f}^\top \in [0, 2]$ [13].

For $K \in \mathbb{N}$, the set $\{1, \dots, K\}$ is denoted as $[K]$. We denote $\boldsymbol{I}$ the identity matrix of a certain size and omit its size for simplicity. Vectors in $\mathbb{R}^d, d \in \mathbb{N}$ are assumed to be **row vectors**, or equivalently matrices in $\mathbb{R}^{1 \times d}$. Given a set $\{\boldsymbol{X}_i\}_{i=1}^m \subseteq \mathbb{R}^{s \times t}$ of m matrices, we use $[\![\boldsymbol{X}_i]\!]_{i \in [m]} := [\boldsymbol{X}_1^\top, \dots, \boldsymbol{X}_m^\top]^\top$ to denote the matrix in $\mathbb{R}^{ms \times t}$ constructed by concatenating $\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_m$ along the rows.

A finite set of row vectors $\{\boldsymbol{f}_i\}_{i=1}^M$ of $\mathbb{R}^N$ is called a *tight frame* (with constant 1) [14] of $\mathbb{R}^N$ if and only if

$$\|\boldsymbol{g}\|^2 = \sum_{i=1}^M |\langle \boldsymbol{f}_i, \boldsymbol{g} \rangle|^2, \quad \forall \boldsymbol{g} \in \mathbb{R}^N,$$

The condition above is equivalent to

$$\boldsymbol{F}^\top \boldsymbol{F} = \boldsymbol{I}, \quad \boldsymbol{F} := [\![\boldsymbol{f}_i]\!]_{i \in [M]} \tag{1}$$

Thus, we can decompose and reconstruct any $\boldsymbol{g}$ using $\{\boldsymbol{f}_i\}_{i=1}^M$ alone. The focus of this paper is to construct tight graph framelet systems for $L^2(\mathcal{G}) \equiv \mathbb{R}^n$.

Under an abuse of notation, we regard $\operatorname{span}(\boldsymbol{X})$ as the span of the row vectors of a matrix $\boldsymbol{X}$. Since it is finite dimensional, the definition of tight frame on $\operatorname{span}(\boldsymbol{X})$ is similar. We have the following key lemma concerning the characterization of the span of a matrix.

Lemma 1. *Let $\boldsymbol{X} := [\![\boldsymbol{\xi}_i]\!]_{i \in [m]} \in \mathbb{R}^{m \times n}$ be defined from the set $\{\boldsymbol{\xi}_i\}_{i=1}^m$ of orthonormal row vectors in $\mathbb{R}^n$. Let $\boldsymbol{A} \in \mathbb{R}^{m_1 \times m}$ and $\boldsymbol{B} \in \mathbb{R}^{m_2 \times m}$ be two matrices such that $1 \le m_1 < m$. Define $\boldsymbol{\Phi} := [\![\boldsymbol{\varphi}_i]\!]_{i \in [m_1]} := \boldsymbol{A}\boldsymbol{X}$ and $\boldsymbol{\Psi} := [\![\boldsymbol{\psi}_i]\!]_{i \in [m_2]} := \boldsymbol{B}\boldsymbol{X}$. Then the following two statements are equivalent.*

(a) *The matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ satisfy $\boldsymbol{A}\boldsymbol{A}^\top = \boldsymbol{I}$, $\boldsymbol{B}\boldsymbol{A}^\top = \boldsymbol{0}$, and* $\operatorname{rank}(\boldsymbol{B}) = m - m_1$.

(b) $\operatorname{span}(\boldsymbol{X}) = \operatorname{span}(\boldsymbol{\Phi}) \oplus \operatorname{span}(\boldsymbol{\Psi})$ *and the set $\{\boldsymbol{\varphi}_i\}_{i \in [m_1]}$ is an orthonormnal basis for* $\operatorname{span}(\boldsymbol{\Phi})$.

Moreover, with the additional assumption of either (a) or (b), the following statements are equivalent.

(i) $\boldsymbol{A}^\top \boldsymbol{A} + \boldsymbol{B}^\top \boldsymbol{B} = \boldsymbol{I}$.

(ii) $\boldsymbol{\Phi}^\top \boldsymbol{\Phi} + \boldsymbol{\Psi}^\top \boldsymbol{\Psi} = \boldsymbol{X}^\top \boldsymbol{X}$.

(iii) $\boldsymbol{B}\boldsymbol{B}^\top \boldsymbol{B} = \boldsymbol{B}$.

(iv) $\{\boldsymbol{\psi}_i\}_{i \in [m_2]}$ *is a tight frame of* $\operatorname{span}(\boldsymbol{\Psi})$.

III. High-Variance Graph Framelets

A. Construction of $\mathcal{V}$-Framelet Systems

We next provide the details for the construction of framelet systems based on a partition tree for a vertex set $\mathcal{V}$, which are the fundamental structures for our construction of graph framelet systems.

Definition 1. *Let $J \in \mathbb{N}$ and $\mathcal{V} = \{v_1, \dots, v_n\}$ be a set of n vertices. A* partition tree *$\mathscr{T}_J(\mathcal{V})$ of $\mathcal{V}$ with $J+1$ levels is a rooted tree such that*

(a) *The root node $p_{0,1}$ is associated with $\mathcal{S}_{0,1} := \mathcal{V}$.*

(b) *Each leaf $p_{J,k}, k \in [n_J]$ with $n_J := n$ is associated with the singleton $\mathcal{S}_{J,k} := \{v_i\}$. The path from $p_{0,1}$ to each $p_{J,k}$ contains exactly J edges.*

(c) *Each tree node $p_{j,k}, k \in [n_j]$ on the $j \in [J-1]$ level (i.e. the path from $p_{0,1}$ to $p_{j,k}$ contains exactly j edges) is associated with a set $\mathcal{S}_{j,k} \subset \mathcal{V}$ of vertices such that*

  (i) $\cup_{k=1}^{n_j} \mathcal{S}_{j,k} = \mathcal{V}$, *$\mathcal{S}_{j,k_1} \cap \mathcal{S}_{j,k_2} = \varnothing$ for $k_1 \neq k_2$, $1 \le k_1, k_2 \le n_j$, and*

  (ii) $\cup_{k' \in \mathcal{C}_{j,k}} \mathcal{S}_{j+1,k'} = \mathcal{S}_{j,k}$ *with $\mathcal{S}_{j+1,k'} \subsetneq \mathcal{S}_{j,k}$,* *where $n_j < n$ is the number of tree nodes on level j and $\mathcal{C}_{j,k} \subseteq [n_{j+1}]$ is the index set of children nodes of $p_{j,k}$.*

See Figure 1 for an illustration. In short, each level in the partition tree is associated with a partition on $\mathcal{V}$, and the partitions in the lower levels are formed by merging clusters in the higher levels. Note that by the condition in item (c), each non-leaf node has at least 2 children, i.e., $|\mathcal{C}_{j,k}| > 1$.

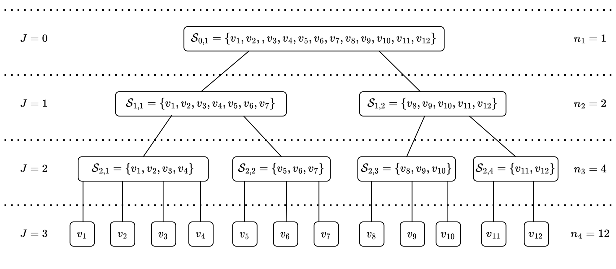

Fig. 1: A partition tree $\mathscr{T}_J(\mathcal{V})$ for a vertex set $\mathcal{V}$ of 12 vertices and with $J = 3$ (4 levels).

In order to construct graph framelet systems analog to the classical wavelet/framelet systems with the multiscale structure, based on a partition tree $\mathscr{T}_J(\mathcal{V})$, we associate each level j with a linear subspace $\mathbb{V}_j$ such that $\mathbb{V}_0 \subsetneq \mathbb{V}_1 \subsetneq \cdots \subsetneq \mathbb{V}_J$. Following Mallat's idea in multi-resolution analysis (MRA) [17], if we have

$$\mathbb{V}_j = \mathbb{V}_{j-1} \oplus \mathbb{W}_{j-1}, \quad \mathbb{V}_{j-1} \perp \mathbb{W}_{j-1}, \quad j \in [J], \tag{2}$$

then by doing the decomposition $J-1$ times, we have

$$\mathbb{V}_J = \mathbb{V}_0 \oplus \mathbb{W}_0 \oplus \mathbb{W}_1 \oplus \cdots \oplus \mathbb{W}_{J-1} \tag{3}$$

and that $\mathbb{V}_0, \mathbb{W}_0, \mathbb{W}_1, \ldots, \mathbb{W}_{J-1}$ are mutually orthogonal. Therefore, to construct a tight frame on $\mathbb{V}_J$, it is sufficient to construct tight frames for each $\mathbb{V}_0, \mathbb{W}_0, \mathbb{W}_1, \ldots, \mathbb{W}_{J-1}$ and take the union.

The partition tree $\mathscr{T}_J(\mathcal{V})$ suggests that each $\mathbb{V}_j$ and $\mathbb{W}_j$ should be further decomposed such that

$$\mathbb{V}_j := \oplus_{k=1}^{n_j} \mathbb{V}_{j,k}, \quad \mathbb{W}_j := \oplus_{k=1}^{n_j} \mathbb{W}_{j,k}, \tag{4}$$

where each element in $\mathbb{V}_{j,k}$ or $\mathbb{W}_{j,k}$ is supported on $\mathcal{S}_{j,k}$ and thus $\mathbb{V}_{j,k}, k \in [n_j]$ (or $\mathbb{W}_{j,k}, k \in [n_j]$) are mutually orthogonal. Furthermore, the nested structure of $\mathscr{T}_J(\mathcal{V})$ further suggests that

$$\oplus_{k' \in \mathcal{C}_{j,k}} \mathbb{V}_{j+1,k'} = \mathbb{V}_{j,k} \oplus \mathbb{W}_{j,k}, \quad \mathbb{V}_{j,k} \perp \mathbb{W}_{j,k} \tag{5}$$

for $j \in \{0, \ldots, J-1\}$ and $k \in [n_j]$.

Assuming that each $\mathbb{V}_{J,k}$ is supported on $\mathcal{S}_{J,k} = \{v_k\}$ at the finest level J, we can define $\mathbb{V}_J := \mathrm{span}\{\boldsymbol{e}_k \,|\, k \in [n]\} = \oplus_{k=1}^{n} \mathbb{V}_{J,k}$, where $\mathbb{V}_{J,k} := \mathrm{span}(\{\boldsymbol{e}_k\})$ is simply the one-dimensional linear space associated with the vertex v_k and generated by the k-th canonical basis vector $\boldsymbol{e}_k := [0, \ldots, 0, 1, 0 \ldots, 0] \in \mathbb{R}^n$. Note that by the parent-children relation, we can rewrite $\mathbb{V}_J$ as

$$\mathbb{V}_J = \oplus_{k \in [n_{J-1}]} (\oplus_{k' \in \mathcal{C}_{J-1,k}} \mathbb{V}_{J,k'}). \tag{6}$$

It is straightforward to see that (4) and (5) implies (2), which eventually implies (3) since we can decompose $\oplus_{k' \in \mathcal{C}_{J-1,k}} \mathbb{V}_{J,k'}$ as

$$\oplus_{k' \in \mathcal{C}_{J-1,k}} \mathbb{V}_{J,k'} = \mathbb{V}_{J-1,k} \oplus \mathbb{W}_{J-1,k} \tag{7}$$

for each $k \in [n_{J-1}]$, and the procedure can continue from bottom $j = J$ to top $j = 0$. Hence, it remains to define a general procedure of decomposition for each non-leaf node in $\mathscr{T}_J(\mathcal{V})$ such that (5) is satisfied. Our next result shows that we indeed can define such a general (bottom-up) procedure and obtain a tight framelet system for $\mathbb{V}_J$ based on Lemma 1 and the partition tree $\mathscr{T}_J(\mathcal{V})$.

Definition 2. *Let $\mathscr{T}_J(\mathcal{V})$ be a partition tree as in Definition 1. Let $\boldsymbol{A}^{[j,k]} \in \mathbb{R}^{1 \times |\mathcal{C}_{j,k}|}$ and $\boldsymbol{B}^{[j,k]} \in \mathbb{R}^{m_{j,k} \times |\mathcal{C}_{j,k}|}$ be two filter matrices associated with the tree node $p_{j,k}$ for $j = 0, \ldots, J-1$ and $k \in [n_j]$.*

(1) (Initialization) *Define $\mathbb{V}_{J,k} := \mathrm{span}(\{\boldsymbol{e}_k\})$ and $\boldsymbol{\Phi}^{[J,k]} := \boldsymbol{e}_k$ for $k \in [n]$. Let $\mathbb{V}_J := \oplus_{k \in [n]} \mathbb{V}_{J,k}$ and $r_J := 1$.*

(2) (Bottom-up Procedure) *Recursively define at each level j from $J-1$ to 0:*

   (a) *For each tree node $p_{j,k}$, obtain $\mathbb{V}_{j,k} := \mathrm{span}(\boldsymbol{\Phi}^{[j,k]})$ and $\mathbb{W}_{j,k} = \mathrm{span}(\boldsymbol{\Psi}^{[j,k]})$ by Lemma 1 with subspace matrices $\{\boldsymbol{X}_{k'} := \boldsymbol{\Phi}^{[j+1,k']} \,|\, k' \in \mathcal{C}_{j,k}\}$ and the two filter matrices $\boldsymbol{A}^{[j,k]}$ and $\boldsymbol{B}^{[j,k]}$. More precisely, let $\boldsymbol{X} := [\![\boldsymbol{X}_{k'}]\!]_{k' \in \mathcal{C}_{j,k}}$. Then we have*

$$\boldsymbol{\Phi}^{[j,k]} = \boldsymbol{A}^{[j,k]} \boldsymbol{X}, \quad \boldsymbol{\Psi}^{[j,k]} = \boldsymbol{B}^{[j,k]} \boldsymbol{X}, \tag{8}$$

   where $\boldsymbol{A}^{[j,k]}, \boldsymbol{B}^{[j,k]}$ are matrices corresponding to $\boldsymbol{A}, \boldsymbol{B}$ in Lemma 1.

   (b) *Define $\mathbb{V}_j := \mathrm{span}(\boldsymbol{\Phi}_j)$ with $\boldsymbol{\Phi}_j := [\![\boldsymbol{\Phi}^{[j,k]}]\!]_{k \in [n_j]}$ and $\mathbb{W}_j := \mathrm{span}(\boldsymbol{\Psi}_j)$ with $\boldsymbol{\Psi}_j := [\![\boldsymbol{\Psi}^{[j,k]}]\!]_{k \in [n_j]}$.*

(3) (Finalization) *For each $J_0 = 0, \ldots, J$, define the $\mathcal{V}$-*framelet system *associate with the partition tree $\mathscr{T}_J(\mathcal{V})$ and determined by the filter matrices as*

$$\begin{aligned} \mathcal{F}_{J_0}^J(\mathcal{V}) :=& \mathcal{F}_{J_0}(\{(\boldsymbol{A}^{[j,k]}, \boldsymbol{B}^{[j,k]}) \,|\, k \in [n_j]\}_{j=J_0}^{J-1}) \\ :=& \boldsymbol{\Phi}_{J_0} \cup \boldsymbol{\Psi}_{J_0} \cup \cdots \cup \boldsymbol{\Psi}_{J-1}. \end{aligned} \tag{9}$$

Now we are ready to state the main theorem for the construction of a general tight $\mathcal{V}$-framelet system for $\mathbb{V}_J$.

Theorem 1. *Adopt the notations in Definition 2. Assume that the filter matrices $\boldsymbol{A}^{[j,k]}$ and $\boldsymbol{B}^{[j,k]}$ satisfy*

$$\boldsymbol{A}^{[j,k]} (\boldsymbol{A}^{[j,k]})^\top = \boldsymbol{I}, \tag{10}$$

$$\boldsymbol{B}^{[j,k]} (\boldsymbol{A}^{[j,k]})^\top = \boldsymbol{0}, \tag{11}$$

$$(\boldsymbol{B}^{[j,k]})^\top \boldsymbol{B}^{[j,k]} = \boldsymbol{I} - (\boldsymbol{A}^{[j,k]})^\top \boldsymbol{A}^{[j,k]}, \tag{12}$$

for $j \in \{0, \ldots, J-1\}$ and $k \in [n_j]$. Then the following statements hold.

(i) *$\mathbb{V}_j = \oplus_{k=1}^{n_j} \mathbb{V}_{j,k}$, $\mathbb{W}_j = \oplus_{k=1}^{n_j} \mathbb{W}_{j,k}$ for $j = 0, \ldots, J-1$, and $\mathbb{V}_j = \mathbb{V}_{j-1} \oplus \mathbb{W}_{j-1}, \mathbb{V}_{j-1} \perp \mathbb{W}_{j-1}$ for $j \in [J]$.*

(ii) *$\mathbb{V}_J = \mathbb{V}_j \oplus \mathbb{W}_j \oplus \mathbb{W}_{j+1} \oplus \cdots \oplus \mathbb{W}_{J-1}$ for $j = 0, \ldots, J-1$. In particular, $\mathbb{V}_J = \mathbb{V}_0 \oplus \mathbb{W}_0 \oplus \mathbb{W}_1 \oplus \cdots \oplus \mathbb{W}_{J-1}$.*

(iii) *$\mathcal{F}_{J_0}^J(\mathcal{V})$ is a tight frame for $\mathbb{V}_J$ for all $J_0 = 0, \ldots, J-1$.*

Remark 1. Note that the construction of $\mathcal{V}$-framelet systems only involves the vertex set $\mathcal{V}$ and the partition tree $\mathscr{T}_J(\mathcal{V})$. No graph structure, such as the edge set $\mathcal{E}$ or the adjacency matrix, is involved. Such construction based on the hierarchical partitions of a vertex set has the benefit of providing a mathematical generality. We discuss the graph framelet systems in the next subsection that utilize both the vertex set $\mathcal{V}$ and the edge set $\mathcal{E}$.

B. Construction of Framelet Systems with High Variance

As we point out in Remark 1, the $\mathcal{V}$-framelet systems do not involve the other graph structure such as the edge set $\mathcal{E}$. However, they do depend on the partition tree $\mathscr{T}_J(\mathcal{V})$ and how to obtain such a partition tree is closely related to the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$. We next discuss how we can obtain the partition tree from a given graph $\mathcal{G}$ and define the so-called graph-involved $\mathcal{G}$-framelet systems.

We first discuss the realization of the nested structure of a partition tree. It is known that edges in graphs intuitively represent a notion of proximity among nodes. Thus, to represent scales (levels) on graphs analog to the Euclidean domains, a common practice is to apply a series of clustering (coarsing) on graphs. In detail, given $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ with $\mathcal{V} := \{v_1, \ldots, v_n\}$, the clusters on $\mathcal{V}$ are formed through certain algorithms, e.g., K-means clustering, based on $\mathcal{E}$ such that connected nodes are more likely to be in the same clusters. Assuming that the clustering algorithm on $\mathcal{G}$ resulted in n' clusters, denoted as $\mathcal{V}' = \{v'_1, \ldots, v'_{n'}\}$, where each $v'_i = \{v_{i_1}, \ldots, v_{i_{n_i}}\} \subseteq \mathcal{V}$, $v'_i \cap v'_j = \varnothing$ for $i \neq j$, and $\cup_{i=1}^{n'} v'_i = \mathcal{V}$. Then a graph $\mathcal{G}' := (\mathcal{V}', \mathcal{E}', \boldsymbol{W}')$ can be formed from these clusters through the definition of its adjacency matrix $\boldsymbol{W}' = (w'_{ij})_{1 \le i,j \le n'}$ as

$$w'_{ij} := \sum_{p \in v'_i} \sum_{q \in v'_j} w_{pq}, \quad i, j = 1, \ldots, n',$$

where w'_{ij} is the weight between $v'_i, v'_j \in \mathcal{V}'$ while w_{pq} is the original weight between vertices $p, q \in \mathcal{V}$.

In summary, the nodes of $\mathcal{G}'$ represent clusters and the edge weight on $\mathcal{G}'$ is determine by summing of the edge weight among nodes of two clusters. We called $\mathcal{G}'$ a *coarse-grained graph* of $\mathcal{G}$ [16]. Based on the coarse-grained graph, we can give the definition of multi-graph partition trees.

Definition 3. *A* multi-graph partition tree $\mathscr{T}_J(\mathcal{G})$ *of* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ *with* $J+1, J \in \mathbb{N}$ *levels is a partition tree* $\mathscr{T}_J(\mathcal{V})$ *as defined in Definition 1 such that each* $j \in \{0, \ldots, J\}$ *is associated with a coarse-grained graph* $\mathcal{G}^j = (\mathcal{V}^j, \mathcal{E}^j)$ *of* $\mathcal{G}$ *and* $\mathcal{V}^j = \{\mathcal{S}_{j,k} \,|\, k \in [n_j]\}$. *In particular,* $\mathcal{G}^J \equiv \mathcal{G}$, *where we consider a vertex in* $\mathcal{G}^J$ *as a cluster of singleton.*

An intuitive and equivalent interpretation of the definition above is that a multi-graph partition tree is a partition tree by generating successively forming coarse-grained graphs $\mathcal{G}^{J-1}, \mathcal{G}^{J-1} \ldots, \mathcal{G}^0$ such that $\mathcal{G}^j$ is a coarse-grained graph of $\mathcal{G}^{j+1}, j \in [J-1]$. Since $\mathscr{T}_J(\mathcal{G})$ is associated with a partition tree $\mathscr{T}_J(\mathcal{V})$, we can construct $\mathcal{V}$-framelet systems as in Definition 2.

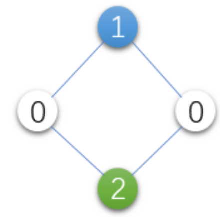

Fig. 2: Assigning non-zero values to only the blue node and its 2-hop neighbor (the green node). Variance occurs on all edges.

However, it is obvious that such constructions result in piecewise constant functions on graphs, in which the pieces are precisely the supports $\mathcal{S}_{j,k}$. Therefore, the variance is generally smaller for framelets in larger scales (i.e., smaller J) since there is no value difference on larger clusters of connected nodes. Such a phenomenon contradicts the purpose of generating high-variance framelets. To solve this problem, instead, we form coarse-grained graphs based on a modified graph $\mathcal{G}_{\text{2hop}}$ at the beginning. In detail, $\mathcal{G}_{\text{2hop}}$ consists of the same node set of $\mathcal{G}$, and a new edge set which is constructed by connecting each node with nodes that are precisely two hops away in $\mathcal{G}$, i.e. connecting node pairs with shortest-path distance of 2 (assuming $\mathcal{G}$ is an unweighted graph). By using $\mathcal{G}_{\text{2hop}}$, we expect the supports in the partition tree contain nodes that are "scattering" with respect to the original $\mathcal{G}$. The intuition of such constructions can be understood in Figure 2. Each edge on circle graphs has a value difference if we skippingly assign values to the nodes, even if we assign the same value. We have the following definition.

Definition 4. *Let* $\mathscr{T}_J(\mathcal{G}_{2hop})$ *be a multi-graph partition tree associated with a induced* 2*-hop graph* $\mathcal{G}_{2hop}$ *from* $\mathcal{G}$. *Let* $\{\boldsymbol{A}^{[j,k]}, \boldsymbol{B}^{[j,k]} \,|\, j = 0, \ldots, J-1; k \in [n_j]\}$ *be the filter matrices satisfying the assumptions of Theorem 1. Then for* $J_0 = 0, \ldots, J-1$, *the* $\mathcal{V}$*-framelet system* $\mathcal{F}_{J_0}^J(\mathcal{V})$ *as in* (9) *is a tight frame for* $\mathbb{R}^n$, *which we call a* ***high-variance framelet (HVF) system****. In such a case, we simply denote* $\mathcal{F}_{J_0}^J(\mathcal{G}) := \mathcal{F}_{J_0}^J(\mathcal{V})$ *to indicate the role played by* $\mathcal{G}$. *In particular, we define for the special case* $J_0 = 0$, *the HVF system* $\mathcal{F}_0^J(\mathcal{G})$ *as*

$$\mathsf{HVF}_J(\mathcal{G}) := \mathcal{F}_0^J(\mathcal{G}) = \boldsymbol{\Phi}_0 \oplus \boldsymbol{\Psi}_0 \oplus \cdots \oplus \boldsymbol{\Psi}_{J-1}. \tag{13}$$

C. Further Details

Since $\mathbb{V}_0$ is one-dimensional, we want its basis vector to be the constant unit-norm vector, which has the least variance. To do so, we simply let each $\boldsymbol{A}^{[j,k]}$ be the constant unit-norm vector of its relevant size. As each $\boldsymbol{A}^{[j,k]}$ is decided, according to Lemma 1, each $\boldsymbol{B}^{[j,k]}$ should be a tight frame of the vector space orthogonal to $\operatorname{span}(\boldsymbol{A}^{[j,k]})$. The simplest example is letting $\boldsymbol{B}^{[j,k]}$ be the remaining $|\mathcal{C}_{j,k}| - 1$ orthonormal vectors in the complement of $\operatorname{span}(\boldsymbol{A}^{[j,k]})$, which can be computed using singular value decomposition. However, in this case,

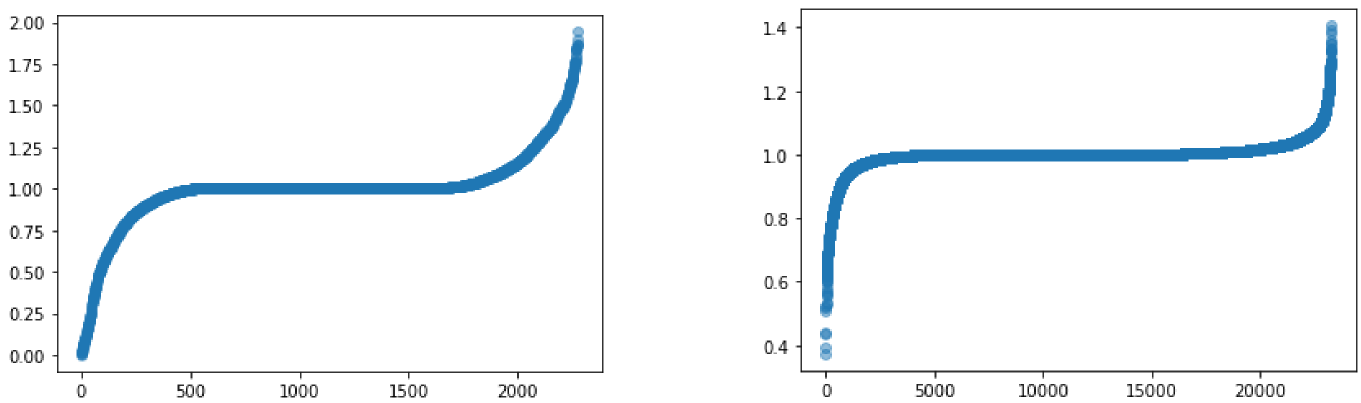

Fig. 3: Distributions of variance on Chameleon of unit-norm vectors. Left: 2277 eigenvectors; Right: 23239 framelets, sorted non-decreasingly according to $\boldsymbol{f}\boldsymbol{L}\boldsymbol{f}^{\top}$.

the $\mathsf{HVF}_J(\mathcal{G})$ will be an orthonormal basis, which has no redundancy. Instead, we find a tight frame $\boldsymbol{F}^{[j,k]}$ in $\mathbb{R}^{|\mathcal{C}_{j,k}|-1}$ as the coefficient matrix for remaining orthonormal vectors (denoted as $\tilde{\boldsymbol{B}}^{[j,k]}$). Then the $\boldsymbol{B}^{[j,k]} := \boldsymbol{F}^{[j,k]}\tilde{\boldsymbol{B}}^{[j,k]}$ will be our final $\boldsymbol{B}$ filters. In details, $\boldsymbol{F}^{[j,k]}$ is obtained via frame completion [18] given $|\mathcal{C}_{j,k}| - 1$ vectors of size $|\mathcal{C}_{j,k}| - 1$ sampled from Gaussian distribution. In this way, the size of $\boldsymbol{F}^{[j,k]}$ is $(|\mathcal{C}_{j,k}| - 1)^2 \times (|\mathcal{C}_{j,k}| - 1)$.

IV. Experiments

A. Settings

We conducted experiments on two heterophilous graphs Chameleon and Squirrel from [19],. These graphs are actually directed, and the adjacency matrices have many zeros columns. To form the $\mathsf{HVF}_J(\mathcal{G})$, we still needed an undirected graph and its 2-hop modified version. This was done by converting the directed graph into an undirected one, in which each directed edge became undirected. In forming the partition tree, we controlled that each $|\mathcal{C}_{j,k}| \leq 16$ by adopting the Python package *scikit-network*. Then the $\mathsf{HVF}_J(\mathcal{G})$ was formed as described in Def. 2. Each framelet was further normalized to have unit norm and sorted non-increasingly according to $\boldsymbol{f}\boldsymbol{L}\boldsymbol{f}^{\top}$. We selected the top 3000 framelets as another feature matrix's columns, which we denoted as $\boldsymbol{F}$. See Fig. 3 for a comparison between the eigenvectors of $\boldsymbol{L}$ and the framelets. Take Chameleon as an example; the majority of framelets have variance ≥ 1, and the number of such framelets (18213) is almost 9 times the number of eigenvectors (2277). This shows the $\mathsf{HVF}_J(\mathcal{G})$'s capability of generating rich high-variance framelets.

As for the network, we used the same architecture and training setup as in [12] except that we replaced the input channels with two matrices $\{\boldsymbol{X}, \boldsymbol{F}\}$ where $\boldsymbol{X}$ is the original node feature matrix. The detailed definition is as follows:

$$\begin{aligned}\boldsymbol{H}_1 &= \alpha_1 \cdot \mathrm{RN}(\boldsymbol{X}\boldsymbol{W}_1) \parallel \alpha_2 \cdot \mathrm{RN}(\boldsymbol{F}\boldsymbol{W}_2),\\ \hat{\boldsymbol{Y}} &= \mathrm{softmax}(\mathrm{ReLU}(\boldsymbol{H}_1)\boldsymbol{W}_3),\end{aligned}$$

where $\parallel$ denotes concatenation operation along columns, $\alpha_1, \alpha_2 \in (0,1)$ are trainable attention weights satisfying $\alpha_1 + \alpha_2 = 1$, RN is the row normalization operation, and $\{\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_3\}$ are trainable parameters.

TABLE I: Dataset statistics, classification accuracy. Results are averaged over 10 public data splits. Percentage: training, 48%; validation, 32%; testing: 20%.

	Chameleon	Squirrel
Node	2,277	5,201
Feature	2,325	2,089
Edge	36,101	217,073
Class	5	5
Framelet	23239	53667
MLP($\boldsymbol{X}$) [12]	46.05	30.24
Ours	**53.42**	**32.69**

B. Avoiding Improper Uses of Adjacency Matrices

The datasets Chameleon and Squirrel are commonly used in heterophilous GNNs for benchmarking. However, there is some issue concerning the strong relation between the connection patterns and the classes of nodes. In fact, it is pointed in [20] that almost half of the nodes are duplicates with identical connection patterns and classes but different features. This is considered as data leakage in [20] and is a crucial factor for achieving high classification accuracy. Following this perspective, we can say that almost all GNNs for heterophilous graphs exploit such data leakage since it is very common to apply aggregation using adjacency matrices. Nonetheless, one might still argue that this is a nice property of the datasets, while it might be too "nice".

For the reasons above, we avoid using the adjacency matrices as aggregations or input features in our model. However, this also makes comparison with a variety of GNNs unreasonable since they do directly apply the adjacency matrices. Therefore, we intend to show that it is possible to improve classification without using adjacency matrices in such ways. This can be seen in Table I in which the results of a 2-layer multilayer preceptron (MLP) are also presented. Note that our method relies on adjacency matrices indirectly by forming coarse-grained graphs and that the goal is to obtain high-variance framelets as features. With $\boldsymbol{F}$ alone as new features, our method still leads to noticeable increases in accuracy.

V. Conclusion and Further Remarks

In this paper, we have proposed a novel design of high-variance graph framelets to generate highly varying features that complement the low-frequency information typically captured in spectral graph methods. These high-variance framelets serve as supplemental features that enhance the expressive power of graph neural networks, especially in the context of heterophilous graphs where traditional message-passing architectures tend to underperform. Our approach is lightweight, intuitively motivated, and avoids the pitfalls of overfitting to specific benchmarks, as it does not rely on data leakage, which remains a critical issue in many existing datasets.

Despite these advantages, the current framework is primarily guided by heuristic and intuitive reasoning. There remains significant room for refining both the theoretical and practical aspects of our design. In particular, a more systematic study on the mathematical selection and characterization of high-variance framelets, possibly guided by task-specific objectives, would enhance the robustness and generality of our method. Additionally, while our current implementation uses fixed framelet bases, it is a promising direction to explore parameterized or learnable framelet systems that can be optimized end-to-end within a graph learning pipeline. From a spectral perspective, our proposed method offers richer spanning sets with substantially more high-frequency components compared to Laplacian eigenvectors, making it a promising building block for the design of future spectral GNN architectures. Nevertheless, a more rigorous comparison with alternative basis functions, as well as a deeper understanding of their convergence properties and stability, is necessary. Experimentally, further validation is needed on newly proposed heterophilous benchmarks that mitigate data leakage, to better assess the practical effectiveness of our method in realistic scenarios. Beyond empirical evaluations, extending our analysis to other graph tasks such as link prediction or community detection under heterophily is another natural step.

Importantly, a compelling future research direction lies in extending our work to heterophilous hypergraph learning [21]. Hypergraphs provide a powerful tool to capture high-order relationships, and applying high-variance framelet ideas in this setting could significantly advance the modeling of heterophilous interactions among groups of nodes, rather than just pairs. Designing hypergraph framelets that preserve or amplify variance across hyperedges—particularly under varying degrees of heterophily—requires new theoretical formulations and computational strategies, but it holds great promise for expanding the applicability of spectral techniques to more complex, non-pairwise graph structures.

In summary, our study opens several promising avenues for exploration, both in terms of theoretical depth and practical breadth. By improving the mathematical formulation, enhancing empirical robustness, and generalizing to the hypergraph domain, especially under heterophily, we expect this line of work to contribute meaningfully to the development of more expressive and versatile graph representation learning frameworks.

Finally, we would like to point out that our work presented in this paper is a specific realization of a general system proposed by us in [22], where detailed discussions and results concerning different aspects, such as proofs of the theorems, computation and storage complexity for constructing the framelets, implementation details, and experiments on larger heterophilous graphs under alternative perspectives and boarder comparison are given.

References

[1] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, “A comprehensive survey on graph neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021.

[2] M. Li, A. Micheli, Y. G. Wang, S. Pan, P. Lió, G. S. Gnecco, and M. Sanguineti, “Guest editorial: Deep neural networks for graphs: Theory, models, algorithms, and applications,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 4367–4372, 2024.

[3] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *ICLR*, 2017.

[4] Q. Li, Z. Han, and X.-M. Wu, “Deeper insights into graph convolutional networks for semi-supervised learning,” in *AAAI*, 2018, pp. 3538–3545.

[5] X. Zheng, Y. Wang, Y. Liu, M. Li, M. Zhang, D. Jin, P. S. Yu, and S. Pan, “Graph neural networks for graphs with heterophily: A survey,” *arXiv preprint arXiv:2202.07082*, 2022.

[6] J. Li, R. Zheng, H. Feng, M. Li, and X. Zhuang, “Permutation equivariant graph framelets for heterophilous graph learning,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 9, no. 35, pp. 11 634–11 648, 2024.

[7] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” in *ICLR*, 2018.

[8] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *NIPS*, vol. 30, 2017.

[9] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *NIPS*, vol. 29, 2016.

[10] M. He, Z. Wei, H. Xu *et al.*, “BernNet: Learning arbitrary graph spectral filters via bernstein approximation,” in *NeulPS*, 2021, pp. 14 239–14 251.

[11] X. Zheng, B. Zhou, J. Gao, Y. Wang, P. Lió, M. Li, and G. Montufar, “How framelets enhance graph neural networks,” in *ICML*, 2021.

[12] S. K. Maurya, X. Liu, and T. Murata, “Simplifying approach to node classification in graph neural networks,” *Journal of Computational Science*, vol. 62, p. 101695, 2022.

[13] K. Huang, Y. G. Wang, M. Li, and P. Lio, “How universal polynomial bases enhance spectral graph neural networks: Heterophily, over-smoothing, and over-squashing,” in *ICML*, 2024, pp. 20 310–20 330.

[14] P. G. Casazza, G. Kutyniok, and F. Philipp, “Introduction to finite frame theory,” *Finite frames: theory and applications*, pp. 1–53, 2013.

[15] Y. Xiao and X. Zhuang, “Adaptive directional haar tight framelets on bounded domains for digraph signal representations,” *Journal of Fourier Analysis and Applications*, vol. 27, pp. 1–26, 2021.

[16] X. Zheng, B. Zhou, Y. G. Wang, and X. Zhuang, “Decimated framelet system on graphs and fast g-framelet transforms,” *Journal of Machine Learning Research*, vol. 23, no. 18, pp. 1–68, 2022.

[17] S. G. Mallat, “A theory for multiresolution signal decomposition: the wavelet representation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[18] J. Ji and L. Wang, “Construction of finite tight frames via singular value decomposition,” *Linear and Multilinear Algebra*, vol. 70, no. 19, pp. 4558–4567, 2022.

[19] H. Pei, B. Wei, K. C.-C. Chang, Y. Lei, and B. Yang, "Geom-GCN: Geometric graph convolutional networks," in *ICLR*, 2020.

[20] O. Platonov, D. Kuznedelev, M. Diskin, A. Babenko, and L. Prokhorenkova, "A critical look at the evaluation of gnns under heterophily: Are we really making progress?" in *ICLR*, 2023.

[21] M. Li, Y. Gu, Y. Wang, Y. Fang, L. Bai, X. Zhuang, and P. Lio, "When hypergraph meets heterophily: New benchmark datasets and baseline," in *AAAI*, 2025, pp. 18 377–18 384.

[22] R. Zheng and X. Zhuang, "Data-adaptive graph framelets with generalized vanishing moments for graph machine learning," 2025. [Online]. Available: https://arxiv.org/abs/2309.03537