Decimated Hypergraph Framelets Based on Spline Wavelets Basis

Chenghao Qiu[1], Ming Li[2], and Xiaosheng Zhuang[1]

[1] Department of Mathematics, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong SAR, China
[2] Zhejiang Key Laboratory of Intelligent Education Technology and Application, Zhejiang Normal University, Jinhua, China
chenghqiu2-c@my.cityu.edu.hk, mingli@zjnu.edu.cn, xzhuang7@cityu.edu.hk

Abstract. Hypergraph Neural Networks (HGNNs) can effectively model high-order relational data but lack robust multi-scale analysis tools and struggle with heterophilic hypergraphs. To address these issues, this paper proposes SWDHFN, a novel Spline-Wavelet-based Decimated Hypergraph Framelets Neural Network. Inspired by Haar wavelets and spline theory, SWDHFN embeds hypergraph structures into spline function spaces to encode multi-entity interactions. SWDHFN leverages the smoothness of spline bases into hypergraph modeling. This ensures the continuity and smooth transition of feature encoding and avoids noise interference during neighborhood information aggregation. We also propose the fast framelet transforms for the spline-Wavelet-based decimated hypergraph framelets, with computational complexity $\mathcal{O}(N \log N)$. Experimental results on benchmark datasets show that SWDHFN achieves state-of-the-art (SOTA) performance on both homophilic and heterophilic hypergraphs, outperforming traditional HGNNs and framelet-based models. Its sparse matrix optimization ensures superior computational scalability.

Keywords: Hypergraph neural networks · hypergraph framelets · decimated framelets · wavelets · splines· heterophily.

1 Introduction

Hypergraphs generalize traditional graphs by allowing hyperedges to connect arbitrary subsets of vertices, providing a natural framework for modeling high-order multi-entity interactions (e.g., group collaborations, multi-modal data associations [1]). Hypergraph Neural Networks (HGNNs) [11] extend graph neural network principles via vertex-hyperedge-vertex propagation, but face two critical limitations: insufficient multi-resolution analysis tools to capture hierarchical features across scales, and computational inefficiencies that hinder scalability to large-scale hypergraphs.

Framelets, as wavelet extensions, enable robust multi-scale signal analysis by decomposing data into global low-frequency trends and local high-frequency

details [5, 12]. While graph framelet systems have been developed for traditional graphs, their adaptation to hypergraphs—especially for capturing complex high-order structures—remains underexplored [8].

To address these gaps, this paper proposes a novel multi-scale decimated hypergraph framelet framework. Inspired by Haar wavelets and spline theory, the framework embeds hypergraphs into spline function spaces, constructs a decimated hypergraph framelet system, and provides efficiency via sparse matrix operations. Experimental results on benchmark datasets demonstrate state-of-the-art performance in node classification and clustering, with significant improvements in computational speed and scalability. This work establishes a new mathematical foundation for integrating framelet theory with hypergraph learning, tackling the challenge of efficient multi-scale analysis for high-order relational data.

Our contributions are summarized as follows:

1) Theoretical Foundation for Decimated Hypergraph Framelets. We construct decimated hypergraph framelet systems on hypergraphs with a global orthogonal B-spline wavelet basis. This framework enables a stable, sparse representation of hypergraph signals by leveraging the smoothness of B-splines and the high-order relationship modeling of hypergraphs, thereby filling the theoretical gap in hypergraph wavelet processing.
2) Spline Wavelet-Based Decimated Hypergraph Framelet Neural Network for Semi-Supervised Learning. We propose a spline wavelet-based decimated hypergraph framelets neural network (SWDHFN) for semi-supervised node classification. The network fuses multiscale spatial information and spectral information, effectively capturing short-range, long-range, and hierarchical hypergraph structures.

The remainder of this paper is organized as follows. Preliminaries are given in Section 2. Details of the constructions of spline wavelet-based decimated hypergraph framelets and hypergraph framelet neural networks are given in Sections 3 and 4, respectively. Numerical experiments are given in Section 5, and conclusions are drawn in the last section.

2 Preliminaries

In this section, we introduce some basic notation and properties on graphs, hypergraphs, and spline wavelets, which are based on the works of [15], [23], [22], and [14].

2.1 Hypergraph and Chains

We first define a weighted hypergraph as $\mathcal{H} = (V, \mathcal{E}, \boldsymbol{w}_V, \boldsymbol{w}_\mathcal{E})$, where V denotes the vertex set, $\mathcal{E}$ denotes the hyperedge set, $\boldsymbol{w}_V = \{w_V(v) \mid v \in V\}$ represents vertex weights and $\boldsymbol{w}_\mathcal{E} = \{w_\mathcal{E}(e) \mid e \in \mathcal{E}\}$ denotes hyperedge weights. Note that each hyperedge $e \in \mathcal{E}$ is a subset of V satisfying $\bigcup_{e\in\mathcal{E}} e = V$.

A hypergraph signal g on $\mathcal{H}$ is a real-valued function $g : V \to \mathbb{R}^n$ (vector form $\boldsymbol{g} = [g(v_1), \ldots, g(v_n)]^\top \in \mathbb{R}^n$). For a vertex set $v \in V$, its degree is defined by $d(v) = \sum_{\{e \in \mathcal{E} | v \in e\}} w_{\mathcal{E}}(e)$. For a hyperedge $e \in \mathcal{E}$, its degree is defined to be the cardinality of e: $\delta(e) = |e|$. We say that there is a hyperpath between vertices v_1 and v_k when there is an alternative sequence of distinct vertices and hyperedges $v_1, e_1, v_2, e_2, \ldots, e_{k-1}, v_k$ such that $\{v_i, v_{i+1}\} \subseteq e_i$ for $1 \leq i \leq k-1$. A hypergraph $\mathcal{H}$ can be represented by a $|V| \times |\mathcal{E}|$ matrix H with entries $H(v, e) = 1$ if $v \in e$ and 0 otherwise, called the incidence matrix of $\mathcal{H}$. Then $d(v) = \sum_{e \in \mathcal{E}} w_{\mathcal{E}}(e) h(v, e)$ and $\delta(e) = \sum_{v \in V} H(v, e)$. Let D_V and $D_{\mathcal{E}}$ denote the diagonal matrices containing the vertex and hyperedge degrees, respectively, and let W denote the diagonal matrix containing the weights of hyperedges. Then the normalized adjacency matrix $\tilde{A}$ of hypergraph $\mathcal{H}$ is defined as $\tilde{A} = D_V^{-1/2} H W D_{\mathcal{E}}^{-1} H^\top D_V^{-1/2}$, where $H^\top$ is the transpose of H.

A hypergraph $\mathcal{H}_c = (V_c, \mathcal{E}_c, \boldsymbol{w}_{V,c}, \boldsymbol{w}_{\mathcal{E},c})$ is called a coarse-grained hypergraph of $\mathcal{H}$ if $V_c = \{C_1, \ldots, C_k\}$ $(k > 1)$ is a partition of V (i.e., $\bigcup_{i=1}^k C_i = V$ and $C_i \cap C_j = \emptyset$ for $i \neq j$), $w_{V,c}(C_i) = \sum_{v \in C_i} w_V(v)$ for each cluster $C_i \in V_c$, $\mathcal{E}_c = \{\tilde{e} \mid e \in \mathcal{E}, |\tilde{e}| \geq 2\}$ with $\tilde{e} = \{C_i \in V_c \mid C_i \cap e \neq \emptyset\}$, and $w_{\mathcal{E},c}(\tilde{e}) = \sum_{\substack{e \in \mathcal{E} \\ \tilde{e} = \{C_i | C_i \cap e \neq \emptyset\}}} w_{\mathcal{E}}(e)$. For integers $J \geq J_0$, a hypergraph coarsening chain $\mathcal{H}_{J \to J_0} := (\mathcal{H}_J, \mathcal{H}_{J-1}, \ldots, \mathcal{H}_{J_0})$ of $\mathcal{H}$ is a sequence where $\mathcal{H}_J \equiv \mathcal{H}$ and each $\mathcal{H}_j$ $(J_0 \leq j < J)$ is a coarse-grained hypergraph of $\mathcal{H}_{j+1}$. The hypergraph distance $\rho_{\mathcal{H}_j}(u, v)$ on $\mathcal{H}_j$ is the shortest weighted path length between u and v, with edge length $l(e) = 1/w_j(a, b)$ for $e = (a, b)$:

$$\rho_{\mathcal{G}_j}(u, v) = \min \left\{ \sum_{e \in P} \frac{1}{w_j(a, b)} \,\middle|\, P \text{ is a path from } u \text{ to } v \right\}.$$

2.2 Non-uniform Spline Wavelets on the Interval

Let $\boldsymbol{t} = (t_i)_{i=1}^{n+m}$ and $\boldsymbol{\tau} = (\tau_i)_{i=1}^{n}$ be two knot vectors in $[0, 1]$, i.e., knots t_i, τ_i are points in $[0, 1]$. A knot vector t can be obtained from τ by inserting m new knots $\boldsymbol{s} = (s_i)_{i=1}^{m}$. Let $S_{d,\boldsymbol{t}}$ denote the space of d-degree B-splines on the knot vector $\boldsymbol{t}$. Let

$$\begin{aligned} \phi_i &\in S_{d,\boldsymbol{\tau}}, \quad i = 1, \ldots, n, \\ \gamma_i &\in S_{d,\boldsymbol{t}}, \quad i = 1, \ldots, n+m. \end{aligned}$$

We also set the spaces

$$V_0 = S_{d,\boldsymbol{\tau}} \quad \textit{and} \quad V_1 = S_{d,\boldsymbol{t}}.$$

And denote the orthogonal complement W of V_0 in V_1 so that

$$V_0 \oplus W = V_1.$$

A function ψ in V_1 with index support $[\ell : r]$, where $r \geq \ell$, is a linear combination of $r - \ell + 1$ fine B-splines $\boldsymbol{\gamma} = (\gamma_\ell, \ldots, \gamma_r)^\top$

$$\psi = \sum_{j=\ell}^{r} w_j \gamma_j = \boldsymbol{\gamma}^\top \boldsymbol{w},$$

and is zero off the interval $[t_\ell, t_{r+d+1}]$. Let us assume that there are $p = p(\ell, r) \geq 0$ old knots in the open interval (t_ℓ, t_{r+d+1}), namely $\tau_{k+1}, \ldots, \tau_{k+p}$, for some $k = k(\ell, r)$. There are then $p + d + 1$ coarse B-splines $\boldsymbol{\phi} = (\phi_{k-d}, \ldots, \phi_{k+p})^\top$, whose supports intersect the interval (t_ℓ, t_{r+d+1}). Due to the local supports of B-splines, the spline ψ is orthogonal to V_0 if and only if

$$\langle \phi_i, \psi \rangle = 0, \quad \text{for } i = k - d, \ldots, k + p,$$

or in terms of the Gram matrix

$$\left\langle \boldsymbol{\phi}, \boldsymbol{\gamma}^\top \right\rangle \boldsymbol{w} = 0.$$

In order to obtain a wavelet ψ with the smallest possible support, we want $r - l$ as small as possible. This motivates the following definition:

Definition 1 (Minimal Interval). *The index interval $[\ell : r]$ is said to be minimal if*

$$r - \ell > p(\ell, r) + d, \tag{1}$$

and there is no true subinterval $[u : v]$ with this property, i.e., if $\ell \leq u < v \leq r$ and

$$v - u < p(u, v) + d, \tag{2}$$

then $u = \ell$ and $v = r$.

The left multiplicity $\lambda_t(i)$ (right multiplicity $\rho_t(i)$) of a knot t_i gives the number of knots in t equal to t_i, but with index less or equal (greater or equal) than i,

$$\lambda_t(i) = \max\{j : t_{i-j+1} = t_i\}, \quad \rho_t(i) = \max\{j : t_{i+j-1} = t_i\}.$$

If s_j is a new knot and $t_{\tilde{\ell}}$ and $t_{\tilde{r}}$ are two knots with $t_{\tilde{\ell}} < s_j < t_{\tilde{r}}$, then we define the integer functions μ_j and ν_j by

$$\begin{aligned} \mu_j(\tilde{\ell}) &= \#\left\{i < j : s_i \in \left(t_{\tilde{\ell}}, s_j\right]\right\}, \\ \nu_j(\tilde{r}) &= \#\left\{i < j : s_i \in [s_j, t_{\tilde{r}})\right\}. \end{aligned}$$

We see that if $[\ell : r]$ is a minimal interval and s_j is a new knot in (t_ℓ, t_{r+d+1}) then the relation

$$\rho_{\boldsymbol{t}}(\ell) + \mu_j(\ell) + \nu_j(r + d + 1) + \lambda_{\boldsymbol{t}}(r + d + 1) = 2d + 2$$

must hold. The following proposition gives a construction of a minimal interval.

Proposition 1. *Let s_j be a new knot, let ℓ be the largest integer less than j such that*

$$\mu_j(\ell) + \rho_t(\ell) = d + 1,$$

and let r be the smallest integer greater than j such that

$$\nu_j(r + d + 1) + \lambda_t(r + d + 1) = d + 1.$$

Then $[\ell : r]$ is a minimal interval.

Then we can compute B-wavelets as follows.

Lemma 1. *Let $[\ell_j : r_j]$ be the index support of the j th B-wavelet ψ_j^*, scaled such that the absolute values of its B-spline coefficients add to one. Let $p_j = p(\ell_j, r_j)$ be the number of old knots in $(t_{\ell_j}, t_{r_j+d+1})$. The nonzero B-spline coefficients $(q_{i,j})_{i=\ell_j}^{r_j}$ of ψ_j^* are given by the solution of the linear system*

$$\begin{pmatrix} \langle \phi_{k-d}, \gamma_{\ell_j} \rangle & \langle \phi_{k-d}, \gamma_{\ell_j+1} \rangle & \cdots & \langle \phi_{k-d}, \gamma_{r_j} \rangle \\ \vdots & \vdots & & \vdots \\ \langle \phi_{k+p_j}, \gamma_{\ell_j} \rangle & \langle \phi_{k+p_j}, \gamma_{\ell_j+1} \rangle & \cdots & \langle \phi_{k+p_j}, \gamma_{r_j} \rangle \\ 1 & -1 & \cdots & (-1)^{r_j-\ell_j} \end{pmatrix} \begin{pmatrix} q_{\ell_j,j} \\ \vdots \\ q_{r_j-1,j} \\ q_{r_j,j} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix},$$

and constitute the nonzero part of column j of $\boldsymbol{Q}$, where $\boldsymbol{\gamma}^\top \boldsymbol{Q} = \psi^\top$.

3 Decimated Hypergraph Framelets with Spline Wavelet Basis

This section elaborates on the construction of decimated hypergraph framelets, with a central focus on adapting framelet theory to hypergraphs.

Let $\mathcal{H}_{J\to J_0} = (\mathcal{H}_J, \mathcal{H}_{J-1}, \ldots, \mathcal{H}_{J_0})$ be a coarse-grained chain of the hypergraphs $\mathcal{H} = (V, \mathcal{E}, \boldsymbol{w}_V, \boldsymbol{w}_\mathcal{E})$ and $\mathcal{H}_j = (V_j, \mathcal{E}_j, \boldsymbol{w}_{V_j}, \boldsymbol{w}_{\mathcal{E}_j})$. For each layer $\mathcal{H}_j$ in $\mathcal{H}_{J\to J_0}$, we map its vertex set V_j to the partition of the interval, where he length of the interval assigned to a coarse-grained node is proportional to the number of vertices in V_J contained in it, to obtain basic knot vector $\boldsymbol{t}'_j$. So each V_j associates with a knots vector $\boldsymbol{t}'_j$ and $\boldsymbol{t}_j := \bigcup_{i=J_0}^{j} \boldsymbol{t}'_i$. For a hypergraph signal g, we define a map $\mathbb{R}^n \to S_{d,\boldsymbol{t}_J}$ by $g \mapsto f = \sum_{i=1}^{n} f(i) B_{i,d,\boldsymbol{t}}$, where $B_{i,d,\boldsymbol{t}}$ is B-spline locating at the i-th knot.

With the above definition, we establish a direct mapping between the coarse-grained chain and spline knot vectors to enable the extension of the spline wavelet on the hypergraph: For each layer $\mathcal{H}_j$ in the coarse-grained chain $\mathcal{H}_{J\to J_0}$, we map its vertex set V_j to the partition of the interval $[0, 1]$ to obtain the basic knot vector $\boldsymbol{t}'_j$. The cumulative knot vector for layer j is $\boldsymbol{t}_j = \bigcup_{i=J_0}^{j} \boldsymbol{t}'_i$, where $\boldsymbol{t}_{J_0}$ (coarsest layer $\mathcal{H}_{J_0}$) serves as the initial knot vector $\boldsymbol{\tau}$. For finer layers ($j > J_0$), $\boldsymbol{t}_j$ is obtained by inserting new knots $\boldsymbol{s}_j = \boldsymbol{t}'_j \setminus \boldsymbol{t}'_{j-1}$ into $\boldsymbol{t}_{j-1}$. With the above definition, any signal $g_j \in \mathbb{R}^{|V_j|}$ on $\mathcal{H}_j$ is embedded into the interval with knot vector $\boldsymbol{t}_j$.

3.1 Coarsening of a Hypergraph

In this paper, we adopt a method similar to that in [22] to construct a coarse-grained chain using a clustering algorithm, the Coarse-Grained Chain for Hypergraphs (CGC-HG) algorithm. The clustering algorithm used determines the coarse-grained chain. Algorithm 1 shows an implementation of the algorithm, which utilizes a modified version of the non-spectral hierarchical clustering (NHC) algorithm given in [3].

Papers [19] and [23] show that hypergraph partition is equivalent to graph partition with a certain weight. Define a weighted edge on the same vertex set V as:

$$w^*(u,v) = \sum_{e:\{u,v\}\subseteq e} \frac{w(e)}{\delta(e)}, \quad (u \neq v).$$

Algorithm 1 explicitly implements this weight to construct a corresponding hypergraph $\mathcal{H}$. Its adjacency matrix is defined as $A_j = H_j \cdot \mathrm{diag}(w_j(e)/\delta_j(e)) \cdot H_j^\top$ (with $\mathrm{diag}(A_j) = 0$). And The graph distance $\rho_{\mathcal{G}_j}(u,v)$ on $\mathcal{G}_j$ is the shortest weighted path length between u and v, with edge length $l(e) = 1/w_j(a,b)$ for $e = (a,b)$:

$$\rho_{\mathcal{G}_j}(u,v) = \min\left\{ \sum_{e\in P} \frac{1}{w_j(a,b)} \,\middle|\, P \text{ is a path from } u \text{ to } v \right\}.$$

3.2 Orthonormal Spline Wavelets Basis for a Coarse-grained Chain

In Section 3.1, we have proposed a method to map coarse-grained chains to knot vectors. In this subsection, we construct the orthonormal B-spline wavelet basis for the coarse-grained chain $\mathcal{H}_{J\to J_0} = (\mathcal{H}_J, \mathcal{H}_{J-1}, \ldots, \mathcal{H}_{J_0})$ via a coarse-to-fine paradigm, bridging the coarse-grained chain with the non-uniform spline wavelet theory [15]. We construct the B-spline wavelet basis as follows.

1) B-spline Scaling Basis for Coarsest Layer $\mathcal{H}_{J_0}$. Corresponds to the initial knot vector $\boldsymbol{\tau} = \boldsymbol{t}_{J_0}$ with the B-spline space $V_0 = S_{d,\boldsymbol{\tau}}$.
2) Recursive Wavelet Basis for Finer Layers ($J_0 < j \leq J$) For each finer layer $\mathcal{H}_j$, the B-spline space decomposes as $V_j = V_{j-1} \oplus W_j$, where V_{j-1} is the scaled space of $\mathcal{H}_{j-1}$ and W_j is the wavelet space. For each new knot $s \in \boldsymbol{s}_j$, compute the minimal interval $[\ell(s) : r(s)]$ via Proposition 1 to obtain the support index of each wavelet. Then, the construct raw wavelet can be computed by solving Lemma 1's linear system [15], and the raw scaling function can be computed by Oslo's algorithm [4, 16]. Finally, we orthogonalize the scaling functions and the spline wavelets to get an orthonormal spline wavelet basis.

3.3 Decimated Hypergraph Framelets

We extend the decimated framelets system to hypergraphs, leveraging the coarse-grained hierarchy to define decimated hypergraph framelets.

Algorithm 1 CGC-HG: Coarse-Grained Chain for Hypergraphs

Input: Hypergraph $\mathcal{H} = (V, \mathcal{E}, \boldsymbol{w}_V, \boldsymbol{w}_\mathcal{E})$; maximum cluster size s_{max}; coarsening ratio $r \in (0, 1)$.
Output: A coarse-grained chain $(\mathcal{H}_{J-1}, \ldots, \mathcal{H}_{J_0})$ of $\mathcal{H}$ with $\mathcal{H}_j = (V_j, \mathcal{E}_j, \boldsymbol{w}_{V_j}, \boldsymbol{w}_{\mathcal{E}_j})$.
Initialization: $j \leftarrow 0$.
while $j > J_0$ **do**
    Step 1: Compute incidence matrix H_j of $\mathcal{H}_j$.
    Compute induced affinity (clique form) for $\mathcal{H}_j$:

$$A_j = H_j \operatorname{diag}\big(w_j(e)/\delta_j(e)\big)\, H_j^\top, \quad \operatorname{diag}(A_j) = 0$$

    where $\delta_j(e) = \sum_v H_j(v, e)$ is the cardinality of hyperedge e.
    Step 2: CGC-style clustering
    Obtain the hypergraph $\mathcal{H}_j$ form adjacent matrix A_j
    Compute the hypergraph distance $\rho_{\mathcal{H}_j}$ of $\mathcal{H}_j$
    $K \leftarrow \max\{1, \lceil r|V_j|\rceil\}$.
    while $\exists \ell \in \{1, \ldots, K\}, |c_\ell| > s_{\max}$ **do**
        $K \leftarrow 2K$
        Randomly choose K vertices $u_1, \ldots, u_K$ from V_j as centers.
        while true **do**
            Construct cluster c_ℓ for $\ell = 1, \ldots, K$ such that $v \in V_j$ belongs to c_ℓ and
            the distance between v and the cluster center u_ℓ is the minimal among
            all clusters:

$$\ell = \arg \min_{1 \le \ell' \le K} \rho_{\mathcal{H}_j}(u_{\ell'}, v).$$

            Update the centers: for each c_ℓ, find a new center $u_\ell \in c_\ell$ such that
            $\sum_{v \in C_\ell} \rho_{\mathcal{H}_j}(u_\ell, v)$ is minimized.
            Break if all centers remain the same.
        end while
    end while
    Step 3: Contract $\mathcal{H}_j$ to obtain $\mathcal{H}_{j+1}$
    Let $V_{j+1} = \{c_1, \ldots, c_K\}$ be the set of coarse vertices.
    Let $\pi : V_j \to V_{j+1}$ be the clustering map, i.e., $\pi(v) = c_\ell \iff v \in c_\ell$.
    For each $e \in \mathcal{E}_j$, define its projection $\Phi_\pi(e) := \{\pi(v) : v \in e\} \subseteq V_{j+1}$.
    Set the coarse hyperedge set $\mathcal{E}_{j+1} := \{\Phi_\pi(e) : e \in \mathcal{E}_j,\ |\Phi_\pi(e)| \ge 2\}$.
    For each $e' \in \mathcal{E}_{j+1}$, compute the coarse hyperedge weight as

$$w_{e,j+1}(e') := \sum_{e \in \mathcal{E}_j:\ \Phi_\pi(e) = e'} w_{e,j}(e).$$

    For each $c_\ell \in V_{j+1}$, compute the coarse vertex weight $w_{V_{j+1}}(c_\ell) := \sum_{v \in C_\ell} w_{V_j}(v)$.
    return $\mathcal{H}_{j+1} = (V_{j+1}, \mathcal{E}_{j+1}, \boldsymbol{w}_{v,j+1}, \boldsymbol{w}_{e,j+1})$.
    $j \leftarrow j - 1$.
end while

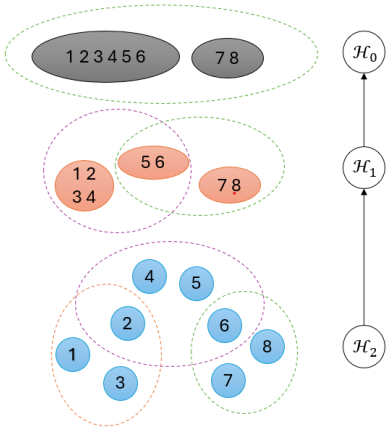

Fig. 1. Coarse-grained chain $\mathcal{H}_{2\to 0} := \mathcal{H}_2 \to \mathcal{H}_1 \to \mathcal{H}_0$ of $\mathcal{H} \equiv \mathcal{H}_3$. Here the bottom hypergraph $\mathcal{H}_3 \equiv \mathcal{H}$ is the underlying hypergraph. Each solid oval in $\mathcal{H}_j = (V_j, \mathcal{E}_j, \boldsymbol{w}_{V_j}, \boldsymbol{w}_{\mathcal{E}_j})$ represents a vertex of $\mathcal{H}_j$, and the dashed ellipses represent hyperedges (with weights omitted for simplicity). Then the vertices of $\mathcal{H}_0$ can be mapped to intervals $[0, \frac{3}{4}]$ and $[\frac{3}{4}, 1]$; the vertices of $\mathcal{H}_1$ can be mapped to intervals $[0, \frac{1}{2}]$, $[\frac{1}{2}, \frac{3}{4}]$ and $[\frac{3}{4}, 1]$; the vertices of $\mathcal{H}_2$ can be mapped to intervals $[0, \frac{1}{8}]$, $[\frac{1}{8}, \frac{1}{4}]$, $[\frac{1}{4}, \frac{3}{8}]$, $[\frac{3}{8}, \frac{1}{2}]$, $[\frac{1}{2}, \frac{5}{8}]$, $[\frac{5}{8}, \frac{3}{4}]$, $[\frac{3}{4}, \frac{7}{8}]$ and $[\frac{7}{8}, 1]$.

Decimated Hypergraph Framelets The definition of decimated hypergraph framelets is given as follows. Let $\mathcal{H}_{J\to J_0} = (\mathcal{H}_J, \mathcal{H}_{J-1}, \ldots, \mathcal{H}_{J_0})$ be a coarse-grained chain of $\mathcal{H}$, where each $\mathcal{H} = (V, \mathcal{E}, \boldsymbol{w}_v, \boldsymbol{w}_e)$ is a hypergraph. $\{(u_\ell, \lambda_\ell)\}_{\ell=1}^N$ is a set of pairs of orthogonal spline wavelet bases constructed in section 3.2 and complex numbers on the interval $[0, 1]$, and Λ_j are scaling factors adapted to the hypergraph scale j.

Let $\Psi_j = \{\alpha_j; \beta_j^{(1)}, \ldots, \beta_j^{(r_j)}\}$ be a set of functions in $L_1(\mathbb{R})$ at scale j for $j = J_0, \ldots, J$. We link the framelet generators in Ψ_j and Ψ_{j-1} by a filter bank $\eta_j = \{a_j; b_j^{(1)}, \ldots, b_j^{(r_{j-1})}\}$ so that, for $\xi \in \mathbb{R}$ and $0 < \Lambda_{J_0} \le \Lambda_{J_0+1} \le \cdots \le \Lambda_J < \infty$,

$$\widehat{\alpha}_{j-1}(\xi/\Lambda_{j-1}) = \widehat{a}_j(\xi/\Lambda_j)\widehat{\alpha}_j(\xi/\Lambda_j), \tag{3}$$

$$\widehat{\beta}_{j-1}^{(n)}(\xi/\Lambda_{j-1}) = \widehat{b}_j^{(n)}(\xi/\Lambda_j)\widehat{\alpha}_j(\xi/\Lambda_j), \quad n = 1, \ldots, r_{j-1}, \tag{4}$$

where $\Lambda_{J_0}, \Lambda_{J_0+1}, \ldots, \Lambda_J$ are called scaling factors.

Definition 2. (Decimated hypergraph framelets) *The decimated framelets* $\varphi_{j,y}(x)$ *and* $\psi_{j,y}^n(x), y, x \in [0, 1]$, *at scale* $j = J_0, \ldots, J$, *for the chain* $\mathcal{H}_{J\to J_0}$ *of*

the hypergraph $\mathcal{H}$, are defined by

$$\begin{aligned}\varphi_{j,y}(x) &:= \sum_{\ell=1}^{N} \widehat{\alpha_j}\left(\frac{\lambda_\ell}{\Lambda_j}\right)\overline{u_\ell(y)}u_\ell(x), \quad y \in [0,1],\\ \psi^n_{j,y}(x) &:= \sum_{\ell=1}^{N} \widehat{\beta^{(n)}_j}\left(\frac{\lambda_\ell}{\Lambda_j}\right)\overline{u_\ell(y)}u_\ell(x), \quad y \in [0,1], n = 1,\dots,r_j.\end{aligned} \tag{5}$$

We call $\varphi_{j,y}$ and $\psi^n_{j,y}$ scaling function and framelets at scale j, respectively.

Definition 3. (Decimated hypergraph framelet system) *The decimated framelet system* $\mathrm{DFS}(\{\Psi_j\}_{j=J_1}^{J}, \{\eta_j\}_{j=J_1+1}^{J})$ *on $\mathcal{H}$, starting from a scale J_1, is a non-homogeneous, nonstationary affine system given by*

$$\begin{aligned}&\mathrm{DFS}\left(\{\Psi_j\}_{j=J_1}^{J}, \{\eta_j\}_{j=J_1+1}^{J}\right) := \mathrm{DFS}\left(\{\Psi_j\}_{j=J_1}^{J}, \{\eta_j\}_{j=J_1+1}^{J}; \mathcal{H}_{J\to J_1}\right)\\ &:= \{\varphi_{J_1,y} : y \in [0,1]\} \cup \left\{\psi^n_{j,y} : y \in [0,1], j = j_2,\dots,J\right\}.\end{aligned} \tag{6}$$

Theoretically, when $\{u_\ell \mid \ell = 1,\dots,N\}$ is an orthogonal basis in $L_2([0,1])$, the decimated hypergraph framelets $\mathrm{DFS}\left(\{\Psi_j\}_{j=J_1}^{J}, \{\boldsymbol{\eta}_j\}_{j=J_1+1}^{J}\right)$ can be constructed to be a tight frame [22].

Hypergraph Filter Banks Here, we define the filter bank using polynomial splines as in [9]. Let $P_m(x)$ be a polynomial given by

$$P_m(x) := \left(\frac{1+x}{2}\right)^m \sum_{k=0}^{m-1}\binom{m-1+k}{k}\left(\frac{1-x}{2}\right)^k,$$

which satisfies $P_m(x) + P_m(-x) = 1$ for any $m \in \mathbb{N}$ [5]. We define a bump function $\nu_{[c_L,c_R];\varepsilon_L,\varepsilon_R}(\xi)$ for $c_L < c_R$ and positive numbers $\varepsilon_L, \varepsilon_R$ satisfying $\varepsilon_L + \varepsilon_R \le c_R - c_L$ by

$$\nu_{[c_L,c_R];\varepsilon_L,\varepsilon_R}(\xi) := \begin{cases} 0, & \xi \le c_L - \varepsilon_L \text{ or } \xi \ge c_R + \varepsilon_R\\ \sin\left(\frac{\pi}{2}P_m\left(\frac{\xi - c_L + \varepsilon_L}{2\varepsilon_L}\right)\right), & c_L - \varepsilon_L \langle \xi \langle c_L + \varepsilon_L\\ 1, & c_L + \varepsilon_L \le \xi \le c_R - \varepsilon_R\\ \cos\left(\frac{\pi}{2}P_m\left(\frac{\xi - c_R + \varepsilon_R}{2\varepsilon_R}\right)\right), & c_R - \varepsilon_R \langle \xi \le c_R + \varepsilon_R \end{cases}.$$

The scaling functions and filters for the decimated framelets are defined by $\nu_{[c_L,c_R];\varepsilon_L,\varepsilon_R}$ and the coarse-grained chain of the hypergraph. Let $0 < \varepsilon < 1$ and define α_ε by

$$\widehat{\alpha_\varepsilon}(\xi) := \nu_{\left[-\frac{1+\varepsilon}{2},\frac{1+\varepsilon}{2}\right];\frac{1-\varepsilon}{2},\frac{1-\varepsilon}{2}}(\xi), \quad \xi \in \mathbb{R}.$$

We then define the scaling functions for scale j by, for $n = 1,\dots,r_j$,

$$\widehat{\beta^{(n)}_j}(\xi/\Lambda_j) := \sqrt{\left|\widehat{\alpha_\varepsilon}(\xi/\Lambda_{j+1})\right|^2 - \left|\widehat{\alpha_\varepsilon}(\xi/\Lambda_j)\right|^2}\cdot\gamma^{(n)}(\xi/\Lambda_{j+1}).$$

The associated framelet generating set is then

$$\Psi_j := \left\{\alpha_\varepsilon; \beta_j^{(1)}, \ldots, \beta_j^{(r_j)}\right\}, \quad j = J_0, \ldots, J.$$

From this, we define the filter bank as follows.

Definition 4. (Filter bank for the coarse-grained chain) *For $j = J_0 + 1, \ldots, J$, the filter bank for the chain $\mathcal{H}_{J\to J_0}$ of hypergraph $\mathcal{H}$ is*

$$\boldsymbol{\eta}_j := \left\{a_j; b_j^{(1)}, \ldots, b_j^{(r_{j-1})}\right\}$$

with

$$\widehat{a_j}\left(\xi/\Lambda_j\right) := \frac{\widehat{\alpha_\varepsilon}\left(\xi/\Lambda_{j-1}\right)}{\widehat{\alpha_\varepsilon}\left(\xi/\Lambda_j\right)}, \widehat{b_j^{(n)}}\left(\xi/\Lambda_j\right) := \frac{\widehat{\beta_j^{(n)}}\left(\xi/\Lambda_{j-1}\right)}{\widehat{\alpha_\varepsilon}\left(\xi/\Lambda_j\right)} \tag{7}$$

for $\xi \in \mathbb{R}, n = 1, \ldots, r_{j-1}$.

3.4 Fast Fourier Transforms and Fast Framelets Transforms

The previous subsection establishes the theoretical foundation of the orthonormal spline wavelet basis for hypergraph coarse-grained chains. To bridge the theoretical construction with practical hypergraph learning tasks, this subsection presents efficient computational algorithms, including fast Fourier transforms for hypergraph signals and fast framelet transforms for decimated hypergraph framelet systems.

Fast Fourier Transform with Spline Wavelets We first define the Fourier transforms. For a finite index set Ω, we denote by $l_2(\Omega) := \{\boldsymbol{c} : \Omega \to \mathbb{C}\}$ all complex-valued sequences supported on Ω. For $j = J_0, \ldots, J$, let

$$\Omega_j := \{\ell : 1 \le \ell \le N_j\},$$

where $N_j := |V_j|$, and $l_2(\Omega_j)$ and $l_2(V_j)$ are the sequences supported on Ω_j and V_j respectively.

Definition 5. (Fourier transform) *The Fourier transform $\mathbf{F}_j^* : L_2([0,1]) \to l_2\left(\Omega_j\right)$ on $\mathcal{H}_j$ is*

$$\left(\mathbf{F}_j^*\mathbf{v}\right)_\ell := \int_{[0,1]} \mathbf{v}(y)\overline{\boldsymbol{u}_\ell(y)}dy, \quad \mathbf{v} \in L_2([0,1]), \ell \in \Omega_j.$$

We say the sequence $\mathbf{F}_j^*\mathbf{v}$ a $(L_2([0,1]), \Omega_j)$-sequence and let $l_2(L_2([0,1]), \Omega_j)$ be the set of all $(L_2([0,1]), \Omega_j)$-sequences.

Definition 6. (Adjoint Fourier transform) *Define $\mathbf{F}_j : l_2\left(\Omega_j\right) \to L_2([0,1])$ the adjoint Fourier transform on $\mathcal{H}_j$ by*

$$\left[\mathbf{F}_j\boldsymbol{c}\right](y) := \sum_{\ell\in\Omega_j} c_\ell \boldsymbol{u}_\ell(y), \quad y \in [0,1], \boldsymbol{c} = (c_\ell)_{\ell=1}^{N_j} \in l_2\left(\Omega_j\right).$$

We say the function $[\mathbf{F}_j\boldsymbol{c}]$ a $(\Omega_j, (L_2([0,1]))$-function and let $L_2(\Omega_j, (L_2([0,1]))$ be the set of all $(\Omega_j, (L_2([0,1]))$-functions.

Then we can apply the spline wavelets theory in [15, Section 4] to achieve the fast Fourier transforms. The computational cost is $\mathcal{O}(N \log N)$.

Fast Framelets Transform In this subsection, we propose decimated framelet transforms on a coarse-grained chain $\mathcal{H}_{J\to J_0}$, including the decomposition and reconstruction algorithms. For a signal $\boldsymbol{f}$ on the interval $[0,1]$, which is mapped from the original hypergraph signal $\boldsymbol{g}$ on the coarse-grained chain $\mathcal{H}_{J\to J_0}$, and a decimated hypergraph framelet system $\mathrm{DFS}\left(\{\Psi_j\}_{j=J_1}^{J}, \{\boldsymbol{\eta}_j\}_{j=J_1+1}^{J}; \mathcal{H}_{J\to J_0}\right)$, $J_1 = J_0, \ldots, J$, the framelet decomposition algorithm produces a sequence of the vectors as the framelet approximation and detail coefficients

$$\left\{\mathbf{v}_{J_0}\right\} \cup \left\{\mathbf{w}_j^n : n = 1, \ldots, r_j, j = J_0, \ldots, J-1\right\}, \tag{8}$$

where for level $j = J_0, \ldots, J$, $\mathbf{v}_j$ is the vector of the approximation framelet coefficients for $\mathcal{H}_j$, and $\mathbf{w}_j^n, n = 1, \ldots, r_j$, is the vector of the detail framelet coefficients for $\mathcal{H}_{j+1}$:

$$\begin{aligned} \mathbf{v}_j(y) &:= \left\langle \boldsymbol{f}, \boldsymbol{\varphi}_{j,y} \right\rangle, y \in [0,1], \\ \mathbf{w}_j^n(y) &:= \left\langle \boldsymbol{f}, \boldsymbol{\psi}_{j,y}^n \right\rangle, y \in [0,1]. \end{aligned} \tag{9}$$

Based on the discrete Fourier transform operators, we can define convolution, downsampling, and upsampling operators. Let $h \in L_2([0,1])$ be a mask and $\mathbf{v} \in L_2(\Omega_j, L_2([0,1]))$ be a $(\Omega_j, L_2([0,1]))$-function. Let $\hat{\mathbf{v}} := (\hat{v}_\ell)_{\ell\in\Omega_j}$ be its discrete Fourier coefficient sequence.

Definition 7. (Discrete convolution) *The discrete convolution $\mathbf{v} *_j h$ is defined as the following sequence:*

$$\left[\mathbf{v} *_j h\right](y) := \sum_{\ell\in\Omega_j} \widehat{\mathbf{v}}_\ell \widehat{h}\left(\frac{\lambda_\ell}{\Lambda_j}\right) \boldsymbol{u}_\ell(y), \quad y \in [0,1]. \tag{10}$$

That is, in the Fourier domain $\left(\widehat{\mathbf{v} *_j h}\right)_\ell = \widehat{\mathbf{v}}_\ell \widehat{h}\left(\frac{\lambda_\ell}{2^j}\right)$ for $\ell \in \Omega_j$.

Definition 8. (Downsampling) *We define the downsampling operator*

$$\downarrow_j \colon l_2\left(\Omega_j, V_j\right) \to l_2\left(\Omega_{j-1}, V_{j-1}\right)$$

for a (Ω_j, V_j)-sequence $\mathbf{v}$ by

$$\left[\mathbf{v} \downarrow_j\right](y) := \sum_{\ell\in\Omega_j} \widehat{\mathbf{v}}_\ell \boldsymbol{u}_\ell(y), \quad y \in [0,1]. \tag{11}$$

Definition 9. (Upsampling) *The upsampling operator*

$$\uparrow_j \colon l_2\left(\Omega_{j-1}, V_{j-1}\right) \to l_2\left(\Omega_j, V_j\right)$$

for a sequence $\mathbf{v} \in l_2\left(\Omega_{j-1}, V_{j-1}\right)$ is defined by

$$\left[\mathbf{v} \uparrow_j\right](y) := \sum_{\ell\in\Omega_{j-1}} \widehat{\mathbf{v}}_\ell \boldsymbol{u}_\ell(y), \quad y \in [0,1]. \tag{12}$$

Then, we can obtain a fast computation with the following relation between the decimated hypergraph framelets transforms and discrete Fourier transforms on $\mathcal{H}$. The proof is similar to [22].

Proposition 2. *For $j = J_0 + 1, \ldots, J$, the decimated hypergraph framelet decomposition and reconstruction at level j can be written as*

$$\mathbf{v}_{j-1} = \mathbf{F}_{j-1}\left(\mathbf{v}_j \ast_j \widehat{(a_j)^\star}\right), \quad \mathbf{w}_{j-1}^n = \mathbf{F}_j\left(\mathbf{v}_j \ast_j \widehat{(b_j^{(n)})^\star}\right), \quad n = 1, \ldots, r_{j-1},$$

and

$$\mathbf{v}_j = \left(\mathbf{F}_j^*\left(\mathbf{v}_{j-1}\right)\right) \ast_j a_j + \sum_{n=1}^{r_{j-1}} \left(\mathbf{F}_j^*\left(\mathbf{w}_{j-1}^n\right)\right) \ast_j b_j^{(n)}.$$

Algorithms 2 and 3 below give a pseudocode for the decomposition and reconstruction of decimated hypergraph transforms based on the formula in Proposition 2. Using the Spline Wavelet basis for the chain(see Section 3.1), the Fast Fourier transforms for the input data with size N have the computational cost $\mathcal{O}(N \log N)$. This, along with Algorithms 3 and 4, then shows that the computational cost of decimated hypergraph transforms is $\mathcal{O}(N \log N)$.

Algorithm 2 Decomposition for Decimated Hypergraph Framelet Transform

Input:: $\mathbf{v}_J$, which is a (Λ_J, Ω_J)-sequence; filter bank
Output:: $\left(\widehat{\mathrm{w}}_{J-1}^1, \ldots, \widehat{\mathrm{w}}_{J-1}^{r_{J-1}}, \ldots, \widehat{\mathrm{w}}_{J_0}^1, \ldots, \widehat{\mathrm{w}}_{J_0}^{r_{J_0}}, \widehat{\mathrm{v}}_{J_0}\right)\ \mathbf{v}_J \longrightarrow \widehat{\mathrm{v}}_J$
for $j \leftarrow J$ to $J_0 + 1$ do
  $\widehat{\mathrm{v}}_{j-1} \longleftarrow \widehat{\mathrm{v}}_j \widehat{a_j}\,(\lambda./\Lambda_j)$ // discrete convolution at level j
  // downsample from level j to $j-1$
  for $n \leftarrow 1$ to r_{j-1} do
    $\widehat{\mathrm{w}}_{j-1}^n \longleftarrow \widehat{\mathrm{v}}_j \widehat{\left(b_j^{(n)}\right)}\,(\lambda \cdot /\Lambda_j)$ // discrete convolution at level j
    $\mathbf{w}_{j-1}^n \longleftarrow \widehat{\mathbf{w}}_{j-1}^n$
  end
  $\mathbf{v}_{J_0} \longleftarrow \widehat{\mathrm{v}}_{J_0}$
end

4 Hypergraph Framelet Neural Networks

In this section, we introduce the neural network model that integrates our constructed Spline-Wavelet Based Decimated Hypergraph Framelets, which we call SWDHFN.

The input to the model includes four key components: the node feature matrix $\boldsymbol{X} \in \mathbb{R}^{N \times n_f}$ (where N denotes the number of nodes and n_f is the number of features), the label matrix $\boldsymbol{Y} \in \mathbb{R}^{N \times M}$(totally M categories), the normalized

Algorithm 3 Reconstruction for Decimated Hypergraph Framelet Transform

Input: $\left(\widehat{\mathrm{w}}_{J-1}^{1},\ldots,\widehat{\mathrm{w}}_{J-1}^{r_{J-1}},\ldots,\widehat{\mathrm{w}}_{J_0}^{1},\ldots,\widehat{\mathrm{w}}_{J_0}^{r_{J_0}},\widehat{\mathrm{v}}_{J_0}\right)$
Output: $\mathbf{v}_J\ \widehat{\mathrm{v}}_{J_0} \longleftarrow \mathbf{v}_{J_0}$
for $j \leftarrow J_0+1$ to J do
  for $n \leftarrow 1$ to r_{j-1} do
    $\widehat{\mathbf{w}}_{j-1}^{n} \longleftarrow \mathbf{w}_{j-1}^{n}$
  end
  $\widehat{\mathrm{v}}_j \longleftarrow (\widehat{\mathrm{v}}_{j-1,\cdot})\,\widehat{a}_j\,(\lambda\cdot/\Lambda_j) + \sum_{n=1}^{r_{j-1}} \widehat{\mathrm{w}}_{j-1,\cdot}^{n}\,\widehat{\left(b_j^{(n)}\right)}\,(\lambda\cdot/\Lambda_j)$
end

hypergraph adjacency matrix $\tilde{\boldsymbol{A}}$, and the scaling function matrix $\boldsymbol{\Phi} \in \mathbb{R}^{N\times N}$ along with the framelet matrix $\boldsymbol{\Psi} \in \mathbb{R}^{N\times N}$.

Suppose there are n channels, associating a series of matrices $\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_n$. For each channel, $\boldsymbol{X}_i \in \mathbb{R}^{N\times d_i}, i \in 1,\ldots,n$. Our model is a two layer neural network, which defined as:

$$\boldsymbol{H}_1 = \|_{i=1}^{n} \alpha_i \cdot \mathfrak{N}\left(\boldsymbol{X}_i \boldsymbol{W}_i\right), \tag{13}$$

$$\hat{\boldsymbol{Y}} = \operatorname{softmax}\left(FFN(\operatorname{ReLU}\left(\boldsymbol{H}_1\right))\boldsymbol{W} + \boldsymbol{b}\right), \tag{14}$$

where $\|$ denotes the concatenation operation, α_i are trainable attention weights satisfying $\alpha_i \in (0,1)$ and $\sum_i \alpha_i = 1, \mathfrak{N}(\cdot)$ is the row normalization operation, and $\boldsymbol{W}_i \in \mathbb{R}^{d_i\times n_h}$ and $\boldsymbol{W} \in \mathbb{R}^{n\times n_h\times M}$ are trainable parameters. Our model comprises several input channels at the beginning and, subsequently, several fully connected layers. Therefore, it is easy to extend with more layers. As usual, we minimize the cross entropy of the labeled nodes using the first l columns of $\hat{\boldsymbol{Y}}$ and $\boldsymbol{Y}$.

Given the decimated hypergraph framelet system $\mathrm{DFS}(\{\Psi_j\}_{j=J_1}^{J}, \{\eta_j\}_{j=J_1+1}^{J})$. We denote $\boldsymbol{\Phi}$, $\boldsymbol{\Psi}$ be the scaling function matrix and framelets matrix. We denote $\boldsymbol{F}_0(\boldsymbol{M}) := \boldsymbol{\Phi}\boldsymbol{M}$ and $\boldsymbol{F}_1(\boldsymbol{M}) := \boldsymbol{\Psi}\boldsymbol{M}$. For our model, we select $n = 1 + r + 2, \left\{\boldsymbol{X}, \tilde{\boldsymbol{A}}\boldsymbol{X}, \tilde{\boldsymbol{A}}^2\boldsymbol{X}, \ldots, \tilde{\boldsymbol{A}}^r\boldsymbol{X},\ \boldsymbol{F}_0(\tilde{\boldsymbol{A}}\boldsymbol{X}), \boldsymbol{F}_1(\tilde{\boldsymbol{A}}\boldsymbol{X})\right\}$.

Remark: In contrast to our model, the model FSGNN[17] adopts the two-layer network model with the following three options of input channels, where A is the adjacency matrix of graph:

1. Homophily: $n = 1 + r, \{X, AX, A^2X, \ldots, A^rX\}$.
2. Heterophily: $n = 1 + r, \{X, AX, A^2X, \ldots, A^rX\}$.
3. All: $n = 1 + 2r, \{X, AX, A^2X, A^rX, A^2X, \ldots, A^rX, A^rX, A^rX\}$.

The computational complexity of SWDHFN is primarily determined by three key operations: multi-channel feature construction, attention-weighted linear transformations, and feed-forward classification. The multi-order neighborhood features $\tilde{\boldsymbol{A}}^k X$ are computed via sparse matrix multiplication with complexity $O(r \cdot Ekn_f)$, where E is the number of hyperedges, k is the average hyperedge size, r is the neighborhood order, and n_f is the feature dimension.

Leveraging the compact support and sparsity of B-spline wavelet bases, the fast framelet transforms $F_0(\tilde{\boldsymbol{A}}\boldsymbol{X}) = \boldsymbol{\Phi}\tilde{\boldsymbol{A}}\boldsymbol{X}$ and $F_1(\tilde{\boldsymbol{A}}X) = \boldsymbol{\Psi}\tilde{\boldsymbol{A}}\boldsymbol{X}$ are performed in $O(N \log N \cdot n_f)$ time. The two-layer neural network further processes these features with linear transformations, resulting in an overall complexity of $O(rEkn_f + N \log N \cdot n_f + (r+3)Nn_f n_h + (r+3)Nn_h M)$, where n_h is the hidden dimension and M is the number of classes.

5 Experiments

5.1 Datasets and Experimental Settings

We evaluate all models on 5 benchmark datasets covering both homophilic and heterophilic hypergraphs, with detailed statistics shown in Table 1. Among these datasets, four are well-established traditional benchmarks (Cora, Citeseer, Senate, House) widely used in hypergraph learning research, while Actor is a newly proposed heterophilic hypergraph dataset from recent work [13].

Table 1. Overview of key statistics for benchmark datasets.

Datasets	Cora	Citeseer	Actor	Senate	House
Hypernodes, $\lvert V\rvert$	2,708	3,312	16,255	282	1,290
Hyperedges, $\lvert E\rvert$	1,579	1,079	10,164	315	340
Avg. hyperedge size	3.0 ± 1.1	3.2 ± 2.0	5.43 ± 2.65	17.2 ± 6.7	34.9 ± 21.4
Features, d	1,433	3,703	932	2	100
Classes, c	7	6	5	2	2
Node hom. ratio, $\mathcal{H}_{\text{node}}$	0.6399	0.5711	0.4815	0.4793	0.5049
Edge hom. ratio, $\mathcal{H}_{\text{edge}}$	0.7462	0.6814	0.4675	0.4642	0.4851

We compare HyperUFG with MLP, HGNN [7], HyperGCN [21], UniGCNII [10], HyperND [18], AllDeepSets and AllSetTransformer [2], ED-HNN [20], SheafHyperGNN [6] FSGNN [17] and HyperUFG [13]. The baseline results are reproduced using their publicly available code, with hyperparameter set according to the original papers.

The experiments are conducted on a NVIDIA RTX 4090 GPU with 24GB of memory. For all new benchmarks, we utilize feature vectors, class labels, and ten random splits (40%/20%/40% of nodes per class for training/validation/testing, respectively). For the setting of SWDHFN, we select the quadratic B-spline function ($d = 2$) as the basis for the coarsest layer $\mathcal{H}_{J_0}$.

5.2 Experimental Results and Analysis

Table 2 presents the comparative performance of all models on the 5 benchmark datasets. As shown, SWDHFN achieves state-of-the-art overall performance,

outperforming all baselines including FSGNN. Notably, the performance gap between SWDHFN and FSGNN exhibits a clear pattern across homophilic and heterophilic datasets, which validates the effectiveness of our framelets design.

Table 2. Comparative performance of various HNNs on homophilic (Cora/Citeseer) and heterophilic (Actor/Senate/House) hypergraphs. The best-performing model is highlighted in lilac, the second-best in blue, and the third-best in gray.

Methods	Homophilic		Heterophilic			Rank
	Cora	Citeseer	Actor	Senate	House	
Edge hom. ratio ($\mathcal{H}_{\text{edge}}$)	0.7462	0.6814	0.4675	0.4642	0.4851	
MLP	75.16 ± 1.41	71.71 ± 1.01	85.45 ± 1.21	52.25 ± 5.17	51.86 ± 2.34	12
HGNN	79.39 ± 1.36	72.45 ± 1.16	74.47 ± 0.32	48.59 ± 4.52	61.39 ± 2.96	10
HyperGCN	78.45 ± 1.26	71.28 ± 0.82	68.67 ± 4.38	42.45 ± 3.67	48.32 ± 2.93	13
UniGCNII	78.81 ± 1.05	73.05 ± 2.21	80.48 ± 1.13	49.30 ± 4.25	67.25 ± 2.57	9
HyperND	79.20 ± 1.14	72.62 ± 1.49	92.52 ± 0.81	52.82 ± 3.20	51.70 ± 3.37	7
AllDeepSets	76.88 ± 1.80	70.63 ± 1.63	82.00 ± 2.33	48.17 ± 5.67	67.82 ± 2.40	11
AllSetTransformer	78.58 ± 1.47	73.08 ± 1.20	83.39 ± 1.73	51.83 ± 5.22	69.33 ± 2.20	8
ED-HNN	80.31 ± 1.35	73.70 ± 1.38	91.86 ± 0.43	64.79 ± 5.14	72.45 ± 2.28	4
SheafHyperGNN	81.30 ± 1.70	74.71 ± 1.23	80.09 ± 2.45	68.73 ± 4.68	73.84 ± 2.30	5
HyperUFG	81.51 ± 0.99	74.72 ± 2.10	89.32 ± 0.75	67.61 ± 7.00	72.82 ± 2.22	2nd
FSGNN	85.65 ± 0.79	75.28 ± 0.96	85.92 ± 0.27	62.37 ± 4.78	70.14 ± 1.50	3rd
SWDHFN	83.55 ± 0.72	74.19 ± 1.41	89.46 ± 0.26	67.89 ± 2.42	73.37 ± 2.22	1st

On Cora and Citeseer (homophilic datasets), FSGNN achieves slightly higher accuracy than SWDHFN: Cora: FSGNN (85.65 ± 0.79) outperforms SWDHFN (83.55 ± 0.72) by 2.1 percentage points. Citeseer: FSGNN (75.28 ± 0.96) outperforms SWDHFN (74.19 ± 1.41) by 1.09 percentage points.

In contrast to homophilic datasets, SWDHFN significantly outperforms FSGNN on all heterophilic datasets, highlighting the critical role of the spline wavelet framework: Actor: SWDHFN (89.46 ± 0.26) outperforms FSGNN (85.92 ± 0.27) by 3.54 percentage points. Senate: SWDHFN (67.89 ± 2.42) outperforms FSGNN (62.37 ± 4.78) by 5.17 percentage points. House: SWDHFN (73.37 ± 2.22) outperforms FSGNN (70.14 ± 1.50) by 3.23 percentage points.

The superior performance of SWDHFN on heterophilic hypergraphs can be attributed to following properties: Traditional models like FSGNN rely solely on spatial features from adjacency matrix powers, which aggregate noisy information from heterophilic neighbors. In contrast, SWDHFN's decimagted hypergraph framelets decompose hypergraph signals into low-frequency global trends and high-frequency local details. On heterophilic datasets, framelets emphasize label-discriminative local variations, while scaling functions preserve global structural integrity. In addition, The inherent smoothness of the spline wavelet basis enables stable and sparse representation of hypergraph signals, smooths local noisy fluctuations in feature values and thus boosts the robustness of hypergraph feature characterization. It has compact support which restricts feature

extraction to a finite local subset of hypergraph nodes and hyperedges, effectively cutting off the spatial propagation of noise from irrelevant heterophilic edges and avoiding meaningless global aggregation of noisy information.

6 Conclusion

This paper proposes a novel method to construct decimated hypergraph framelets with spline wavelets basis. The results show that combining hypergraph framelets and multihop aggregation improves the performance of node classification on heterophilic hypergraphs. Moreover, compared with using multihop aggregation alone, results show that our decimated hypergraph framelets can extract the feature of heterophilic hypergraphs better. We would also like to mention that choosing a better way to generate the coarse-grained chain $\mathcal{H}_{J\to J_0}$ has the potential to produce better hypergraph framelets, which will be a future experimental direction to be explored. Building on our work, it would be beneficial to theoretically and empirically investigate the potential relation between key issues such as homophily versus heterophily and their spectral properties from the perspective of hypergraph framelets to enhance the performance of hypergraph framelets or extending the spline wavelet-based framelet framework to capture complex high-order relational differences beyond node-hyperedge neighborhood discrepancies.

Acknowledgments. The work of C.Qiu and X.Zhuang was supported in part by the Research Grants Council of Hong Kong (Project no. CityU 11302023, CityU 11301224, and CityU 11300825).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Luís Cavique, Nuno C Marques, and António Gonçalves. A data reduction approach using hypergraphs to visualize communities and brokers in social networks. *Social Network Analysis and Mining*, 8(1):60, 2018.
2. Eli Chien, Changping Pan, Jian Peng, and Olgica Milenkovic. You are allset: A multiset function framework for hypergraph neural networks. In *arXiv preprint arXiv:2106.13264*, 2021.
3. Charles K Chui, H N Mhaskar, and Xiaosheng Zhuang. Representation of functions on big data associated with directed graphs. *Applied and Computational Harmonic Analysis*, 44(1):165–188, 2018.
4. Elaine Cohen, Tom Lyche, and Richard Riesenfeld. Discrete b-splines and subdivision techniques in computer-aided geometric design and computer graphics. *Computer graphics and image processing*, 14(2):87–111, 1980.
5. Ingrid Daubechies. *Ten lectures on wavelets*. SIAM, 1992.
6. Iulia Duta, Gennaro Cassarà, Fabrizio Silvestri, and Pietro Liò. Sheaf hypergraph networks. In *Advances in Neural Information Processing Systems*, volume 36, pages 12087–12099, 2023.

7. Yifan Feng, Haoxuan You, Zhe Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3558–3565, 2019.
8. 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.
9. Bin Han. Matrix splitting with symmetry and symmetric tight framelet filter banks with two high-pass filters. *Applied and Computational Harmonic Analysis*, 35(2):200–227, 2013.
10. Jing Huang and Jie Yang. Unignn: A unified framework for graph and hypergraph neural networks. In *arXiv preprint arXiv:2105.00956*, 2021.
11. Ming Li, Yujie Fang, Dongrui Shen, Han Feng, Xiaosheng Zhuang, Kelin Xia, and Pietro Liò. High-pass matters: Theoretical insights and sheaflet-based design for hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 23039–23046, 2026.
12. Ming Li, Yujie Fang, Yi Wang, Han Feng, Yongchun Gu, Lu Bai, and Pietro Liò. Deep hypergraph neural networks with tight framelets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 18385–18392, 2025.
13. Ming Li, Yongchun Gu, Yi Wang, Yujie Fang, Lu Bai, Xiaosheng Zhuang, and Pietro Liò. When hypergraph meets heterophily: New benchmark datasets and baseline. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 18377–18384, 2025.
14. Ming Li, Yi Wang, Chengling Gao, Lu Bai, Yujie Fang, Xiaosheng Zhuang, and Pietro Liò. Permutation equivariant framelet-based hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 23079–23086, 2026.
15. T Lyche, K Mørken, and E Quak. Theory and algorithms for non-uniform spline wavelets. *Multivariate Approximation and Applications*, 152, 2001.
16. Tom Lyche and Knut Mørken. Making the oslo algorithm more efficient. *SIAM Journal on Numerical Analysis*, 23(3):663–675, 1986.
17. S K Maurya, X Liu, and T Murata. Simplifying approach to node classification in graph neural networks. *Journal of Computational Science*, 62:101695, 2022.
18. Konstantin Prokopchik, Austin R Benson, and Francesco Tudisco. Nonlinear feature diffusion on hypergraphs. In *International Conference on Machine Learning*, pages 17945–17958. PMLR, 2022.
19. Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
20. Ping Wang, Shuang Yang, Yuyang Liu, Zhen Wang, and Ping Li. Equivariant hypergraph diffusion neural operators. In *arXiv preprint arXiv:2207.06680*, 2022.
21. Naganand Yadati, Madhav Nimishakavi, Preetish Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. Hypergcn: A new method for training graph convolutional networks on hypergraphs. In *Advances in Neural Information Processing Systems*, volume 32, 2019.
22. Xuebin Zheng, Bingxin Zhou, Yu Guang Wang, and Xiaosheng Zhuang. Decimated framelet system on graphs and fast g-framelet transforms. *Journal of Machine Learning Research*, 23(18):1–68, 2022.
23. Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. *Advances in Neural Information Processing Systems*, 19, 2006.