Appendix for "Deep Hypergraph Neural Networks with Tight Framelets"

Ming Li, Yujie Fang, Yi Wang*, Han Feng, Yongchun Gu, Lu Bai*, Pietro Lio

AAAI 2025

A Theoretical Properties of Tight Framelets on Hypergraphs

As discussed in Section 3.2, our theoretical results, including Theorem 1 and its proof, demonstrate that when $\|\mathbf{P}x\|_2 = 0$, the feature vectors in a deep GCN collapse into a one-dimensional space, represented $x = \gamma \mathbf{D}_v^{1/2}\mathbf{1}$ for some $\gamma \in \mathbb{R}$. This collapse leads to a loss of node distinguishability, which is a key factor in the oversmoothing problem. To address this, we introduce more advanced filters that perform convolutional operations on a channel-by-channel basis, capturing both low-pass and high-pass features. This approach preserves a richer set of features across network layers, enhancing the model's ability to maintain node distinguishability and offering deeper insights into the underlying data structure. From an implementation perspective, our proposed FrameHGNN model employs framelet-based hypergraph convolutions, incorporating tight framelet transforms with both low-pass and high-pass components. In this appendix, we explore the theoretical properties of hypergraph framelets, focusing on the performance of framelet-based decomposition and reconstruction operators when applied to hypergraph signals. Specifically, we investigate how the collaborative contribution of low-pass and high-pass components facilitates effective processing of hypergraph signals, leading to a more comprehensive understanding of signal representation and reconstruction.

Recalling the notations introduced in Section 4, consider a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N nodes and hypergraph Laplacian $\mathcal{L}$. Let $\mathbf{U} = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_N]$ denote the matrix of eigenvectors of $\mathcal{L}$, and $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_N)$ be the diagonal matrix of the corresponding eigenvalues. Framelets over the hypergraph are generated by a set of scaling functions $\Phi = \{\gamma; \rho^{(1)}, \cdots, \rho^{(n)}\} \subset L_1(\mathbb{R})$ associated with a filter bank $\eta = \{a; b^{(1)}, \cdots b^{(n)}\}$, which satisfy the relations for any $\xi \in \mathbb{R}$:

$$\widehat{\gamma}(2\xi) = \widehat{a}(\xi)\widehat{\gamma}(\xi) \quad \widehat{\rho^{(r)}}(2\xi) = \widehat{b^{(r)}}(\xi)\widehat{\gamma}(\xi). \tag{S-1}$$

These functions are defined as follows (for clarity, Equations (4) and (5) from Section 4 of the main manuscript are restated here for reference):

$$\psi_{j,p}(\mu) = \sum_{q=1}^{N} \widehat{\gamma}\left(\frac{\lambda_q}{2^j}\right) u_q(p)u_q(\mu), \tag{S-2}$$

$$\phi^r_{j,p}(\mu) = \sum_{q=1}^{N} \widehat{\rho^{(r)}}\left(\frac{\lambda_q}{2^j}\right) u_q(p)u_q(\mu),\ r = 1, \ldots, n, \tag{S-3}$$

where $u_q(p)$ represent the eigenvector $\boldsymbol{u}_q$ at node p.

For integers J, J_1 such that $J > J_1$, we define a tight framelet system on hypergraphs (denoted as ***TifHyper*** $(\Phi, \eta; \mathcal{G})$), starting from scale J_1, as a non-homogeneous, stationary affine system:

$$\textbf{\textit{TifHyper}}^J_{J_1}(\Phi, \eta; \mathcal{G}) = \{\psi_{J_1,p} : p \in \mathcal{V}\} \cup \{\phi^r_{j,p} : p \in \mathcal{V}, j = J_1, \ldots, J\}^n_{r=1}. \tag{S-4}$$

In light of the theoretical background discussed above, we now present the following formal properties of tight framelets on hypergraphs.

Theorem A1 (Properties of Tight Framelets on Hypergraphs). Let $J \geq 1$ be an integer, and consider the hypergraph framelet system ***TifHyper***$^J_{J_1}(\Phi, \eta; \mathcal{G})$ defined in (S-4), with hypergraph framelets $\psi_{j,p}$ and $\phi^r_{j,p}$. The following statements are equivalent:

(i) For each $J_1 = 1, \ldots, J$, the framelet system on hypergraphs, ***TifHyper***$^J_{J_1}(\psi, \eta; \mathcal{G})$, is a tight frame for $l_2(\mathcal{G})$. That is, $\forall f \in l_2(\mathcal{G})$,

$$\|f\|^2 = \sum_{p\in\mathcal{V}} \Big| \langle f, \psi_{J_1,p}\rangle \Big|^2 + \sum_{j=J_1}^{J}\sum_{r=1}^{n}\sum_{p\in\mathcal{V}} \Big| \langle f, \phi^r_{j,p}\rangle \Big|^2. \tag{S-5}$$

(ii) For all $f \in l_2(\mathcal{G})$ and for $j = 1, \ldots, J-1$, the following identities hold:

$$f = \sum_{p\in\mathcal{V}} \langle f, \psi_{J,p}\rangle\, \psi_{J,p} + \sum_{r=1}^{n}\sum_{p\in\mathcal{V}} \langle f, \phi^r_{J,p}\rangle\, \phi^r_{J,p}, \tag{S-6}$$

$$\sum_{p\in\mathcal{V}} \langle f, \psi_{j+1,p}\rangle\, \psi_{j+1,p} = \sum_{p\in\mathcal{V}} \langle f, \psi_{j,p}\rangle\, \psi_{j,p} + \sum_{r=1}^{k}\sum_{p\in\mathcal{V}} \langle f, \phi^r_{j,p}\rangle\, \phi^r_{j,p}. \tag{S-7}$$

(iii) For all $f \in l_2(\mathcal{G})$ and for $j = 1, \ldots, J-1$, the following identities hold:

$$\|f\|^2 = \sum_{p\in\mathcal{V}} \big|\langle f, \psi_{J,p}\rangle\big|^2 + \sum_{r=1}^{n}\sum_{p\in\mathcal{V}} \big|\langle f, \phi^r_{J,p}\rangle\big|^2, \tag{S-8}$$

$$\sum_{p\in\mathcal{V}} \big|\langle f, \psi_{j+1,p}\rangle\big|^2 = \sum_{p\in\mathcal{V}} \big|\langle f, \psi_{j,p}\rangle\big|^2 + \sum_{r=1}^{n}\sum_{p\in\mathcal{V}} \big|\langle f, \phi^r_{j,p}\rangle\big|^2. \tag{S-9}$$

(iv) The scaling functions in Φ satisfy

$$1 = \left|\widehat{\gamma}\left(\frac{\lambda_q}{2^J}\right)\right|^2 + \sum_{r=1}^{k}\left|\widehat{\rho^{(r)}}\left(\frac{\lambda_q}{2^J}\right)\right|^2 \quad \forall q = 1,\ldots,N, \tag{S-10}$$

$$\left|\widehat{\gamma}\left(\frac{\lambda_q}{2^{j+1}}\right)\right|^2 = \left|\widehat{\gamma}\left(\frac{\lambda_q}{2^j}\right)\right|^2 + \sum_{r=1}^{k}\left|\widehat{\rho^{(r)}}\left(\frac{\lambda_q}{2^j}\right)\right|^2 \quad \forall \begin{array}{l} q = 1,\ldots,N, \\ j = 1,\ldots,J-1. \end{array} \tag{S-11}$$

(v) The identities in (S-10) hold and the filters in the filter bank η satisfy

$$\left|\widehat{a}\left(\frac{\lambda_q}{2^j}\right)\right|^2 + \sum_{r=1}^{n}\left|\widehat{b^{(r)}}\left(\frac{\lambda_q}{2^j}\right)\right|^2 = 1 \quad \forall q \in \sigma_\lambda^{(j)},\ j = 2,\ldots,J, \tag{S-12}$$

with

$$\sigma_\lambda^{(j)} := \left\{q \in \{1,\ldots,N\} : \widehat{\gamma}\left(\frac{\lambda_q}{2^j}\right) \neq 0\right\}.$$

Proof: (i)$\Longleftrightarrow$(ii). Let $\Psi_j := \operatorname{span}\{\psi_{j,p} : p \in \mathcal{V}\}$ and $\Phi_j^r := \operatorname{span}\{\phi_{j,p}^r : p \in \mathcal{V}\}$. Define projections $\mathbf{P}_{\Psi_j}, \mathbf{P}_{\Phi_j^r}$, $r = 1,\ldots,n$ by

$$\mathbf{P}_{\Psi_j}(f) := \sum_{p\in\mathcal{V}} \langle f, \psi_{j,p}\rangle\, \psi_{j,p}, \quad \mathbf{P}_{\Phi_j^r} := \sum_{p\in\mathcal{V}} \langle f, \phi_{j,p}^r\rangle\, \phi_{j,p}^r, \quad f \in l_2(\mathcal{G}). \tag{S-13}$$

Since ***TifHyper***$_{J_1}^J(\Psi,\eta)$ is a tight frame on hypergraphs for $l_2(\mathcal{G})$ ($J_1 = 1,\ldots,J$), we obtain by polarization identity,

$$f = \mathbf{P}_{\Psi_{J_1}}(f) + \sum_{j=J_1}^{J}\sum_{r=1}^{k}\mathbf{P}_{\Phi_j^r}(f) = \mathbf{P}_{\Psi_{J_1+1}}(f) + \sum_{j=J_1+1}^{J}\sum_{r=1}^{k}\mathbf{P}_{\Phi_j^r}(f) \tag{S-14}$$

for all $f \in l_2(\mathcal{G})$ and for all $J_1 = 1,\ldots,J$. Thus, for $J_1 = 1,\ldots,J-1$,

$$\mathbf{P}_{\Psi_{J_1+1}}(f) = \mathbf{P}_{\Psi_{J_1}}(f) + \sum_{r=1}^{k}\mathbf{P}_{\Phi_{J_1}^r}(f), \tag{S-15}$$

which is (S-7). Moreover, when $J_1 = J$, (S-14) gives (S-6). Consequently, (i)$\Longrightarrow$(ii). Conversely, recursively using (S-15) gives

$$\mathbf{P}_{\Psi_{m+1}}(f) = \mathbf{P}_{\Psi_{J_1}}(f) + \sum_{j=J_1}^{m}\sum_{r=1}^{k}\mathbf{P}_{\Phi_j^r}(f) \tag{S-16}$$

for all $J_1 \le m \le J-1$. Taking $m = J-1$ together with (S-6), we deduce (S-14), which is equivalent to (S-5). Thus, (ii)$\Longrightarrow$(i).

(ii)$\Longleftrightarrow$(iii). The equivalence between (ii) and (iii) simply follows from the polarization identity.

(ii)$\Longleftrightarrow$(iv). By the orthonormality of $\boldsymbol{u}_p$,

$$\langle f, \psi_{j,q} \rangle = \sum_{q=1}^{N} \widehat{\gamma}\left(\frac{\lambda_q}{2^j}\right) \widehat{f}_\ell \, u_q(p), \quad \langle f, \phi^r_{j,q} \rangle = \sum_{q=1}^{N} \widehat{\rho^{(r)}}\left(\frac{\lambda_q}{2^j}\right) \widehat{f}_\ell \, u_q(p),$$

where $\widehat{f}_q = \langle f, \boldsymbol{u}_q \rangle$ is the Fourier coefficient of f with respect to $\boldsymbol{u}_q$. This together with (S-13), (S-2) and (S-3) gives, for $j \geq 1$ and $r = 1, \ldots, n$, the Fourier coefficients for the projections $\mathbf{P}\phi_j(f)$ and $\mathbf{P}\psi_j(f)$:

$$\left(\widehat{\mathbf{P}_{\Psi_j}(f)}\right)_p = \left|\widehat{\gamma}\left(\frac{\lambda_q}{2^j}\right)\right|^2 \widehat{f}_q, \quad \left(\widehat{\mathbf{P}_{\Phi^r_j}(f)}\right)_q = \left|\widehat{\rho^{(r)}}\left(\frac{\lambda_p}{2^j}\right)\right|^2 \widehat{f}_q, \quad \forall q = 1, \ldots, N, \tag{S-17}$$

which implies that (S-6) and (S-7) are equivalent to (S-10) and (S-11) respectively. Thus, (ii)$\Longleftrightarrow$(iv).

(iv)$\Longleftrightarrow$(v). Based on the relations (S-1) that $\widehat{\gamma}(2\xi) = \widehat{a}(\xi)\widehat{\gamma}(\xi)$ and $\widehat{\rho^{(r)}}(2\xi) = \widehat{b^{(r)}}(\xi)\widehat{\gamma}(\xi)$ for any $\xi \in \mathbb{R}$, it can be deduced that for $q = 1, \ldots, N$ and $j \geq 1$,

$$\left|\widehat{\gamma}\left(\frac{\lambda_q}{2^j}\right)\right|^2 + \sum_{r=1}^{n}\left|\widehat{\rho^{(r)}}\left(\frac{\lambda_q}{2^j}\right)\right|^2 = \left(\left|\widehat{a}\left(\frac{\lambda_q}{2^{j+1}}\right)\right|^2 + \sum_{r=1}^{n}\left|\widehat{b^{(r)}}\left(\frac{\lambda_q}{2^{j+1}}\right)\right|^2\right)\left|\widehat{\gamma}\left(\frac{\lambda_q}{2^{j+1}}\right)\right|^2.$$

This shows that (S-11) is equivalent to (S-12). Therefore, (iv)$\Longleftrightarrow$(v).

B Additional Details on Experimental Studies

We provide the pseudocode for implementing FrameHGNN in the following **Algorithm 1**.

Algorithm 1. FrameHGNN

Input: Hypergraph $\mathcal{G}$, incidence matrix $\mathbf{H}$, hypergraph Laplacian $\mathcal{L}$, feature matrix $\mathbf{X}$, the number of layers l

Output: $\mathbf{X}^{(\ell)}$

1. Compute the eigenvalue and eigenvector pairs $\{(\lambda_q, \boldsymbol{u}_q)\}_{j=1}^{N}$ for $\mathcal{L}$ of hypergraph $\mathcal{G}$ with N nodes.
2. Define low-pass and high-pass framelets using Eq. (4) and Eq. (5).
3. Compute low-pass and high-pass coefficients $V_0, W_j^r \in \mathbb{R}^{N\times d}$ using Eq. (6) and Eq. (7).

For $i = 0, 1, ..., l-1$, **do:**

4. **Decomposition:** Use $\mathcal{W}_{r,j}$ as decomposition operators to represent the framelet transform matrices for decomposition as shown in Eq. (8).
5. **Reconstruction:** Use $\mathcal{W}_{r,j}^{\top}$ for reconstruction
6. Add initial residual and identity mapping techniques using Eq. (10)

Table B-1 presents an overview of the eight datasets utilized in our experiments, highlighting their key characteristics.

Table B-1: The statistics of datasets

Dataset	Nodes	Edges	Classes	Features
Cora	2708	1579	7	1433
Citeseer	3312	1709	6	3703
Pubmed	19717	7963	3	500
Cora-CA	2708	1072	7	1433
Senate	282	315	2	100
House	1290	340	2	100
NTU2012	2012	2012	67	100
ModelNet40	12311	12311	40	100

Table B-2 details the parameter search space explored in the node classification experiments. We note that all experiments were conducted using a single NVIDIA RTX A6000 GPU and implemented in PyTorch.

To further investigate the key characteristics of the tight framelet-based convolution, whose theoretical properties are detailed in Appendix A, we provide an additional analysis of the hyperparameter γ. This parameter governs the balance between the framelet-based convolution, which incorporates both low-pass and high-pass filters, and the standard low-pass filter $\mathbf{F}\mathbf{X}^{(\ell)}$ (with $\mathbf{F} := \mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{D}_e^{-1}\mathbf{H}^{\top}\mathbf{D}_v^{-1/2}$). Then we have $\mathcal{F}(\mathbf{X}^{(\ell)}) = \gamma\sum_{(r,j)\in\Gamma}\mathcal{W}_{r,j}^{\top}\text{diag}(\theta_{r,j})\mathcal{W}_{r,j}\mathbf{X}^{(\ell)} + (1-\gamma)\mathbf{F}\mathbf{X}^{(\ell)}$ where $\theta_{r,j} \in \mathbb{R}^N$ are

Table B-2: Hyperparameter searching space for node classification.

Hyperparameters	**Searching space**
Learning rate	1e-3,2e-3,3e-3
Weight decay	1e-3,5e-3,1e-4,2e-4,5e-4,1e-5
Hidden Size	32,64,128,256,512
Dropout ratio	0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9
Level	1,2,3
Alpha	0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9
Gamma	0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9
Lambda	0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9
Seed	50,200,500,1000

learnable filter, and $\Gamma = \{(r, j) : r = 1, \ldots, R, j = 0, 1, \ldots, J\} \cup \{(0, J)\}$ is the index set for all framelet decomposition matrices.

Moreover, the choice of the parameter β is crucial for ensuring the adaptive decay of the weight matrix as we increase the number of layers. In practice, we set $\beta_\ell = \log\left(\frac{\lambda}{\ell} + 1\right)$, where λ serves as a refined parameter analogous to β. To further understand the sensitivity of the model to these parameters, we conduct experiments analyzing the impact of α (see Eq. (10) in Section 4), γ, and λ. Specifically, Figure B-1 presents the sensitivity analysis for the Senate and Citeseer datasets, demonstrating results that are consistent with those shown in Figure 4 of the main manuscript. Overall, FrameHGNN exhibits stable performance across various configurations of these parameters, suggesting that the model is relatively insensitive to changes in α, γ, and λ.

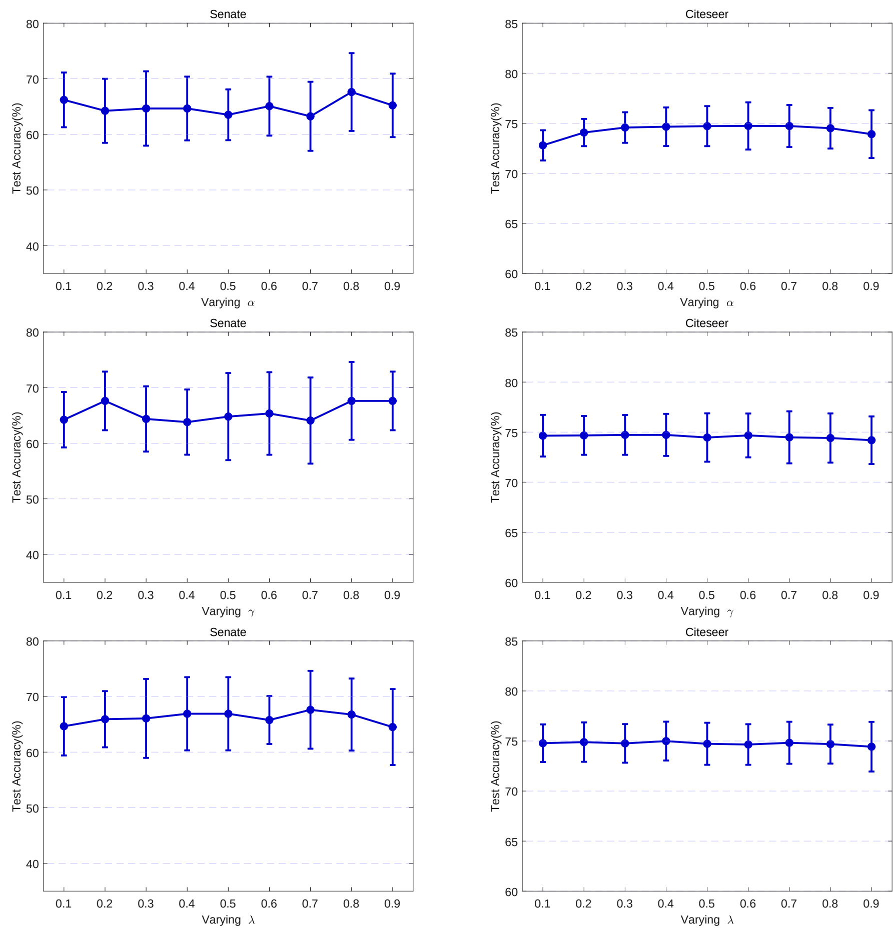

Figure B-1: Parameter sensitivity analysis for FrameHGNN on the Senate (left) and Citeseer (right) datasets.

Table B-3 specifies the reproducible parameters corresponding to the optimal results obtained in these experiments.

Table B-3: Hyperparameter settings for different datasets used in the experiments.

Dataset	Hyperparameter Setting	
Cora	Learning rate: 2e-3 Weight decay: 1e-3 Hidden Size: 512 Dropout ratio: 0.6 Level: 3	Alpha: 0.2 Gamma: 0.2 Lambda: 0.3 Seed: 50
Citeseer	Learning rate: 2e-3 Weight decay: 1e-5 Hidden Size: 512 Dropout ratio: 0.5 Level: 1	Alpha: 0.7 Gamma: 0.4 Lambda: 0.4 Seed: 500
Pubmed	Learning rate: 2e-3 Weight decay: 1e-5 Hidden Size: 512 Dropout ratio: 0.5 Level: 1	Alpha: 0.2 Gamma: 0.5 Lambda: 0.5 Seed: 1000
Cora-CA	Learning rate: 2e-3 Weight decay: 1e-3 Hidden Size: 512 Dropout ratio: 0.7 Level: 3	Alpha: 0.2 Gamma: 0.2 Lambda: 0.6 Seed: 50
Senate	Learning rate: 3e-3 Weight decay: 1e-3 Hidden Size: 256 Dropout ratio: 0.8 Level: 3	Alpha: 0.8 Gamma: 0.9 Lambda: 0.7 Seed: 200
House	Learning rate: 3e-3 Weight decay: 1e-3 Hidden Size: 512 Dropout ratio: 0.6 Level: 1	Alpha: 0.1 Gamma: 0.5 Lambda: 0.8 Seed: 500
NTU2012	Learning rate: 2e-3 Weight decay: 1e-5 Hidden Size: 512 Dropout ratio: 0.2 Level: 1	Alpha: 0.5 Gamma: 0.5 Lambda: 0.1 Seed: 50
ModelNet40	Learning rate: 2e-3 Weight decay: 1e-4 Hidden Size: 256 Dropout ratio: 0.4 Level: 1	Alpha: 0.4 Gamma: 0.4 Lambda: 0.7 Seed: 50

C Computational Complexity Analysis

In this section, we provide an overview of the training computational complexity for four state-of-the-art hypergraph neural networks and our proposed FrameHGNN model. Table C-1 summarizes the estimated training complexity analysis, where the following notations are used:

- N is the number of nodes in the given hypergraph
- M refers to the number of hyperedges in the given hypergraph
- M' is the number of edges in the clique expansion (when transforming the hypergraph into a regular graph)
- $||\mathbf{H}||_0$ represents the number of non-zero values in the incidence matrix $\mathbf{H}$
- T refers to the number of training epochs
- L is the number of layers
- d is the feature dimension
- n indicates the number of high-pass filters used in FrameHGNN
- J is the scale level used in FrameHGNN
- K refers to the largest number of non-zero values in the framelet transform matrices $\mathcal{W}r, j$ (see more details in Section 4)

Importantly, n, J, K are constants independent of the given hypergraph, and both n and J typically take small values in practical implementations. The sparsity property of the constructed hypergraph framelets ensures that K is generally not large and may even be smaller than or approximately equivalent to$||\mathbf{H}||_0$. As a result, FrameHGNN offers competitive performance without imposing any additional computational burden compared to existing methods. Specifically, the computational complexity of Frame-HGNN is approximately on par with models like AllDeepSetss (Chien et al. 2022)and ED-HNN (Wang et al. 2023).

Table C-1: Summary of training computational complexity for UniGCNII, Deep-HGNN, AllDeepSets, ED-HNN, and our proposed FrameHGNN model.

Name	Training Computational Complexity
UniGCNII (Huang and Yang 2021)	$\mathcal{O}\Big(TL(N + M + \|\mathbf{H}\|_0)d + TLNd^2\Big)$
Deep-HGCN (Chen et al. 2022)	$\mathcal{O}\Big(TLM'd + TLNd^2\Big)$
AllDeepSets (Chien et al. 2022)	$\mathcal{O}\Big(TL\|\mathbf{H}\|_0 d + TL(N + M)d^2\Big)$
ED-HNN (Wang et al. 2023)	$\mathcal{O}\Big(TL\|\mathbf{H}\|_0 d + TL(N + M)d^2\Big)$
FrameHGNN (Ours)	$\mathcal{O}\Big(TL(nJ + 1)Kd + TL(N + M)d^2\Big)$

References

Chen, G.; Zhang, J.; Xiao, X.; and Li, Y. 2022. Preventing over-smoothing for hypergraph neural networks. *arXiv preprint arXiv:2203.17159*.

Chien, E.; Pan, C.; Peng, J.; and Milenkovic, O. 2022. You are AllSet: a multiset function framework for hypergraph neural networks. In *ICLR*.

Huang, J.; and Yang, J. 2021. UniGNN: a unified framework for graph and hypergraph neural networks. In *IJCAI*, 2563–2569.

Wang, P.; Yang, S.; Liu, Y.; Wang, Z.; and Li, P. 2023. Equivariant hypergraph diffusion neural operators. In *ICLR*.