Appendix for " Permutation Equivariant Framelet-based Hypergraph Neural Networks"

Ming Li, Yi Wang*, Chengling Gao*, Lu Bai, Yujie Fang, Xiaosheng Zhuang, Pietro Lio

AAAI 2026

1 Notation Summary

For clarity and ease of reference, we summarize the frequently used mathematical symbols and notations throughout Section 3 (in the main manuscript) in Table 1.

Table 1: Summary of the frequently used symbols

Symbol	Description
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Hypergraph with vertex set $\mathcal{V}$ and hyperedge set $\mathcal{E}$
$N = \lvert\mathcal{V}\rvert$, $M = \lvert\mathcal{E}\rvert$	Number of vertices and hyperedges
$\mathbf{X} \in \mathbb{R}^{N\times d}$	Vertex feature matrix
$\mathbf{Y} \in \mathbb{R}^{M\times o}$	Hyperedge feature matrix
$\mathbf{H} \in \{0,1\}^{N\times M}$	Incidence matrix of the hypergraph
$\mathbf{D}_v, \mathbf{D}_e$	Diagonal matrices of vertex and hyperedge degrees
$\mathbf{f} \in \mathbb{R}^N$	Hypergraph signal defined on vertices
$L_2(\mathcal{G})$	Hilbert space of square-integrable signals on $\mathcal{G}$
$\langle \mathbf{f}, \mathbf{g} \rangle$	Inner product in $L_2(\mathcal{G})$
$\mathcal{P}_K = \{\mathcal{V}_j\}_{j=1}^K$	K-level hierarchical clustering of $\mathcal{V}$
$s_{\mathbf{\Lambda}}$	Cluster (node group) indexed by $\mathbf{\Lambda}$
$\mathbf{\Lambda} \in \mathbb{N}^j$	Index vector of depth-j in the hierarchy
$\dim(\mathbf{\Lambda})$	Level (depth) of index $\mathbf{\Lambda}$
$L_{\mathbf{\Lambda}}$	Number of children of cluster $s_{\mathbf{\Lambda}}$
$\phi_{\mathbf{\Lambda}}$	Scaling (low-pass) vector supported on $s_{\mathbf{\Lambda}}$
$\psi_{(\mathbf{\Lambda},i)}$	Framelet (high-pass) vector corresponding to pairwise difference
$\mathbf{B}_{\mathbf{\Lambda}}$	Difference matrix used to construct framelets
$\mathcal{F}_{j_0}(\mathcal{P}_K)$	Framelet system at level j_0 over tree $\mathcal{P}_K$
$\mathcal{F} \in \mathbb{R}^{I_{\mathcal{G}}\times N}$	Matrix form of all framelet vectors
$I_{\mathcal{G}}$	Total number of vectors in $\mathcal{F}_{j_0}(\mathcal{P}_K)$
$\mathrm{nnz}(\mathcal{F})$	Number of nonzero entries in $\mathcal{F}$
$\hat{\mathbf{f}} = \mathcal{F}\mathbf{f}$	Framelet coefficients of signal $\mathbf{f}$
$\pi : \mathcal{V} \to \mathcal{V}$	Permutation (reordering) of vertices
$\mathbf{P}_\pi \in \mathbb{R}^{N\times N}$	Permutation matrix corresponding to π
$\pi(\mathcal{G})$	Permuted hypergraph
$\pi(\mathcal{E}) = \{\pi(e) : e \in \mathcal{E}\}$	Permuted hyperedges
$\mathcal{A}(\mathcal{G}, \mathcal{P}_K)$	Framelet matrix constructed on hypergraph $\mathcal{G}$ with tree $\mathcal{P}_K$
$\mathcal{A}(\pi(\mathcal{G}), \mathcal{P}_K)$	Framelet matrix on permuted hypergraph
$\mathcal{F}^\top\mathcal{F} = \mathbf{I}$	Tight frame property of $\mathcal{F}_{j_0}(\mathcal{P}_K)$
$p_{(\mathbf{\Lambda},\ell)}$	Coefficients in the recursive construction of $\phi_{\mathbf{\Lambda}}$
$\mathrm{supp}(\phi_{\mathbf{\Lambda}})$	Support of the scaling vector
$\lVert\phi_{\mathbf{\Lambda}}\rVert = 1$	Normalization of scaling vector
$[I] = \{1, \ldots, I\}$	Shorthand for indexing set
$\mathbf{M}_{:i}$, $\mathbf{M}_{i:}$	i-th column and row of matrix $\mathbf{M}$

2 Additional Details for Experimental Implementation

Datasets. We evaluate PEF-HNN on seven representative hypergraph datasets, summarized in Table 2. This collection includes widely-used homophilic datasets such as co-citation networks (Cora, Citeseer) and co-authorship networks (Cora-CA) (Yadati et al., 2019), as well as heterophilic datasets including Actor, Twitch (Li et al., 2025), and U.S. Senate/House voting records (Fowler, 2006). These datasets span a broad range of sizes, structures, and relational patterns, providing a comprehensive testbed for evaluating the effectiveness of hypergraph models in both homophilic and heterophilic settings.

Table 2: Statistics of the benchmark datasets used in our experiments.

Datasets	Hypernodes	Hyperedges	Depth of Binary Tree	Size of Framelet Matrix
Cora	2708	1579	5	113351×2708
Citeseer	3312	1079	6	83811×3312
Cora-CA	2708	1072	8	13261×2708
Actor	16255	10164	10	122455×16255
Twitch	16812	2627	11	62816×16812
Senate	282	315	6	552×282
House	1290	341	8	2884×1290

Baselines. We compare PEF-HNN against representative methods from two major categories of hypergraph neural networks (HNNs):

- *Homophily-oriented HNNs:* Models such as HGNN (Feng et al., 2019), HyperGCN (Yadati et al., 2019), AllDeepSets and AllSetTransformer (Chien et al., 2022), and the UniGNN family (Huang and Yang, 2021) (including UniGCN, UniSAGE, UniGAT, and UniGCNII) have shown strong performance on homophilic datasets. However, their evaluations primarily focus on such settings, with limited consideration for heterophilic hypergraphs.
- *Heterophily-aware HNNs:* ED-HNN (Wang et al., 2023) and HyperUFG (Li et al., 2025) are more recent efforts that address heterophilic hypergraph learning. Although these models mark a step forward, their evaluations often rely on limited-scale or weakly heterophilic datasets, as discussed in the main manuscript.

Hyperparameter Settings. Table 4 provides the complete hyperparameter settings corresponding to the best-performing results reported in Table 1 of the main paper. Key hyperparameters, such as learning rate, weight decay, hidden dimension, number of layers, dropout rate, α, and β, are optimized through univariate sensitivity analysis within the search space defined in Table 3.

Table 3: Hyperparameter searching space.

Hyperparameters	Searching space
Learning rate	{5e-3, 3e-3, 2e-3, 1e-3, 5e-2, 3e-2, 2e-2, 1e-2}
Weight decay	{5e-5, 1e-5, 5e-4, 1e-4, 5e-3, 1e-3}
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}
Layers	{1, 2, 4, 8, 16, 32, 64, 128}
Alpha	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}
Beta	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}

Table 4: Hyperparameter settings for each dataset used in the experiments.

Dataset	Hyperparameter Setting	
Cora	Learning rate: 2e-3 Weight decay: 1e-3 Hidden Size: 512 Layers: 4	Dropout ratio: 0.7 Alpha: 0.2 Beta: 0.3
Citeseer	Learning rate: 2e-3 Weight decay: 1e-5 Hidden Size: 512 Layers: 128	Dropout ratio: 0.5 Alpha: 0.7 Beta: 0.5
Cora-CA	Learning rate: 2e-3 Weight decay: 1e-3 Hidden Size: 512 Layers: 64	Dropout ratio: 0.7 Alpha: 0.2 Beta: 0.6
Actor	Learning rate: 3e-3 Weight decay: 2e-4 Hidden Size: 512 Layers: 4	Dropout ratio: 0.1 Alpha: 0.1 Beta: 0.8
Twitch	Learning rate: 3e-3 Weight decay: 2e-4 Hidden Size: 512 Layers: 4	Dropout ratio: 0.1 Alpha: 0.1 Beta: 0.8
Senate	Learning rate: 2e-3 Weight decay: 1e-5 Hidden Size: 256 Layers: 8	Dropout ratio: 0.8 Alpha: 0.8 Beta: 0.7
House	Learning rate: 3e-3 Weight decay: 2e-3 Hidden Size: 512 Layers: 8	Dropout ratio: 0.6 Alpha: 0.1 Beta: 0.8

3 Additional Experimental Results and Discussion

3.1 Additional Visualization Results

In **Section 4.7 of the main paper**, we only present a comparative analysis of raw features and learned embeddings for the Citeseer and House datasets (see Figure 3 in the main manuscript) due to space limitations. Here, we complement those findings by providing supplementary t-SNE visualizations on three additional datasets: Cora-CA (top row), Senate (middle row), and Actor (bottom row), as shown in Figure 1. Across all three datasets, we observe that the representations learned by PEF-HNN yield more discriminative and well-clustered structures compared to other baselines. On the Cora-CA dataset, which is homophilic in nature, PEF-HNN effectively preserves class boundaries while enhancing intra-class compactness, outperforming HGNN and UniGCNII in terms of visual separability. For the Senate dataset, which exhibits strong heterophily, the embeddings learned by PEF-HNN avoid the degenerate structures observed in ED-HNN and UniGCNII (e.g., serpentine or linear clusters), and instead produce a more meaningful global layout that better reflects underlying label relationships. Similarly, on the Actor dataset, PEF-HNN generates clearer class separability than competing methods, with less entanglement among different classes.

These results further support our claims in the main paper: PEF-HNN is effective not only in homophilic settings but also demonstrates strong capacity to model and disentangle complex relational patterns in heterophilic hypergraphs.

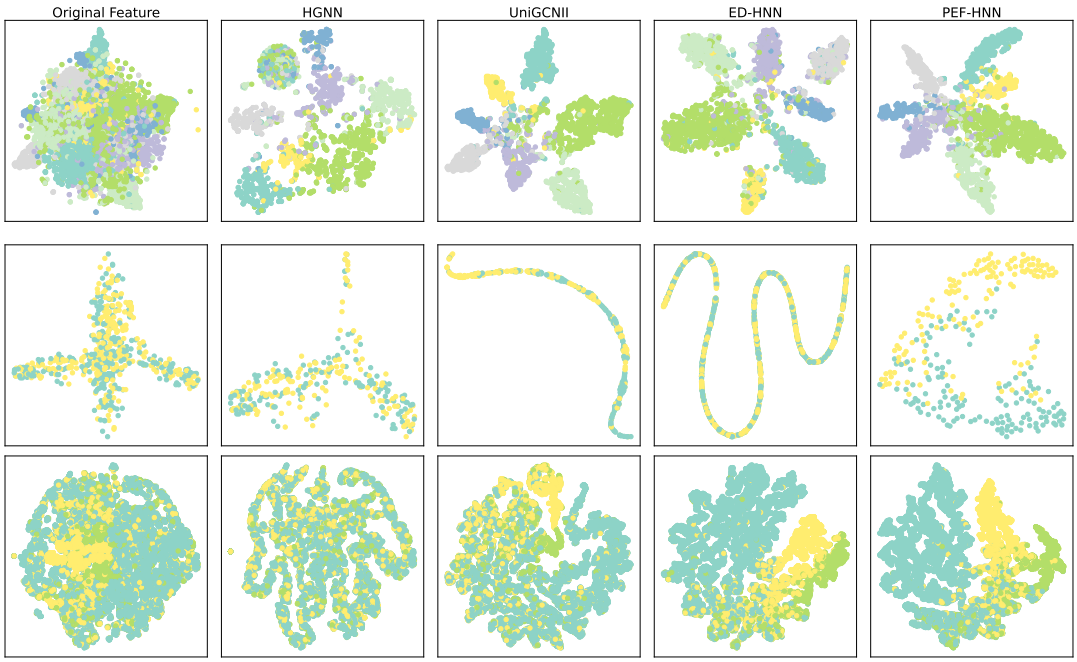

Figure 1: Visualization comparison of raw input features and learned representations on Cora-CA (top), Senate (middle), and Actor (bottom).

3.2 Further Details on Visual Object Classification

Real-World Datasets and Experimental Settings. We evaluate our method on two widely used benchmark datasets for visual object classification: the Princeton ModelNet40 dataset (Wu et al., 2015) and the National Taiwan University (NTU) 3D model dataset (Chen et al., 2003). A summary of these datasets is provided in Table 5.

ModelNet40 consists of 12,311 3D objects across 40 common object categories. Following the HGNN experimental protocol, we adopt the standard split of 9,843 training samples and 2,468 test samples. The NTU dataset comprises 2,012 3D shapes spanning 67 categories (e.g., car, chair, chess, chip, clock, cup, door, frame, pen, and plant leaf), which we partition into 80% for training and 20% for testing.

The reported results for GCN and HGNN are reproduced from (Feng et al., 2019). For GCN, performance on these non-graph visual datasets is obtained by constructing probabilistic graphs using distance-based metrics. For both HGNN and PEF-HNN, the hypergraph structures are constructed following the same procedure illustrated in Figure 2.

Table 5: The detailed information of the ModelNet40 and the NTU2012 datasets.

Dataset	Objects	MVCNN Feature	GVCNN Feature	Training Nodes	Testing Nodes	Classes	Size of Framelet Matrix
ModelNet40	12311	4096	2048	9843	2468	40	33149×12311
NTU2012	2012	4096	2048	1639	373	67	7189×2012

Hypergraph Structure Construction. Figure 2 presents the workflow for constructing hypergraphs from 3D visual data, highlighting the key steps in hyperedge formation. We begin by extracting shape features from 3D object datasets using two well-established representation methods: Multi-view Convolutional Neural Network (MVCNN) (Su et al., 2015) and Group-View Convolutional Neural Network (GVCNN) (Feng et al., 2018). These feature embeddings are then used to define hypergraph structures. Specifically, we construct hyperedge indicator matrices $\mathbf{H}_1$ and $\mathbf{H}_2$ by applying the k-nearest neighbor (k-NN) algorithm in the MVCNN and GVCNN feature spaces, respectively. In the single-view setting, we utilize either $\mathbf{H}_1$ or $\mathbf{H}_2$ to build the hypergraph. In the multi-view setting, we concatenate both matrices to generate a unified hypergraph consisting

of $2N$ hyperedges. This construction process produces visual hypergraph data that can be directly utilized by the PEF-HNN model for node-level classification in visual object recognition tasks.

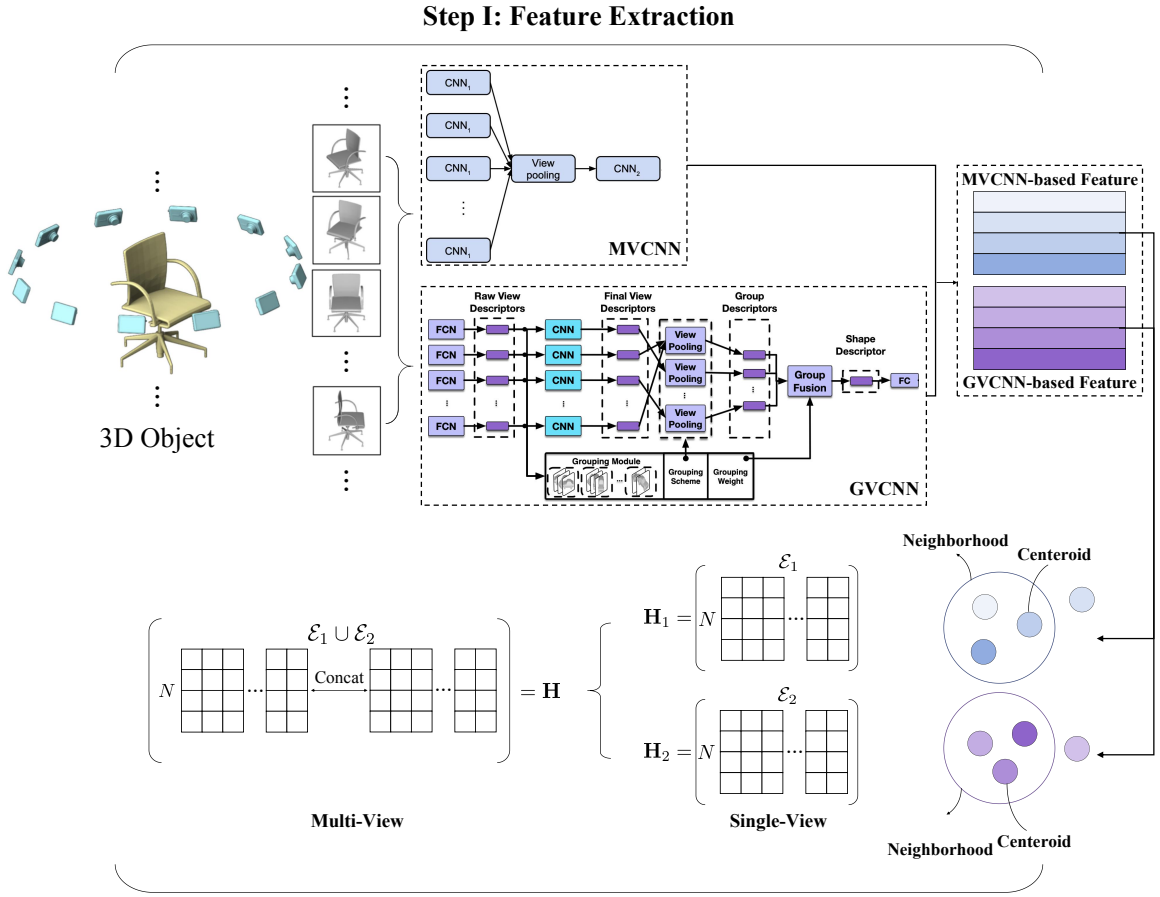

Figure 2: Hypergraph construction pipeline for visual object classification.

Table 6 details the reproducible hyperparameters used to obtain the visual object classification results reported in the main paper.

Table 6: Hyperparameter settings for ModelNet40 and NTU2012 datasets.

Dataset	Hyperparameter Setting	
ModelNet40	Learning rate: 2e-3 Weight decay: 1e-4 Hidden Size: 256 Layers: 4	Dropout ratio: 0.4 Alpha: 0.4 Beta: 0.7
NTU2012	Learning rate: 2e-3 Weight decay: 1e-4 Hidden Size: 512 Layers: 4	Dropout ratio: 0.2 Alpha: 0.5 Beta: 0.1

Further Ablation Study. Table 7 presents the ablation results on the ModelNet40 and NTU2012 datasets, providing empirical evidence for the importance of jointly incorporating low-pass and high-pass components

in the visual object classification task. The full model, which leverages both spectral components, achieves the highest accuracy on both datasets (98.42% on ModelNet40 and 91.40% on NTU2012). Removing the high-pass component leads to a noticeable drop in performance (-0.32% on ModelNet40 and -1.02% on NTU2012), indicating that high-frequency information contributes valuable discriminative features, particularly for heterogeneous structures. Similarly, excluding the low-pass component also degrades performance (-0.21% on ModelNet40 and -1.52% on NTU2012), highlighting the necessity of capturing smooth, semantically coherent patterns. These findings confirm that visual data typically involve both homophilic and heterophilic associations, for example, an airplane's fuselage and wings are structurally and semantically distinct yet semantically related within the object context. By jointly modeling information across frequency spectra, the proposed approach enables more adaptive and semantically enriched representations, ultimately enhancing classification accuracy.

Table 7: Ablation study on the contributions of low-pass and high-pass components.

	ModelNet40	**NTU2012**
Full model	$\mathbf{98.42 \pm 0.14}$	$\mathbf{91.40 \pm 1.33}$
w/o high pass	98.10 ± 0.15	90.38 ± 1.46
w/o low pass	98.21 ± 0.15	89.88 ± 1.26

References

Chen, D.-Y.; Tian, X.-P.; Shen, Y.-T.; and Ouhyoung, M. 2003. On visual similarity based 3D model retrieval. *Computer Graphics Forum*, 22(3): 223–232.

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

Feng, Y.; You, H.; Zhang, Z.; Ji, R.; and Gao, Y. 2019. Hypergraph neural networks. In *AAAI*, 3558–3565.

Feng, Y.; Zhang, Z.; Zhao, X.; Ji, R.; and Gao, Y. 2018. GVCNN: Group-view convolutional neural networks for 3D shape recognition. In *CVPR*, 264–272.

Fowler, J. H. 2006. Legislative cosponsorship networks in the US House and Senate. *Social Networks*, 28(4): 454–465.

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

Li, M.; Gu, Y.; Wang, Y.; Fang, Y.; Bai, L.; Zhuang, X.; and Lio, P. 2025. When hypergraph meets heterophily: New benchmark datasets and baseline. In *AAAI*, 18377–18384.

Su, H.; Maji, S.; Kalogerakis, E.; and Learned-Miller, E. 2015. Multi-view convolutional neural networks for 3D shape recognition. In *ICCV*, 945–953.

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

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3D ShapeNets: A deep representation for volumetric shapes. In *CVPR*, 1912–1920.

Yadati, N.; Nimishakavi, M.; Yadav, P.; Nitin, V.; Louis, A.; and Talukdar, P. 2019. HyperGCN: A new method for training graph convolutional networks on hypergraphs. In *NeurIPS*, 1511–1522.