Appendix for " When Hypergraph Meets Heterophily: New Benchmark Datasets and Baseline"

Authors: M. Li, Y. Gu, Y. Wang*, Y. Fang, L. Bai*, X. Zhuang, P. Lio

HHL: Heterophilic Hypergraph Learning

(Repository: https://kellysylvia77.github.io/HHL)

AAAI 2025

Appendix A: Additional Details for Experimental Implementation

Code and Dataset Availability

The newly developed heterophilic hypergraph benchmark datasets, along with the full experimental implementation code for HyperUFG, are available at https://kellysylvia77.github.io/HHL. These resources are provided to support reproducibility and further exploration of the results presented in this paper.

Existing Hypergraph Datasets

In addition to the introduction of four new benchmark hypergraphs, we utilize seven established hypergraph datasets to comprehensively assess the performance of HyperUFG. The details of these datasets are summarized in Table S-1. Our dataset selection includes widely-used co-citation networks (Cora, Citeseer, Pubmed) (Yadati et al. 2019), co-authorship networks (Cora-CA, DBLP-CA) (Yadati et al. 2019), as well as the U.S. Senate and House voting records (Fowler 2006). These datasets exhibit diverse scales and structures, although they predominantly exhibit homophilic relationships, making them well-suited for evaluating baseline performance in homophilic settings.

Table S-1: Overview of key statistics for several benchmark hypergraph datasets.

Datasets	Cora	Citeseer	Pubmed	Cora-CA	DBLP-CA	Senate	House
Hypernodes, $\lvert\mathcal{V}\rvert$	2, 708	3, 312	19, 717	2, 708	43, 413	282	1, 290
Hyperedges, $\lvert\mathcal{E}\rvert$	1, 579	1, 079	7, 963	1, 072	22, 535	315	340
Avg. hyperedge size	3.0 ± 1.1	3.2 ± 2.0	4.3 ± 5.7	4.2 ± 4.1	4.7 ± 6.1	17.2 ± 6.7	34.9 ± 21.4
Features, d	1, 433	3, 703	500	1, 433	1, 425	100	100
Classes, c	7	6	3	7	6	2	2
Node hom. ratio, $\mathcal{H}_{node}$	0.6399	0.5771	0.5499	0.7279	0.8557	0.4793	0.5049
Edge hom. ratio, $\mathcal{H}_{edge}$	0.7462	0.6814	0.7765	0.7797	0.8656	0.4642	0.4851

Baseline Models

We benchmark HyperUFG against the graph-agnostic method MLP and several existing hypergraph neural networks (HNNs).

- MLP: As a graph-agnostic model, MLP serves as a baseline for assessing the impact of incorporating relational learning in HNNs. Its performance provides a reference point for evaluating the advantages of HNNs.
- While models such as HNN (Feng et al. 2019), HyperGCN (Yadati et al. 2019), HyperND (Prokopchik, Benson, and Tudisco 2022), AllDeepSets, AllSetTransformer (Chien et al. 2022), and the UniGNN family (Huang and Yang 2021) (including UniGCN, UniSAGE, UniGAT, and UniGCNII) are widely recognized in the HNN domain, they primarily focus on datasets with homophilic relationships and pay little attention to heterophilic hypergraph datasets. As a result, their performance evaluations are often limited to homophilic scenarios.
- ED-HNN (Wang et al. 2023) and SheafHyperGNN (Duta et al. 2023) are recent models that have started to address the issue of heterophilic hypergraph learning. While they begin to touch on this challenge, their exploration remains limited, as the

Table S-2: Performance comparison between UniGNN models and HyperUFG on heterophilic hypergraphs. The best-performing model is highlighted inh lilac, the second-best in blue, and the third-best in grey.

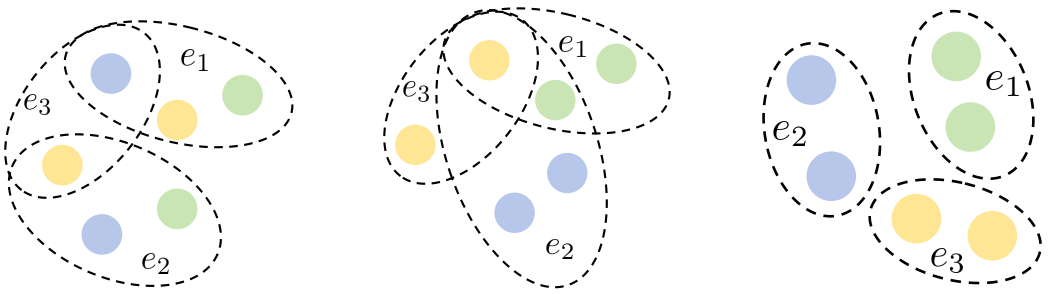
Datasets **Edge hom. ratio,** $\mathcal{H}_{edge}$	**Actor** 0.4675	**Amazon-ratings** 0.3677	**Twitch-gamers** 0.4857	**Pokec** 0.4529	**Senate** 0.4642	**House** 0.4851
MLP	85.45 ± 1.21	26.70 ± 2.82	52.77 ± 1.81	56.92 ± 2.46	52.25 ± 5.17	51.86 ± 2.34
UniGCN	70.55 ± 3.90	24.65 ± 1.88	51.87 ± 0.75	51.64 ± 0.87	50.56 ± 5.99	54.52 ± 2.91
UniSAGE	67.76 ± 3.59	26.38 ± 3.01	51.94 ± 0.61	54.19 ± 2.53	49.44 ± 5.43	53.53 ± 3.53
UniGAT	71.57 ± 2.14	26.39 ± 1.61	52.06 ± 0.35	51.32 ± 0.45	49.30 ± 5.00	56.04 ± 2.03
UniGCNII	80.48 ± 1.13	26.63 ± 1.32	50.84 ± 0.76	54.25 ± 2.70	49.30 ± 4.25	67.25 ± 2.57
HyperUFG	89.32 ± 0.75	40.53 ± 2.25	52.35 ± 0.04	62.30 ± 0.12	67.61 ± 7.00	72.82 ± 2.22

datasets used for evaluation are either small in scale or not fully representative of true heterophily, as discussed in the main manuscript.

Hyperparameters Selection

All baseline models are implemented using the PyTorch framework (Paszke et al. 2017), alongside the Torch-Geometric library for geometric deep learning (Fey and Lenssen 2019). Detailed experimental configurations and hyperparameter settings for each baseline model are provided to ensure the integrity and reproducibility of our results.

- **MLP**: learning rate in {1e-2, 1e-3, 2e-3, 3e-3}, weight decay in {1e-2, 1e-3, 1e-4, 2e-3, 2e-4, 5e-3, 5e-4}, hidden dimensions in {32, 64, 128, 256, 512}, layers number in {1, 2, 8, 16, 32, 64}, dropout in {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}.
- **HGNN**: learning rate in {1e-2, 1e-3, 2e-3, 3e-3}, weight decay in {1e-2, 1e-3, 1e-4, 2e-3, 2e-4, 5e-3, 5e-4}, hidden dimensions in {32, 64, 128, 256, 512}, dropout in {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}.
- **HyperGCN**: learning rate in {1e-2, 1e-3, 2e-3, 3e-3}, weight decay in {1e-2, 1e-3, 1e-4, 2e-3, 2e-4, 5e-3, 5e-4}, hidden dimensions in {32, 64, 128, 256, 512}, dropout in {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}.
- **HyperND**: learning rate in {1e-2, 1e-3, 2e-3, 3e-3}, weight decay in {1e-2, 1e-3, 1e-4, 2e-3, 2e-4, 5e-3, 5e-4}, hidden dimensions of MLP in {32, 64, 128, 256, 512}, hidden dimensions of Classifier in {32, 64, 128, 256, 512}, layers number in {1, 2, 8, 16, 32, 64}, restart alpha in {0.0, 0.5}, input dropout in {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}, dropout in {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}.
- **AllDeepSets**: learning rate in {1e-2, 1e-3, 2e-3, 3e-3}, weight decay in {1e-2, 1e-3, 1e-4, 2e-3, 2e-4, 5e-3, 5e-4}, hidden dimensions of MLP in {32, 64, 128, 256, 512}, hidden dimensions of Classifier in {32, 64, 128, 256, 512}, All layers number in {1, 2, 8, 16, 32, 64}, layers number of MLP in {1, 2, 8, 16, 32, 64}, layers number of MLP2 in {1, 2, 8, 16, 32, 64}, layers number of MLP3 in {1, 2, 8, 16, 32, 64}, layers number of Classifier in {1, 2, 8, 16, 32, 64}, restart alpha in {0.0, 0.5}, dropout in {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}.
- **AllSetTransformer**: learning rate in {1e-2, 1e-3, 2e-3, 3e-3}, weight decay in {1e-2, 1e-3, 1e-4, 2e-3, 2e-4, 5e-3, 5e-4}, hidden dimensions of MLP in {32, 64, 128, 256, 512}, hidden dimensions of Classifier in {32, 64, 128, 256, 512}, All layers number in {1, 2, 8, 16, 32, 64}, layers number of MLP in {1, 2, 8, 16, 32, 64}, layers number of MLP2 in {1, 2, 8, 16, 32, 64}, layers number of MLP3 in {1, 2, 8, 16, 32, 64}, layers number of Classifier in {1, 2, 8, 16, 32, 64}, restart alpha in {0.0, 0.5}, dropout in {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}.
- **UniGNNs (including UniGCN, UniSAGE, UniGAT and UniGCNII)**: learning rate in {1e-2, 1e-3, 2e-3, 3e-3}, weight decay in {1e-2, 1e-3, 1e-4, 2e-3, 2e-4, 5e-3, 5e-4}, lambda in {0.0, 0.5}, hidden dimensions in {32, 64, 128, 256, 512}, layers number in {1, 2, 8, 16, 32, 64}, dropout in {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}.
- **ED-HNN**: learning rate in {1e-2, 1e-3, 2e-3, 3e-3}, weight decay in {1e-2, 1e-3, 1e-4, 2e-3, 2e-4, 5e-3, 5e-4}, hidden dimensions of MLP in {32, 64, 128, 256, 512}, hidden dimensions of Classifier in {32, 64, 128, 256, 512}, All layers number in {1, 2, 8, 16, 32, 64}, layers number of MLP in {1, 2, 8, 16, 32, 64}, layers number of MLP2 in {1, 2, 8, 16, 32, 64}, layers number of MLP3 in {1, 2, 8, 16, 32, 64}, layers number of Classifier in {1, 2, 8, 16, 32, 64}, restart alpha in {0.0, 0.5}, dropout in {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}.



(a) $\mathcal{H}_{edge} = 0, \mathcal{H}_{node} = 0$ (b) $\mathcal{H}_{edge} = \frac{1}{2}, \mathcal{H}_{node} = \frac{11}{24}$ (c) $\mathcal{H}_{edge} = 1, \mathcal{H}_{node} = 1$

Fig. S-1: An intuitive illustration of toy hypergraphs with varying homophily/heterophily ratios.

Table S-3: Detailed statistics of the newly-developed heterophilic hypergraphs.

Datasets	**Actor**	**Amazon-ratings**	**Twitch-gamers**	**Pokec**
Hypernodes, $\lvert\mathcal{V}\rvert$	16, 255	22, 299	16, 812	14, 998
Hyperedges, $\lvert\mathcal{E}\rvert$	10, 164	2, 090	2, 627	2, 406
Avg. hyperedge size	5.43 ± 2.65	3.10 ± 0.62	6.23 ± 3.37	2.29 ± 0.65
Features, d	50	111	7	65
Classes, c	3	5	2	2
Proportions of each category	62:15:23	19:10:20:26:25	50:50	49:51
Node hom. ratio, $\mathcal{H}_{node}$	0.4815	0.4805	0.4893	0.4952
Edge hom. ratio, $\mathcal{H}_{edge}$	0.4675	0.3677	0.4857	0.4529

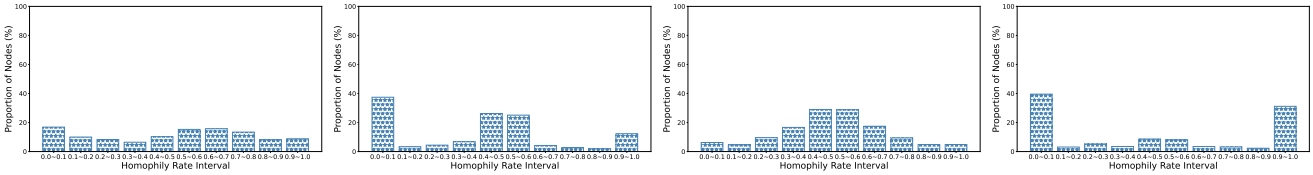
- **SheafHyperGNN**: learning rate in {1e-2, 1e-3, 2e-3, 3e-3}, weight decay in {1e-2, 1e-3, 1e-4, 2e-3, 2e-4, 5e-3, 5e-4}, hidden dimensions of MLP in {32, 64, 128, 256, 512}, hidden dimensions of Classifier in {32, 64, 128, 256, 512}, All layers number in {1, 2, 8, 16, 32, 64}, layers number of MLP in {1, 2, 8, 16, 32, 64}, layers number of MLP2 in {1, 2, 8, 16, 32, 64}, layers number of MLP3 in {1, 2, 8, 16, 32, 64}, layers number of Classifier in {1, 2, 8, 16, 32, 64}, dropout in {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}.
- **HyperUFG**: learning rate in {1e-2, 1e-3, 2e-3, 3e-3}, weight decay in {1e-2, 1e-3, 1e-4, 2e-3, 2e-4, 5e-3, 5e-4}, alpha in {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}, gamma in {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}, lambda in {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}, Lev in {1, 2, 3, 4, 5}, hidden dimensions in {32, 64, 128, 256, 512}, layers number in {1, 2, 8, 16, 32, 64}, dropout in {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}.

Appendix B: Additional Experimental Results

The comparative performance of HyperUFG against the UniGNN family is presented in Table S-2. The results clearly demonstrate that HyperUFG consistently outperforms all UniGNN variants. Additionally, we observe that MLP achieves superior performance compared to the UniGNN models. This finding aligns with the observations in the main manuscript, where we discuss the limitations of UniGNNs in handling heterophilic data. Specifically, UniGNNs struggle with heterophily, which results in their performance being surpassed by MLP in such scenarios.

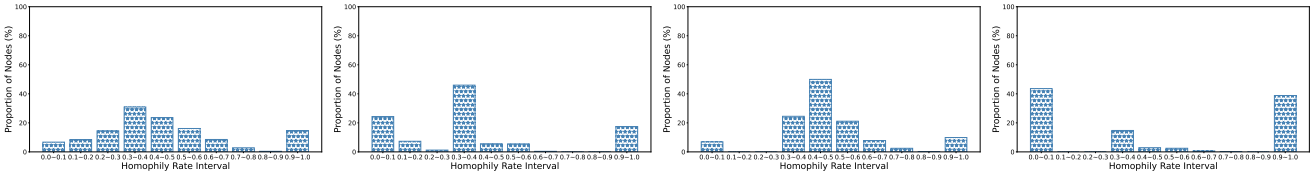
Appendix C: Further Clarification on Homophily Measures

To provide a clearer understanding of homophily in hypergraphs, we present several toy schematic diagrams that illustrate how hyperedges and nodes change characteristics under varying homophily ratios (see Fig. S-1). In particular, Fig. S-1 (a) and Fig. S-1 (c) demonstrate that the homophily measures from both perspectives can accurately capture the extremes of complete homophily and complete heterophily, as we intuitively expect in hypergraphs. In more intermediate cases, slight differences between the two homophily measures can emerge, as illustrated in Fig. S-1 (b).



(a) Actor (b) Amazon-ratings (c) Twitch-gamers (d) Pokec

Fig. S-2: Distribution of node homophily ratios across four newly-developed heterophilic hypergraphs. The horizontal axis represents intervals of homophily ratios, while the vertical axis indicates the frequency of nodes in each interval.



(a) Actor (b) Amazon-ratings (c) Twitch-gamers (d) Pokec

Fig. S-3: Distribution of hyperedge homophily ratios across four newly-developed heterophilic hypergraphs. The horizontal axis represents intervals of homophily ratios, while the vertical axis indicates the frequency of hyperedges in each interval.

Appendix D: Additional Datasets Statistics

Table S-3 provides additional statistical information on the proposed datasets, including the distribution of nodes across different classes. These statistics indicate that the datasets are not only relatively larger in scale compared to the Senate and House datasets presented in Table S-1, but also mostly class-balanced, offering favorable support for validating research on heterophilic hypergraph learning.

To further explore the relationship between homophily and heterophily in hypergraph datasets, we visualize the distribution of node and hyperedge homophily using the two proposed homophily quantification methods. These visualizations are presented in Figs. S-2 and S-3. From the results, it is evident that the majority of nodes and hyperedges in the four benchmark datasets have a homophily rate of less than 0.5, indicating that these hypergraphs are predominantly heterophilic. This poses a significant challenge for the design of effective hypergraph filtering methods.

Appendix E: Theoretical Properties of Hypergraph Framelet System

As highlighted in Section 4 (in the main manuscript), HyperUFG integrates both low-pass and high-pass filters within its hypergraph convolution layer, effectively addressing the limitations of spatial-based message passing that may impede effective neighbor aggregation in heterophilic hypergraph learning scenarios. Consistent with the principles that guide the development of GNNs for heterophilic graphs (Bo et al. 2021; Li, Pan, and Kang 2024; Huang et al. 2024), we believe that low-pass filters primarily capture similarities among homophilic neighbors, whereas high-pass filters emphasize extracting differentiating information among heterophilic neighbors. In this section, we further explore the theoretical properties of hypergraph framelets to enhance understanding of how framelet-based decomposition and reconstruction operators function when applied to hypergraph signals. Specifically, we investigate how the collaborative contribution of low-pass and high-pass components aids in the effective processing of hypergraph signals, providing deeper insights into their roles in signal representation and reconstruction.

Recalling the notations introduced in Section 4, consider a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a vertex set $\mathcal{V}$ containing $N = |\mathcal{V}|$ vertices. Let $\mathbf{U} = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_N]$ represent the matrix of eigenvectors of the hypergraph Laplacian $\mathcal{L}_h$, and let $\Lambda = \text{diag}(\lambda_1, \cdots, \lambda_N)$ be the diagonal matrix of the eigenvalues. We define a set of scaling functions $\xi = \{\delta; \eta^{(1)}, \cdots, \eta^{(k)}\}$ associated with a *filter bank* $\rho = \{a; b^{(1)}, \cdots b^{(k)}\}$. These functions satisfy $\widehat{\delta}(2\vartheta) = \widehat{a}(\vartheta)\widehat{\delta}(\vartheta)$ and $\widehat{\eta^{(r)}}(2\vartheta) = \widehat{b^{(r)}}(\vartheta)\widehat{\delta}(\vartheta)$ for any $\vartheta \in \mathbb{R}$, where $\widehat{f}(\vartheta)$ denotes the Fourier transform of f. Additionally, the functions $\Phi_{j,p}(\nu)$ and $\Psi^r_{j,p}(\nu)$ represent the low-pass and high-pass framelets, respectively, at node ν and are associated with node t at scale level $j \in \{1, \cdots, J\}$. These functions are defined as follows (for clarity, we copy and paste the Equations (3) and (4) from Section 4 of the main manuscript

here for reference):

$$\text{Low-pass: } \Phi_{j,t}(\nu) = \sum_{p=1}^{N} \widehat{\delta}\left(\frac{\lambda_p}{2^j}\right) u_p(t)u_p(\nu), \tag{A-1}$$

$$\text{High-pass: } \Psi^r_{j,t}(\nu) = \sum_{p=1}^{N} \widehat{\eta^{(r)}}\left(\frac{\lambda_p}{2^j}\right) u_p(t)u_p(\nu),\ r = 1, \ldots, k, \tag{A-2}$$

where $u_p(t)$ represents the t-th component of the eigenvector $\boldsymbol{u}_p$.

For two integers J, J_1 such that $J > J_1$, we define a **hypergraph framelet system** (denoted as $\mathcal{HFS}(\xi, \rho; \mathcal{G})$), starting from a scale J_1, as a non-homogeneous, stationary affine system:

$$\mathcal{HFS}^J_{J_1}(\xi, \rho; \mathcal{G}) = \{\Phi_{J_1,t} : t \in \mathcal{V}\} \cup \{\Psi^r_{j,t} : t \in \mathcal{V}, j = J_1, \ldots, J\}^k_{r=1}. \tag{A-3}$$

The system $\mathcal{HFS}^J_{J_1}(\xi, \rho; \mathcal{G})$ is referred to as a hypergraph tight frame for $l_2(\mathcal{G})$, and its elements are called hypergraph framelets on are called hypergraph framelets on $\mathcal{G}$.

Theorem S1 (Equivalence of Hypergraph Framelet Tightness). Let $J \geq 1$ be an integer, and consider the hypergraph framelet system $\mathcal{HFS}^J_{J_1}(\xi, \rho; \mathcal{G})$, where $J_1 = 1, \ldots, J$, as defined in(A-3), with hypergraph framelets $\Phi_{j,t}$ and $\Psi^r_{j,t}$. Then, the following statements are equivalent:

(i) For each $J_1 = 1, \ldots, J$, the hypergraph framelet system $\mathcal{HFS}^J_{J_1}(\xi, \rho; \mathcal{G})$ is a tight frame for $l_2(\mathcal{G})$, that is, $\forall f \in l_2(\mathcal{G})$,

$$\|f\|^2 = \sum_{t\in\mathcal{V}} \left| \langle f, \Phi_{J_1,t}\rangle \right|^2 + \sum_{j=J_1}^{J} \sum_{r=1}^{k} \sum_{t\in\mathcal{V}} \left| \langle f, \Psi^r_{j,t}\rangle \right|^2. \tag{A-4}$$

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

$$f = \sum_{t\in\mathcal{V}} \langle f, \Phi_{J,t}\rangle \Phi_{J,t} + \sum_{r=1}^{k} \sum_{t\in\mathcal{V}} \langle f, \Psi^r_{J,t}\rangle \Psi^r_{J,t}, \tag{A-5}$$

$$\sum_{t\in\mathcal{V}} \langle f, \Phi_{j+1,t}\rangle \Phi_{j+1,t} = \sum_{t\in\mathcal{V}} \langle f, \Phi_{j,t}\rangle \Phi_{j,t} + \sum_{r=1}^{k} \sum_{t\in\mathcal{V}} \langle f, \Psi^r_{j,t}\rangle \Psi^r_{j,t}. \tag{A-6}$$

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

$$\|f\|^2 = \sum_{t\in\mathcal{V}} \left|\langle f, \Phi_{J,t}\rangle\right|^2 + \sum_{r=1}^{k} \sum_{t\in\mathcal{V}} \left|\langle f, \Psi^r_{J,t}\rangle\right|^2, \tag{A-7}$$

$$\sum_{t\in\mathcal{V}} \left|\langle f, \Phi_{j+1,t}\rangle\right|^2 = \sum_{t\in\mathcal{V}} \left|\langle f, \Phi_{j,t}\rangle\right|^2 + \sum_{r=1}^{k} \sum_{t\in\mathcal{V}} \left|\langle f, \Psi^r_{j,t}\rangle\right|^2. \tag{A-8}$$

(iv) The functions in ξ satisfy

$$1 = \left|\widehat{\delta}\left(\frac{\lambda_p}{2^J}\right)\right|^2 + \sum_{r=1}^{k} \left|\widehat{\eta^{(r)}}\left(\frac{\lambda_p}{2^J}\right)\right|^2 \quad \forall p = 1, \ldots, N, \tag{A-9}$$

$$\left|\widehat{\delta}\left(\frac{\lambda_p}{2^{j+1}}\right)\right|^2 = \left|\widehat{\delta}\left(\frac{\lambda_p}{2^j}\right)\right|^2 + \sum_{r=1}^{k} \left|\widehat{\eta^{(r)}}\left(\frac{\lambda_p}{2^j}\right)\right|^2 \quad \forall \begin{array}{l} p = 1, \ldots, N, \\ j = 1, \ldots, J-1. \end{array} \tag{A-10}$$

(v) The identities in (A-9) hold and the filters in the filter bank ρ satisfy

$$\left|\widehat{a}\left(\frac{\lambda_p}{2^j}\right)\right|^2+\sum_{r=1}^{k}\left|\widehat{b^{(r)}}\left(\frac{\lambda_p}{2^j}\right)\right|^2=1\quad\forall p\in\sigma_\delta^{(j)},\ j=2,\ldots,J, \tag{A-11}$$

with

$$\sigma_\delta^{(j)}:=\left\{p\in\{1,\ldots,N\}:\widehat{\delta}\left(\frac{\lambda_p}{2^j}\right)\neq 0\right\}.$$

Proof: (i)$\Longleftrightarrow$(ii). Let $\phi_j:=\operatorname{span}\{\Phi_{j,t}:t\in\mathcal{V}\}$ and $\psi_j^r:=\operatorname{span}\{\Psi_{j,t}^r:t\in\mathcal{V}\}$. Define projections $\mathbf{P}\phi_j,\mathbf{P}\psi_j^r,r=1,\ldots,k$ by

$$\mathbf{P}\phi_j(f):=\sum_{t\in\mathcal{V}}\langle f,\Phi_{j,t}\rangle\,\Phi_{j,t},\quad \mathbf{P}\psi_j^r(f):=\sum_{t\in\mathcal{V}}\left\langle f,\Psi_{j,t}^r\right\rangle\Psi_{j,t}^r,\quad f\in l_2(\mathcal{G}). \tag{A-12}$$

Since $\mathcal{HFS}_{J_1}^J(\xi,\rho)$ is a hypergraph tight frame for $l_2(\mathcal{G})$ for $J_1=1,\ldots,J$, we obtain by polarization identity,

$$f=\mathbf{P}\phi_{J_1}(f)+\sum_{j=J_1}^{J}\sum_{r=1}^{k}\mathbf{P}\psi_j^r(f)=\mathbf{P}\phi_{J_1+1}(f)+\sum_{j=J_1+1}^{J}\sum_{r=1}^{k}\mathbf{P}\psi_j^r(f) \tag{A-13}$$

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

$$\mathbf{P}\phi_{J_1+1}(f)=\mathbf{P}\phi_{J_1}(f)+\sum_{r=1}^{k}\psi_j^r(f), \tag{A-14}$$

which is (A-6). Moreover, when $J_1=J$, (A-13) gives (A-5). Consequently, (i)$\Longrightarrow$(ii). Conversely, recursively using (A-14) gives

$$\mathbf{P}\phi_{m+1}(f)=\mathbf{P}\phi_{J_1}(f)+\sum_{j=J_1}^{m}\sum_{r=1}^{k}\mathbf{P}\psi_j^r(f) \tag{A-15}$$

for all $J_1\leq m\leq J-1$. Taking $m=J-1$ together with (A-5), we deduce (A-13), which is equivalent to (A-4). 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 $\boldsymbol{u}_p$,

$$\langle f,\Phi_{j,t}\rangle=\sum_{p=1}^{N}\widehat{\delta}\left(\frac{\lambda_p}{2^j}\right)\widehat{f}_p\,u_p(t),\quad \left\langle f,\Psi_{j,t}^r\right\rangle=\sum_{p=1}^{N}\widehat{\eta^{(r)}}\left(\frac{\lambda_p}{2^j}\right)\widehat{f}_p\,u_p(t),$$

where $\widehat{f}_p=\langle f,\boldsymbol{u}_p\rangle$ is the Fourier coefficient of f with respect to $\boldsymbol{u}_p$. This together with (A-12), (A-1) and (A-2) gives, for $j\geq 1$ and $r=1,\ldots,k$, the Fourier coefficients for the projections $\mathbf{P}\phi_j(f)$ and $\mathbf{P}\psi_j^r(f)$:

$$\left(\widehat{\mathbf{P}\phi_j(f)}\right)_p=\left|\widehat{\delta}\left(\frac{\lambda_p}{2^j}\right)\right|^2\widehat{f}_p,\quad \left(\widehat{\mathbf{P}\psi_j^r(f)}\right)_p=\left|\widehat{\eta^{(r)}}\left(\frac{\lambda_p}{2^j}\right)\right|^2\widehat{f}_p,\quad\forall p=1,\ldots,N, \tag{A-16}$$

which implies that (A-5) and (A-6) are equivalent to (A-9) and (A-10) respectively. Thus, (ii)$\Longleftrightarrow$(iv).

(iv)$\Longleftrightarrow$(v). Based on the relations that $\widehat{\delta}(2\vartheta)=\widehat{a}(\vartheta)\widehat{\delta}(\vartheta)$ and $\widehat{\eta^{(r)}}(2\vartheta)=\widehat{b^{(r)}}(\vartheta)\widehat{\delta}(\vartheta)$ for any $\vartheta\in\mathbb{R}$, it can be deduced that for $p=1,\ldots,N$ and $j\geq 1$,

$$\left|\widehat{\delta}\left(\frac{\lambda_p}{2^j}\right)\right|^2+\sum_{r=1}^{k}\left|\widehat{\eta^{(r)}}\left(\frac{\lambda_p}{2^j}\right)\right|^2=\left(\left|\widehat{a}\left(\frac{\lambda_p}{2^{j+1}}\right)\right|^2+\sum_{r=1}^{k}\left|\widehat{b^{(r)}}\left(\frac{\lambda_p}{2^{j+1}}\right)\right|^2\right)\left|\widehat{\delta}\left(\frac{\lambda_p}{2^{j+1}}\right)\right|^2.$$

This shows that (A-10) is equivalent to (A-11). Therefore, (iv)$\Longleftrightarrow$(v).

Table S-4: Summary of training computational complexity for UniGCNII, Deep-HGNN, AllDeepSets, ED-HNN, and our proposed HyperUFG model. N denotes the number of nodes in the given hypergraph, M represents the number of hyperedges, M' is the number of edges in the clique expansion (when transforming the hypergraph into a regular graph), $\|\mathbf{H}\|_0$ refers to the number of non-zero values in the incidence matrix $\mathbf{H}$, T is the number of training epochs, L is the number of layers, d refers to the feature dimension, k represents the number of high-pass filters in HyperUFG, J is the scale level in HyperUFG, and K denotes the largest number of non-zero values in the framelet transform matrices $\mathcal{W}r, j$.

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

Appendix F: Computational Complexity Analysis

This appendix provides an analysis of the computational complexity associated with training four state-of-the-art hypergraph neural networks and our proposed HyperUFG model. A summary of the estimated training complexities is presented in Table S-4.

As shown in Table S-4, n, J, K are constants that are independent of the specific hypergraph. In practical scenarios, n and J usually take on relatively small values. On the other hand,thanks to the sparsity of the constructed hypergraph framelets, K tends to be small and may even be less than or approximately equal to $\|\mathbf{H}\|_0$. Consequently, HyperUFG maintains competitive performance without incurring significant additional computational overhead compared to existing models. Specifically, the computational complexity of HyperUFG is comparable to that of models such as AllDeepSets (Chien et al. 2022) and ED-HNN (Wang et al. 2023).

Appendix G: Exploratory Thoughts on Open Problems in HHL

As discussed in Section 2.1 (in the main manuscript), to the best of our knowledge, only ED-HNN (Wang et al. 2023) and SheafHyperGNN (Duta et al. 2023) briefly mention the issue of heterophilic hypergraph learning (that is why a related work section is absent from the main manuscript), albeit without a focused examination of the most direct challenges such as evaluation metrics and data sources. This gap underscores the importance of our work. Our study represents an initial effort in the promising field of heterophilic hypergraph learning. To inspire further research in this emerging area, we propose several open problems, some of which are currently under exploration within our research group:

- Developing new metrics and conducting an in-depth investigation of the various characteristics of these metrics, particularly their effectiveness and/or distinctiveness in accurately evaluating the homophily/heterophily of a given hypergraph. Key questions include: "*What is missing in our current understanding of homophily metrics for hypergraph?", "Can an effective bridge be established between metrics for graphs and those for hypergraphs?"* Lessons might possibly be drawn from recent advances in (Telyatnikov et al. 2023; Zheng, Luan, and Chen 2024; Luan et al. 2024);
- Conducting a thorough theoretical analysis of the key factors influencing the performance of heterophilic HNNs, including investigating the expressive power, universal approximation ability, and generalization capability of specifically designed HNN architectures.
- Formulating formally and developing methods for modeling dynamic hypergraphs that exhibit heterophily, or more broadly, enabling continue/lifelong/incremental learning on heterophilic hypergraphs;
- Extending heterophilic graph and hypergraph learning to more general topological domains (Hajij et al. 2022), such as simplicial complexes, cell complexes, and combinatorial complexes;

In addition to these challenges, we call for further research on creating more complex and diverse datasets, developing more advanced HNNs and open-source library/toolkits for heterophilic hypergraph learning, and exploring advanced applications in both scientific and industrial contexts (Heydaribeni et al. 2024; Gao et al. 2024).

References Mentioned in Appendix

Bo, D.; Wang, X.; Shi, C.; and Shen, H. 2021. Beyond low-frequency information in graph convolutional networks. In *AAAI*, 3950–3957.

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*.

Duta, I.; Cassarà, G.; Silvestri, F.; and Liò, P. 2023. Sheaf hypergraph networks. In *NeurIPS*, 12087–12099.

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

Fey, M.; and Lenssen, J. E. 2019. Fast graph representation learning with PyTorch Geometric. In *RLGM Workshop at ICLR*.

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

Gao, Y.; Lu, J.; Li, S.; Li, Y.; and Du, S. 2024. Hypergraph-Based Multi-View Action Recognition Using Event Cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(10): 6610–6622.

Hajij, M.; Zamzmi, G.; Papamarkou, T.; Miolane, N.; Guzmán-Sáenz, A.; Ramamurthy, K. N.; Birdal, T.; Dey, T. K.; Mukherjee, S.; Samaga, S. N.; et al. 2022. Topological deep learning: Going beyond graph data. *arXiv preprint arXiv:2206.00606*.

Heydaribeni, N.; Zhan, X.; Zhang, R.; Eliassi-Rad, T.; and Koushanfar, F. 2024. Distributed constrained combinatorial optimization leveraging hypergraph neural networks. *Nature Machine Intelligence*, 6: 664–672.

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

Huang, K.; Wang, Y. G.; Li, M.; et al. 2024. How universal polynomial bases enhance spectral graph neural networks: Heterophily, over-smoothing, and over-squashing. In *ICML*.

Li, B.; Pan, E.; and Kang, Z. 2024. PC-Conv: Unifying homophily and heterophily with two-fold filtering. In *AAAI*, 13437–13445.

Luan, S.; Lu, Q.; Hua, C.; Wang, X.; Zhu, J.; Chang, X.-W.; Wolf, G.; and Tang, J. 2024. Are heterophily-specific gnns and homophily metrics really effective? evaluation pitfalls and new benchmarks. *arXiv preprint arXiv:2409.05755*.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch. In *Autodiff Workshop at NeurIPS*.

Prokopchik, K.; Benson, A. R.; and Tudisco, F. 2022. Nonlinear feature diffusion on hypergraphs. In *ICML*, 17945–17958.

Telyatnikov, L.; Bucarelli, M. S.; Bernardez, G.; Zaghen, O.; Scardapane, S.; and Lio, P. 2023. Hypergraph neural networks through the lens of message passing: a common perspective to homophily and architecture design. *arXiv preprint arXiv:2310.07684*.

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

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.

Zheng, Y.; Luan, S.; and Chen, L. 2024. What is missing in homophily? disentangling graph homophily for graph neural networks. *arXiv preprint arXiv:2406.18854*.