Deep Hypergraph Neural Networks with Tight Framelets

Ming Li[1,2], Yujie Fang[3,1], Yi Wang[3,1*], Han Feng[4], Yongchun Gu[1], Lu Bai[5*], Pietro Lio[6]

[1]Zhejiang Key Laboratory of Intelligent Education Technology and Application, Zhejiang Normal University
[2]Zhejiang Institute of Optoelectronics
[3]School of Computer Science and Technology, Zhejiang Normal University
[4]Department of Mathematics, City University of Hong Kong
[5]School of Artificial Intelligence, and Engineering Research Center of Intelligent Technology and Educational Application, Ministry of Education, Beijing Normal University
[6]Department of Computer Science and Technology, Cambridge University
{mingli, yjfang, wangyi, guyongchun}@zjnu.edu.cn, hanfeng@cityu.edu.hk, bailu@bnu.edu.cn, pl219@cam.ac.uk

Abstract

Hypergraphs provide a flexible framework for modeling high-order (complex) interactions among multiple entities, extending beyond traditional pairwise correlations in graph structures. However, deep hypergraph neural networks (HGNNs) often face the challenge of oversmoothing with increasing depth, similar to issues in graph neural networks (GNNs). While oversmoothing in GNNs has been extensively studied, its implications in relation to hypergraphs are less explored. This paper addresses this gap by first theoretically exploring the reasons behind oversmoothing in deep HGNNs. Our novel insights suggest that a spectral-based hypergraph convolution, equipped with both low-pass and high-pass filters, can potentially mitigate these effects. Motivated by these findings, we introduce FrameHGNN, a framework that utilizes framelet-based hypergraph convolutions integrating tight framelet transforms with both low-pass and high-pass components, as well as the commonly used strategies in designing deep GNN architecture: initial residual and identity mappings. The experiment results on diverse benchmark datasets demonstrate that FrameHGNN outperforms several state-of-the-art models, effectively reducing oversmoothing while improving predictive accuracy. Our contributions not only advance the theoretical understanding of deep hypergraph learning but also provide a practical spectral-based approach for HGNNs, emphasizing the design of multifrequency channels.

Appendix — https://mingli-ai.github.io/FrameHGNN.pdf

1 Introduction

Hypergraphs offer a favorable framework for modeling high-order interactions among multiple entities, surpassing the capabilities of traditional graphs which are limited to pairwise correlations. This flexibility makes hypergraphs particularly well-suited for representing complex relationships in diverse real-world datasets. Recently, hypergraphs have gained significant attention in various fields, including recommendation systems (Li et al. 2024a; Xia, Huang, and Zhang 2022), social network analysis (Sun et al. 2023), cross-modal retrieval (Li et al. 2024b), and bioinformatics (Murgas, Saucan, and Sandhu 2022). Their broad applicability underscores the value of hypergraphs in providing richer and more detailed data representations.

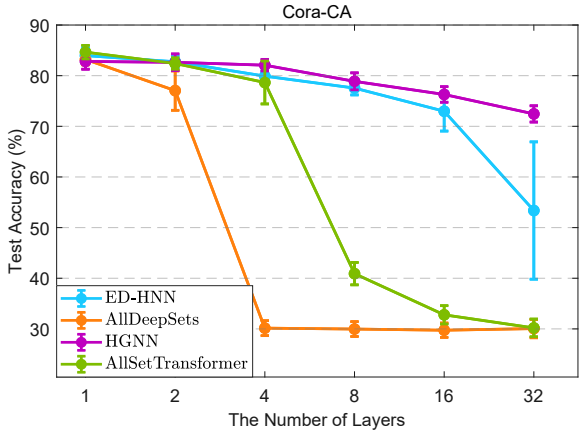

Figure 1: Illustration of oversmoothing effects in various classic HGNNs on the Cora-CA dataset.

Most existing hypergraph neural networks (HGNNs), however, suffer from the issue of oversmoothing, a challenge also well-recognized in graph neural networks (Rusch, Bronstein, and Mishra 2023). As the number of network layers increases, node representations in the hypergraph tend to become homogenized, leading to a significant decline in performance. Figure 1 illustrates this phenomenon, showing how the performance of four classic HGNN models, i.e., vanilla HGNN (Feng et al. 2019), ED-HNN (Wang et al. 2023), AllDeepSets and AllSetTransformer (Chien et al. 2022), degrades as the number of layers increases, with each model experiencing varying degrees of performance deterioration. While the problem of oversmoothing in GNNs has been extensively studied (Xu et al. 2018; Zhao and Akoglu 2020; Chen et al. 2020), there has been limited research specifically addressing why oversmoothing occurs in hypergraphs and how it can be prevented (Chen et al. 2022). To fill this gap, we offer a new theoretical perspective on why deep HGNNs are prone to oversmoothing, and then, motivated by these theoretical insights, explore how to develop deep HGNNs that effectively alleviate the oversmoothing problem.

In this paper, we introduce a framework for construct-

*Corresponding authors: Yi Wang, Lu Bai

ing deep HGNNs using tight framelets on hypergraphs. We begin by theoretically investigating the underlying reasons behind the oversmoothing issue in deep HGNNs. Our theoretical findings, supported by rigorous mathematical proofs, reveal that both low-pass and high-pass components of the hypergraph signal are integral to the hypergraph learning process. This insight motivates us to develop a spectral-based hypergraph convolution with multifrequency channels. To achieve this, we introduce framelet transforms on hypergraphs for both decomposition and reconstruction, resulting in a novel framelet-based hypergraph convolution layer that incorporates both low-pass and high-pass filters. We also theoretically establish the reversibility of frame decomposition and reconstruction, ensuring the stability of the proposed method. The use of Haar-type filters and approximated Chebyshev polynomials further enhances computational efficiency. Building on the techniques used in GCNII (Chen et al. 2020), specifically initial residual and identity mappings, we design deep HGNNs with multiple framelet-based hypergraph convolution layers, which we term FrameHGNN. Our experiments conducted on eight benchmark datasets demonstrate that FrameHGNN outperforms several baselines and shows significant potential in preventing oversmoothing.

The main contributions are summarized as follows:

- We conduct a theoretical study on the oversmoothing issue in deep HGNNs. Our analysis reveals that hypergraph convolution with both low-pass and high-pass filters is instrumental in mitigating oversmoothing, although the balance between these components is data-dependent when implemented practically on a given hypergraph dataset.
- Motivated by our theoretical findings, we develop FrameHGNN, a deep HGNN framework that leverages tight framelets for hypergraph convolutions. FrameHGNN integrates low-pass and high-pass components along with commonly used strategies in deep GNN architectures, such as initial residual and identity mappings.
- We perform extensive experiments on eight benchmark datasets, demonstrating the effectiveness of the proposed FrameHGNN model in comparison with state-of-the-art methods. These experiments also provide insights into the characteristics and challenges of different hypergraph datasets.

2 Related Work

Hypergraph Learning. The development of HGNNs significantly advances the field of hypergraph learning. HGNN (Feng et al. 2019) marks a key milestone by designing convolution operations based on the hypergraph Laplacian operator and truncated Chebyshev polynomials. HyperGCN (Yadati et al. 2019) is a novel approach for training graph convolutional networks on hypergraphs using tools from hypergraph spectral theory. HNHN (Dong, Sawin, and Bengio 2020) further enhances this area by applying nonlinear activation functions to update both vertex and hyperedge representations. HCHA (Bai, Zhang, and Torr 2021) incorporates hypergraph convolution and attention mechanisms, while HCoN (Wu, Yan, and Ng 2023) integrates information from vertices and hyperedges to achieve latent representation learning, preserving the original relationships between nodes and hyperedges. Additionally, ED-HNN (Wang et al. 2023) combines star expansions with standard message-passing neural networks, demonstrating effectiveness in handling heterogeneous hypergraph classification tasks.

Oversmoothing in GNNs. The issue of oversmoothing in GNNs is widely recognized and studied (Li, Han, and Wu 2018; Oono and Suzuki 2020; Chen et al. 2020; Rusch, Bronstein, and Mishra 2023). Researchers conduct theoretical analyses to understand the expressive power of GNNs and to develop strategies to mitigate oversmoothing. For instance, JKNet (Xu et al. 2018) introduces jump connections to enhance the adaptability of node representation learning. DropEdge (Rong et al. 2019) propose to reduce oversmoothing by randomly removing a proportion of edges from the input graph. PPNP and APPNP (Klicpera, Bojchevski, and Günnemann 2019) tackle oversmoothing by extending the neighborhood range of GCNs through an improved propagation scheme based on personalized PageRank. GRAND (Feng et al. 2020) improves model generalization by incorporating random propagation and consistency regularization, while PairNorm (Zhao and Akoglu 2020) introduces a novel normalization layer to address oversmoothing. DAGNN (Liu, Gao, and Ji 2020) decouples transformation and propagation in graph convolution operations, and GCNII (Chen et al. 2020) achieves strong performance in deep networks by integrating initial residuals and identity mapping into GCNs.

Oversmoothing in HGNNs. Addressing oversmoothing in HGNNs has also gained attention in recent research. UniGCNII (Huang and Yang 2021) employs two-stage aggregation and incorporates jumping knowledge from initial and previous features, effectively mitigating the oversmoothing issue in deep models. ED-HNN (Wang et al. 2023) suggests that equivariant operators distributing different messages across nodes help overcome oversmoothing, while parameter sharing across layers may reduce the risk of overfitting. DeepHGCN (Chen et al. 2022) analyzes the causes of oversmoothing from the perspectives of random walks and Dirichlet energy, applying GCNII's initial residuals and identity mapping to hypergraphs to effectively address the issue. A recent study (Tang, Chen, and Dong 2024) proposes the HypergraphMLP model, which, due to its lack of node feature aggregation during forward propagation, is inherently unaffected by the oversmoothing problem. Additionally, HDS^{ode} (Yan et al. 2024) introduces a control-diffusion hypergraph dynamic system based on ordinary differential equations (ODEs), enhancing the controllability of multi-layer HGNNs.

Remark. In contrast to existing studies, our research emphasizes the spectral properties of HGNNs and focuses on a theoretical investigation into the causes of oversmoothing in deep HGNNs. Motivated by these theoretical insights, we propose a spectral-based model specifically designed to tackle oversmoothing.

3 Theoretical Studies on Oversmoothing in Deep HGNNs

In this section, we provide details of the theoretical studies which explore the reasons behind the oversmoothing issue in

deep HGNNs. To begin, we revisit the fundamental concepts related to hypergraphs.

3.1 Notation and Preliminaries

A hypergraph is represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})^*$, comprising a vertex set $\mathcal{V}$ of size $N = |\mathcal{V}|$, a hyperedge set $\mathcal{E}$ of size $M = |\mathcal{E}|$. Suppose that vertices and hyperedges have feature dimensions d and m, respectively, we have the representation of vertex data as $\mathbf{X} \in \mathbb{R}^{N\times d}$ and hyperedge data as $\mathbf{Y} \in \mathbb{R}^{M\times o}$.

The hypergraph structure, from a vertex perspective, is defined by an incidence matrix $\mathbf{H} \in \{0,1\}^{N\times M}$ where $\mathbf{H}(v,e) = 1$ if vertex v is contained in hyperedge e, and 0 otherwise, as represented by:

$$\mathbf{H}(v,e) = \begin{cases} 1, & \text{if } v \in e; \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The degrees of vertex v and hyperedge e are denoted by diagonal matrices $\mathbf{D}_v \in \mathbb{R}^{N\times N}$ and $\mathbf{D}_e \in \mathbb{R}^{M\times M}$, calculated as $\sum_{e\in\mathcal{E}} \mathbf{H}(v,e)$ and $\sum_{v\in\mathcal{V}} \mathbf{H}(v,e)$, respectively.

In (Feng et al. 2019), the classic form of hypergraph convolution layer $f(\mathbf{X}, \mathbf{\Theta})$ is defined as the following formulation

$$\mathbf{X}^{(\ell+1)} = \sigma(\mathbf{F}\mathbf{X}^{(\ell)}\mathbf{\Theta}^{(\ell)}), \tag{2}$$

where $\mathbf{F} := \mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{D}_v^{-1/2}$, $\mathbf{X}^{(\ell)} \in \mathbb{R}^{N\times C}$ is the signal of the hypergraph at ℓ layer, $\mathbf{X}^{(0)} = \mathbf{X}$ and σ denotes the nonlinear activation function.

3.2 Our Theoretical Results

Let $\mathbf{h}_1$ be the normalization of $\mathbf{D}_v^{1/2}\mathbf{1}$ where $\mathbf{1} = (1,1,\ldots,1)^\top \in \mathbb{R}^N$. It is straightforward to verify that $\mathbf{F}\mathbf{h}_1 = \mathbf{h}_1$ and that all entries of $\mathbf{h}_1$ are positive. Therefore, we can identify $\mathbf{h}_1$ as the low-frequency eigenvector of $\mathbf{F}$. For any $x \in \mathbb{R}^N$, we define the projection operator $\mathbf{P}$ by

$$\mathbf{P}x := x - \langle x, \mathbf{h}_1\rangle \mathbf{h}_1,$$

which projects x onto the orthogonal complement space of $\mathbf{h}_1$ (the high-frequency subspace of $\mathbf{F}$). The high-frequency energy of x is then given by $\|\mathbf{P}x\|_2$.

The following theorem demonstrates that the classical hypergraph model, as described by Eq. (2), is susceptible to the over-smoothing problem, where the high-frequency energy tends to decrease towards zero as the number of layers increases.

Theorem 1. Let $\mathbf{X}^{(\ell)} \in \mathbb{R}^{N\times C}$ be the ℓth layer of a hypergraph as defined by (2) and $\mathbf{X}_c^{(\ell)} \in \mathbb{R}^N$ the feature at channel c, for $c = 1,\ldots,C$. If $\|\mathbf{\Theta}^{(\ell)}\|_{Fro} \le 1$, then the high frequency energy of $\mathbf{X}^{(\ell)}$ can be bounded by

$$\sum_{c=1}^{C} \|\mathbf{P}\mathbf{X}_c^{(\ell)}\|_2 \le |\lambda_{max}|^\ell \|\mathbf{X}^{(0)}\|_{Fro},$$

where $\|\cdot\|_{Fro}$ denotes the Frobenius norm, and λ_{max} is the largest high-frequency eigenvalue of $\mathbf{F}$. In addition, $\lambda_{max} < 1$.

*For simplicity, we do not consider the diagonal matrices indicating the weights of vertices and hyperedges

Proof. Recall that $\mathbf{F} := \mathbf{D}_\mathbf{v}^{-1/2}\mathbf{H}\mathbf{D}_\mathbf{e}^{-1}\mathbf{H}^\top\mathbf{D}_\mathbf{v}^{-1/2}$. For any $x \in \mathbb{R}^N$, letting $z = \mathbf{D}_v^{-1/2}x$, then

$$x^\top\mathbf{F}x = z^\top\mathbf{H}\mathbf{D}_e^{-1}\mathbf{H}^\top z = \sum_j \frac{1}{\sum_i \mathbf{H}_{ij}} \left(\sum_i \mathbf{H}_{ij} z_i\right)^2$$

and $x^\top x = \sum_i \left(\sum_j \mathbf{H}_{ij}\right) z_i^2 = \sum_j \left(\sum_i \mathbf{H}_{ij} z_i^2\right)$. Note that $H_{ij} \in \{0,1\}$ and for any sequence $\{a_n\}_{n=1}^m$

$$\begin{aligned}\left(\sum_{n=1}^m a_n\right)^2 &= \sum_n a_n^2 + \sum_{i\neq j} a_i a_j \le \sum_n a_n^2 + \sum_{j=1}^m \sum_{i\neq j} a_i^2 \\ &= \sum_n a_n^2 + \sum_{i=1}^m \sum_{j\neq i} a_i^2 = m\sum_n a_n^2.\end{aligned}$$

Moreover, the inequivalent holds if and only if all a_n's take a same value. This implies that

$$x^\top\mathbf{F}x \le x^\top x, \quad \forall x \in \mathbb{R}^N.$$

Thus, the eigenvalues of $\mathbf{F}$ are located in $[0,1]$ and $\lambda_{max} < 1$.

For any $x \in \mathbb{R}^N$, let $x_+ := \sigma(x), \quad x_- := x_+ - x$. We first claim that for any

$$\|\mathbf{P}x\|_2^2 \ge \|\mathbf{P}x_+\|_2^2 = \|\mathbf{P}\sigma(x)\|_2^2, \quad x \in \mathbb{R}^N. \tag{3}$$

To see it, without loss of generality, we assume $\|\mathbf{P}x_+\|_2 \neq 0$, otherise, (3) holds true trivially. Let $\{v_1, v_2, \ldots, v_{N-1}\} \subset \{\mathbf{h}_1\}^\perp$ be an orthonormal basis, where

$$v_1 := \frac{\mathbf{P}x_+}{\|\mathbf{P}x_+\|_2} = \frac{x_+ - \langle x_+, \mathbf{h}_1\rangle \mathbf{h}_1}{\|\mathbf{P}x_+\|_2}.$$

One can verify that $\langle x_+, v_1\rangle = \|\mathbf{P}x_+\|_2 \ge 0$ and

$$\langle x_-, v_1\rangle = -\frac{\langle x_+, \mathbf{h}_1\rangle \langle x_-, \mathbf{h}_1\rangle}{\|\mathbf{P}x_+\|_2} \le 0.$$

This yields that

$$\begin{aligned}\|\mathbf{P}\sigma(x)\|_2 &= |\langle x_+, v_1\rangle| \le |\langle x_+, v_1\rangle - \langle x_-, v_1\rangle| \\ &= |\langle x, v_1\rangle| \le \|\mathbf{P}x\|_2.\end{aligned}$$

Then by using (3) for all channels,

$$\begin{aligned}\sum_{c=1}^C \left\|\mathbf{P}\mathbf{X}_c^{(\ell)}\right\|_2 &\le \sum_{c=1}^C \left\|\mathbf{P}\mathbf{F}\mathbf{X}_c^{(\ell-1)}\right\|_2 \\ &\le \lambda_{\max} \left\|\mathbf{P}\mathbf{X}^{(\ell-1)}\right\|_{Fro},\end{aligned}$$

which by induction implies that

$$\left\|\mathbf{P}\mathbf{X}^{(\ell)}\right\|_{Fro} \le \lambda_{max}^\ell \left\|\mathbf{X}^{(0)}\right\|_{Fro}.$$

Remark on Theoretical Insights. Our theoretical findings demonstrate that when $\|\mathbf{P}x\|_2 = 0$, the resulting feature vectors in a deep GCN collapse into a single one-dimensional space, represented as $x = \gamma\mathbf{D}_v^{1/2}\mathbf{1}$ 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 propose the use of more sophisticated filters that perform convolutional operations on a channel-by-channel basis while capturing both low-pass and high-pass features. This approach is motivated by the need to preserve a richer set of features throughout the network layers, enhancing the model's ability to maintain the distinguishability of nodes and providing a more nuanced understanding of the underlying data. In the next section, we introduce FrameHGNN, which implements these concepts through framelet-based hypergraph convolutions, integrating tight framelet transforms with both low-pass and high-pass components.

4 FrameHGNN: Tackling Oversmoothing

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 $\Lambda = \text{diag}(\lambda_1, \cdots, \lambda_N)$ be the diagonal matrix of the 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* $\eta = \{a; b^{(1)}, \cdots b^{(n)}\}$, satisfying $\widehat{\gamma}(2\xi) = \widehat{a}(\xi)\widehat{\gamma}(\xi)$ and $\widehat{\rho^{(r)}}(2\xi) = \widehat{b^{(r)}}(\xi)\widehat{\gamma}(\xi)$, for any $\xi \in \mathbb{R}$, where $\widehat{h}(\xi)$ denotes the Fourier transform of h, defined by $\widehat{h}(\xi) := \sum_{k\in\mathbb{Z}} h(k)e^{-2\pi ik\xi}$. Here, n represents the number of high-pass filters.

The functions $\psi_{j,p}(\mu)$ and $\phi^r_{j,p}(\mu)$ represent the low-pass and high-pass framelets at node μ associated with node p at scale level $j \in \{1, \cdots, J\}$ respectively, defined as follows:

$$\psi_{j,p}(\mu) = \sum_{q=1}^{N} \widehat{\gamma}\left(\frac{\lambda_q}{2^j}\right) u_q(p)u_q(\mu), \tag{4}$$

$$\phi^r_{j,p}(\mu) = \sum_{q=1}^{N} \widehat{\rho^{(r)}}\left(\frac{\lambda_q}{2^j}\right) u_q(p)u_q(\mu),\ r = 1, \ldots, n, \tag{5}$$

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

These low-pass (i.e., Eq. (4)) and high-pass framelets (i.e., Eq. (5)) distill coarse-grained and fine-grained information from hypergraph signals. The framelet coefficients $V_0, W^r_j \in \mathbb{R}^{N\times d}$ are respectively the low-pass and high-pass coefficient, defined as the inner product of framelets and the hypergraph signal $\mathbf{X} \in \mathbb{R}^{N\times d}$, i.e.,

$$V_0 = \langle \psi_{0,\cdot}, \mathbf{X}\rangle = \mathbf{U}^\top \widehat{\gamma}(\frac{\Lambda}{2})\mathbf{U}\mathbf{X}, \tag{6}$$

$$W^r_j = \left\langle \phi^r_{j,\cdot}, \mathbf{X}\right\rangle = \mathbf{U}^\top \widehat{\rho^{(r)}}(\frac{\Lambda}{2^{j+1}})\mathbf{U}\mathbf{X}. \tag{7}$$

Let $\mathcal{W}_{r,j}$ denote the decomposition operators such that $V_0 = \mathcal{W}_{0,J}\mathbf{X}$ and $W^r_j = \mathcal{W}_{r,j}\mathbf{X}$. Referring to Eq. (6) and Eq. (7), the framelet transform matrices for decomposition are expressed as:

$$\begin{aligned}
\mathcal{W}_{0,J} &= \mathbf{U}^\top \widehat{a}(2^{-S+J-1}\Lambda)\cdots\widehat{a}(2^{-S}\Lambda)\mathbf{U} := \mathbf{U}^\top\Lambda_{0,J}\mathbf{U},\\
\mathcal{W}_{r,1} &= \mathbf{U}^\top \widehat{b^{(r)}}(2^{-S}\Lambda)\mathbf{U} := \mathbf{U}^\top\Lambda_{r,1}\mathbf{U},\\
\mathcal{W}_{r,j} &= \mathbf{U}^\top \widehat{b^{(r)}}(2^{-S+j-1}\Lambda)\widehat{a}(2^{-S+j-2}\Lambda)\cdots\widehat{a}(2^{-S}\Lambda)\mathbf{U}\\
&:= \mathbf{U}^\top\Lambda_{r,j}\mathbf{U}.
\end{aligned} \tag{8}$$

Here, S is chosen to be sufficiently large such that the largest eigenvalue λ_{max} of the hypergraph Laplacian satisfies $\lambda_{max} \le 2^S\pi$.

The tightness of the framelet system can be guaranteed by the following condition:

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

This ensures that framelet decomposition and reconstruction are invertible, i.e., $\mathcal{W}^\top_{0,J}\mathcal{W}_{0,J}\mathbf{X} + \sum_{r,j}\mathcal{W}^\top_{r,j}\mathcal{W}_{r,j}\mathbf{X} = \mathbf{X}$.

In particular, the use of Haar-type filters ensures a computationally efficient multi-scale system (Zheng et al. 2021; Dong 2017). Specifically, for a framelet system with two scale levels (i.e., $j = 1, 2$) and one high-pass filter (i.e., $r = 1$), the filters are constructed as follows: $\widehat{\gamma}(\frac{\Lambda}{2}) = \cos(\frac{\Lambda}{8})\cos(\frac{\Lambda}{16})$, $\widehat{\rho}(\frac{\Lambda}{2}) = \sin(\frac{\Lambda}{8})\cos(\frac{\Lambda}{16})$ and $\widehat{\rho}(\frac{\Lambda}{4}) = \sin(\frac{\Lambda}{16})$, yielding one low-pass and two high-pass components, see Figure 2. For the purpose of addressing the computational challenges associated with the eigendecomposition of the hypergraph Laplacian, we use an approximation approach using Chebyshev polynomials, inspired by the methodology introduced in (Dong 2017). Specifically, the method utilizes fixed-degree Chebyshev polynomials $\mathcal{T}_0, \cdots, \mathcal{T}_k$, where filters $a \approx \mathcal{T}_0$ and $b^{(r)} \approx \mathcal{T}_r$. Consequently, the operators defined in Eq. (8) can be approximated as follows:

$$\begin{aligned}
\mathcal{W}_{0,J} &\approx \mathbf{U}^\top\mathcal{T}_0(2^{-S+J-1}\Lambda)\cdots\mathcal{T}_0(2^{-S}\Lambda)\mathbf{U}\\
&= \mathcal{T}_0(2^{-S+J-1}\mathcal{L})\cdots\mathcal{T}_0(2^{-S}\mathcal{L}),\\
\mathcal{W}_{r,1} &\approx \mathbf{U}^\top\mathcal{T}_r(2^{-S}\Lambda)\mathbf{U} = \mathcal{T}_r(2^{-S}\mathcal{L}),\\
\mathcal{W}_{r,j} &\approx \mathbf{U}^\top\mathcal{T}_r(2^{-S+j-1}\Lambda)\mathcal{T}_0(2^{-S+j-2}\Lambda)\cdots\mathcal{T}_0(2^{-S}\Lambda)\mathbf{U}\\
&= \mathcal{T}_r(2^{-S+j-1}\mathcal{L})\mathcal{T}_0(2^{-S+j-2}\mathcal{L})\cdots\mathcal{T}_0(2^{-S}\mathcal{L}).
\end{aligned}$$

With these preparations, we define the framelet-based (spectral) hypergraph convolution operator as follows:

$$\mathcal{F}(\mathbf{X}^{(\ell)}) = \sum_{(r,j)\in\Gamma}\mathcal{W}^\top_{r,j}\text{diag}(\theta_{r,j})\mathcal{W}_{r,j}\mathbf{X}^{(\ell)}, \tag{9}$$

where $\theta_{r,j} \in \mathbb{R}^N$ are 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.

Subsequently, a deep HGNN with tight framelets (FrameHGNN) can be constructed using initial residual and identity mapping techniques introduced in (Chen et al. 2020), that is,

$$\mathbf{X}^{(\ell+1)} = \sigma\left(\Big((1-\alpha_\ell)\mathcal{F}(\mathbf{X}^{(\ell)}) + \alpha_\ell\mathbf{X}\Big)\Big((1-\beta_\ell)\mathbf{I} + \beta_\ell\mathbf{\Theta}^{(\ell)}\Big)\right). \tag{10}$$

Figure 2 provides a detailed overview of the FrameHGNN architecture. Starting with the node feature information and adjacency matrix of a hypergraph, the model outputs node classification labels after processing the data through multiple convolution layers and a multilayer perceptron (MLP). The convolution layers operate by first decomposing the input data into high-pass and low-pass components using a Haar-type filter, which performs two stages of decomposition corresponding to two scale levels. The resulting matrices,

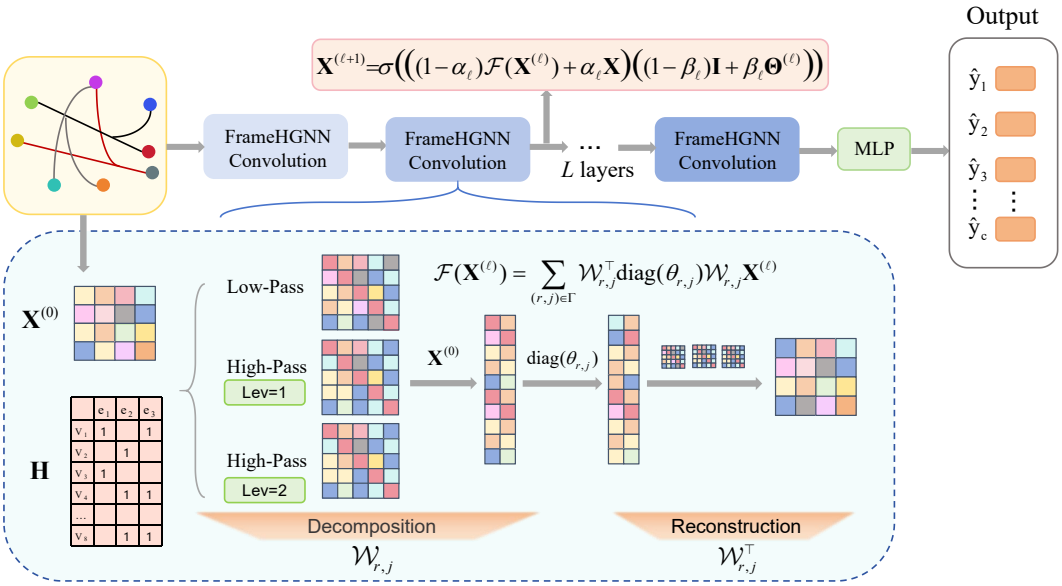

Figure 2: Schematic overview of our proposed FrameHGNN architecture.

representing low-frequency and high-frequency components, are then multiplied by the input feature matrix to generate frame coefficients. These coefficients are subsequently filtered through learnable filters, and the frame transform matrix is applied to reconstruct the coefficients, yielding the convolution output. In particular, to build a deep architecture, residual connections and identity mappings (Chen et al. 2020) are integrated between the convolution layers. These mechanisms ensure that the initial node information is preserved and effectively utilized throughout the training process.

For a more comprehensive understanding of the characteristics of tight framelets on hypergraphs, we provide an extended discussion of their theoretical properties in the **Appendix A**.

5 Experiments

This section presents a comprehensive evaluation of FrameHGNN on the task of node classification using eight benchmark datasets. The performance of FrameHGNN is compared against several classic HGNN models.

5.1 Experimental Setup

We conducted our experiments using eight publicly available datasets: Cora, CiteSeer, Pubmed, and Cora-CA (cocitation and coauthorship networks) obtained from (Yadati et al. 2019), as well as Senate (Fowler 2006), House (Chodrow, Veldt, and Benson 2021), NTU2012 (Chen et al. 2003), and ModelNet40 (Wu et al. 2015), which cover a variety of applications. Detailed descriptions of these datasets are provided in the **Appendix B**. The datasets were randomly divided into training, validation, and test sets, with proportions of 50%, 25%, and 25%, respectively. To prevent overfitting, an early stopping strategy was employed, halting training if no improvement in the validation performance was observed for 200 consecutive epochs, with a maximum of 1000 training epochs. To ensure the robustness of the results, we repeated each experiment ten times and calculated the standard deviation of the outcomes as the final evaluation metric. For comparison, we selected eight baseline methods: HGNN (Feng et al. 2019), HCHA (Bai, Zhang, and Torr 2021), HNHN (Dong, Sawin, and Bengio 2020), HyperGCN (Yadati et al. 2019), UniGCNII (Huang and Yang 2021), AllDeepSets (Chien et al. 2022), AllSetTransformer (Chien et al. 2022), and ED-HNN (Wang et al. 2023).

5.2 Results on Hypergraph Node Classification

The node classification results, measured by accuracy and its standard deviation, are summarized in Table 1, showing clearly that our method, FrameHGNN, outperforms seven of the eight datasets. Notably, on the Cora, CiteSeer, Cora-CA, NTU2012, and ModelNet40 datasets, FrameHGNN surpasses previous methods by over 1%. There is an even more significant improvement on Senate dataset with a nearly 3% gain, underscoring the effectiveness of FrameHGNN.

5.3 Ability to Alleviate Oversmoothing

This subsection examines the variation in accuracy as the number of layers increases, highlighting the ability of FrameHGNN to mitigate the oversmoothing issue. We compared FrameHGNN with HGNN, UniGCNII, AllSet models, and ED-HNN, with the results displayed in Figure 3. UniGCNII, known for its deep GCN architecture with residual connections, is effective in preventing oversmoothing; however, its performance fluctuates across different datasets and layer depths. While UniGCNII maintains high accuracy beyond 16 layers on the Cora-CA dataset, it shows a declining trend on the Senate and House datasets. ED-HNN, although performing well on Senate and House, exhibits a noticeable decline after 32 layers on Cora-CA, with an increasing standard deviation. Other methods experience a reduction in accuracy when

	Cora	Citeseer	Pubmed	Cora-CA	Senate	House	NTU2012	ModelNet40
HGNN	79.39 ± 1.36	72.45 ± 1.16	86.44 ± 0.44	82.64 ± 1.65	48.59 ± 4.52	61.39 ± 2.96	87.72 ±1.35	95.44 ± 0.33
HCHA	79.14 ± 1.02	72.42 ± 1.42	86.41 ± 0.36	82.55 ± 0.97	48.62 ± 4.41	61.36 ± 2.53	87.48 ± 1.87	94.48 ± 0.28
HNHN	76.36 ± 1.92	72.64 ± 1.57	86.90 ± 0.30	77.19 ± 1.49	50.93 ± 6.33	67.80 ± 2.59	89.11 ± 1.44	97.84 ± 0.25
HyperGCN	78.45 ± 1.26	71.28 ± 0.82	82.84 ± 8.67	79.48 ± 2.08	42.45 ± 3.67	48.32 ± 2.93	56.36 ± 4.86	75.89 ± 5.26
UniGCNII	78.81 ± 1.05	73.05 ± 2.21	88.25 ± 0.40	83.60 ± 1.14	49.30 ± 4.25	67.25 ± 2.57	89.30 ± 1.33	98.07 ± 0.23
AllDeepSets	76.88 ± 1.80	70.83 ± 1.63	88.75 ± 0.33	81.97 ± 1.50	48.17 ± 5.67	67.82 ± 2.40	88.09 ± 1.52	96.98 ± 0.26
AllSetTransformer	78.58 ± 1.47	73.08 ± 1.20	88.72 ± 0.37	83.63 ± 1.47	51.83 ± 5.22	69.33 ± 2.20	88.69 ± 1.24	98.20 ± 0.20
ED-HNN	80.31 ± 1.35	73.70 ± 1.38	**89.03 ± 0.53**	83.97 ± 1.55	64.79 ± 5.14	72.45 ± 2.28	88.67 ± 0.92	97.83 ± 0.33
FrameHGNN (ours)	**81.51± 0.99**	**74.72 ± 2.10**	88.73 ± 0.42	**85.18 ± 0.69**	**67.61 ± 5.27**	**72.82 ± 2.22**	**89.98 ± 2.02**	**98.41 ± 0.18**

Table 1: Performance comparison on classification accuracy (%) across eight datasets.

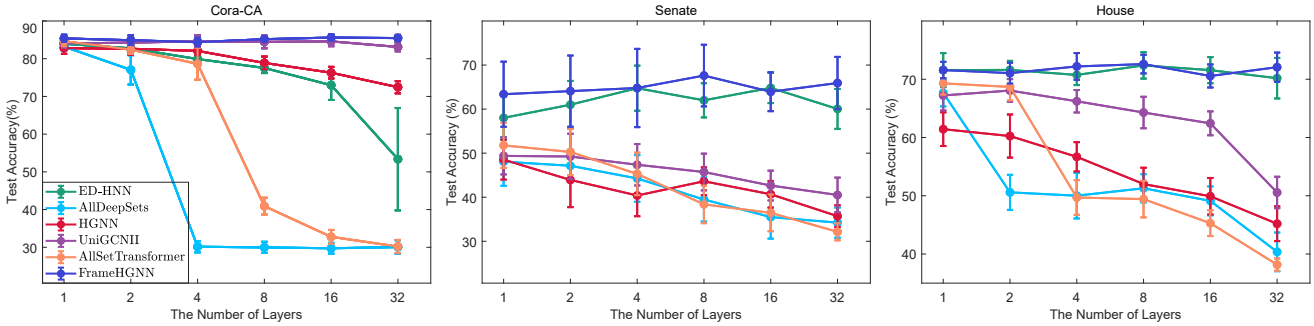

Figure 3: Comparison of the models' ability to alleviate oversmoothing: Cora-CA (LHS), Senate (Middle), House (RHS).

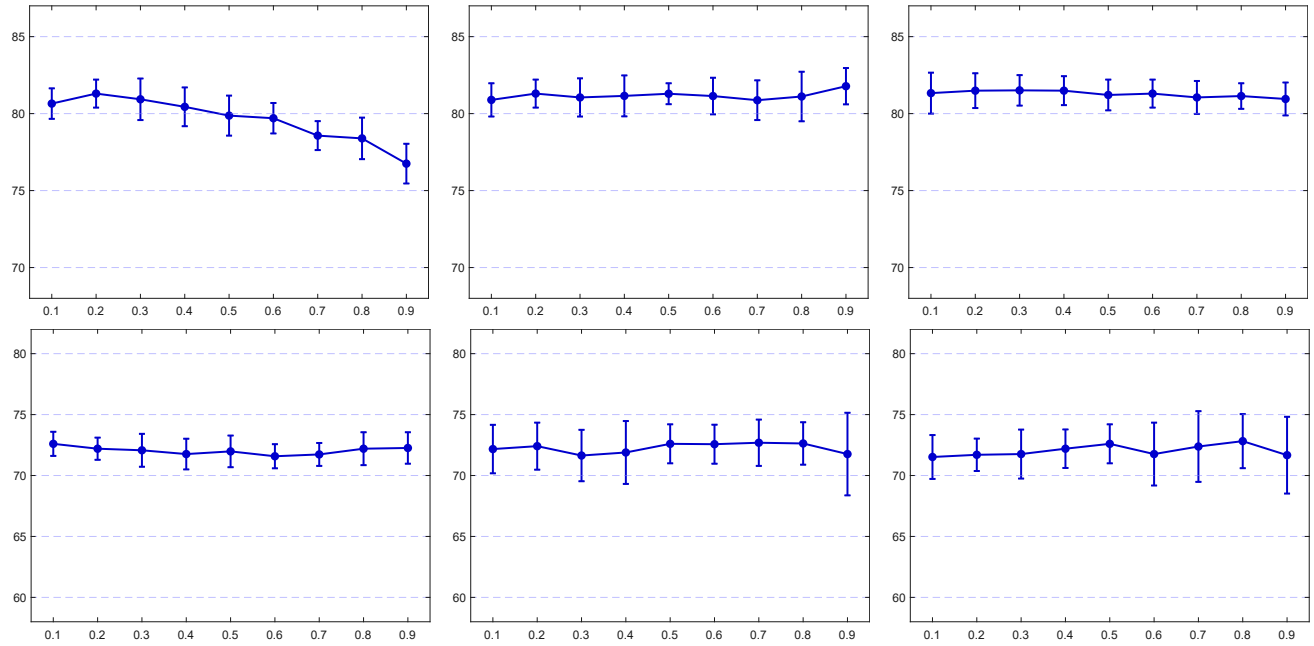

Figure 4: Parameter sensitivity analysis of α,γ,λ (L/M/RHS) for FrameHGNN on the Cora (top) and House (bottom) datasets.

the number of layers exceeds two. In contrast, FrameHGNN consistently improves performance with an increasing number of layers across all datasets. Specifically, on the Senate dataset, FrameHGNN achieves a 2% higher accuracy with 32 layers compared to 1 layer.

5.4 Parameter Sensitivity Analysis

FrameHGNN model includes three key hyperparameters: α, γ, and λ, where α controls the balance between current and initial layer features, γ balances features derived from random filtering and graph convolution, and λ adjusts the weight of each layer's feature updates relative to the initial features through the scaling parameter θ. We perform experiments on Cora and House, with the results illustrated in Figure 4. The values of α, γ, and λ were varied within the range$\{0.1, 0.2, 0.3, ..., 0.9\}$. The results indicate that suitable values of α can enhance performance, while γ and λ influence the stability of the model, as reflected in the standard deviation. It should be noted that additional empirical results for the parameter sensitivity analysis conducted on other benchmark datasets are provided in the **Appendix B**.

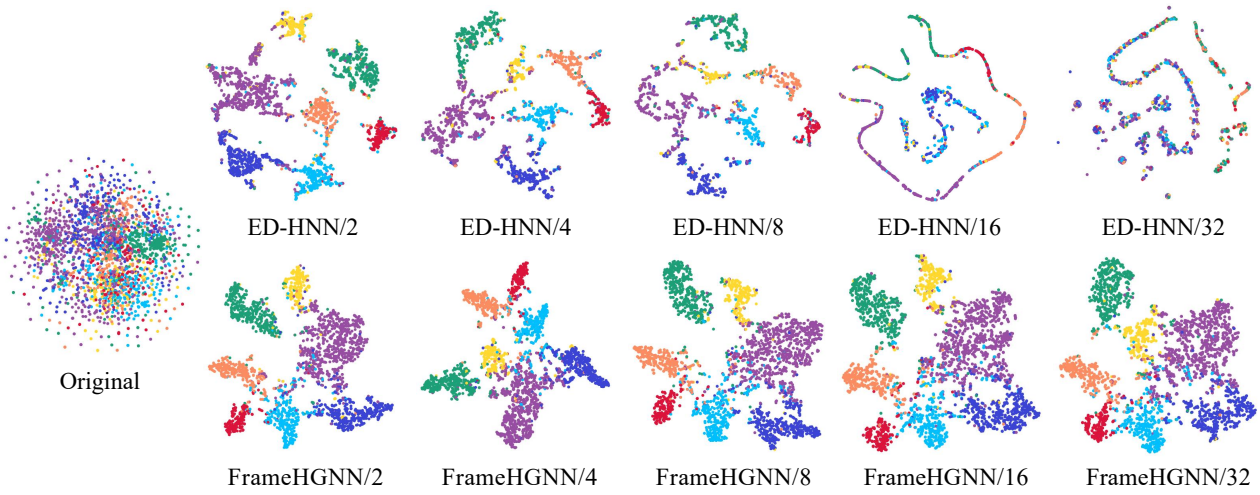

Figure 5: Node embedding visualization on the Cora-CA dataset: FrameHGNN vs. ED-HNN.

	Cora	Citeseer	Pubmed	Cora-CA	Senate	House	NTU2012	ModelNet40
Full model	81.51 ± 0.99	74.72 ± 2.10	88.73 ± 0.42	85.18 ± 0.69	67.61 ± 5.27	72.82 ± 2.22	89.98 ± 2.02	98.41 ± 0.18
w/o high pass	80.34 ± 1.48	74.65 ± 1.95	88.67 ± 0.44	85.01 ± 0.85	66.34 ± 6.03	71.76 ± 1.81	89.88 ± 2.20	98.51 ± 0.13
w/o low pass	81.42 ± 0.87	74.82 ± 2.14	88.75 ± 0.41	85.54 ± 0.94	65.07 ± 7.24	72.29 ± 2.37	89.15 ± 2.29	98.24 ± 0.17

Table 2: Ablation study on the contributions of low-pass and high-pass components.

5.5 Ablation Study

Motivated by our theoretical findings presented in Section 3, FrameHGNN integrates both low-pass and high-pass information from hypergraph signals. This section details the ablation study aiming to examine the role of the low-pass and high-pass components. Results in Table 2 provide compelling empirical support for our theoretical insights. Specifically:

i) The integration of low-pass and high-pass information in our model markedly enhances classification accuracy across most datasets. This outcome aligns with our theoretical expectation that maintaining a diverse set of features, both global (low-pass) and local (high-pass), across layers helps preserve node distinguishability. By leveraging this richer feature set, the convolutional operations effectively capture subtle variations in the data, mitigating feature collapse and preventing oversmoothing.

ii) Ablation results reveal that certain datasets, such as ModelNet40, benefit from the exclusion of high-pass information, suggesting that in some cases, high-pass features may introduce noise rather than useful details. This is consistent with the theoretical notion that not all types of information are equally beneficial across different datasets. FrameHGNN's capability to selectively incorporate or exclude high-pass information based on the specific characteristics of the dataset underscores its adaptability.

iii) On datasets such as Cora, CiteSeer, and Pubmed, excluding low-pass information results in better performance, highlighting the significance of high-pass features in these contexts. The theoretical insight that deep GCNs suffer from oversmoothing when feature diversity is lost is empirically supported by these results. By focusing on high-pass information, which often captures local variations and edge-specific details, our model successfully avoids the oversmoothing trap, ensuring that even deeper layers retain meaningful and distinguishable features.

5.6 Visualization

Figure 5 visualizes the vertex embeddings of ED-HNN and FrameHGNN at varying layer depths. Both models effectively differentiate between classes at depths of fewer than 16 layers. However, as the number of layers increases to 32, the embeddings in ED-HNN become homogeneous, making class distinction difficult. In contrast, FrameHGNN maintains distinct class separations as the layers deepen, demonstrating its resilience to oversmoothing.

6 Conclusion

This paper addresses the challenge of oversmoothing in deep HGNNs by conducting a theoretical investigation into its causes, particularly focusing on the spectral properties of hypergraph signals. Our analysis reveals the critical roles of both low-pass and high-pass components in mitigating oversmoothing, leading to the development of FrameHGNN, a framework that leverages tight framelet transforms to perform hypergraph convolutions with multifrequency channels. By integrating initial residual connections and identity mappings, FrameHGNN effectively addresses oversmoothing and demonstrates superior performance across eight benchmark datasets in hypergraph node classification tasks. Future work will explore adaptive mechanisms for balancing spectral components and will extend the framework to different learning paradigms (e.g., contrastive learning or continual learning) and more complex high-order networks or topological domains (e.g., simplicial/cell/combinatorial complexes).

Acknowledgements

This work was supported in part by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang (No. 2024C03262), the National Natural Science Foundation of China (No. U21A20473, No. 62172370) and the Jinhua Science and Technology Plan (No. 2023-3-003a). L. Bai was supported by the National Natural Science Foundation of China (No. T2122020). H. Feng was supported in part by Research Grants Council of Hong Kong under Project CityU11315522 and CityU11303821.

References

Bai, S.; Zhang, F.; and Torr, P. H. 2021. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 110: 107637.

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.

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

Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; and Li, Y. 2020. Simple and deep graph convolutional networks. In *ICML*, 1725–1735.

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

Chodrow, P. S.; Veldt, N.; and Benson, A. R. 2021. Generative hypergraph clustering: from blockmodels to modularity. *Science Advances*, 7(28): eabh1303.

Dong, B. 2017. Sparse representation on graphs by tight wavelet frames and applications. *Applied and Computational Harmonic Analysis*, 42(3): 452–479.

Dong, Y.; Sawin, W.; and Bengio, Y. 2020. HNHN: hypergraph networks with hyperedge neurons. In *ICML Graph Representation Learning and Beyond Workshop*.

Feng, W.; Zhang, J.; Dong, Y.; Han, Y.; Luan, H.; Xu, Q.; Yang, Q.; Kharlamov, E.; and Tang, J. 2020. Graph random neural networks for semi-supervised learning on graphs. In *NeurIPS*, 22092–22103.

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

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.

Klicpera, J.; Bojchevski, A.; and Günnemann, S. 2019. Combining neural networks with personalized pagerank for classification on graphs. In *ICLR*.

Li, M.; Li, Z.; Huang, C.; Jiang, Y.; and Wu, X. 2024a. EduGraph: Learning Path-based Hypergraph Neural Networks for MOOC Course Recommendation. *IEEE Transactions on Big Data*, 10(6): 706–719.

Li, M.; Zhou, S.; Chen, Y.; Huang, C.; and Jiang, Y. 2024b. EduCross: Dual adversarial bipartite hypergraph learning for cross-modal retrieval in multimodal educational slides. *Information Fusion*, 109: 102428.

Li, Q.; Han, Z.; and Wu, X.-M. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, 3538–3545.

Liu, M.; Gao, H.; and Ji, S. 2020. Towards deeper graph neural networks. In *KDD*, 338–348.

Murgas, K. A.; Saucan, E.; and Sandhu, R. 2022. Hypergraph geometry reflects higher-order dynamics in protein interaction networks. *Scientific Reports*, 12(1): 20879.

Oono, K.; and Suzuki, T. 2020. Graph neural networks exponentially lose expressive power for node classification. In *ICLR*.

Rong, Y.; Huang, W.; Xu, T.; and Huang, J. 2019. DropEdge: towards deep graph convolutional networks on node classification. In *ICLR*.

Rusch, T. K.; Bronstein, M. M.; and Mishra, S. 2023. A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993*.

Sun, X.; Cheng, H.; Liu, B.; Li, J.; Chen, H.; Xu, G.; and Yin, H. 2023. Self-supervised hypergraph representation learning for sociological analysis. *IEEE Transactions on Knowledge and Data Engineering*, 35(11): 11860–11871.

Tang, B.; Chen, S.; and Dong, X. 2024. HypergraphMLP: learning on hypergraphs without message passing. In *ICASSP*, 13476–13480.

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

Wu, H.; Yan, Y.; and Ng, M. K.-P. 2023. Hypergraph collaborative network on vertices and hyperedges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3): 3245–3258.

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.

Xia, L.; Huang, C.; and Zhang, C. 2022. Self-supervised hypergraph transformer for recommender systems. In *KDD*, 2100–2109.

Xu, K.; Li, C.; Tian, Y.; Sonobe, T.; Kawarabayashi, K.-i.; and Jegelka, S. 2018. Representation learning on graphs with jumping knowledge networks. In *ICML*, 5453–5462.

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.

Yan, J.; Feng, Y.; Ying, S.; and Gao, Y. 2024. Hypergraph dynamic system. In *ICLR*.

Zhao, L.; and Akoglu, L. 2020. PairNorm: tackling oversmoothing in GNNs. In *ICLR*.

Zheng, X.; Zhou, B.; Gao, J.; Wang, Y. G.; Lió, P.; Li, M.; and Montúfar, G. 2021. How framelets enhance graph neural networks. In *ICML*, 12761–12771.