Check for updates

ACM DIGITAL LIBRARY | Association for Computing Machinery | acm open

Latest updates: https://dl.acm.org/doi/10.1145/3637528.3672047

RESEARCH-ARTICLE

DPHGNN: A Dual Perspective Hypergraph Neural Networks

SIDDHANT SAXENA, Indian Institute of Technology Delhi, New Delhi, DL, India

SHOUNAK GHATAK, Indian Institute of Technology Delhi, New Delhi, DL, India

RAGHU KOLLA

DEBASHIS MUKHERJEE

TANMOY CHAKRABORTY, Indian Institute of Technology Delhi, New Delhi, DL, India

Open Access Support provided by:

Indian Institute of Technology Delhi

PDF Download
3637528.3672047.pdf
11 March 2026
Total Citations: 0
Total Downloads: 523

Published: 24 August 2024

Citation in BibTeX format

KDD '24: The 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining
August 25 - 29, 2024
Barcelona, Spain

Conference Sponsors:
SIGMOD
SIGKDD

DPHGNN: A Dual Perspective Hypergraph Neural Networks

Siddhant Saxena
IIT Delhi
New Delhi, India
siddhantsaxenaphy@gmail.com

Shounak Ghatak
IIIT Delhi
New Delhi, India
shounak19109@iiitd.ac.in

Raghu Kolla
Meesho
Bangalore, India
raghu.kolla@meesho.com

Debashis Mukherjee
Meesho
Bangalore, India
debashis.mukherjee@meesho.com

Tanmoy Chakraborty
IIT Delhi
New Delhi, India
tanchak@iitd.ac.in

ABSTRACT

Message passing on hypergraphs has been a standard framework for learning higher-order correlations between hypernodes. Recently-proposed hypergraph neural networks (HGNNs) can be categorized into spatial and spectral methods based on their design choices. In this work, we analyze the impact of change in hypergraph topology on the suboptimal performance of HGNNs and propose DPHGNN, a novel dual-perspective HGNN that introduces equivariant operator learning to capture lower-order semantics by inducing topology-aware spatial and spectral inductive biases. DPHGNN employs a unified framework to dynamically fuse lower-order explicit feature representations from the underlying graph into the super-imposed hypergraph structure. We benchmark DPHGNN over eight benchmark hypergraph datasets for the semi-supervised hypernode classification task and obtain superior performance compared to seven state-of-the-art baselines. We also provide a theoretical framework and a synthetic hypergraph isomorphism test to express the power of spatial HGNNs and quantify the expressivity of DPHGNN beyond the Generalized Weisfeiler Leman (1-GWL) test. Finally, DPHGNN was deployed by our partner e-commerce company, Meesho for the Return-to-Origin (RTO) prediction task, which shows 7% higher macro F1-Score than the best baseline.

CCS CONCEPTS

• **Computing methodologies** → *Semi-supervised learning settings*; **Semantic networks**.

KEYWORDS

Graph neural network; Hypergraph neural networks; Topological deep learning; RTO prediction

ACM Reference Format:
Siddhant Saxena, Shounak Ghatak, Raghu Kolla, Debashis Mukherjee, and Tanmoy Chakraborty. 2024. DPHGNN: A Dual Perspective Hypergraph Neural Networks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3672047

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDD '24, August 25–29, 2024, Barcelona, Spain
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3672047

1 INTRODUCTION

In recent years, the use of spatial and spectral message-passing neural networks (MPNN) [37, 43] on graph topology has grown exponentially to solve various downstream tasks such as node label classification, link prediction, and graph classification [26]. Graph Neural Networks (GNNs) achieve exceptional performance for representation learning on graph-structured data. GNNs improve contextual feature representation of nodes in the graph via layer-wise spatial feature aggregation and message propagation. A wide range of applications involving graph-based semi-supervised node classification, such as molecular property prediction [44], topic modeling of research papers from a scientific citation network [23], user-item recommendations in e-commerce networks [11], have been explored. However, the graph structure limits modeling the interactions to pair-wise node connections and, therefore, is restricted to capturing only the lower-order correlation and relationships between entities.

Hypergraphs provide a flexible mechanism to model higher-order data correlation and complex relationships by allowing hyperedges of two or more hypernodes. Many real-world applications of semi-supervised node classification involve higher-order relation modeling [14], such as academic citation networks consisting of hypernodes as scientific authors and hyperedges as coauthorship relations among authors. [15] used hypergraph structure to describe quantum optical experiments. [20] improved collaborative filtering, and [2] used session-based recommendations using higher-order relations. [27] considered semi-dynamic hypergraphs for 3d pose estimation. Recent studies proposed models for Hypergraph Neural Networks (HGNNs) [12], Hypergraph Convolution and Hypergraph Attention [3]. However, despite the flexibility and advances of message-passing neural networks on hypergraphs, there exist some topological and design-related challenges, as mentioned below:

- Performance gap between spatial and spectral HGNNs with modeling data into hypergraph topology.
- Effects of over-smoothing and feature collapse in hypergraph topology with resource-constrained setting (e.g., class imbalance, sparse incidence structure, a limited set of features, etc.) .
- Exploiting the underlying graph structure to explicitly incorporate information from the lower-order structure to the higher-order message-passing framework.

Notation	Description	Notation	Description
$G = (V, E)$	An undirected graph with a set of nodes V and a set of edges E	$HG = (V, \xi)$	An undirected hypergraph with a set of hypernodes V and a set of hyperedges ξ
A	Adjacency Matrix	H	Hypergraph incidence matrix
D_v	Node degree matrix	D_v	Hypernode degree matrix
L_{sym}	Symmetric Laplacian Matrix	D_e	Hyperedge degree matrix
G_c, A_c	Clique decomposition graph and adjacency	Δ_{HGNN}	HGNN Laplacian matrix
G_*, A_*	Star decomposition graph and adjacency	Δ_{sym}	Symmetric Laplacian matrix
G_{hyp}, A_{hyp}	HyperGCN decomposition graph and adjacency	Δ_{rw}	Random-walk Laplacian matrix

Table 1: A summary of notations used throughout the paper.

To address these challenges, we propose DPHGNN (**D**ual **P**erspective **H**yper**g**raph **N**eural **N**etwork), a mixed spectral and spatial learning framework that improves feature representation learning on resource-constrained hypergraph settings. DPHGNN adopts – (i) a dual-layered feature update mechanism, (ii) a static update layer to provide spectral inductive biases and lower-order relational features to update the static feature matrix of hypernodes, and (iii) a dynamic update layer to fuse the explicitly aggregated features from the underlying graph topology in the hypergraph spatial message propagation. Moreover, with extensive empirical and theoretical analyses, we show that DPHGNN can tackle the above-mentioned challenges and produce improved feature representations for the downstream hypernode classification task.

We summarize our major contributions below[1]:

- DPHGNN introduces a novel message propagation framework that explicitly diffuses lower-order graph features to super-imposed hypergraph structure.
- We introduce equivariant operator learning (EOL) over TAA and SIB inductive biases. EOL creates an information-rich feature mixture, rather than cold-start with HG initial features. This also constitutes maximally expressive, k-order equivariant layers in DPHGNN.
- We perform extensive empirical analysis on DPHGNN's generalized performance over eight benchmark datasets, and a new isomorphic HG classification task. We formalize strong mathematical characterization on 3-GWL expressivity of DPHGNN, automorphism groups, and EOL.
- We introduce CO-RTO, a new real-world application to tackle the challenge of the RTO problem in e-commerce. DPHGNN being robust to variations in topological constraints, beats HGNN baselines with a large margin.

2 PRELIMINARIES, BACKGROUND AND RELATED WORK

Let $HG = (V, \xi)$ denote an undirected hypergraph without self-loops, where V is a set of hypernodes and a hyperedge $e \in \xi$ is composed of a set of nodes; therefore, $\{e \subset V\} \wedge \{e \neq \emptyset\}$. The incident edges of hypernode i are denoted by $E_i = \{e \in \xi \mid i \in e\}$. Table 1 summarizes the notations used throughout the paper.

Graph Neural Networks. Graph Neural Networks (GNNs) have been successful by following the message-passing neural network (MPNN) paradigm to update representations of nodes in the graph topology by aggregating the feature information from neighboring nodes. A simple message propagation in GNNs can be formulated as follows: spectral in

$$\hat{x}_v^{(l)} = \text{Update}^{(l)}\left(x_v^{(l)}, \text{Aggr}^{(l)}\left(\left\{x_u^{(l-1)} \forall u \in N(v)\right\}\right)\right)$$

where $\text{Aggr}^{(l)}$ aggregates node embeddings of the neighborhood $N(\cdot)$ of node v at layer $l-1$, and $\text{Update}^{(l)}$ assigns aggregated embedding to node v at layer l. $N(v)$ is the neighbors of node v, and $x_u^{(l-1)}$ is the representation of node u at $(l-1)$ layer. The foundation message-passing GNN models include GCN [21], GraphSAGE [16], GIN [38], JK-GCN [39] and GAT [36].

Spatial Hypergraph Networks. To generalize MPNN over hypergraph, several attempts have been made in which spatial HGNNs tend to aggregate the hypernode embeddings from a hyperedge, aggregate the embedding from neighboring hyperedges and update the hypernode feature information [10, 13]. A simple spatial message propagation in HGNNs can be formulated as follows:

$$\hat{x}_v^{(l)} = \text{Update}^{(l)}\left(x_v^{(l)}, \text{Aggr}^{(l)}\left(\left\{\left\{\text{Aggr}^{(l-1)}\left(x_u^{(l-1)}\right) \forall u \in e\right\} \forall e \in E_i\right\}\right)\right) \quad (1)$$

The benchmark models following spatial HGNN mechanism include HGNNP [13], HyperSAGE[2], HNHN [10], HAIN [4] and the family of UniGNNs [18].

Spectral Hypergraph Networks. An HGNN layer uses spectral convolution to learn node/edge representation via approximating each cluster to a clique and requires quadratic computation. The HyperGCN layer [41] considers a linear number of edges resulting in a reduction in training times and uses a Laplacian operator for hypergraphs to approximate the hyperedges followed by a regular graph convolution operation on the resulting graph. MPNN-R [40] considers hyperedges as new vertices $V = \{V \cup \xi\}$ and presents the hypergraph with a $|V| \times |\xi|$ matrix. However, it fails to describe the intersections between hyperedges, resulting in a loss of complex semantic relations. These approaches can be implemented as smoothing features with trainable hypergraph Laplacian operators. Some frequently used Laplacian operators include HGNN Laplacian matrix $\Delta_{HGNN} = D_v^{-1/2} H D_e^{-1} H^T D_v^{-1/2}$, Symmetric Laplacian matrix $\Delta_{sym} = I - D_v^{-1/2} H D_e^{-1} H^T D_v^{-1/2}$, Random walk Laplacian matrix $\Delta_{rw} = I - D_v^{-1} H D_e^{-1} H^T$ [30] (see Table 1 for notations).

[1]The source code of DPHGNN is available at https://github.com/mr-siddy/DPHGNN. The CO-RTO dataset can't be released due to privacy concerns.

Algorithm 1 DPHGNN: Dual Perspective Hypergraph Neural Network

Input Hypergraph incidence matrix: $H \in R^{m \times n}$
Hypernode feature matrix: X_{HG}
Target labels: $\{y\}_{\text{train}}$
Epochs: e
Num Layers: k
Output Learned representations $\widehat{X}_{\text{HG}}$ for downstream tasks.

for i = 1 to e + 1 epochs **do**
  Project: $f(x) : X_{HG} \rightarrow X'_{HG}$ hidden dimension
  Compute: $\Delta_{\text{spectral}} = \left[(\Delta_{rw} \oplus \Delta_{\text{sym}}) \,|; |\Delta_{HGNN} \right]$
  Update hypernodes: $X_{HG_{\text{spectral}}} = \sigma \left[(I + \lambda \Delta_{\text{spectral}}) X'_{HG} \theta \right]$
  Compute TAA: $\widehat{x}_\kappa = \Sigma_{\kappa \in N_\beta} \widehat{\alpha}_{\beta Y} \kappa, \widehat{x}_z = \Sigma_{Z \in N_x} \widehat{\alpha}_{XY} Z$
  Update hypernodes:
  $X_{\text{eqv}} = MLP^1(\hat{x}_\kappa |; |\hat{x}_Z) \odot \sigma(\text{ReLU}(MLP^2(X_{HG_{\text{spectral}}})))$
  Update static features
  $X_{\text{static}} = X_{\text{eqv}} |; | MLP^3(X_{\text{HG}})$
  for l in $k-1$ layers **do**
    Update hypernodes as:
    Feature Fusion:
    $X_{\text{Fused}} = \left(H^T D_v^{-1/2} X_{\text{static}} + D_v^{-1} \phi_{\text{mask}} \ \{AX_G*\} \right)$
    $X_{\text{DPHGNN}} = \sigma \left[X_{\text{static}} + H D_e^{-1} X_{\text{Fused}} \Theta \right]$
  Update hypernodes as:
  $\widehat{X}_{HG} = \sigma \left(D_v^{-1/2} X_{\text{DPHGNN}} \bar{D}_e^{-1/2} \cdot D_e^{-1} H^T D_v^{-1/2} X_{\text{DPHGNN}} \Theta \right)$
  Compute cross-entropy loss between $\{y\}_{\text{train}}$ and predictions $\left\{ \widehat{X}_{HG} \right\}_{\text{train}}$
  Backpropagate loss and finetune parameter set using Adam optimizer

3 DUAL PERSPECTIVE HYPERGRAPH NEURAL NETWORK (DPHGNN)

Our proposed DPHGNN architecture is presented in Figure 1 and summarized in Algorithm 1. DPHGNN first induces various structural and spectral inductive biases to learn the underlying lower-order relations and spectral features. Following this, a dynamic feature fusion mechanism fuses explicit aggregated features from specific graph topology with the hypergraph message passing. We finetune different sub-layers in the static and dynamic induction blocks with cross-entropy loss by making the convolution layer an end-to-end differentiable pipeline. Specific building blocks are explained in the remaining section.

3.1 Topology-Aware Attention (TAA)

A hypergraph can be decomposed into various substructures representing underlying pairwise connections between hypernodes. We carefully decompose it into three particular graph topologies – (i) **Clique expansion** G_c is used to learn the interconnection of node-pair entities. (ii) **Star expansion** G_* is used to generate explicit supernodes utilized in the dynamic feature fusion module by masking the synthetic supernodes and nodes already present in the graph. This can be achieved by introducing a mask defined as,

$$\phi_{\text{mask}} = \begin{cases} 1 \text{ if } x_i \in V^* \cup S^* \\ 0 \text{ if } x_i \in (V^* \cup S^*) \backslash V^* \end{cases}$$

(iii) **HyperGCN expansion** G_{hyp} is used to learn lower-order graph functions, approximate the hypergraph learning functions and update the feature representation using a single-layer MPNN update. The choice of respective graph MPNN is made to maximize the induction of feature updates in respective graph topologies with the following rules:

$$X_c = \sigma \left[\left(I + D_v^{-1} A_c \right) X\theta \right]; X_* = \sigma \left[\left(I + D_v^{-1} A_* \right) X\theta \right]$$
$$X_{hyp} = \sigma \left(\widehat{D}_v^{-1/2} \widehat{A}_{hyp} \widehat{D}_v^{-1/2} X\theta \right)$$

where X_c, X_*, and X_{hyp} represent single-layer MPNN update of nodes on G_c, G_*, and G_{hyp}, respectively. We also compute smoothened features through graph Laplacian smoothing as $L_* = D_v - A_*$, $L_c = D_v - A_c$, $L_{hyp} = D_v - A_{hyp}$ on G_*, G_c, and G_{hyp}, respectively. This allows aggregating feature representations induced from different possible semantic relations among lower-order entities. Moreover, we adopt a cross-attention mechanism for graphs [42] to generate topology-aware feature representations described below.

3.1.1 TAA on Spatial Features. Let $\beta = \{\beta_i = x_i | \forall x_i \in \phi_{mask}(X_*)\}$, $\gamma = \{\gamma_j = x_j | \forall x_j \in X_c\}$, and $\kappa = \{\kappa_k = x_k | \forall x_k \in X_{hyp}\}$ be the sets of feature representation of decomposed graph topology G_*, G_c, and G_{hyp}. The topology-aware attention weights $\widehat{\alpha}_{\beta\gamma}$ would be along with trainable attention parameters $\delta \in \mathbb{R}^{2d}$ with ";" denoting concatenation operation. The updated feature representation for G_{hyp} is then obtained by,

$$\alpha_{\beta\gamma} = \sigma \left(\delta^T [W\beta |; | W\gamma] \right); \ \widehat{\alpha}_{\beta\gamma} = \frac{\exp\left(\alpha_{\beta\gamma}\right)}{\sum_{p \in N_\beta} \exp\left(\alpha_{\beta\gamma}\right)};$$
$$\widehat{x}_\kappa = \Sigma_{p \in N_\beta} \widehat{\alpha}_{\beta\gamma} W\kappa$$

3.1.2 TAA on Spectral Features. Let $X = \{l_i = x_i | \forall x_i \in \phi_{mask}[L_* \otimes X_*]\}$, $Y = \{m_j = x_j | \forall x_j \in [L_c \otimes X_c]\}$, and $Z = \{n_k = x_k | \forall x_k \in [L_{hyp} \otimes X_{hyp}]\}$ be the sets of Laplacian smoothen features of decomposed topologies G_*, G_c, and G_{hyp}, respectively. Similarly, the attention-weighted features for $L_{hyp} \otimes X_{hyp}$ feature matrix would be obtained by,

$$\alpha_{XY} = \sigma \left(\delta^T [WX |; | WY] \right); \ \widehat{\alpha}_{XY} = \frac{\exp\left(\alpha_{XY}\right)}{\sum_{p \in N_X} \exp\left(\alpha_{XY}\right)};$$
$$\widehat{x}_Z = \Sigma_{p \in N_X} \widehat{\alpha}_{XY} WZ$$

Here, W represents a linear transformation matrix. After the topological attention-based re-weighting of the graph's structural and spectral features, we elapse them to the Feature Mixer Module.

3.2 Feature Mixture Module

The sub-optimal performance of existing HGNNs, with the change in hypergraph topology, naturally motivates us to construct a generalized learning mechanism. To this end, we introduce a feature mixture module that first aggregates the spectral inductive biases (SIB) and then produces a mixture of features satisfying the following desiderata: TAA concatenation operation (i) provides spatial and spectral features of lower-order graph topology for more rich semantic feature representations; (ii) prevents over-smoothing of relevant information at the time of message passing over sparse hypergraph topology. SIB concatenation operation provides symmetrical random walk-based HGNN Laplacian smoothen features,

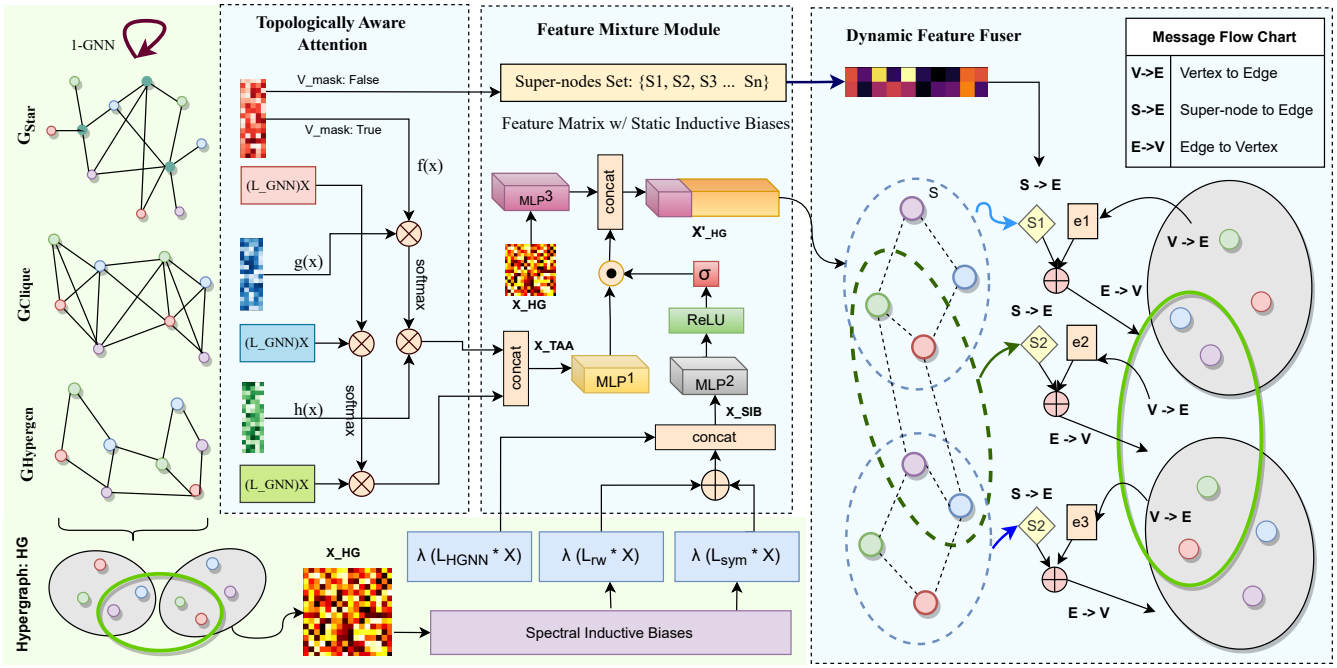


Figure 1: A schematic diagram of our proposed architecture, DPHGNN. Left: Hypergraph decomposition and topology-aware Attention (TAA) mechanism. Middle: Feature Mixture that generates static features by incorporating spectral inductive biases from hypergraph Laplacian smoothing and TAA. Right: Dynamic feature fusion (DFF) that fuses explicitly learned graph embedding in supernodes with the hypergraph message-passing module.

inherently giving unique identifiers to hypernodes. This helps us break automorphism groups.

Aggregating Spectral Inductive Biases DPHGNN leverages smoothened features from symmetrical random walk-based HGNN Laplacian, denoted by $\Delta_{\text{spectral}} = \left[(\Delta_{\text{rw}} \oplus \Delta_{\text{sym}}) \,|;|\, \Delta_{\text{HGNN}}\right]$. The SIB block is inherently inspired by different approaches to formulating the hypergraph Laplacian, where the smoothened features $X_{\text{HG}_{\text{spectral}}} = \sigma\left[X + \lambda\Delta_{\text{spectral}}X\right]$, and the concatenation operation $|;|$ is crucial to prevent the nullification of spectral features. This is because the element-wise sum of Laplacian matrices $\Delta_{\text{HGNN}} \oplus \Delta_{\text{sym}} \oplus \Delta_{\text{rw}}$ trivially results in $3I - D_v^{-1}HD_e^{-1}H^T$, i.e., it contains only the random-walk Laplacian information. These features provide unique identifiers to hypernodes, implicitly helping to break automorphism groups in hypergraph topology.

Equivariant Feature Mixing. We employ an equivariant operator learning mechanism to perform static feature updates on the hypernode features. The aggregated hypernode representations from TAA and SIB modules are propagated through downscaling MLPs, $MLP^1 : \mathbb{R}^{n\times 2d} \rightarrow \mathbb{R}^{n\times d/2}$ and $MLP^2 : \mathbb{R}^{n\times 2d} \rightarrow \mathbb{R}^{n\times d/2}$ in parallel. The original features from hypernodes are propagated through $MLP^3 : \mathbb{R}^{n\times d} \rightarrow \mathbb{R}^{n\times d/2}$. We perform a permutation equivariant operation on aggregated features:

$$X_{\text{eqv}} = MLP^1(\hat{x}_K|;|\hat{x}_Z) \odot \sigma(\text{ReLU}(MLP^2(X_{HG_{\text{spectral}}}))) \quad (2)$$

where $\odot$ represents Hadamard product; the static feature update finally upscales the dense information-rich features through a concatenation $|;|$ operation as $X_{\text{static}} = X_{\text{eqv}}|;|MLP^3(X_{\text{HG}})$.

Model	Time Complexity
HGNN	$O(nd^3)$
HGNN+	$O(nd^2 + md_e^2)$
HyperGCN	$O(mn^2d^2)$
HNHN	$O(nd^2 + md^2)$
UniGCN	$O(md + nd^2)$
DPHGNN	$O((n+s_*)\bar{d}_v\bar{d} + n_{hyp}\bar{d}^2 + m_{hyp}\bar{d}^2)$

Table 2: Time Complexity of different models.

3.3 Dynamic Feature Fusion (DFF)

Here, we describe the dynamic feature fusion mechanism to explicitly capture the lower-order semantics from pair-wise node interactions and diffuse it into the hypergraph message-passing step to learn generalizable feature representations. The star expansion produces graph topology G_* by connecting existing nodes in the edges to a synthetic node for the hyperedge set [45]. This allows us to extract aggregated information of a neighborhood in the graph in these synthetic nodes (called *supernodes*, hereafter). We utilize this feature representation from supernodes and fuse it into features aggregated from hypernodes towards hyperedges. We then update the representations in the neighborhood from the fused hyperedge towards subsequent hypernodes. Following this, we reiterate the feature updation without dynamic features on the hypergraph topology for the final prediction layer and perform class prediction. The dynamic message-passing mechanism described here can be formulated using the following update rules:

$$
\begin{aligned}
m_p^{(l)} &= \phi_{\text{mask}}\left[\text{Aggr}^{(l)}\left(\left\{m_u^{(l-1)} \forall u \in N(p)\right\}\right)\right] \\
m_e^{(l-1)} &= \text{Aggr}^{(l-1)}\left(\left\{m_u^{(l-1)} \forall u \in e\right\} \cup \left\{m_p^{(l)} \forall p \in E\right\}\right) \\
m_e^{(l)} &= \text{Aggr}^{(l)}\left(\left\{m_e^{(l-1)} \forall e \in E_i\right\}\right) \\
x_v^{(l)} &= \text{Update}^{(l)}\left(\left\{x_v^{(l-1)}, m_e^{(l)}\right\}\right)
\end{aligned}
$$

The embedding update mechanism for hypernodes proposed in DPHGNN can be summarised below:

$$
\widehat{X}_{\text{DPHGNN}} = \sigma\left[X_{\text{static}} + HD_e^{-1}\left(H^T D_v^{-\frac{1}{2}} X_{\text{static}} + D_e^{-1}\phi_{\text{mask}}\left\{AX_{G_*}\right\}\right)\Theta\right] \tag{3}
$$

We initialize the hypernode features with X_{static}. DPHGNN performs summation over supernodes and hypernodes at the time of aggregation. The dimensions of the DPHGNN matrix multiplications are as follows: $H \in \mathbb{R}^{n\times m}$, $H^T \in \mathbb{R}^{m\times n}$, $A \in \mathbb{R}^{n\times n}$. $X_{\text{static}} \in \mathbb{R}^{n\times d}$, and $X_{G_*} \in \mathbb{R}^{n+m\times d}$ are input matrices. $D_v \in \mathbb{R}^{n\times n}$, and $D_e \in \mathbb{R}^{m\times m}$ are the diagonal matrices. The dimensions involved in the operations are $D_e^{-1}\phi_{\text{mask}}\{AX_{G_*}\}$, $\mathbb{R}^{m\times m} \times \mathbb{R}^{m\times d} \rightarrow \mathbb{R}^{m\times d}$, $H^T D_v^{-1/2} X_{\text{static}}$, $\mathbb{R}^{m\times m} \times (\mathbb{R}^{n\times n} \times \mathbb{R}^{n\times d}) \rightarrow \mathbb{R}^{m\times d}$. At the DFF, $X \in \mathbb{R}^{m\times d}$, and finally, $\mathbb{R}^{n\times m} \times (\mathbb{R}^{m\times m} \times \mathbb{R}^{m\times d}) \rightarrow \mathbb{R}^{n\times d}$.

3.4 Time Complexity Analysis

Give an attributed hypergraph $\text{HG}(V, \xi, d_v, d_e, X)$, where d_v, d_e, and X are average degree of n hypernodes, average degree of m hyperedges, and multiset of features $\{x_v\}_{v\in V}$, where $x_v \in \mathbb{R}^d$ is of dimension d. The computation time of decomposing hypergraph into multiple views of graph (clique, hypergcn, star) structure and applying graph convolution is $O(n_c d_v d_e d + n_{hyp} d_v d + (n+s)_* d_v d_e d) \leq O((n+s)_* d_v d_e d + n_{hyp} d_v d)$. For inductive priors, the compute time is upper bounded by $O(n_{hyp} d_v d_e d + (n+m)_{hyp} d^2)$. The time complexity for DPHGNN message update mechanism is $O((n+s)_* d_v d + n_{hyp} d + m_{hyp} d + n_{hyp} d^2 + m_{hyp} d^2) \leq O((n+s)_* d_v d + n_{hyp} d^2 + m_{hyp} d^2)$, subscripts $*$, c, and hyp denote the respective vertex set G_*, G_c, and G_{hyp}.Table 2 presents a comparative analysis between the proposed DPHGNN and baseline HGNNs. We also provide corresponding run-time analysis in Table 11 in the appendix.

4 THEORETICAL CHARACTERIZATION

4.1 Equivariant Operator Learning

The DPHGNN architecture generalizes the performance across the change in hypergraph topology; several operators are used to bridge the tradeoffs of spatial and spectral HGNNs. In proposition 4.1 we prove that the feature mixer module encodes features from TAA and SIB blocks in a permutation equivariant manner. We have provided an in-depth analysis of the correlation between specified model blocks and hypergraph dataset in the Ablation Study section in Experiments.

Proposition 4.1. *The encoding function $f : (X_{HG}, X_{TAA}, X_{SIB}) \rightarrow X_{static}$ learned by equation 2 is permutation equivariant, i.e, if π is a bijective function; $f(\pi \cdot X) = \pi \cdot f(X)$.*

Proof. Let $A = MLP^1(\hat{x}_\kappa|; |\hat{x}_Z)$, and $B = (MLP^2(X_{HG_{\text{spectral}}}))$ are the matrices obtained by equation 2 with elements a_{ij}, b_{ij} respectively. Let $a'_{ij} = a_{\pi(i)\pi(j)}$ and $b'_{ij} = b_{\pi(i)\pi(j)}$ are elements of matrics A, B after permutation π. The Hadamard product matrix (C) of elements results in $c'_{ij} = a'_{ij} \odot b'_{ij}$. The original Hadamard Product $C = A \odot B$ has elements $c_{ij} = a_{ij} \odot b_{ij}$, applying permutation π to C will result in the same operation as with π to original matrices, i.e. $c_{\pi(i)\pi(j)} = a_{\pi(i)\pi(j)} \odot b_{\pi(i)\pi(j)}$. Hence, function f is permutation equivariant. □

4.2 Expressive Power of DPHGNN

The expressive power of HGNNs is determined by their ability to learn a function on hypergraph that can distinguish two non-isomorphic hypergraphs [5] and their local substructures. [18] formulated a variant of the Generalized Weisfeiler Leman (1-GWL) test for measuring the expressive power of UniGNNs, following [6]. However, the increase in symmetry from automorphisms in graph/hypergraph structure GWL inherits several failure cases in distinguishing complex substructures. In Proposition 4.2, we generalize the 1-GWL test for spatial HGNNs which follow message passing (c.f. Eq. 1) through the lens of the hypergraph color refinement algorithm. We then analyze the expressive power of DPHGNN in theorem 4.3 and prove that providing explicit representation information of underlying graph structure helps break automorphisms [31] via learning equivariant functions.

Proposition 4.2. *Given a function $f_\theta \in F^{HGNN}$ (learned by Eq. 1), and two non-isomorphic hypergraphs, H_1 and H_2, f_θ can distinguish $f_\theta(H_1) \neq f_\theta(H_2)$ if and only if for some $t > 0$, the updated coloring $HC_{(t)}(V_1, H_1) \neq HC_{(t)}(V_2, H_2)$.*

Proof Sketch. Here, we provide the skeleton of the proof (see A.1 Supplementary for the detailed proof). We first generalize the color refinement for hypergraphs by providing tensor representations for colors. We then establish an injective mapping between the HASH function of iterative hypergraph color refinement (c.f. Eq. 4) and general spatial HGNNs update (c.f. Eq. 1)

$$
\begin{aligned}
HC_{t+1}^H(v) = \text{HASH}\{\{\{HC_t^H(u) \mid u \in f_H(e)\} \\
\mid e \in \xi \text{ with } v \in f_H(e)\}\}
\end{aligned} \tag{4}
$$

Theorem 4.3. *Given a function $g_\theta \in F^{DPHGNN}$ (learned by Eq. 3), and two non-isomorphic hypergraphs H_1 and H_2 that can be distinguished by 3-GWL test, there is at least a function g_θ that can also decide $g_\theta(H_1) \neq g_\theta(H_2)$ if g_θ is equipped with the following structure: (i) g_θ is an order invariant function learned by the composition of k-equivariant layer ($k \geq 3$) HGNN network. (ii) Aggregation and Readout functions of DPHGNN are injective.*

Proof Sketch. Here, we provide the skeleton of the proof (see A.2 Supplementary for the detailed proof). We first build upon the work of [28, 29] to generalize the notion of the F-WL (Folklore-WL) test [19] for hypergraphs. We then argue that using explicit feature representations from underlying graph structure as the composition of equivariant layers is at least as powerful as the F-GWL test. We then prove the second part of the theorem.

Dataset	$\lvert V\rvert$	$\lvert\xi\rvert$	$\lvert\mu\rvert$	$\lvert M\rvert$	d	C
CA-DBLP	2708	1072	0.79	4.2	1433	7
CA-Cora	43413	22535	1.03	4.7	1425	6
CC-Citeseer	2708	1579	1.16	3.0	1433	6
CC-Cora	3312	1079	1.03	3.2	3703	6
YelpRestaurant	50758	679302	26.7	6.6	1862	11
HouseCommittees	1290	341	0.52	34.7	1290	3
Cooking200	7403	2755	0.74	19.9	7403	20
News20	16342	100	0.01	654.5	100	4
CO-RTO	66790	27528	0.82	1.6	10	2

Table 3: Dataset statistics: $|V|$ and $|\xi|$ represent the number of hypernodes and hyperedges, respectively. $|\mu| = \frac{2|\xi|}{|V|}$ is. the average hyperedge density, and $|M| = \frac{1}{|\xi|}\sum_{e\in\xi}|e|$ is the average hypernode density of hyperedges; d and C are the dimension of features and the number of classes, respectively.

5 EXPERIMENTS

This section presents benchmark datasets, baseline methods, comparative analyses and ablation studies of our model.

Benchmark Datasets. We experiment with publicly available benchmark datasets – Cora, DBLP, and Citeseer, extensively used in previous studies [18, 41]. Two standard ways to construct hypergraphs from these datasets result in coauthorship (CA) networks from Cora and DBLP and cocitation networks (CC) from Cora and Citeseer. Moreover, we also benchmark on the pre-constructed standard hypergraph datasets – YelpRestaurant, HouseCommittees, Cooking200, and News20 [8]. We utilize the hypergraph convolution layers from the DeepHyperGraph (DHG) package for our baseline experiments. Table 3 provides detailed statistics of these datasets.

Baseline Methods. We compare DPHGNN with seven standard baselines designed for semi-supervised node classification on hypergraphs. The baselines include a mix of spatial and spectral approaches for hypergraph representation learning – (i) HGNN: It parametrizes the smoothing of features from HGNN Laplacian matrix [12]; (ii) HGNN+: It uses adaptive hyperedge group fusion strategy to introduce spatial message passing for different modalities [13]; (iii) HyperGCN: It introduces Laplacian operator to approximate underlying graph structure and perform message passing over lower-order graph [41]; (iv) HNHN: Spatial HGNN with nonlinear activation functions are applied over both hypernodes and hyperedges [10]; UniGNN [41] generalises spatial learning GNNs for hypergraphs and formulates various architectures such as (v) UniGCN, (vi) UniSAGE, and (vii) UniGAT; (viii) HENNs [17]: It exploits multiple spectrally-similar graph representations of hypergraphs; (ix) HyGNN [33]: Spatial HGNN with attention based hypergraph encoder for optimal hyperedge prediction.

Experimental Setup. For DPHGNN, we deploy a dual-layer training mechanism (see Figure 4 in Supplementary), in which the static layer is trained to learn the underlying graph structural and hypergraph spectral inductive biases, and the dynamic layer is used to optimize learning on hypergraphs. We use a suitable GNN layer to the decomposed structure for graph structural learning and update the feature representation for the 2-hop neighborhood. Moreover, we use two-layer message passing for hypergraph structural learning, one for updating the neighbor hypernode features and the other for a class-wise prediction head. Table 10 in Supplementary includes specific hyperparameter details for our experiments.

5.1 Performance Comparison

We evaluate the performance of the model based on the mean accuracy, macro F1 score and micro F1 score. We report the average performance of the model over ten runs, along with the standard deviation.

5.1.1 Comparative Analysis. Table 4 presents the comparative analysis. We observe that spatial HGNN methods outperform the spectral ones for co-authorship datasets (by a margin of ∼5%). However, it is the opposite in the case of co-citation datasets in which spectral methods stand out (by a significant margin of 7%-9%).

Our proposed DPHGNN method, with TAA, SIB, and DFF+HGNN modules, is robust to the change in hypergraph topology. We quantify the performance improvement in DPHGNN by comparing it with the best-performing spatial HGNN on co-authorship datasets. DPHGNN outperforms UniGCN, UniSAGE, and UniGAT by 1.1%, 1.2%, and 2.3%, respectively, on CA-DBLP, and 3.4%, 2.5%, and 2.6%, respectively, on CA-Cora. For the co-citation datasets, with the induction of SIB and TAA (spectral), DPHGNN outperforms spatial UniGNNs (UniGCN, UniSAGE, UniGAT) by 6.3%, 4.2%, and 9.6%, respectively. In CC-Citeseer, DPHGNN outperforms the best baseline by 2.2% and gives the best performance on the F1-Score within the standard deviation range (std). On CC-Cora, HGNN, being a fully spectral baseline, surpasses DPHGNN by a small margin of 1.8%; however, DPHGNN outperforms the rest of the spatial baselines, namely UniGCN, UniSAGE and UniGAT, with a significant margin of 6.4%, 2.6%, 11.6%, respectively. For HouseCommittees, UniGNNs surpass DPHGNN by a small margin of 2.8% (within mean and std range). On YelpRestaurant, Cooking200, and News20 with dense hypernode distributions, DPHGNN outperforms best spatial baselines (HGNN+, UniSAGE, UniGAT) with 2.9%, 3.3%, 0.8% and best spectral baselines (HGNN, HGNN, HyperGCN) with 5.5%, 2.3%, 1.4% respectively. Moreover, in Table 6, we present a performance comparison between fully-spatial (HyGNN) and spectral (HEGNNs) baselines; DPHGNN outperforms with significant absolute differences of 1.93% and 3.06% for HENN and HyGNN, respectively. The ablation study below empirically validates the effect of each module in improving the performance of DPHGNN.

Embedding Update Visualization. The performance of a classification model is inherently dependent on interpolating the underlying distribution and clustering the data points for respective classes. Figure 2 provides the visualization of the t-SNE [35] of the embedding update of DPHGNN and that of the best baseline (HGNN). We observe the former segregating classes more effectively than the latter.

5.1.2 Ablation Study. We study the impact of each module of DPHGNN and report the change in the model's performance with the structure of hypergraph datasets in Table 7.

- On co-authorship datasets (CA-DBLP, CA-Cora), most impact is due to spatial information flow. Without TAA, the performance of DPHGNN decreases (1.3% ↓, 1.5% ↓); however, DFF propagates the lower-order structural information. Without DFF (2.5% ↓,

Methods	CA-DBLP			CA-Cora			CC-Citeseer			CC-Cora		
	Mean Acc	Macro F1	Micro F1	Mean Acc	Macro F1	Micro F1	Mean Acc	Macro F1	Micro F1	Mean Acc	Macro F1	Micro F1
HGNN	68.24±9.8	67.61±8.5	68.24±9.9	63.92±4.2	53.88±5.4	64.01±5.2	68.56±0.3	60.52±1.1	68.56±0.2	69.38±3.8	69.03±3.9	68.92±5.4
HGNN+	84.66±5.1	81.12±1.8	84.66±5.1	68.76±4.8	61.90±5.2	69.76±3.6	42.02±5.3	37.21±3.0	42.02±6.2	46.26±5.2	49.3±3.2	46.26±5.2
HyperGCN	84.03±5.2	83.29±4.2	84.01±6.2	63.91±7.1	51.22±9.2	68.99±5.5	63.2±1.8	60.24±2.2	63.2±1.1	67.07±2.1	66.52±4.9	67.87±2.1
HNHN	65.32±1.4	61.09±2.3	65.32±1.5	65.28±4.2	54.11±7.3	68.91±4.8	33.45±4.5	31.65±3.9	33.45±4.5	48.03±4.3	42.55±0.5	48.03±4.3
UniGCN	88.07±0.1	86.03±0.4	88.07±0.2	71.68±4.4	57.31±6.2	79.74±1.3	52.47±8.2	47.54±4.8	52.47±8.2	61.1±7.4	55.24±8.2	61.1±7.4
UniSAGE	87.97±0.2	85.37±0.1	87.96±0.4	72.54±3.4	57.26±6.4	78.69±3.4	57.59±7.5	55.2±3.2	57.59±7.2	64.87±6.0	63.26±1.5	64.87±6.0
UniGAT	86.69±0.1	84.24±1.8	87.97±1.1	72.48±2.8	58.69±4.5	75.2±1.8	52.68±8.4	46.95±2.9	51.96±8.4	55.83±6.5	51.58±4.2	55.83±6.5
DPHGNN	89.06±1.2	86.48±2.5	87.16±1.8	75.12±1.2	68.44±1.9	78.94±3.3	65.77±4.2	56.52±3.5	63.77±5.4	67.51±6.2	64.55±8.2	68.51±3.4

Methods	Yelp Restaurant			House Committees			Cooking200			News20		
	Mean Acc	Macro F1	Micro F1	Mean Acc	Macro F1	Micro F1	Mean Acc	Macro F1	Micro F1	Mean Acc	Macro F1	Micro F1
HGNN	30.69±3.2	14.37±6.7	28.24±2.8	52.33±5.1	50.06±5.5	50.03±3.2	53.45±2.6	40.91±5.8	50.83±2.9	80.46±3.2	78.74±0.4	80.46±2.4
HGNN+	32.92±1.8	12.84±4.2	30.53±1.5	53.1±5.9	52.58±4.2	51.27±3.5	49.64±1.2	40.07±8.6	48.27±1.5	81.25±2.8	78.89±1.1	81.25±1.2
HyperGCN	30.77±1.1	13.3±2	28.41±3.1	NA	NA	NA	9.32±2.8	4.28±1.6	6.64±2.6	81.16±0.8	78.95±0.5	80.16±1.8
HNHN	26.61±4.4	7.64±1.2	24.75±2.5	49.61±4.2	47.96±1.4	46.68±4.2	25.29±4.7	20.1±4.7	23.33±3.3	76.87±5.4	78.75±0.3	81.13±0.6
UniGCN	28.56±2.5	10.38±0.9	26.6±4.2	56.04±5.5	54.87±2.5	53.32±4.8	49.49±3.3	38.53±1.6	47.43±1.4	80.97±0.5	78.82±0.1	80.97±0.5
UniSAGE	27.3±0.4	10.41±1.5	25.93±1.6	61.72±8.4	50.87±6.7	61.21±0.5	52.32±3.9	42.24±2.5	49.64±2.2	81.49±1.5	79.47±1.8	81.49±0.3
UniGAT	OOM	OOM	OOM	57.36±3.1	56.31±0.2	54.73±2.9	34.99±1.8	26.17±3.6	33.73±0.9	78.76±4.4	78.15±0.3	80.76±3.3
DPHGNN	35.82±1.4	20.55±0.3	36.33±1.2	56.87±4.3	51.72±0.3	54.33±3.3	55.74±2.1	46.51±1.5	51.73±3.4	81.57±1.3	79.2±0.5	81.43±0.3

Table 4: Performance comparison on the benchmark hypergraph datasets over mean accuracy(%), macro F1-score, and micro F1-score (± standard deviation). For every dataset, the best performance is highlighted in red, the best baseline is highlighted in blue, results within 1-std. dev. are highlighted in brown. NA indicates structural restriction for method; OOM is out of memory.

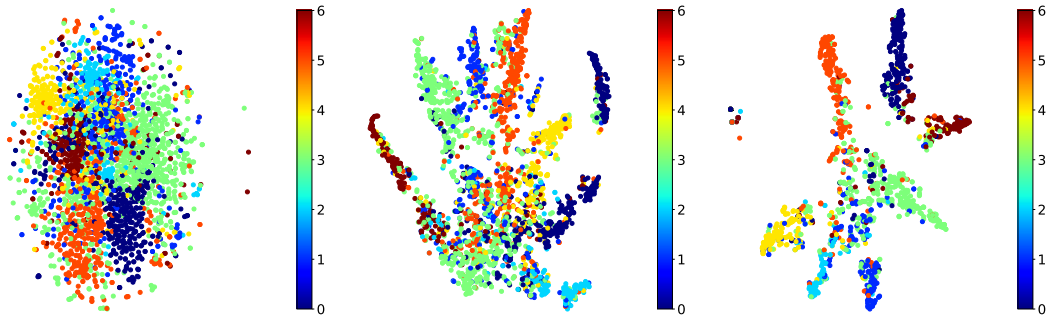


Figure 2: Visualization of feature embedding update on the CC-Citeseer dataset (6 classes) – initial embedding (left), HGNN embedding update (middle), DPHGNN embedding update (right).

2.7% ↓), TAA can only provide cross-attention information of node entities as inductive bias.

- On co-citation datasets, CC-Citeseer, and CC-Cora, the TAA (spectral) and SIB modules of DPHGNN have the majority of contributions, without which the performance of DPHGNN deteriorates by 6.3% ↓, and 5.9% ↓, respectively. This supports the empirical analysis of spectral HGNNs performing better than spatial baselines over co-citation datasets. Without TAA, the performance of DPHGNN degrades (2.9% ↓, 3.3% ↓) as TAA propagates spectral inductive biases.
- On HouseCommittees and Cooking200, features are initialised with one-hot encoding of node indices. Therefore, TAA and SIB have the most impact on empowering features with spatial and spectral inductive biases; the performance deteriorates without TAA (4.1% ↓, 5.7% ↓) and without SIB (3.2% ↓, 6.2% ↓). Moreover, due to sparse connectivity ($|\mu| \sim 0.5, 0.7$) with dense hypernode density ($|M| \sim 34, 19$) TAA (spectral) complements in the absence of SIB, and TAA (spatial) for DFF.
- On News20, most of the HGNNs suffer from over-smoothing of features as the highly dense hypernodes ($|M| \sim 654$) diffuse features via highly sparse hyperedges ($|\mu| \sim 0.01$), resulting in similar performance irrespective of HGNN architecture. Without SIB (4.5% ↓), DPHGNN struggles to break automorphisms in this symmetric structure.
- On YelpRestaurant, with densely connected hypergraph, the performance drops without TAA and DFF (5.2% ↓, 3.3% ↓) as they provide a major contribution in the spatial message flow. Without SIB (1.3% ↓), TAA (spectral) provides desired inductive biases.

We also study the model's performance with increased message-passing blocks (i.e., the TAA and DFF blocks). The DFF+HGNN layers in Table 8 indicate the increase in hypergraph message-passing layers; DFF stays constant throughout the layer increment. Moreover, the spectral inductive biases remain unchanged. However, we observe that with the increase in HGNN layers (4 to 8), the performance drops significantly (5.1% ↓, 30% ↓); over-smoothing of features [25] remains the primary reason behind the performance drop with the increase in layers.

5.1.3 Convergence Analysis. Figure 3 shows the significance of topological structure for the loss convergence in HGNN. We observe that the loss function tends to have a weak convergence for the co-citation networks compared to co-authorship networks.

Method	Mean Acc	Macro F1	Micro F1
HGNN	80.51±1.7	45.01±0.6	80.51±1.6
HGNN+	80.47±0.9	44.74±1.5	80.47±0.9
HyperGCN	80.50±0.6	44.59±0.2	80.49±0.6
HNHN	80.49±0.8	44.60±0.2	80.49±0.8
UniGCN	80.50±2.6	44.86±3.2	80.49±2.8
UniSAGE	81.66±2.9	46.66±3.1	81.66±2.9
UniGAT	80.48±0.6	44.70±0.4	80.48±0.6
DPHGNN	86.02±1.2	51.39±0.8	86.02±1.2

Table 5: Performance comparison on the CO-RTO dataset. The best performance is highlighted in red, and the best baseline is highlighted in blue.

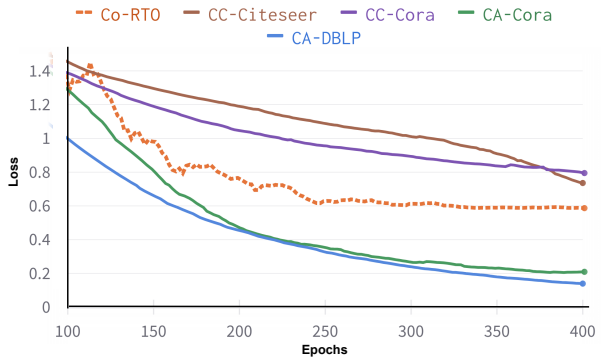


Figure 3: Convergence analysis of DPHGNN over benchmark hypergraph and CO-RTO datasets.

Datasets	Overall	w/o TAA	w/o SIB	w/o DFF
CA-DBLP	89.06	87.72	88.65	86.54
CA-Cora	75.12	73.6	74.52	72.34
CC-Citeseer	65.77	63.86	59.47	64.05
CC-Cora	67.51	64.12	61.15	65.98
YelpRestaurant	35.82	30.56	34.69	32.45
HouseCommitees	56.87	52.86	53.66	55.68
Cooking200	55.74	50.04	49.53	53.09
News20	81.07	80.23	76.55	80.33
CO-RTO	51.39	48.32	46.75	49.12

Table 7: Ablation study to measure the impact of three blocks of DPHGNN – Topology-Aware Attention (TAA), Spectral Inductive Biases (SIB), and Dynamic Feature Fusion (DFF). For benchmark datasets, we report mean accuracy (%); since the CO-RTO dataset is imbalanced, we report macro F1-score.

Datasets	TAA Layers			DFF Layers		
	2-GCN	4-GCN	8-GCN	2-HGNN	4-HGNN	8-HGNN
CA-DBLP	89.06	88.34	86.95	89.06	88.92	46.03
CA-Cora	75.12	74.85	73.8	75.12	69.01	32.54
CO-Citeseer	65.77	64.98	63.94	65.77	59.82	25.86
CO-Cora	67.51	66.02	64.22	67.51	58.25	28.05
CoRTO	51.39	49.95	48.38	51.39	28.67	21.03
YelpRestaurant	35.82	34.23	30.23	35.82	30.37	15.35
HouseCommitees	56.87	55.01	51.95	56.87	50.4	20.91
Cooking200	55.74	54.02	50.6	55.74	49.59	18.95
News20	81.07	79.88	76.78	81.07	74.55	39.26

Table 8: Ablation study to measure the impact of increment in message-passing layers over the TAA and DFF blocks.

Dataset \Baseline	HENN	HyGNN	DPHGNN
CA-DBLP	85.64±4.7	88.01±0.4	89.06±1.2
CA-Cora	72.33±3.5	70.42±1.5	75.12±1.2
CC-Citeseer	68.15±2.2	55.86±0.5	65.77±4.2
CC-Cora	70.56±5.6	59.67±4.1	67.51±6.2
Yelp Restaurant	33.47±1.8	33.43±2.4	35.82±1.4
House Committees	54.83±4.9	59.99±3.2	56.87±4.3
Cooking200	50.09±5.5	50.42±4.6	55.74±2.1
News20	81.32±0.6	80.33±2.5	81.57±1.3
CO-RTO	45.07±2.8	48.23±0.9	51.39±0.8

Table 6: Performance comparison (Mean Accuracy, for CO-RTO: Macro F1) between HENN, HyGNN, and the proposed DPHGNN. The best performance is highlighted in red.

Model	Iso HG	Non-Iso HG
HGNN	55.23	53.25
HGNN++	40.89	63.69
HyperGCN	57.31	58.25
HNHN	38.85	77.3
UniGCN	44.83	80.55
UniSAGE	46.35	85.00
UniGAT	48.84	84.21
HENN	53.54	75.26
HyGNN	51.63	80.64
DPHGNN	73.38	87.58

Table 9: Performance comparison on the Isomorphic and Non-Isomorphic datasets. The best performance is highlighted in red, and the best baseline is highlighted in blue.

Performance on Hypergraph Isomorphism Test. To validate our theoretical framework behind equivariant operator learning (EOL) within the DPHGNN model, we constructed two synthetic datasets; *"Isomorphic (Iso) HG" and "Non-Isomorphic (Non-Iso) HG"*. The classification results on these synthetic tasks are presented in Table 9. The results demonstrate that DPHGNN achieves significant improvements, which are attributed to its robust spectral and spatial inductive biases.

6 THE RETURN-TO-ORIGIN (RTO) TASK

This section describes modeling resource-constrained real-world data with hypergraph structure and reports the efficacy of DPHGNN for an industrial application – the Return-to-Origin (RTO) task.

Task Description. RTO refers to the situation where a product is returned to the point of origin instead of being delivered to the intended recipient [24]. A product being RTO-ed substantially impacts global e-commerce and encourages a vicious cycle of financial losses for merchants, complex fraudulent transactions, etc. We model RTO as a semi-supervised hypernode classification task. In collaboration with an e-commerce giant, we deploy our proposed DPHGNN to address the RTO task.

6.1 The CO-RTO Dataset

Here, we describe the carefully curated CO-RTO (Co-order RTO) hypergraph structure. An RTO involves various physical entities from an order transaction cycle (customer, product, supplier, and order) that can not be expressed using pairwise relations. This naturally motivates us to model higher-order interactions using

hyperedges, containing entities as hypernodes. The RTO prediction task boils down to a binary hypernode classification problem – predicting whether order-related nodes are RTOs. We obtained the CO-RTO dataset from our partner e-commerce company, Meesho (Table 3).

Hypergraph Construction. The CO-RTO dataset is an orderly transaction and entity relation hypergraph dataset. The raw data accompanies the feature information collected from partner e-commerce companies. It mainly comprises four major entities that play key roles in a complete order transaction cycle namely, user, supplier, courier, and product. In this work, we use an autoencoder for the conversion of heterogeneous features to homogeneous features to enable message flow. We formulate a hypergraph structure in the form of nodes that represent these entities with their corresponding feature vectors and family of hyperedges where each hyperedge comprises entity nodes from an order transaction flow.

6.2 Experimental Results

Table 5 reports the performance of the competing models for the RTO task. Due to sparse connectivity, symmetrical structure, and class imbalance in CO-RTO, baselines suffer from over-smoothing of features over the message propagation. DPHGNN achieves 51% macro-F1, outperforming other baselines by a significant margin (close to 7%). We further report critical observations below.

Symmetric and Sparse Incidence Structure. In the CO-RTO hypergraph, the edge connectivity is sufficiently sparse (avg. hypernode density $|M|$ is 1.6, hyperedge density $|\mu|$ is 0.82; c.f. Table 3), constraining message passing between subsequent layers in HGNN. The fixed cardinality in the hyperedges of the CO-RTO hypergraph and the highly symmetrical structure tend to increase the multiplicity of eigenvalues, which is inherently linked with the Cheeger constant [32] and the over-squashing phenomenon [34].

Automorphism. The hypergraph $HG = (V, \xi)$ of the CO-RTO dataset is a uniform hypergraph containing an automorphism group. More precisely, if π is a permutation function over hypernode set V of HG; $\pi : V \rightarrow V'$, with $|V| = |V'|$ and for each hyperedge $e \in \xi$ and $|e| = k$ (here $k = 4$). Here π satisfies $\pi(v) = v'$, where $v \in V$ and $v' \in V'$. Also $\forall e \in \xi \; \exists \; e' \in \xi$ such that $e' = \{\pi(v) | v \in e\}$. Now as every hyperedge $e_i \in \xi$, the image of e_i under π is also a hyperedge in ξ. For example: if $e_i = \{v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}\}$ then $\pi(e_i) = \{\pi(v_{i,1}), \pi(v_{i,2}), \pi(v_{i,3}), \pi(v_{i,4})\}$ is also in ξ. Given the general spatial HGNN message-passing framework, automorphism in hypergraph preserves the incidence relations between vertices u and edge e as $HG_{u,e} = HG_{\pi(u),\pi(e)}$. Given two hypernodes, (a, b) under automorphism π are $a = \phi(b)$, their feature update equations 5 and 6 are:

$$\hat{x}_a^{(l)} = \text{Update}^{(l)}\left(x_a^{(l)}, \text{Aggr}^{(l)}\left(\left\{\left\{\text{Aggr}^{(l-1)}\left(x_u^{(l-1)}\right) \forall u \in e\right\} \forall e \in E_a\right\}\right)\right) \tag{5}$$

$$\hat{x}_b^{(l)} = \text{Update}^{(l)}\left(x_b^{(l)}, \text{Aggr}^{(l)}\left(\{\{\text{Aggr}^{(l-1)}(x_{\pi(u)}^{(l-1)}) \forall \pi(u) \in \pi(e)\} \forall \pi(e) \in E_b\}\right)\right) \tag{6}$$

Hence given the automorphism π, the feature update leads to $\hat{x}_a^{(l)} = \hat{x}_b^{(l)}$. Therefore, the spatial HGNNs bounded by 1-GWL expressivity perform sub-optimally due to symmetry-induced automorphism groups. To our knowledge, our method (DPHGNN) is the first HGNN that breaks the automorphism groups and is 3-GWL expressive.

Over-smoothing. We perform experiments with dual-layer HGNNs. Still, the features get smoothed over message passing due to the sparse incidence structure. This is the main reason behind the suboptimal performance of baselines on CO-RTO. Along with propagating messages with skip connections [7], DPHGNN leverages static inductive priors to prevent over-smoothing of relevant feature information.

Class Imbalance. In CO-RTO, class imbalance occurs naturally (a ratio of 6568:26827 for RTO:Non-RTO), and HGNN tends to overfit the majority class. These observations lead to a drastic performance drop with the hypergraph-based MPNN models.

6.3 Deployment Workflow

This section presents a brief description of the deployment of DPHGNN in Meesho's large-scale e-commerce platform with a transductive setting [9].

Data Pipelines. The large-scale e-commerce transaction data is pulled through the Apache Spark streaming service [1] and distributed across multiple clusters to compute the homogeneous hypernode features and specify user, product, supplier, and courier entities inside a co-order hyperedge. These node features and edges are hosted in a feature store infrastructure built on top of a data lake and partitioned for efficient read and write operations.

Test-Train Hypergraph creation. Based on this model's inference, some of the orders get actioned. Therefore, the possible outcome of RTO remains unknown due to the model's action, and hence, data is not useful for future model training. To resolve this, a holdout sample is maintained where no actions are taken and are used as training data for transductive training.

Model Building and Inference. In the current transductive setting, the target variable is only available for the past holdout transactions; the model is trained only on those hypernodes and hyperedges using training masks. All the edges and nodes go through the forward pass. Once the model is trained, the predictions for the latest transactions are collected and stored. We used NVIDIA T4 GPU clusters with 20 workers for network training and inference. The orders flagged are published in a Kafka queue [22] to be consumed and blocked by the order management system.

7 CONCLUSION

In this paper, we proposed DPHGNN, a novel hypergraph message-passing approach for representation learning. It explicitly learns static and dynamic feature representations from multiple views of lower-order graph topology. DPHGNN outperformed seven SOTA baselines over eight benchmark datasets. We also showed its expressive power theoretically. DPHGNN was also tested on a new e-commerce dataset (CO-RTO) and deployed by our industry partner for RTO prediction.

ACKNOWLEDGEMENT

This research was financially supported by Meesho. We thank them for their generous support in providing the CO-RTO dataset and their overall contributions to our work.

REFERENCES

[1] Michael Armbrust, Tathagata Das, Joseph Torres, Burak Yavuz, Shixiong Zhu, Reynold Xin, Ali Ghodsi, Ion Stoica, and Matei Zaharia. 2018. Structured streaming: A declarative api for real-time applications in apache spark. In *Proceedings of the 2018 International Conference on Management of Data*. 601–613.

[2] Devanshu Arya, Deepak K. Gupta, Stevan Rudinac, and Marcel Worring. 2020. HyperSAGE: Generalizing Inductive Representation Learning on Hypergraphs. *CoRR* abs/2010.04558 (2020). http://dblp.uni-trier.de/db/journals/corr/corr2010.html#abs-2010-04558

[3] Song Bai, Feihu Zhang, and Philip H. S. Torr. 2019. Hypergraph Convolution and Hypergraph Attention. *CoRR* abs/1901.08150 (2019). http://dblp.uni-trier.de/db/journals/corr/corr1901.html#abs-1901-08150

[4] Sambaran Bandyopadhyay, Kishalay Das, and M. Narasimha Murty. 2020. Hypergraph Attention Isomorphism Network by Learning Line Graph Expansion. In *2020 IEEE International Conference on Big Data (Big Data)*. 669–678. https://doi.org/10.1109/BigData50022.2020.9378335

[5] Alain Bretto. 2013. *Hypergraph Theory*. Springer, Heidelberg. https://doi.org/10.1007/978-3-319-00080-0

[6] Jan Böker. 2019. Color Refinement, Homomorphisms, and Hypergraphs.. In *WG (Lecture Notes in Computer Science, Vol. 11789)*, Ignasi Sau and Dimitrios M. Thilikos (Eds.). Springer, 338–350. http://dblp.uni-trier.de/db/conf/wg/wg2019.html#Boker19

[7] Guanzi Chen, Jiying Zhang, Xi Xiao, and Yang Li. 2022. Preventing Over-Smoothing for Hypergraph Neural Networks. arXiv:2203.17159 [cs.LG]

[8] Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. 2022. You are AllSet: A Multiset Function Framework for Hypergraph Neural Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=hpBTIv2uy_E

[9] Giorgio Ciano, Alberto Rossi, Monica Bianchini, and Franco Scarselli. 2021. On inductive–transductive learning with graph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 2 (2021), 758–769.

[10] Yihe Dong, Will Sawin, and Yoshua Bengio. 2020. HNHN: Hypergraph Networks with Hyperedge Neurons. *ArXiv* abs/2006.12278 (2020). https://api.semanticscholar.org/CorpusID:219965680

[11] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The world wide web conference*. 417–426.

[12] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph Neural Networks.. In *AAAI*. AAAI Press, 3558–3565. http://dblp.uni-trier.de/db/conf/aaai/aaai2019.html#FengYZJG19

[13] Yue Gao, Yifan Feng, Shuyi Ji, and Rongrong Ji. 2022. HGNN+: General Hypergraph Neural Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45 (2022), 3181–3199. https://api.semanticscholar.org/CorpusID:249644940

[14] Yue Gao, Zizhao Zhang, Haojie Lin, Xibin Zhao, Shaoyi Du, and Changqing Zou. 2022. Hypergraph Learning: Methods and Practices. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 5 (2022), 2548–2566. https://doi.org/10.1109/TPAMI.2020.3039374

[15] Xuemei Gu, Lijun Chen, and Mario Krenn. 2020. Quantum experiments and hypergraphs: Multiphoton sources for quantum interference, quantum computation, and quantum entanglement. *Physical Review A* 101, 3 (mar 2020). https://doi.org/10.1103/physreva.101.033816

[16] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*. 11.

[17] Mikhail Hayhoe, Hans Matthew Riess, Michael M. Zavlanos, VICTOR PRECIADO, and Alejandro Ribeiro. 2023. Transferable Hypergraph Neural Networks via Spectral Similarity. In *The Second Learning on Graphs Conference*. https://openreview.net/forum?id=cHuii4NOB9

[18] Jing Huang and Jie Yang. 2021. UniGNN: a Unified Framework for Graph and Hypergraph Neural Networks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, Zhi-Hua Zhou (Ed.). ijcai.org, 2563–2569. https://doi.org/10.24963/ijcai.2021/353

[19] Ningyuan Teresa Huang and Soledad Villar. 2021. A short tutorial on the weisfeiler-lehman test and its variants. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 8533–8537.

[20] Shuyi Ji, Yifan Feng, Rongrong Ji, Xibin Zhao, Wanwan Tang, and Yue Gao. 2020. Dual channel hypergraph collaborative filtering. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2020–2029.

[21] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. http://arxiv.org/abs/1609.02907 cite arxiv:1609.02907Comment: Published as a conference paper at ICLR 2017.

[22] Jay Kreps, Neha Narkhede, Jun Rao, et al. 2011. Kafka: A distributed messaging system for log processing. In *Proceedings of the NetDB*, Vol. 11. Athens, Greece, 1–7.

[23] Guillaume Lachaud, Patricia Conde-Cespedes, and Maria Trocan. 2022. Graph neural networks-based multilabel classification of citation network. In *Asian Conference on Intelligent Information and Database Systems*. Springer, 128–140.

[24] Rashmee Lahon. 2022. Return to Origin- Why it Happens, Its Impact, and How to Solve it? https://razorpay.com/blog/return-to-origin-challenges-and-how-to-solve-it

[25] Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[26] Fan Liang, Cheng Qian, Wei Yu, David W. Griffith, and Nada Golmie. 2022. Survey of Graph Neural Networks and Applications. *Wireless Communications and Mobile Computing* (2022). https://api.semanticscholar.org/CorpusID:251192566

[27] Shengyuan Liu, Pei Lv, Yuzhen Zhang, Jie Fu, Junjin Cheng, Wanqing Li, Bing Zhou, and Mingliang Xu. 2020. Semi-Dynamic Hypergraph Neural Network for 3D Pose Estimation.. In *IJCAI*, Christian Bessiere (Ed.). ijcai.org, 782–788. http://dblp.uni-trier.de/db/conf/ijcai/ijcai2020.html#LiuLZFCLZX20 Scheduled for July 2020, Yokohama, Japan, postponed due to the Corona pandemic..

[28] Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. 2019. Provably powerful graph networks. *Advances in neural information processing systems* 32 (2019).

[29] Christopher Morris, Gaurav Rattan, and Petra Mutzel. 2020. Weisfeiler and Leman go sparse: Towards scalable higher-order graph embeddings. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 21824–21840. https://proceedings.neurips.cc/paper/2020/file/f81dee42585b3814de199b2e88757f5c-Paper.pdf

[30] Raffaella Mulas, Christian Kuehn, Tobias Böhle, and Jürgen Jost. 2021. Random walks and Laplacians on hypergraphs: When do they match? arXiv:2106.11663 [math.SP]

[31] Raffaella Mulas, Rubén J Sánchez-Garcıa, and Ben D MacArthur. 2020. Hypergraph automorphisms. *arXiv preprint arXiv:2010.01049* (2020).

[32] Raffaella Mulas and Dong Zhang. 2021. Spectral theory of Laplace operators on oriented hypergraphs. *Discrete Mathematics* 344, 6 (jun 2021), 112372. https://doi.org/10.1016/j.disc.2021.112372

[33] Khaled Mohammed Saifuddin, Bri Bumgardnerr, Farhan Tanvir, and Esra Akbas. 2022. HyGNN: Drug-Drug Interaction Prediction via Hypergraph Neural Network. *2023 IEEE 39th International Conference on Data Engineering (ICDE)* (2022), 1503–1516. https://api.semanticscholar.org/CorpusID:250072419

[34] Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. 2022. Understanding over-squashing and bottlenecks on graphs via curvature. arXiv:2111.14522 [stat.ML]

[35] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[36] Petar Velickovic, Guillem Cucurull, A. Casanova, A. Romero, P. Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *ArXiv* abs/1710.10903 (2018).

[37] Janu Verma, Srishti Gupta, Debdoot Mukherjee, and Tanmoy Chakraborty. 2019. Heterogeneous edge embedding for friend recommendation. In *Advances in Information Retrieval: 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14–18, 2019, Proceedings, Part II 41*. Springer, 172–179.

[38] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How Powerful are Graph Neural Networks?. In *ICLR*. OpenReview.net. http://dblp.uni-trier.de/db/conf/iclr/iclr2019.html#XuHLJ19

[39] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation Learning on Graphs with Jumping Knowledge Networks.. In *ICML (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer G. Dy and Andreas Krause (Eds.). PMLR, 5449–5458. http://dblp.uni-trier.de/db/conf/icml/icml2018.html#XuLTSKJ18

[40] Naganand Yadati. 2020. Neural Message Passing for Multi-Relational Ordered and Recursive Hypergraphs. In *Advances in Neural Information Processing Systems (NeurIPS) 33*. Curran Associates, Inc.

[41] Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. 2019. Hypergcn: A new method for training graph convolutional networks on hypergraphs. *Advances in neural information processing systems* 32 (2019).

[42] Kai Yang, Hao Sun, Chunbo Zou, and Xiaoqiang Lu. 2021. Cross-attention spectral–spatial network for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), 1–14.

[43] Jonathan S. Yedidia. 2011. Message-Passing Algorithms for Inference and Optimization. *Journal of Statistical Physics* 145 (2011), 860–890. https://api.semanticscholar.org/CorpusID:120991239

[44] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2018. Graph Neural Networks: A Review of Methods and Applications. *ArXiv* abs/1812.08434 (2018). https://api.semanticscholar.org/CorpusID:56517517

[45] J.Y. Zien, M.D.F. Schlag, and P.K. Chan. 1999. Multilevel spectral hypergraph partitioning with arbitrary vertex sizes. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 18, 9 (1999), 1389–1399. https://doi.org/10.1109/43.784130

A SUPPLEMENTARY MATERIAL

A.1 Proof of Proposition

Proof. We generalize the work of [6, 28] on hypergraph color refinement from (see Eq. (4) in the main text) and using tensors to encode the color, for hypergraphs having k vertices inside an edge $HC \in \Sigma^{n^k}$, as the color of a node is induced by the coloring of its edges by a map $f_H(e) : V^k \to \Sigma$. Therefore, the tensor $HC \in \mathbb{R}^{n^k \times a}$ encodes k vertices inside an edge $i \in [n]^k$ with $HC_i \in \Sigma$ and color $\Sigma \in \mathbb{R}^a$. Initialising $HC_0^H(v) = 1$ for all $v \in V(H)$, the color refinement is given by:
$HC_{t+1}^H(v) = \text{HASH}\{\{\{HC_t^H(u) \mid u \in f_H(e)\} \mid e \in E_i \text{ with } v \in f_H(e)\}\}$, $\{\{.\}\}$ represents multiset notation. H_1 and H_2 are non-isomorphic if

$$\left\{\left\{HC_i^{H_1}(v) \mid v \in V(H_1)\right\}\right\} \neq \left\{\left\{HC_i^{H_2}(v) \mid v \in V(H_2)\right\}\right\}$$

We start by proving for any HGNN of the form of Eq1, there exists an injective mapping Γ between every iteration of color refinement and the layer of message propagation such that $\Gamma^{(t,l)}\left(HC_v^t\right) \to \left(h_v^l\right)$.

Assume that injective mapping $\Gamma^{(t,l)}$ exists. For $(t, l) = 0$ trivially, both tensors are initially assigned some value under injectivity HC_v^0, h_v^0.

For $(t+1, l+1)$, the feature update is given by,

$$h_v^{l+1} = \phi_3\left(h_{v'}^l, \phi_2\left\{\left\{\phi_1\left(\left\{\left\{h_u^l \mid u \in e\right\}\right\}\right) \mid e \in E_i\right\}\right\}\right)$$

as ϕ_1, ϕ_2, ϕ_3 are all injective, they could be composed into an injective function φ, therefore,

$$\begin{aligned}
&= \varphi\left(h_v^l, \left\{\left\{\left(\left\{\left\{h_u^l \mid u \in e\right\}\right\}\right) \mid e \in E_i\right\}\right\}\right) \\
&= \varphi\left(\Gamma^{(t,l)}\left(HC_v^t\right), \left\{\left\{\left(\left\{\left\{\Gamma^{(t,l)}\left(HC_u^t\right) \mid u \in e\right\}\right\}\right) \mid e \in E_i\right\}\right\}\right) \\
&= \Gamma^{(t,l+1)}\left(HC_{v'}^t, \{\{(\{\{(HC_u^t) \mid u \in e\}\}) \mid e \in E_i\}\}\right) \\
&= \Gamma^{(t+1,l+1)}\left(HC_v^{t+1}\right)
\end{aligned}$$

Where injective function $\Gamma^{(t+1,l+1)}$ is induced by $\varphi, \Gamma^{(t,l)}$. Hence by principles of induction, we can say, there exists an injective function $\Gamma^{(t,l)}$ at time t and layer l such that $\Gamma^{(t,l)}\left(HC_v^t\right) \to \left(h_v^l\right)$.

Therefore, at iteration T, if $HC_{(t)}(V_1, H_1) \neq HC_{(t)}(V_2, H_2)$, then $\left\{\left\{h_{v,H_1}^T\right\}\right\}_{v \in V_1} \neq \left\{\left\{h_{v,H_2}^T\right\}\right\}_{v \in V_2}$. □

A.2 Proof of Theorem 1

Proof. **Part 1:** We first establish that a function $g_\theta \in F^{\text{DPHGNN}}$ belongs to a family of k-order invariant networks.

Let S_n denote the permutation group on n elements. In the dynamic feature fusion (DFF), dynamic features $m_p^{(l)} = \phi_{mask}[\text{L}]$ where $p \in V_*$ comes from the update rule:
$L = \left[\text{Aggr}^{(l)}\left(\left\{m_u^{(l-1)} \forall u \in N(p)\right\}\right)\right]$. The function ϕ_{mask}, is bijective; therefore, the updated representations satisfy $\text{L}(f \cdot m_p^{(l)}) = f \cdot L(m_p^{(l)})$, $\text{f} \in S_n$. Hence, the update mechanism of DPHGNN follows the form (a k-order invariant graph network) $F = m \circ h \circ L_d \circ \sigma \circ \cdots \circ \sigma \circ L_1$, where h is an invariant linear layer, σ is non-linear activation function and L_i are equivariant layers, m is the multilayer perceptron (FC). [28], in Theorem 1, proved the expressive power of k-order invariant graph network equivalent to the k-WL test (for graphs).

Part 2: We quantify the expressive power of DPHGNN as a 1-FWL test. The main difference is between the update rules of WL and FWL tests. FWL assigns distinct colorings to the k-tuple of nodes. The feature fusion mechanism in DPHGNN updates the hypernode aggregation as

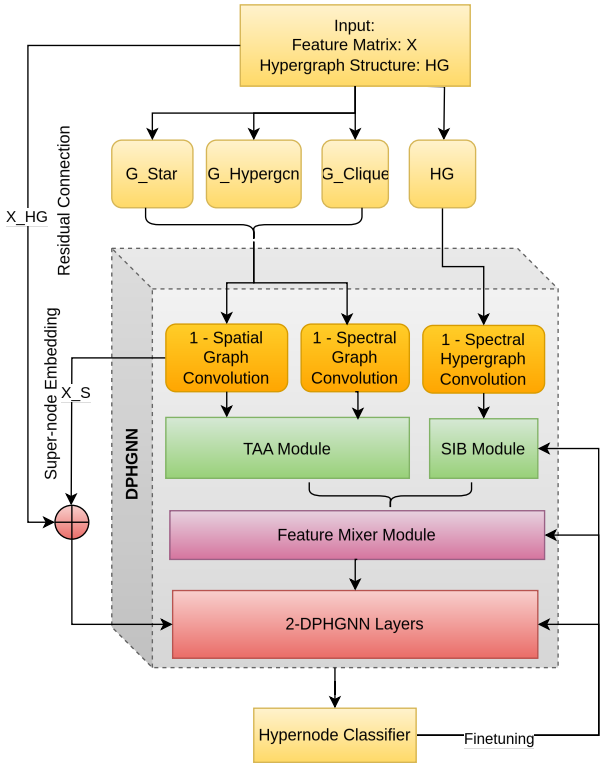


Figure 4: Generalized version of experimentation pipeline.

$$m_e^{(l-1)} = \text{Aggr}^{(l-1)}\left(\{\{m_u^{(l-1)} \forall u \in e\}\} \cup \{\{m_p^{(l)} \forall p \in E\}\}\right).$$

Therefore, now the tensor $\text{HC} \in \mathbb{R}^{(n^k \times a)+b}$ encodes the hypernodes, where coloring b is assigned by map $f_G(e) : V \to V'$ and $V' \in \mathbb{R}^b$. We formulate the FWL version of hypergraph color refinement as,
$\text{HC}_{t+1}^H(v) = \text{HASH}\{\{\{HC_t^H(u) \mid u \in \psi_H(e)\} \mid e \in E_i \text{with } v \in \psi_H(e)\}\}$, where map $\psi_H(e) := (f_G \circ f_H)(e)$. The update mechanism of the hypernodes in DPHGNN is given by:
$x_v^{(l)} = \text{Update}^{(l)}(\{x_v^{(l-1)}, \text{Aggr}^{(l)}\{m_e^{(l-1)} \forall e \in E_i\}\})$. The injectivity of Aggregation and Readout functions follows similarly from Proposition . The expressive power of k-FWL test is equivalent to $(k+1)$-WL for $k \geq 2$. The function $g_\theta \in F^{\text{DPHGNN}}$ is at least as powerful as the 3-GWL test. □

A.2.1 Convergence Analysis. Here, we describe the convergence analysis of DPHGNN for the standard hypergraph datasets and the CO-RTO dataset. Figure 3 reflects the behavior of cross-entropy loss over the change in hypergraph topology. The convergence for co-citation is slow as the spectral information is propagated as inductive priors. However, DPHGNN achieves a fast convergence rate for the Co-authorship datasets as explicit lower-order information is propagated as dynamic feature fusion. To learn in resource-constrained settings for the Co-RTO dataset, the loss initially increases and converges better than in co-citation networks. This signifies the flow of inductive priors. Due to the sparse and highly symmetrical structure, the initial performance suffers from providing better representations, the static and dynamic fusion of lower-order information helps message propagation to achieve better convergence.

Datasets	Modules	lr	weight_decay	dropout	attention_heads	hidden	# layers
CA-Cora	GNN module	0.1	5.0E-04	0.2	-	64	2
	TAA module	0.001	0.001	0.5	2	32	1
	SIB module	0.01	0.0005	0.4	-	64	1
	DFF module	0.01	5.0E-04	0.5	-	64	2
CA-DBLP	GNN module	0.1	5.0E-05	0.6	-	64	2
	TAA module	0.1	0.001	0.5	2	32	1
	SIB module	0.01	0.0005	0.4	-	64	1
	DFF module	0.01	5.0E-04	0.6	-	64	2
CC-Cora	GNN module	0.001	0.001	0.5	-	64	2
	TAA module	0.01	5.0E-04	0.5	8	8	1
	SIB module	0.01	0.0005	0.4		64	1
	DFF module	0.01	5.0E-04	0.6	-	64	2
CC-Citeseer	GNN module	0.1	1.0E-05	0.9	-	64	2
	TAA module	0.001	0.001	0.5	8	8	1
	SIB module	0.01	0.0005	0.4	-	64	1
	DFF module	0.01	5.0E-04	0.6	-	64	2
CO-RTO	GNN module	0.01	5.0E-05	0.3	-	64	2
	TAA module	0.001	0.001	0.5	2	32	1
	SIB module	0.01	0.0005	0.4	-	64	1
	DFF module	0.01	5.0E-04	0.5	-	64	2
YelpRestaurant	GNN module	0.01	5.0E-05	0.3	-	64	2
	TAA module	0.001	0.001	0.5	2	64	1
	SIB module	0.01	0.0005	0.4	-	64	1
	DFF module	0.01	5.0E-04	0.5	-	64	2
HouseCommittees	GNN module	0.01	5.0E-05	0.3	-	64	2
	TAA module	0.001	0.001	0.5	4	32	4
	SIB module	0.01	0.0005	0.4	-	64	1
	DFF module	0.01	5.0E-04	0.5	-	64	2
Cooking200	GNN module	0.01	5.0E-05	0.3	-	64	2
	TAA module	0.001	0.001	0.5	8	64	1
	SIB module	0.01	0.0005	0.4	-	64	1
	DFF module	0.01	1.0E-03	0.5	-	64	2
News20	GNN module	0.01	1.0E-2	0.3	-	64	2
	TAA module	0.001	0.001	0.5	4	32	1
	SIB module	0.01	0.0005	0.4	-	64	1
	DFF module	0.01	0.01	0.5	-	64	2

Table 10: Hyperparameters used in our experiments.

Dataset	HGNN	HGNN+	HyperGCN	HNHN	UniGCN	UniSAGE	UniGAT	DPHGNN
CA-DBLP	605.9	401	563.4	358	432	455	502	528
CA-Cora	228.4	145	183.4	122	161	157	174	182
CC-CiteSeer	290.1	122	171.1	135	140	152	186	168
CC-Cora	301.4	167	205.4	130	174	193	201	220
Yelp Restaurant	570	194	495	185	218	223	247	312.2
House Committees	192	114	153	66	72	96	150	168
Cooking200	502	156	258	144	174	216	216	233
News20	406	138	210	108	102	108	156	174
CO-RTO	384	192	306	108	138	150	174	205

Table 11: Runtime analysis between proposed DPHGNN with baseline architectures (in seconds).

A.3 Implementation Details

Hyperparameter Details. Table 10 summarizes the hyperparameter details for each submodule of DPHGNN architecture. We used GridSearchCV (from the sklearn library) for hyperparameter tuning. The sub-modules of DPHGNN, namely (i) the GNN module comprises the graph convolution training of multiple views of decomposed graph topology (as described in the TAA section in the main paper); (ii) the TAA module contains a multi-head wrapper of cross-attention modules with specific heads for each of the hypergraph datasets; (iii) the SIB module is trained as a hybrid spectral convolution layer. For the model parameter optimizations, we use a specific parameter group for each of the submodules of the DPHGNN architecture.

Experimentation Details. For the semi-supervised hypernode classification task, we train the model on each dataset for 400 epochs and infer the trained model from the last epoch. We perform 50 iterations of DPHGNN on each dataset, with five train/test splits and ten different random seeds. Figure 4 presents a concise view of the training mechanism adopted by DPHGNN.