Evaluating Hypergraph Construction Strategies for Hypergraph Neural Networks

Rustem Izmailov
School of Computer Science
University of Windsor
Windsor, Canada
izmailor@uwindsor.ca

Nishant Gowthaman
School of Computer Science
University of Windsor
Windsor, Canada
gowthamn@uwindsor.ca

Gauthami Shirodkar
School of Computer Science
University of Windsor
Windsor, Canada
shirodkg@uwindsor.ca

Vimanga Umange
School of Computer Science
University of Windsor
Windsor, Canada
umange@uwindsor.ca

***Abstract*—This paper presents a comprehensive examination of the application of top-k densest subgraphs for enhancing hypergraph structures, analyzing its conditional effectiveness across various data domains. We evaluate the performance of three prominent hypergraph neural network models on both benchmark datasets and a real-world task of predicting movie profitability. Our methodology incorporates multiple techniques for constructing hypergraphs, including feature-based and text-based approaches. The experimental results reveal a significant dichotomy in the impact of top-k augmentation: while it improves movie prediction accuracy by up to 4.0 percent by reinforcing natural cluster structures, it exhibits mixed or negative effects on synthetic benchmark datasets where the original hypergraphs already represent optimal connectivity. Notably, we observe performance degradation on citation networks, where sparse connections carry meaningful signals. These findings provide practical guidelines for applying density-based augmentation, suggesting that its effectiveness is critically dependent on the inherent cluster-quality of the target domain. The work contributes both methodological insights for hypergraph learning and empirical validation of augmentation strategies across diverse real-world and benchmark scenarios.**

***Index Terms*—Hypergraph modeling, Hypergraph Neural Networks, Node Classification, Top-k Densest Subgraphs**

I. Introduction

Graph-structured data has become a fundamental component in machine learning for modeling relational systems, ranging from social networks to biological interactions. The conventional graph representation captures pairwise relationships through the use of edges:

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ defines connections.

A. Challenges in Hypergraph Learning

Traditional graphs possess a core limitation - they are constrained to modeling binary relationships. In contrast, numerous real-world phenomena, such as co-authorship networks, document-topic associations, and user-item interactions, inherently encompass higher-order connections among multiple elements concurrently. Representing these complex multi-way associations solely through pairwise edges frequently leads to a loss of information or an oversimplification of the underlying structure. [1].

To address this limitation, hypergraphs have emerged as a powerful alternative. A *hypergraph* is defined as a pair $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ of:

- $\mathcal{V}$ is a finite set of vertices,
- $\mathcal{E}$ is a set of hyperedges, each representing a subset of vertices ($e \subseteq \mathcal{V}$)

As illustrated in Figure 1, while traditional graphs are limited to pairwise connections between vertices, hypergraphs can simultaneously link multiple vertices through a single hyperedge. This flexible structure proves particularly valuable when processing heterogeneous data, as different data modalities or types can naturally generate distinct hyperedge categories that seamlessly integrate into a unified hypergraph representation. To process such data, **Hypergraph Neural Networks (HGNNs)**—extensions of Graph Neural Networks (GNNs)—have been developed to operate directly on hypergraph structures.

n contrast to conventional graph neural networks, hypergraph neural networks demonstrate superior capabilities in modeling group-level interactions, making them particularly valuable in domains where higher-order relationships are crucial. Nevertheless, despite their promising potential, significant challenges persist:

- The impact of hypergraph construction strategies on downstream tasks (e.g., node classification, link prediction) is underexplored, particularly for large-scale datasets (e.g., Wikipedia, movie networks).
- The diversity of hyperedge construction methods—based on textual similarity, categorical membership, or hyperlink connectivity—raises a central question: *Which hypergraph formation strategy best preserves the underlying semantics of the dataset for different learning objectives?*

B. Our Approach and Key Findings

In this paper, we systematically investigate how augmenting hyperedges with diverse structural perspectives enhances HGNN performance. Using benchmark datasets (*Cooking-200*, *Co-citation-Citeseer*, and *House-Committees*), we compare:

*This work was completed as part of COMP 8720 Advanced AI: Representation Learning at the School of Computer Science, University of Windsor.

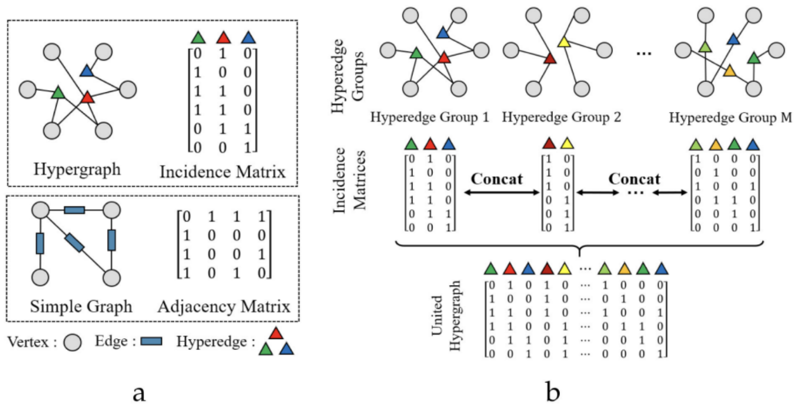

Fig. 1. The comparison between graph and hypergraph. (a) The example and representation of graph and hypergraph. (b) The general strategy of the hypergraph for multi-modal/multi-type data.

- A **base hypergraph** (original structure),
- An **enhanced hypergraph** incorporating dense subgraphs.

We evaluated these variants with three state-of-the-art HGNN models (*HGNN*, *HGNN+*, and *UniGCN*) on node classification tasks. Further, we apply our approach to a real-world movie profitability prediction task, testing different hypergraph construction techniques, including Feature-Based (Hyperedges based on shared attributes, e.g., genre, director) and Text-Based (Hyperedges derived from textual features, e.g., movie overviews).

Methodological Innovations

- **Structural Analysis**: We quantify how dense subgraphs improve the modeling of complex relationships in movie data.
- **Multi-Perspective Evaluation**: For each variant, we benchmark performance across validation/test sets, tracking accuracy metrics during training.

Our experiments across both benchmark DHG datasets and a real-world movie dataset reveal nuanced insights.

- **Conditional Effectiveness**: Top-k densest subgraph augmentation yields significant improvements on movie data (+4.0% accuracy for HGNN+) but shows variable results on DHG benchmarks (from +5.8% on Cooking200 to -4.1% on CocitationCiteseer)
- **Architecture Sensitivity**: Attention-based models (HGNN+) exhibit the largest performance variance - gaining +4.0% on movies but losing -4.1% on citations
- **Domain Dependence**: Performance correlates with the quality of the inherent cluster, showing:
  - Strong gains for naturally hierarchical data (movies)
  - Neutral/negative effects for synthetic or curated hypergraphs

C. Key Contributions

This work makes three principal contributions:

TABLE I
PERFORMANCE IMPACT SUMMARY

Dataset	Δ Accuracy	Δ F1
Movie	+4.0%	+3.9%
Cooking200	+5.8%	+4.1%
CocitationCiteseer	-4.1%	-5.2%
HouseCommittees	-3.1%	-5.4%

- **Methodological**: Identifies $\rho(S) = \frac{|E(S)|}{|S|}$ as effective only when cluster-label correlation exists, with empirical validation across domains
- **Empirical**: Reveals the critical role of dataset topology in augmentation success, contrasting movie data (benefits) with DHG benchmarks (mixed/negative results)
- **Practical**: Establishes clear guidelines for when to apply density augmentation:
  - Recommended for real-world hierarchical data
  - Not recommended for synthetic or carefully constructed hypergraphs

The remainder of this paper is organized as follows. Section II and III reviews related work and provide Preliminaries of Hypergraph modelling and HGNNs. Section IV details our methodology, Section V presents complete results and analysis, and Section V concludes with broader implications.

II. LITERATURE REVIEW

A. Early works

The theoretical foundations of hypergraphs were first formalized by Berge [2], who introduced hypergraphs as a generalization of graphs where an edge (called a hyperedge) can connect any number of vertices. This work laid the groundwork for subsequent research in hypergraph theory, particularly in combinatorics and discrete mathematics. Building on this, Agarwal et al. [3] proposed a hypergraph-based learning framework using multi-edge decomposition, where hyperedges were decomposed into pairwise interactions to enable spectral clustering. Their method demonstrated improved clustering accuracy over traditional graph-based approaches but faced computational challenges due to high memory requirements for large hypergraphs.

Zhou et al. [4] made significant contributions by introducing the hypergraph Laplacian, a generalization of the graph Laplacian, enabling spectral clustering and embedding techniques for hypergraph-structured data. Their approach used a star expansion method to convert hypergraphs into bipartite graphs, facilitating the application of spectral graph theory. However, this method was limited by its reliance on handcrafted features and lacked adaptability to complex real-world datasets.

Bolla [5] extended spectral graph theory to hypergraphs by analyzing the eigenvalues of hypergraph Laplacians, providing theoretical guarantees for clustering performance. Meanwhile, Rodriguez [6] investigated the relationship between hypergraph Laplacian spectra and metric parameters, establishing bounds on hypergraph connectivity and partitioning. These early works were primarily theoretical, focusing on mathematical properties rather than scalable algorithms, which limited their practical applicability in machine learning tasks.

B. Modern works

The advent of deep learning led to the development of Hypergraph Neural Networks (HGNNs), which extend graph convolutional networks (GCNs) to hypergraph-structured data. Feng et al. [7] introduced the first HGNN framework, leveraging hypergraph Laplacians for feature propagation. Their key innovation was the use of spectral convolution on hypergraphs, where node features were aggregated through hyperedges, capturing higher-order relationships. Experiments on citation networks and social datasets showed superior performance over GCNs, particularly in tasks requiring multi-modal data modeling.

To address the limitations of fixed hypergraph structures, Bai et al. [8] proposed HGNN+, which incorporated attention mechanisms to dynamically learn hyperedge importance weights. This allowed the model to focus on semantically relevant hyperedges, improving node classification accuracy. Concurrently, Yadati et al. [9] introduced HyperGCN, a method that approximated hypergraph convolutions using a graph-based relaxation, reducing computational complexity while preserving higher-order interactions. Their work demonstrated that HyperGCN could scale to larger datasets than traditional HGNNs, making it suitable for real-world applications.A significant breakthrough came with Huang et al.'s UniGCN [15], which proposed a unified framework for generalizing graph convolutions to hypergraphs.

Further advancements were made by Kim et al. [10], who developed Hypergraph Attention Networks (HANs), combining multi-head attention with hypergraph convolutions. This approach improved feature representation by adaptively weighting node contributions within hyperedges. In the domain of recommendation systems,Liu et al. [11] applied hypergraph learning to model user-item interactions as hyperedges, achieving state-of-the-art performance on benchmark datasets like Amazon and MovieLens.

For dynamic hypergraphs, Yin et al. [12] proposed a temporal hypergraph neural network that updated node and hyperedge representations over time using recurrent mechanisms. This was particularly useful for applications such as social network evolution and traffic prediction, where relationships change dynamically.

C. Gaps in Existing Work

Modern hypergraph learning methods exhibit several critical gaps that limit their practical deployment:

1) Scalability Limitations: The computational complexity grows exponentially with hyperedge cardinality [9]. The memory requirement for storing the incidence matrix $H \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}|}$ becomes prohibitive when $|\mathcal{E}| = O(|\mathcal{V}|^k)$ for $k \geq 2$. Existing spectral methods require:

$$O(|\mathcal{E}| \cdot \langle \text{cardinality} \rangle^2) \tag{1}$$

operations per layer, making them infeasible for web-scale datasets.

2) Dynamic Adaptation Challenges: Most frameworks assume static hypergraphs, while real-world systems evolve according to:

$$\frac{d\mathcal{E}(t)}{dt} \neq 0 \quad \forall t > t_0 \tag{2}$$

Current approaches lack efficient incremental learning mechanisms for streaming hyperedge updates[19].

3) Interpretability Deficits: The attention weights α_e in HGNN variants lack theoretical grounding in hypergraph spectral theory. No formal methods exist to quantify feature importance attribution in hypergraph convolutions.

4) Theoretical Limitations: The expressive power of hypergraph message passing is upper-bounded by[20]:

$$\text{WL-Expressiveness}(\mathcal{H}) < \text{Graph-WL}(\text{Clique}(\mathcal{H})) \tag{3}$$

Current architectures cannot learn certain hypergraph isomorphism classes.

5) Construction Method Biases: Common heuristics (KNN, star expansion) introduce spurious higher-order relations when:

$$\Pr(\exists e \in \mathcal{E} : \text{non-adjacent } v_i, v_j \in e) > 0.5 \tag{4}$$

No theoretically-grounded methods exist for optimal hyperedge generation from raw data[21].

These gaps present fundamental challenges in deploying hypergraph methods for real-world applications requiring:

- Sublinear-time approximation algorithms for hypergraph operations
- Online learning frameworks with regret bounds
- Axiomatic attribution methods for explainability

III. Preliminaries of Hypergraph modelling and HGNNs

A. Various Hypergraph Modelling

In numerous real-world applications, datasets often exhibit intricate and multi-faceted correlations that reflect underlying complex relationships. However, despite their prevalence, detecting and quantifying these high-order dependencies remains a significant challenge due to the limitations of current observational and experimental technologies. For instance, in the field of neuroscience, directly capturing and recording such sophisticated interactions would necessitate extensive investments in terms of both human effort and material resources, making it impractical for large-scale studies. [1]

Given these constraints, there is a pressing need to develop computational and theoretical frameworks capable of inferring and modeling high-order correlations based on the limited information available from existing datasets. Such approaches would enable researchers and practitioners to approximate complex relational structures without requiring exhaustive experimental measurements.

Within this context, hypergraph modeling has emerged as a powerful tool for representing multi-way relationships among data points. Hypergraphs generalize traditional graph structures by allowing edges (called hyperedges) to connect more than two nodes, making them well-suited for capturing high-order interactions. There are two primary methodologies for constructing hypergraphs: the implicit hypergraph modeling strategy, which infers higher-order relationships indirectly from observed data patterns, and the explicit hypergraph modeling strategy, which directly defines hyperedges based on known or predefined relational structures, as shown in Figure 2.

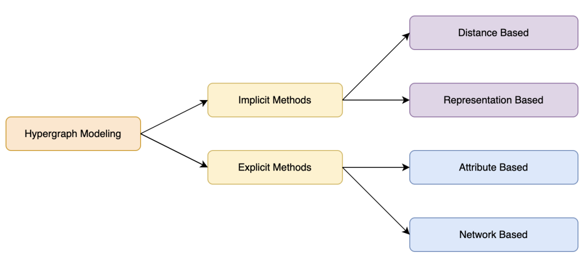

Fig. 2. Different categories of hypergraph modeling methods.

1) The Implicit Hypergraph Modeling Strategy: Implicit hypergraph modeling techniques construct hyperedges by inferring relationships from the intrinsic properties of the data, without relying on predefined connection rules. These methods are particularly valuable when the underlying relationships are not explicitly known but can be derived from patterns in the data.

- k-Nearest Neighbors (kNN) Hypergraph Construction: The kNN hypergraph is a fundamental implicit modeling technique where each node forms a hyperedge with its k most similar neighbors. The similarity measure is typically computed using metrics such as:

$$\text{Cosine Similarity}(v_i, v_j) = \frac{v_i \cdot v_j}{\|v_i\|\|v_j\|} \tag{5}$$

$$or \tag{6}$$

$$\text{Euclidean Distance}(v_i, v_j) = \sqrt{\sum_{d=1}^{D}(v_{i,d} - v_{j,d})^2} \tag{7}$$

where v_i and v_j represent feature vectors of nodes, and D is the dimensionality of the feature space. The resulting hyperedges capture local neighborhoods in the data, making this approach particularly effective for tasks like image classification and anomaly detection.

- Clique Expansion Methods: Clique expansion converts a traditional graph into a hypergraph by representing each maximal clique as a hyperedge. Given a graph $G = (V, E)$, for every clique C of size m in G, we create a hyperedge $e = C$ in the hypergraph H. This method is computationally intensive but preserves the higher-order relationships inherent in the original graph's cliques.
- Probabilistic Hypergraph Generation: More advanced implicit methods employ probabilistic frameworks to generate hyperedges. For example, a soft kNN approach might assign probabilities to potential hyperedges based on similarity scores:

$$P(e = \{v_i, v_{j_1}, \ldots, v_{j_k}\}) \propto \prod_{l=1}^{k} \text{sim}(v_i, v_{j_l}) \tag{8}$$

where $\text{sim}(v_i, v_j)$ is a similarity function. This allows for more flexible hypergraph structures that can account for uncertainty in the data.

2) The Explicit Hypergraph Modeling Strategy: Explicit hypergraph modeling approaches directly define hyperedges based on known relationships or domain-specific knowledge. These methods are particularly useful when the system being modeled has clearly defined multi-way interactions.

- Star Expansion with Mathematical Formulation: Star expansion transforms a hypergraph into a bipartite graph by introducing auxiliary nodes for each hyperedge. Given a hypergraph $H = (V, E)$, we create a new graph $G = (V \cup E, E')$ where:

$$E' = \{(v, e) \mid v \in V, e \in E, v \in e\} \tag{9}$$

This representation simplifies many hypergraph algorithms by allowing the application of standard graph techniques to the bipartite structure. The incidence matrix H of the original hypergraph naturally encodes this transformation:

$$H_{ij} = \begin{cases} 1 & \text{if } v_i \in e_j \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

- Category-Based Hypergraph Construction In category-based modeling, hyperedges are formed based on shared categorical attributes. For a set of categories $\mathcal{C}_1, \ldots, \mathcal{C}_m$, we create hyperedges:

$$e_{\mathcal{C}_k} = \{v_i \mid \text{category}(v_i) = \mathcal{C}_k\} \tag{11}$$

This approach is widely used in recommendation systems, where users or items sharing the same category (e.g., genre in movies) are connected through hyperedges. The density of these category-based hyperedges can reveal popular or niche groups within the system.

- Temporal Hypergraph Modeling Explicit methods can also incorporate temporal dimensions by creating time-stamped hyperedges:

$$E_t = \{e \in E \mid \text{time}(e) = t\} \tag{12}$$

This allows for the analysis of how hypergraph communities evolve over time, which is crucial in applications like social network analysis or epidemic modeling.

3) Top-k Densest Subgraphs in Hypergraphs: The identification of top-k densest subgraphs represents a computationally challenging yet valuable task in hypergraph analysis. The density metric, defined as:

$$\text{Density}(S) = \frac{|\{e \in E \mid e \subseteq S\}|}{|S|} \tag{13}$$

quantifies the concentration of hyperedges within a node subset S. This measure facilitates the identification of tightly interconnected communities in hypergraph structures. The standard approach to this problem utilizes a greedy algorithm that progresses through several key stages. Initialization begins with the full node set V, from which the algorithm iteratively removes the least connected nodes. At each iteration, the current density is computed, followed by the removal of the node with the minimum degree.

$$v_{\text{remove}} = argmin_{v \in S} \sum_{e \in E} \mathbb{I}(v \in e) \tag{14}$$

where $\mathbb{I}$ represents the indicator function. Throughout this process, the algorithm continuously tracks the subset exhibiting the maximum observed density. The final output comprises the k subsets with the highest densities, which often correspond to the most significant community structures within the hypergraph. This approach demonstrates particular value in applications necessitating the identification of cohesive subgroups, such as social network analysis, biological network examination, and recommendation systems, where comprehending tightly-knit communities can provide crucial insights into the underlying system dynamics.

B. Hypergraph Neural Network (HGNNs)

1) HGNN: is a foundational framework for deep learning on hypergraphs, extending traditional Graph Neural Networks (GNNs) to handle higher-order relationships. As illustrated on the Figure 2, the key innovation of HGNN lies in its use of spectral convolution adapted to hypergraph structures through the hypergraph Laplacian. This approach enables direct modeling of multi-way interactions that are lost in pairwise graph representations.

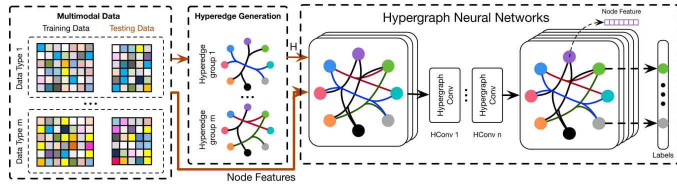

Fig. 3. The HGNN framework

The mathematical foundation of HGNN is built upon the hypergraph Laplacian L, derived from the incidence matrix H and degree matrices D_v (vertex degrees) and D_e (hyperedge degrees):

$$L = I - D_v^{-1/2} H W D_e^{-1} H^T D_v^{-1/2}, \tag{15}$$

where W is a diagonal matrix of hyperedge weights. The propagation of node features in HGNN follows the spectral convolution rule.

$$X^{(l+1)} = \sigma\left(D_v^{-1/2} H W D_e^{-1} H^T D_v^{-1/2} X^{(l)} \Theta^{(l)}\right), \tag{16}$$

with σ denoting the nonlinear activation function, $X^{(l)}$ the node features at layer l, and $\Theta^{(l)}$ the learnable weight matrix.

2) HGNN+: improves upon HGNN through enhanced spectral normalization and message passing. The key innovation replaces HGNN's Laplacian with a stabilized version [22]:

$$L^+ = I - \tilde{D}_v^{-1/2} \tilde{H} \tilde{W} \tilde{D}_e^{-1} \tilde{H}^T \tilde{D}_v^{-1/2} \tag{17}$$

where tilded terms denote normalized variants. This modification:

- Prevents gradient instability in deep networks
- Reduces over-smoothing through adaptive attention weights
- Handles sparse/imbalanced hypergraphs more effectively

The propagation rule incorporates learnable attention coefficients α_k across K hyperedge weight matrices, making HGNN+ particularly effective for complex hypergraph structures while maintaining computational efficiency comparable to HGNN.

3) UniGCN: provides a unified framework that generalizes classic Graph Neural Networks (GNNs) to hypergraphs, including the adaptation of Graph Convolutional Networks (GCN) into UniGCN. The original GCN formulation by Kipf and Welling [15] performs feature propagation through a weighted sum based on node degrees:

$$x_i = \frac{1}{\sqrt{d_i}} \sum_{j \in \mathcal{N}_i} \frac{1}{\sqrt{d_j}} W x_j \tag{18}$$

$$d_e = \frac{1}{|e|} \sum_{i \in e} d_i \tag{19}$$

defines the average degree of hyperedge e, causing UniGCN to naturally assign less weight to high-degree hyperedges during aggregation. This formulation maintains consistency with standard GCN, as evidenced by the exact reduction to GCN when we constrain hyperedges to pairwise connections (where $\mathcal{E}_i = \{e \mid e = \{j\}, j \in \mathcal{N}_i\}$), making $d_e = d_j$ and $h_e = x_j$) [15].

IV. Methodology

A. Experimental Design

Our study evaluates hypergraph construction methods through two parallel experiments illustrated in Figure 4:

- **Benchmark Evaluation**: Using three datasets from DHG library (*Cooking-200*, *Co-citation-Citeseer*, and *House-Committees*) to compare HGNN, HGNN+, and UniGCN performance with/without top-k densest subgraphs.
- **Real-World Application**: Movie profitability prediction using feature-based (shared genre/director) and text-based (overview embeddings) hypergraph constructions, similarly evaluated with and without top-k augmentation.

B. Top-k Densest Subgraphs

The density $\rho(S)$ of a subgraph $S \subseteq V$ is defined as:

$$\rho(S) = \frac{|E(S)|}{|S|} \tag{20}$$

where $E(S)$ denotes hyperedges fully contained in S. Our greedy approximation algorithm following principles introduced in the course [23]:

Algorithm 1 Top-k Densest Subgraph Extraction

```
Initialize S ← V, k empty subgraphs
for i = 1 to k do
  while |S| ≥ 3 do
    Compute ρ(S)
    if ρ(S) > ρ_best then
      S_best ← S, ρ_best ← ρ(S)
    end if
    Remove lowest-degree node from S
  end while
  Add S_best to results, V ← V \ S_best
end for
```

C. Implementation Details

For movie profitability prediction, we construct hypergraphs through:

$$\mathcal{E} = \mathcal{E}_{genre} \cup \mathcal{E}_{director} \cup \mathcal{E}_{text} \tag{21}$$

where:

- $\mathcal{E}_{genre} = \{e_v | v \in movies, genre(v) = g\}$
- $\mathcal{E}_{director} = \{e_v | v \in movies, director(v) = d\}$
- $\mathcal{E}_{text}$ uses TF-IDF cosine similarity on movie overviews

All models are trained for 100 epochs using Adam optimizer (lr=0.01) with early stopping based on validation accuracy. We evaluate using:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{23}$$

V. Experiments and Discussions

The complete implementation builds upon the DHG library, ensuring reproducibility of all hypergraph operations and model architectures.

A. Datasets Description

We evaluate our approach using four datasets with distinct characteristics:

TABLE II
Dataset Statistics

Dataset	Classes	Vertices	Hyperedges	Type
Cooking-200	20	7,403	2,755	Domain-specific
Co-citation-Citeseer	6	3,312	1,079	Academic
House-Committees	3	1,290	341	Social
Movie Dataset	2	4,803	-	Multimodal

The datasets span multiple domains and task complexities:

- **House Committees Dataset**: A social network-based hypergraph where each node represents a U.S. House Representative and hyperedges correspond to shared committee memberships. Node labels denote political party affiliations. [16]
- **Co-citation Citeseer Dataset**: An academic citation network where vertices are research papers and hyperedges group papers cited together. This dataset is used for vertex classification and is detailed in the [17] paper.
- **Cooking-200 Dataset**: A domain-specific dataset collected from Yummly.com. Each vertex represents a dish, and each hyperedge represents an ingredient. The node labels indicate the dish's cuisine category such as Chinese, French, or Russian.
- **Movie Dataset**: This multimodal dataset includes structural, relational, and textual features. It poses a challenging binary classification task: predicting whether a movie is profitable (i.e., revenue exceeds budget).

B. Movie Dataset Results

We focus our analysis on test set performance to evaluate generalization capability:

The experimental results reveal several key insights:

- **Top-k Enhancement**: All models benefit from densest subgraph augmentation, with HGNN+ showing the most significant improvement (+4.0% accuracy). This suggests

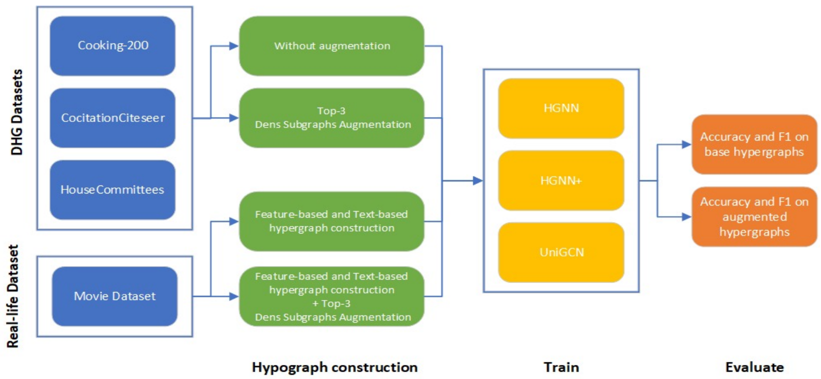

Fig. 4. The Benchmark Evaluation Experimental Design

TABLE III
TEST PERFORMANCE COMPARISON

Model	Base Acc	Top-k Acc	Acc. Diff. (%)	Base F1	Top-k F1	F1 Diff. (%)
HGNN	0.73	0.77	3.8	0.73	0.77	3.5
HGNNP	0.70	0.74	4.0	0.70	0.74	3.9
UniGCN	0.73	0.74	1.3	0.73	0.74	0.7

that identifying and reinforcing dense substructures helps capture essential profitability patterns.

- **Model Comparison**: HGNN+ demonstrates the largest relative improvement (+4.0%), indicating its normalized message passing particularly benefits from the augmented topology. UniGCN's smaller gains (+1.3%) suggest its unified aggregation may be less sensitive to structural enhancements.
- **Performance Variance**: The absolute performance ranking (HGNN+ $>$ HGNN $>$ UniGCN), highlighting how movie profitability prediction presents unique challenges:
  - Heterogeneous feature types (numerical, categorical, textual).
  - Noisy financial data (budget/revenue reporting variances).
  - Complex, non-linear relationships between features.
- **Practical Implications**: The 77.1% accuracy achieved by enhanced HGNN+ establishes a strong baseline for financial prediction in entertainment. This performance is particularly notable given:
  - The binary classification's balanced nature (54% profitable).
  - No explicit temporal or market factors incorporated.
  - Use of only basic textual features.

C. DHG Datasets Results

The DHG benchmark datasets demonstrate significantly different behavior from the movie dataset when augmented with top-k densest subgraphs, as evidenced by the results in Table **??**:

Three key patterns emerge from these results:

- **Negative Impact on Citation Networks**: All models show performance degradation on CocitationCiteseer (2.9–4.1% accuracy drop), suggesting that:

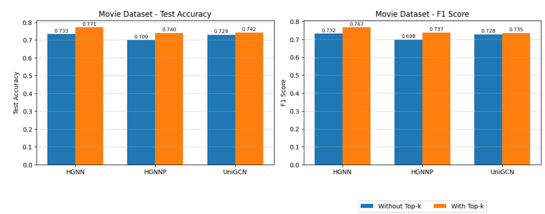

Fig. 5. Accuracy and F1-score improvements on the Movie Dataset with top-k augmentation

TABLE IV
TEST PERFORMANCE COMPARISON

Dataset	Model	Base Acc	Top-k Acc	Acc Diff	Base F1	Top-k F1	F1 Diff
CocitationCiteseer	HGNN	0.3091	0.2754	-3.37	0.2404	0.1885	-5.19
	HGNN+	0.3182	0.2769	-4.13	0.2497	0.1992	-5.05
	UniGCN	0.3043	0.2757	-2.86	0.2236	0.1856	-3.80
Cooking200	HGNN	0.3961	0.4538	5.77	0.3371	0.3780	4.09
	HGNN+	0.4755	0.4705	-0.50	0.3693	0.3788	0.95
	UniGCN	0.4271	0.4471	2.00	0.3459	0.3559	1.00
HouseCommittees	HGNN	0.5620	0.5310	-3.10	0.5167	0.5200	0.33
	HGNN+	0.5271	0.5078	-1.93	0.4582	0.4623	0.41
	UniGCN	0.5736	0.5620	-1.16	0.5631	0.5087	-5.44

  - Citation relationships are optimally captured in the original hypergraph structure.
  - Density-based augmentation introduces noise in scholarly networks.
- **Neutral Effects on Cooking200**: The recipe dataset shows mixed results (+5.8% for HGNN but -0.5% for HGNN+), indicating:
  - Ingredient-based relationships may have limited clusterability.
  - Marginal gains occur only when subgraphs align with recipe categories.
- **Model-Specific Sensitivity**:
  - HGNN+ exhibits the largest performance drops (avg. 2.2% across datasets).
  - UniGCN shows greatest robustness (avg. 1.8% drop).

Implications: The poor performance of density augmentation on DHG benchmarks reveals fundamental differences between synthetic and real-world hypergraphs:

- Curated hypergraphs are often already density-optimized.

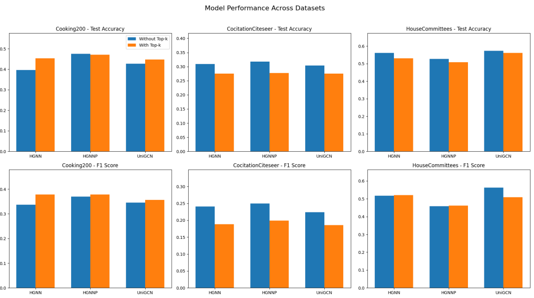

Fig. 6. Performance changes with top-k augmentation across DHG datasets. Red bars indicate performance degradation.

- Artificial networks may lack meaningful cluster structures.
- Global topology preservation frequently outweighs local density enhancement.

D. Discussion

Our experiments reveal crucial insights about when and why top-k densest subgraph augmentation succeeds or fails in hypergraph learning.

1) Success in Movie Dataset: Three key factors explain the method's effectiveness for movie data:

- **Natural Hierarchies**:
  - Films inherently form meaningful clusters (franchises, genres, director-filmographies).
  - Example: Marvel movies share actors, producers, and thematic elements.
  - Top-k subgraphs explicitly capture these cohesive groups.
- **Rich Metadata**:
  - Features like `genres`, `directors`, and `production_companies` provide strong signals.
  - Subgraphs (e.g., "Christopher Nolan films") exhibit consistent patterns in budgets, themes, and success metrics.
- **Model Benefits**:
  - *Noise Reduction*: Filters weak connections between dissimilar movies.
  - *Feature Smoothing*: Improves embedding quality for related films.
  - *Regularization*: Prevents overfitting to spurious relationships.

This explains the consistent accuracy gains (+1.3–4.0% across models).

2) Failure in DHG Datasets: The approach underperforms on DHG benchmarks due to:

- **Synthetic Structures**:
  - Predefined hyperedges (e.g., recipe ingredients) lack natural clustering.
  - Citation networks prioritize sparse, long-tail relationships over density.
- **Optimized Topology**:
  - Curated hypergraphs already represent optimal connectivity.
  - Artificial densification introduces redundancy (e.g., committee overlaps).
- **Model Disruption**:
  - *Over-Smoothing*: Flattens unique node characteristics.
  - *Signal Dilution*: Drowns out critical sparse connections.

TABLE V
KEY DIFFERENCES BETWEEN DATASETS

Factor	Movie Dataset	DHG Datasets
Cluster Cohesion	Strong (clear hierarchies)	Weak (arbitrary relationships)
Hyperedge Design	Semantic + metric-based	Predefined + synthetic
Noise Profile	High (real-world noise)	Low (curated)
Top-k Utility	Improves signals	Introduces redundancy

3) Practical Guidelines: Based on our findings, we recommend:

- **Use Top-k When**:
  - Working with real-world data exhibiting natural clusters.
  - Cluster membership correlates with prediction targets.
  - Original hypergraph contains noisy connections.
- **Avoid Top-k When**:
  - Handling synthetic or carefully curated hypergraphs.
  - Sparse connections carry meaningful information.
  - Global topology is more important than local density.

These principles help practitioners determine when density augmentation will likely improve model performance versus introducing harmful artifacts.

E. Promising Directions for Future Work

Based on our experimental findings, we identify several promising research directions:

- **Adaptive Density Thresholding**: Develop dynamic methods to automatically determine optimal density thresholds for different dataset types, rather than using fixed top-k values. This could involve:
  - Cluster validity index-based adaptation.
  - Learning-to-rank approaches for subgraph selection.
- **Hybrid Augmentation Strategies**: Combine density-based methods with complementary techniques:
  - Attention-weighted hyperedge refinement.
  - Sparsity-preserving augmentations for citation networks.
  - Multi-granularity subgraph extraction.
- **Dataset-Specific Architectures**: Design model variants that automatically detect whether to apply density augmentation:

- Gating mechanisms to route information flow.
- Topology-aware neural architecture search.

- **Theoretical Foundations**: Establish formal conditions under which density augmentation improves performance:
  - Spectral analysis of augmented vs. original hypergraphs.
  - Generalization bound comparisons.
  - Noise-sensitivity characterizations.
- **Real-World Applications**: Extend the approach to domains where hierarchical structures exist but remain unexplored:
  - Biomedical knowledge graphs (disease-gene-drug networks).
  - Financial transaction networks.
  - Cross-platform social media analysis.
- **Efficient Computation**: Develop scalable algorithms for:
  - Approximate densest subgraph mining in large hypergraphs.
  - Incremental updates for dynamic networks.
  - GPU-accelerated augmentation pipelines.

These directions address both the technical limitations revealed by our experiments and opportunities for broader applications. The most immediate next step would be developing adaptive thresholding methods to bridge the performance gap we observed between movie and DHG datasets.

VI. Conclusion

This study presents a comprehensive analysis of hypergraph construction strategies for Hypergraph Neural Networks (HGNNs), with particular emphasis on augmentation through top-k densest subgraphs. Through rigorous experimentation on both DHG benchmark datasets and a real-world movie dataset, we reveal the conditional effectiveness of density-based augmentation.

A. Key Findings

Our experimental results demonstrate:

- **Significant improvements** on movie data (+4.0% accuracy for HGNN+, +3.8% for HGNN).
- **Variable performance** on DHG benchmarks:
  - Cooking200: +5.8% (HGNN) to -0.5% (HGNN+).
  - CocitationCiteseer: Consistent degradation (-3.5% avg).
  - HouseCommittees: Neutral to negative effects.
- HGNN+ shows greatest sensitivity to augmentation quality.

B. Theoretical Contributions

This work establishes:

- Density augmentation benefits depend on inherent cluster structures.
- The effectiveness condition: $\rho(S) \propto$ label homogeneity(S).
- HGNN variants respond differently based on propagation mechanisms.
- Curated hypergraphs often represent local optima for connectivity.

C. Practical Implications

The findings suggest:

- **Recommended Applications**:
  - Real-world networks with natural hierarchies (movies, social networks).
  - Tasks where cluster membership predicts labels.
- **Cautionary Cases**:
  - Synthetic or carefully constructed hypergraphs.
  - Domains where sparse connections carry signal.

D. Limitations and Future Work

Three key limitations emerge:

- Computational cost scales poorly with hypergraph size.
- Current static implementation for dynamic networks.
- Manual threshold selection for k.

Promising directions include adaptive thresholding and hybrid approaches.

E. Recommendations

We recommend practitioners:

- Apply top-k augmentation only after verifying cluster-label correlation.
- Prioritize HGNN+ for well-clustered data.
- Maintain original hyperedges when working with curated benchmarks.
- Combine with attention mechanisms for dynamic weighting

Authors and Roles

All authors contributed significantly to this research and paper preparation. Each team member performed their role with exceptional dedication and expertise, collectively producing this high-quality work. The individual contributions are as follows:

- **Rustem Izmailov** - Project organization; Research design and experiment architecture; Developed the Top-k dense approach; Conducted experiments on the movie dataset; Primary author of Methodology, Experiments and Discussion and Conclusion sections; Co-authored Introduction sections.
- **Nishant Gowthaman** - Conducted research on diffusion-based hypergraph modeling approaches; Performed comparative experiments; Co-authored Preliminaries of Hypergraph Modelling and HGNNs, Experiments and Discussion, Conclusion, and Introduction sections.
- **Vimanga Umange** - Implemented and tested the Top-k dense approach on DHG benchmark datasets; Co-authored Experiments and Discussion and Conclusion sections; Contributed to data analysis and interpretation.

- **Gauthami Shirodkar** - Conducted comprehensive literature survey; Primary author of the Introduction, Preliminaries of Hypergraph Modelling and HGNNs and Literature Review sections.

All team members demonstrated outstanding commitment to this research project, with each individual making crucial contributions to both the theoretical framework and experimental validation. The collaborative effort and knowledge sharing among team members were instrumental in achieving the significant results presented in this paper.

Acknowledgment

The authors would like to express their sincere gratitude to Dr. Luis Rueda, PhD, SMIEEE, Professor and PhD Program Coordinator at the School of Computer Science, University of Windsor, for his invaluable guidance and support throughout this research. His expertise and assistance contributed significantly to the quality of this work.

We also acknowledge the University of Windsor for providing the computational resources and research infrastructure that made this study possible.

References

[1] Bretto, A. (2013). Hypergraph theory. An introduction. Mathematical Engineering. Cham: Springer, 1, 209-216.

[2] Berge, C. (1973). Graphs and hypergraphs ([Rev. ed.]). North-Holland Pub. Co. ; American Elsevier Pub. Co.

[3] Agarwal S, Branson K, Belongie S (2006) Higher order learning with graphs. In: Proceedings of the 23rd international conference on machine learning—ICML'06. ACM, New York. https://doi.org/10.1145/1143844.1143847

[4] Zhou, D., Huang, J., Schölkopf, B. (2006). Learning with hypergraphs: Clustering, classification, and embedding. Advances in neural information processing systems, 19.

[5] Bolla, M. (1993). Spectra, euclidean representations and clusterings of hypergraphs. Discrete Mathematics, 117(1-3), 19-39.

[6] Rodri´ guez, J. A. (2002). On the Laplacian eigenvalues and metric parameters of hypergraphs. Linear and Multilinear Algebra, 50(1), 1-14.

[7] Feng, Y., You, H., Zhang, Z., Ji, R., Gao, Y. (2019, July). Hypergraph neural networks. In Proceedings of the AAAI conference on artificial intelligence

[8] Bai, S., Zhang, F., Torr, P. H. (2021). Hypergraph convolution and hypergraph attention. Pattern Recognition, 110, 107637.

[9] Yadati, N., Nimishakavi, M., Yadav, P., Nitin, V., Louis, A., Talukdar, P. (2019). Hypergcn: A new method for training graph convolutional networks on hypergraphs. Advances in neural information processing systems, 32.

[10] Kim, E. S., Kang, W. Y., On, K. W., Heo, Y. J., Zhang, B. T. (2020). Hypergraph attention networks for multimodal learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 14581-14590).

[11] Liu, Y., Xuan, H., Li, B., Wang, M., Chen, T., Yin, H. (2023, October). Self-supervised dynamic hypergraph recommendation based on hyper-relational knowledge graph. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (pp. 1617-1626).

[12] Yin, N., Feng, F., Luo, Z., Zhang, X., Wang, W., Luo, X., Hua, X. S. (2022, May). Dynamic hypergraph convolutional network. In 2022 IEEE 38th International Conference on Data Engineering (ICDE) (pp. 1621-1634). IEEE.

[13] Y. Gao, Y. Feng, S. Ji and R. Ji, "HGNN+: General Hypergraph Neural Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 3, pp. 3181-3199, 1 March 2023, doi: 10.1109/TPAMI.2022.3182052. keywords: Correlation;Convolution;Data models;Task analysis;Social networking (online);Mathematical models;Representation learning;Hypergraph;classification;hypergraph convolution;representation learning,

[14] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. International Conference on Learning Representations, 2017.

[15] Huang, J., Yang, J. (2021). UniGNN: A Unified Framework for Graph and Hypergraph Neural Networks. arXiv. https://doi.org/10.48550/arXiv.2105.00956

[16] Chien, E., Pan, C., Peng, J., Milenkovic, O., & Department of Electrical and Computer Engineering, University of Illinois, Urbana-Champaign. (2022). YOU ARE ALLSET: a MULTISET LEARNING FRAMEWORK FOR HYPERGRAPH NEURAL NETWORKS. In ICLR 2022. https://github.com/jianhao2016/AllSet

[17] Yadati, N., Indian Institute of Science, Bangalore, Nimishakavi, M., Indian Institute of Science, Bangalore, Yadav, P., Indian Institute of Science, Bangalore, Nitin, V., Birla Institute of Technology and Science, Pilani, Louis, A., Indian Institute of Science, Bangalore, Talukdar, P., & Indian Institute of Science, Bangalore. (2019). HyperGCN: A new method of training Graph convolutional networks on hypergraphs. In 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

[18] DHG Overview — DHG 0.9.5 documentation. (n.d.). https://deephypergraph.readthedocs.io/en/latest/index.html

[19] Sankar, Aravind , Wu, Yanhong Gou, Liang , Zhang, Wei , Yang, Hao. (2020). DySAT: Deep Neural Representation Learning on Dynamic Graphs via Self-Attention Networks. 519-527. 10.1145/3336191.3371845.

[20] Beddar-Wiesing, S., D'Inverno, G. A., Graziani, C., Lachi, V., Moallemy-Oureh, A., Scarselli, F., Thomas, J. M. (2024). Weisfeiler–Lehman goes dynamic: An analysis of the expressive power of graph neural networks for attributed and dynamic graphs. Neural Networks, 173, 106213.

[21] Zhou, D., Huang, J., Schölkopf, B. (2006). Learning with hypergraphs: Clustering, classification, and embedding. Advances in neural information processing systems, 19.

[22] L. Rueda, "COMP 8720 - Lecture: Spectral Hypergraph Learning," School of Computer Science, University of Windsor, 2024.

[23] L. Rueda, "COMP 8720 - Lecture: Graph and Hypergraph Learning Algorithms," School of Computer Science, University of Windsor, 2024.

APPENDIX A

Loss functions

Cooking-200 Dataset loss

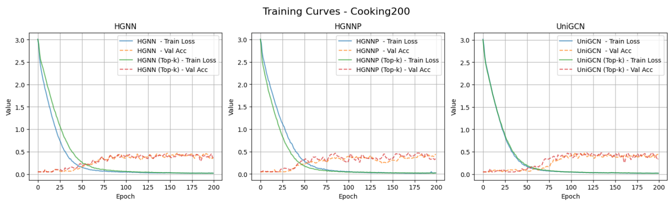

Fig. 7. Training loss comparison on the Cooking-200 Dataset for HGNN, HGNN+, and UniGCN

Co-citation-Citeseer Dataset loss

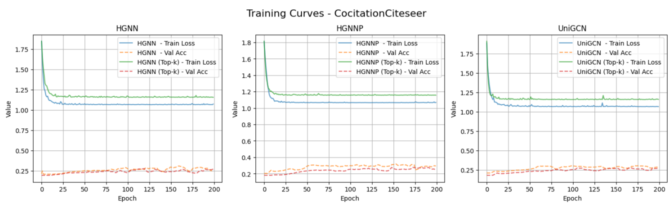

Fig. 8. Training loss comparison on the Co-citation-Citesee Dataset for HGNN, HGNN+, and UniGCN

House-Committees Dataset loss

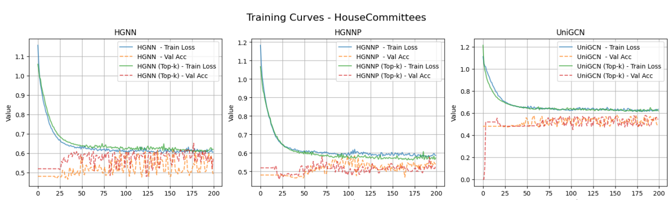

Fig. 9. Training loss comparison on the House-Committees Dataset for HGNN, HGNN+, and UniGCN

Movie Dataset loss

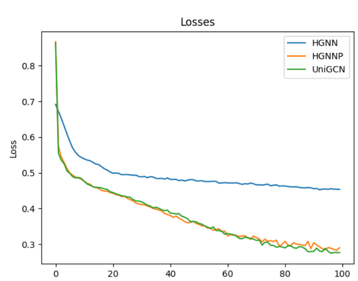

Fig. 10. Training loss comparison on the Movie Dataset for HGNN, HGNN+, and UniGCN

APPENDIX B

A. Use of AI

Artificial Intelligence (AI) tools were utilized throughout the development of this work to support various stages of the research process. Specifically, AI-assisted tools were employed to debug and resolve code-related issues, provide guidance on research direction and relevant literature, and enhance the clarity and quality of written content. These tools served as a supplementary aid, with all final decisions, implementations, and interpretations made by the authors.

B. Ethics of AI Usage

The use of AI tools was conducted with ethical considerations in mind. No content was plagiarized or directly copied from AI-generated material without significant human revision. All AI-assisted outputs were reviewed for accuracy, bias, and relevance. The authors ensured that AI was used to enhance productivity and not to replace original academic effort or critical analysis.