Heterophily-aware Contrastive Learning for Heterophilic Hypergraphs

Ming Li[1], Yongqi Li[2], Yuting Chen[3*], Feilong Cao[4], Ke Lv[5,6]

[1]Zhejiang Key Laboratory of Intelligent Education Technology and Application, Zhejiang Normal University, Jinhua, China
[2]School of Computer Science and Technology, Zhejiang Normal University, Jinhua, China
[3]Centre for Learning Sciences and Technologies, The Chinese University of Hong Kong, Hong Kong, China
[4]School of Mathematical Sciences, Zhejiang Normal University, Jinhua, China
[5]School of Engineering Science, University of Chinese Academy of Sciences, Beijing, China
[6]Peng Cheng Laboratory, Shenzhen, China
mingli@zjnu.edu.cn, liyongqi@zjnu.edu.cn, yuting.chen@cuhk.edu.hk, caofeilong88@zjnu.edu.cn, luk@ucas.ac.cn

Abstract

Hypergraph neural networks (HNNs) have emerged as powerful tools for modeling high-order relationships in complex systems. However, most existing HNNs are designed under the assumption of homophily, which does not hold in many real-world scenarios where connected nodes often exhibit diverse semantics, i.e., heterophily. This inconsistency leads to suboptimal aggregation and degraded performance, especially in low-label regimes. While a few recent methods have attempted to enhance heterophilic hypergraph learning, they often rely heavily on label supervision and overlook the potential of self-supervised techniques. In this paper, we propose HeroCL, a heterophily-aware contrastive learning framework that improves hypergraph representation under both structural heterogeneity and label scarcity. Specifically, HeroCL integrates a multi-hop neighbor encoding module to capture informative higher-order context and incorporates two complementary contrastive objectives, label-aware and structure-aware, to guide representation learning from both semantic and relational perspectives. A multi-granularity contrastive strategy is introduced to exploit latent signals across multiple neighborhood levels. Extensive experiments on several benchmark datasets against 10 existing baselines demonstrate that HeroCL achieves consistent and significant performance gains, particularly under strong heterophily and limited supervision, validating its robustness and effectiveness.

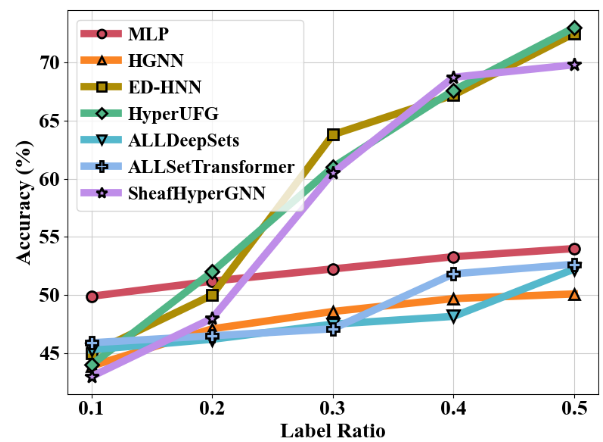

Figure 1: Classification accuracy on the heterophilic hypergraph dataset (Senate) under varying label ratios. The performance drop under low supervision **highlights the need for heterophily-aware contrastive learning**.

1 Introduction

Hypergraphs provide a powerful and expressive modeling paradigm for real-world systems involving high-order interactions, where a single hyperedge can connect an arbitrary number of nodes (Wang and Kleinberg 2024; Millán et al. 2025). This generalization from pairwise graphs enables hypergraphs to represent group-wise relationships across domains such as social networks, recommender systems, and intelligent education (Li et al. 2025a, 2024). With the increasing availability of hypergraph-structured data, hypergraph neural networks (HNNs) (Kim et al. 2024) have emerged as a promising class of models that learn effective node or hyperedge representations by aggregating information over hypergraph topology. Recent advances have demonstrated the success of HNNs in various applications, underscoring the importance of developing principled and effective learning frameworks for hypergraph representation learning (Antelmi et al. 2023; Li et al. 2025b).

While recent years have witnessed growing interest in HNNs, the majority of existing methods still assume a homophilic setting, where connected nodes tend to share similar semantics or labels. However, this assumption is often violated in real-world scenarios, where heterophilic hypergraphs are prevalent and characterized by diverse semantics within hyperedges. Such inconsistency can severely degrade performance, as traditional message passing tends to aggregate conflicting information. Several works, such as ED-HNN (Wang et al. 2023) and SheafHyperGNNs (Duta et al. 2023), have briefly acknowledged this issue in experimental discussions, but without systematically addressing it. To the best of our knowledge, HyperUFG (Li et al. 2025c) is the first method to explicitly focus on heterophilic hypergraph learning, incorporating tools such as equivariant diffusion and multi-frequency transforms to improve representation capability. Nevertheless, these methods still as-

*Corresponding author

sume access to sufficient label supervision and largely overlook their model robustness in low-label regimes. As demonstrated in Figure 1, when the proportion of labeled nodes is small, heterophily-aware models like ED-HNN and HyperUFG suffer significant performance drops, often falling behind even shallow methods such as MLP, highlighting their vulnerability to label scarcity. This observation suggests that addressing heterophily alone is insufficient without accounting for the common scenario of limited annotations.

To overcome these limitations, we explore the integration of self-supervised contrastive learning into the context of heterophilic hypergraph learning. Contrastive learning enables models to learn meaningful representations by pulling semantically similar samples closer and pushing dissimilar ones apart, even without access to explicit labels. This self-supervised paradigm is particularly appealing in heterophilic and weakly supervised environments, as it encourages nodes to capture their own semantic identity and reduce reliance on potentially misleading neighbors. In this work, we propose HeroCL, a novel Heterophily-aware contrastive learning framework for hypergraph representation learning. Specifically, we first introduce a **Multi-Hop Neighbor Encoding** module that models high-order structural information, which is essential in capturing indirect but informative relations in heterophilic settings. On top of this, we design two complementary contrastive modules: **Label-Aware Contrastive Learning**, which enhances class-wise separability based on limited labels, and **Structure-Aware Contrastive Learning**, which promotes locality-aware consistency by contrasting multi-hop neighbors and perturbations. Furthermore, we incorporate **Granularity-Aware Contrastive** objective to extract latent semantics from different relational levels. Extensive experiments conducted on 11 public benchmark datasets against 10 representative baselines demonstrate that HeroCL consistently achieves superior performance, particularly under strong heterophily and limited label supervision, highlighting its robustness, generalization ability, and effectiveness in label-scarce and structurally challenging hypergraph scenarios.

In summary, our key contributions are as follows:

- We are the first to systematically explore and integrate contrastive learning into heterophilic hypergraph representation learning, addressing the challenge of label scarcity and neighbor inconsistency.
- We propose a unified framework, HeroCL, which combines multi-hop neighbor encoding with a dual-view contrastive learning design, leveraging both label-aware and structure-aware objectives, to enhance representational quality under heterophily.
- We conduct comprehensive experiments across 11 public datasets and 10 baseline methods, showing that HeroCL consistently outperforms existing approaches and offering empirical insights into the interplay between contrastive learning and heterophilic hypergraph modeling.

2 Preliminaries and Notation

Basics on Hypergraphs. A hypergraph is represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, comprising a vertex set $\mathcal{V}$ of size $N = |\mathcal{V}|$, a hyperedge set $\mathcal{E}$ of size $M = |\mathcal{E}|$. Suppose that vertices have feature dimensions d, we have the representation of vertex data as $\mathbf{X} \in \mathbb{R}^{N\times d}$. The hypergraph structure, from a vertex perspective, is defined by an incidence matrix $\mathbf{H} \in \{0,1\}^{N\times M}$ where $\mathbf{H}(v,e) = 1$ if vertex v is contained in hyperedge e, and 0 otherwise, as represented by:

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

The degrees of vertex v and hyperedge e are denoted by diagonal matrices $\mathbf{D}_v \in^{N\times N}$ and $\mathbf{D}_e \in^{M\times M}$, calculated as $\sum_{e\in\mathcal{E}} \mathbf{H}(v,e)$ and $\sum_{v\in\mathcal{V}} \mathbf{H}(v,e)$, respectively. A hypergraph is a generalization of a standard graph where each hyperedge can connect an arbitrary number of nodes. Formally, a hypergraph is defined as $H = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_{|\mathcal{V}|}\}$ denotes the set of nodes and $\mathcal{E} = \{e_1, e_2, \ldots, e_{|\mathcal{E}|}\}$ denotes the set of hyperedges, with each hyperedge $e_j \subseteq \mathcal{V}$ being a non-empty subset of nodes.

Metrics for Hypergraph Homophily/Heterophily. To quantify the degree of heterophily in hypergraphs, we adopt the evaluation metrics introduced by (Li et al. 2025c), which to the best of our knowledge provide the first formal definitions for measuring homophily and heterophily in hypergraph learning. These metrics enable principled analysis from both node-centric and hyperedge-level perspectives.

Definition 1 (Hyperedge Homophily). The hyperedge homophily score H_{edge} measures the average proportion of same-label node pairs within each hyperedge:

$$H_{\text{edge}} = \frac{1}{|\mathcal{E}|} \sum_{e_j\in\mathcal{E}} \frac{|\{(u,v) \in e_j \mid 1(y_u = y_v)\}|}{\binom{n_j}{2}},$$

where $1(\cdot)$ is the indicator function and $n_j = |e_j|$.

Definition 2 (Node Homophily). The node homophily score H_{node} evaluates the average proportion of same-label neighbors for each node across its incident hyperedges:

$$H_{\text{node}} = \frac{1}{|\mathcal{V}|} \sum_{v\in\mathcal{V}} \frac{1}{|R_v|} \sum_{e_j\in R_v} \frac{|\{(u,v) \in e_j \mid 1(y_u = y_v)\}|}{n_j},$$

where $R_v = \{e_j \in \mathcal{E} \mid v \in e_j\}$ is the set of hyperedges incident to node v, and $n_j = |e_j|$.

3 Proposed Method: HeroCL

To address the challenge of learning on heterophilic hypergraphs under limited supervision, we propose HeroCL, a heterophily-aware contrastive learning framework that enhances node representations through view-aligned supervision. HeroCL follows a dual-branch architecture and integrates multi-hop neighborhood encoding with two complementary contrastive objectives: structure-aware and label-aware contrast. As illustrated in Figure 2, the overall architecture comprises three core components:

- **Hypergraph Augmentation.** Two structurally distinct views are constructed via perturbations applied to both the hypergraph topology and node features. These augmentations preserve essential semantic content while promoting representation diversity across views, enabling the contrastive objective to effectively learn invariant representations.

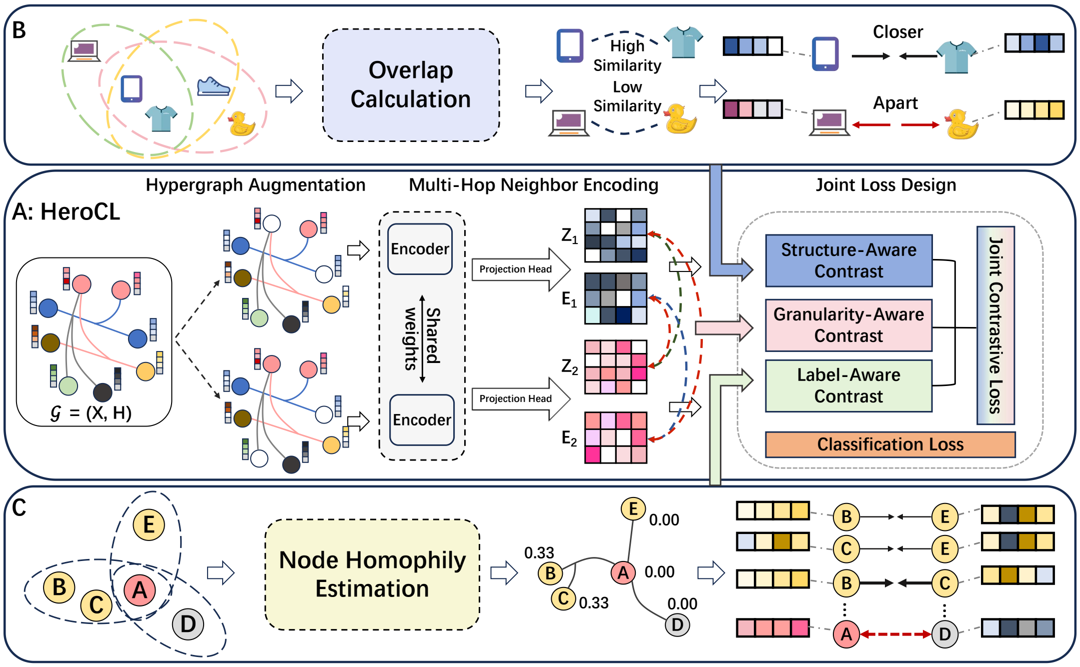

Figure 2: Schematic illustration of the proposed HeroCL framework: (A) Overall model architecture; (B) Structure-aware contrastive learning branch; (C) Label-aware contrastive learning branch.

- **Multi-Hop Neighbor Encoding.** To account for long-range dependencies and alleviate noise from immediate dissimilar neighbors, we employ second-order neighborhood aggregation. A shared encoder processes both augmented views, producing node embeddings Z_1, Z_2 and hyperedge embeddings E_1, E_2, which are passed through a projection head for contrastive learning.
- **Joint Contrastive and Supervised Objectives.** We introduce two contrastive branches: (i) a *structure-aware* contrastive module (Figure 2-B), which compares nodes based on their structural overlap to enforce topological consistency; and (ii) a *label-aware* contrastive module (Figure 2-C), which estimates node homophily and adaptively selects semantically aligned positive/negative samples. These contrastive signals are integrated into a unified loss, alongside a supervised classification loss over labeled nodes. Our design is further enhanced by a granularity-aware contrastive learning module to support fine-grained semantic alignment.

3.1 Hypergraph Augmentation

Given the original hypergraph $\mathcal{G} = (\mathbf{X}, \mathbf{H})$, HeroCL generates two augmented views $\mathcal{G}_1 = (\mathbf{X}_1, \mathbf{H}_1)$ and $\mathcal{G}_2 = (\mathbf{X}_2, \mathbf{H}_2)$ by applying perturbations to both node features and hypergraph structure. For node-level augmentation, we randomly mask positions in the node feature matrix $\mathbf{X}$ and inject Gaussian noise at the masked locations to retain feature diversity. For structure-level augmentation, a subset of node-hyperedge connections in the incidence matrix $\mathbf{H}$ is randomly masked to alter the topological structure. These augmentations introduce both semantic and structural diversity across views.

3.2 Multi-Hop Neighbor Encoding

In heterophilic hypergraphs, nodes within the same hyperedge often belong to different classes, resulting in semantic conflicts during message passing. The conventional "node $\rightarrow$ hyperedge $\rightarrow$ node" propagation can aggregate incompatible features, undermining representation quality. To mitigate this, we introduce a multi-hop neighbor encoding scheme that captures latent semantic correlations by leveraging both first-order and second-order structural information.

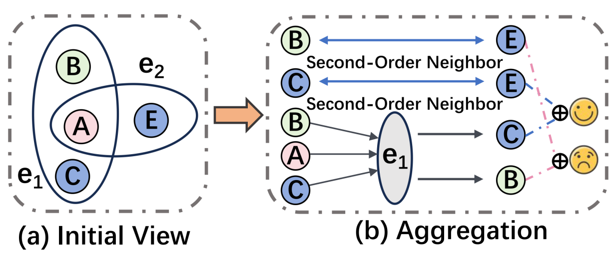

Figure 3: Encoding process of HeroCL for generating node and hyperedge representations.

First-Order Propagation. For each augmented view $\mathcal{H}_1$ and $\mathcal{H}_2$, standard hypergraph message passing is performed as:

$$\mathbf{X}_e^{(l)} = \mathbf{D}_e^{-1}\mathbf{H}^\top\mathbf{X}^{(l-1)}, \quad \mathbf{X}_v^{(l)} = \mathbf{D}_v^{-1}\mathbf{H}\mathbf{X}_e^{(l)}, \quad (2)$$

where $\mathbf{H}$ is the incidence matrix, $\mathbf{D}_v$ and $\mathbf{D}_e$ are the node and hyperedge degree matrices, and $\mathbf{X}_v^{(l)}$ denotes the node representations after the l-th layer.

Second-Order Aggregation. As shown in Figure 3, we define second-order neighbors based on two-hop connectivity through shared hyperedges. A second-order adjacency matrix $\hat{A}^{(2)}$ is constructed accordingly. To filter noisy neighbors, we compute a feature similarity matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$ using cosine similarity: $\mathbf{S}_{i,j} = \frac{x_i^\top x_j}{\|x_i\| \cdot \|x_j\|}$ and apply it to reweight $\hat{A}^{(2)}$ via element-wise multiplication. The second-order propagation is then defined as:

$$\mathbf{X}_{2hop}^{(l)} = \hat{\mathbf{D}}_2^{-1} \left(\hat{\mathbf{A}}^{(2)} \odot \mathbf{S} \right) \hat{\mathbf{D}}_2^{-1} \mathbf{X}^{(l-1)} \mathbf{W}, \quad (3)$$

where $\hat{\mathbf{D}}_2$ is the degree matrix of the reweighted second-order graph, W is a learnable transformation, and $\odot$ denotes the Hadamard product.

Embedding Fusion and Projection. The final node embedding is obtained by weighted residual fusion:

$$\mathbf{X}_v^{(l)} = \alpha_{2hop} \cdot \mathbf{X}_v^{(l)} + (1 - \alpha_{2hop}) \cdot \mathbf{X}_{2hop}^{(l)}, \quad (4)$$

where $\alpha_{2hop} \in [0, 1]$ controls the contribution of second-order neighbors. To prepare for contrastive learning, we apply separate two-layer MLPs to obtain projected embeddings:

$$\mathbf{Z} = \text{proj}_v(\mathbf{X}_v^{(l)}), \quad \mathbf{E} = \text{proj}_e(\mathbf{X}_e^{(l)}). \quad (5)$$

3.3 Label-Aware Contrastive Learning

To enhance semantic consistency in node representations, we introduce a label-aware contrastive learning module that leverages available label supervision. This component is designed to encourage nodes sharing the same label to have similar embeddings, even when structural connections are weak or inconsistent, which indeed is an issue commonly observed in heterophilic hypergraphs. The framework consists of two core steps: label homophily modeling and contrastive loss formulation.

To quantify the label coherence of a node's local neighborhood, we define a label homophily score H_v for each labeled node v which measures the average proportion of same-label nodes across its incident hyperedges:

$$H_v = \frac{1}{|R_v|} \sum_{e_j \in R_v} \frac{|\{u \in e_j \mid 1(y_u = y_v)|}{n_j}, \quad (6)$$

where R_v denotes the set of hyperedges containing node v, $n_j = |e_j|$ is the size of hyperedge e_j, and y_u, y_v represent the labels of nodes u and v, respectively.

Based on these scores, we compute a label similarity weight matrix to softly quantify semantic alignment between node pairs: $W_{i,j} = \exp(-|H_i - H_j|)$, where H_i is defined above in Eq. (6) . Let $z_i^{(1)}$ and $z_i^{(2)}$ denote the embeddings of node $i \in V_l$ from two augmented views. For each labeled node i, we define its positive sample j as a randomly selected node sharing the same label ($y_j = y_i, j \neq i$), and its negative samples as all nodes with different labels: $\mathcal{N}(i) = \{k \mid y_k \neq y_i\}$. The label-aware contrastive loss for each view is computed as: The label-aware contrastive loss for each view $t \in \{1, 2\}$ is defined as:

$$\ell_l^{(t)}(z_i^{(t)}, z_j^{(t)}) = -\log \frac{\exp(s(z_i^{(t)}, z_j^{(t)})/\tau_l)}{\sum\limits_{k \in \mathcal{N}(i) \cup \{j\}} \exp(s(z_i^{(t)}, z_k^{(t)})/\tau_l)},$$

where $s(\cdot, \cdot)$ denotes cosine similarity and τ_l is a temperature hyperparameter.

The overall *label-aware* contrastive loss aggregates across all labeled node pairs, weighted by their label similarity:

$$\mathcal{L}_l = \frac{1}{2|V_l|} \sum_{i \in V_l} \sum_{j \in V_l} W_{i,j} \Big(\ell_l^{(1)}(z_i^{(1)}, z_j^{(1)}) + \ell_l^{(2)}(z_i^{(2)}, z_j^{(2)}) \Big).$$

3.4 Structure-Aware Contrastive Learning

While heterophilic hypergraphs often connect nodes with dissimilar labels, structural relationships can still provide valuable learning signals. Motivated by the concept of hyperedge set overlap (Lee, Choe, and Shin 2021), we define a structural similarity function between nodes v and u as:

$$\text{Sim}(v, u) = \frac{|R_v \cap R_u|}{|R_v \cup R_u|}, \quad (7)$$

where R_v and R_u denote the sets of hyperedges incident to nodes v and u, respectively. By definition, $\text{Sim}(v, v) = 1$, indicating perfect self-similarity.

To select structurally aligned node pairs, we introduce a threshold $\tau \in [0, 1]$ and define the positive sample set as $\mathcal{P} = \{(v, u) \mid \text{Sim}(v, u) \geq \tau, v \neq u\}$. For each pair $(i, j) \in \mathcal{P}$, we compute the structure-aware contrastive loss on both augmented views. Letting $t \in \{1, 2\}$ denote the view index, the per-view contrastive loss is defined as:

$$\ell_s^{(t)}(z_i^{(t)}, z_j^{(t)}) = -\log \frac{\exp\left(s(z_i^{(t)}, z_j^{(t)})/\tau_s\right)}{\sum_{k \in \mathcal{V}} \exp\left(s(z_i^{(t)}, z_k^{(t)})/\tau_s\right)},$$

where $s(\cdot, \cdot)$ is the cosine similarity, and τ_s is a temperature hyperparameter.

The overall *structure-aware* contrastive loss aggregates these terms over all positive node pairs:

$$\mathcal{L}_s = \frac{1}{2|\mathcal{P}|} \sum_{(i,j) \in \mathcal{P}} W_{i,j} \left(\ell_s^{(1)}(z_i^{(1)}, z_j^{(1)}) + \ell_s^{(2)}(z_i^{(2)}, z_j^{(2)}) \right).$$

3.5 Granularity-Aware Contrastive Learning

In addition to the label-aware and structure-aware contrastive objectives, we incorporate a granularity-aware contrastive learning module to enhance representation consistency across different levels of the hypergraph structure. Specifically, this component models contrastive signals at the node, hyperedge (group), and node–hyperedge (member) levels, allowing the model to capture fine-grained semantic alignments within and across hypergraph components. Inspired by the design principle of TriCL (Lee and Shin 2023), we adapt and integrate its hierarchical contrastive scheme into HeroCL to complement our heterophily-aware design.

Let z_i and e_j denote the embeddings of node v_i and hyperedge e_j, respectively. Cosine similarity $s(\cdot,\cdot)$ is used as the similarity function, with separate temperature parameters τ_n, τ_g, τ_m for each contrastive objective.

Node-Level Contrastive Loss. Each node v_i has two views $z_i^{(1)}$ and $z_i^{(2)}$. The node-level contrastive loss is defined as:

$$\mathcal{L}_n = \frac{1}{2|\mathcal{V}|}\sum_{i=1}^{|\mathcal{V}|}\left(\ell_n(z_i^{(1)}, z_i^{(2)}) + \ell_n(z_i^{(2)}, z_i^{(1)})\right),$$

$$\ell_n(z_i^{(1)}, z_i^{(2)}) = -\log\frac{\exp(s(z_i^{(1)}, z_i^{(2)})/\tau_n)}{\sum_{k=1}^{|\mathcal{V}|}\exp(s(z_i^{(1)}, z_k^{(2)})/\tau_n)}.$$

Group-Level Contrastive Loss. For each hyperedge e_j, the two views $e_j^{(1)}$ and $e_j^{(2)}$ are contrasted as:

$$\mathcal{L}_g = \frac{1}{2|\mathcal{E}|}\sum_{j=1}^{|\mathcal{E}|}\left(\ell_g(e_j^{(1)}, e_j^{(2)}) + \ell_g(e_j^{(2)}, e_j^{(1)})\right),$$

$$\ell_g(e_j^{(1)}, e_j^{(2)}) = -\log\frac{\exp(s(e_j^{(1)}, e_j^{(2)})/\tau_g)}{\sum_{k=1}^{|\mathcal{E}|}\exp(s(e_j^{(1)}, e_k^{(2)})/\tau_g)}.$$

Member-Level Contrastive Loss. We use $h_{ij} \in \{0, 1\}$ to denote the element of the incidence matrix $\mathbf{H}$, where $h_{ij} = 1$ if node $v_i \in e_j$, and $h_{ij} = 0$ otherwise. We define a discriminator function $D(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ to measure the similarity between node and hyperedge embeddings. Therefore, the member-level contrastive loss is defined as:

$$\mathcal{L}_m = \frac{1}{2K}\sum_{i=1}^{|\mathcal{V}|}\sum_{j=1}^{|\mathcal{E}|} h_{ij}\cdot\left[\ell_m(z_i^{(1)}, e_j^{(2)}) + \ell_m(z_i^{(2)}, e_j^{(1)})\right],$$

$$\ell_m(z_i, e_j) = -\log\frac{\exp(D(z_i, e_j)/\tau_m)}{\sum_{k=1}^{|\mathcal{V}|}\exp(D(z_k, e_j)/\tau_m)} - \log\frac{\exp(D(z_i, e_j)/\tau_m)}{\sum_{k=1}^{|\mathcal{E}|}\exp(D(z_i, e_k)/\tau_m)},$$

where $K = \sum_{i=1}^{|\mathcal{V}|}\sum_{j=1}^{|\mathcal{E}|} h_{ij}$ is the total number of positive node–hyperedge pairs.

The overall granularity-aware contrastive loss is defined as a weighted sum of the three components:

$$\mathcal{L}_{gra} = \lambda_n\mathcal{L}_n + \lambda_g\mathcal{L}_g + \lambda_m\mathcal{L}_m, \tag{8}$$

where $\lambda_n, \lambda_g, \lambda_m$ are hyperparameters that control the contribution of each term.

3.6 Training Objective

To enable effective representation learning under heterophily and label scarcity, we combine both supervised and contrastive objectives into a unified training framework. Specifically, we use a standard cross-entropy loss over labeled nodes to encourage class-discriminative embeddings:

$$\mathcal{L}_{\text{cls}} = -\sum_{v_i\in\mathcal{V}_L}\sum_{c=1}^{C} y_{i,c}\log\hat{y}_{i,c}, \tag{9}$$

where $\hat{\mathbf{y}}_i$ denotes the predicted class distribution for node v_i, and $\mathbf{y}_i$ is the one-hot encoded ground-truth label.

The final training objective of HeroCL integrates the label-aware contrastive loss $\mathcal{L}_l$, the structure-aware contrastive loss $\mathcal{L}_s$, the granularity-aware contrastive loss $\mathcal{L}_{gra}$, and the supervised classification loss:

$$\mathcal{L}_{\text{total}} = \lambda_l\mathcal{L}_l + \lambda_s\mathcal{L}_s + \mathcal{L}_{gra} + \mathcal{L}_{\text{cls}}, \tag{10}$$

where λ_l and λ_s are hyperparameters that balance the contributions of the label-aware and structure-aware components.

4 Experiments

4.1 Experimental Setups

Datasets. We evaluate HeroCL on 11 benchmark datasets. Following common practice, we group them into two categories based on the hyperedge homophily ratio: homophilic ($H_{\text{edge}} > 0.5$) and heterophilic datasets ($H_{\text{edge}} \leq 0.5$).

Baselines and Experimental Settings. We compare HeroCL with 10 baselines, including MLP, HGNN (Feng et al. 2019), HyperGCN (Yadati et al. 2019), UniGCNII (Huang and Yang 2021), HyperND (Prokopchik, Benson, and Tudisco 2022), AllDeepSets (Chien et al. 2022), AllSetTransformer (Chien et al. 2022), ED-HNN (Wang et al. 2023), SheafHyperGNN (Duta et al. 2023), and HyperUFG (Li et al. 2025c). Notably, ED-HNN and SheafHyperGNN include heterophilic datasets in their evaluations, whereas HyperUFG is the first method tailored to heterophilic hypergraphs; thus we group baselines by their relevance to heterophily. All baselines are re-implemented from official repositories with hyperparameters set as in the original papers. For heterophilic datasets, we follow HyperUFG's split (40/20/40 train/val/test). Each experiment is run 10 times with different seeds, and we report mean accuracy ± standard deviation.

4.2 Overall Performance Comparison

Table 1 presents node classification accuracies on five homophilic hypergraph datasets. HeroCL achieves the best performance on four out of five datasets and ranks first overall, showcasing its strong capacity to capture class-consistent structures in homophilic settings. In particular, it outperforms the second-best method by 1.5%, 0.2%, 1.9%, and 0.8% on Cora, Citeseer, Pubmed, and Cora-CA, respectively. While slightly trailing ED-HNN on DBLP-CA, the performance gap remains small. Table 2 summarizes results on six heterophilic datasets (Actor, Amazon, Twitch, Pokec, Senate, and House), where class-inconsistent neighborhoods pose greater modeling challenges. Despite the datasets' structural and semantic diversity, HeroCL consistently ranks among the top performers, achieving the best accuracy on Senate and House with margins exceeding 6.6% and 3.4%, respectively. It also shows competitive stability on noisy datasets (Pokec, Twitch), indicating robustness to complex neighbor-label distributions.

4.3 Robustness under Varying Label Ratios

Figure 4 shows classification accuracy on Cora and Senate across different label ratios. On the homophilic Cora dataset,

Model	Cora (0.75)	Citeseer (0.68)	Pubmed (0.78)	Cora-CA (0.78)	DBLP-CA (0.87)	Ave. Rank
MLP	75.16 ± 1.41	71.71 ± 1.01	87.20 ± 0.34	75.17 ± 1.41	84.37 ± 0.33	10.0
HGNN	79.39 ± 1.36	72.45 ± 1.16	86.44 ± 0.44	82.64 ± 1.65	91.03 ± 0.20	7.60
HyperGCN	78.45 ± 1.26	71.28 ± 0.82	82.84 ± 8.67	79.48 ± 2.08	89.38 ± 0.25	10.0
UniGCNII	78.81 ± 1.05	73.05 ± 2.21	88.25 ± 0.40	83.60 ± 1.14	91.69 ± 0.19	5.40
HyperND	79.20 ± 1.14	72.62 ± 1.49	86.68 ± 0.43	80.62 ± 1.32	90.35 ± 0.26	8.00
AllDeepSets	76.88 ± 1.80	70.83 ± 1.63	88.75 ± 0.33	81.97 ± 1.50	91.27 ± 0.27	7.80
AllSetTransformer	78.58 ± 1.47	73.08 ± 1.20	88.72 ± 0.37	83.63 ± 1.47	91.53 ± 0.23	5.80
ED-HNN	80.31 ± 1.35	73.70 ± 1.38	89.03 ± 0.53	83.97 ± 1.55	91.90 ± 0.19	3.00
SheafHyperGNN	81.30 ± 1.70	74.71 ± 1.23	87.68 ± 0.60	85.52 ± 1.28	91.59 ± 0.24	4.00
HyperUFG	81.51 ± 0.99	74.72 ± 2.10	88.73 ± 0.42	85.18 ± 0.69	91.67 ± 0.31	2.80
HeroCL (Ours)	83.03 ± 0.98	74.96 ± 0.35	90.99 ± 1.01	86.37 ± 1.08	91.61 ± 0.21	1.60

Table 1: Node classification accuracy (%) on homophilic hypergraph datasets, with H_{edge} values shown in parentheses next to each dataset name. The best results are highlighted in red, and the second-best in cyan.

Model	Actor (0.47)	Amazon (0.37)	Twitch (0.49)	Pokec (0.45)	Senate (0.46)	House (0.49)	Ave. Rank
MLP	85.45 ± 1.21	26.70 ± 2.82	52.77 ± 1.81	56.92 ± 2.46	52.25 ± 5.17	51.86 ± 2.34	5.17
HGNN	74.47 ± 0.32	23.79 ± 0.24	51.88 ± 0.26	49.82 ± 0.27	48.59 ± 4.52	61.39 ± 2.96	8.33
HyperGCN	68.67 ± 4.38	22.53 ± 3.94	51.32 ± 1.02	52.43 ± 3.68	42.45 ± 3.67	48.32 ± 2.93	9.50
UniGCNII	80.48 ± 1.13	26.63 ± 1.32	50.84 ± 0.76	54.25 ± 2.70	49.30 ± 4.25	67.25 ± 2.57	7.50
HyperND	92.52 ± 0.81	26.08 ± 0.33	51.44 ± 0.67	55.94 ± 0.45	52.82 ± 3.20	51.70 ± 3.37	5.67
AllDeepSets	82.00 ± 2.33	18.60 ± 0.17	50.72 ± 0.96	51.11 ± 1.04	48.17 ± 5.67	67.82 ± 2.40	8.92
AllSetTransformer	83.39 ± 1.73	18.60 ± 0.17	50.45 ± 0.76	58.40 ± 0.42	51.83 ± 5.22	69.33 ± 2.20	7.25
ED-HNN	91.86 ± 0.43	26.21 ± 0.36	50.86 ± 0.88	59.11 ± 0.57	64.79 ± 5.14	72.45 ± 2.28	4.50
SheafHyperGNN	80.09 ± 2.45	26.93 ± 3.04	51.03 ± 0.76	55.34 ± 4.39	68.73 ± 4.68	73.84 ± 2.30	5.00
HyperUFG	89.32 ± 0.75	40.53 ± 2.25	52.35 ± 0.04	62.30 ± 0.12	67.61 ± 7.00	72.82 ± 2.22	2.33
HeroCL (Ours)	85.58 ± 0.23	30.81 ± 0.28	52.82 ± 0.12	59.83 ± 0.28	75.35 ± 4.46	77.28 ± 3.10	1.83

Table 2: Node classification accuracy (%) on heterophilic hypergraph datasets, with H_{edge} values shown in parentheses next to each dataset name. The best results are highlighted in red, and the second-best in cyan.

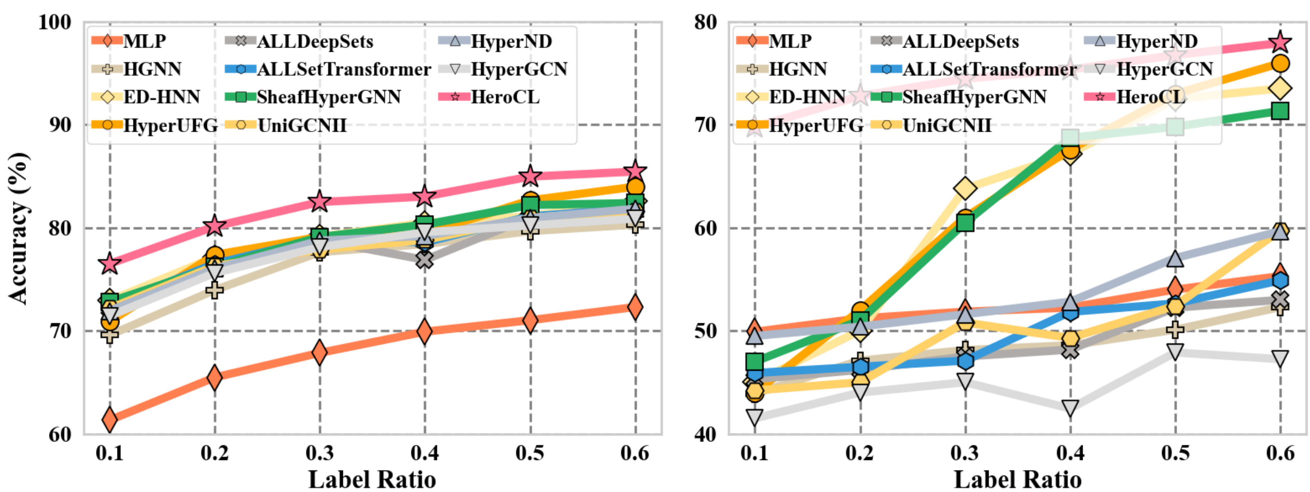

Figure 4: Classification accuracy (%) on Cora (left) and Senate (right) datasets under varying label ratios. The results illustrate model robustness with respect to supervision sparsity in homophilic (Cora) and heterophilic (Senate) settings.

all methods degrade smoothly as supervision decreases, consistently outperforming the MLP baseline. In contrast, on the heterophilic Senate dataset, most baselines suffer severe performance drops under low label availability, occasionally falling below MLP. HeroCL, however, retains strong performance even at low label ratios. These results indicate that HeroCL attains higher accuracy across homophilic and heterophilic settings, particularly with limited supervision, with good robustness owing to the integration of label- and structure-aware contrastive objectives..

4.4 Ablation Study

To evaluate the contribution of each key component in HeroCL, we perform a comprehensive ablation study with four model variants. Specifically, **w/o LabelCL** removes the label-aware contrastive module; **w/o StructCL** disables the structure-aware contrastive module; **w/o LS-CL** eliminates both; and **w/o MHNE** replaces our Multi-Hop Neighbor Encoder with a standard HNN architecture (Feng et al. 2019).

Table 3 summarizes the performance of these variants across four representative datasets. We draw the following key observations: i) The label-aware contrastive learning component consistently improves performance across all datasets, supporting our theoretical motivation that aligning embeddings of semantically similar nodes enhances representation quality and discriminative power. ii): The structure-aware contrastive learning component offers limited gains on homophilic hypergraphs and can even slightly degrade performance (e.g., higher accuracy on Cora-CA and Citeseer after its removal), indicating that structural signals may be redundant or noisy when local neighborhoods already exhibit strong label consistency. iii): Combining both contrastive components with MHNE yields the best results, especially on heterophilic datasets such as Senate and House, confirming that MHNE is crucial for capturing higher-order dependencies and that the dual contrastive objectives jointly strengthen the exploitation of structural and semantic signals under heterophily.

Model Variant	Cora-CA	Citeseer	Senate	House
w/o LabelCL	83.18±1.42	72.97±0.50	75.28±4.47	76.52±1.20
w/o StructCL	86.54±1.33	75.18±0.43	75.30±4.63	76.57±1.28
w/o LS-CL	83.24±1.67	73.37±0.33	75.18±4.61	76.48±1.19
w/o MHNE	85.37±0.75	74.29±0.36	73.82±4.12	76.64±1.19
Full	86.37±1.08	74.96±0.35	75.35±4.46	77.28±3.10

Table 3: Ablation results on four datasets.

4.5 Hyperparameter Sensitivity Analysis

To investigate the influence of key hyperparameters in HeroCL, particularly those introduced in Sections 3.3 3.4, 3.5, we present a detailed study of the structure-aware and label-aware contrastive loss weights (λ_s and λ_l) on both homophilic (Cora-CA) and heterophilic (House) hypergraphs.

Effects of λ_s and λ_l. As shown in Figure 5, moderate values of the structure-aware contrastive weight λ_s (e.g., 0.3–0.5) yield the best performance on Cora-CA, while overly large values lead to a decline, likely due to overemphasis on redundant local structure in homophilic graphs. In contrast, the label-aware contrastive weight λ_l provides consistent benefits across a wide range, validating its role in enforcing semantic consistency. On House, both λ_s and λ_l exhibit stable performance across different settings, indicating that these complementary contrastive signals contribute to improved generalization and robustness under heterophilic conditions. These results confirm the effectiveness of the dual contrastive scheme: label-aware contrast provides stable semantic benefits across both settings, while structure-aware contrast requires careful tuning under homophilic hypergraphs to avoid potential redundancy.

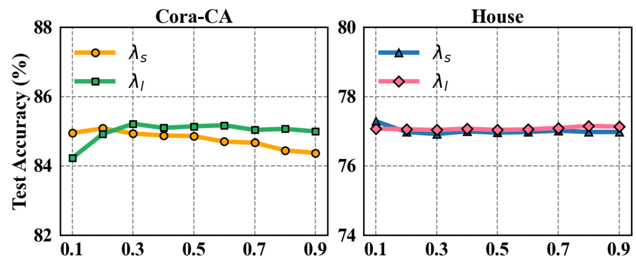

Figure 5: Impact of λ_s, λ_l.

5 Related Work

Hypergraph neural networks have been widely studied for capturing high-order interactions (Yang and Xu 2025). Early approaches such as HGNN (Feng et al. 2019) and HyperGCN (Yadati et al. 2019) extended graph convolution to hypergraphs via spectral filtering and clique expansion. UniGCN (Huang and Yang 2021) unified various message-passing paradigms into a general framework, while AllDeepSets and AllSetTransformer (Chien et al. 2022) introduced permutation-invariant set functions to model hyperedges more flexibly. Although some methods like ED-HNN (Wang et al. 2023) and SheafHyperGNNs (Duta et al. 2023) briefly touch on heterophily in their experiments, they do not explicitly target heterophilic hypergraph learning. In contrast, HyperUFG (Li et al. 2025c) is the first to systematically address this challenge through framelet-based spectral design tailored for heterophilic structures. Meanwhile, contrastive learning has emerged as a promising direction in hypergraph representation. TriCL (Lee and Shin 2023) employs contrastive objectives at multiple granularities (node, group, member), and CHGNN (Song et al. 2024) integrates label-guided contrastive loss to enhance semantic consistency and class discrimination. Our work presents the first contrastive framework tailored to heterophilic hypergraphs, addressing both semantic inconsistency and label scarcity.

6 Conclusion

This work studies learning from heterophilic hypergraphs with limited supervision, where existing methods assuming homophily or requiring many labels are less effective. We propose HeroCL, a contrastive learning framework that combines multi-hop neighbor encoding, dual-view contrastive objectives, and multi-granularity alignment, enabling nodes to preserve their semantic identities while mitigating noise from heterophilic neighbors. As a future direction, it is desirable to extend this framework to heterogeneous heterophilic hypergraph learning.

Acknowledgements

This work was supported in part by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang (No. 2024C03262), and the National Natural Science Foundation of China (No. 62536006, No. 62172370, No. 62576371, No. U23A20388, No. 62320106007).

References

Antelmi, A.; Cordasco, G.; Polato, M.; Scarano, V.; Spagnuolo, C.; and Yang, D. 2023. A survey on hypergraph representation learning. *ACM Computing Surveys*, 56(1): 1–38.

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

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

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

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

Kim, S.; Lee, S. Y.; Gao, Y.; Antelmi, A.; Polato, M.; and Shin, K. 2024. A survey on hypergraph neural networks: An in-depth and step-by-step guide. In *KDD*, 6534–6544.

Lee, D.; and Shin, K. 2023. I'm me, we're us, and i'm us: Tri-directional contrastive learning on hypergraphs. In *AAAI*, 8456–8464.

Lee, G.; Choe, M.; and Shin, K. 2021. How do hyperedges overlap in real-world hypergraphs?-patterns, measures, and generators. In *Proceedings of the web conference 2021*, 3396–3407.

Li, M.; Cheng, Y.; Bai, L.; Cao, F.; Lv, K.; Liang, J.; and Lio, P. 2025a. EduLLM: Leveraging Large Language Models and Framelet-Based Signed Hypergraph Neural Networks for Student Performance Prediction. In *ICML*, 34119–3414.

Li, M.; Fang, Y.; Wang, Y.; Feng, H.; Gu, Y.; Bai, L.; and Lio, P. 2025b. Deep hypergraph neural networks with tight framelets. In *AAAI*, 18385–18392.

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

Li, M.; Li, Z.; Huang, C.; Jiang, Y.; and Wu, X. 2024. EduGraph: Learning path-based hypergraph neural networks for mooc course recommendation. *IEEE Transactions on Big Data*, 10(6): 706–719.

Millán, A. P.; Sun, H.; Giambagli, L.; Muolo, R.; Carletti, T.; Torres, J. J.; Radicchi, F.; Kurths, J.; and Bianconi, G. 2025. Topology shapes dynamics of higher-order networks. *Nature Physics*, 21: 353—361.

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

Song, Y.; Gu, Y.; Li, T.; Qi, J.; Liu, Z.; Jensen, C. S.; and Yu, G. 2024. CHGNN: a semi-supervised contrastive hypergraph learning network. *IEEE Transactions on Knowledge and Data Engineering*, 36(9): 4515–4530.

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

Wang, Y.; and Kleinberg, J. 2024. From Graphs to Hypergraphs: Hypergraph Projection and its Reconstruction. In *ICLR*.

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

Yang, M.; and Xu, X.-J. 2025. Recent advances in hypergraph neural networks. *arXiv preprint arXiv:2503.07959*.