Hypergraph Self-supervised Learning with Sampling-efficient Signals

Fan Li[1] , **Xiaoyang Wang**[1*] , **Dawei Cheng**[2] , **Wenjie Zhang**[1] , **Ying Zhang**[3] and **Xuemin Lin**[4]

[1]University of New South Wales, Sydney, Australia
[2]Tongji University, Shanghai, China
[3]University of Technology Sydney, Sydney, Australia
[4]Shanghai Jiaotong University, Shanghai, China

{fan.li8, xiaoyang.wang1,wenjie.zhang}@unsw.edu.au, dcheng@tongji.edu.cn, ying.zhang@uts.edu.au,
xuemin.lin@gmail.com

Abstract

Self-supervised learning (SSL) provides a promising alternative for representation learning on hypergraphs without costly labels. However, existing hypergraph SSL models are mostly based on contrastive methods with the instance-level discrimination strategy, suffering from two significant limitations: (1) They select negative samples arbitrarily, which is unreliable in deciding similar and dissimilar pairs, causing training bias. (2) They often require a large number of negative samples, resulting in expensive computational costs. To address the above issues, we propose SE-HSSL, a hypergraph SSL framework with three sampling-efficient self-supervised signals. Specifically, we introduce two sampling-free objectives leveraging the canonical correlation analysis as the node-level and group-level self-supervised signals. Additionally, we develop a novel hierarchical membership-level contrast objective motivated by the cascading overlap relationship in hypergraphs, which can further reduce membership sampling bias and improve the efficiency of sample utilization. Through comprehensive experiments on 7 real-world hypergraphs, we demonstrate the superiority of our approach over the state-of-the-art method in terms of both effectiveness and efficiency.

1 Introduction

The hypergraph, where each hyperedge can connect any number of nodes, is a generalization data structure of the simple pairwise graph. It provides a natural and expressive way to model complex high-order relationships among entities in diverse real-world applications such as recommender systems [Xia *et al.*, 2022b], computer vision [Yu *et al.*, 2012], and neuroscience [Xiao *et al.*, 2019].

In recent years, hypergraph representation learning has attracted increasing attention from both academia and industry to deal with hypergraph data [Yi and Park, 2020; Jia *et al.*, 2021; Antelmi *et al.*, 2023]. Hypergraph neural networks (HGNNs) have become a popular tool in this domain,

demonstrating remarkable effectiveness [Feng *et al.*, 2019; Chien *et al.*, 2021]. However, task-specific labels can be extremely scarce in hypergraph datasets [Wei *et al.*, 2022], and deep models are prone to overfitting when trained with sparse supervised signals [Thakoor *et al.*, 2021]. Therefore, it is crucial yet challenging to explore self-supervised learning for HGNNs, where only limited or even no labels are needed.

Current hypergraph SSL models are mostly contrastive-based, which aim to maximize the agreement between two augmented views derived from the original hypergraph. Previous studies, such as HCCF [Xia *et al.*, 2022a] and S^2-HHGR [Zhang *et al.*, 2021b], employ contrastive strategies to mitigate label scarcity in hypergraph-based recommender systems, but they are not general methods designed for hypergraph contrastive learning (HCL). HyperGCL [Wei *et al.*, 2022], as the first comprehensive hypergraph SSL method, focuses on generating augmented views that better preserve high-order relations compared to fabricated augmentations. TriCL [Lee and Shin, 2023] proposes tri-directional contrast objectives in hypergraphs to capture multi-view structural information, achieving state-of-the-art performance.

Nevertheless, existing contrastive-based approaches suffer from two limitations: (1) They employ the instance-level discrimination strategy, that is, pulling together the representation of the same node/hyperedge in the two augmented hypergraphs while pushing apart every other node/hyperedge pairs. However, this approach is unreliable for determining similar/dissimilar pairs. For example, consider a co-author hypergraph, papers (nodes) published by the same author (hyperedge) are more likely to have similar topics. Authors who co-publish (overlap) a significant number of papers may share similar research interests. Treating them as negative pairs arbitrarily would introduce training bias, resulting in less discriminative representations. (2) They require a large number of negative pairs to achieve the best performance. However, generating a large number of negative pairs often comes with prohibitive computational costs, particularly for large hypergraphs. In TriCL, in addition to node-level and group-level contrast, it introduces membership-level contrast to maximize the agreements between each group (hyperedge) and its members (nodes) in two augmented views. Although this signal has been proven to be highly effective, it requires a large number of node-hyperedge negative pairs to achieve the SOTA performance, leading to a complexity

---

of $O(|\mathcal{V}| \times |\mathcal{E}|)$ in scoring function computation, where $\mathcal{V}$ and $\mathcal{E}$ denote the number of nodes and hyperedges, respectively. This would greatly limit the training speed. Moreover, it also follows the instance-level contrast strategy when handling positive/negative membership pairs, which can introduce the training bias mentioned above.

To address the above issues, we introduce SE-HSSL, a sampling-efficient framework for hypergraph self-supervised learning. Specifically, it first employs an HGNN to encode the original hypergraph and two augmented views, generating multi-view node and hyperedge representations. Then, to capture high-order relations in hypergraphs and reduce sampling bias caused by previous contrastive-based methods, we propose tri-directional sampling-efficient self-supervised signals. For node-level and group-level learning, we design two optimization objectives based on the idea of canonical correlation analysis (CCA) [Andrew *et al.*, 2013]. These two signals allow the model to maximize the agreement between two augmented views without reliance on any negative sample. Furthermore, instead of relying on instance-level membership discrimination, which focuses on distinguishing between "real" and "fake" node-hyperedge memberships, we present a novel hierarchical membership-level contrast signal based on the overlap structure within hypergraphs. This design effectively addresses sampling bias in membership-level learning and substantially reduces the number of negative pairs, leading to improved effectiveness and efficiency. Finally, the model jointly optimizes the three objectives, facilitating comprehensive representation learning in hypergraphs. Our main contributions are summarized as follows:

- We propose SE-HSSL, a hypergraph self-supervised learning method that can address training bias and sampling inefficiency challenges in existing HCL methods.

- In our HSSL framework, we introduce sampling-free CCA-based node- and group-level objectives to reduce training bias and generate more discriminative representations. Furthermore, our novel hierarchical membership contrast signal can effectively preserve membership relations with only a few sampled node-hyperedge pairs.

- We conduct a comprehensive evaluation of SE-HSSL on 7 real-world datasets. The experimental results demonstrate its superiority over strong baselines across various benchmark downstream tasks. The efficiency test shows that it can achieve at least 2x speedup during training compared to the state-of-the-art on most datasets.

2 Preliminary

Notations. Let $\mathcal{H}(\mathcal{V}, \mathcal{E})$ represent a hypergraph with vertex set $\mathcal{V} = \{v_i\}_{i=1}^{|\mathcal{V}|}$ and hyperedge set $\mathcal{E} = \{e_j\}_{j=1}^{|\mathcal{E}|}$. $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times F}$ denotes the F dimensional node feature matrix. As each hyperedge $e \in \mathcal{E}$ is a subset of $\mathcal{V}$, we denote $|e|$ as the size of e. Each hyperedge e_j is associated with a positive number w_j as hyperedge weight, and all the weights formulate a diagonal matrix $\mathbf{W} \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|}$. The hypergraph structure can be represented by an incidence matrix $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}|}$, where each entry $h_{ij} = 1$ if $v_i \in e_j$ and $h_{ij} = 0$ otherwise. We represent the degree of vertices using the diagonal matrix

$\mathbf{D}_v \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, where each entry is $d(v_i) = \sum_{e_j \in \mathcal{E}} w_j \cdot h_{ij}$. The degree of hyperedges is denoted by the diagonal matrix $\mathbf{D}_e \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|}$, with each element $\delta(e_j) = \sum_{v_i \in e_j} h_{ij}$ representing the number of nodes connected by e_j.

Hypergraph Neural Networks (HGNNs). HGNNs apply a two-stage neighborhood aggregation scheme to learn hypergraph representations. This entails updating the hyperedge representation by aggregating representations of its incident nodes, and updating the node representation by propagating information from representations of its incident hyperedges:

$$
\begin{aligned}
z_{e,j}^{(l)} &= f_{\mathcal{V} \to \mathcal{E}}^{(l)}(z_{e,j}^{(l-1)}, \{z_{v,k}^{(l-1)} | v_k \in e_j\}) \\
z_{v,i}^{(l)} &= f_{\mathcal{E} \to \mathcal{V}}^{(l)}(z_{v,i}^{(l-1)}, \{z_{e,k}^{(l)} | v_i \in e_k\})
\end{aligned}
\tag{1}
$$

where $z_{e,j}^{(l)}, z_{v,i}^{(l)}$ are embeddings of e_j and v_i at layer l, respectively. $f_{\mathcal{V} \to \mathcal{E}}^{(l)}$ and $f_{\mathcal{E} \to \mathcal{V}}^{(l)}$ are two permutation-invariant functions which aggregates information from nodes and hyperedges, respectively, at the l-th layer. Recent works have proposed different $f_{\mathcal{V} \to \mathcal{E}}^{(l)}$ and $f_{\mathcal{E} \to \mathcal{V}}^{(l)}$, leading to multiple HGNN variants [Kim *et al.*, 2020; Chien *et al.*, 2021].

Deep Canonical Correlation Analysis (DCCA). CCA is a classical multivariate analysis method designed to identify the linear combinations of variables from each set that are maximally correlated with each other. Recent research has explored the application of CCA in multi-view learning by leveraging deep neural networks instead of linear transformations [Andrew *et al.*, 2013; Jing *et al.*, 2014]. This technique helps capture complex and nonlinear relationships that are commonly shared between two sets of multidimensional vectors. Assuming X_1 and X_2 are input data from two views, the DCCA optimization objective can be formulated as:

$$
\begin{aligned}
\min_{\theta_1, \theta_2} & \mathcal{L}_C(f_{\theta_1}(X_1), f_{\theta_2}(X_2)) \\
& + \lambda(L_D(f_{\theta_1}(X_1)) + L_D(f_{\theta_2}(X_2)))
\end{aligned}
\tag{2}
$$

where $f_{\theta_1}, f_{\theta_2}$ are two neural networks, and λ is a Lagrangian multiplier. L_C is the invariance term that measures the correlation between the two views, while L_D is the feature-level decorrelation term. Following the popular setting in soft CCA [Chang *et al.*, 2018], L_D computes the L_1 distance between $f_{\theta_i}(X_i)$ and identity matrix, for $i = 1, 2$.

Problem statement. Given a hypergraph $\mathcal{H}(\mathbf{H}, \mathbf{X})$, our objective is to train an optimal hypergraph encoder f_θ : $(\mathbf{H}, \mathbf{X}) \to (\mathbf{Z}_v, \mathbf{Z}_e)$ in an unsupervised manner. This encoder maps the node and hyperedge to low-dimensional embeddings, which can be beneficial for various downstream tasks, such as node classification and node clustering.

3 Methodology

In this section, we introduce our proposed Hypergraph SSL framework. We begin by briefly presenting an overview of SE-HSSL. Then, we bring forward the details of its three modules. Finally, we describe the joint optimization strategy.
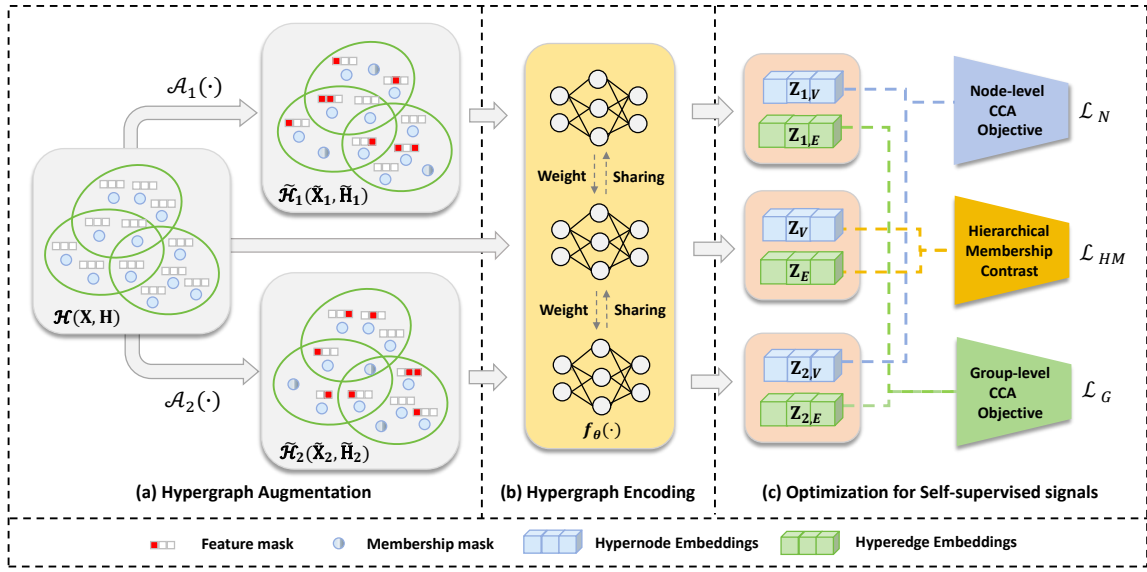
Figure 1: The architecture of the hypergraph self-supervised learning framework SE-HSSL.

3.1 Overview

As illustrated in Figure 1, our hypergraph SSL architecture consists of the following three major components: hypergraph augmentation, hypergraph encoding, and optimization for self-supervised signals. Initially, the given input hypergraph $\mathcal{H}(\mathbf{H}, \mathbf{X})$ undergoes data augmentation to obtain two correlated views $\tilde{\mathcal{H}}_1(\tilde{\mathbf{H}}_1, \tilde{\mathbf{X}}_1)$ and $\tilde{\mathcal{H}}_2(\tilde{\mathbf{H}}_2, \tilde{\mathbf{X}}_2)$. Next, we utilize a shared HGNN encoder f_θ to generate node and hyperedge representations for the original hypergraph $\mathcal{H}$ and the two augmented views, $\tilde{\mathcal{H}}_1$ and $\tilde{\mathcal{H}}_2$. While most existing HSSL approaches apply instance-level contrast signals that could lead to sampling bias and heavy computation burden, in this paper, we construct three novel sampling-efficient self-supervised signals: a node-level CCA objective, a group-level CCA objective, and a hierarchical membership-level contrast objective. Lastly, we jointly optimize the loss function of these signals, denoted as $\mathcal{L}_N$, $\mathcal{L}_G$, and $\mathcal{L}_{HM}$, respectively.

3.2 Hypergraph Augmentation

We consider two manually designed hypergraph augmentation strategies: node feature masking and membership masking. In node feature masking, we first sample a random vector $\tilde{m} \in \{0, 1\}^F$, where each dimension is independently drawn from a Bernoulli distribution, i.e., $\tilde{m}_j \sim \mathcal{B}(1 - p^f), \forall i$. The augmented feature matrix $\tilde{X}$ is then computed as follows:

$$\tilde{X} = [x_1 \odot \tilde{m}; x_2 \odot \tilde{m}, ..., x_{|\mathcal{V}|} \odot \tilde{m}]^T \quad (3)$$

where $\odot$ denotes the element-wise multiplication and $[\cdot; \cdot]$ is the concatenate operator. Regarding membership masking, we aim to disrupt high-order relations in hypergraphs by randomly kicking out vertices from hyperedges. We construct a masking matrix $\mathbf{M} \in \{0, 1\}^{|\mathcal{V}| \times |\mathcal{E}|}$ where each entry $m_{ij} \sim \mathcal{B}(1 - p^m)$. The membership masking augmentation on hypergraph topology can be formulated as:

$$\tilde{\mathbf{H}} = \mathbf{M} \odot \mathbf{H} \quad (4)$$

p^f and p^m denote drop probability for node features and node-hyperedge membership links, respectively.

3.3 Hypergraph Encoder

Following the classic hypergraph message passing scheme, we take the HGNN with the element-wise mean pooling layer as our hypergraph encoder. Formally, the l-th layer of HGNN can be represented as:

$$\begin{aligned} \mathbf{Z}_{\mathcal{E}}^{(l)} &= \phi(\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{Z}_{\mathcal{V}}^{(l-1)}\mathbf{\Theta}_{\mathcal{E}}^{(l)}) \\ \mathbf{Z}_{\mathcal{V}}^{(l)} &= \phi(\mathbf{D}_v^{-1}\mathbf{H}\mathbf{W}\mathbf{Z}_{\mathcal{E}}^{(l)}\mathbf{\Theta}_{\mathcal{V}}^{(l)}) \end{aligned} \quad (5)$$

where $\mathbf{Z}_{\mathcal{E}}^{(l)} \in \mathbb{R}^{|\mathcal{V}| \times D}, \mathbf{Z}_{\mathcal{V}}^{(l)} \in \mathbb{R}^{|\mathcal{E}| \times D}$ are hyperedge and node representations at the l-th layer. D is the embedding dimensionality. $\mathbf{\Theta}_{\mathcal{E}}^{(l)}$ and $\mathbf{\Theta}_{\mathcal{V}}^{(l)}$ denote trainable parameters for $f_{\mathcal{V} \to \mathcal{E}}^{(l)}$ and $f_{\mathcal{E} \to \mathcal{V}}^{(l)}$, respectively. $\phi(\cdot)$ represents PReLU nonlinear activation function. $\mathbf{W}$ is initialized as an identity matrix, which means equal weights for all hyperedges.

3.4 Self-supervised Signal Construction

To address the issues of training bias and heavy computation burden caused by arbitrary negative sampling, we propose tri-directional sampling-efficient self-supervised signals.

Node-level CCA objective. Most contrastive methods rely on negative pairs to avoid degenerated solutions in feature space [Zhang *et al.*, 2021a], that is, different dimensions are highly correlated and can capture the same information, making learned representations less discriminative. Following the DCCA, we can maximize the agreement of two augmented views with the invariance term and prevent degenerated solutions with the feature-level decorrelation term. In this way, we are not required to select negative pairs arbitrarily, thereby reducing sampling bias. Specifically, we first normalize node embedding matrix $\mathbf{Z}_{\mathcal{V}}$ along instance dimension so that each

feature dimension follows a distribution with 0-mean and $\frac{1}{\sqrt{|\mathcal{V}|}}$-standard deviation:

$$\hat{\mathbf{Z}}_{\mathcal{V}} = \frac{\mathbf{Z}_{\mathcal{V}} - \mu(\mathbf{Z}_{\mathcal{V}})}{\sigma(\mathbf{Z}_{\mathcal{V}}) \times \sqrt{|\mathcal{V}|}} \qquad (6)$$

where $\mu(\cdot), \sigma(\cdot)$ denote computing mean value and standard deviation for each feature dimension, respectively. Next, with normalized matrices $\hat{\mathbf{Z}}_{1,\mathcal{V}}$ and $\hat{\mathbf{Z}}_{2,\mathcal{V}}$ for two augmented views, we construct node-level CCA loss $\mathcal{L}_N$ which consists of an invariance term $\mathcal{L}_N^{inv}$ and a decorrelation term $\mathcal{L}_N^{dec}$ as:

$$\mathcal{L}_N^{inv} = \left\lVert \hat{\mathbf{Z}}_{1,\mathcal{V}} - \hat{\mathbf{Z}}_{2,\mathcal{V}} \right\rVert_F^2 \qquad (7)$$

$$\mathcal{L}_N^{dec} = \left\lVert \hat{\mathbf{Z}}_{1,\mathcal{V}}^T \hat{\mathbf{Z}}_{1,\mathcal{V}} - \mathbf{I} \right\rVert_F^2 + \left\lVert \hat{\mathbf{Z}}_{2,\mathcal{V}}^T \hat{\mathbf{Z}}_{2,\mathcal{V}} - \mathbf{I} \right\rVert_F^2 \qquad (8)$$

$$\mathcal{L}_N = \mathcal{L}_N^{inv} + \lambda_N \mathcal{L}_N^{dec} \qquad (9)$$

where $\mathbf{I} \in \mathbb{R}^{D \times D}$ is the indentity matrix and λ_N represents a non-negative weight. $\lVert \cdot \rVert_F^2$ denote the square of Frobenius norm. $\mathcal{L}_N^{inv}$ preserves the node-wise invariant information while $\mathcal{L}_N^{dec}$ can prevent dimension collapse [Hua *et al.*, 2021], leading to more discriminative representations.

Group-level CCA objective. Group-level SSL signal aims to distinguish the representations of the same hyperedge in the two augmented views from other hyperedge representations, which helps the model preserve group-wise information in the hypergraph. Similar to the node-wise SSL signal, we can formulate the CCA loss in group-wise relation as:

$$\mathcal{L}_G^{inv} = \left\lVert \hat{\mathbf{Z}}_{1,\mathcal{E}} - \hat{\mathbf{Z}}_{2,\mathcal{E}} \right\rVert_F^2 \qquad (10)$$

$$\mathcal{L}_G^{dec} = \left\lVert \hat{\mathbf{Z}}_{1,\mathcal{E}}^T \hat{\mathbf{Z}}_{1,\mathcal{E}} - \mathbf{I} \right\rVert_F^2 + \left\lVert \hat{\mathbf{Z}}_{2,\mathcal{E}}^T \hat{\mathbf{Z}}_{2,\mathcal{E}} - \mathbf{I} \right\rVert_F^2 \qquad (11)$$

$$\mathcal{L}_G = \mathcal{L}_G^{inv} + \lambda_G \mathcal{L}_G^{dec} \qquad (12)$$

where $\hat{\mathbf{Z}}_{1,\mathcal{E}}$ and $\hat{\mathbf{Z}}_{2,\mathcal{E}}$ are normalized hypergraph embedding matrices from two augmented views. λ_G is the non-negative coefficient for the decorrelation term. $\mathcal{L}_G, \mathcal{L}_G^{inv}, \mathcal{L}_G^{dec}$ represent group-wise SSL loss, group-wise invariant term, and group-wise decorrelation term, respectively.

Hierarchical membership-level contrast. Membership-level contrast aims to differentiate a "real" node-hyperedge membership from a "fake" one across the two augmented views initially. In this work, we only sample and contrast membership pairs from the original view as shown in Figure 1. This can avoid bias caused by random augmentation. Following the instance-level discrimination in the previous work, each anchor node v_i and one of its incident hyperedge e_j would form a positive pair and all (v_i, e_k) where $v_i \notin e_k$ are considered as negative pairs with the same status. However, this kind of contrast strategy may result in learning bias. For instance, consider the hypergraph in Figure 2(a) as a co-author network, where $v_1 \in e_1$, but $v_1 \notin e_3$. We can observe a close collaboration relationship (significant overlap) between authors e_1 and e_3, indicating shared research interests. Although paper v_1 has a stronger correlation with incident author e_1, its content may also be similar to works of neighborhood author e_3. This membership-level similarity strength can be easily extended to a broader neighborhood
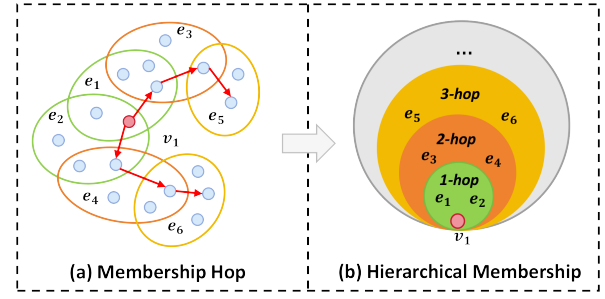

Figure 2: Hierachical membership relation

through cascading overlap, leading to a hierarchical membership structure. Based on this insight, we formally introduce two concepts: membership hop and k-hop membership set.

Definition 1. *(membership hop) Given a hypergraph $\mathcal{H}(\mathcal{V}, \mathcal{E})$, we connect each vertex with other vertices at the same hyperedge to form a clique expansion graph. For each $v_i \in \mathcal{V}$ and $e_j \in \mathcal{E}$, the membership hop is defined as:*

$$\mathcal{M}_{hop}(v_i, e_j) = \max\{hop(v_i, v_k)|v_k \in e_j\}$$

where $hop(v_i, v_k)$ denotes the hop number between v_i and v_k in the expansion graph. In particular, if e_j is not reachable from v_i, $\mathcal{M}_{hop}(v_i, e_j) = \infty$.

Definition 2. *(k-hop membership set) Given a hypergraph $\mathcal{H}(\mathcal{V}, \mathcal{E})$, for each $v_i \in \mathcal{V}$, the v_i's k-hop membership set is:*

$$\mathcal{M}_k(v_i) = \{e_j | \mathcal{M}_{hop}(v_i, e_j) = k\}$$

This hierarchical membership relation has been illustrated in Figure 2. To describe it as a self-supervised signal during training, in k-th membership hop, we consider each $\mathcal{M}_k(v_i)$ as the positive set $\mathcal{P}_k$ while $\mathcal{M}_{k+1}(v_i)$ as the negative set $\mathcal{N}_k$. Inspired by MIL-NCE [Miech *et al.*, 2020] which can handle multiple positive samples, we design a loss function in k-th membership hop for v_i as:

$$\mathcal{L}_k(v_i) = -\log \frac{\sum_{e_p \in \mathcal{P}_k} e^{\mathcal{D}(v_i, e_p)/\tau}}{\sum_{e_p \in \mathcal{P}_k} e^{\mathcal{D}(v_i, e_p)/\tau} + \sum_{e_q \in \mathcal{N}_k} e^{\mathcal{D}(v_i, e_q)/\tau}} \qquad (13)$$

where $\mathcal{D}(v, e)$ represents a trainable discriminator that predict the similarity score between v and e. we implement it using a bilinear function, as described in [Lee and Shin, 2023]. τ denotes the temperature parameter. In this way, our objective can be interpreted as hyperedges in $\mathcal{M}_k(v_i)$ is overall more similar with v_i than those in $\mathcal{M}_{k+1}(v_i)$. This multi-positive setting can enhance model robustness to noisy positive samples. Then, it is natural to enumerate this objective in different membership ranges, formulating our final hierarchical membership loss $\mathcal{L}_{HM}$ as:

$$\mathcal{L}_{HM} = -\sum_{v_i \in \mathcal{V}} \frac{1}{K} \sum_{k=1}^{K} \log \left[\min\{e^{-\mathcal{L}_k(v_i)}, \alpha\} \right] \qquad (14)$$

where $\alpha \in (0, 1)$ is the secure threshold to avoid over-pushing two neighboring membership hops. K is a hyperparameter that denotes the chosen membership range. However,

using all the samples in different membership sets is computationally prohibitive for large hypergraphs. To deal with this, we uniformly sample a fixed number of d hyperedges in each $\mathcal{M}_k(v_i)$ in implementation. The experiments in Section 4 show that only a small sample number can make SE-HSSL achieve superior performance than the state-of-the-art.

We consider the frequency of calculating similarity function $D(v,e)$ when evaluating the time complexity in membership contrast. The current state-of-the-art method TriCL requires a complexity of up to $O(|\mathcal{V}| \times |\mathcal{E}|)$, whereas the time complexity of SE-HSSL is $O(|\mathcal{V}|Kd)$.

3.5 Model Optimization

To preserve multi-view structural information during self-supervised hypergraph representation learning, we combine tri-directional signals and jointly optimize the node-level CCA loss $\mathcal{L}_N$, group-level CCA loss $\mathcal{L}_G$, and hierarchical membership-level contrast loss $\mathcal{L}_{HM}$ as:

$$\mathcal{L} = \mathcal{L}_N + \lambda_1 \mathcal{L}_G + \lambda_2 \mathcal{L}_{HM} + \lambda_3 ||\boldsymbol{\Theta}||^2 \qquad (15)$$

where $\lambda_1, \lambda_2, \lambda_3$ are hyperparameters to control the strengths of group-level loss, membership-level loss, and L_2 regularization respectively. $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_{\mathcal{E}}, \boldsymbol{\Theta}_{\mathcal{V}}\}$ represents the learnable parameters in hypergraph encoder.

4 Experiment

In this section, We first briefly introduce the experimental setups. Subsequently, we conduct extensive experiments to validate the effectiveness and efficiency of our model.

4.1 Experimental Setups

Datasets. We empirically evaluate the model performance on 7 commonly used hypergraph benchmark datasets, whose details are shown in Table 1.

Baselines. To comprehensively validate our method, we compare it against 11 strong baselines, including 4 semi-supervised models (i.e., HGNN [Feng *et al.*, 2019], HyperGCN [Yadati *et al.*, 2019], UniGIN [Cai *et al.*, 2022]), and AllSet [Chien *et al.*, 2021], 5 common graph based self-supervised methods (i.e., DGI [Veličković *et al.*, 2018], GRACE [Zhu *et al.*, 2020], BGRL [Thakoor *et al.*, 2021], CCA-SSG [Zhang *et al.*, 2021a], and COSTA [Zhang *et al.*, 2022]), and 2 hypergraph contrastive learning methods (i.e., HyperGCL [Wei *et al.*, 2022] and the current SOTA method TriCL [Lee and Shin, 2023]). For the evaluation of graph-based SSL methods, we follow [Lee and Shin, 2023] by applying clique expansion to hypergraphs, thereby converting them into simple graphs.

Evaluation protocol. In this work, we focus on two commonly used benchmark hypergraph learning tasks: node classification and node clustering [Zhou *et al.*, 2006; Lee and Shin, 2023]. For the node classification task, we follow the linear evaluation scheme utilized in previous work [Wei *et al.*, 2022; Lee and Shin, 2023]. Specifically, we first train the hypergraph encoder in a self-supervised manner as described in Section 3. Afterward, with the trained model, we generate node embeddings and randomly split them into training, validation, and test samples using splitting percentages of 10%,

Name	Type	$|\mathcal{V}|$	$|\mathcal{E}|$	F	Classes
Cora [Wei *et al.*, 2022]	citation	2708	1579	1433	7
Citeseer [Wei *et al.*, 2022]	citation	3312	1079	3703	6
Pubmed [Wei *et al.*, 2022]	citation	19717	7963	500	3
Cora-CA [Yadati *et al.*, 2019]	co-author	2708	1072	1433	7
NTU2012 [Yang *et al.*, 2022]	graphics	2012	2012	100	67
ModelNet40 [Yang *et al.*, 2022]	graphics	12311	12311	100	40
Zoo [Hein *et al.*, 2013]	animal	101	43	16	7

Table 1: Statistics of datasets

10%, and 80%, respectively. Finally, we employ the obtained embeddings from the training set to train a logistic regression classifier and evaluate its performance on test node embeddings. For fair evaluation, we report the average test accuracy along with its corresponding standard deviation based on 20 random initializations for each dataset.

For the node clustering task [Lee and Shin, 2023], we apply the k-means clustering algorithm to the output node representations and get the predicted results. As in [Zhao *et al.*, 2021], the normalized mutual information (NMI), and adjusted rand index (ARI) are used to measure the performance of clustering. We randomly perform k-means on generated node embeddings 5 times and report the averaged results.

Implementation details. For all baselines, we report their performance based on the official implementations and default hyper-parameters from original papers. For SE-HSSL, we set the number of encoder layers to 1 for all datasets. We set the group-level coefficient $\lambda_1 = 1$ and regularization coefficient $\lambda_3 = 0.05$. The sampling number d in each membership set is fixed to 10. The membership-level coefficient λ_2 is tuned over $\{0.1, 0.18, 0.20\}$. Besides, we tune the embedding size D in $\{256, 512, 784, 1024\}$, the hyperedge hop K in $\{1, 2, 3, 4\}$, the temperature τ in $\{0.4, 0.5, 0.6\}$, and threshold α in $\{0.62, 0.65\}$. We have released our code in https://github.com/Coco-Hut/SE-HSSL.

4.2 Node Classification Evaluation

We first employ the node classification task to evaluate the effectiveness of SE-HSSL. The performance of all compared methods is summarized in Table 2. The first 4 lines present the results of supervised HGNNs. It can be observed that the AllSet network demonstrates superior performance compared to other methods. This is because its learnable multiset functions allow for higher expressive power. However, we notice that these methods perform poorly compared to the HSSL models. This is because, when labels are limited, supervised models are prone to overfitting and fail to generalize well to unseen data. For SSL frameworks, they can capture the underlying semantics of data without labels. Lines 5-9 show the results of several popular graph SSL methods. Although they achieve superior overall performance compared to supervised methods, their accuracy scores are considerably lower than those of HSSL methods. This is because, during the expansion of a hypergraph to a simple graph, a significant amount of high-order information is lost. The above observations suggest that it is necessary to devise specific SSL models for hypergraph learning. The last 3 lines compare

Method	Cora	Citeseer	Pubmed	Cora-CA	NTU2012	Zoo	ModelNet40	Avg.Rank
HGNN	73.36±3.40	64.66±2.95	79.22±1.59	73.26±2.86	70.53±7.68	77.56±11.34	91.14±0.35	9.3
HyperGCN	74.32±2.39	65.28±2.85	77.90±1.39	72.17±1.92	71.68±2.40	78.37±10.90	90.81±0.30	8.7
UniGIN	71.81±2.36	63.58±3.07	77.71±3.39	71.20±2.50	68.83±3.60	74.51±10.87	90.65±0.71	11.0
AllSet	77.12±0.61	67.88±1.36	81.55±0.29	76.89±1.63	73.70±0.50	78.19±10.39	95.92±0.24	5.4
DGI	76.54±1.48	66.78±1.56	77.68±2.24	74.56±1.44	70.46±2.45	63.11±13.14	91.48±0.57	9.7
GRACE	76.90±2.25	66.45±2.57	80.64±0.46	75.02±2.06	69.77±2.38	62.16±15.45	89.96±0.39	9.7
BGRL	78.01±1.53	67.11±1.78	81.38±0.61	77.97±2.84	71.49±1.78	65.50±12.36	92.28±2.52	6.9
CCA-SSG	79.29±1.51	67.48±1.77	82.14±0.64	78.45±1.16	72.81±2.43	78.35±10.85	93.91±0.18	4.3
COSTA	77.49±1.83	68.74±1.61	81.89±0.56	78.03±2.07	72.14±2.18	69.69±9.46	91.64±2.18	5.7
HyperGCL	78.86±1.09	69.49±1.62	83.95±0.40	77.90±1.15	**75.44±1.15**	67.74±10.34	96.42±0.31	4.1
TriCL	81.61±1.24	72.05±1.14	84.24±0.63	82.17±0.92	75.21±2.46	80.09±11.14	97.05±0.13	<u>2.0</u>
SE-HSSL	**82.47±1.14**	**72.74±0.98**	**84.38±0.71**	**82.59±0.93**	75.19±2.36	**80.68±11.13**	**97.17±0.13**	**1.3**

Table 2: Evaluation results for node classification: mean accuracy (%) ± standard deviation. For each dataset, the top and runner-up performances are indicated by boldface and underlined formatting, respectively.

the results of SE-HSSL with the latest HCL models. The proposed SE-HSSL achieves the best performance on most datasets. For example, we achieve mean accuracy scores of 82.47% and 72.74% on the Cora and Citeseer datasets, respectively. These results show a relative improvement of 0.86% and 0.69% over the current SOAT method TriCL. It is noteworthy that SE-HSSL outperforms TriCL and Hyper-GCL even without the utilization of node-wise and group-wise negative samples. Furthermore, it only requires a small number of membership pairs. These highlight the effectiveness of our tri-directional sampling-efficient signals.

4.3 Node Clustering Evaluation

In the subsection, we conduct the node clustering task on 7 datasets. We compare SE-HSSL with 2 HCL methods and 3 strong graph SSL models. Table 3 shows that SE-HSSL achieves the best clustering results on 6 out of 7 datasets and obtains 1st place in terms of the average rank. Moreover, we find it achieves an average improvement of 1.80% in NMI and 2.78% in ARI compared to TriCL across all datasets. This is because our sampling-efficient signals effectively reduce training bias and avoid degenerated solutions, making learned embeddings more informative and discriminative.

4.4 Ablation Study

In this subsection, we evaluate the effectiveness of each designed SSL objective. The results are summarized in Table 4. We find that the performance degrades when any of the objectives is dropped, which indicates the effectiveness of each signal. We also observe that SE-HSSL-w/o N achieves lower accuracy compared to the other two variants on most datasets. This is because we focus on the node-level downstream task. Notably, the removal of hierarchical membership loss brings about even larger performance degradation than SE-HSSL-w/o N on Citeseer and Zoo datasets, which shows that this signal is essential in boosting model performance.

4.5 Parameter Sensitivity Analysis

We perform sensitivity analysis on two critical parameters to further validate the robustness of SE-HSSL. We report mean accuracy w.r.t. the node classification task in Figure 3.
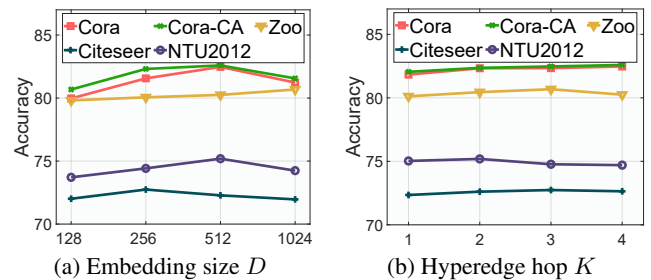

(a) Embedding size D (b) Hyperedge hop K

Figure 3: Parameter sensitivity test.

Embedding size D. Figure 3(a) illustrates that as we vary the embedding size from 128 to 512, the model performance improves overall, suggesting that our model benefits from a larger representation capacity. However, we notice a decline in performance as the embedding size further increases on most datasets. This is because the embedding space becomes sparse when D is too large. Consequently, there is limited meaningful information available in each dimension of the embedding, with many dimensions being irrelevant or redundant. This makes it challenging for the downstream classifier to extract meaningful features from embeddings.

Hyperedge hop K. In the hierarchical membership contrast, we sample neighborhood hyperedges ranging from 1 to K hops for each node. As depicted in Figure 3(b), we observe that the model can achieve near-optimal performance when K is set to 1. Increasing the neighborhood range only leads to marginal performance gains on most datasets. This suggests that our model exhibits robustness and is not highly sensitive to the number of hops when sampling hyperedges. We attribute this to our multi-positive membership loss which does not enforce a higher similarity for all positive pairs and can effectively disregard the noisy positives.

4.6 Efficiency Evaluation

We measure the training time to validate the efficiency of the proposed SE-HSSL. We mainly compare it with TriCL,

Method	Cora		Citeseer		Pubmed		Cora-CA		NTU2012		Zoo		ModelNet40		Avg.Rank
	NMI↑	ARI↑	NMI↑	ARI↑	NMI↑	ARI↑	NMI↑	ARI↑	NMI↑	ARI↑	NMI↑	ARI↑	NMI↑	ARI↑	
BGRL	40.81	22.54	31.69	24.87	15.98	16.80	32.48	21.72	67.76	33.52	70.55	58.65	73.63	45.36	5.9
CCA-SSG	49.08	40.57	38.45	36.02	17.22	17.24	38.13	26.18	73.72	42.26	79.59	74.70	79.08	56.44	3.9
COSTA	45.58	36.08	34.83	28.18	20.41	15.92	36.09	22.90	71.77	40.32	76.25	61.95	75.23	46.33	4.9
HyperGCL	40.24	32.35	39.06	38.88	28.72	24.73	46.15	39.33	84.27	70.88	77.40	79.36	93.71	89.91	2.9
TriCL	56.95	49.77	42.63	42.44	30.80	29.06	44.39	37.21	83.75	68.33	90.87	88.62	93.80	87.07	2.2
SE-HSSL	59.41	53.37	43.66	43.01	32.79	29.73	50.56	44.47	83.31	68.26	91.71	91.48	94.36	91.63	1.3

Table 3: Evaluation results for node clustering: NMI and ARI

Method	Cora	Citeseer	Pubmed	Cora-CA	NTU2012	Zoo	ModelNet40	Avg.Rank
SE-HSSL-w/o N	80.21±1.51	72.16±1.14	83.09±0.71	80.04±2.34	74.22±2.68	80.31±10.94	97.13±0.10	3.4
SE-HSSL-w/o G	81.23±0.85	72.28±0.95	83.84±0.74	82.27±0.93	75.08±2.21	79.81±10.88	97.12±0.11	2.7
SE-HSSL-w/o HM	81.71±1.41	71.67±1.21	83.90±0.76	82.15±1.02	74.81±2.25	79.51±10.89	97.15±0.14	2.9
SE-HSSL	82.47±1.14	72.74±0.98	84.38±0.71	82.59±0.93	75.19±2.36	80.68±11.13	97.17±0.13	1.0

Table 4: Ablation study on different SSL signals. The variants SE-HSSL-w/o N, SE-HSSL-w/o G, and SE-HSSL-w/o HM represent models that have removed node-level CCA loss, group-level CCA loss, and hierarchical membership loss, respectively.
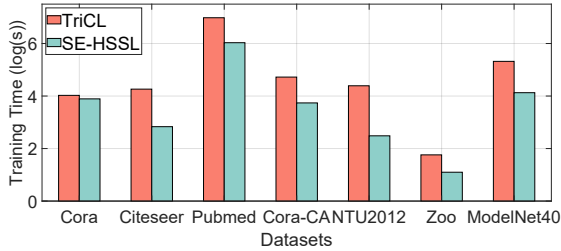
Figure 4: The training time comparison between SE-HSSL and TriCL. log(·) represents the natural logarithm.

which is the current SOTA approach with tri-directional contrast objectives. The results are shown in Figure 4. For better visibility, we take the logarithm of training time when plotting. It can be found that SE-HSSL is consistently faster than TriCL across all datasets. What stands out is that our model is at least 2.5 times faster than TriCL on large datasets, such as PubMed and ModelNet40. This efficiency improvement can be attributed to three key factors: 1) Sampling-free CCA objectives in node-level and group-level learning reduce computation burden during training. 2) Hierarchical contrast in membership-level learning only requires a small number of node-hyperedge pairs in each neighborhood range, thereby considerably enhancing the training speed. 3) Our method is projection head-free, further reducing the computation cost.

5 Related Work

5.1 Contrastive Learning on Graph

Contrastive learning (CL), which was originally utilized in computer vision [He *et al.*, 2020], has recently been applied to the graph domain. The basic idea of graph contrastive learning (GCL) is to maximize agreement between instances (e.g., node, subgraph, and graph) of different views augmented from the original graph. DGI [Veličković *et al.*, 2018] maximizes the mutual information between the local and global representations of nodes to learn their embeddings. GraphCL [You *et al.*, 2020] and GCA [Zhu *et al.*, 2021] em-

ploy multiple augmentation strategies and treat nodes/graphs with distinct IDs from the anchor node/graph as negative samples. COSTA [Zhang *et al.*, 2022] and SFA [Zhang *et al.*, 2023] are two feature-level augmentation strategies that aim to construct more effective contrast views than random augmentations. However, the above methods can not effectively learn high-order information in hypergraphs, so it is necessary to design specific HSSL approaches.

5.2 Contrastive Learning on Hypergraph

Hypergraph contrastive learning has received little attention and remains largely unexplored. While some investigations in recommender systems have applied contrastive strategy in hypergraphs [Yang *et al.*, 2021; Xia *et al.*, 2022a; Liang *et al.*, 2023], they are not designed for general HCL. HyperGCL [Wei *et al.*, 2022] is the first general HCL framework. It designs a variational auto-encoder to generate augmentation views that can preserve the high-order information in the original hypergraph. TriCL [Lee and Shin, 2023] uses tri-directional contrastive objectives to capture group-level and membership-level information in hypergraphs. However, existing methods select a large number of negative samples arbitrarily, which can be both unreliable and inefficient.

6 Conclusion

In this paper, we introduce SE-HSSL, a sampling-efficient hypergraph self-supervised learning framework, to address the efficiency bottleneck and training bias in instance-level contrastive methods. To achieve this, we propose sampling-free CCA-based objectives as self-supervised signals for learning at both the node and group levels. This approach allows us to effectively avoid degenerate solutions in the learned embeddings and reduce sampling bias. Additionally, we develop a novel hierarchical membership contrast objective, which only requires a small number of node-hyperedge pairs to achieve effective membership-level self-supervised training. Through extensive experiments conducted on 7 real-world datasets, we empirically demonstrate the superior effectiveness and efficiency of SE-HSSL.

Acknowledgments

References

[Andrew *et al.*, 2013] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *International conference on machine learning*, pages 1247–1255. PMLR, 2013.

[Antelmi *et al.*, 2023] Alessia Antelmi, Gennaro Cordasco, Mirko Polato, Vittorio Scarano, Carmine Spagnuolo, and Dingqi Yang. A survey on hypergraph representation learning. *ACM Computing Surveys*, 56(1):1–38, 2023.

[Cai *et al.*, 2022] Derun Cai, Moxian Song, Chenxi Sun, Baofeng Zhang, Shenda Hong, and Hongyan Li. Hypergraph structure learning for hypergraph neural networks. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 1923–1929, 2022.

[Chang *et al.*, 2018] Xiaobin Chang, Tao Xiang, and Timothy M Hospedales. Scalable and effective deep cca via soft decorrelation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1488–1497, 2018.

[Chien *et al.*, 2021] Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. You are allset: A multiset function framework for hypergraph neural networks. In *International Conference on Learning Representations*, 2021.

[Feng *et al.*, 2019] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3558–3565, 2019.

[He *et al.*, 2020] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[Hein *et al.*, 2013] Matthias Hein, Simon Setzer, Leonardo Jost, and Syama Sundar Rangapuram. The total variation on hypergraphs-learning on hypergraphs revisited. *Advances in Neural Information Processing Systems*, 26, 2013.

[Hua *et al.*, 2021] Tianyu Hua, Wenxiao Wang, Zihui Xue, Sucheng Ren, Yue Wang, and Hang Zhao. On feature decorrelation in self-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9598–9608, 2021.

[Jia *et al.*, 2021] Renqi Jia, Xiaofei Zhou, Linhua Dong, and Shirui Pan. Hypergraph convolutional network for group recommendation. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 260–269. IEEE, 2021.

[Jing *et al.*, 2014] Xiao-Yuan Jing, Rui-Min Hu, Yang-Ping Zhu, Shan-Shan Wu, Chao Liang, and Jing-Yu Yang. Intraview and inter-view supervised correlation analysis for multi-view feature learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

[Kim *et al.*, 2020] Eun-Sol Kim, Woo Young Kang, Kyoung-Woon On, Yu-Jung Heo, and Byoung-Tak Zhang. Hypergraph attention networks for multimodal learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14581–14590, 2020.

[Lee and Shin, 2023] Dongjin Lee and Kijung Shin. I'm me, we're us, and i'm us: Tri-directional contrastive learning on hypergraphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8456–8464, 2023.

[Liang *et al.*, 2023] Yongquan Liang, Qiuyu Song, Zhongying Zhao, Hui Zhou, and Maoguo Gong. Bagnn: Behavior-aware graph neural network for session-based recommendation. *Frontiers of Computer Science*, 17(6):176613, 2023.

[Miech *et al.*, 2020] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020.

[Thakoor *et al.*, 2021] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. Bootstrapped representation learning on graphs. In *ICLR 2021 Workshop on Geometrical and Topological Representation Learning*, 2021.

[Veličković *et al.*, 2018] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *International Conference on Learning Representations*, 2018.

[Wei *et al.*, 2022] Tianxin Wei, Yuning You, Tianlong Chen, Yang Shen, Jingrui He, and Zhangyang Wang. Augmentations in hypergraph contrastive learning: Fabricated and generative. *Advances in neural information processing systems*, 35:1909–1922, 2022.

[Xia *et al.*, 2022a] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*, pages 70–79, 2022.

[Xia *et al.*, 2022b] Lianghao Xia, Chao Huang, and Chuxu Zhang. Self-supervised hypergraph transformer for recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2100–2109, 2022.

[Xiao *et al.*, 2019] Li Xiao, Junqi Wang, Peyman H Kassani, Yipu Zhang, Yuntong Bai, Julia M Stephen, Tony W Wilson, Vince D Calhoun, and Yu-Ping Wang. Multi-hypergraph learning-based brain functional connectivity analysis in fmri data. *IEEE transactions on medical imaging*, 39(5):1746–1758, 2019.

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

[Yang *et al.*, 2021] Haoran Yang, Hongxu Chen, Lin Li, S Yu Philip, and Guandong Xu. Hyper meta-path contrastive learning for multi-behavior recommendation. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 787–796. IEEE, 2021.

[Yang *et al.*, 2022] Chaoqi Yang, Ruijie Wang, Shuochao Yao, and Tarek Abdelzaher. Semi-supervised hypergraph node classification on hypergraph line expansion. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2352–2361, 2022.

[Yi and Park, 2020] Jaehyuk Yi and Jinkyoo Park. Hypergraph convolutional recurrent neural network. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3366–3376, 2020.

[You *et al.*, 2020] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33:5812–5823, 2020.

[Yu *et al.*, 2012] Jun Yu, Dacheng Tao, and Meng Wang. Adaptive hypergraph learning and its application in image classification. *IEEE Transactions on Image Processing*, 21(7):3262–3272, 2012.

[Zhang *et al.*, 2021a] Hengrui Zhang, Qitian Wu, Junchi Yan, David Wipf, and Philip S Yu. From canonical correlation analysis to self-supervised graph neural networks. *Advances in Neural Information Processing Systems*, 34:76–89, 2021.

[Zhang *et al.*, 2021b] Junwei Zhang, Min Gao, Junliang Yu, Lei Guo, Jundong Li, and Hongzhi Yin. Double-scale self-supervised hypergraph learning for group recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 2557–2567, 2021.

[Zhang *et al.*, 2022] Yifei Zhang, Hao Zhu, Zixing Song, Piotr Koniusz, and Irwin King. Costa: covariance-preserving feature augmentation for graph contrastive learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2524–2534, 2022.

[Zhang *et al.*, 2023] Yifei Zhang, Hao Zhu, Zixing Song, Piotr Koniusz, and Irwin King. Spectral feature augmentation for graph contrastive learning and beyond. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11289–11297, 2023.

[Zhao *et al.*, 2021] Han Zhao, Xu Yang, Zhenru Wang, Erkun Yang, and Cheng Deng. Graph debiased contrastive learning with joint representation clustering. In *IJCAI*, pages 3434–3440, 2021.

[Zhou *et al.*, 2006] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. *Advances in neural information processing systems*, 19, 2006.

[Zhu *et al.*, 2020] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*, 2020.

[Zhu *et al.*, 2021] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*, pages 2069–2080, 2021.