Multi-view Mixed Attention for Contrastive Learning on Hypergraphs

Jongsoo Lee
Hanyang University
Seoul, South Korea
leejongsoo@hanyang.ac.kr

Dong-Kyu Chae
Hanyang University
Seoul, South Korea
dongkyu@hanyang.ac.kr

ABSTRACT

Hypergraphs are effective in learning high-order relationships between nodes, which naturally represent *group interactions* as hyperedges (i.e., arbitrary-sized subsets of nodes). However, most approaches currently used for learning hypergraph representations do not consider *pairwise* relationships between nodes. While high-order relationships provide insight into the general connections among nodes in a group, they do not reveal the pairwise relationships between individual nodes within that group. Considering that it is unlikely for all nodes in the same group to share identical relationships, we argue that considering pairwise relationships is a critical aspect. In this paper, we propose **M**ulti-view **M**ixed **A**ttention for **C**ontrastive **L**earning (**MMACL**) to address the aforementioned problem. MMACL proposes *Mixed-Attention*, which blends high-order relationships derived from the hypergraph attention network and pairwise relationships derived from the graph attention network. Then, it performs node-level contrastive learning to the graph structure with different views learned at each layer to finally obtain an expressive node representation. Our extensive experimental results on several popular datasets validate the effectiveness of the proposed MMACL for hypergraph node classification. Our code is available at: https://github.com/JongsooLee-HYU/MMACL

CCS CONCEPTS

• **Mathematics of computing → Hypergraphs**; • **Computing methodologies → Neural networks**.

KEYWORDS

Hypergraphs, Contrastive learning, Graph representation learning

ACM Reference Format:
Jongsoo Lee and Dong-Kyu Chae. 2024. Multi-view Mixed Attention for Contrastive Learning on Hypergraphs. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3626772.3657897

1 INTRODUCTION

Hypergraphs are a generalized form of graph structures that extend the concept of an edge from connecting only two vertices to connecting *any number of vertices* via *hyperedges*, naturally representing group interactions [2]. They provide a powerful tool for representing higher-order, many-to-many relationships between entities where standard graphs may fall short. Because of such unique structure, hypergraphs have widespread applications in several real-world scenarios, each of which benefits from their ability to capture complex relationships [7, 13, 23, 25]. On the other hand, *Graph neural networks* (GNNs) have gained significant attention due to its efficacy of learning pairwise relationships between entities. The high-quality representations of general graphs learned through GNNs have been successfully applied across various domains [6, 10, 14, 15, 19, 22]. However, learning representations for hypergraphs [7] remains a significant challenge.

Recently, *contrastive learning* [9] has gained attention for hypergraph representation learning [13]. Its main idea, contrasting semantically similar (positive) data pairs with dissimilar (negative) ones to learn representations that bring positive pairs closer and push negative pairs apart, has shown high performance in fields like Computer Vision [17] and Natural Language Processing [3]. Applying contrastive learning to hypergraphs has been quite successful for learning hypergraph representations [23]. However, we believe there is still room for further advancement due to the following point. Previous studies have augmented hypergraphs with node feature masking and hyperedge masking during contrastive learning, but both augmented views focus only on high-order relationships. While high-order relationships can represent the general relations among nodes within a group, they fail to consider the pairwise relationships between individual nodes. In real-world scenarios, nodes within the same group do not always share identical relationships. Therefore, it is crucial to consider pairwise relationships between nodes, but existing studies have focused solely on high-order relationships for hypergraph representation learning.

In this paper, we introduce a novel framework called **M**ulti-view **M**ixed **A**ttention for **C**ontrastive **L**earning (**MMACL**) on Hypergraphs. It considers both the high-order relationships of hypergraphs and the pairwise relationships of general graphs to improve the quality of node representations. Towards this end, MMACL adopts two types of attention layers: one for the *hypergraph structure* and the other for the *general graph structure*. MMACL inputs a hypergraph into the hypergraph attention layer to derive high-order relationships. At the same time, by applying *clique expansion* [20] to the given hypergraph, MMACL creates its general graph structure, which is then fed into the graph attention layer to derive pairwise relationships. During the node feature aggregation process in each attention layer, one layer incorporates attention scores derived from the other layer, blending the two different types of relationships. Having the two different views (high-order

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657897

and pairwise relationships) induced by the same hypergraph, our contrastive learning adopts the same node in the two views as a positive pair, and all other node pairs as negative pairs. As a result, MMACL achieves expressive node representations by considering both high-order and pairwise relationships. Empirically, MMACL achieved the highest accuracy compared to state-of-the-art hypergraph learning methods, and our ablation study highlighted the effectiveness of the proposed multi-view mixed attention.

2 RELATED WORK AND MOTIVATION

Hypergraph Learing. Hypergraphs, with their ability to model intricate relationships, have been actively used to illustrate the complex associations inherent in high-dimensional data. Hypergraph Neural Network (HGNN) [7] is the first work that applies GNNs for learning hypergraphs, involving the convolution operation with a hypergraph Laplacian and truncated Chebyshev polynomials. UniGNN [11] proposes a unified framework that generalizes the message-passing processes for both graph and hypergraph neural networks. HCNH [24] conducts convolution operations on both nodes and hyperedges, which are then used to reconstruct the hypergraph. HCoN [25] collaboratively aggregates information from previous nodes and hyperedges.

Hypergraph Contrastive Learing. More recently, many studies have utilized contrastive learning, one of the self-supervised learning methods, for hypergraph representation learning. TriCL [13] introduced a tri-directional contrastive learning that considers node-, group-, and membership-level contrast. CCL [23] proposed a contrastive learning method working on the outputs of Graph Convolutional Network (GCN) and Hypergraph Convolutional Network (HGCN) to consider information obtained from both general graph and hypergraph structures.

Our Motivation. Most related work has not concurrently considered the pairwise relationships between nodes and high-order relationships. While CCL [23] aimed to incorporate information derivable from the general graph structure, it derives neighborhood node information through convolution layers, leading to predefined or equally considered importance across all neighborhood nodes. Thus, it fails to capture the nuanced differences in high-order and pairwise relationships between nodes.

3 METHOD

This section presents the proposed MMACL (Multi-view Mixed Attention for Contrastive Learning). Figure 1 illustrates an overview of MMACL. It blends (1) the pairwise relationships between nodes derived from the general graph structure and (2) the high-order relationships from the hypergraph structure, during the node feature aggregation process. It then conducts node-level contrastive learning on the two views focusing on distinct relationships.

3.1 Preliminaries

A general graph $\mathcal{G}$ can be presented as a $\mathcal{G} = (V, E)$, where $V = \{v_1, v_2, ..., v_{|V|}\}$ denotes a set of nodes and $V \in \mathbb{R}^{|V| \times m}$ where m is dimension of feature vectors. $E \in \mathbb{R}^{|V| \times |V|}$ indicates a set of edges. $\mathcal{N}_i = \{j | (i, j) \in E\}$ stands for a set of neighboring nodes of v_i. Such a general graph structure can effectively reflect pairwise relationships between two nodes, but it faces a challenge to capture various high-order relationships that are prevalent in real-world.

Unlike the general graph structure, in a *hypergraph*, a single *hyperedge* can connect multiple vertices. A hypergraph $\mathcal{H}$ can be represented as $\mathcal{H} = (V, \mathcal{E})$, where $\mathcal{E} = \{e_1, e_2, ..., e_{|\mathcal{E}|}\}(\mathcal{E} \in \mathbb{R}^{|V| \times |\mathcal{E}|})$ indicates a set of hyperedges and each hyperedge is a non-empty subset of nodes. And the set of all nodes connected to the hyperedge e_i is defined as $\mathcal{Y}_i$, and the set of hyperedges connected to node n_i is defined as $\mathcal{P}_i$.

3.2 Hypergraph Attention Layer

The goal of this layer is to learn the high-order relationships between nodes. Our hypergraph attention layer takes the initial node features of the hypergraph as input. It then updates the representation of hyperedges, and aggregates information from updated hyperedges to update node representations. In this hypergraph representation learning process, certain nodes may be informative while others may not. Hyperedges can also have varying degrees of importance. To account for this, we design the following two types of attention mechanisms to ensure that important nodes and hyperedges have a relatively higher influence during aggregation.

Hyperedge Attention. During the representation learning of a hyperedge, we aggregate the information from all nodes connected to each hyperedge. Here, the hyperedge attention mechanism [4] captures the different importances of all nodes $n_p \in \mathcal{Y}_j$ connected to a given hyperedge e_j. The attention score A_{ji} that node n_i has for hyperedge e_j can be calculated as follows:

$$A_{ji} = \frac{S\left(\mathbf{W}_1 \mathbf{n}_i, \mathbf{u}\right)}{\sum_{p \in \mathcal{Y}_j} S\left(\mathbf{W}_1 \mathbf{n}_p, \mathbf{u}\right)} \quad (1) \qquad S(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}^T \mathbf{b}}{\sqrt{D}} \quad (2)$$

where $\mathbf{W}_1$ is a learnable parameter matrix and $\mathbf{u}$ is a trainable weight vector. As a similarity function $S(\cdot, \cdot)$, we adopt the Scaled Dot-Product Attention mechanism.

Node Attention. After then, node representations can be obtained by aggregating the updated hyperedge representations. Similarly, the node attention here determines the different importance of different hyperedges. Formally, the attention score B_{ij} of hyperedge e_j for node n_i is computed by:

$$B_{ij} = \frac{S\left(\mathbf{W}_2 e_j, \mathbf{W}_3 n_i\right)}{\sum_{p \in \mathcal{P}_i} S\left(\mathbf{W}_2 e_p, \mathbf{W}_3 n_i\right)} \quad (3)$$

where $\mathbf{W}_2$ and $\mathbf{W}_3$ are the learnable parameter matrices. We also utilize the Scaled Dot-product attention as a similarity function.

3.3 Graph Attention Layer

The goal of this layer is to learn the pairwise relationships between nodes in a hypergraph. To achieve this, the given hypergraph needs to be converted into a general graph structure. Here, we apply *clique expansion* [20], where every hyperedge is expanded into a *clique*, meaning all nodes connected by the same hyperedge become fully connected. Then, the converted graph structure is fed into the graph attention layer [21], where the attention score G_{ij} that node i has for node j can be calculated by :

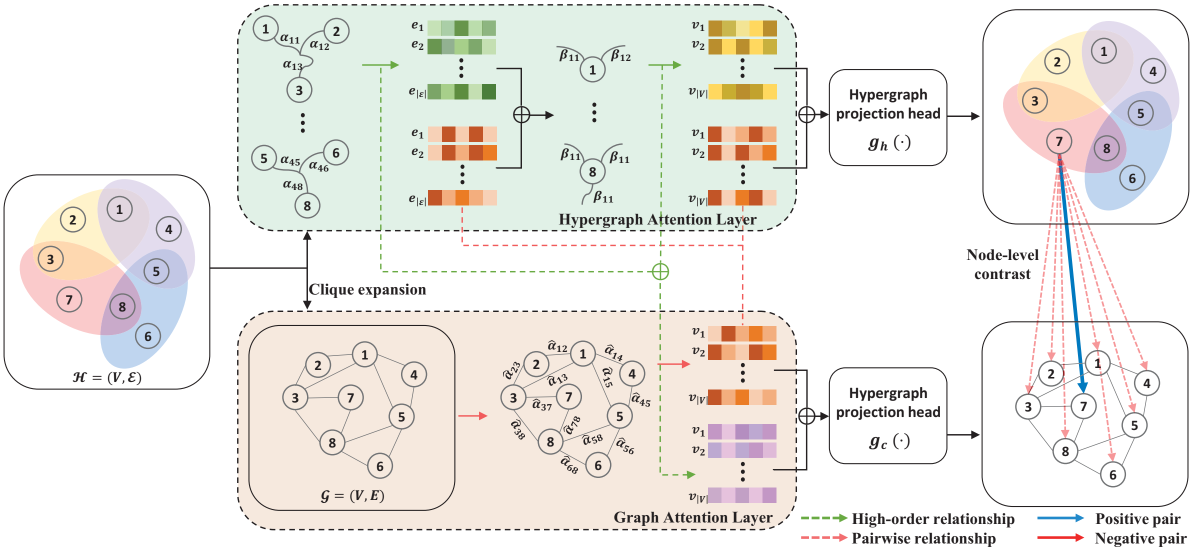

Figure 1: The overview of our MMACL framework. The given hypergraph is fed into the hypergraph attention layer, and its transformation to the general graph structure is given to the graph attention layer. The attention scores derived from one layer are passed to the other attention layer (the green and orange dashed arrows). The two layers finally produce the different views, induced from the same hypergraph. The same node in the two views are considered as the positive pair for contrastive learning.

$$G_{ij} = \frac{exp(LeakyReLU(\mathbf{W}_4 \cdot [\mathbf{n}_i \parallel \mathbf{n}_j]))}{\sum_{k \in \mathcal{N}(i)} exp(LeakyReLU(\mathbf{W}_4 \cdot [\mathbf{n}_i \parallel \mathbf{n}_k]))} \tag{4}$$

where $\mathbf{W}_4$ is a learnable parameter matrix and $[\,\cdot \parallel \cdot\,]$ indicates the concatenation operation.

3.4 Mixed Attention

Our hypergraph attention and general graph attention consider high-order relationships and pairwise relationships, respectively. The next step involves each attention layer considering both types of relationships for representation learning. To achieve this, we propose a Mixed attention schema. Specifically, the graph attention layer, which focuses on learning pairwise relationships, mixes the high-order attention scores derived from the hypergraph attention layer during the node feature aggregation process, and vice versa.

Having the pairwise attention score matrix $G \in \mathbb{R}^{|V| \times |V|}$ obtained from the graph attention layer and the high-order attention matrices $A \in \mathbb{R}^{|\mathcal{E}| \times |V|}$ and $B \in \mathbb{R}^{|V| \times |\mathcal{E}|}$ derived from the hypergraph attention layer, the mixed attention score $\hat{G}$ after adopting the mixed attention schema in the graph attention layer is:

$$\hat{G} = G \oplus (BB^T \oplus A^T A) \tag{5}$$

where $\oplus$ indicates element-wise summation.

Next, we reflect the attention scores derived from the graph attention layer to the hypergraph attention. We claim that by aggregating the pairwise relationships among nodes within a group, we can capture the general relationship within the group. Therefore, the high-order relationship attention matrix $M \in \mathbb{R}^{|V| \times |\mathcal{E}|}$ derived from pairwise relationships, and the resulting high-order attention scores $\hat{A}$ and $\hat{B}$ are calculated as follows:

$$M_{ij} = \frac{\sum_{k \in \mathcal{Y}(j) \setminus \{i\}} G_{ik}}{|\mathcal{Y}(j)| - 1} \tag{6}$$

$$\hat{A} = A \oplus M^T \tag{7}$$ $$\hat{B} = B \oplus M \tag{8}$$

where $|\mathcal{Y}(j)|$ is the number of nodes connected to hyperedge e_j.

Finally, we derive the hyperedge representation $\mathbf{e}_j$ and the updated node representations $z_{h,i}$ and $z_{g,i}$ by aggregating their neighboring information as follows:

$$\mathbf{e}_j = \sum_{i \in \mathcal{Y}_j} \hat{A}_{ji} \mathbf{W}_1 n_i \tag{9}$$

$$z_{h,i} = \sigma\Big(\sum_{j \in \mathcal{P}_i} \hat{B}_{ij} \mathbf{W}_2 e_j\Big) \tag{10}$$ $$z_{g,i} = \sigma\Big(\sum_{j \in \mathcal{N}(i)} \hat{G}_{ij} n_j\Big) \tag{11}$$

A number of prior studies have empirically demonstrated that mapping representations to another latent space via a non-linear projection head contributes to performance improvements in contrastive loss [1, 13]. Consequently, we mapped the outputs of our hypergraph attention and graph attention layers to another latent space through a projection head (see Figure 1). The projection head consists of two MLP layers and an ELU activation function.

3.5 Contrastive Learning

Having the two views of the same hypergraph, each obtained from different attention layers, our contrastive learning treats the same nodes in the two views as positive pairs and all other nodes as negative pairs. The objective function of our contrastive learning is based on the average over positive pairs as follows[1]:

$$\ell\left(z_{h,i}, z_{g,i}\right) = -\log \frac{e^{s(z_{h,i}, z_{g,i})/\tau}}{\sum_{k=1}^{|V|} e^{s(z_{h,i}, z_{g,k})/\tau}} \tag{12}$$

$$\mathcal{L} = \frac{1}{2|V|} \sum_{i=1}^{|V|} \left\{\ell\left(z_{h,i}, z_{g,i}\right) + \ell\left(z_{g,i}, z_{h,i}\right)\right\} \tag{13}$$

[1]Following prior work [13], we only consider positive node pairs.

where τ is the temperature parameter and $s(\cdot,\cdot)$ indicates the cosine similarity of two vectors.

4 EXPERIMENTAL RESULTS

We employed the three widely-used benchmark datasets: **Citeseer** [8], **Cora** [16] and **Pubmed** [18]. They are co-citation datasets, where each dataset's nodes represent documents, and the citations between documents are represented as edges. Table 1 summarizes the statistics of each data. These datasets were divided into a 1:1:8 ratio for training, validation, and testing respectively, following most of the related works [13, 23]. For additional implementation details, visit our code: https://github.com/JongsooLee-HYU/MMACL

Table 1: A summary statistics of the datasets.

Dataset	# nodes	# edges	# classes	# features
Citeseer	1,498	1,107	6	3,703
Cora	1,434	1,579	7	1,433
Pubmed	3,840	7,963	3	500

4.1 Accuracy Improvements

Table 2 shows the hypergraph node classification accuracy of each model on the three co-citation datasets. Our MMACL demonstrates quite competitive results against the baselines. We also observed that the methods based on contrastive learning on hypergraphs [13, 23] outperformed other baselines, indicating that considering valuable information from an augmented graph structure enhances representation learning. We also confirmed that simultaneously considering both the general graph structure and the hypergraph structure leads to superior performance compared to other baselines. While CCL [23] tries to take both structures into account, it involves the convolution operations for representation learning, which does not account for the relative importance of each node. Our MMACL significantly benefits from the mixed attention mechanism, which collaboratively utilizes high-order relationships and pairwise relationships from the two different layers, playing a key role in enhancing performance.

4.2 Ablation Study

The main component of our proposed model is the mixed attention that learns diverse relationships between nodes. Our ablation study aimed to verify whether considering various types of mixed attention in each layer's neighborhood aggregation process aids performance improvement. To this end, we conducted experiments on all possible combinations to validate the performance of each component. Table 3 reports the ablation results, which demonstrate that applying all types of mixed attention achieved the highest performance. Moreover, in most cases, considering a single type of mixed attention showed higher performance compared to not applying mixed attention at all, and considering two types of mixed attention was more effective than using just one. This proves that incorporating various types of mixed attention in representation learning significantly contributes to improving hypergraph representation learning and thus enhancing the downstream task performance.

Table 2: Accuracy (%, mean ± standard deviation) of different methods derived from a total of five experimental runs. The best accuracy in each dataset is highlighted in bold, and the second best is denoted by underline. The results of the compared baselines are borrowed from [23].

Method	Citeseer	Cora	Pubmed
NN	27.0 ± 9.0	24.5 ± 5.7	41.2 ± 2.7
SVM	21.0 ± 1.3	32.6 ± 5.2	41.3 ± 4.1
GCN [12]	63.0 ± 1.9	72.1 ± 1.9	69.6 ± 6.3
HyperGCN [26]	54.7 ± 9.8	52.2 ± 11.4	60.0 ± 10.7
FastHyGCN [26]	56.1 ± 11.2	50.4 ± 11.7	54.4 ± 10.0
HGNN [7]	61.1 ± 2.2	76.9 ± 1.8	63.3 ± 2.2
HNHN [5]	64.8 ± 1.6	71.3 ± 1.9	75.9 ± 1.5
UniGCN [11]	70.9 ± 1.0	78.3 ± 1.7	78.8 ± 1.7
UniGAT [11]	70.7 ± 1.0	78.5 ± 1.8	78.7 ± 1.7
HCNH [24]	71.4 ± 1.2	78.5 ± 1.7	77.1 ± 3.6
HCoN [25]	71.2 ± 2.7	79.0 ± 1.5	80.4 ± 1.1
TriCL [13]	71.1 ± 0.8	**81.0 ± 1.0**	80.3 ± 1.6
CCL [23]	<u>71.9 ± 0.8</u>	79.7 ± 1.1	<u>81.9 ± 0.9</u>
Ours	**72.3 ± 1.2**	<u>80.6 ± 1.2</u>	**82.6 ± 0.6**

Table 3: Ablation study.

$\hat{A}$	$\hat{B}$	$\hat{G}$	Citeseer	Cora	Pubmed
-	-	-	68.9 ± 1.8	77.6 ± 1.5	81.4 ± 0.6
✓	-	-	69.3 ± 1.7	77.6 ± 1.5	81.5 ± 0.5
-	✓	-	69.1 ± 1.8	77.6 ± 1.5	81.4 ± 0.5
-	-	✓	69.0 ± 1.7	78.3 ± 1.6	82.3 ± 0.6
✓	✓	-	69.5 ± 1.6	77.6 ± 1.5	81.8 ± 0.5
✓	-	✓	69.8 ± 1.7	79.3 ± 1.4	82.8 ± 0.6
-	✓	✓	69.7 ± 1.7	79.6 ± 1.3	82.4 ± 0.6
✓	✓	✓	72.3 ± 1.2	80.6 ± 1.2	82.6 ± 0.6

5 CONCLUSION

In this paper, we discussed a novel way towards improving hypergraph representation learning. We proposed MMACL, introducing a mixed attention mechanism that considers both pairwise relationships between nodes derivable from general graph structures and high-order relationships from hypergraph structures. Such mixed attention based on various node relationships produces two different views, and contrastive learning based on the same nodes associated with the two views results in more impressive representations. Our experiments on several co-citation datasets and the ablation study confirmed the effectiveness of our approach's components in enhancing node classification performance.

ACKNOWLEDGEMENT

This work was partly supported by (1) the National Research Foundation of Korea (NRF) grant funded by the Korea government (*MSIT) (No.2018R1A5A7059549) and (2) the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2020-0-01373,Artificial Intelligence Graduate School Program (Hanyang University)). *Ministry of Science and ICT

REFERENCES

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[2] Hyunjin Choo and Kijung Shin. 2022. On the persistence of higher-order interactions in real-world hypergraphs. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*. SIAM, 163–171.

[3] Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljačić, Shang-Wen Li, Scott Yih, Yoon Kim, and James Glass. 2022. DiffCSE: Difference-based Contrastive Learning for Sentence Embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4207–4218.

[4] Kaize Ding, Jianling Wang, Jundong Li, Dingcheng Li, and Huan Liu. 2020. Be more with less: Hypergraph attention networks for inductive text classification. *arXiv preprint arXiv:2011.00387* (2020).

[5] Yihe Dong, Will Sawin, and Yoshua Bengio. 2020. Hnhn: Hypergraph networks with hyperedge neurons. *arXiv preprint arXiv:2006.12278* (2020).

[6] Wenqi Fan, Xiaorui Liu, Wei Jin, Xiangyu Zhao, Jiliang Tang, and Qing Li. 2022. Graph trend filtering networks for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 112–121.

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

[8] C Lee Giles, Kurt D Bollacker, and Steve Lawrence. 1998. CiteSeer: An automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*. 89–98.

[9] Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, Vol. 2. IEEE, 1735–1742.

[10] Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. 2022. Vision gnn: An image is worth graph of nodes. *Advances in Neural Information Processing Systems* 35 (2022), 8291–8303.

[11] Jing Huang and Jie Yang. 2021. UniGNN: a Unified Framework for Graph and Hypergraph Neural Networks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. International Joint Conferences on Artificial Intelligence Organization, 2563–2569.

[12] Thomas N Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

[13] Dongjin Lee and Kijung Shin. 2023. 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*, Vol. 37. 8456–8464.

[14] Jongsoo Lee, Byeongtae Park, and Dong-Kyu Chae. 2023. DuoGAT: Dual Time-oriented Graph Attention Networks for Accurate, Efficient and Explainable Anomaly Detection on Time-series. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 1188–1197.

[15] Xien Liu, Xinxin You, Xiao Zhang, Ji Wu, and Ping Lv. 2020. Tensor graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 8409–8416.

[16] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval* 3 (2000), 127–163.

[17] Taesung Park, Alexei Efros, Richard Zhang, and Jun-Yan Zhu. 2020. Contrastive Learning for Unpaired Image-to-Image Translation. *Computer Vision–ECCV 2020. ECCV 2020. Lecture Notes in Computer Science*, 12354 (2020).

[18] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine* 29, 3 (2008), 93–93.

[19] Eunseon Seong, Harim Lee, and Dong-Kyu Chae. 2024. Self-Supervised Framework based on Subject-wise Clustering for Human Subject Time Series Data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 22341–22349.

[20] Liang Sun, Shuiwang Ji, and Jieping Ye. 2008. Hypergraph spectral learning for multi-label classification. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 668–676.

[21] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

[22] Azmine Toushik Wasi, Taki Hasan Rafi, Raima Islam, and Dong-Kyu Chae. 2024. BanglaAutoKG: Automatic Bangla Knowledge Graph Construction with Semantic Neural Graph Filtering. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*.

[23] Hanrui Wu, Nuosi Li, Jia Zhang, Sentao Chen, Michael K Ng, and Jinyi Long. 2024. Collaborative contrastive learning for hypergraph node classification. *Pattern Recognition* 146 (2024), 109995.

[24] Hanrui Wu and Michael K Ng. 2022. Hypergraph convolution on nodes-hyperedges network for semi-supervised node classification. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 16, 4 (2022), 1–19.

[25] Hanrui Wu, Yuguang Yan, and Michael Kwok-Po Ng. 2022. Hypergraph collaborative network on vertices and hyperedges. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 3 (2022), 3245–3258.

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