sensors

MDPI

Article

Node Classification Method Based on Hierarchical Hypergraph Neural Network

Feng Xu [1,2], Wanyue Xiong [1], Zizhu Fan [3,*] and Licheng Sun [4]

[1] School of Electrical and Automation Engineering, East China Jiaotong University, Nanchang 330013, China; 2022029081100008@ecjtu.edu.cn (F.X.); 2021088085400002@ecjtu.edu.cn (W.X.)
[2] School of Mechanical and Electrical Engineering, Quzhou College of Technology, Quzhou 324000, China
[3] College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai 200090, China
[4] Mechanical and Electrical Room, Quzhou Special Equipment Inspection & Testing Research Institute, Quzhou 324000, China; amenslc@163.com
* Correspondence: zzfan3@163.com

Abstract: Hypergraph neural networks have gained widespread attention due to their effectiveness in handling graph-structured data with complex relationships and multi-dimensional interactions. However, existing hypergraph neural network models mainly rely on planar message-passing mechanisms, which have limitations: (i) low efficiency in encoding long-distance information; (ii) underutilization of high-order neighborhood features, aggregating information only on the edges of the original graph. This paper proposes an innovative hierarchical hypergraph neural network (HCHG) to address these issues. The HCHG combines the high-order relationship-capturing capability of hypergraphs, uses the Louvain community detection algorithm to identify community structures within the network, and constructs hypergraphs layer by layer. In the bottom-level hypergraph, the model establishes high-order relationships through direct neighbor nodes, while in the top-level hypergraph, it captures global relationships between aggregated communities. Through three hierarchical message-passing mechanisms, the HCHG effectively integrates local and global information, enhancing the multi-resolution representation ability of node representations and significantly improving performance in node classification tasks. In addition, the model performs excellently in handling 3D multi-view datasets. Such datasets can be created by capturing 3D shapes and geometric features through sensors or by manual modeling, providing extensive application scenarios for analyzing three-dimensional shapes and complex geometric structures. Theoretical analysis and experimental results show that the HCHG outperforms traditional hypergraph neural networks in complex networks.

Keywords: hypergraph neural networks; hierarchical representations; Nnode classification

Citation: Xu, F.; Xiong, W.; Fan, Z.; Sun, L. Node Classification Method Based on Hierarchical Hypergraph Neural Network. *Sensors* **2024**, *24*, 7655. https://doi.org/10.3390/s24237655

Academic Editor: Liang-Jian Deng

Received: 30 October 2024
Revised: 23 November 2024
Accepted: 25 November 2024
Published: 29 November 2024

1. Introduction

Graph neural networks (GNNs) have garnered significant attention recently, emerging as powerful tools for processing graph-structured data. They are widely applied in various domains such as social networks [1,2], Photogrammetry [3,4], 3D object classification [5,6], the Internet of Things [7], and bioinformatics [8,9]. GNNs can effectively capture local relationships between nodes within a graph by aggregating information from neighboring nodes, enabling tasks such as node classification and link prediction [10,11]. For instance, in 3D object classification, GNNs utilize point cloud and depth data from sensors like LiDAR and RGB-D cameras, leveraging spatial relationships among nodes to enhance classification accuracy by capturing geometric features, especially in complex scenes. Classic GNN models, including GCN [12], GAT [13], and GraphSAGE [8], have achieved substantial advances in representation learning for graph data. However, these models exhibit limitations when handling complex graph structures and long-range dependencies among nodes.

Specifically, the flat message-passing mechanism of traditional GNNs makes it difficult to capture relationships between distant nodes effectively [14,15]. Furthermore, existing

graph structures are often overly simplistic, primarily designed for binary relationships, which limits their ability to express multi-relational interactions fully. This directly results in traditional graph neural networks performing poorly in capturing global and local graph information, affecting classification effectiveness. Lastly, GNNs face memory and GPU memory constraints when handling complex graphs, making it challenging to scale to practical applications such as community detection [8].

The introduction of hypergraph structures presents a new approach to addressing these issues. Hypergraphs can capture high-order relationships among multiple nodes, transcending superficial binary relationships, thus better representing multilateral interactions in complex networks [16]. For example, hypergraphs are embedded into a low-dimensional space for clustering analysis, revealing the underlying group structures within the data. However, traditional hypergraph structures may have limitations when handling dynamically changing datasets, as they cannot adapt to rapidly evolving relationships, leading to delayed classification results [17]. At the same time, although hyperedge convolution layers can learn higher-order relationships in complex data, their high computational complexity affects the practicality of the model [18].

The hierarchical mechanism offers a new approach to addressing long-range dependencies and expressing hierarchical information. The hierarchical message-passing mechanism, which progressively aggregates local and global information, can effectively enhance the robustness and expressiveness of node representations [19]. For instance, hierarchical graph pooling methods, such as G-U-Net and DiffPool, have significantly improved in graph classification tasks [20,21]. Although some studies have combined hierarchical structures with graph learning models, there remains a lack of research on integrating hierarchical mechanisms with hypergraph neural networks for node classification.

To address key challenges in traditional GNNs and hypergraph neural networks, we introduce the Hierarchical Hypergraph Neural Network (HCHG). While GNNs struggle with long-range dependencies and global context, hypergraph neural networks excel at capturing higher-order relationships but are computationally expensive. HCHG constructs hypergraphs layer by layer, with the first layer capturing local relationships and subsequent layers aggregating community nodes to model global relationships, enhancing the model's ability to represent local and global interactions. Additionally, HCHG uses a multi-layer message-passing mechanism, including bottom-up, lateral, and top-down flows, which strengthens node representations and reduces the computational burden, efficiently handling complex networks.

The main contributions of this paper are as follows:

1. We propose the HCHG model. This novel approach combines hierarchical structures with hypergraph neural networks to effectively capture local and global relationships in node classification and link prediction tasks, improving performance on complex graphs.
2. The HCHG model introduces a hierarchical construction method, using the Louvain community detection algorithm to build higher-order relationship networks, enhancing the model's ability to represent complex network structures.
3. Our method performs excellently on six classification datasets and three link prediction datasets, achieving significant performance improvements across multiple tasks.

2. Related Work

Node classification is a core task in graph representation learning, aiming to predict a node's category based on its structure and attributes. Although Graph Neural Networks (GNNs), such as GCNs [12] and GraphSAGE [8], have achieved significant progress in node classification tasks, they still face challenges in capturing long-range dependencies and handling sparse graph structures, which are critical for node classification in complex networks.

In recent years, hypergraph representation learning has provided new solutions by modeling higher-order relationships between nodes. For example, end-to-end hypergraph

convolution [22] and Dynamic Hypergraph Neural Networks (DHGNN) [23] effectively capture complex node interactions. While simplifying computation, HyperGCN [24] may lose some high-order structural information. Furthermore, HyperSAGE [25] enhances generalization capabilities through an inductive message-passing mechanism. However, these methods still face limitations in integrating global and local information, particularly in dynamic and complex network scenarios. Researchers have introduced hierarchical structures into node classification tasks to enhance the capacity for multi-granularity semantic modeling. Methods such as DiffPool [21] and ASAP [19] significantly improve representation through node aggregation, but their applications in node-level tasks remain limited by the shortcomings of single-layer models.

We propose the HCHG, which leverages a multi-layer structure and diverse message-passing mechanisms to effectively integrate local and global information while enhancing adaptability and generalization. Experiments demonstrate that HCHG achieves outstanding performance across multiple node classification datasets, particularly excelling in modeling higher-order relationships and complex node interactions.

3. Motivation and Background

Given a hypergraph structure $HG = (V, E, H)$, the objective is to construct a mapping function $F : HG \to Z \in \mathbb{R}^{|V| \times d}$ that captures the features of node v_i and its relationships within the hypergraph. The effectiveness of F will be evaluated through tasks such as node classification and link prediction.

In hypergraph neural networks, the node update mechanism differs from traditional Graph Neural Networks. A hypergraph consists of nodes and edges (hyperedges); each can connect multiple nodes. Below, we will detail the fundamental mechanism of node updates in hypergraph neural networks. Consider the hypergraph $HG = (V, E, H)$, where V is the set of nodes, E is the set of edges, and H is the set of hyperedges. Each hyperedge $h_i \in H$ connects multiple nodes. The node update process in hypergraph neural networks typically involves the following two steps:

1. Message Aggregation

At layer l, the message aggregation for node v is completed through all hyperedges connected to v. The aggregation can be represented as follows:

$$m_{agg}^{(l)} = \text{Aggregate}_N\left(\{W_{kv}^{(l)}, h_k^{(l)} \mid k \in N(v)\}\right) \tag{1}$$

where $\text{Aggregate}_N(\cdot)$ is a differentiable aggregation function, $W_{kv}^{(l)}$ is the association matrix between node k and node v, $h_k^{(l)}$ represents the features of node k at layer l, and $N(v)$ is the set of neighbor nodes directly connected to node v.

2. Node Feature Update

Using the aggregated message $m_{agg}^{(l)}$, the feature of node v is updated as follows:

$$\begin{cases} h_v^{(l)} = \text{Combine}(m_{agg}^{(l)}, m_{vgg}^{(l)}) \\ m_{vgg}^{(l)} = \text{Aggregate}_I\left(\{W_{kv}^{(l)} \mid k \in N(v)\}\right) h_v^{(l-1)} \end{cases} \tag{2}$$

where $\text{Combine}(\cdot)$ is a nonlinear fusion function, $\text{Aggregate}_I(\cdot)$ is another aggregation function, and $h_v^{(l-1)}$ refers to the features of node v from the previous layer.

4. Hierarchical Hypergraph Neural Networks

We propose the HCHG framework to enable node representations to receive long-range messages and multi-granularity semantics through a hierarchical hypergraph structure. As illustrated in Figure 1, this framework first creates a progressively refined hierarchical structure, processing the input hypergraph in layers. Next, hypergraphs are constructed to facilitate message-passing between supernodes based on the connectivity among hierarchical nodes. To ensure adequate information flow, we design three message propagation mechanisms: bottom-up, intra-layer, and top-down. These mechanisms en-

sure information exchange both within the same level and across different levels. Finally, we train the model using task-specific loss functions and gradient descent algorithms, optimizing node representations and overall performance.

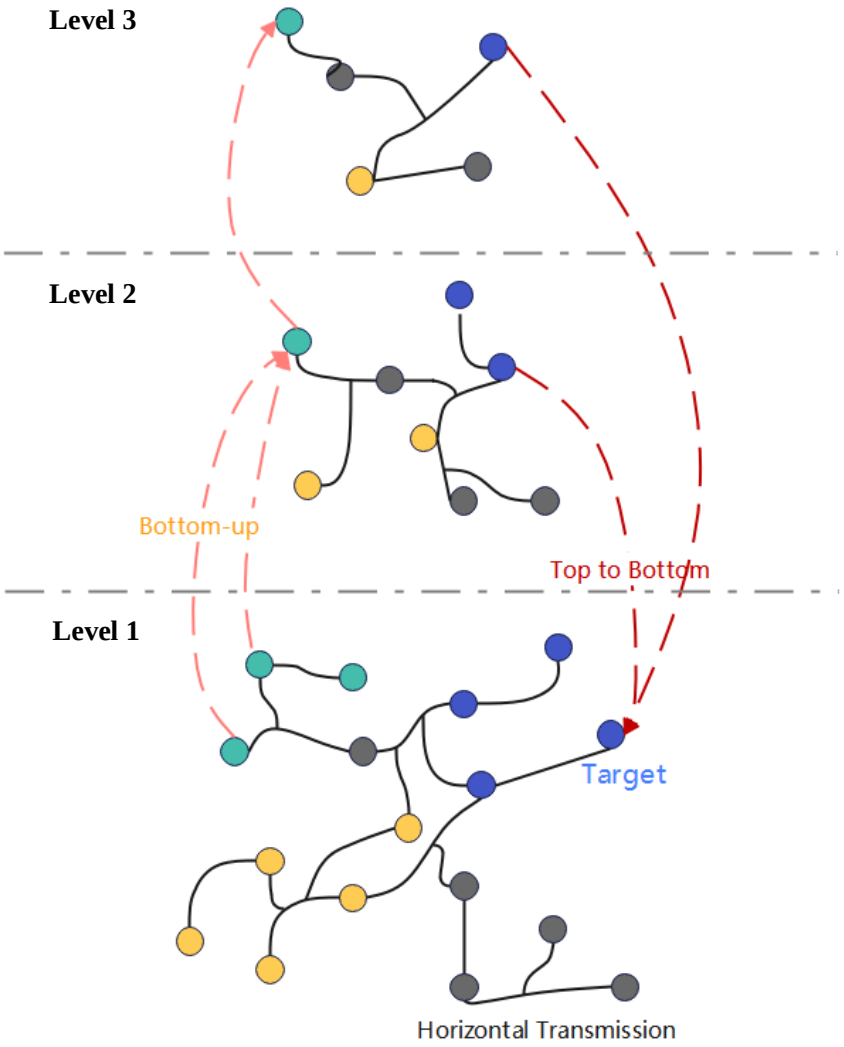

Figure 1. Hierarchical Hypergraph Neural Network Framework (Different colors represent different types of nodes).

4.1. Hierarchical Structure Partitioning

For complex multi-node relational systems, we first represent the raw data as a graph $G = (V, E)$, where V is the set of nodes representing different entities, and E is the set of edges depicting relationships between these entities. To capture the higher-order relationships among nodes more effectively, we introduce hyperedges H, defined as $h = \{v_1, v_2, \ldots, v_n\}$, where $v_i \in V$. This constructs a hypergraph $HG = (V, E, H)$.

On this basis, our study employs a hierarchical mechanism that simplifies the graph's structure through layer-wise abstraction and aggregation. In each layer, we convert the node-set V into a higher-level set of supernodes by partitioning the nodes into communities. For instance, the supernodes in the first layer arise from the community partitioning of the original nodes. In contrast, subsequent layers' supernodes are formed by combining the supernodes from the preceding layer. This approach allows the hierarchical structure to evolve progressively from the lower to the upper layers, facilitating higher-level analysis and modeling.

4.2. Hierarchical Hypergraph Construction

In constructing the hypergraph, the first layer hypergraph starts from the original. We utilize the Louvain community detection algorithm to identify communities within the node set V through modularity optimization. Each identified community $C_i \subseteq V$ forms a supernode, resulting in a supernode set $S_1 = \{SC_1^1, SC_2^1, \ldots, SC_p^1\}$, considered the first layer. Once the supernodes are established, we create edges between them. Suppose an edge exists between two communities, C_i and C_j (for example, through common original nodes or connecting hyperedges). In that case, we create an edge e_{ij} between their corresponding supernodes SC_i and SC_j.

We view the original nodes within each community as components of the hyperedges for hyperedge construction. Specifically, for the set of original nodes $V_{C_i} = \{v_k \mid v_k \in C_i\}$ within each community C_i, we define a hyperedge h_i that includes all original nodes within community C_i, i.e., $h_i = V_{C_i}$. Thus, the set of hyperedges H_1 of the first layer hypergraph HG_1 consists of all hyperedges corresponding to the communities: $H_1 = \{h_i \mid i = 1, 2, \ldots, p\}$. Ultimately, the first layer hypergraph can be represented as $HG_1 = (S_1, E_1, H_1)$, where E_1 is the set of edges between supernodes, and H_1 is the collection of hyperedges within S_1.

When generating the second layer hypergraph, we again apply the Louvain community detection algorithm to aggregate the first layer supernodes, forming the second layer supernode set $S_2 = \{SC_1^2, SC_2^2, \ldots, SC_m^2\}$. During this process, if there exists an edge between the second layer supernodes SC_k^2 and SC_l^2, we establish an edge e_{kl} between them. Additionally, we construct hyperedges for the second layer supernodes, defining each hyperedge h_k formed by the supernodes in community C_k. Specifically, each hyperedge h_k in the second layer consists of all first layer supernodes SC_i^1 that belong to community C_k, i.e., $h_k = \{SC_i^1 \mid SC_i^1 \in C_k\}$. Ultimately, the second layer hypergraph is represented as $HG_2 = (S_2, E_2, H_2)$, where H_2 is the collection of hyperedges within S_2.

4.3. Hierarchical Information Propagation

The hierarchical message-passing mechanism enhances node representations through long-range interaction and neighborhood aggregation. This mechanism does not interfere with the process of learning planar node representations, thereby effectively preserving the original information of the nodes. The hierarchical message-passing mechanism consists of the following three methods.

After obtaining the node representations of the $(t-1)$-th layer hypergraph $h_{s_j^{t-1}}^{(l)}$ using the node update mechanism, these representations are aggregated to update the supernode representations in the t-th layer. A schematic illustration is shown in Figure 2, with the mathematical expression for aggregation given by:

$$a_{SC_i^t}^{(l)} = \frac{1}{|SC_i^t| + 1} \left(\sum_{s_j^{t-1} \in SC_i^t} h_{s_j^{t-1}}^{(l)} + h_{SC_i^t}^{(l-1)} \right) \tag{3}$$

where SC_i^t is the supernode in the t-th layer, s_j^{t-1} represents the nodes belonging to SC_i^t in the $(t-1)$-th layer, $|SC_i^t|$ is the number of nodes belonging to SC_i^t in the $(t-1)$-th layer, and $h_{SC_i^t}^{(l-1)}$ represents the node representations of the supernode SC_i^t in the t-th layer of the $(l-1)$-th layer.

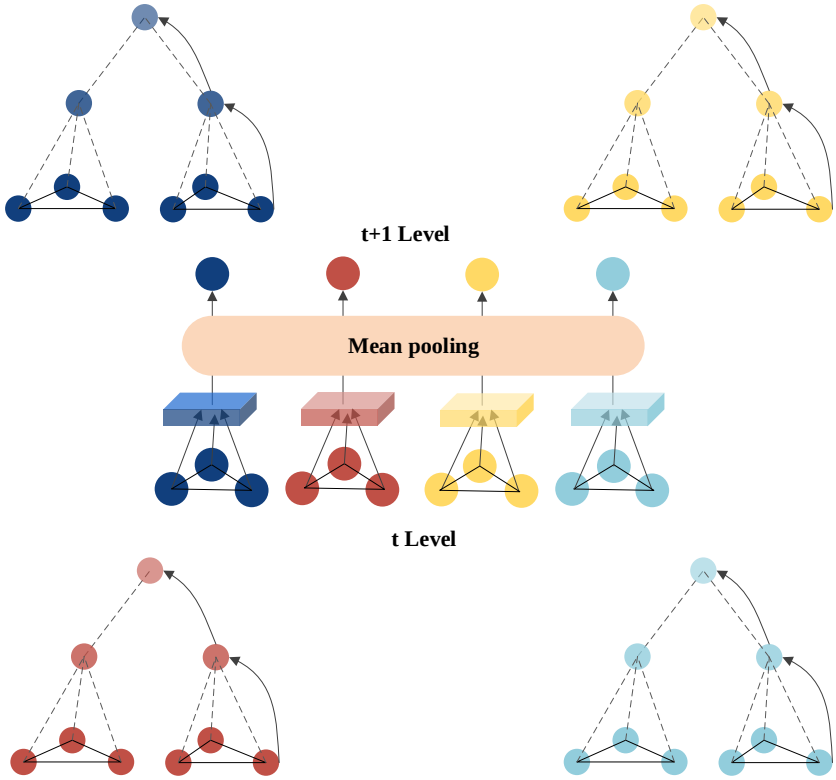

Figure 2. Schematic Diagram of Bottom-Up Propagation.

4.4. Inter-Layer Propagation

Inter-layer propagation primarily relies on hypergraph neural networks' planar message-passing mechanism to aggregate neighboring information and update node representations within the same layer. A schematic illustration is shown in Figure 3. Based on the bottom-up propagation, the aggregation process of information from higher-layer supernodes is represented as follows:

$$\begin{cases} m_{agg}^{(l)} = \text{Aggregate}_N\left(\{W_{kv}^{(l)}, a_u^{(l)} \mid k \in N(v)\}\right) \\ m_{vgg}^{(l)} = \text{Aggregate}_I\left(\{W_{kv}^{(l)} \mid k \in N(v)\}\right) h_v^{(l-1)} \\ b_v^{(l)} = \text{Combine}(m_{agg}^{(l)}, m_{vgg}^{(l)}) \end{cases} \tag{4}$$

where $a_u^{(l)}$ is the representation of supernode u after bottom-up propagation, and $b_v^{(l)}$ is the updated feature representation of node v in layer l. The meanings of the remaining parameters are consistent with those in Section 3.

The node representations from the hypergraphs $\{G_2, \ldots, G_T\}$ are used to update the node representations in the original graph G. The importance of information from different layers varies depending on the specific task. Therefore, an attention mechanism proposed by Veličković et al. is employed to adaptively learn the weights of the information during the top-down integration process [26]. A schematic illustration is shown in Figure 4, represented as follows:

$$h_v^l = \text{ReLU}\left(\sum_{u \in N^l(v)} \alpha_{uv}^l W^l \cdot \text{MEAN}(b_u^l)\right) \tag{5}$$

where α_{uv}^l is a trainable attention coefficient that represents the connection weight between nodes v and u across different layers. Here, b_u is the bias term, MEAN denotes the element-wise average operation, and ReLU is the activation function. Ultimately, the node information representation from the last layer L is output using the following formula [27]:

$$z_v = \sigma\left(\sum_{u \in N^L(v)} \alpha_{uv}^L W^L \cdot \mathrm{MEAN}(b_u^L)\right) \tag{6}$$

where σ is the Euclidean normalization function that adjusts values to the range of $[0, 1]$. The final generated node representations $Z \in \mathbb{R}^{n \times d}$ are used for classification, with each row $z_v \in Z$ representing the representation of a node v.

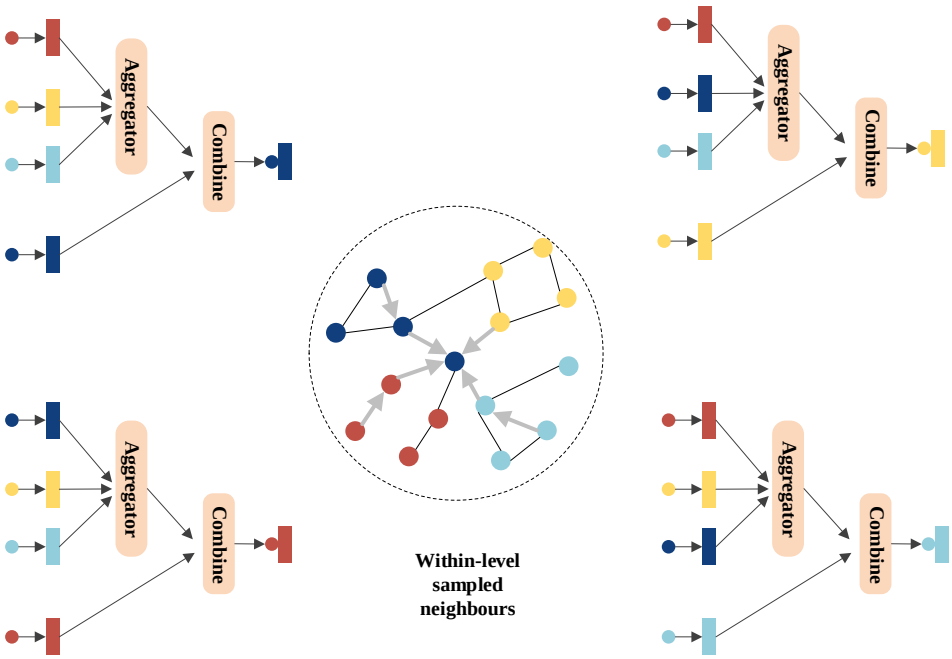

Figure 3. Schematic Diagram of Inter-Layer Propagation.

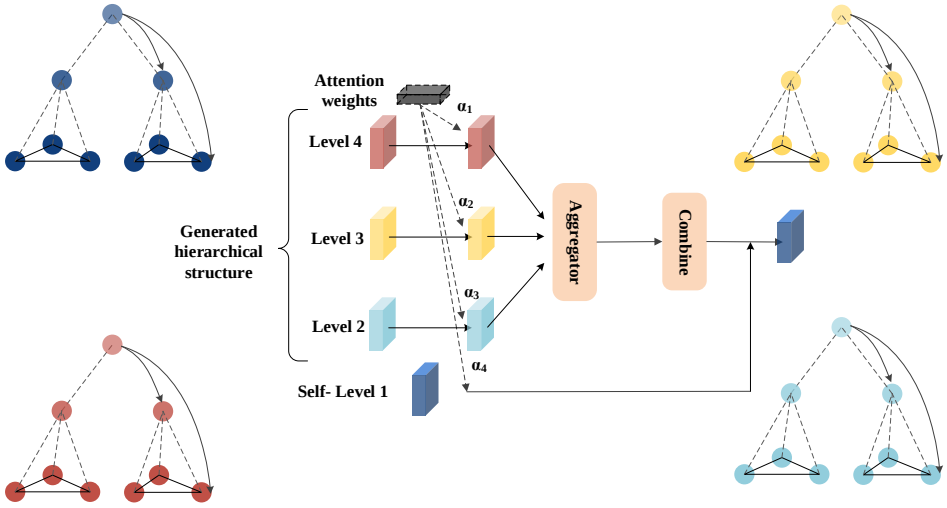

Figure 4. Schematic Diagram of Top-Down Propagation.

4.5. Model Training

In the experimental section, HCHG is applied to the training and prediction of semi-supervised node classification. The designed cross-entropy loss function is defined as follows:

$$L = -\sum_{v \in V} y_v^T \log(\mathrm{Softmax}(z_v)) \tag{7}$$

where y_v^T is the one-hot encoded label vector representing node v. The loss function L can be specifically modified based on different tasks.

The computational complexity analysis presented in this paper consists of three main components: the hierarchical construction of graphs, the construction of hypergraphs, and the information propagation mechanism. The time complexity of the Louvain algorithm during the hierarchical construction process at each layer is $O(m_t \log n_t)$. The complexity of hypergraph construction is $O(n_t + e_t)$. The complexity of the information propagation mechanism is $O(n_t d_t + e_t)$. Ultimately, the overall computational complexity can be expressed as:

$$O\left(\sum_{t=1}^{T}(m_t \log n_t + n_t + e_t + n_t d_t)\right) \tag{8}$$

where T is the number of layers in the graph, and m_t, n_t, e_t, and d_t represent the number of edges, nodes, hyperedges, and the average degree of nodes in layer t, respectively.

The following Algorithm 1 provides a brief summary of the construction process of the HCHG model.

Algorithm 1: Hierarchical Hypergraph Neural Network

Input : Graph $HG = (V, E, H)$
Output: Node representation $Z \in \mathbb{R}^{n \times d}$

$h_v^{(0)} \leftarrow x_v$;
$H = \{G_t \mid t = 1, 2, \ldots, T\}$;
for $l \leftarrow 1$ **to** L **do**
  $h_v^{(l)} = \mathrm{ReLU}\left(\sum_{u \in N(v)} \frac{W^{(l)} h_u^{(l-1)}}{\sqrt{|N(u)||N(v)|}} + \frac{W^{(l)} h_v^{(l-1)}}{\sqrt{|N(v)||N(v)|}}\right)$;
  for $t \leftarrow 2$ **to** T **do**
    $a_{SC_i^t}^{(l)} = \frac{1}{|SC_i^t|+1}\left(\sum_{s_j^{t-1} \in SC_i^t} h_{s_j^{t-1}}^{(l)} + h_{SC_i^t}^{(l-1)}\right)$
    $b_v^{(l)} = \mathrm{Re}LU(\sum_{u \in N(v)} \frac{W^{(l)} a_u^{(l)}}{\sqrt{|N(u)||N(u)|}} + \frac{W^{(l)} a_v^{(l)}}{\sqrt{|N(v)||N(v)|}}), \forall v \in HG$;
  end
  for $v \in G$ **do**
    if $l < L$ **then**
      $h_v^l = \mathrm{ReLU}\left(\sum_{u \in N^l(v)} \alpha_{uv}^l W^l \cdot \mathrm{MEAN}(b_u^l)\right)$
    end
    else
      $z_v = \sigma\left(\sum_{u \in N^L(v)} \alpha_{uv}^L W^L \cdot \mathrm{MEAN}(b_u^L)\right)$
    end
  end
end

5. Experimental Analysis

5.1. Datasets

To verify the model's overall performance, several commonly used datasets were used in the experiments, including graph-structured and multiview datasets. The graph-structured datasets contain rich node and edge relationships. In contrast, multi-view

datasets can be generated by creating images from 3D models or using sensors to capture data from different angles, providing a comprehensive representation of the objects. Table 1 summarizes all the datasets used in the experiments.

Table 1. Categorical Dataset.

Dataset	Nodes/Feature	Train/(val)/Test	Class
Cora	2708/1433	140/500/1000	7
Citeseer	3312/3703	140/500/1000	6
Pubmed	19,717/500	60/500/1000	3
Zoo	101/16	66/35	7
Grid	400	-	-
ModelNet40	12,311/(2048/4096)	9843/2468	44
NTU2012	2012/(2048/4096)	1640/372	67

1. Graph-structured Datasets

Cora and Citeseer are citation network datasets introduced by Sen et al. [28]. Cora consists of 2708 scientific publications and 5429 links. Each publication is represented as a node with a 1433-dimensional word vector as its feature. Citeseer consists of 3312 scientific publications and 4660 links, with each node having a 3703-dimensional word vector.

Pubmed, introduced by Namata et al. [29], includes 19,717 scientific publications about diabetes from the Pubmed database, divided into three classes. Each node is represented by a TF/IDF weighted word vector consisting of 500 words.

The Zoo dataset is downloaded from the UCI website. Each sample contains 17 Boolean attributes. Hyperedges are created for nodes with the same classification feature value.

Grid is a synthetic 2D grid graph representing a 20×20 grid with 400 nodes and no node features. This dataset is only for link prediction.

2. Multiview Datasets

The ModelNet40 dataset consists of 12,311 objects from 40 popular categories, split into training and test sets, with 9843 objects for training and 2468 objects for testing. NTU2012 (National Taiwan University (NTU) 3D Dataset) is a dataset from the computer vision/graphics field. It comprises 2012 3D shapes from 67 categories, including cars, chairs, chessboards, chips, clocks, cups, doors, frames, pens, plant leaves, etc. In the NTU2012 dataset, 80% of the data is used for training, and the remaining 20% is used for testing.

In the experiments, each 3D object is represented by extracted features. Two state-of-the-art shape representation methods, Multi-View Convolutional Neural Network (MVCNN) and Group-View Convolutional Neural Network (GVCNN), are adopted here. These two methods have shown satisfactory performance in representing 3D objects. Following the experimental settings of MVCNN and GVCNN, multiple views of each 3D object are generated.

5.2. Experimental Setup and Results

Applying the HCHG model to the node classification task on the dataset, multiple averaged results are preserved in the experiments. For HGNN convolutional layers $\{256, 128, 64, 32\}$, experiments are conducted with 2 or 3 convolutional layers and a 1×10^{-5} learning rate. The correlation between different views is modeled for the multiview data by generating a hypergraph G. For each view's data, an adjacency matrix H_i of the hypergraph is constructed based on the HGNN proposed by Feng et al. [22]. As shown in Figure 5, the adjacency matrices H_i of different views are concatenated to construct the adjacency matrix H of the multiview hypergraph, thus creating a hypergraph structure with multiview features. Since the macro-level graph, G_T of the macro-level layer is relatively small, during the experiment, we averaged the input label information by pooling G_T and added the loss function separately to calculate the edge information in the T layer. The experimental results showed no impact on the data. The possible reason is that the macro-level layer contains very little information and has minimal influence on the target nodes after top-

down propagation. Therefore, the small changes in the node connection mode in G_T have a negligible effect on the results.

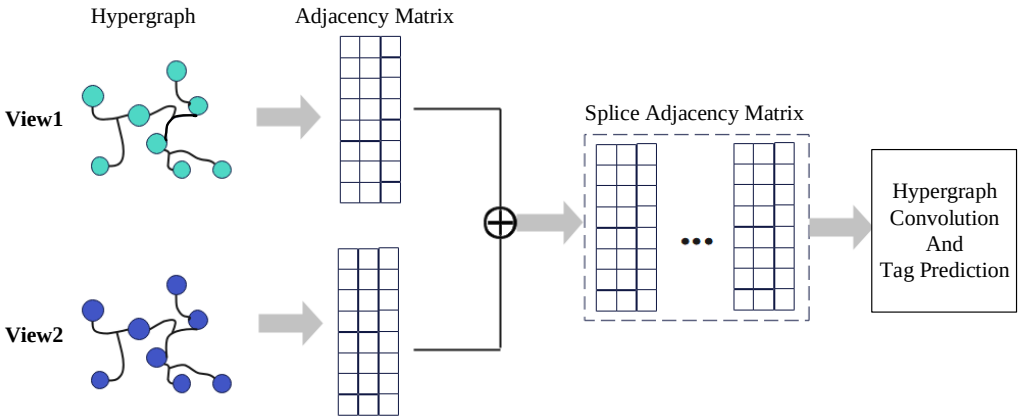

Figure 5. Multimodal Data Fusion.

A comparison of the HCHG model with other graph neural network models for node classification was conducted. Experimental results were collected for eight models, including Hyper-Conv, HC-GNN [27], GCN [12], GAT [13], FastGCN [30], LADIES [31], and DNGNN [32], HJRL [33] on standard datasets, as shown in Table 2. For the link prediction, six models—GCN, GraphSAGE [8], GIN [14], G-U-Net [20], GXN [34], and HC-GNN—were compared, as shown in Table 3. The results with node and without node features were evaluated in the experiments with the Cora dataset. The results show that HCHG performs well in terms of node classification and link prediction accuracy.

Table 2. Average Test Accuracy (%) ± Standard Deviation for Node Classification Tasks.

Model (Author, Year)	Zoo	Cora	Pubmed	Citeseer
GCN [12]	60.0 ± 1.5	80.2 ± 0.9	77.9 ± 1.1	64.8 ± 1.4
GAT [13]	48.5 ± 1.2	77.2 ± 1.0	77.5 ± 0.8	62.0 ± 1.3
FastGCN [30]	37.8 ± 1.6	78.0 ± 0.8	74.4 ± 1.3	63.5 ± 1.5
LADIES [31]	37.8 ± 1.7	78.3 ± 0.7	76.8 ± 1.2	65.0 ± 1.1
Hyper-Conv [27]	93.1 ± 0.4	82.7 ± 0.5	78.4 ± 0.6	71.2 ± 0.7
LE [27]	97.0 ± 0.2	82.3 ± 0.4	78.7 ± 0.5	70.4 ± 0.6
HC-GNN [27]	85.7 ± 0.5	79.0 ± 0.6	78.7 ± 0.4	65.9 ± 1.0
HJRL [29]	96.3 ± 0.3	77.6 ± 0.5	77.3 ± 0.6	65.1 ± 1.2
HCHG (Ours)	97.1 ± 0.2	79.8 ± 0.6	79.4 ± 0.5	66.2 ± 1.0

Table 3. Average Test Accuracy (%) ± Standard Deviation for Link Prediction Tasks.

Model (Author, Year)	Grid	Cora-Feat	Cora-noFeat
GCN [12]	76.3 ± 1.2	86.9 ± 0.9	78.5 ± 1.1
GraphSAGE [8]	77.5 ± 1.1	87.0 ± 0.7	74.1 ± 1.3
GIN [14]	75.6 ± 1.0	86.2 ± 0.8	78.2 ± 1.2
G-U-Net [20]	70.1 ± 1.5	90.9 ± 0.6	77.2 ± 1.0
GXN [34]	64.2 ± 1.4	88.9 ± 0.8	78.1 ± 1.1
HC-GNN [27]	80.1 ± 1.3	89.4 ± 0.7	77.6 ± 1.0
HCHG (Ours)	87.8 ± 0.9	82.1 ± 1.2	78.5 ± 1.1

For the multiview dataset (Table 4), a comparison was made between Hyper-Conv, LADIES, HGNN, HJRL, HGNN+ [35], and HC-HGNN. HCHG showed an improvement of approximately 6% on the NTU2012 dataset compared to other models, while HCHG and HC-GNN showed improvements on the ModelNet40 dataset. The superior performance of HCHG may be attributed to its hierarchical structure, which allows the model to capture the

topological information of the graph, i.e., the message propagated from distant nodes in the graph. Moreover, the intermediate and macro-level semantics reflected in the hierarchical structure are encoded through bottom-up, intra-layer, and top-down propagation. On the ModelNet40 dataset, the HCHG model achieved an accuracy of 97%, surpassing other models such as Hyper-Conv, which attained 92%, demonstrating its adaptability to diverse data structures.

Table 4. Average Test Accuracy (%) ± Standard Deviation for Node Classification Tasks.

Model (Author, Year)	NTU2012	ModelNet40
Hyper-Conv [27]	79.4 ± 0.8	91.1 ± 1.2
LE [27]	83.2 ± 1.6	94.1 ± 1.3
HGNN [35]	84.2 ± 1.4	96.7 ± 1.2
HGNN+ [35]	84.2 ± 1.5	96.9 ± 1.1
HC-GNN [27]	83.3 ± 1.0	98.1 ± 0.8
HJRL [29]	86.1 ± 1.3	95.8 ± 1.4
HCHG (ours)	90.0 ± 1.1	97.4 ± 1.3

Additionally, with its ability to simultaneously aggregate information from multiple nodes, the hypergraph structure enables better capture of community structure information during the learning process. For example, on the NTU dataset, the HCHG model achieved an accuracy of 90%, a significant improvement compared to other models like HC-GNN, which achieved 85%.

The HCHG model demonstrates strong generalization capabilities when handling multimodal data, effectively adapting to various datasets. Its performance across diverse datasets, including Cora, Pubmed, Citeseer, Zoo, NTU, and ModelNet40, surpasses other models.

By integrating hypergraph structures and hierarchical information, the HCHG model can more effectively capture complex relationships surrounding nodes. In ablation experiments, the HCHG model consistently yielded favorable results across different numbers of node neighbors, confirming its adaptability to graphs of varying scales.

In summary, by combining hypergraph structures and hierarchical information, the HCHG model efficiently captures the intricate associations among multimodal data and achieves outstanding performance in node classification across various datasets.

Finally, in the experiments, HC-GNN, which is constructed using GCN, was compared to HCHG to investigate the impact of the hypergraph structure on the experimental results. It was found that the effect varies for different datasets. Not all graph community structures are suitable for the hypergraph structure, and this needs to be considered in different problem scenarios. Table 5 presents the ablation experiments on the multiview data, comparing the classification results for the same nodes under various scenarios of single-view and multiview. The results from the three models in the experiment demonstrated that multiview data carries more information, which is beneficial for classification. Finally, a comparison was made on the number of neighboring nodes using the ModelNet40 and NTU2012 datasets. As shown in Figure 6, the different performances of the HC-GNN and the proposed HCHG model with varying numbers of neighboring nodes are displayed. It can be observed that the HCHG model achieves the best results even with fewer neighboring nodes, indicating that the hypergraph structure can aggregate neighbor information more quickly and accurately.

Table 5. Average Test Accuracy (%) ± Standard Deviation for Multi-view Data vs. Single-view Data Comparison.

View	HC-GNN [27]	HGNN [35]	HJRL [29]	HCHG (Ours)
NTU (mvcnn)	73.7 ± 1.1	69.8 ± 0.9	69.6 ± 1.2	70.7 ± 1.0
NTU (gvcnn)	69.7 ± 0.8	79.5 ± 0.7	80.3 ± 1.6	85.7 ± 1.2
NTU (mvc. and gvc.)	83.3 ± 1.0	84.2 ± 1.4	86.1 ± 1.3	90.0 ± 0.8
ModelNet40 (mvcnn)	98.1 ± 1.3	90.8 ± 1.0	92.3 ± 1.7	93.9 ± 1.1
ModelNet40 (gvcnn)	97.3 ± 1.1	92.8 ± 0.9	90.5 ± 1.4	81.5 ± 1.5
ModelNet40 (mvc. and gvc.)	98.1 ± 0.8	96.7 ± 1.2	95.8 ± 1.4	97.4 ± 1.3

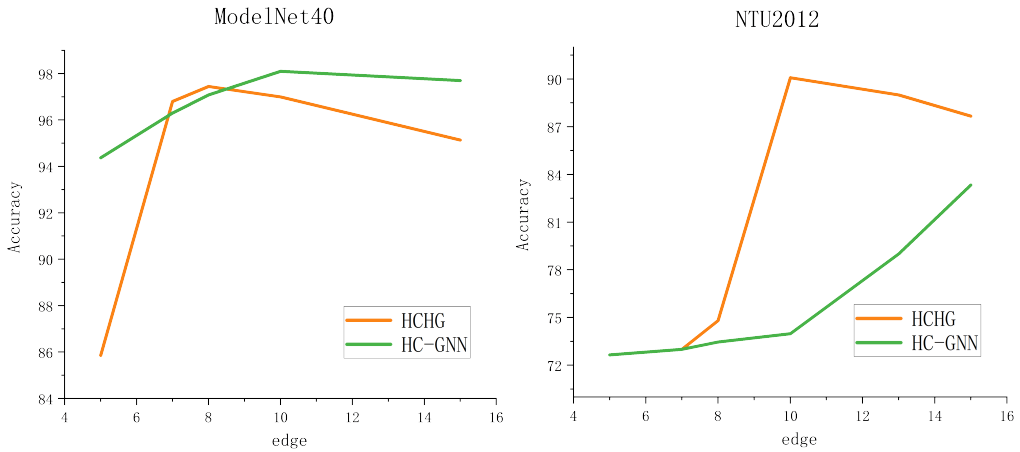

Figure 6. The number of neighboring node points affects the classification.

5.3. Visualization

The core dataset was visualized to compare the learning abilities of graph-based and hypergraph-based methods intuitively. The t-SNE method was used to visualize the output of the last layer convolution. The results are shown in Figure 7. It can be seen from the results that compared to the graph-based method; the hypergraph-based HCHG method produces recognizable clusters, which qualitatively validates the effectiveness of the proposed method.

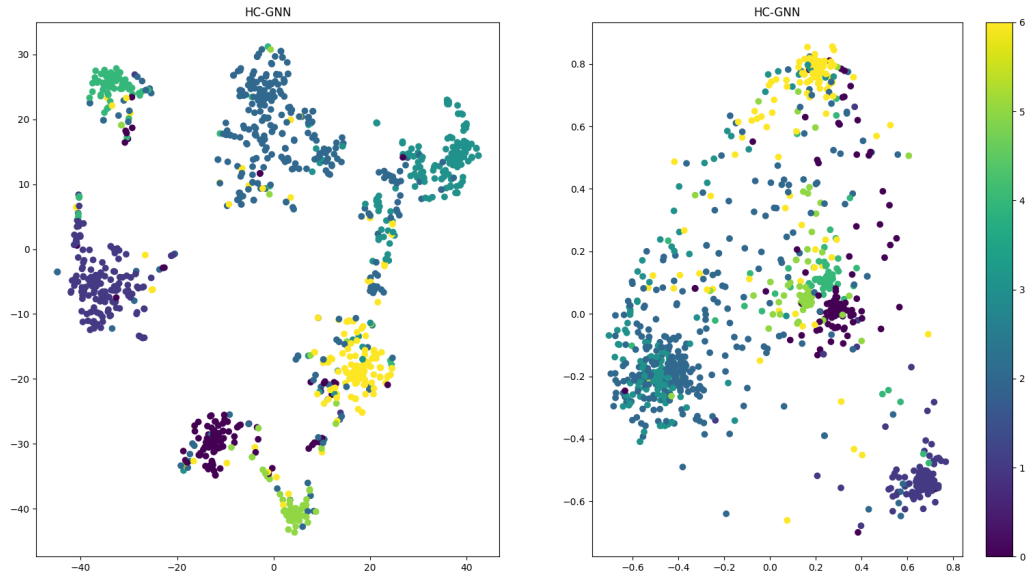

Figure 7. *Cont.*

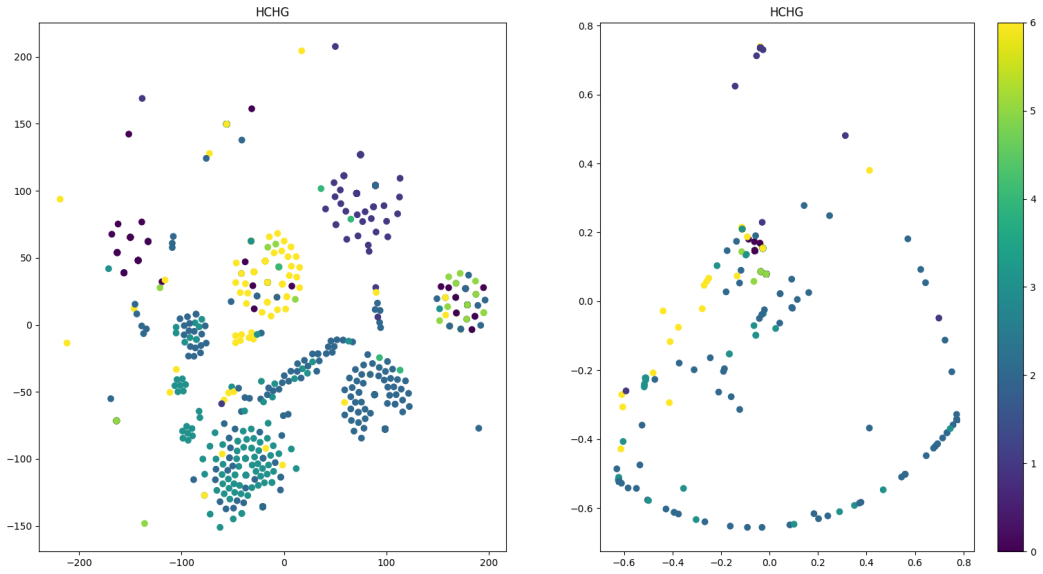

Figure 7. Visualization of the clustering results.

6. Conclusions

This paper proposes a novel Hierarchical message-passing Hypergraph Convolutional (HCHG) model that combines hypergraphs and hierarchical message-passing using a layered community detection algorithm. The HCHG model constructs a hierarchical structure of hypergraph neural networks and performs layered message-passing to handle multi-view data. The model structure of HCHG enables nodes to capture information-rich interactions from distant nodes effectively. Extensive experiments are conducted on five datasets, and the results are analyzed. The experiments demonstrate that HCHG performs excellently in graph-structured dataset classification and 3D model classification tasks. HCHG allows for different choices and customized designs of hierarchical structures, making it easily applicable to various task-specific data. In the future, our goal is to optimize the learning of hypergraph hierarchical structures further and extend the framework to handle complex multimodal data in real-life scenarios.

7. Limitations and Future Work

Despite the impressive performance of the HCHG model on multiple datasets, there are still some limitations. Future work will focus on improving the model's computational efficiency, particularly its scalability in complex networks, and developing more efficient algorithms to reduce computational complexity. Additionally, we plan to optimize the model's ability to handle complex social hierarchies and nested structural networks. Finally, we will conduct an in-depth analysis of how different community detection methods impact the generation of hierarchical structures, exploring their applicability and limitations in real-world scenarios. Overall, future research will further enhance the performance and broad applicability of the HCHG model.

Author Contributions: Conceptualization, W.X. and F.X.; methodology, F.X. and W.X.; software, F.X. and W.X.; validation, F.X. and L.S.; formal analysis, Z.F.; resources, F.X.; writing—original draft preparation, F.X. and Z.F.; supervision, Z.F.; funding acquisition, F.X., L.S. and Z.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Quzhou City Science and Technology Plan Project (2023K263, 2023K265, 2023K045), the General Research Project of the Zhejiang Provincial Department of Education (2023) (Y202353440, Y202353289), and Quzhou Vocational and Technical College university-level scientific research project (QZYZ2305-2023).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Min, S.; Gao, Z.; Peng, J.; Wang, L.; Qin, K.; Fang, B. STGSN—A spatial–temporal graph neural network framework for time-evolving social networks. *Knowl.-Based Syst.* **2021**, *214*, 106746. [CrossRef]
2. Dhelim, S.; Aung, N.; Ning, H. Mining user interest based on personality-aware hybrid filtering in social networks. *Knowl.-Based Syst.* **2020**, *206*, 106227. [CrossRef]
3. Leahy, J.; Jabari, S. Enhancing Aerial Camera-LiDAR Registration through Combined LiDAR Feature Layers and Graph Neural Networks. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2024**, *48*, 25–31. [CrossRef]
4. Yuan, W.; Yuan, X.; Fan, Z.; Guo, Z.; Shi, X.; Gong, J.; Shibasaki, R. Graph neural network based multi-feature fusion for building change detection. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2021**, *43*, 377–382. [CrossRef]
5. Shi, W.; Rajkumar, R. Point-gnn: Graph neural network for 3d object detection in a point cloud. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1711–1719.
6. Meraz, M.; Ansari, M.A.; Javed, M.; Chakraborty, P. DC-GNN: Drop channel graph neural network for object classification and part segmentation in the point cloud. *Int. J. Multimed. Inf. Retr.* **2022**, *11*, 123–133. [CrossRef]
7. Wu, Y.; Dai, H.N.; Tang, H. Graph neural networks for anomaly detection in industrial Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 9214–9231. [CrossRef]
8. Hamilton, W.; Ying, Z.; Leskovec, J. Inductive representation learning on large graphs. In Proceedings of the International Conference on Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; Volume 30, pp. 1025–1035.
9. Huang, K.; Xiao, C.; Glass, L.M.; Zitnik, M.; Sun, J. SkipGNN: Predicting molecular interactions with skip-graph networks. *Sci. Rep.* **2020**, *10*, 21092. [CrossRef] [PubMed]
10. Zitnik, M.; Agrawal, M.; Leskovec, J. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* **2018**, *34*, 457–466. [CrossRef]
11. Zhang, M.; Chen, Y. Link prediction based on graph neural networks. In Proceedings of the 2018 Annual Conference on Neural Information Processing Systems (NeurIPS), Montréal, QC, Canada, 3–8 December 2018; pp. 5171–5181.
12. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
13. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph attention networks. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
14. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How powerful are graph neural networks? In Proceedings of the 2019 International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019.
15. Min, Y.; Wenkel, F.; Wolf, G. Scattering GCN: Overcoming oversmoothness in graph convolutional networks. In Proceedings of the 2020 Annual Conference on Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 10–16 December 2020.
16. Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 701–710.
17. Zhou, D.; Huang, J.; Schölkopf, B. Learning with hypergraphs: Clustering, classification, and embedding. In Proceedings of the Advances in Neural Information Processing Systems, Kelowna, BC Canada, 4–7 December 2006; Volume 19, pp. 1601–1608.
18. Feng, Y.; You, H.; Zhang, Z.; Ji, R.; Gao, Y. Hypergraph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 3558–3565.
19. Ranjan, E.; Sanyal, S.; Talukdar, P.P. ASAP: Adaptive structure aware pooling for learning hierarchical graph representations. In Proceedings of the 2020 AAAI Conference on Artificial Intelligence (AAAI), New York, NY, USA, 7–12 February 2020; pp. 5470–5477.
20. Gao, H.; Ji, S. Graph U-Nets. In Proceedings of the 2019 International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019.
21. Ying, R.; You, J.; Morris, C.; Ren, X.; Hamilton, W.L.; Leskovec, J. Hierarchical graph representation learning with differentiable pooling. In Proceedings of the 2018 Annual Conference on Neural Information Processing Systems (NeurIPS), Montréal, QC, Canada, 3–8 December 2018; pp. 4805–4815.
22. Bai, S.; Zhang, F.; Torr, P.H. Hypergraph convolution and hypergraph attention. *Pattern Recognit.* **2021**, *110*, 107637. [CrossRef]
23. Jiang, J.; Wei, Y.; Feng, Y.; Cao, J.; Gao, Y. Dynamic hypergraph neural networks. In Proceedings of the IJCAI, Macao, China, 10–16 August 2019; pp. 2635–2641.
24. Yadati, N.; Nimishakavi, M.; Yadav, P.; Nitin, V.; Louis, A.; Talukdar, P. Hypergcn: A new method for training graph convolutional networks on hypergraphs. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
25. Arya, D.; Gupta, D.K.; Rudinac, S.; Worring, M. Hypersage: Generalizing inductive representation learning on hypergraphs. *arXiv* **2020**, arXiv:2010.04558.

26. Ye, Z.; Zhao, H.; Zhang, K.; Zhu, Y.; Xiao, Y. Tri-party deep network representation learning using inductive matrix completion. *J. Cent. South Univ.* **2019**, *26*, 2746–2758. [CrossRef]
27. Zhong, Z.; Li, C.T.; Pang, J. Hierarchical message-passing graph neural networks. *Data Min. Knowl. Discov.* **2023**, *37*, 381–408. [CrossRef]
28. Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; Eliassi-Rad, T. Collective classification in network data. *AI Mag.* **2008**, *29*, 93–93. [CrossRef]
29. Namata, G.; London, B.; Getoor, L.; Huang, B. Query-driven active surveying for collective classification. In Proceedings of the 2012 International Workshop on Mining and Learning with Graphs, Edinburgh, UK, 1 July 2012; p. 8.
30. Chen, J.; Ma, T.; Xiao, C. FastGCN: Fast learning with graph convolutional networks via importance sampling. *arXiv* **2018**, arXiv:1801.10247.
31. Yang, C.; Wang, R.; Yao, S.; Abdelzaher, T. Hypergraph learning with line expansion. *arXiv* **2020**, arXiv:2005.04843.
32. Sunil, K.M. Feature Selection: Key to Enhance Node Classification with Graph Neural Networks). *CAAI Trans. Intell. Technol.* **2023**, *8*, 14–28. Available online: https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/cit2.12166 (accessed on 29 October 2024).
33. Yan, Y.; Chen, Y.; Wang, S.; Wu, H.; Cai, R. Hypergraph Joint Representation Learning for Hypervertices and Hyperedges via Cross Expansion. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 26–27 February 2024; Volume 38, pp. 9232–9240.
34. Li, M.; Chen, S.; Zhang, Y.; Tsang, I.W. Graph cross networks with vertex infomax pooling. In Proceedings of the 2020 Annual Conference on Neural Information Processing Systems (NeurIPS), Virtual, 6–12 December 2020.
35. Gao, Y.; Feng, Y.; Ji, S.; Ji, R. HGNN+: General Hypergraph Neural Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 3181–3199. [CrossRef] [PubMed]