Pattern Recognition 168 (2025) 111775

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A novel hypergraph neural network combining multi-view learning with density awareness

Jianpeng Liao [a], Jun Yan [b], Qian Tao [a,c,*], Enze Zhang [a], Yanchao Zhang [a]

[a] School of Software, South China University of Technology, Guangzhou, Guangdong, 510006, China
[b] Concordia Institute for Information Systems Engineering, Concordia University, Montréal, Québec, H3G 1M8, Canada
[c] Pazhou Lab, Guangzhou, Guangdong, 510335, China

ARTICLE INFO

Keywords:
Semi-supervised node classification
Hypergraph neural networks
Graph neural networks
Hypergraph learning
Density-aware attention

ABSTRACT

Graph-based semi-supervised node classification has become a vital area in pattern recognition, with numerous applications of substantial research importance. However, existing approaches typically rely solely on either the intrinsic graph structure or artificially constructed graphs, potentially leading to inaccuracies in capturing the true data correlations, thus diminishing performance in downstream graph neural networks. Additionally, while current methods predominantly utilize explicit graph structures, there exists untapped potential in leveraging implicit information such as density for enhanced performance. To address these limitations, this paper introduces the Multi-view Density-aware Hypergraph Neural Network (MD-HGNN), a novel dual connection model that simultaneously integrates hypergraph structure and representation learning in a unified architecture. The MD-HGNN begins by utilizing a multi-view hypergraph learning (MVHL) network to explore the optimal hypergraph structure from multiple views, constrained by a consistency loss proposed to improve its generalization. Subsequently, it employs a density-aware hypergraph attention (DAHAT) network to explore the high-order semantic correlation among data points based on the density-aware attention mechanism. Extensive experiments across diverse benchmark datasets validate the efficacy of the proposed MD-HGNN approach, showcasing its superior performance in semi-supervised node classification tasks.[1]

1. Introduction

In the recent years, graph neural networks (GNNs) have been widely studied because of their ability to effectively deal with non-Euclidean data and achieve amazing performance. Researchers have applied GNNs in various machine learning tasks, including computer vision [1], recommendation systems [2], link prediction [3] and others.

Graph-based learning has been shown to be a highly effective approach for semi-supervised deep learning, a family of machine learning that uses both labeled and unlabeled data to improve the generalization ability of deep neural networks. A key challenge in graph-based pattern recognition is how to exploit the correlation between labeled and unlabeled samples and the implicit information among data. As a solution, graph-based semi-supervised node classification can exploit the connectivity relationship between small amounts of labeled samples and a relatively large number of unlabeled samples to improve the performance of classification. Such methods have seen success in various applications, such as misinformation detection [4], scientific paper topic assignment [5,6], credit card fraud detection [7], and hyperspectral image classification [8].

Currently, many graph-based semi-supervised node classifiers have been proposed [9]. However, most of these methods focused only on pairwise connections among data on the graph structure. The data correlation in real practice could be beyond pairwise relationships and even far more complicated. Under such conditions, only exploring the pairwise connections and modeling it as a graph may lose the high-order semantic correlation among data, especially for complicated data such as image and citation network datasets. The traditional graph structure cannot fully formulate the data correlation, thus limiting the application of GNN models [10]. To tackle this limitation, hypergraph neural networks (HGNNs) have been proposed, introducing hyperedges that can link any number of nodes to improve the learning performance. Compared with the simple graph, the hyperedges in HGNNs allow the latter to more effectively represent the high-order semantic relationship among data [10–12]. This work will also leverage the

* Corresponding author at: School of Software, South China University of Technology, Guangzhou, Guangdong, 510006, China.
E-mail addresses: sejianpengliao@mail.scut.edu.cn (J. Liao), jun.yan@concordia.ca (J. Yan), taoqian@scut.edu.cn (Q. Tao), sefranciszhang@mail.scut.edu.cn (E. Zhang), 202320145364@mail.scut.edu.cn (Y. Zhang).

[1] Our code can be found here: https://github.com/FrancisGrace/MD-HGNN.

https://doi.org/10.1016/j.patcog.2025.111775
Received 30 September 2024; Received in revised form 2 April 2025; Accepted 27 April 2025
Available online 12 May 2025

hypergraph to explore the high-order semantic correlation among data for semi-supervised node classification.

This paper dives into two challenges in graph-based semi-supervised node classification. The first challenge is, we note that the success of GNNs and HGNNs is widely attributed to the graph-structure data inputs, which are typically derived from either intrinsic known structures (e.g., citation networks [13]) or explicit hand-crafted ones (e.g., k-nearest neighbor graphs [14]). However, it cannot be guaranteed that the original intrinsic or artificially established graph is optimal for semi-supervised node classification in the downstream GNNs. The original graph is also usually constructed from the original feature space in which the similarity between samples may not be accurately measured. In other words, the original graph may have some redundant or missing edges. Therefore, it may not accurately reflect the "true" correlation among data, which can be detrimental to the learning process of GNNs. Meanwhile, hand-crafted structures like k-nearest neighbor graphs are mainly based on a single fixed similarity metric (e.g., Euclidean distance), which may fail to accurately measure a potentially diversified set of similarities among all samples. These challenges underscore the necessity for more sophisticated modeling and learning techniques to derive a suitable graph and hypergraph structure.

The second challenging problem is how to exploit the implicit information among data to improve the performance of model. Most existing semi-supervised node classification methods like [5,10,15] only utilize the explicit graph structure information. Some implicit information among data like the density information has been demonstrated to improve significantly for semi-supervised node classification [16], yet it is rarely exploited in depth. Li et al. [16] first utilized density information for graph-based deep semi-supervised visual recognition. However, it still only explores the graph-structure relationships among data, ignoring the high-order semantic correlations.

To tackle these challenges, we propose the Multi-view Density-aware Hypergraph Neural Network (MD-HGNN), a novel GNN model that performs hypergraph structure learning and hypergraph representation learning for graph-based semi-supervised node classification. MD-HGNN contains two main sub-networks: a multi-view hypergraph learning (MVHL) network and a density-aware hypergraph attention (DAHAT) network. The MVHL network overcomes the first challenge by learning the hypergraph structure from multiple views to measure the sample similarity more accurately. By adopting different learnable similarity metrics for each view, MVHL can capture a wider range of relationships between nodes, overcoming the limitations of single-metric approaches like k-nearest neighbors. Moreover, by introducing a consistency loss, MD-HGNN can effectively improve the ability of generalizing the hypergraph learning. The consistency loss not only improves the quality of the learned hypergraph but also mitigates the risks of overfitting to any one specific metric. For the second challenge, the DAHAT network explicitly encodes the density information into the learning process. We define a density rule for hypergraphs and build a density-aware attention mechanism which effectively guides the model in capturing higher-order semantic correlations and learning more accurate hypergraph representation to improve node classification performance. In short, MD-HGNN jointly optimizes the MVHL network and the DAHAT network to learn an optimal hypergraph suitable for downstream graph-based semi-supervised node classification tasks. Meanwhile, based on the suitable hypergraph learned from the MVHL network, the performance of the DAHAT network can be improved simultaneously. The experiments show that integrating these two HGNNs into the proposed architecture can reveal better results on graph-based semi-supervised node classification.

The previous version of this work [17] has been accepted by the 2023 International Joint Conference on Neural Networks. This paper extends it in several significant aspects: First, we enhance the design and validation of imbalanced data, a common challenge in many real-world knowledge and data engineering tasks [18], by theoretically analyzing the advantages of our method on imbalanced learning tasks and validating it on multiple imbalanced datasets. Second, additional ablation studies, parameter comparisons, and visual analysis have also been carried out to present a comprehensive validation of the proposed method.

The main contributions of this paper are summarized as follows:

- We propose MD-HGNN, a novel GNN architecture combining hypergraph structure learning and hypergraph representation learning for semi-supervised node classification.
- We design a new multi-view hypergraph learning network to learn an optimal hypergraph structure suitable for downstream semi-supervised node classification from multiple views with different learned similarity metrics. Furthermore, the study utilizes the explicit density information inherent in hypergraphs to introduce a density-aware hypergraph attention network. By establishing a hypergraph density rule and crafting a density-aware attention mechanism, the proposed approach effectively enhances the performance of hypergraph representation learning.
- We pay more attention to imbalanced datasets, theoretically analyzing the advantages of MD-HGNN on imbalanced datasets, and experiments on multiple imbalanced datasets prove its superiority.
- Extensive experiments have been conducted to demonstrate the effectiveness of the MD-HGNN in semi-supervised node classification tasks. Additionally, an ablation study has been conducted to validate both the multi-view hypergraph learning and the density-aware attention mechanism, further underlining their significance in improving the performance of the proposed approach.

The remainder of the paper is organized as follows. In Section 2, we review the important studies regarding graph-based semi-supervised node classification. Section 3 provides some preliminary definitions and the problem statement. Then, we introduce MD-HGNN in Section 4. Subsequently, in Section 5, we conducted extensive experiments to evaluate the effectiveness of our method and discuss the experimental results. Finally, we conclude the entire paper in Section 6.

2. Related work

This section briefly describes several vital works related to graph-based semi-supervised learning, mainly from two branches: hypergraph neural networks and graph-based semi-supervised node classification.

2.1. Hypergraph neural networks

In real world, data like citation networks, social networks and knowledge graphs can be modeled by simple graph structure. Lots of GNNs have shown great performance in graph representation learning and various downstream tasks [19]. However, the simple graph structure may not fully formulate the high-order data correlation, for which hypergraphs can provide an effective solution. Hypergraph is a more generalized graph structure, in which each hyperedge can link to any number of nodes. Therefore, HGNNs can more effectively mine the high-order semantic correlation among samples than GNNs. Feng et al. [10] proposed a hypergraph neural network (HGNN) for node-edge-node transform through hyperedge convolution operations to better refine the feature representation. Hypergraph attention networks (HGATs) [11] introduced the attention mechanism into HGNNs to encode the high-order data correlation. Jiang et al. [12] integrated dynamic hypergraph construction and convolution modules to propose dynamic HGNNs (DHGNN) to further improve hypergraph representation learning. Recently, many improvements have been proposed, including hypergraph label propagation networks (HLPN) [20], hypergraph convolution and hypergraph attention (HCHA) [21] and hypergraph convolution on the nodes-hyperedges network (HCNH) [22], among others. Yang et al. [23] proposed a new topological mapping for

hypergraphs named the line expansion, transforming the hypergraph into a homogeneous structure without losing any structure information. Tian et al. [24] proposed a dynamic heterogeneous hypergraph network for spatial–temporal activity prediction, which more fully models high-order relationships and heterogeneity and improves the encoding of heterogeneous spatial–temporal activity hyperedges. However, most of these studies focused only on hypergraph representation learning based on the original hypergraph that may not accurately reflect the "true" data correlation and is not optimal for downstream HGNNs, and this motivates the accurate learning of a suitable hypergraph structure to improve the performance of HGNNs.

2.2. Graph-based semi-supervised node classification

Semi-supervised node classification plays an important role in the field of graph-based pattern recognition. Numerous graph-based semi-supervised learning approaches have been proposed to facilitate this task. Klicpera et al. [25] proposed APPNP by introducing a personalized PageRank propagation scheme, achieving graph information propagation in a larger neighborhood. Iscen et al. [26] designed a graph-based transductive label propagation to generate pseudo-labels and combine them with a deep inductive framework. Yet they all neglected the learning of graph structure. Jiang et al. [6] introduced a graph learning module to learn an optimal graph structure that makes GCNs better for semi-supervised learning. In the same way, we introduce a hypergraph learning module in our method. Rong et al. [27] randomly removed a certain number of edges from the input graph to realize data enhancement. Similarly, Tang et al. [28] presented GRAND by designing a random propagation strategy based on the drop node mechanism and introducing consistency regularization to further mitigate the over-smoothing and non-robustness of GNNs. Yet both [27,28] only use a sub-optimal graph structure for semi-supervised node classification. Lin et al. [29] suggested exploring global and local graph structures to improve semi-supervised node classification performance. Lee et al. [30] proposed GraFN to learn discriminative node representations through supervised and unsupervised consistency between two augmented graphs. Yu et al. [31] suggested encoding the label semantics into the learning process of GNNs by modeling it as a virtual center for intra-class nodes and learning the representations of both nodes and labels. Li et al. [32] constructed a cooperative dual-view graph neural network regarding different views as the reasoning processes of two GNN models. Zheng et al. [33] presented the Adaptive Graph Smoothing Network (AGSN) which can reduce inter-class edge weights and produce smoother predictions. Wang et al. [34] proposed a semi-supervised method based on pairwise constraints and subset allocation (PCSA-DEC) by redefining the similarity-based constraint loss. Wu et al. [35] introduced transferable graph autoencoders model which solves cross-network node classification problem by obtaining transferable features between the source and target networks. Most of these methods only utilized the explicit graph structure information, calling for an effective mechanism to explore implicit information in the hypergraphs, such as the density, to improve the performance of semi-supervised node classification. In this paper, we adopt a dual connection model in our MD-HGNN based on the effective combination of two hypergraph neural networks.

3. Definitions

This section describes some notations used in this paper and states our learning problem.

Definition 1 (*Hypergraph*)**.** A hypergraph can be defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ is the set of nodes or vertices with size $n = |\mathcal{V}|$ and $\mathcal{E} = \{e_1, e_2, \dots, e_m\}$ is the set of hyperedges with size $m = |\mathcal{E}|$. Let $X \in \mathbb{R}^{n\times d}$ be a matrix containing d-dimensional data vectors of all nodes, and x_i denotes the ith row of X, as well as the feature vector of node v_i. The structure of a hypergraph can be represented by an incidence matrix $H \in \mathbb{R}^{n\times m}$. If the node v_i is connected by the hyperedge e_k, $H_{i,k} = 1$, otherwise $H_{i,k} = 0$. The labels of nodes can be represented by $Y \in \mathbb{R}^{n\times c}$ and the ith row corresponds to a c-dimensional one-hot label representation of node v_i, where c is the number of classes. The goal of our work is to integrate both hypergraph structure learning and hypergraph representation learning simultaneously and to mine more implicit information among data to promote the performance of graph-based semi-supervised node classification.

Definition 2 (*Semi-Supervised Node Classification*)**.** For graph-based semi-supervised node classification, that is, given a hypergraph $\mathcal{G}$ with the features of nodes X and hyperedges represented by the incidence matrix H and some labels Y_L of nodes $L \subset \mathcal{V}$, we aim to optimize a classification function $Y = f(X, H)$ to predict the labels of the unlabeled nodes $U \subset \mathcal{V} \setminus L$.

4. Methodology

In this section, we provide the architecture of MD-HGNN as shown in Fig. 1 . Firstly, a multi-view hypergraph learning (MVHL) network is adopted to generate a suitable hypergraph structure on multiple views with different similarity metrics. Secondly, a density-aware hypergraph attention (DAHAT) network based on a density-aware attention mechanism is employed to perform hypergraph representation learning for class prediction. Finally, the losses calculated from the output of two sub-networks and used to perform backpropagation are linearly combined to update the parameters of these two modules at the same time. The specific designs are elaborated as follows.

4.1. Multi-view hypergraph learning network

The core concept of hypergraph learning is to learn an optimal hypergraph structure for semi-supervised node classification of the downstream HGNNs. This is achieved by optimizing both hypergraph structure learning and hypergraph representation learning simultaneously. We construct a hypergraph based on the similarity between pairs of samples. In the proposed MD-HGNN, a multi-view hypergraph learning network is employed to adaptively learn a suitable hypergraph. The principle of the proposed MVHL network is metric learning, that is, learning the similarity between samples in an embedding space to construct a new hypergraph based on the learned sample similarity. The MVHL network learns the hypergraph structure from multiple views with different learnable similarity metrics. This allows for a more accurate fitting of the similarities between samples, avoiding the defect that the single fixed distance measure function may not be able to accurately measure the similarity between all samples. The final hypergraph structure can be obtained by merging the hypergraph learned in each view.

To remove the influence of noise and redundant data in the original feature space, we adopt a fully connected layer to map the initial feature matrix X from the original feature space to a low-dimensional embedding space. This operation can be implemented by multiplying a learnable embedding matrix $W^E \in \mathbb{R}^{d\times d'}$, that is,

$$\tilde{X} = XW^E. \tag{1}$$

To accurately measure the similarity between samples, our MVHL network can perform hypergraph learning on multiple views, allowing us to adopt different learnable similarity metrics (e.g. cosine similarity and inner product) on different views. The similarity between pairs of nodes in the kth view can be represented by a matrix $S^{(k)} \in \mathbb{R}^{n\times n}$, and each element can be formulated as

$$S^{(k)}_{i,j} = \mathrm{sim}_k(\tilde{x}_i, \tilde{x}_j). \tag{2}$$

where $\mathrm{sim}_k(\cdot,\cdot)$ can be implemented as different functions in different views.

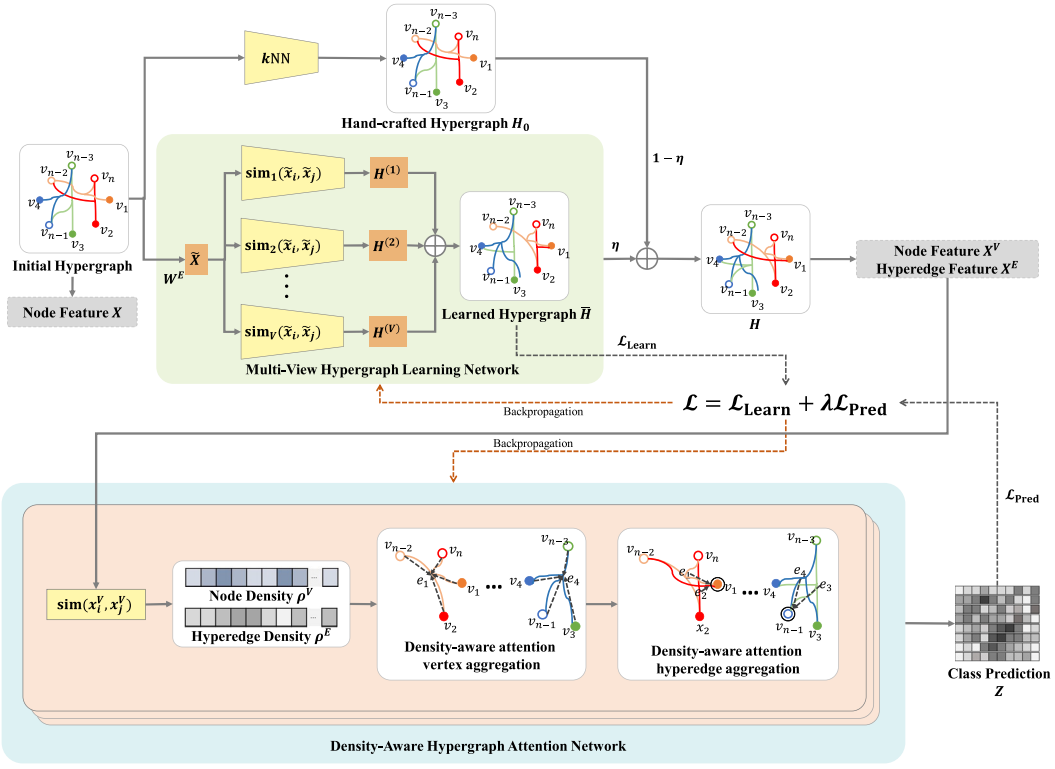


Fig. 1. Overview of the proposed MD-HGNN framework. MD-HGNN contains two main sub-networks: a multi-view hypergraph learning network (the green part), and a density-aware hypergraph attention network (the blue part). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

To decrease the computational cost brought by the aforementioned fully connected graph, we perform sparse sampling for each similarity matrix $S^{(k)}$ by employing a predefined threshold δ_1 to remove lower similarity values, which can be formulated as follows:

$$\tilde{S}_{i,j}^{(k)} = \begin{cases} S_{i,j}^{(k)}, & S_{i,j}^{(k)} \geq \delta_1 \\ 0, & S_{i,j}^{(k)} < \delta_1. \end{cases} \tag{3}$$

Then, we construct the hyperedges and obtain the learned hypergraph incidence matrix $H^{(k)}$ based on the sparse similarity matrix $\tilde{S}^{(k)}$:

$$H^{(k)} = \text{Ctt}(\tilde{S}^{(k)}), \tag{4}$$

The final hypergraph structure is obtained by calculating the mean of hypergraph incidence matrices learned from all views, which can be formulated as follows:

$$\bar{H} = \frac{1}{V}\sum_{i=1}^{V} H^{(i)}, \tag{5}$$

where V is the number of views in the MVHL network.

To constrain the similarity of hypergraph structures learned from the views, we introduce a consistency loss into the MVHL network. With the consistency loss, we can effectively use a large number of unlabeled samples to provide weak supervision that can improve the generalization of the MVHL network. The consistency loss can be defined as the sum of squared L2 distances between each output and its mean, which can be formulated as follows:

$$\mathcal{L}_{\text{con}} = \frac{1}{V}\sum_{i=1}^{V}\left\|H^{(i)} - \bar{H}\right\|_2^2 \tag{6}$$

Considering that the original hypergraph may contain useful information, we merge the learned hypergraph with the original hypergraph and obtain the final incidence matrix H, which can be formulated as follows:

$$H = \eta\bar{H} + (1-\eta)H_0, \tag{7}$$

where η is a trade-off parameter. H_0 can be either a known intrinsic hypergraph or a hand-crafted k-nearest neighbor hypergraph.

The loss function of the hypergraph learning network is

$$\mathcal{L}_{\text{Learn}} = \frac{\alpha}{n^2}\,\text{tr}(\tilde{X}^\top \hat{H}\tilde{X}) + \frac{\beta}{n^2}\,\|H\|_{\text{F}}^2 - \frac{\gamma}{n}1^\top \log \hat{H}1 + \frac{\mu}{n^2}\mathcal{L}_{\text{con}}, \tag{8}$$

where $\hat{H} = D_V^{-1/2} H D_E^{-1} H^\top D_V^{-1/2}$ is the hypergraph Laplacian, in which D_E and D_V are the diagonal matrices of the hyperedge degrees and the node degrees, respectively. α, β, γ, and μ are the hyperparameters whose chosen values will be discussed later. $\text{tr}(\cdot)$ denotes the trace of matrix. $\|\cdot\|_{\text{F}}$ is the Frobenius norm. $\cdot^\top$ denotes transposition. In this loss function, the first term restricts adjacent nodes to having similar features and learning a smooth incidence matrix; the second one constrains learning a sparse hypergraph; the third term penalizes the formation of disconnected hypergraphs; the last one is the consistency loss.

4.2. Density-aware hypergraph attention network

The MD-HGNN employs the density information of the hypergraph structure to introduce the DAHAT network. The density-aware design integrates the density information with attention to structure a density-aware attention mechanism before leveraging hypergraph representation learning through density-aware neighborhood feature aggregation. The detailed design is elaborated as follows.

The input of the DAHAT network is the hypergraph H learned from the MVHL network, including its node feature matrix $X^V \in \mathbb{R}^{n\times d}$ and hyperedge feature matrix $X^E \in \mathbb{R}^{m\times d}$. We perform hypergraph message propagation to obtain the node and the hyperedge feature matrix, which can be expressed as:

$$X^E = D_E^{-1/2} H^\top D_V^{-1/2} X, \tag{9}$$

$$X^V = D_V^{-1/2} H D_E^{-1/2} X^E. \tag{10}$$

where $X \in \mathbb{R}^{n\times d}$ is the initial node feature matrix.

The DAHAT network mainly consists of density-aware attention vertex aggregation and density-aware attention hyperedge aggregation. The former aggregates the information of the connected vertices to the hyperedge to enhance hyperedge features, and the latter aggregates the hyperedge information to generate a new node representation. More details are described as follows.

4.2.1. Density-aware attention vertex aggregation

The density-aware attention vertex aggregation module integrates the density information of each node as a part of attention weight and then performs attention vertex aggregation to enhance hyperedge features. We define the density for each node as the sum of the similarities of neighboring nodes whose similarity with the target node is greater than a predefined threshold. The density of node v_i can be formulated as:

$$\rho_i^V = \sum_{v_j \in \mathcal{N}(v_i)} \begin{cases} \mathrm{sim}\left(x_i^V, x_j^V\right), & \text{if sim}\left(x_i^V, x_j^V\right) > \delta_2 \\ 0, & \text{if sim}\left(x_i^V, x_j^V\right) \le \delta_2, \end{cases} \tag{11}$$

where $\mathcal{N}(v_i)$ denotes the set of neighbors nodes of node v_i. δ_2 is a predefined threshold. $\mathrm{sim}(\cdot)$ is a similarity measure function that can adopt cosine similarity in implementation.

Intuitively, nodes with higher densities tend to have more similar neighbors. In other words, the target node is located in a more densely distributed area. Based on the density-peak assumption, nodes with higher density are closer to the cluster center. Therefore, higher weights for these nodes need to be assigned when performing neighborhood feature aggregation. Traditional attention mechanisms only consider feature similarity, which may aggregate limited information. Fusing the density can effectively avoid this and achieve more accurate attention to neighborhood aggregation.

In the density-aware attention vertex aggregation module, we compute the attention weight of each node relative to the hyperedges it is on. First, we design a trainable linear transformation weight matrix $W \in \mathbb{R}^{d\times d}$ to project the node and hyperedge features into a higher-level. Subsequently, we adopt an attention mechanism $\mathrm{Attention}(\cdot)$ to calculate the weight between node v_i and hyperedge e_k, which can be represented as:

$$a_{i,k} = \mathrm{Attention}(x_i^V W, x_k^E W), \tag{12}$$

The density-aware attention mechanism can be then structured by combining the density information with the attention weight, which is shown in the following:

$$a'_{i,k} = a_{i,k} + \tilde{\rho}_i^V, \tag{13}$$

where $\tilde{\rho}_i^V \in [0, \max(a_V)]$ is the normalized density, and a_V is the collection of attention weight $a_{i,k}$.

$$A_{i,k}^V = \frac{\exp\left(\mathrm{LeakyReLU}\left(\left(x_i^V W \| x_k^E W\right)\alpha_V\right) + \tilde{\rho}_i^V\right)}{\sum_{v_j \in \mathcal{H}(e_k)} \exp\left(\mathrm{LeakyReLU}\left(\left(x_j^V W \| x_k^E W\right)\alpha_V\right) + \tilde{\rho}_j^V\right)} \tag{14}$$

where $\mathcal{H}\left(e_k\right)$ denotes the set of nodes connected by the hyperedge e_k, $\mathrm{LeakyReLU}(\cdot)$ is the leaky rectified linear unit activation function, and $\|$ represents the concatenation operation. This attention means the importance of node v_i in the process of neighborhood feature aggregation to generate hyperedge embedding x_k^E.

After obtaining the density-aware attention matrix $A^V \in \mathbb{R}^{n\times m}$, we utilize this attention coefficient matrix to perform feature aggregation, which is formulated as:

$$\tilde{E} = \sigma((A^V)^\top X^V W) \tag{15}$$

where $\sigma(\cdot)$ is an activation function, which can be exponential linear unit (ELU) in implementation, and $\tilde{E} \in \mathbb{R}^{m\times d}$ is the updated hyperedge representation.

4.2.2. Density-aware attention hyperedge aggregation

This module integrates the density information of each hyperedge as a part of the attention and aggregates the connected hyperedge to enhance the node embeddings. We define the density for each hyperedge as the sum of the density of all nodes connected by this hyperedge, which can be formulated as:

$$\rho_k^E = \sum_{v_j \in \mathcal{H}(e_k)} \rho_j^V \tag{16}$$

Intuitively, if a hyperedge has a higher density, it would be located in a node-dense area, requiring more attention in hyperedge feature aggregation. Accordingly, in the density-aware attention hyperedge aggregation module, we calculate the density-aware attention weights of each hyperedge concerning each node connected by this hyperedge. We employed an attention mechanism similar to the above density-aware attention vertex aggregation module, that is,

$$A_{k,i}^E = \frac{\exp\left(\mathrm{LeakyReLU}\left(\left(x_k^E W \| x_i^V W\right)\alpha_E\right) + \tilde{\rho}_k^E\right)}{\sum_{e_j \in \mathcal{H}(v_i)} \exp\left(\mathrm{LeakyReLU}\left(\left(x_j^E W \| x_i^V W\right)\alpha_E\right) + \tilde{\rho}_j^E\right)} \tag{17}$$

where $\mathcal{H}(v_i)$ represents the set of hyperedges connecting to node v_i. $\alpha_E \in \mathbb{R}^{2d}$ is a weight vector to be trained. And $\tilde{\rho}_k^E$ is the normalized density.

Afterward, we can obtain the density-aware attention matrix $A^E \in \mathbb{R}^{m\times n}$. Finally, we utilize this density-aware attention matrix to aggregate hyperedge features and update the node embedding, which can be formulated as:

$$\tilde{X} = \sigma((A^E)^\top \tilde{E}) \tag{18}$$

where $\tilde{X} \in \mathbb{R}^{n\times d}$ is the updated node embedding.

We combine the two modules described above to form a density-aware hypergraph attention layer shown as follows:

$$\tilde{X} = \mathrm{ELU}\left((A^E)^\top \ \mathrm{ELU}\left((A^V)^\top X^V W\right)\right) \tag{19}$$

where $\mathrm{ELU}(\cdot)$ is the exponential linear unit activation function.

In each density-aware hypergraph attention layer, we first pay a density-aware attention weight to each node and gather the node features to enhance hyperedge features. Then, we assign a density-aware attention weight to each hyperedge and aggregate the connected hyperedge features to generate new node features. This node-hyperedge-node feature transform can efficiently explore high-order semantic correlation among data.

Our design applies the multi-head attention mechanism [15] for the first density-aware hypergraph attention layer to enhance the neighborhood aggregation. The output feature representation of this layer is obtained by concatenating the output features of each head, which can be expressed as:

$$\tilde{X} = \|_{t=1}^{T} \mathrm{ELU}\left((A^{E(t)})^\top \ \mathrm{ELU}((A^{V(t)})^\top X^V W^{(t)})\right) \tag{20}$$

where $\|_{t=1}^{T}$ denotes the concatenation operation, and T is the number of attention heads. For the second density-aware hypergraph attention layer, we adopted one-head attention. The final output of the MD-HGNN is a low-dimensional node feature embedding and the class prediction $Z \in \mathbb{R}^{n\times c}$ can be obtained by applying the softmax function.

The cross-entropy loss is adopted as the optimization function, which can be formulated as follows:

$$\mathcal{L}_{\mathrm{Pred}} = -\sum_{i\in L}\sum_{j=1}^{c} Y_{i,j} \ln Z_{i,j}, \tag{21}$$

where L is the set of labeled samples.

Accordingly, we jointly optimize the MVHL network and the DAHAT network by linearly combining the hypergraph learning loss and the cross-entropy loss, which can be defined as:

$$\mathcal{L} = \mathcal{L}_{\mathrm{Learn}} + \lambda \mathcal{L}_{\mathrm{Pred}}, \tag{22}$$

where λ is a trade-off parameter balancing the two losses.

The entire workflow of the proposed MD-HGNN is summarized in Algorithm 1.

Algorithm 1: The algorithm of MD-HGNN.

Input: Node feature embedding $X \in \mathbb{R}^{n\times d}$, initial hypergraph incidence matrix $H \in \mathbb{R}^{n\times m}$;
Output: The model parameters after training;
Initialize the parameters W^E, W, α_V, α_E in MD-HGNN;
while *not converges* **do**
  for *each view of the hypergraph learning network* **do**
    $S \leftarrow$ Calculate sample similarity matrix using Eq. (2);
    $\tilde{S} \leftarrow$ Sparse sampling according to Eq. (3);
    Construct hyperedges via Eq. (4);
  end
  Calculate the learned $\tilde{H}$ using Eq. (5);
  Calculate H using Eq. (7);
  X^V, $X^E \leftarrow$ Perform hypergraph message propagation according to Eq. (9) and Eq. (10);
  for *each density-aware hypergraph attention layer* **do**
    $\rho_1^V, \rho_2^V, ..., \rho_n^V \leftarrow$ Calculate node density via Eq. (11);
    $A^V \leftarrow$ Calculate density-aware attention using Eq. (14);
    Perform attention vertex aggregation according to Eq. (15);
    $\rho_1^E, \rho_2^E, ..., \rho_m^E \leftarrow$ Calculate hyperedge density via Eq. (16);
    $A^E \leftarrow$ Calculate density-aware attention using Eq. (17);
    Perform attention hyperedge aggregation according to Eq. (18);
  end
  Calculate $\mathcal{L}_{\text{Learn}}$ using Eq. (8);
  Calculate $\mathcal{L}_{\text{Pred}}$ using Eq. (21);
  Calculate $\mathcal{L}$ via Eq. (22);
  Update parameters by performing back-propagation;
end

4.3. Advantages on imbalanced datasets

Traditional hypergraph crafting approaches, such as k-nearest neighbors, assume that the data is uniformly distributed and each class have sufficient number of nodes in the graph. However, in many real-world scenarios, the class distributions can be imbalanced, with some classes having significantly fewer samples than others. The k-nearest neighbors method uses hyperedges with a same weight to connect each node with k other nodes with nearest features regardless of classes with a hyperedge. When the sample size of a minority class is smaller than k, nodes with the minority class will be connected with samples from different classes. As a result, inter-class connections become more frequent, negatively impacting the process of feature aggregation and potentially degrading the performance of model.

In contrast, the multi-view learning process in the MVHL network does not require a fixed number of connected nodes and a fixed edge weight for each hyperedge, reducing the occurrence of inter-class connections and enhancing the structure of the obtained hypergraph. Meanwhile, the density-aware attention mechanism in the DAHAT network assigns higher attention weights to more similar neighbors, or neighbors with higher densities, which improves feature aggregation, particularly for nodes from majority classes.

4.4. Computational complexity analysis

The goal of the MVHL network is to construct a hypergraph based on multi-view learning. First, we need to map the initial feature matrix $X \in \mathbb{R}^{n\times d}$ to a d'-dimensional embedding space, and the time complexity of the multiplication operation is $O(ndd')$. As shown in Algorithm 1, we then calculate the similarity matrices $S^{(k)}$ of V different views based on n^2 pairs of d'-dimensional node feature vectors, so the total cost of the process is $O(Vn^2d')$. Since the calculation of the final hypergraph H in Eqs. (5) and (7) only involves linear operations of matrices, the complexity is just $O(n^2)$. As the number of views V is usually less than 5, the overall complexity of the MVHL network is $O(n^2d')$.

In the DAHAT network, hypergraph message propagation to obtain X^V and X^E is performed first, and the complexities of both Eqs. (9) and (10) are $O(nm(n+m))$. By applying the attention mechanism, the complexity of calculating the density-aware attention matrix A^V is $O(nmd)$, and the complexity of attention vertex aggregation is $O(mn^2)$. Similarly, the complexity of attention hyperedge aggregation is $O(nmd)$.

Based on the analysis before, the total time complexity of the proposed MD-HGNN is $O(nm(n+m))$.

5. Experiments

In this section, we conduct extensive experiments to evaluate the effectiveness of the proposed MD-HGNN on semi-supervised node classification tasks. We first introduce the datasets used in our experiments and the experimental settings. After that, we present the experimental results and discussions. Finally, we provide the ablation study and the parameter analysis.

5.1. Datasets

We evaluate our proposed MD-HGNN on three imbalanced datasets, including Scene15 [36], CIFAR10-IBL [37] and MNIST-IBL [38], and three widely used balanced image datasets, including CIFAR10 [37], MNIST [38], and SVHN [39]. Each dataset is used in a semi-supervised learning setup where only a subset of samples is labeled. We summarize the datasets in our experiments in Table 1 and briefly introduce them as follows.

- **Scene15.** This dataset consists of 4485 RGB images from 15 different scene categories, including both indoor and outdoor scenes; each category contains 200 to 600 samples. Our experiments used all 4485 samples. For each image, we use the 3000-dimensional feature provided in the previous work [40].
- **CIFAR10-IBL.** It is an artificially established imbalanced dataset built from CIFAR10 datasets, which contains 4320 32×32 RGB natural images coming from 10 classes, and each category contains 100 to 800 samples. To represent each image, we used the same 13-layer CNN networks as in [26,41] to extract the features.
- **MNIST-IBL.** This imbalanced dataset is constructed from the MNIST dataset. It has 3820 grayscale images containing 10 classes of hand-written digits, and each category contains 100 to 800 samples. In our experiments, similar to the prior work [6, 29], we use the 784-dimensional feature vectors converted from grayscale to represent each sample.
- **CIFAR10.** It has a training set of 50,000 natural images and a test set of 10,000 images, all from 10 classes. Each image is 32×32 with RGB channels. In our experiments, we use 10,000 images from the independent test set to evaluate our method. Similarly, we used the same 13-layer CNN networks as in [26,41] to extract the features of each image.
- **MNIST.** This dataset contains 10 classes of images of hand-written digits. It has 60,000 training images and 10,000 test images, all in grayscale. We randomly selected 1000 images from each class to obtain 10,000 images for our experiments. Similarly, we use the 784-dimensional feature vectors converted from grayscale to represent the samples.
- **SVHN.** This dataset contains 73,257 training images and 26,032 test images, each of which is a 32×32 RGB image. Similarly to the CIFAR10 dataset, we randomly select 1000 images for each class from the independent test set and get 10,000 images at last for our evaluation. We also use the same 13-layer CNN networks as in [26,41] to extract features to represent each image.

For the imbalanced datasets (Scene15, CIFAR10-IBL, and MNIST-IBL), we randomly selected 250, 500, 750, and 1000 labeled images as

Table 1
Datasets statistics and the extracted features in experiments.

Dataset	Samples	Features	Classes	No. per class	Training	Validating	Testing
Scene15	4485	3000	15	200~600	250 / 500 / 750 / 1000	500	3735 / 3485 / 3235 / 2985
CIFAR10-IBL	4320	128	10	100~800	250 / 500 / 750 / 1000	500	3570 / 3320 / 3070 / 2820
MNIST-IBL	3820	784	10	100~800	250 / 500 / 750 / 1000	500	3070 / 2820 / 2570 / 2320
CIFAR10	10,000	128	10	1000	500 / 1000 / 2000 / 3000	1000	8500 / 8000 / 7000 / 6000
MNIST	10,000	784	10	1000	500 / 1000 / 2000 / 3000	1000	8500 / 8000 / 7000 / 6000
SVHN	10,000	128	10	1000	500 / 1000 / 2000 / 3000	1000	8500 / 8000 / 7000 / 6000

labeled samples. We randomly pick 500 images for validation among the unlabeled samples, and the remaining images are used as test samples. For the balanced datasets CIFAR10, MNIST, and SVHN, we randomly select 50, 100, 200, and 300 labeled images from each of the 10 classes to obtain 500, 1000, 2000, and 3000 images as labeled samples, respectively. For the unlabeled samples, we pick 100 images at random per class and get 1000 images used for validation. The remaining 8500, 8000, 7000, and 6000 images are used as test samples.

5.2. Baseline methods

To highlight the outstanding performance of our method, we compare the MD-HGNN with some representative and state-of-the-art graph-based semi-supervised node classification methods. The details of these baselines are given as follows:

- **GCNs** [5] are the most classic GNNs, which performed label prediction based on graph neighborhood aggregation.
- **GATs** [15] introduced a self-attention mechanism to specify different attention weights to different nodes.
- **GraphSAGE** [42] generated embeddings for new nodes by neighborhood sampling and feature aggregation.
- **APPNP** [25] introduced a personalized PageRank propagation to graph information propagation.
- **HGNN** [10] performed node-edge-node transform through hyperedge convolution operations.
- **DHGNN** [12] employed a dynamic hypergraph construction module to update the hypergraph structure.
- **SGC** [43] removed the nonlinear activation function and collapsed weight matrices from GCNs.
- **DropEdge** [27] randomly removed some edges from the input graph to augment the data.
- **GCNII** [44] extended GCNs by introducing initial residual and identity mapping.
- **GRAND** [28] designed a random propagation strategy based on the drop node mechanism.
- **ElasticGNN** [45] introduced L1 and L2 regularization and provided an elastic message passing scheme.
- **HCNH** [22] performs filtering on both nodes and hyperedges and recovers the original hypergraph with the least information loss.
- **LEGNN** [31] presents a label-enhanced learning framework for GNNs to bring into full play the rich information of labels.

5.3. Experimental setup

For the model architecture of our MD-HGNN, we adopt the MVHL network with two views to learn a hypergraph and employ a two-layer DAHAT network for hypergraph representation learning where the first layer uses the multi-head attention mechanism with two heads. The similarity metrics we adopt in the MVHL network are cosine similarity and inner product. The output dimension of W^E is set to 70 for the Scene15, CIFAR10-IBL, CIFAR10, and SVHN datasets and 128 for the MNIST-IBL and MNIST datasets, respectively. On all the six datasets, the hyperparameters α, β, γ and μ are tuned from $[0.2, 0.8]$ with an interval of 0.1. On all the six datasets, the predefined threshold δ_1 is tuned from $\{0.7, 0.8, 0.9\}$. The predefined threshold δ_2 is set to 0.4 for the Scene15, CIFAR10-IBL, MNIST-IBL, and MNIST datasets and 0.8 for the CIFAR10 and SVHN datasets. The L2-normalization is introduced into each head of the hypergraph learning network and each density-aware hypergraph attention layer. The number of units in the density-aware hypergraph attention hidden layer is 64. We employ the Xavier algorithm [46] for the initialization of W^E, W, α_V and α_E. We adopt Adam optimizer [47] with a learning rate of 0.2, 0.001, 0.005, 0.01, 0.002, and 0.1 for the Scene15, CIFAR10-IBL, MNIST-IBL, CIFAR10, MNIST, and SVHN datasets, respectively. The learning rate decays to half after every 100 epochs. We train MD-HGNN for a maximum of 2000 epochs and stop training if the validation loss does not decrease for 100 consecutive epochs, as suggested in work [5,6]. For a fair comparison, we construct a k-NN graph for all methods with $k = 15$. We retrain all the baseline methods, and all the reported results are averaged over 10 runs.

5.4. Performance analysis

5.4.1. Performance on imbalanced datasets

Table 2 and 3 summarize the classification accuracy and F1-scores (with 500 labeled samples) comparison results on three imbalanced datasets, that is, Scene15, CIFAR10-IBL, and MNIST-IBL. The best results are marked. From these results, we can make several observations as follows.

- On all three imbalanced datasets, MD-HGNN significantly outperforms all baseline methods and achieves a lower standard deviation in most cases simultaneously, indicating that the proposed multi-view learning strategy and density-aware attention mechanism effectively improve the accuracy and stability of node classification even if the distribution in the datasets is imbalanced.
- The classification accuracy of ElasticGNN achieves excellently on all three imbalanced datasets. This could be because the elastic message-passing mechanism of ElasticGNN is evidently advantageous for handling imbalanced data. Despite this, our MD-HGNN still outperforms ElasticGNN, especially on the Scene15 dataset; MD-HGNN achieves at least 1.6% and at most 2.17% improvements.
- The F1-scores of HCNH are prominent among the baseline methods, since HCNH constructs hypergraphs to exploit higher-order relations among the data. Besides, HCNH explicitly conducts convolution operations on both nodes and hyperedges, utilizing the information in the hypergraph adequately. Our MD-HGNN outperforms HCNH by adopting a density-aware attention mechanism in the DAHAT network to enhance the features of both nodes and hyperedges.
- On all three imbalanced datasets, nearly all GNNs and HGNNs baselines are severely suppressed, resulting in poor performance. This is because these baselines do not consider the data imbalance problem and are with the assumption that the data are well balanced, so they do not work well on imbalanced datasets. However, our MD-HGNN still achieves excellent classification accuracy on imbalanced datasets, which also demonstrates the advantages of our MD-HGNN.

Table 2
Classification accuracy (%) on three imbalanced datasets with different numbers of labeled samples.

Datasets	Methods	No. of labels			
		250	500	750	1000
Scene15	GCNs [5]	89.96 ± 1.17	94.02 ± 1.04	94.75 ± 0.96	95.86 ± 0.64
	GATs [15]	97.06 ± 0.44	98.01 ± 0.28	98.26 ± 0.25	98.32 ± 0.18
	GraphSAGE [42]	95.95 ± 0.87	97.74 ± 0.32	97.98 ± 0.24	98.26 ± 0.26
	APPNP [25]	96.44 ± 0.73	97.39 ± 0.32	97.59 ± 0.27	97.89 ± 0.25
	HGNN [10]	90.49 ± 1.47	94.11 ± 0.41	94.52 ± 0.47	95.58 ± 0.52
	DHGNN [12]	94.29 ± 0.51	95.14 ± 0.26	95.42 ± 0.35	95.58 ± 0.29
	SGC [43]	95.34 ± 0.60	96.27 ± 0.56	96.41 ± 0.52	96.89 ± 0.44
	DropEdge [27]	85.04 ± 2.09	91.55 ± 0.63	93.16 ± 0.53	94.16 ± 0.62
	GCNII [44]	96.03 ± 1.58	96.64 ± 1.90	97.10 ± 1.33	97.30 ± 1.60
	GRAND [28]	88.46 ± 0.81	90.41 ± 0.85	91.12 ± 0.67	91.69 ± 0.56
	ElasticGNN [45]	96.46 ± 0.08	96.54 ± 0.16	97.09 ± 0.12	97.25 ± 0.09
	HCNH [22]	97.08 ± 0.49	97.84 ± 0.22	97.83 ± 0.22	98.01 ± 0.27
	$LEGNN_{GCN}$ [31]	95.02 ± 0.96	97.26 ± 0.38	97.90 ± 0.44	98.31 ± 0.30
	MD-HGNN (ours)	**98.55 ± 0.09**	**98.71 ± 0.06**	**98.79 ± 0.07**	**98.85 ± 0.10**
CIFAR10-IBL	GCNs [5]	91.24 ± 0.56	91.94 ± 0.61	92.52 ± 0.37	92.83 ± 0.48
	GATs [15]	93.55 ± 0.39	94.10 ± 0.21	94.09 ± 0.24	94.26 ± 0.16
	GraphSAGE [42]	93.51 ± 0.26	93.89 ± 0.22	94.11 ± 0.42	94.36 ± 0.47
	APPNP [25]	90.81 ± 2.03	91.29 ± 1.36	91.96 ± 0.63	91.92 ± 0.72
	HGNN [10]	90.93 ± 0.70	91.14 ± 0.76	91.82 ± 0.49	91.82 ± 0.49
	DHGNN [12]	94.12 ± 0.28	94.36 ± 0.20	94.17 ± 0.41	94.47 ± 0.39
	SGC [43]	94.19 ± 0.33	94.58 ± 0.21	94.56 ± 0.21	94.60 ± 0.32
	DropEdge [27]	92.13 ± 0.50	92.59 ± 0.48	92.66 ± 0.74	93.10 ± 0.35
	GCNII [44]	90.83 ± 1.56	91.73 ± 0.93	92.33 ± 0.63	92.46 ± 0.92
	GRAND [28]	92.90 ± 1.03	93.50 ± 0.45	93.78 ± 0.55	94.09 ± 0.32
	ElasticGNN [45]	94.63 ± 0.20	94.49 ± 0.12	94.67 ± 0.05	94.70 ± 0.12
	HCNH [22]	93.24 ± 0.85	93.87 ± 0.36	94.14 ± 0.24	94.44 ± 0.19
	$LEGNN_{GCN}$ [31]	93.49 ± 0.62	93.80 ± 0.59	94.09 ± 0.27	94.31 ± 0.34
	MD-HGNN (ours)	**94.68 ± 0.18**	**94.76 ± 0.17**	**94.91 ± 0.14**	**94.82 ± 0.14**
MNIST-IBL	GCNs [5]	84.77 ± 1.11	89.60 ± 1.09	91.12 ± 0.55	91.79 ± 0.74
	GATs [15]	88.47 ± 0.97	90.92 ± 0.70	92.01 ± 0.69	92.66 ± 0.76
	GraphSAGE [42]	88.04 ± 0.88	91.02 ± 0.75	92.30 ± 0.57	92.90 ± 0.73
	APPNP [25]	76.44 ± 1.80	79.14 ± 1.32	78.73 ± 1.32	77.87 ± 1.19
	HGNN [10]	85.62 ± 1.09	89.03 ± 0.63	90.30 ± 0.53	90.61 ± 0.66
	DHGNN [12]	86.41 ± 0.96	87.49 ± 1.20	89.04 ± 1.08	88.69 ± 1.65
	SGC [43]	86.17 ± 1.07	88.68 ± 0.81	89.23 ± 1.04	89.78 ± 0.54
	DropEdge [27]	88.09 ± 0.95	90.34 ± 0.76	91.47 ± 0.70	91.98 ± 0.78
	GCNII [44]	82.29 ± 1.44	83.50 ± 2.02	83.47 ± 0.83	83.94 ± 1.48
	GRAND [28]	74.38 ± 0.69	77.41 ± 0.76	78.11 ± 0.40	78.44 ± 0.51
	ElasticGNN [45]	92.10 ± 0.21	92.78 ± 0.25	92.75 ± 0.26	93.57 ± 0.13
	HCNH [22]	90.56 ± 1.13	92.82 ± 0.50	93.61 ± 0.69	93.65 ± 0.30
	$LEGNN_{GCN}$ [31]	90.22 ± 0.70	92.54 ± 0.53	93.39 ± 0.37	93.92 ± 0.78
	MD-HGNN (ours)	**92.20 ± 0.20**	**93.30 ± 0.29**	**93.86 ± 0.26**	**94.10 ± 0.26**

Table 3
F1-scores (%) on three imbalanced datasets with 500 labeled samples.

Methods	Scene15		CIFAR10-IBL		MNIST-IBL	
	F1-macro	F1-weighted	F1-macro	F1-weighted	F1-macro	F1-weighted
GCN [5]	97.70 ± 0.15	97.80 ± 0.14	91.61 ± 0.97	92.87 ± 0.95	90.65 ± 0.23	92.50 ± 0.20
GAT [15]	97.85 ± 0.14	97.96 ± 0.12	92.34 ± 0.30	93.19 ± 0.25	88.87 ± 0.49	90.97 ± 0.39
SGC [43]	97.74 ± 0.11	97.81 ± 0.10	93.00 ± 0.55	94.08 ± 0.39	89.70 ± 0.28	91.56 ± 0.21
DropEdge [27]	96.14 ± 0.24	96.23 ± 0.29	91.63 ± 0.52	92.80 ± 0.41	87.62 ± 0.84	90.31 ± 0.64
GRAND [28]	96.20 ± 0.28	96.45 ± 0.23	83.12 ± 1.27	86.41 ± 0.68	74.12 ± 1.73	79.09 ± 1.17
HCNH [22]	98.07 ± 0.08	98.12 ± 0.07	92.87 ± 0.10	93.98 ± 0.03	90.92 ± 0.46	92.52 ± 0.29
$LEGNN_{GCN}$ [31]	97.17 ± 0.37	97.31 ± 0.37	93.58 ± 0.40	94.28 ± 0.38	90.84 ± 0.78	92.53 ± 0.49
MD-HGNN(ours)	**98.30 ± 0.17**	**98.38 ± 0.16**	**93.72 ± 0.38**	**94.42 ± 0.27**	**91.20 ± 0.66**	**92.79 ± 0.54**

5.4.2. Performance on balanced datasets

Table 4 summarizes the classification accuracy comparison results on three balanced datasets, including CIFAR10, MNIST, and SVHN, which are widely used in many classification tasks. Similarly, the best results are marked. We can make a few observations from these results.

- Compared to imbalanced data, both MD-HGNN and other baselines can achieve good performance on three balanced datasets with various numbers of labeled samples. At the same time, our method has more significant improvements on CIFAR10 and SVHN datasets, especially in the case of a few labeled samples, such as less than 1000 labels. This demonstrates that by modeling the correlation among data as a hypergraph and performing density-aware attention neighborhood aggregation, MD-HGNN is more accurate on semi-supervised node classification. MD-HGNN performs better than GATs and receives an improvement at most of 0.49%, 2.17%, and 0.57% on the CIFAR10, MNIST, and SVHN datasets, respectively, which further demonstrates the effectiveness of density-aware attention neighborhood aggregation.
- On the CIFAR10 and SVHN datasets, MD-HGNN significantly outperforms all the baseline methods. On the MNIST dataset, MD-HGNN outperforms most baseline methods. Even compared to ElasticGNN [45], MD-HGNN also obtains competitive performance, especially in the case of a few labeled samples ($\leq$1000). This may be because the elastic message passing mechanism

Table 4
Classification accuracy (%) on three balanced datasets with different numbers of labeled samples.

Datasets	Methods	No. of labels			
		500	1000	2000	3000
CIFAR10	GCNs [5]	91.48 ± 0.25	91.70 ± 0.12	92.55 ± 0.13	92.64 ± 0.14
	GATs [15]	93.80 ± 0.13	93.59 ± 0.42	93.90 ± 0.13	93.97 ± 0.09
	GraphSAGE [42]	92.73 ± 0.10	92.49 ± 0.21	92.08 ± 0.11	92.04 ± 0.22
	APPNP [25]	92.46 ± 0.56	92.52 ± 0.32	92.69 ± 0.28	92.97 ± 0.33
	HGNN [10]	90.97 ± 0.41	91.26 ± 0.19	91.35 ± 0.30	91.68 ± 0.11
	DHGNN [12]	93.95 ± 0.13	93.77 ± 0.26	93.88 ± 0.18	93.95 ± 0.13
	SGC [43]	90.64 ± 0.25	92.24 ± 0.17	93.32 ± 0.32	94.19 ± 0.08
	DropEdge [27]	91.99 ± 0.46	92.42 ± 0.34	92.88 ± 0.24	93.04 ± 0.32
	GCNII [44]	92.99 ± 0.26	93.10 ± 0.23	93.06 ± 0.40	93.24 ± 0.33
	GRAND [28]	93.57 ± 0.14	93.74 ± 0.19	93.88 ± 0.19	93.79 ± 0.15
	ElasticGNN [45]	93.81 ± 0.24	94.03 ± 0.09	93.92 ± 0.25	94.15 ± 0.07
	HCNH [22]	93.23 ± 0.20	93.79 ± 0.19	94.03 ± 0.23	94.06 ± 0.15
	$\text{LEGNN}_{\text{GCN}}$ [31]	93.38 ± 0.45	93.65 ± 0.63	93.85 ± 0.35	93.88 ± 0.24
	MD-HGNN (ours)	**93.99 ± 0.14**	**94.08 ± 0.07**	**94.10 ± 0.16**	**94.24 ± 0.17**
MNIST	GCNs [5]	90.37 ± 0.32	90.42 ± 0.39	90.28 ± 0.34	90.30 ± 0.38
	GATs [15]	91.40 ± 0.14	92.44 ± 0.12	92.99 ± 0.20	93.05 ± 0.18
	GraphSAGE [42]	89.74 ± 0.92	90.72 ± 0.60	91.88 ± 0.25	92.27 ± 0.60
	APPNP [25]	86.17 ± 1.14	86.21 ± 1.28	86.70 ± 1.16	87.75 ± 1.16
	HGNN [10]	88.70 ± 0.46	90.26 ± 0.53	91.34 ± 0.45	92.25 ± 0.32
	DHGNN [12]	86.68 ± 2.38	88.64 ± 0.91	89.40 ± 1.46	89.17 ± 1.38
	SGC [43]	90.11 ± 1.17	91.83 ± 0.65	93.21 ± 0.32	94.21 ± 0.32
	DropEdge [27]	88.98 ± 0.85	90.96 ± 0.27	92.21 ± 0.50	93.09 ± 0.43
	GCNII [44]	86.93 ± 1.51	87.67 ± 1.18	87.70 ± 1.23	88.38 ± 1.49
	GRAND [28]	84.66 ± 0.91	86.80 ± 0.88	88.41 ± 0.90	89.33 ± 0.82
	ElasticGNN [45]	93.25 ± 0.17	93.77 ± 0.14	95.09 ± 0.06	94.88 ± 0.21
	HCNH [22]	92.78 ± 0.38	94.14 ± 0.29	94.80 ± 0.41	94.87 ± 0.38
	$\text{LEGNN}_{\text{GCN}}$ [31]	92.08 ± 0.84	93.79 ± 0.42	**95.12 ± 0.30**	**95.49 ± 0.19**
	MD-HGNN (ours)	**93.57 ± 0.29**	**94.43 ± 0.15**	94.56 ± 0.20	94.89 ± 0.15
SVHN	GCNs [5]	95.52 ± 0.12	95.17 ± 0.20	95.54 ± 0.11	95.70 ± 0.11
	GATs [15]	96.15 ± 0.09	96.23 ± 0.10	96.33 ± 0.04	96.38 ± 0.13
	GraphSAGE [42]	96.09 ± 0.11	96.18 ± 0.09	96.12 ± 0.29	95.96 ± 0.08
	APPNP [25]	95.58 ± 0.49	95.72 ± 0.39	95.91 ± 0.21	95.96 ± 0.28
	HGNN [10]	94.20 ± 0.58	94.61 ± 0.53	94.73 ± 0.49	94.73 ± 0.33
	DHGNN [12]	96.61 ± 0.15	96.68 ± 0.18	96.66 ± 0.16	96.67 ± 0.20
	SGC [43]	94.94 ± 0.68	95.77 ± 0.39	96.21 ± 0.17	96.60 ± 0.08
	DropEdge [27]	95.06 ± 0.51	95.27 ± 0.40	95.86 ± 0.31	95.94 ± 0.31
	GCNII [44]	96.16 ± 0.33	96.25 ± 0.21	96.35 ± 0.20	96.30 ± 0.27
	GRAND [28]	96.33 ± 0.26	96.40 ± 0.29	96.35 ± 0.21	96.44 ± 0.16
	ElasticGNN [45]	96.48 ± 0.04	96.49 ± 0.10	96.65 ± 0.04	96.64 ± 0.07
	HCNH [22]	96.15 ± 0.20	96.33 ± 0.19	96.58 ± 0.24	96.69 ± 0.11
	$\text{LEGNN}_{\text{GCN}}$ [31]	96.14 ± 0.27	96.34 ± 0.36	96.32 ± 0.29	96.53 ± 0.39
	MD-HGNN (ours)	**96.67 ± 0.08**	**96.69 ± 0.12**	**96.67 ± 0.11**	**96.70 ± 0.17**

in ElasticGNN can only play its advantages when more labeled samples are available. At the same time, its performance will be limited when there are fewer labels. This also proves the advantage of MD-HGNN in the case of fewer labels.

- MD-HGNN significantly outperforms HGNN with the least margins 2.56% on the CIFAR10 dataset, 2.64% on the MNIST dataset, and 1.94% on the SVHN dataset, which straightforwardly indicates the higher predictive accuracy on semi-supervised node classification of MD-HGNN by performing multi-view hypergraph learning and density-aware attention neighborhood aggregation.

To have a more intuitive comparison, we further visualize the learned feature embeddings of GCNs, GATs, GraphSAGE, APPNP, HGNN, DHGNN, SGC, HCNH, GCNII, $\text{LEGNN}_{\text{GCN}}$, ElasticGNN, and the proposed MD-HGNN, respectively, on the MNIST dataset with 1000 labeled samples, which is shown in Fig. 2 . From Fig. 2, we can observe that for GCNs, HGNN, and DHGNN, the points representing the same classes fail to form distinct clusters, suggesting that these methods do not effectively learn consistent metrics within each class. In contrast, for GATs, APPNP, GCNII, and ElasticGNN, the clusters formed by different classes show varying degrees of overlap, indicating that these methods do not adequately distinguish between different classes. Although the points of GraphSAGE, HCNH, and $\text{LEGNN}_{\text{GCN}}$ do form clusters, the clusters produced by MD-HGNN are notably more compact and well-structured. The data from different classes in MD-HGNN are distributed more clearly and cohesively, further highlighting the superior performance of MD-HGNN in graph representation learning.

5.5. Ablation study

5.5.1. Effectiveness of multi-view hypergraph learning

Ablation studies are conducted to verify the effectiveness of the MVHL network. We first remove the MVHL network from the proposed MD-HGNN and denote it as *MD-HGNN w/o HGL*. That means the *MD-HGNN w/o HGL* only performs hypergraph representation learning based on an original k-NN hypergraph. The proposed version is denoted as *MD-HGNN w/i HGL*. The ablation experiments are conducted on the Scene15, CIFAR10-IBL, MNIST-IBL, CIFAR10, MNIST, and SVHN datasets, and the comparison results between *MD-HGNN w/o HGL* and *MD-HGNN w/i HGL* are shown in Fig. 3.

From these results, we can observe that employing the MVHL network to learn the hypergraph structure can achieve higher classification accuracy than only using the original k-NN hypergraph and receives an improvement at most of 0.51%, 0.45%, and 4.35% on imbalanced datasets Scene15, CIFAR10-IBL and MNIST-IBL, and 0.29%, 1.09% and 0.29% on balanced datasets CIFAR10, MNIST and SVHN, respectively. Furthermore, in most cases, the standard deviation of accuracy is lower when the MVHL network is used than when it is not. This demonstrates that the MVHL network can not only improve the performance of classification but also enhance the stability of training.

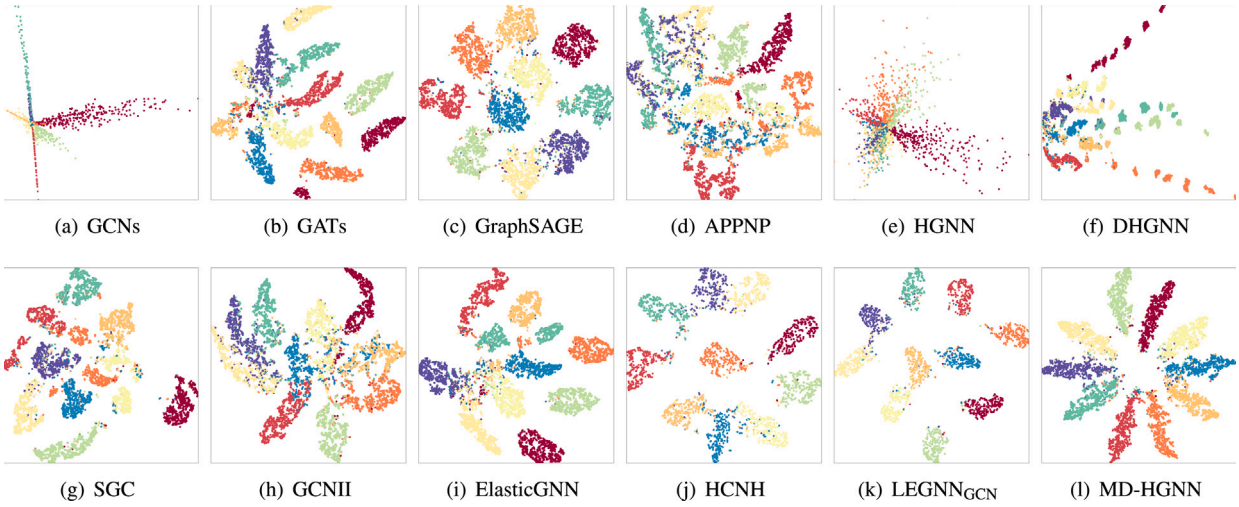


Fig. 2. 2D t-SNE visualizations of the feature embedding of GCNs, GATs, GraphSAGE, APPNP, HGNN, DHGNN, SGC, GCNII, ElasticGNN, HCNH, LEGNN$_{GCN}$, and the proposed MD-HGNN respectively on MNIST dataset with 1000 labeled samples. Each dot in the figure corresponds to one sample, and different classes are marked by different colors. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

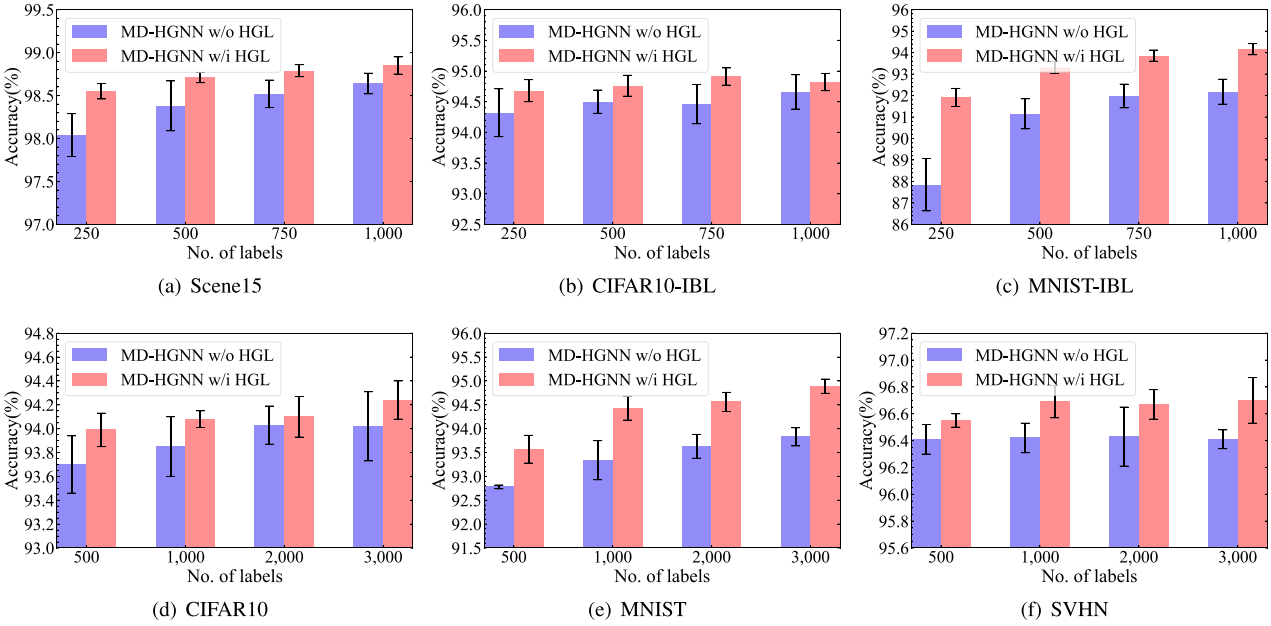


Fig. 3. Effectiveness of multi-view hypergraph learning network. The average accuracy of MD-HGNN w/o HGL and MD-HGNN w/i HGL on (a) Scene15, (b) CIFAR10-IBL, (c) MNIST-IBL, (d) CIFAR10, (e) MNIST, and (f) SVHN datasets.

5.5.2. *Effectiveness of density-aware attention mechanism*

We also conduct an ablation study to evaluate the effectiveness of the proposed density-aware attention mechanism. We remove the density information from the proposed MD-HGNN and only keep the common attention mechanism, which only considers feature similarity, and we denote it as *MD-HGNN w/o Density*. This means that the *MD-HGNN w/o Density* only has two traditional hypergraph attention layers. Correspondingly, the proposed version is denoted as *MD-HGNN w/i Density*.

We conduct ablation experiments on all six datasets and the comparison between *MD-HGNN w/o Density* and *MD-HGNN w/i Density* is shown in Fig. 4. From these results, we can observe that integrating the density-aware attention mechanism can significantly improve the accuracy of hypergraph representation learning. It receives at most 0.44%, 0.86%, and 1.26% improvements on Scene15, CIFAR10-IBL and MNIST-IBL datasets, and 0.31%, 0.49% and 0.28% improvements on CIFAR10, MNIST and SVHN datasets, respectively. Similar to the results of the comparison in the previous section, the standard deviation of accuracy decreases when the DAHAT network is used. Notably, in several cases, the standard deviation of accuracy is as high as 3% or more. This demonstrates the effectiveness of the density-aware attention mechanism on graph-based semi-supervised node classification.

5.6. *Parameter analysis*

5.6.1. *The multi-view hypergraph learning network*

Our MVHL network is provided to learn a suitable hypergraph from multiple views with different similarity metrics. To verify how the MVHL network benefits the MD-HGNN, we conduct some analytical experiments on the Scene15 dataset. We utilize different multi-view hypergraph learning mechanisms with different similarity metrics in

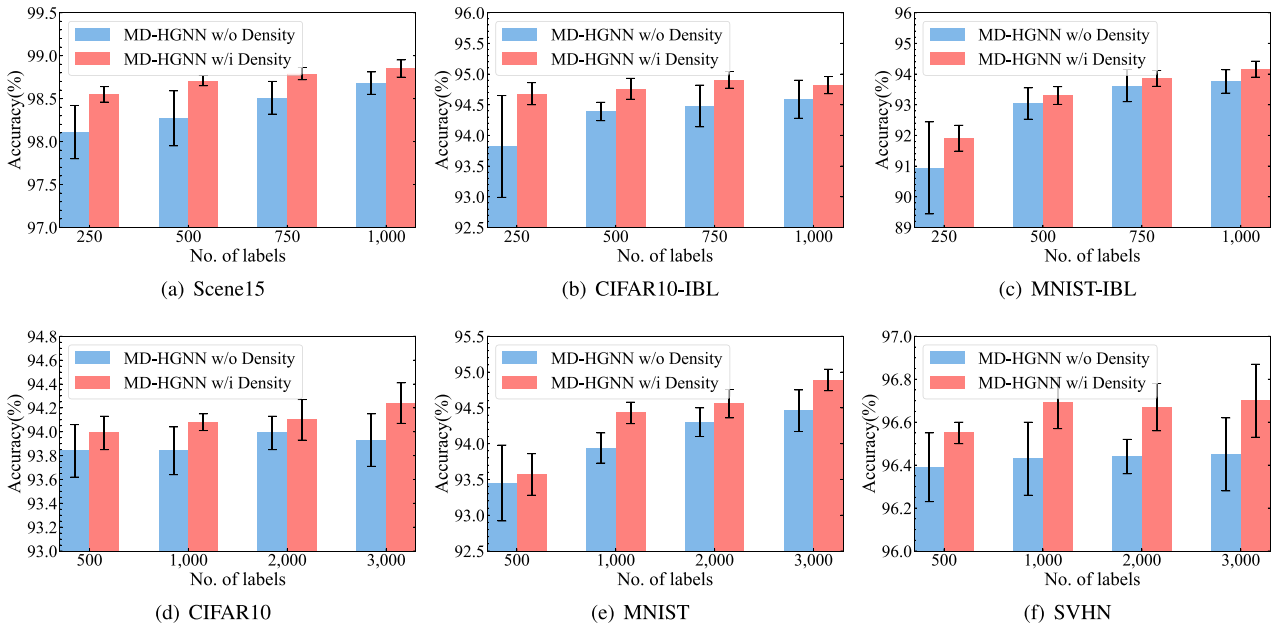


Fig. 4. Effectiveness of density-aware attention weight. The average accuracy of MD-HGNN w/o density and MD-HGNN w/i density on (a) Scene15, (b) CIFAR10-IBL, (c) MNIST-IBL, (d) CIFAR10, (e) MNIST, and (f) SVHN datasets.

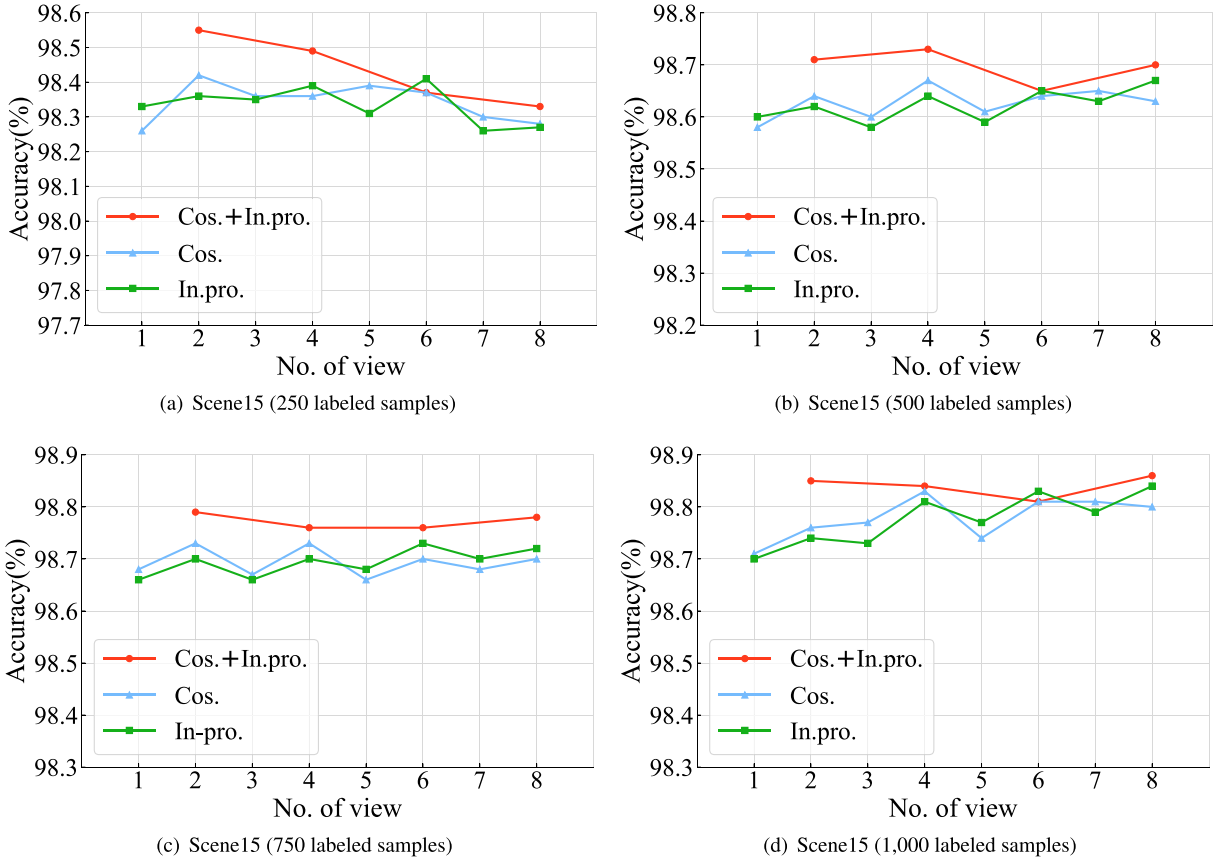
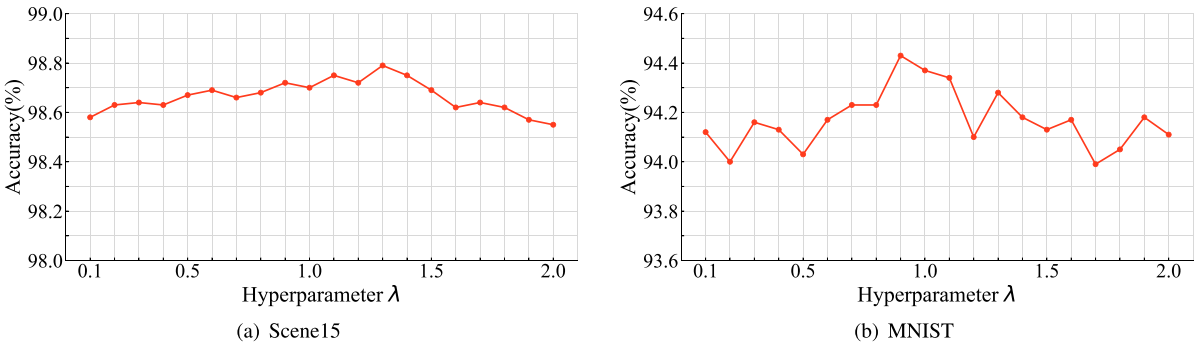


Fig. 5. Results of multi-view hypergraph learning network analysis: the average accuracy of MD-HGNN utilize different multi-view hypergraph learning mechanisms with different similarity metrics on Scene15 dataset with (a) 250, (b) 500, (c) 750, and (d) 1000 labeled samples.

the MVHL network. More specifically, we utilize both cosine similarity and inner product similarity metrics at the same time to instantiate the multi-view hypergraph learning mechanism and denote it as "*Cos.+In.pro.*". The compared baselines adopt only a single similarity metric, and are denoted as "*Cos.*" or "*In-pro.*" accordingly. We modify the number of views used in the MVHL network from 1 to 8, and the experimental results are shown in Fig. 5.

The results indicate that using cosine similarity alone performs comparably to using inner product similarity alone. However, combining both metrics significantly improves the performance of classification,



(a) Scene15 (b) MNIST

Fig. 6. Results of parameter analysis: (a) the parameter sensitivity with respect to λ on Scene15 dataset; (b) the parameter sensitivity with respect to λ on MNIST dataset.

Table 5
Comparison of inference times (Mean±Std seconds) on Scene15 dataset.

Methods	Inference time
HCNH [22]	5.20 ± 0.42
$\text{LEGNN}_{\text{GCN}}$ [31]	0.04 ± 0.01
MD-HGNN (ours)	0.24 ± 0.01

especially when fewer views are used in the hypergraph learning network. When using more views, such as 6 or 8, adopting only one similarity metric can achieve satisfactory accuracy but also bring a large computational overhead. If both cosine similarity and inner product similarity metrics are used simultaneously, we can obtain satisfactory performance even with only two views. In summary, different similarity metrics in our MVHL network to learn a hypergraph structure from multiple views can achieve higher classification performance and lower computing overhead, which once again demonstrates the effectiveness of the proposed MVHL network.

5.6.2. Effect of λ

The proposed MD-HGNN jointly optimizes the MVHL network and the DAHAT network by linearly combining the two losses. $\lambda \geq 0$ is the trade-off parameter of the hypergraph learning loss $\mathcal{L}_{\text{Learn}}$ and the prediction loss $\mathcal{L}_{\text{Pred}}$ in Eq. (22). We conduct a parameter analysis experiment to verify how different values of λ influence the performance of MD-HGNN. For ease of presentation, we only show the results on Scene15 and MNIST datasets when setting λ from 0.1 to 2.0 in Fig. 6. It can be observed that the accuracy variation remains within 0.5% across different values of λ, suggesting that the performance is not highly sensitive to this parameter. Choosing an appropriate value for λ can slightly increase the classification accuracy of MD-HGNN to a certain extent, which is in line with our expectations of jointly optimizing the MVHL network and the DAHAT network. However, setting λ too large will also degrade the performance. In general, the performance of the model is not particularly sensitive to λ, and it is optimized when the value of λ is set to 1.3, 1.1, 0.9, and 0.9 for Scene15, CIFAR10, MNIST, and SVHN datasets, respectively.

5.7. Efficiency analysis

To evaluate the efficiency of the proposed MD-HGNN, we measure the elapsed time for the inference stage on the Scene15 dataset. In this part, we select HCNH [22] and $\text{LEGNN}_{\text{GCN}}$ [31] as the baseline methods and compare the inference time of both methods with MD-HGNN. Table 5 presents the inference times for MD-HGNN and the two chosen baseline models on Scene15 dataset with 250 labeled samples. All reported results are averaged over 10 independent runs.

The results in Table 5 show that HCNH costs the most time due to its complex encoder–decoder architecture. The inference time of $\text{LEGNN}_{\text{GCN}}$ is on par with our method, as both involve graph construction and attention-based message passing. Considering the accuracy of prediction, our method achieves superior performance while consuming a reasonable inference time.

6. Conclusion

In this paper, we propose a Multi-view Density-aware Hypergraph Neural Network (MD-HGNN), which integrates both hypergraph structure learning and hypergraph representation learning simultaneously in a unified network architecture and performs joint optimization for semi-supervised node classification. The MD-HGNN first adopts a multi-view hypergraph learning (MVHL) network to learn a hypergraph structure from multiple views with different similarity metrics. Then, MD-HGNN employs a density-aware hypergraph attention (DAHAT) network based on a density-aware attention mechanism to perform hypergraph representation learning. We have conducted extensive experiments on three imbalanced datasets and three balanced datasets and demonstrated the effectiveness of the MD-HGNN on various semi-supervised node classification tasks. Notably, we pay more attention to the situation that the dataset is imbalanced. Experiments on three imbalanced datasets have reflected the robustness of MD-HGNN towards data imbalance issue. In addition, the conducted ablation study can also prove the validity of the proposed multi-view hypergraph learning mechanism and the density-aware attention mechanism.

Although our MD-HGNN has achieved excellent performance, there are still some limitations to be overcome in the future. More specifically, the time complexity of MD-HGNN is relatively high, since it contains plenty of parameters to be trained. Moreover, MD-HGNN is designed for node-level classification task and unable to perform other types downstream tasks. In our future endeavors, we intend to investigate approaches to reduce the structure of MD-HGNN, and extend MD-HGNN to graph-level tasks, such as graph classification.

CRediT authorship contribution statement

Jianpeng Liao: Writing – original draft, Visualization, Methodology, Investigation, Formal analysis, Conceptualization. **Jun Yan:** Writing – review & editing. **Qian Tao:** Supervision, Project administration. **Enze Zhang:** Writing – review & editing. **Yanchao Zhang:** Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 62276101) and the Natural Science Foundation of Guangdong Province (Grant No. 2024A1515510014).

Data availability

Data will be made available on request.

References

[1] S. Zeng, B. Zhang, J. Gou, Y. Xu, W. Huang, Fast and robust dictionary-based classification for image data, ACM Trans. Knowl. Discov. Data 15 (6) (2021) 97:1–97:22.

[2] M. Zhang, S. Wu, X. Yu, Q. Liu, L. Wang, Dynamic graph neural networks for sequential recommendation, IEEE Trans. Knowl. Data Eng. 35 (5) (2023) 4741–4753.

[3] K. Yao, W. Li, Asymmetric learning for graph neural network based link prediction, ACM Trans. Knowl. Discov. Data 18 (5) (2024) 106:1–106:18.

[4] G. Zhang, S. Zhang, G. Yuan, Bayesian graph local extrema convolution with long-tail strategy for misinformation detection, ACM Trans. Knowl. Discov. Data 18 (4) (2024) 89:1–89:21.

[5] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations, 2017.

[6] B. Jiang, Z. Zhang, D. Lin, J. Tang, B. Luo, Semi-supervised learning with graph learning-convolutional networks, in: 2019 CVF/IEEE Conference on Computer Vision and Pattern Recognition, CVF/IEEE, 2019, pp. 11313–11320.

[7] J. Wang, R. Wen, C. Wu, Y. Huang, J. Xiong, FdGars: Fraudster detection via graph convolutional networks in online app review system, in: Companion of the 2019 World Wide Web Conference, ACM, 2019, pp. 310–316.

[8] F. He, R. Wang, W. Jia, Fast semi-supervised learning with anchor graph for large hyperspectral images, Pattern Recognit. Lett. 130 (2020) 319–326.

[9] Z. Song, X. Yang, Z. Xu, I. King, Graph-based semi-supervised learning: A comprehensive review, IEEE Trans. Neural Netw. Learn. Syst. 34 (11) (2023) 8174–8194.

[10] Y. Feng, H. You, Z. Zhang, R. Ji, Y. Gao, Hypergraph neural networks, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI Press, 2019, pp. 3558–3565.

[11] C. Chen, Z. Cheng, Z. Li, M. Wang, Hypergraph attention networks, in: 19th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, IEEE, 2020, pp. 1560–1565.

[12] J. Jiang, Y. Wei, Y. Feng, J. Cao, Y. Gao, Dynamic hypergraph neural networks, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, ijcai.org, 2019, pp. 2635–2641.

[13] L. Shi, H. Tong, J. Tang, C. Lin, Vegas: Visual influence graph summarization on citation networks, IEEE Trans. Knowl. Data Eng. 27 (12) (2015) 3417–3431.

[14] A. Bryant, K. Cios, RNN-DBSCAN: A density-based clustering algorithm using reverse nearest neighbor density estimates, IEEE Trans. Knowl. Data Eng. 30 (6) (2017) 1109–1121.

[15] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: 6th International Conference on Learning Representations, OpenReview.net, 2018.

[16] S. Li, B. Liu, D. Chen, Q. Chu, L. Yuan, N. Yu, Density-aware graph for deep semi-supervised visual recognition, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVF/IEEE, 2020, pp. 13397–13406.

[17] J. Liao, J. Yan, Q. Tao, DualHGNN: A dual hypergraph neural network for semi-supervised node classification based on multi-view learning and density awareness, in: International Joint Conference on Neural Networks, IJCNN 2023, IEEE, 2023, pp. 1–10.

[18] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.

[19] W. Ju, Z. Fang, Y. Gu, Z. Liu, Q. Long, Z. Qiao, Y. Qin, J. Shen, F. Sun, Z. Xiao, J. Yang, J. Yuan, Y. Zhao, Y. Wang, X. Luo, M. Zhang, A comprehensive survey on deep graph representation learning, Neural Netw. 173 (2024) 106207.

[20] Y. Zhang, N. Wang, Y. Chen, C. Zou, H. Wan, X. Zhao, Y. Gao, Hypergraph label propagation network, in: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI Press, 2020, pp. 6885–6892.

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

[22] H. Wu, M.K. Ng, Hypergraph convolution on Nodes-Hyperedges network for semi-supervised node classification, ACM Trans. Knowl. Discov. Data 16 (4) (2022) 1–19.

[23] C. Yang, R. Wang, S. Yao, T.F. Abdelzaher, Semi-supervised hypergraph node classification on hypergraph line expansion, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, ACM, 2022, pp. 2352–2361.

[24] C. Tian, Z. Zhang, F. Yao, Z. Guo, S. Yan, X. Sun, Tackling higher-order relations and heterogeneity: Dynamic heterogeneous hypergraph network for spatiotemporal activity prediction, Neural Netw. 166 (2023) 70–84.

[25] J. Klicpera, A. Bojchevski, S. Günnemann, Predict then propagate: Graph neural networks meet personalized PageRank, in: 7th International Conference on Learning Representations, OpenReview.net, 2019.

[26] A. Iscen, G. Tolias, Y. Avrithis, O. Chum, Label propagation for deep semi-supervised learning, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVF/IEEE, 2019, pp. 5070–5079.

[27] Y. Rong, W. Huang, T. Xu, J. Huang, DropEdge: Towards deep graph convolutional networks on node classification, in: International Conference on Learning Representations, 2020.

[28] W. Feng, J. Zhang, Y. Dong, Y. Han, H. Luan, Q. Xu, Q. Yang, E. Kharlamov, J. Tang, Graph random neural networks for semi-supervised learning on graphs, in: Advances in Neural Information Processing Systems 2020, vol. 33, 2020, pp. 22092–22103.

[29] G. Lin, X. Kang, K. Liao, F. Zhao, Y. Chen, Deep graph learning for semi-supervised classification, Pattern Recognit. 118 (2021) 108039.

[30] J. Lee, Y. Oh, Y. In, N. Lee, D. Hyun, C. Park, GraFN: Semi-supervised node classification on graph with few labels via non-parametric distribution assignment, in: Proceedings of the Forty-Fifth International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 2243–2248.

[31] L. Yu, L. Sun, B. Du, T. Zhu, W. Lv, Label-enhanced graph neural network for semi-supervised node classification, IEEE Trans. Knowl. Data Eng. (2022) 1–13.

[32] P. Li, Y. Yang, M. Pagnucco, Y. Song, CoGNet: Cooperative graph neural networks, in: Proceedings of the 2022 International Joint Conference on Neural Networks, 2022, pp. 1–8.

[33] R. Zheng, W. Chen, G. Feng, Semi-supervised node classification via adaptive graph smoothing networks, Pattern Recognit. 124 (2022) 108492.

[34] Y. Wang, J. Zou, K. Wang, C. Liu, X. Yuan, Semi-supervised deep embedded clustering with pairwise constraints and subset allocation, Neural Netw. 164 (2023) 310–322.

[35] H. Wu, L. Tian, Y. Wu, J. Zhang, M.K. Ng, J. Long, Transferable graph auto-encoders for cross-network node classification, Pattern Recognit. 150 (2024) 110334.

[36] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: 2006 CVF/IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVF/IEEE, 2006, pp. 2169–2178.

[37] A. Krizhevsky, G. Hinton, et al., Learning Multiple Layers of Features from Tiny Images, Tech. Rep., University of Toronto, 2009.

[38] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[39] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A.Y. Ng, Reading digits in natural images with unsupervised feature learning, in: NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011, 2011.

[40] Z. Jiang, Z. Lin, L.S. Davis, Label consistent K-SVD: learning a discriminative dictionary for recognition, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2651–2664.

[41] A. Tarvainen, H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, in: Annual Conference on Neural Information Processing Systems 2017, 2017, pp. 1195–1204.

[42] W.L. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Annual Conference on Neural Information Processing Systems 2017, 2017, pp. 1024–1034.

[43] F. Wu, A.H.S. Jr., T. Zhang, C. Fifty, T. Yu, K.Q. Weinberger, Simplifying graph convolutional networks, in: Proceedings of the 36th International Conference on Machine Learning, vol. 97, PMLR, 2019, pp. 6861–6871.

[44] M. Chen, Z. Wei, Z. Huang, B. Ding, Y. Li, Simple and deep graph convolutional networks, in: Proceedings of the 37th International Conference on Machine Learning, vol. 119, PMLR, 2020, pp. 1725–1735.

[45] X. Liu, W. Jin, Y. Ma, Y. Li, H. Liu, Y. Wang, M. Yan, J. Tang, Elastic graph neural networks, in: Proceedings of the 38th International Conference on Machine Learning, vol. 139, PMLR, 2021, pp. 6837–6849.

[46] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, vol. 9, JMLR.org, 2010, pp. 249–256.

[47] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: International Conference on Learning Representations, 2015.