A Network Representation Learning Method Fusing Multi-dimensional Classification Information of Nodes

Chenze Huang, Ying Zhong*

Abstract—**The network representation learning fusing multi-dimensional classification information of nodes aims to effectively combine nodes multi-dimensional classification information and network structure information, thus improving the performance of network representation. However, existing methods only consider multi-dimensional classification information as a priori feature to assist the representation learning of the network structure information. They lack coping mechanisms in the case of missing data, and have low robustness in the case of incomplete information. To address these issues, we propose a representation learning method that fuses node multidimensional categorical information. On the one hand, data transformation is performed by a stochastic perturbation strategy to improve the adaptability of the model to incomplete information. On the other hand, the effective semantic information in the incomplete information is further mined by an attribute similarity preservation method based on the homogeneity principle. Experiments show that the method can effectively deal with the problem of incomplete information and improve the performance of node classification and link prediction tasks.**

Index Terms—**Representation learning, complex network, deep learning, semantic information**

I. Introduction

A Large number of complex systems in real life can be modeled as networks for analysis [1]. However, with the emergence of large-scale social networks, traditional network representation methods based on network structure often suffer from computational inefficiency and difficulty in effectively integrating heterogeneous information [2]. Network representation learning [3] is a machine learning approach to learn a low-dimensional vector representation of each node in the network [4]. These vectors can fully preserve the structural and other heterogeneous information of the nodes in the original network. In addition, it can be effectively used as a feature vector for subsequent network analysis tasks [5], such as node classification and link prediction [6]. A network representation learning task fusing node multidimensional classification information, which aims to improve the performance of network representation by effectively combining node multidimensional classification information with network structure information [7].

In real life, node multidimensional classification information [2] is very common. For example, the basic personal information of users in social networks includes gender, age, interests and so on. Multi-dimensionality is reflected in the node contains more than one aspect of attribute information. Classification information means that the information type is classified data, and its value range only contains a limited number of values [2], such as gender information only includes male or female. In the process of social network evolution [8], in addition to the similarity of network structure, nodes with similar attributes are more likely to be connected to each other. These attribute information provides important a priori knowledge in the process of network structure evolution [9], which can provide accurate auxiliary information for network representation learning tasks. Therefore, how to make full use of attribute similarity information while mining network structure similarity is important for improving network representation performance.

Since the node contains various attributes, and each attribute is independent of each other, such as hobbies and locations. The same attribute is filled with different values, for example, the nationality is filled with ”U.S.A” or ”United States of America”. The multi-dimensional classification information of nodes is characterized by independence and inconsistency, which makes it difficult to mine effective semantic information. In addition, it is difficult to collection multidimensional classification information of nodes in social networks, mainly because of user confidentiality and platform privacy protection, so that the multidimensional classification data that can be collected is incomplete. Taking the university datasets [10] on the Facebook platform as an example, statistical analysis reveals that the average missing rate of categorical information is close to 20%, and the maximum missing rate of individual categorical information is 50%.

How to design reasonable fusion representation learning methods in case of incomplete node multidimensional classification information [11]. Make it learn more robust node representations. The following issues need to be considered: (1) How to design an effective fusion method for the incompleteness of multidimensional classification information, so that the model can still learn the robust node representation in the absence of information [12]. (2) For the independence and inconsistency of multidimensional classification information, how to mine the effective semantic information, so that it can assist the network structure information for fusion representation learning [13].

To address the above problems, we propose a network representation learning method that fuses multidimensional

Manuscript received March 21, 2022; revised November 29, 2022.

This work was supported in part by Shenzhen Fundamental Research Program under Grant No. 20210317191843003 and Shaanxi Provincial Key R&D Program under Grant No. 2021ZDLGY05-01.

Chenze Huang is a student of Research and Development Institute of Northwestern Polytechnical University, Shenzhen, Guangdong 518057 China (e-mail: 676401141@qq.com).

Ying Zhong is a lecturer of Research and Development Institute of Northwestern Polytechnical University, Shenzhen, Guangdong 518057 China (Corresponding author, phone: 86-029-88460825; fax: 86-029-88460825; yxy_0713@qq.com).

classification information of nodes, called IMCIN. First, to address the problem of incomplete multidimensional classification information, in the data encoding stage, we propose a random perturbation strategy based on missing proportional weighting to make the model mining more robust features. Secondly, we introduce an advance fusion strategy, and and use the nonlinear characteristics of the deep neural network to extract the knowledge interaction of information, so as to achieve the fusion feature extraction. Finally, on the basis of existing network structure similarity retention methods, to effectively mine the semantic information contained in the multi-dimensional classification information of nodes, we propose an attribute similarity preservation method based on homogeneity constraint, establish a joint optimization objective function, and realize the fusion representation learning of node information. The contributions of this paper are as follows.

- We propose a network representation learning model that integrates node multi-dimensional classification information to improve the performance of network representation learning by using node multi-dimensional classification information, which plays a key role in the evolution of the network.
- To cope with the problem of incomplete information in the node multi-dimensional classification information, we propose a data enhancement strategy based on random perturbation, which makes the model more robust.
- Finally, we conduct a large number of experiments on three real network datasets, and the experimental results show that the proposed algorithm can effectively fuse the multidimensional classification information of nodes and improve the performance of node representation vectors in link prediction and node classification tasks.

II. Related Work

The representation learning method of fusing node multi-dimensional classification information [3] refers to the method of fusing node multi-dimensional classification information and network structure information for representation learning. Multi-dimensionality [12] refers to the multi-dimensional attribute information such as age, gender, occupation, and hobby. Classification [13] refers to the values of a single attribute, which contains a limited number of independent values. For example, gender only includes male and female.

Some algorithms [8] have tried to fuse multi-dimensional classification information of nodes for representation learning. Huang et al. [8] proposed the AANE algorithm, which first constructs the property similarity matrix through the node attribute information, then decomposes the node representation vector through the matrix decomposition, and finally realizes the fusion representation of the two aspects of information by adding the constraint term based on the structure topology in the loss function. In the case of incomplete node multidimensional classification information, the calculation of attribute similarity will inevitably generate errors. Zhang et al. [8] proposed the UPP-SNE algorithm, which firstly uses K-kernel mapping the multi-dimensional classification information of nodes to the high dimensions, and then encodes the network structure information into the multi-dimensional classification information, which realizes the fusion representation.

Wang et al. proposed the NTI_NRL algorithm [13], which performs NRL by preserving the structural similarity and textual similarity of the network, therefore, the representation vector has both structural and textual information. Nguyen et al. [14] proposed the NAWAL algorithm, which embeds nodes into a low-dimensional space, captures structural neighborhoods on the original network by spatial distance, and then leverages generative adversarial deep neural networks to coordinate the space without relying on hand-crafted features or specific domains supervision. NAIE [14] employed an ensemble auto-encoder to learn node representations by reconstructing a combination of smooth structure and attribute information. Li et al. [3] designed two different structural role proximity enhancement strategies in this model, which can effectively capture the network topology and attribute information. Furthermore, the neighbor-modified Skip-Gram model efficiently and seamlessly integrates network topology and attribute information, which can significantly improve the final performance of the algorithm. HANE [15] established a hierarchy of continuous smaller attribute networks from fine to coarse through a fast granulation strategy that integrates topology and node attributes. Although the data noise caused by missing can be alleviated by performing matrix decomposition or K-kernel mapping on the multi-dimensional classification information of nodes [13], however, the attribute feature representation without considering the network structure information is not sufficient. Liao et al. [15] proposed the SNE algorithm, based on the advance fusion strategy, the one-hot encoded node ID encoding vector and the node attribute information encoding vector are concatenated as the input of the neural network. However, the model only aims to maintain the similarity of the network structure as the optimization goal, and its semantic information mining is not enough.

The existing fusion methods [10], [15] are less robust under the condition that the node multi-dimensional classification information is incomplete. The basic idea of the existing fusion methods is to first obtain the node multi-dimensional classification information feature vector by decomposing the matrix singular value for a given one-hot encoding multi-dimensional classification information matrix, and then use it as an auxiliary feature to realize the fusion representation in the representation learning of the node network structure information. However, once the multi-dimensional classification information is incomplete and some attributes are missing, there are large errors in the obtained node multi-dimensional classification information features, which leads to the deterioration of the quality of the obtained representation vector.

Therefore, how to design a reasonable fusion representation learning method to learn a more robust node representation in the case of incomplete multi-dimensional classification information. The focus of this paper is on how to effectively model the interaction between text content information and network structure information, we design a reasonable fusion representation learning method, and realize the fusion representation of these two aspects of information.

III. Problem Definition

Given a network *G*=(*V*, *S*, *C*), which contains multi-dimensional classification information of nodes. The symbol $V = \{v_1, v_2, ..., v_n\}$ represents a set of nodes, and *n* represents the number of nodes in the network. *S* represents a weight matrix, s_{ij} represents the edge weight, s_{ij}>0 denotes the link relationship between nodes v_i and v_j, and s_{ij}=0 denotes no link relationship between nodes v_i and v_j. $C = \bigcup_{i=1}^{n} P_i$ represents the multi-dimensional classification information set of nodes in the network, $P_i = \{p_{i1}, p_{i2}, ..., p_{im}\}$ represents the multi-dimensional classification information of the node v_i, and *m* represents the different types of classification information contained in the node.

For each node $v \in V$, the network representation learning aims to establish a mapping relationship *f*, which maps each node *v* in the network to a low-dimensional vector representation: $f : v \to y, y \in R^d$. Where *d* represents the vector dimension and $d \ll |V|$. In this paper, the node representation vector *y* contains the multi-dimensional classification information of nodes and network structure information.

IV. Proposed method

In this section, we mainly introduce the overall framework of the IMCIN model. Then, the joint training of the representation model and its parameters update method are introduced. Finally, the pseudo-code of the algorithm is given. The IMCIN model framework is shown in Fig. 1. Specifically, it includes three stages: data encoding, feature extraction, and data decoding.

A. Data Encoding

To facilitate feature extraction using deep neural network, firstly, the node network structure information and multi-dimensional classification information are encoded and represented respectively.

B. Network structure information encoding

For any network *G*, its adjacency matrix *S* encodes all network structure information. Therefore, for any node v_i, its corresponding column vector s_i in *S* is taken as the encoding vector of its structure information, which encodes its local neighbor structure information.

C. Multi-dimensional classification information encoding

The data characteristic of the node multi-dimensional classification information is that each attribute is independent of each other, and without direct dependence. Due to limited attribute values, there is usually no obvious order and size distinction. Therefore, we employ one-hot encoding to encode the attribute classification data into binary feature vectors. For example, a node whose gender is male, the representation vector for this attribute is [1, 0]. By ensuring the independence of attribute values in the coding space, the original attribute value disorder characteristics are maintained. Considering the independence between each attribute, for any node v_i, we splice the representation vectors of each attribute as the final node multi-dimensional classification information encoding vector a_i. Let a_{ij} be the encoding vector of the attribute p_{ij} corresponding to the node v_i, and $\oplus$ represents the vector splicing.

$$a_i = a_{i1} \oplus a_{i2} \oplus \cdots \oplus a_{im} \tag{1}$$

To improve the robustness of the representation vector of the model in the case of incomplete information, inspired by the random disturbance data set enhancement strategy in image processing, in the training process, a random perturbation to the input encoding of multi-dimensional classification information of nodes is introduced. and robust representation learning is realized in the case of incomplete information through data set transformation. The basic idea is shown follows.

First, for the multi-dimensional classification information of any node, random perturbation is added to it with probability ε, where ε is the perturbation probability. Secondly, considering that in real life scenarios, different user attribute sensitivities lead to different attribute missing ratios, we design a random perturbation method based on missing ratio weighting. According to the missing ratio perturbation of attributes in the training set, randomly sample the perturbed attribute values. Assuming that the proportion of missing data in the training set is $\{MP_1$,MP_2,...,$MP_m\}$, for any node v_i, its input multi-dimensional classification information p_i, n_s is denoted as the perturbation attribute number, then the perturbation probability distribution of each attribute is as follows.

$$P\left(n_s = k\right) = Mp_k / \sum_{j=1}^{m} Mp_j \tag{2}$$

Finally, in the form of random perturbation, to fit the problem of attributes missing in real life scenes and avoid the interference of random noise on the data, random perturbation is carried out for randomly selected attribute information by setting the attribute value to zero to simulate the situation of data missing.

D. Feature extraction

Considering the positive role played by the multi-dimensional classification information of nodes in the network evolution process, the known network structure information can be used to effectively supplement the multi-dimensional classification information of nodes. The advanced fusion strategy of combining structure encoding and attribute encoding as the input of the neural network is adopted to promote the knowledge interaction of the two aspects of information. Then, the nonlinear feature extraction ability of deep neural network is utilized to mine the complex nonlinear relationship between the two aspects of information and realize nodes fusion feature extraction.

For the construction of input vectors, an advance fusion strategy is adopted to promote knowledge interaction. The node network structure information encoding vector and the node multi-dimensional classification information encoding vector are spliced as the input of the deep neural network, and the complementary and restrictive role of the network structure information on the node multi-dimensional classification information is fully utilized, and the incomplete problem of the node multi-dimensional classification information is alleviated. The advance fusion strategy achieves better performance in a large number of end-to-end deep learning models, and can effectively realize the knowledge interaction of two aspects of information. Define the input vector x_i as follows.

$$x_i = s_i \oplus \tilde{a}_i \tag{3}$$

For the neural network design, we use a tower-like deep feedforward network to extract the abstract feature representation of the data. Universal approximation theorem shows that the deep feedforward network has a very powerful data fitting ability, and also shows good performance in network data feature extraction. Therefore, the node input vector x_i is fed into the deep feedforward network, and the number of hidden layer units is reduced layer by layer to achieve node fusion feature extraction. Suppose the output vector of the hidden layer is $y_i^{(1)}, y_i^{(2)}, ..., y_i^{(K)}$, and the specific definition is as follows.

$$y_i^{(k)} = \mathrm{Re}lu\left(W^{(k)} y_i^{(k-1)} + b^{(k)}\right), k = 1, 2, ..., K \tag{4}$$

where $W^{(k)}$ and $b^{(k)}$ respectively represent the weight matrix and bias vector of the *k*-th layer, and $y_i^{(0)} = x_i$ represents the input vector. Considering the gradient dispersion problem of the Sigmoid function, the line-ar rectification function *Relu* is used as the activation function, i.e *Relu*(*x*)=*max*(0, *x*). The output vector $y_i^{(K)}$ of the last layer of the neural network is used as the finally represent the vector y_i of the node v_i, i.e $y_i = y_i^{(K)}$.

E. Data decoding

The purpose of network representation learning is to preserve the original network information as effectively as possible, and to mine potential semantic information, so as to better serve the subsequent network data mining tasks. Therefore, the node representation vector y obtained by the algorithm in this paper should simultaneously realize the preservation of the similarity information of the network structure and the attribute similarity information. We design the corresponding optimization loss functions respectively. These loss functions are described in detail below.

1) Structural similarity preservation: Similar to the word vector model in natural language processing [16] to predict context words through the word vector of the target node, the neighbors of the target node can be regarded as a kind of "context", and the probability of the target node predicting its neighbor "context" node can be maximized. The similarity of the network structure is preserved. The structure optimization loss function L_{str} is defined as follows.

$$\begin{aligned} L_{str} = -\sum_{v_i \in V} \log p\left(Nr\left(v_i\right) | v_i\right) = \\ -\sum_{v_i \in V} \sum_{v_j \in Nr(v_i)} \log p\left(v_j | v_i\right) \end{aligned} \tag{5}$$

where *Nr*(v_i) represents the set of neighbors of the node v_i. The structural similarity measurement function $p(v_j|v_i)$ is defined as follows.

$$p\left(v_j | v_i\right) = \frac{\exp\left(y_j^{T\prime} y_i\right)}{\sum_{k=1}^{|V|} \exp\left(y_j^{T\prime} y_i\right)} \tag{6}$$

where y_i and y_i' respectively represent the representation vector of the node v_i and the representation vector of the "context" node (auxiliary parameters). Considering that the calculation of *p*($v_j|v_i$) needs to traverse all network nodes, to improve the computational efficiency, we use the negative sampling strategy proposed in [17]. According to the suggestions of literature [15], the number of negative samples is 5, and *degree*(*v*) is the node degree, then the sampling probability distribution of negative samples is as follows.

$$P_n\left(v\right) \propto \left[degree\left(v\right) / \sum_{v \in V} degree\left(v\right)\right]^{3/4} \tag{7}$$

2) Attribute similarity preservation: To effectively preserve the similarity of the original network attributes, the traditional methods directly take the distance between the cosine similarity value of the vector representation of the nodes and the calculated value of the original attribute as the loss function [14], or based on the idea of self-encoding neural network, take the distance between the reconstructed input data and the input of the original network as the loss function [18]. However, from the perspective of attribute similarity calculations, the inconsistency and incompleteness of node multi-dimensional classification information will cause calculation errors. From the perspective of reconstructed input data, the reconstructed missing data obviously does not meet expectations.

The homogeneity of the network shows that nodes with similar attribute information are prone to have connection relations [14], [16], [19]. Then, it is considered that there is a certain potential similarity between nodes that are known to have a connection relationship. Therefore, we propose an attribute similarity preservation method based on homogeneity constraints. This method can make the nodes with the connection relationship closer in the representation space, and the attribute optimization loss function is defined as follows.

$$L_{attri} = -\sum_{v_i \in V} \sum_{v_j \in Nr(v_i)} s_{ij} \log d\left(v_i, v_j\right) \tag{8}$$

where s_{ij} represents the connection weight between the node pair (v_i, v_j), and *d*(v_i, v_j) is the normalized attribute similarity measurement function of the node in the low-dimensional representation space. The specific definition is as follows.

$$d\left(v_i, v_j\right) = \frac{2}{1 + \exp\left(\left\|y_i - y_j\right\|_2^2\right)} \tag{9}$$

The attribute similarity between nodes is inversely proportional to the distance of the vector representation between nodes. When the distance between the nodes in the low-dimensional space is close, *d*(v_i, v_j) tends to 1. On the contrary, *d*(v_i, v_j) tends to 0. The connection weight s_{ij} between the node pair (v_i, v_j) can be regarded as a penalty coefficient to learn more similar representation vectors for nodes with close connections. By minimizing the attribute optimization loss function in formula (10), the attribute similarity information between network nodes is retained.

F. Joint optimization and parameter training

To effectively retain the original network information and realize the fusion representation of the two aspects of information, the joint training optimization loss function is defined as shown below.

$$L = L_{str} + \lambda L_{attri} \tag{10}$$

where λ is the weight coefficient reserved for attribute similarity. By adjusting different weight ratios, the importance of two aspects of information can be dynamically adjusted to better fit different data scenarios. Let the model parameter set be $\theta = \theta_h \cup \theta_c$, where θ_h is the hidden layer model parameter set, i.e $\theta_h = \left\{w^{(k)}, b^{(k)}; k = 1, 2, ..., K\right\}$. θ_c indicates that the node is used as a "context" node to indicate the auxiliary parameter of the vector, i.e $\theta_c = \{y_i'; i = 1, 2, ..., |V|\}$. The model training process is to minimize the loss function L, and realize the training update of the network parameter set θ through gradient propagation. The most critical step is to find the partial derivative $\partial L/\partial\theta$. The specific derivation process is given below. First, the partial derivative of formula (11) can be obtained.

$$\frac{\partial L}{\partial \theta} = \frac{\partial L_{str}}{\partial \theta} + \lambda \frac{\partial L_{attri}}{\partial \theta} \tag{11}$$

Then, solve the two terms on the right side of formula (12) separately. For $\partial L_{str}/\partial\theta$, to facilitate the derivation, the equivalent transformation of L_{str} in formula (5) is expressed as follows.

$$\begin{aligned} L_{str} = -\sum_{v_i \in V} \sum_{v_j \in Nr(v_i)} \sum_{v_k \in \{v_j\} \cup Ns(v_j)} \\ \{I(v_k|v_j)\log[\delta(y_k' y_i)] + (1 - I(v_k|v_j))\log[1 - \delta(y_k' y_i)]\} \end{aligned} \tag{12}$$

where $I(v_k|v_j)$ is the indicative function, when v_k=v_j, the value of $I(v_k|v_J)$ is 1, otherwise it is 0. To simplify the expression, the cubic cumulative sum term of L_{str} in formula (13) is:

$$\begin{aligned} L_{str}(v_i, v_j, v_k) = -\, & I(v_k|v_j)\log[\delta(y_k' y_i)] + \\ & (1 - I(v_k|v_j))\log[1 - \delta(y_k' y_i)] \end{aligned} \tag{13}$$

For the node as the "context" means the auxiliary parameter y_k' of the vector.

$$\begin{aligned} \frac{\partial L_{str}(v_i, v_j, v_k)}{\partial y_k'} &= -I(v_k|v_j)\log[\delta(y_k' y_i)] + \\ & (1 - I(v_k|v_j))\log[1 - \delta(y_k' y_i)]\, y_i \\ &= \left[\delta\left(y_k^{T\prime} y_i\right) - I(v_k|v_j)\right] y_i \end{aligned} \tag{14}$$

For the hidden layer model parameter θ_h, the chain rule can be obtained.

$$\frac{\partial L_{str}(v_i, v_j, v_k)}{\partial \theta_h} = \frac{\partial L_{str}(v_i, v_j, v_k)}{\partial y_i} \cdot \frac{\partial y_i}{\partial \theta_h} \tag{15}$$

For $\frac{\partial L_{str}(v_i,v_j,v_k)}{\partial y_i}$, similar to formula (15) can be calculated as shown below.

$$\frac{\partial L_{str}(v_i, v_j, v_k)}{\partial y_i} = \left[\delta\left(y_k^{T\prime} y_i\right) - I(v_k|v_j)\right] y_k' \tag{16}$$

For $\frac{\partial y_i}{\partial \theta_h}$, the back-propagation parameter update method of the deep feedforward neural network has been introduced in detail in [11]. Based on back propagation, $\frac{\partial y_i}{\partial W^{(k)}}$ and $\frac{\partial y_i}{\partial b^{(k)}}$ (k=1,2. . .,K-1) can be obtained iteratively.

For $\frac{\partial L_{attri}}{\partial \theta}$, to facilitate the derivation, the equivalent transformation of L_{attri} in formula (9) is expressed as follows.

$$\begin{aligned} & L_{attri} = \\ & -\sum_{v_i \in V} \sum_{v_j \in Nr(v_i)} s_{ij} \left\{\log\left[1 + \exp\left(\|y_i - y_j\|_2^2\right)\right] - \log 2\right\} \end{aligned} \tag{17}$$

The two cumulative terms of L_{str} in formula (18) are as follows.

$$\begin{aligned} & L_{attri}(v_i, v_j) = \\ & s_{ij} \left\{\log\left[1 + \exp\left(\|y_i - y_j\|_2^2\right)\right] - \log 2\right\} \end{aligned} \tag{18}$$

Since $L_{attri}(v_i, v_j)$ is only a function of θ_h, it is only required to solve $\frac{\partial L_{attri}(v_i,v_j)}{\partial \theta_h}$. It can be obtained by the chain rule.

$$\begin{aligned} \frac{\partial L_{attri}(v_i, v_j)}{\partial \theta_h} = & \frac{\partial L_{attri}(v_i, v_j)}{\partial y_i} \cdot \frac{\partial y_i}{\partial \theta_h} + \\ & \frac{\partial L_{attri}(v_i, v_j)}{\partial y_j} \cdot \frac{\partial y_j}{\partial \theta_h} \end{aligned} \tag{19}$$

For $\frac{\partial L_{attri}(v_i,v_j)}{\partial y_i}$,

$$\frac{\partial L_{attri}(v_i, v_j)}{\partial y_i} = \frac{s_{ij}\exp\left(\|y_i - y_j\|_2^2\right)}{1 + \exp\left(\|y_i - y_j\|_2^2\right)} \cdot 2(y_i - y_j) \tag{20}$$

For $\frac{\partial L_{attri}(v_i,v_j)}{\partial y_j}$,

$$\frac{\partial L_{attri}(v_i, v_j)}{\partial y_j} = \frac{s_{ij}\exp\left(\|y_i - y_j\|_2^2\right)}{1 + \exp\left(\|y_i - y_j\|_2^2\right)} \cdot 2(y_j - y_i) \tag{21}$$

For $\frac{\partial y_i}{\partial \theta_h}$ and $\frac{\partial y_j}{\partial \theta_h}$, corresponding parameter updates can be completed based on back propagation, $\frac{\partial y_i}{\partial W^{(k)}}$ and $\frac{\partial y_i}{\partial b^{(k)}}$ (k=1,2,. . .,K-1) can be obtained iteratively. Algorithm 1 describes the flow of IMCIN algorithm.

V. Experiments and Result Analysis

To verify whether the IMCIN model can obtain an effective fusion feature representation under the condition of incomplete information, through link prediction and node classification tasks (as shown in Fig.2), the comparison experiments are conducted on real network datasets with typical benchmark methods to verify the effectiveness of the proposed method. In addition, we also analyze the

Algorithm 1 IMCIN

```
Input: Network G(V, S, C), random perturbation ratio $\varepsilon$, training batch size bs, attribute similarity weight coefficient $\lambda$
Output: Node fusion representation vector matrix $Y$
Randomly sample the edge set to get the test set $E_test$, the validation set $E_validation$ and the training network $G'(V, S', C)$
Initialize hidden parameters $\theta_h$ and $\theta_c$
repeat
    Node set to be sampled $V_c$=$V$=$v_1$,$v_2$,...,$v_N$
    repeat
        $V_t$←Randomly sample a single training set containing bs nodes from $V_c$
        $V_c$←$V_c$-$V_t$
        Feedforward neural network input matrix $X$=$\phi$
        for $v_i$ in $V_t$ do
            $x_i = s_i \oplus \tilde{a}_i$
            $X$ append $x_i$
        end for
        $g \leftarrow \nabla_\theta L\left(f\left(X;\theta\right), G'\right)$
        $\triangle\theta = Adam\left(g\right)$
        $\theta \leftarrow \theta - \triangle\theta$
    until $V_c \neq \oslash$
    Calculate the link prediction index AUC on $E_validation$
until The AUC value on the validation set is less than the average AUC value of the previous 10 times
Feedforward neural network input matrix $X$=$\phi$
for $v_i$ in $V$ do
    $x_i = s_i \oplus a_i$
    $X$ append $x_i$
end for
Calculate neural network output $Y$=$f$($X$,$\theta$)
return $Y$
```

impact of hyperparameter selection on the performance of the proposed method. The model is implemented using Python 3.5, the experimental system platform is Urbanu16.04LTS, Intel Core^{TM} i7-5930K six-core twelve-thread processor, and the memory is 64GB.

A. Datasets

We select three commonly used real network datasets, as shown in TABLE I, which are introduced as follows.

GPLUS. The dataset is derived from the user data of the Google+ social network platform [20]. This dataset is a social network collected by McAuley et al. We extracted a self-centered network of one user, which contains 4,450 nodes and 1,473,709 edges. The node multi-dimensional classification information contains six aspects of attribute information: gender, institution, occupation, surname, region, and univer-sity. In this paper, gender is used as the class label of the node classification task.

OKLAHOMA/UNC. The dataset is derived from Facebook, which is a social relationship dataset collected by Traud et al. [10] in universities: OKLAHOMA and UNC. OKLAHOMA contains 17,425 nodes and 892,528 edges. UNC contains 18,163 and 766,800 edges. The node multi-dimensional classification information contains seven aspects of attribute information: student/teacher status label, gender, major, minor, address, grade, graduation high school. In this paper, status labels are used as class labels for node classification tasks.

DBLP. The dataset is an integrated database system of computer English literature [7]. We built a network from the collaboration of 28,085 authors in ten research fields over a ten-year period. The authors belong to the research field in which they have the most publications.

B. Baselines

We use the following representation learning method as the benchmark method for comparison, and the specific introduction is as follows.

DeepWalk [21]: The method samples network structure information through random walk to generate random walk sequences, and then uses the word vector model to learn network node representations. Which only utilizes network structure information.

TADW [4]: The method combines the node attribute information matrix to perform joint matrix de-composition, and finally obtains the fusion representation vector of the node.

UPP-SNE [5]: This method is based on Deepwalk, which introduces the encoding vector of node at-tribute information into the word vector model to assist the representation learning of network structure information.

SNE [4]: This method utilizes the nonlinear feature extraction ability of deep neural network, takes the encoding vector of node attribute information and the encoding vector of network structure information as the input of the neural network, mines the complex nonlinear relationship between nodes, and obtains the node fusion representation vector.

NTF [13]: NTF adopts the idea of energy dissipation, introduces the concept of energy level to measure the influence of adjacent nodes on the central node, and constructs the influence subgraph of the central node on this basis.

NTI_NRL [13]: This method performs NRL by preserving the structural similarity and textual similarity of the network, therefore, the resulting representation vector has both structural and textual information.

NAWAL [14]: The model first embeds network nodes into a low-dimensional space, captures structural neighborhoods on the original network by spatial distance, and then leverages generative adversarial deep neural networks to coordinate the space without relying on hand-crafted features or specific domains supervision.

NAIE [14]: This method employs an ensemble autoencoder to learn node representations by reconstructing a combination of smooth structure and attribute information.

RolEANE [3]: Two different structural role proximity enhancement strategies are designed in this model, which can effectively capture the network topology and attribute information. Furthermore, the neighbor-modified Skip-Gram model can efficiently and seamlessly integrate network topology and attribute information, which can significantly improve the final performance of the algorithm.

HANE [14]: This method first establishes a hierarchy of continuous smaller attribute networks from fine to coarse through a fast granulation strategy that integrates topology and node attributes. After learning the node representation of the coarsest network using any unsupervised network embedding method, HANE refines the node representation of the hierarchical attribute network from coarse to fine.

IMCNRL_R: In the training process, IMCNRL method without random disturbance is used as a comparison method to verify the different roles of attribute similarity retention method based on homogeneity constraint and data set enhancement strategy based on random disturbance.

C. link prediction

1) Experimental setup: First, 10% of edges were used as the test set, 10% of edges were used as the validation set for hyperparameter adjustment and model convergence judgment, and the remaining 80% of edges were used as the training set. Then we use the training set and the multi-dimensional classification information of all nodes to perform representation learning, take the AUC value [11] on the validation set less than the average of the first 10 times as the stopping condition of model training, and finally count the AUC value on the test set. The above experiments are repeated ten times and the average results are recorded.

In the link prediction experiment, the cosine similarity is used as the score value of unknown edges between nodes, and the AUC index is used to measure the accuracy of the algorithm as a whole. In the calculation of AUC, the method of sampling comparison is used to obtain approximate values. AUC refers to the probability that the score value of an edge in the test set is higher than the score value of a randomly selected edge that does not exist. if the number of times p is independently compared, and the number of comparisons in which the score value of the edge in the test set is greater than the fraction value of the non-existent edge is p', and the number of comparisons with equal score values is p''. AUC is defined as follows.

$$AUC = \frac{p' + 0.5p''}{p} \tag{22}$$

For parameter settings, the representation vector dimension is set to 128, which is consistent with the literature [13]. In addition to representing vector dimensions, the SVD algorithm and TADW algorithm based on matrix decomposition have no other parameters. The Deepwalk algorithm and UPP-SNE algorithm based on random walk are set according to the parameters provided in [13], and the number of random walks of each node is r=40, the length of random walk is l=40, the window size is t=10, and the number of iterations is d=40. The SNE algorithm based on deep neural network is set according to the optimal parameters provided by the original literature. For the newly added dataset GPLUS, the batch training size parameter is bs=64, and the initial learning rate is η= 0.0001. For the IMCIN algorithm proposed in this paper, its parameter settings refer to the idea of [7]. By comparing the link prediction performance on the validation set in a given parameter space, the optimal parameter settings are obtained as shown in TABLE II.

TABLE I: Optimal hyperparameter setting of IMCIN algorithm on different datasets

Datasets	ε	bs	λ
GPLUS	0.5	64	0.4
OKLAHOMA	0.5	128	0.3
UNC	0.5	128	0.3

TABLE II: Link prediction performance AUC values on different datasets

Datasets	GPLUS	OKLAHOMA	UNC
Deepwalk	88.93	93.62	92.87
TADW	90.81	94.96	93.83
IMCNRL_R	92.08	94.58	96.84
UPP-SNE	92.63	95.65	94.68
SNE	93.05	96.09	95.77
HANE	93.1	95.95	96.24
NTF	94.04	96.15	95.98
RolEANE	94.17	96.51	96.01
NTI_NRL	94.85	96.65	96.07
NAIE	94.35	95.71	97.02
NAWAL	90.91	91.74	92.52
IMCIN	95.01	96.89	97.08

2) Result analysis: TABLE III records the AUC values of IMCIN algorithm and other comparison algorithms on three experimental data sets. As can be seen from the experimental results, compared with other benchmark algorithms, the performance of the proposed algorithm is improved by 0.19% to 6.24%, with an average improvement of 2.51%. The fusion algorithm (TADW, UPP-SNE, SNE and IMCNRL) has an average improvement of 3.2% compared with the Deepwalk algorithm which only considers network structure information. The NAWAL algorithm mainly depends on the neighbors of the nodes, and does not consider the attribute information of the nodes, therefore, the performance of NAWAL is significantly lower than IMCIN. Which also proves the significance of the prior information provided by node multidimensional classification information for improving network representation learning performance.

In addition, although NTI_NRL, RolEANE and HANE also consider the fusion of network structure and nodes information, and the results have achieved good results. However, these methods mainly consider the completeness of nodes information, while ignoring the incompleteness of nodes information. Therefore, compared to the NTI_NRL, RolEANE and HANE algorithms, our algorithm has an average improvement of 0.47%, 0.62%, and 0.41%, respectively.

D. Node Classification

1) Experimental setup: The model training hyperparameter settings are consistent with the parameter settings in the link prediction task. The training set contains the remaining 80% of network edges and multi-dimensional classification information of all nodes (excluding class labels). First, the node representation vector is obtained through the representation learning process, and then the nodes are divided into training data set and test data set according to the given supervised training rate. The node representation vector in the training data set is used as the node feature, which is input to the Liblinear toolkit [6]. Supervised training is performed in the provided one-to-many logistic regression classifier, and finally the node classification performance of the test dataset on the trained classification model is counted. The above experiment is repeated 10 times and the average results are recorded.

To reflect the comprehensive performance of the representation vector on the node classification task, the *F1* index is used as the evaluation index [9]. For any class label A, TP(A), FP(A), and FN(A) represent the number of correctly predicted positive class, the number of wrongly predicted negative class, and the number of wrongly predicted positive classes whose prediction result is label A, respectively. C is the set of all label lists. Then *F1* is defined as follows.

$$F_1 = \frac{2 \cdot P_r \cdot R}{P_r + R} \tag{23}$$

where

$$P_r = \frac{\sum_{A \in C} TP\left(A\right)}{\sum_{A \in C} \left(TP\left(A\right) + FP\left(A\right)\right)} \tag{24}$$

$$R = \frac{\sum_{A \in C} TP\left(A\right)}{\sum_{A \in C} \left(TP\left(A\right) + FN\left(A\right)\right)} \tag{25}$$

2) Result analysis: In the node classification task, by setting different proportions of supervised training sets, the node classification performance of each method with different amounts of supervised information is investigated. The average *F1* value of the node classification is shown in TABLE IV-VI.

First, as with the link prediction results, the Deepwalk algorithm uses only the network structure information has the lowest performance, which once again proves the important role of node multi-dimensional classification information for more accurate node representation. Secondly, the proposed algorithm in this paper still has advantages compared with the TADW algorithm and UPP-SNE algorithm, which also consider two aspects of information. In the case of less training data, the node classification performance of the TADW algorithm on the OKLAHOMA and UNC datasets is even lower than that of the Deepwalk algorithm. It is because the TADW algorithm directly performs matrix decomposition and dimension reduction on the TF-IDF coding matrix to obtain the representation vector of the multi-dimensional classification information of the node, which is difficult to effectively deal with the incomplete data in the multi-dimensional classification information of the node. Although NTI_NRL, RolEANE and HANE also achieved good results, the effect is lower than our method due to the lack of strategic research on missing data.

In contrast, the UPP-SNE algorithm uses the word vector model to effectively realize the fusion of two aspects of information, and the node classification performance is further improved. However, because the word vector model is a shallow model, it is difficult to effectively mine the complex nonlinear relationship between the node network structure information and multi-dimensional classification information, therefore, the performance is not optimal.

Finally, compared with the SNE algorithm that also uses deep neural network for fusion representation learning, the proposed algorithm in this paper can more effectively utilize the multi-dimensional classification information of nodes. On the one hand, in the process of model design, an optimization loss function based on homogeneity constraints is added, the IMCIN_R method can fully retain the attribute similar-ity information between the original network nodes and mine the latent semantic information of the multi-dimensional classification information of nodes. The node classification performance on the three datasets is improved by an average of 2.28%. On the other hand, in the process of model training, random perturbation of the multi-dimensional classification information of nodes is introduced, so that the IMCIN method can still obtain a highly discriminative representation vector when the multi-dimensional classification information of nodes is incomplete.

It is observed that with a training rate of only 2%, the node classification performance is improved by 3.14% (GPLUS dataset), 1.73% (OKLAHOMA dataset) and 1.91% (UNC dataset), respectively. Which also verifies the robustness of the IMCIN algorithm in the case of incomplete multi-dimensional classification information of nodes.

E. Hyperparameter sensitivity analysis

The IMCIN model contains two main hyperparameters, the attribute similarity retention weight coefficient (λ) and the random perturbation probability of the input data (ε). This section further analyzes the impact of hyperparameters selection on the performance of the IMCIN model through experiments. During the experiment, except for the test parameters, the rest of the parameters are set to default values, and the proportion of supervised training data for the classifier is set to 10%. Fig. 3 shows the node classification *F1* values of the IMCIN model under different parameters.

By adjusting the λ value, the influence of the attribute similarity retention weight coefficient on the node classification performance is studied. When λ is small, the node classification performance gradually improves as λ increases. As shown in Fig. 3 (left), when λ is around 0.3, it achieves better performance in all three datasets. When λ is larger, the node classification performance gradually decreases, because the larger λ makes all node representation vectors tend to be the same, which weakens the influence of structural feature information between nodes on the representation vector, resulting in a decrease in classification performance. Therefore, the default value is set to 0.3 in the simulation of the IMCIN algorithm.

It is found from the comparative experiment as shown in Fig.3 (right) that as the random disturbance probability increases, the node classification performance gradually improves. When ε exceeds a certain value, the node classification performance gradually decreases, indicating that the introduction of excessive random noise leads to the loss of key information of node attributes. Therefore, the default value is set to 0.5 in the simulation of the IMCIN algorithm.

Fig. 4 shows the experimental results of the sensitivity of the IMCIN model to the parameters of the representation learning dimensions. In our experiments, we set the training ratio for node classification to 50% and the learning dimension from 50 to 800. it can be seen from the Micro-F1 scores and Macro-F1 scores of the dimensionality analysis experiments that the performance of the proposed model increases as the representation learning dimension increases, but the performance decreases significantly when the representation learning dimension is too high. However, one of the objectives of the network representation learning algorithm is to reduce the dimensionality while preserving the network feature information, so it is not practical to set the representation learning dimension too large.

The analytical experiments of the IMCIN model on the training ratio are shown in Fig. 5. In the experiments, we set the representation learning dimension to 128 and the training ratio of node classification changes from 10% to 90%. From the experimental results, we can see that the performance of the IMCIN model increases with the increase of the training ratio and there is no overfitting phenomenon. The results show that our model has strong robustness.

F. Network Visualization

We select the representation vectors of 2663 authors in the DBLP dataset that belong to four fields of data mining, artificial intelligence, information retrieval and computer vision for visualization, and use the t-SNE dimensionality reduction algorithm to reduce the 128 dimensional representation vector to 2 dimensions. In our experiments, we use different color dots to represent the 2-dimensional visualization results of authors in different research fields. The blue dots indicate authors in the field of information retrieval, the red dots indicate authors in the field of computer vision research, the green dots indicate authors in the field of data mining, and the purple dots indicate authors in the field of artificial intelligence. It can be seen from Fig. 4 that the IMCIN algorithm proposed in this paper can map authors belonging to different fields to different communities, and its visualization effect is better.

VI. Conclusion

In this paper, we propose a network representation learning model that fuses multi-dimensional classification information of nodes, and utilizes the prior information of nodes multi-dimensional classification information, which plays a key role in the evolution of the network, to improve the performance of network representation learning. In the future work, we will try to fill in missing attributes based on dynamic network representation learning method, and then mining valuable user privacy attributes.

References

[1] Q. Li, Z. Cao, M. Tanveer, H. M. Pandey, and C. Wang, “An effective reliability evaluation method for power communication network based on community structure,” *IEEE Transactions on Industry Applications*, vol. 56, no. 4, pp. 4489–4500, 2019.

[2] A. Shahrbanoonezhad and A. Molahosseini, “Residue number system sign detector design based on dynamic range partitioning.” *Engineering Letters*, vol. 30, no. 2, pp. 546–553, 2022.

[3] M. Li, F. Wang, X. Chang, and X. Liang, “Auxiliary signal-guided knowledge encoder-decoder for medical report generation,” 2020. [Online]. Available: https://arxiv.org/abs/2006.03744

[4] P. Goyal and E. Ferrara, “Graph embedding techniques, applications, and performance: A survey,” *Knowledge-based Systems*, vol. 151, pp. 78–94, 2018.

[5] H. Cai, V. W. Zheng, and K. C.-C. Chang, “A comprehensive survey of graph embedding: Problems, techniques, and applications,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1616–1637, 2018.

[6] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, and M. Zhou, “Sentiment embeddings with applications to sentiment analysis,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 496–509, 2015.

[7] L. Liao, X. He, H. Zhang, and T.-S. Chua, “Attributed social network embedding,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 12, pp. 2257–2270, 2018.

[8] W. Wang, D. Ma, G. Xin, Y. Han, J. Huang, and B. Wang, “A network representation learning method based on topology,” *Information Sciences*, vol. 571, pp. 443–458, 2021.

[9] H. Yu, W. Dong, and J. Shi, “Raneddi: Relation-aware network embedding for drug-drug interaction prediction,” *Information Sciences*, vol. 582, pp. 167–180, 2022.

[10] A. L. Traud, P. J. Mucha, and M. A. Porter, “Social structure of facebook networks,” *Physica A: Statistical Mechanics and Its Applications*, vol. 391, no. 16, pp. 4165–4180, 2012.

[11] C. Zhu, Z. Yang, X. Xia, N. Li, F. Zhong, and L. Liu, “Multimodal reasoning based on knowledge graph embedding for specific diseases,” *Bioinformatics*, vol. 38, no. 8, pp. 2235–2245, 2022.

[12] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, “Graph neural networks: A review of methods and applications,” *AI Open*, vol. 1, pp. 57–81, 2020.

[13] S. Zhou, H. Xu, Z. Zheng, J. Chen, J. Bu, J. Wu, X. Wang, W. Zhu, M. Ester *et al.*, “A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions,” *arXiv preprint arXiv:2206.07579*, 2022.

[14] P. Cui, X. Wang, J. Pei, and W. Zhu, “A survey on network embedding,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 5, pp. 833–852, 2018.

[15] Y. Qin, F. Zhao, Y. Fang, H. Luo, and C. Wang, “Memory attention enhanced graph convolution long short-term memory network for traffic forecasting,” *International Journal of Intelligent Systems*, vol. 37, no. 9, pp. 6555–6576, 2022.

[16] K. Chowdhary, “Natural language processing,” *Fundamentals of Artificial Intelligence*, pp. 603–649, 2020.

[17] J. Chen, Z. Gong, W. Wang, and W. Liu, “Hns: Hierarchical negative sampling for network representation learning,” *Information Sciences*, vol. 542, pp. 343–356, 2021.

[18] X. Liu, C. Zhuang, T. Murata, K.-S. Kim, and N. Kertkeidkachorn, “How much topological structure is preserved by graph embeddings?” *Computer Science and Information Systems*, vol. 16, no. 2, pp. 597–614, 2019.

[19] E. Nasiri, K. Berahmand, Z. Samei, and Y. Li, “Impact of centrality measures on the common neighbors in link prediction for multiplex networks,” *Big Data*, vol. 10, no. 2, pp. 138–150, 2022.

[20] X. Luo, Z. Liu, L. Jin, Y. Zhou, and M. Zhou, “Symmetric nonnegative matrix factorization-based community detection models and their convergence analysis,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 1203–1215, 2021.

[21] C. TU, C. YANG, Z. LIU, and M. SUN, “Network representation learning: An overview,” *Scientia Sinica Informationis*, vol. 47, no. 8, pp. 980–996, 2017.

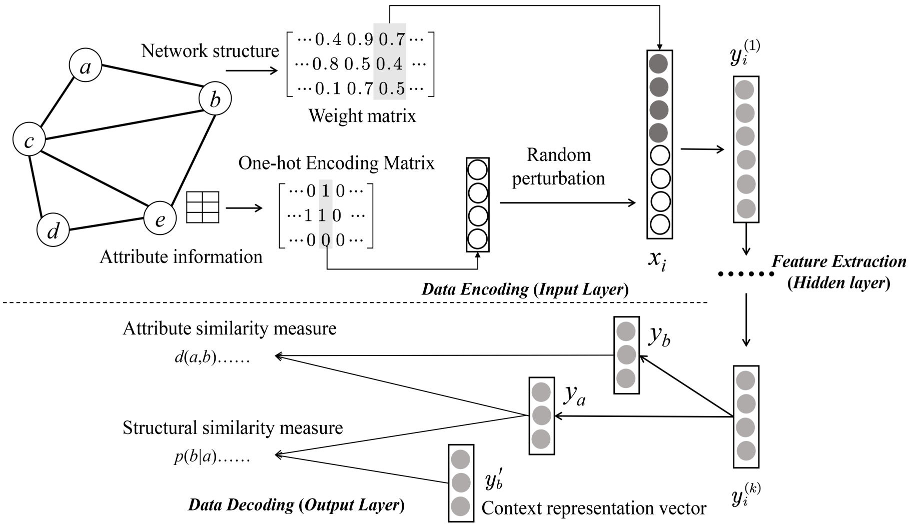
Network structure
Weight matrix
One-hot Encoding Matrix
Attribute information
Random perturbation
Data Encoding (Input Layer)
Feature Extraction (Hidden layer)
Attribute similarity measure
d(a,b)……
Structural similarity measure
p(b|a)……
Data Decoding (Output Layer)
Context representation vector

Fig. 1: Schematic diagram of IMCIN model.

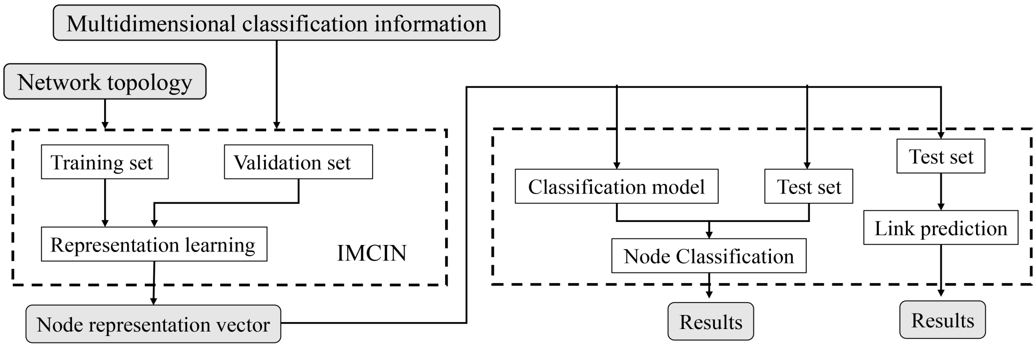

Fig. 2: Experimental flow chart.

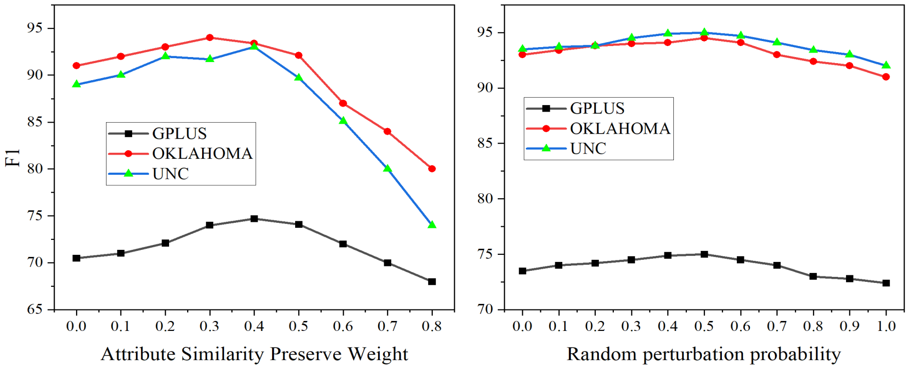

Fig. 3: Influence of IMCIN model hyperparameters on node classification performance.

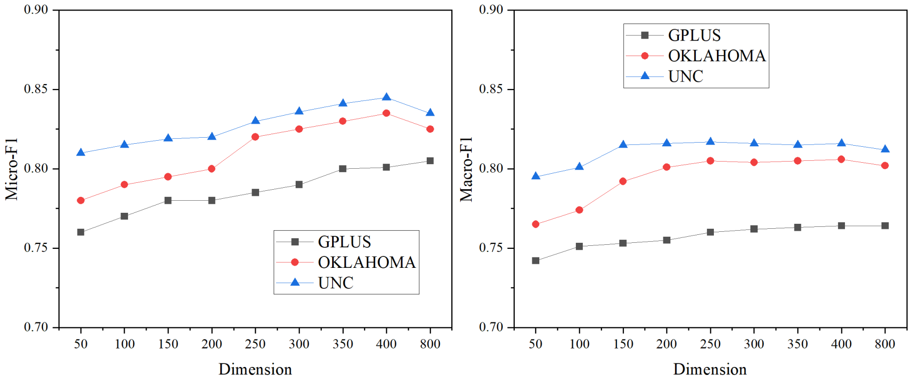

Fig. 4: Results of different dimensions.

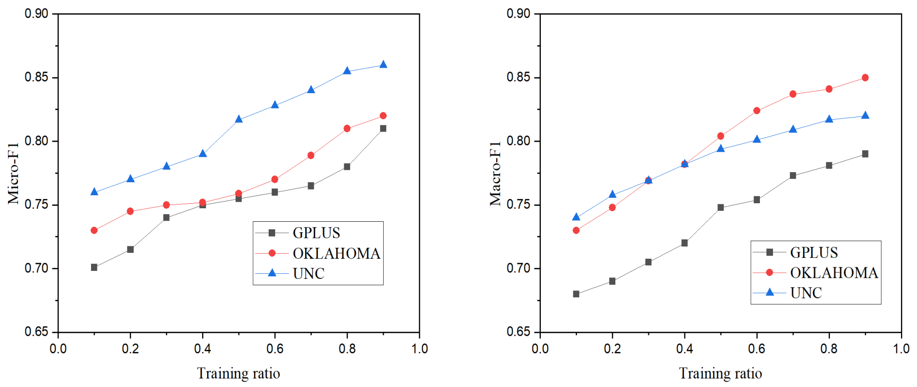

Fig. 5: Results of different training ratios.

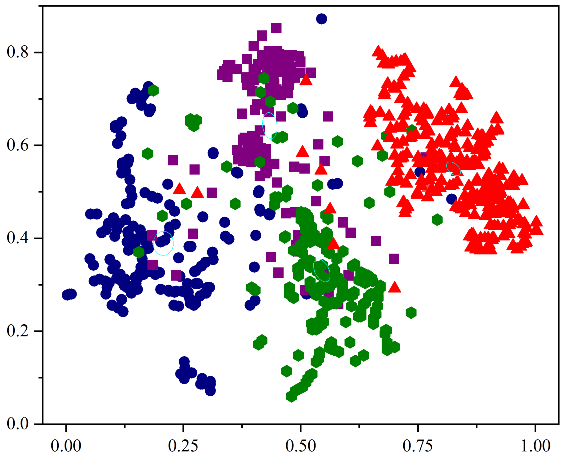

Fig. 6: Visualization of DBLP datasets.

TABLE III: Node classification performance F1 value on GPLUS network

Datasets	2%	4%	6%	8%	10%	20%	40%
Deepwalk	54.21	60.34	63.51	65.23	66.37	67.14	68.39
TADW	55.93	61.78	65.39	68.17	70.65	71.39	72.16
IMCNRL_R	68.14	67.98	67.95	68.56	68.34	69.81	71.42
UPP-SNE	66.78	67.52	68.33	69.08	69.54	70.29	71.12
SNE	67.82	68.97	69.91	71.45	72.48	73.24	73.89
HANE	67.41	68.25	69.87	70.9	69.98	71.58	72.64
NTF	68.47	70.75	70.98	71.28	71.94	72.08	73.14
RolEANE	69.54	71.17	72.03	73.49	72.14	74.64	75.09
NTI_NRL	70.81	71.93	72.07	74.14	74.56	75.04	75.19
NAIE	72.07	73.14	72.98	74.05	74.99	75.13	75.64
NIS	71.83	72.05	72.79	73.94	74.66	74.93	75.21
IMCIN	72.14	73.78	73.02	74.17	75.89	75.97	76.03

TABLE IV: Node classification performance F1 value on OKLAHOMA network

Datasets	2%	4%	6%	8%	10%	20%	40%
Deepwalk	82.84	84.73	86.22	87.38	88.17	88.66	88.99
TADW	79.53	82.36	84.77	86.93	87.51	89.14	89.68
IMCNRL_R	84.97	85.18	85.62	88.04	90.91	90.19	92.35
UPP-SNE	82.45	86.81	87.58	89.13	89.67	90.97	91.49
SNE	83.16	86.49	88.68	89.54	91.04	92.17	92.81
HANE	82.95	85.79	87.09	89.07	92.8	92.57	91.58
NTF	85.96	87.75	88.04	89.45	90.57	91.18	91.86
RolEANE	86.11	87.97	89.41	90.17	91.28	91.61	92.94
NTI_NRL	86.14	88.87	90.54	91.27	92.07	92.86	93.31
NAIE	87.08	90.15	91.67	92.52	92.54	93.98	94.41
NIS	86.83	89.81	91.26	91.93	92.78	93.94	94.36
IMCIN	88.34	91.08	92.41	92.87	93.58	94.53	94.83

TABLE V: Node classification performance F1 value on UNC network

Datasets	2%	4%	6%	8%	10%	20%	40%
Deepwalk	81.41	83.28	84.79	85.92	86.81	87.67	88.24
TADW	80.18	80.93	81.14	83.67	84.59	86.42	87.35
IMCNRL_R	84.67	86.43	86.94	90.47	90.85	89.76	92.14
UPP-SNE	82.76	85.57	87.09	87.83	88.26	89.37	89.81
SNE	84.87	86.09	87.33	88.42	89.31	90.45	91.16
HANE	83.95	86.13	87.59	87.94	89.07	90.94	91.7
NTF	85.87	86.14	86.57	87.52	88.17	88.96	89.75
RolEANE	86.21	86.45	87.51	88.69	88.49	89.07	90.64
NTI_NRL	86.41	87.14	88.24	89.01	89.58	90.12	91.62
NAIE	87.15	88.64	89.62	91.24	91.25	92.34	92.99
NIS	86.76	88.42	89.56	90.57	91.43	92.17	92.85
IMCIN	88.42	89.53	90.69	91.54	92.21	92.64	93.18