Dual Bidirectional Graph Convolutional Networks for Zero-shot Node Classification

Qin Yue
School of Computer and Information Technology, Shanxi University, Taiyuan, China
993203718@qq.com

Jiye Liang*
School of Computer and Information Technology, Shanxi University, Taiyuan, China
ljy@sxu.edu.cn

Junbiao Cui
School of Computer and Information Technology, Shanxi University, Taiyuan, China
945546899@qq.com

Liang Bai
School of Computer and Information Technology, Shanxi University, Taiyuan, China
bailiang@sxu.edu.cn

ABSTRACT

Zero-shot node classification is a very important challenge for classical semi-supervised node classification algorithms, such as Graph Convolutional Network (GCN) which has been widely applied to node classification. In order to predict the unlabeled nodes from unseen classes, zero-shot node classification needs to transfer knowledge from seen classes to unseen classes. It is crucial to consider the relations between the classes in zero-shot node classification. However, the GCN only considers the relations between the nodes, not the relations between the classes. Therefore, the GCN can not handle the zero-shot node classification effectively. This paper proposes a Dual Bidirectional Graph Convolutional Networks (DBiGCN) that consists of dual BiGCNs from the perspective of the nodes and the classes, respectively. The BiGCN can integrate the relations between the nodes and between the classes simultaneously in an united network. In addition, to make the dual BiGCNs work collaboratively, a label consistency loss is introduced, which can achieve mutual guidance and mutual improvement between the dual BiGCNs. Finally, the experimental results on real-world graph data sets verify the effectiveness of the proposed method.

CCS CONCEPTS

• **Information systems** → **Data mining**; • **Mathematics of computing** → **Graph theory**; • **Computing methodologies** → **Machine learning**.

KEYWORDS

Dual Bidirectional GCNs, Zero-shot Node Classification, Graph Data Analysis

ACM Reference Format:
Qin Yue, Jiye Liang, Junbiao Cui, and Liang Bai. 2022. Dual Bidirectional Graph Convolutional Networks for Zero-shot Node Classification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3534678.3539316

1 INTRODUCTION

Graph is an important representation of data in many real-world applications, such as social network [8, 17, 19], citation network [18] etc. The graph data analysis is ubiquitous and attracts increasing attentions recently. And one of the most frequently applied tasks on graph data is node classification. The traditional node classification aims to predict the unlabeled nodes with a few labeled nodes and usually assumes that the classes of labeled nodes covers all the classes. With the rapid development of deep learning, the Graph Convolutional Network (GCN) [10] becomes one of the most widely used method for traditional node classification task on graph data. However, in many practical applications, the classes from the labeled nodes can not cover the all classes, since some novel classes usually emerge. For example, as shown in Figure 1, there is a need to assign the scientific papers from a citation network into topics, but new research topics are emerging all the time. Besides, in a biological interaction network, the role of a protein is needed to be classified. But the new roles of proteins will continue to be discovered. Therefore, the labeling process of data is hard to cover the all classes.

To sum up, node classification on graph data faces the challenge of zero-shot node classification [24] that is how to classify the nodes from novel classes with only labeled nodes from seen classes. The key step of the traditional GCN is to aggregate the neighbor information on graph, which results in similar nodes on graph with similar representations. Because of the labeled nodes only from seen classes in zero-shot node classification, the learned representations of nodes by traditional GCN is not discriminative to seen and unseen classes. Therefore, the traditional GCN methods can not handle the zero-shot node classification effectively.

In addition, zero-shot learning has received great attention in computer vision [4, 20, 26] and natural language processing [2, 29, 30]. And the goal of zero-shot learning is to recognize the novel (unseen) classes with the labeled training data from seen classes. Generally, classes semantic descriptions are introduced to establish

[*] Jiye Liang is the corresponding author.

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539316

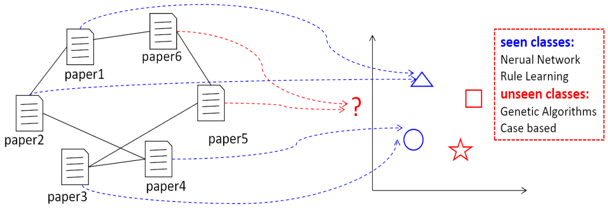


Figure 1: An example of zero-shot node classification.

a communication between seen and unseen classes. The key of zero-shot learning is to transfer knowledge learned from seen classes to unseen classes. But these methods are not designed for graph data and can not deal with directly the zero-shot node classification.

Recently, the literature is already emerging and has achieved some success for zero-shot node classification. For example, Decomposed Graph Prototype Network (DGPN) [24] was proposed. In this method, the representations of nodes are learned by following the principles of locality and compositionality in GCN, and then the learned representations of nodes are needed to be close to corresponding classes semantic descriptions. In test phase, the predictions of nodes from unseen classes are made by computing the similarity between the representations of nodes and classes semantic descriptions. Similar to zero-shot learning, the key of the zero-shot node classification also is to transfer knowledge from seen classes to unseen classes. Thus, full consideration of the relations between the classes is helpful to improve the performance of the zero-shot node classification. However, the relations between the classes are not fully considered in the method DGPN.

To solve the above challenges, this paper proposes a Dual Bidirectional Graph Convolutional Networks (DBiGCN) for zero-shot node classification. The graph is an effective and commonly used data structure for mining the relations contained in data. Therefore, the adjacency graph of the classes is constructed, which can intuitively reflect the relations between the classes in this paper. What's more, the joint representations of the nodes and the classes are learned, which is the more discriminative for zero-shot node classification. And the joint representations not only can fuse the relations between the nodes, but also fuse the relations between the classes simultaneously. Finally, it is natural that the two joint representations can be learned from perspective of the nodes and the classes, respectively. And the consistency of two joint representations are considered, so that their mutual guidance and mutual improvement can be achieved.

In summary, the high-lights of the proposed method are as follows:

Table 1: Notations

Notation	Doamin	Description
G	-	the graph data
V	-	the set of the n nodes
E	-	the set of edges between the nodes
n	$\mathbb{N}$	the number of the nodes
d	$\mathbb{N}$	the dimension of node features
d_c	$\mathbb{N}$	the dimension of classes semantic descriptions
c_s	$\mathbb{N}$	the number of seen classes
c_u	$\mathbb{N}$	the number of unseen classes
c	$\mathbb{N}$	the total number of classes
$\mathbf{x}_i$	$\mathbb{R}^d$	the vector description of the ith node
$\mathbf{a}_k$	$\mathbb{R}^{d_c}$	the vector description of the kth class
$\mathbf{X}$	$\mathbb{R}^{n\times d}$	the features matrix of the nodes
$\mathbf{A}$	$\mathbb{R}^{c\times d_c}$	the attribute matrix of the classes
$\mathbf{S}^V$	$\mathbb{R}^{n\times n}$	the adjacency matrix of the nodes
$\mathbf{Y}_L^{\text{true}}$	$\mathbb{R}^{l\times c}$	the true label matrix of the l labeled nodes
$\mathbf{S}^{\mathbf{A}}$	$\mathbb{R}^{c\times c}$	the adjacency matrix of the classes
$\mathbf{Y}^V$	$\mathbb{R}^{n\times c}$	the joint representations matrix based on the BiGCN_X
$\mathbf{Y}^{\mathbf{A}}$	$\mathbb{R}^{c\times n}$	the joint representations matrix based on the BiGCN_A

- To obtain the more discriminative joint representations of the nodes and the classes, the BiGCN is designed for zero-shot node classification. And the two joint representations are obtained based on BiGCN from perspective of the nodes and classes, respectively.
- To achieve the cooperation of the two BiGCNs, the label consistency loss is designed to constrain the two joint representations of the nodes and the classes.
- Experimental results demonstrate that the proposed method DBiGCN performs well on 3 real-world graph data sets. And we conduct ablation study that furthermore illustrates the effectiveness of the proposed method.

2 RELATED WORK

2.1 Node Classification

Node classification is one of the frequent task on graph data. The proposal of Graph Convoluntional Network (GCN) [10] that can directly operate on graph has greatly promoted the development of node classification. The GCN can encode both adjacency information and the features of nodes. Subsequently, the Graph Attention Network (GAT) [21] is proposed, which introduce the attention-based architecture to distribute different weights for different neighbors. Up to now, various graph neural network (GNN) based methods [25, 31] have been proposed and become popular for node classification.

The traditional node classification usually assumes that the classes of the labeled nodes can cover the all classes. However, the novel classes will emerge all the time. Unfortunately, the GNNs can not deal with the scenario of the unlabeled nodes from novel classes effectively.

2.2 Zero-shot Learning

Zero-shot learning has received a lot of attentions especially in the field of computer vision. The existing zero-shot learning methods can be roughly grouped into two-types: embedding-based methods [1, 5, 11] and generative-based methods [3, 7, 16, 27, 28]. The former methods aim to learn a embedding function that can align the features of images and the corresponding classes semantic descriptions. The goal of the latter methods is to learn a generator that can generate features of unseen classes, which can alleviate the domain shift problems in zero-shot learning. The key of zero-shot learning is to transfer knowledge from seen classes to unseen classes. Specifically, the relations between features and corresponding classes semantic descriptions are transferred to image recognition from unseen classes. Besides, there are some zero-shot learning methods in the field of natural language processing [2, 29, 30].

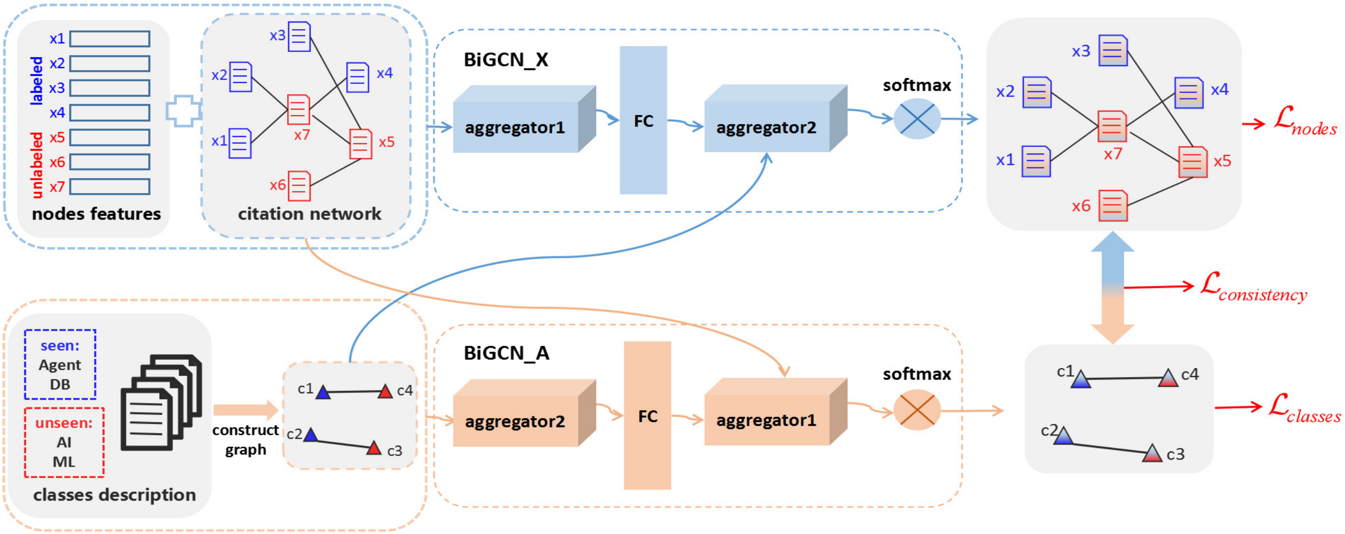


Figure 2: A schematic overview of DBiGCN. The DBiGCN consists of the dual BiGCNs from perspective of the nodes and the classes respectively and the mutual guidance between the dual BiGCNs is achieved via the consistency loss, which is united into a network. The aggregator 1 and 2 are used for aggregating the adjacency information of the nodes and the classes.

Among the above methods, some zero-shot learning methods have realized the importance of the relations between the classes. In these methods, the graph of the classes is constructed to model the relations between the classes. And then the Graph Convolutional Network is used for learning the more discriminative representations of classes[14, 23]. However, the above methods are not designed for graph data, so they can not handle directly the graph data.

Recently, the zero-shot node classification on graph data begins to be concerned [24]. Up to now, the number of the studies for zero-shot node classification on graph data is still very few.

3 PROPOSED METHOD

This section presents a formalized expression of the proposed method, and illustrates the function and work mechanism of each part of the expression.

3.1 Problem Formulation

Let $G = (V, E, \mathbf{X}, \mathbf{S}^V)$ denote an attribute graph with vertices $V = \{v_1, v_2, ..., v_n\}$ and edges $E \subseteq V \times V$. Let $\mathbf{S}^V \in \mathbb{R}^{n\times n}$ be the adjacency matrix, where s_{ij}^V is the edge weight between node v_i and v_j. Each node v_i is described by an attribute vector $\mathbf{x}_i \in \mathbb{R}^d$ and $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the attribute matrix of the nodes.

For zero-shot node classification, the space of class labels consists of two disjoint parts, i.e. $\mathcal{Y} = \mathcal{Y}_s \bigcup \mathcal{Y}_u$ and $\mathcal{Y}_s \bigcap \mathcal{Y}_u = \phi$. For the sake of discussion, we assume that there are c_s seen classes: $\mathcal{Y}_s = \{1, 2, \cdots, c_s\}$ and c_u unseen classes: $\mathcal{Y}_u = \{c_s+1, c_s+2, \cdots, c_s+c_u = c\}$. Each class is described by a semantic description vector $\mathbf{a}_k \in \mathbb{R}^{d_c}$, $k = 1, 2, \cdots, c$ and $\mathbf{A} \in \mathbb{R}^{c\times d_c}$ is the matrix of semantic description vectors of all classes.

Without loss of generality, we assume that the first l nodes are labeled and the rest u nodes are unlabeled and $l + u = n$. All the labeled nodes are from the seen classes $\mathcal{Y}_s$ and all the unlabeled nodes are from the unseen classes $\mathcal{Y}_u$. The goal of zero-shot node classification is to predict the class labels of the u unlabeled nodes.

Besides, let $\mathbf{Y}_L^{\text{true}} \in \{0, 1\}^{l\times c_s}$ be the true label matrix of the l labeled nodes. And $\forall i = 1, 2, \cdots, l$, $\forall j \in \mathcal{Y}_s$, if the class label of ith node is j, then $y_{L_{ij}}^{\text{true}} = 1$, otherwise $y_{L_{ij}}^{\text{true}} = 0$. And more notations are listed in Table 1.

3.2 Preliminaries on GCNs

For traditional node classification on graph data, the Graph Convolutional Network (GCN) [10] is a most popular method. Given a symmetric adjacency matrix $\mathbf{S}^V$, let $\mathbf{D} = \text{diag}(d_1, d_2, \cdots, d_n)$, $d_i = \sum_{j=1}^{n} s_{ij}^V$ be the degree matrix of $\mathbf{S}^V$. The $\hat{\mathbf{S}}^V = \mathbf{D}^{-\frac{1}{2}}\mathbf{S}^V\mathbf{D}^{-\frac{1}{2}}$ is calculated first. Then the general form of GCN can be written as $\mathbf{Y}^V = \sigma\left(\hat{\mathbf{S}}^V\mathbf{X}\mathbf{W}\right)$, where $\mathbf{W} \in \mathbb{R}^{d\times d'}$ is a matrix of filter parameters and $\sigma(\cdot)$ is the nonlinear activation function.

For the traditional semi-supervised node classification on graph data, we consider briefly a one-layer GCN and we have

$$\mathbf{Y}^V = \text{softmax}\left(\hat{\mathbf{S}}^V\mathbf{X}\mathbf{W}\right), \tag{1}$$

where softmax$(\cdot)$ is a nonlinear activation function and $\mathbf{W} \in \mathbb{R}^{d\times c_s}$ is an input-output weight matrix. Traditional semi-supervised node classification assumes that the the classes of the labeled nodes can cover the classes of the unlabeled nodes. Therefore, the dimension of predicting label vector is the same as the number of classes of the labeled nodes. The rows of the $\mathbf{Y}^V$ can be regarded as the representations of the nodes, and the columns of the $\mathbf{Y}^V$ can be regarded as the representations of the classes. So the goal of the GCN is learning the joint representations of the nodes and the classes.

Finally, the cross-entropy loss is calculated over all labeled nodes, and we have

$$\mathcal{L} = -\sum_{i=1}^{l}\sum_{j=1}^{c_s} y_{L_{ij}}^{\text{true}} \ln y_{ij}^{V}, \tag{2}$$

where the $y_{L_{ij}}^{\text{true}}$ is the ith row and jth column entity of the matrix $\mathbf{Y}_L^{\text{true}}$ and denotes the membership of the ith node belonging to the class j. And the y_{ij}^{V} is the ith row and jth column entity of the matrix $\mathbf{Y}^V$ and denotes the predicting probability of the ith node belonging to the class j by the GCN.

The GCN for semi-supervised node classification only aggregates the relations between the nodes, not considers the relations between the classes. But zero-shot nodes classification assumes that the classes of the unlabeled nodes are not emerging at the classes of the labeled nodes. And full consideration of the relations between the classes is crucial for zero-shot node classification. Therefore, the traditional GCN can not deal with the zero-shot node classification effectively.

3.3 Model Formulation

This paper proposes DBiGCN for zero-shot node classification. The classes of labeled nodes can not cover the classes of the unlabeled nodes, so it is crucial that how to fully consider the relations between the classes in the zero-shot node classification. In this paper, the BiGCN is designed for obtaining the joint representations of the nodes and the classes. Specifically, the BiGCN can integrate the adjacency information between the nodes and between the classes in an united network. Therefore, the learned joint representations is more discriminative for zero-shot node classification. In addition, the BiGCN can be implemented on the nodes and the classes, respectively. Thus the two joint representations of the nodes and the classes can be learned. Finally, we hope these dual BiGCNs from perspective of the nodes and the classes can work collaboratively, which can achieve their mutual guidance and mutual improvement. Therefore, the label consistency loss is introduced to constrain the two joint representations. These above three aspects are considered simultaneously in the proposed method. The final objective function can be formulated as

$$\mathcal{L}_{\text{overall}} = \mathcal{L}_{\text{nodes}} + \alpha\mathcal{L}_{\text{classes}} + \beta\mathcal{L}_{\text{consistency}}, \tag{3}$$

where $\mathcal{L}_{\text{nodes}}$ and $\mathcal{L}_{\text{classes}}$ are the loss terms of BiGCN from perspective of the nodes and the classes, respectively and $\mathcal{L}_{\text{consistency}}$ is the loss term of constraining the dual BiGCNs to collaboratively work. Besides, α and β are the trade-off parameters to balance the terms $\mathcal{L}_{\text{classes}}$ and $\mathcal{L}_{\text{consistency}}$.

As seen from formula (3), the dual BiGCNs are collaboratively trained in an unified framework, which enables the joint representations of nodes and classes to fuse the relations between the nodes and between the classes simultaneously. And the pipeline of the DBiGCN is shown in Figure 2.

The functions of these three terms are described in the following sections. We first introduce the BiGCN for zero-shot node classification and illustrate the difference between the BiGCN and GCN.

3.3.1 BiGCN. For zero-shot node classification, the relations between the classes should be fully considered during the learning process of the joint representations of the nodes and the classes. Motivated by this, a BiGCN is proposed in this paper. The graph of the classes is constructed first, which can be used for exploring the relations between the classes. Then, the relations between the nodes and between the classes need to be considered in a network. Simply, we consider a one-layer BiGCN that adopts bidirectional aggregation mechanism. And the general form can be written as

$$\mathbf{Y}^V = \text{softmax}\left(\text{relu}\left(\hat{\mathbf{S}}^V \mathbf{X}\mathbf{W}^{(1)}\right)\mathbf{W}^{(2)}\hat{\mathbf{S}}^{\mathbf{A}}\right), \tag{4}$$

where $\hat{\mathbf{S}}^{\mathbf{A}}$ is the normalized adjacency matrix of the classes defined by the distances between the classes, which can intuitively reflect the relations between the classes. And $\mathbf{W}^{(1)} \in \mathbb{R}^{d\times d'}$, $\mathbf{W}^{(2)} \in \mathbb{R}^{d'\times c}$ are the learnable parameters. In BiGCN, dimension of predicting label vector is the same as the number of all classes including seen and unseen classes. The learned joint representations of the nodes and the classes via the BiGCN not only fuses the relations between the nodes, but also fuses the relations between the classes simultaneously in an united network .

Therefore, the BiGCN can learn and transfer knowledge from seen classes to unseen classes, which is consistent with the goal of zero-shot node classification. As shown in Figure 3, the difference between the GCN and the BiGCN is exhibited. The GCN aggregates the adjacency information of nodes that reflects the relations between nodes, while the BiGCN can aggregate the adjacency information of the nodes and the classes simultaneously. Therefore, the learned joint representations of the nodes and the classes by the BiGCN is more discriminative for seen and unseen classes. So the BiGCN is more suitable for zero-shot node classification.

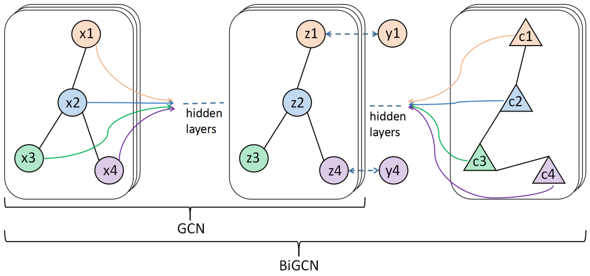


Figure 3: A schematic depiction of BiGCN. The circles represent the nodes and the black lines between the circles represent the relations between the nodes. And the triangles represent the classes and the black lines between the triangles represent the relations between the classes.

To sum up, the idea of the BiGCN is two-fold: (1) The learned joint representations of the nodes and the classes fuses the relations between the nodes, which results in the similar nodes on graph with the similar representations; (2) The learned joint representations of the nodes and the classes fuses the relations between the classes, which results in the similar classes on the graph with similar representations. Benefit from the aggregation of the relations between the classes, the zero-shot node classification become possible.

3.3.2 BiGCN from Perspective of the Nodes. The definition of the BiGCN from perspective of the nodes is shown as formula (4), and the output-layer is used for classification. The BiGCN from

Table 2: Information of the graph data sets

Data set	Nodes	Edges	Features	Classes	The space of class labels
Cora	2708	5429	1433	7	{ Neural Network, Rule Learning, Reinforcement Learning, Probabilistic Methods, Theory, Genetic Algorithms, Cased based }
Citeseer	3312	4732	3703	6	{ Agent, Information Retrieval, Database, Artificial Intelligence, Human Computer Interaction, Machine Learning }
C-M10M	4464	5804	128	6	{ Biology, Computer Science, Finacial Economics, Industrial Engineering, Physics, Social Science }

perspective of the nodes learns the joint representations of the nodes and the classes, which can preserve the relations between the nodes and between the classes in an united network. So transferring knowledge from seen classes to unseen classes can be achieved.

After obtaining the joint representations of the nodes and the classes $\mathbf{Y}^V$ by the formula (4), the cross-entropy loss function is also employed over all labeled nodes and the loss can be defined as

$$\mathcal{L}_{\text{nodes}} = -\sum_{i=1}^{l}\sum_{j=1}^{c} y_{L_{ij}}^{\text{true}} \ln y_{ij}^{V}, \tag{5}$$

where y_{ij}^V is the ith row and jth column entity of the matrix $\mathbf{Y}^V$ and denotes the predicting probability of the ith nodes belonging to class j based on BiGCN from perspective of the nodes. The BiGCN from perspective of the nodes is referenced as BiGCN_X.

3.3.3 BiGCN from Perspective of the Classes. The BiGCN from perspective of the classes also can learn the joint representations of the nodes and the classes, which can preserve the relations between the nodes and between the classes in an united network. Therefore, the BiGCN from perspective of the classes can be formulated as

$$\mathbf{Y}^{\mathbf{A}} = \text{softmax}\left(\hat{\mathbf{S}}^{\mathbf{A}}\mathbf{A}\mathbf{W}^{(3)}\hat{\mathbf{S}}^{V}\right), \tag{6}$$

where $\hat{\mathbf{S}}^{\mathbf{A}}$ is the normalized adjacency matrix of the classes that is can be defined by the distances between the classes and $\mathbf{W}^{(3)} \in \mathbb{R}^{d_c \times n}$ is the learnable parameter. The rows of $\mathbf{Y}^{\mathbf{A}} \in \mathbb{R}^{c \times n}$ can be regarded as the representations of the classes, and the columns can be regarded as the representations of the nodes. Finally, the cross-entropy loss function also be applied to all labeled nodes, we have

$$\mathcal{L}_{\text{classes}} = -\sum_{i=1}^{l}\sum_{j=1}^{c} y_{L_{ij}}^{\text{true}} \ln y_{ji}^{\mathbf{A}}, \tag{7}$$

where $y_{ji}^{\mathbf{A}}$ is the jth row and ith column entity of the matrix $\mathbf{Y}^{\mathbf{A}}$ and denotes the predicting probability of the ith nodes belonging to class j based on the BiGCN from perspective of the classes.

Different from the formula (4), the joint representations of the nodes and the classes can be learned from the aspect of classes, which can better exploit the relations between seen and unseen classes. This implement is fully consistent with the basic assumption of zero-shot node classification. The BiGCN from perspective of the classes is referenced as BiGCN_A.

3.3.4 Label Consistency Loss. We can obtain the two joint representations of the nodes and the classes by the BiGCN_X and the BiGCN_A, respectively. It is a natural idea that how can these two BiGCNs work collaboratively to achieve mutual guidance and mutual improvement between them. To achieve the goal, the label consistency loss is designed, which constrains the outputs of the BiGCN_X consistent with the outputs of the BiGCN_A. In addition, these two outputs need to be aligned with the true labels on all pairs of the labeled nodes.

Therefore, the label consistency loss is defined as

$$\mathcal{L}_{\text{consistency}} = \sum_{i=1}^{l}\sum_{j=1}^{l}\left(\mathbf{y}_i^V\mathbf{y}_j^{\mathbf{A}} - \mathbf{y}_i^{\text{true}}\left(\mathbf{y}_j^{\text{true}}\right)^T\right)^2, \tag{8}$$

where $\mathbf{y}_i^V \in [0,1]^{1\times c}$ denotes the ith row of the $\mathbf{Y}^V$ and is the predicting label probability vector of the ith nodes based on the BiGCN_X. Similarly, $\mathbf{y}_i^{\mathbf{A}} \in [0,1]^{c\times 1}$ denotes the ith column of the $\mathbf{Y}^{\mathbf{A}}$ and is the predicting label probability vector of the ith nodes based on the BiGCN_A. And $\mathbf{y}_i^{\text{true}}$ is the true one-hot label vector of the ith nodes.

For simplicity, formula (8) can be formulated as

$$\mathcal{L}_{\text{consistency}} = \left\|\mathbf{Y}_L^V\mathbf{Y}_L^{\mathbf{A}} - \mathbf{Y}_L^{\text{true}}\left(\mathbf{Y}_L^{\text{true}}\right)^T\right\|_F^2, \tag{9}$$

where $\mathbf{Y}_L^V \in [0,1]^{l\times c}$ is the predicting label matrix of the l labeled nodes based on the BiGCN_X. Similarly, $\mathbf{Y}_L^{\mathbf{A}} \in [0,1]^{c\times l}$ is the predicting label matrix of the l labeled nodes based on the BiGCN_A. $\mathbf{Y}_L^{\text{true}}$ is the true label matrix of the l labeled nodes.

3.4 Solution and Prediction

In this paper, the gradient descent method is employed to solve the optimization problem (3). Specifically, the Adam optimizer [9] implemented by Pytorch[1] is used in this paper.

In the test phase, the goal of the zero-shot node classification is predicting the u unlabeled nodes from unseen classes, and the predicting label of the ith unlabeled node based on the method DBiGCN is

$$\hat{y}_i = \arg\max_{j\in\mathcal{Y}_u} y_{ij}^V. \tag{10}$$

Finally, the procedure of DBiGCN is summarized in Algorithm 1.

4 EXPERIMENTS

To evaluate the performance of DBiGCN proposed in this paper, we conduct the experiments compared with the representative zero-shot learning metods in computer vision and the latest zero-shot node classification method. In addition, we design the different

[1] https://pytorch.org/

Algorithm 1: Dual Bidirectional Graph Neural Network for Zero-shot Node Classification (DBiGCN)

```
Input: The graph data $G = (V, E, \mathbf{X}, \mathbf{S}^V)$, the classes
       semantic descriptions matrix $\mathbf{A}$, the true labels
       matrix of the $l$ labeled nodes $\mathbf{Y}_L^{\text{true}}$, and the trade-off
       parameters $\alpha$, $\beta$.
Output: the predicting labels of the $u$ unlabeled nodes.
Initialize the network parameters $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$, and $\mathbf{W}^{(3)}$;
Calculate and normalize the adjacency matrix of the classes;
repeat
    Update the network parameters $\mathbf{W}^{(1)}$, $\mathbf{W}^{(2)}$, and $\mathbf{W}^{(3)}$
      with the formula (3);
until $\mathcal{L}_{\text{overall}}$ convergence;
Predict the labels of $u$ unlabeled nodes with the formula (10).
Return $\hat{y}_i, i = 1, 2, \cdots, u$.
```

experimental settings to exhibit the effectiveness and applicability of the proposed method DBiGCN for zero-shot node classification.

4.1 Data Sets

We conduct experiments on three real-world citation network, which are Cora [13], Citeseer [6], and C-M10M [24]. And the basic information of the three data sets are summarized as Table 2.

For zero-shot node classification, two kind of classes semantic descriptions (CSDs) [24], LABEL-CSDs and TEXT-CSDs, are used in the experiments. And the splits of the seen and unseen classes follows the setting of the literature [24] for comparison.

4.2 Experimental Settings

In the method DBiGCN proposed in this paper, there are two trade-off parameters α and β that both are selected within $\{0.01, \ 0.1, 1, 10, \ 100\}$. And the output dimension of $\mathbf{W}^{(1)}$ is simply fixed 512 in the whole experiment.

Some classical and recent Zero-shot learning methods are selected for comparison, including DAP (Direct Attribute Prediction) [11] and the variant DAP (CNN), ESZSL (Embarrassingly Simple Zero-shot Learning) [15], ZS-GCN (GCN for Zero-shot Learning) [23] and the variant ZS-GCN (CNN), WDVSc (Wasserstein-Distance-based Visual Structure Constraint) [22] and Hyperbolic-ZSL [12], which are designed for image recognition in computer vision. In addition, the method DGPN [24] concerned on zero-shot node classification also is selected as comparison method. Finally, the *RandomGuess* (randomly guessing the label for the unlabel nodes) is regarded as a baseline. And the more detail experimental setting can be found in Appendix.

4.3 Performance Analysis on the Proposed Method for Zero-shot Node Classification

The zero-shot node classification accuracy using the TEXT-CSDs of the proposed method and comparison methods on the three graph data sets are presented in Table 3.

For the class split I, we have the following observations:

(a) The method DBiGCN outperforms the *RandomGuess* and eight comparison methods for zero-shot node classification accuracy on the all experimental data sets. First, compared with the classical zero-shot learning methods, the DBiGCN has a significant improvement. Second, the DBiGCN also performs well compared with the DGPN desighed for zero-shot node classification and achieves 16.55% average accuracy improvement.

(b) The performance of the method DGPN and DBiGCN that are designed for the zero-shot node classification on graph data is superior to the methods that are designed for image recognition in computer vision. The main reason is that the method DGPN and DBiGCN can better integrate the information of the graph data.

(c) Compared with the method DGPN, the method DBiGCN achieves higher performance on all data sets. The main reason is that the method DBiGCN can obtain more discriminative representations for zero-shot node classification by integrating the BiGCN_X and BiGCN_A into an united network.

Table 3: Zero-shot node classification accuracy (%) using the TEXT-CSDs

		Cora	Citeseer	C-M10M
Class Split I	RandomGuess	25.35	24.86	33.21
	DAP	26.56	34.01^3	38.71^3
	DAP(CNN)	27.80	30.45	32.97
	ESZSL	27.35	30.32	37.00
	ZS-GCN	25.73	28.62	37.89
	ZS-GCN(CNN)	16.01	21.18	36.44
	WDVSc	30.62^3	23.46	38.12
	Hyperbolic-ZSL	26.36	34.18	35.80
	DGPN	33.78^2	38.02^2	41.98^2
	DBiGCN	**45.14**^{1}	**40.97**^{1}	**45.45**^{1}
	Improve rate	33.63%	7.76%	8.27%
Class Split II	RandomGuess	32.69	50.48	49.73
	DAP	30.22	53.30	46.79
	DAP(CNN)	29.83	50.07	46.29
	ESZSL	38.82^3	55.32^3	56.07^3
	ZS-GCN	29.53	52.22	56.07
	ZS-GCN(CNN)	33.20	49.27	51.37
	WDVSc	34.13	52.70	46.26
	Hyperbolic-ZSL	37.02	46.27	55.07
	DGPN	46.40^2	**61.90**^{1}	62.46^2
	DBiGCN	**49.20**^{1}	60.11^2	**71.86**^{1}
	Improve rate	6.03%	-2.89%	15.05%

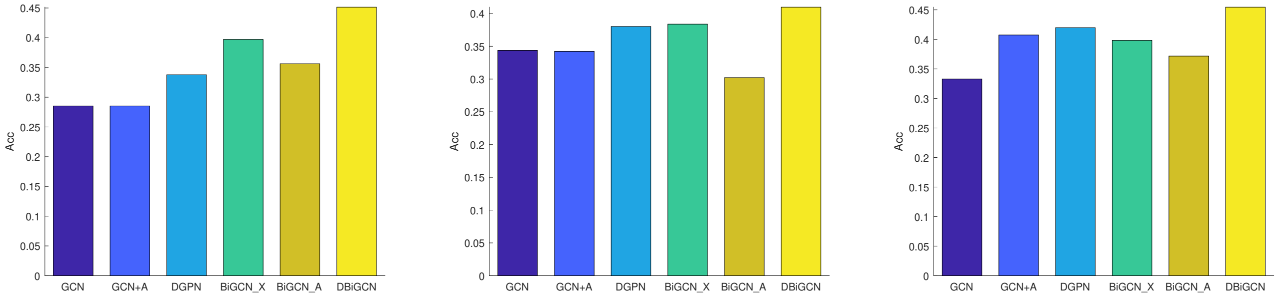
Under the class split II, there are similar conclusions with the class split I. And the method DBiGCN obtains a improved performance on the data sets Cora and C-M10M. On the data set Citeseer, the performance of the is inferior to the method DPGN. Finally, the method DBiGCN achieves 5.26% average accuracy improvement.

4.4 The Comparison of the Zero-shot Node Classification Accuracy Using the Different CSDs

There are two kind of classes semantic descriptions provided in the literature [24], the TEXT-CSDs and the LABEL-CSDs. According

Table 4: The Comparison of zero-shot node classification accuracy (%) using the different CSDs

		Cora			**Citeseer**			**C-M10M**		
		TEXT	LABEL	Decline rate	TEXT	LABEL	Decline rate	TEXT	LABEL	Decline rate
Class Split I	DAP	26.56	25.34	-4.59%	34.01	30.01	-11.76%	38.71	32.67	-15.60%
	ESZSL	27.35	25.79	-5.70%	30.32	28.52	-5.94%	37.00	35.02	-5.35%
	ZS-GCN	25.73	23.73	-7.77%	28.62	26.11	-8.77%	37.89	33.32	-12.06%
	WDVSc	30.62	18.73	-38.83%	23.46	19.70	-16.02%	38.12	30.82	-19.15%
	Hyperbolic-ZSL	26.36	25.47	-3.38%	34.18	21.04	-38.44%	35.80	34.49	-3.66%
	DGPN	33.78	32.55	-3.64%	38.02	31.83	-16.28%	41.98	35.05	-16.51%
	DBiGCN	**45.14**	**39.05**	-13.49%	**40.97**	**39.10**	-3.10%	**45.45**	**43.71**	-3.83%



(a) Cora

(b) Citeseer

(c) C-M10M

Figure 4: The comparison of the different methods based on Graph Convolutional Network for zero-shot node classification. The abscissa represents the different methods and the ordinate represents the accuracy of the zero-shot node classification.

the literature [24], the quality of the TEXT-CSDs is higher that the LABEL-CSDs. In this section, we conduct the experiments of the zero-shot node classification using these two different CSDs. The comparison results are exhibited in the Table 4 and we have

(a) In the zero-shot node classification using the LABEL-CSDS, the performance of the method DBiGCN has a significant improvement, compared with the other zero-shot learning or zero-shot node classification methods.

(b) The performance of zero-shot node classification using the LABEL-CSD is lower than the performance using the TEXT-CSDs, which is consistent with the conclusion concluded by literature [24]. The reason is that the TEXT-CSDs contains more information than the LABEL-CSDs during the process of the feature extraction.

(c) Some methods, including the ESZSL and DBiGCN, have more stable performance, though the quality of the classes semantic descriptions deteriorates. However, the others have a sharp decline under the same scenario.

Therefore, the method proposed in this paper achieves a better and more stable performance under the scenario of using the classes semantic descriptions of different qualities.

4.5 Advantages of the BiGCN

To demonstrate the advantages of BiGCN, this section designs the experiments of the traditional GCN [10] and BiGCN under different scenarios. Specifically, the comparison methods includes GCN, GCN+A, DGPN, BiGCN_X, BiGCN_A and DBiGCN.

The method GCN is the traditional Graph Convolutional Network that does not consider the classes semantic descriptions.

The method GCN+A is that the learned representations of nodes by the traditional Graph Convolutional Network are needed to be aligned with the corresponding classes semantic description.

The method DGPN is implemented by the Decomposed Graph Convolutional Network that considers the local and global information. Besides, the classes semantic descriptions also are considered.

The method BiGCN_X is the Bidirectional Graph Convolutional Network from perspective of the nodes (see section 3.3.2).

The method BiGCN_A is the Bidirectional Graph Convolutional Network from perspective of the classes (see section 3.3.3).

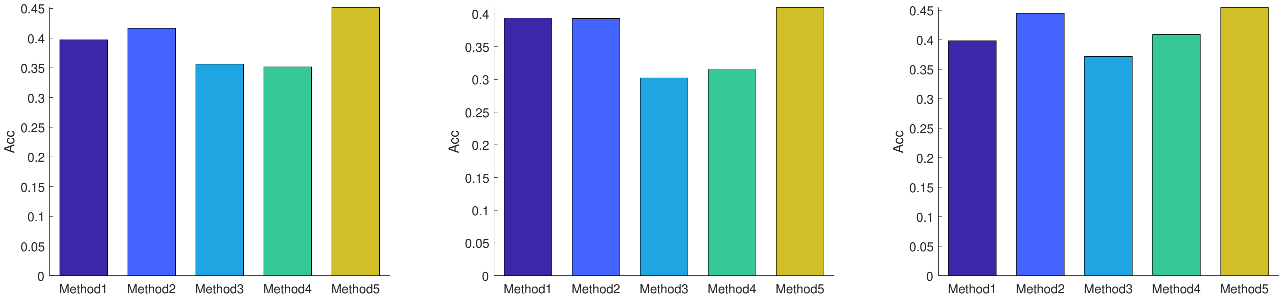
The method DBiGCN integrates the BiGCN_X and BiGCN_A into an united framework, namely the full model.

The performance of the above methods is shown in the Figure 4. And we have the following conclusions:

(a) The performance of the method GCN+A is equal to or better than the method GCN. Because the GCN does not consider the classes semantic descriptions.

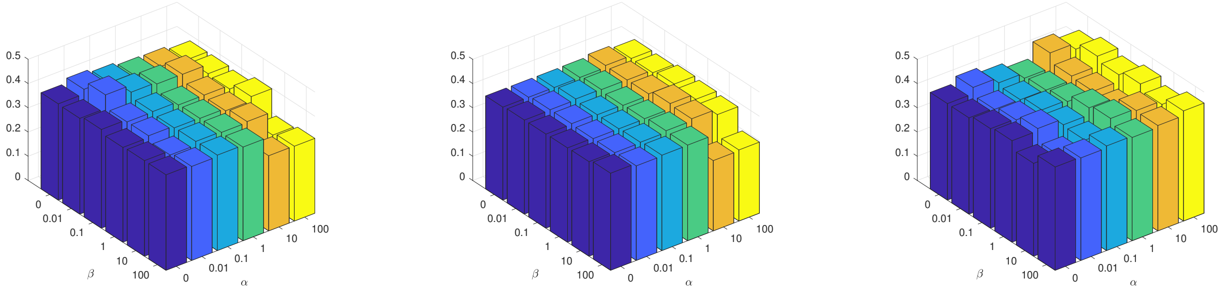
(b) The method DGPN is superior to the GCN+A. The method DGPN considering the local and global information is a variation of the GCN.

(c) Compared with the DGPN, the method BiGCN_X with Bidirectional Graph Convolutional Network has an improved performance



(a) Cora (b) Citeseer (c) C-M10M

Figure 5: The zero-shot node classification accuracy of the five ablative methods from the proposed model.



(a) Cora (b) Citeseer (c) C-M10M

Figure 6: The variations of the zero-shot node classification accuracy of the proposed method under different parameters α and β on all data sets.

on the data sets Cora and Citeseer. However, the performance of the method BiGCN_X is less than the method DGPN. The main reason is that the classes semantic descriptions are not fully exploited.

(d) The performance of the method BiGCN_A is inferior to the method BiGCN_X. This phenomenon shows that the features of nodes is more informative and discriminative than the the attributes of the classes.

(e) Compare with all the other methods, the method DBiGCN has a significant improvement on the all data sets. The main reason is that the method DBiGCN integrates the BiGCN_X and the BiGCN_A into an united network and the more discriminative joint representations of the nodes and the classes can be obtained for zero-shot node classification.

4.6 Ablation Study

To provide further insight into the method DBiGCN, we conduct the ablation studies to evaluate the effectiveness and function of the different components. The method DBiGCN consists of three components. And the following 5 different variants are designed.

- Method 1: refers to BiGCN_X and BiGCN_A with the loss term $\mathcal{L}_{\text{nodes}}$
- Method 2: refers to BiGCN_X and BiGCN_A with the loss term $\mathcal{L}_{\text{nodes}}$ and the loss term $\mathcal{L}_{\text{consistency}}$
- Method 3: refers to BiGCN_X and BiGCN_A with the loss term $\mathcal{L}_{\text{classes}}$
- Method 4: refers to BiGCN_X and BiGCN_A with the loss term $\mathcal{L}_{\text{classes}}$ and the loss term $\mathcal{L}_{\text{consistency}}$
- Method 5: refers to BiGCN_X and BiGCN_A with the all loss terms, namely the full model

The zero-shot node classification accuracy of the above five methods are shown in Figure 5. The following observations can be seen

(a) The performance of the Method 2 considering the consistency loss is better than the Method 1 on the all data sets. Similarly, the performance of the Method 4 considering the label consistency loss is better than the Method 3 on the data set Citeseer and C-M10M and equal to the Method 3 on the data set Cora. The results intuitively illustrate that the effectiveness of the label consistency loss $\mathcal{L}_{\text{consistency}}$.

(b) On the all data sets, the performance of the Method 5 consisting of dual Bidirectional Convolutional Networks is superior to the Method 1 and the Method 3 with singe BiGCN. The experimental results demonstrate the effectiveness and necessity of the DBiGCN.

4.7 Parameters Sensitivity Analysis

There are two trade-off parameters, α and β, needed to be determined for the method DBiGCN. This section conducts the experiments on the three data sets to study the parameter sensitivity of the method DBiGCN to parameters variations of the α and β in terms of the zero-shot node classification accuracy using the TEXT-CSDs under the class split I (see Figure 6).

Figure 6 shows that the zero-shot node classification accuracy of the method DBiGCN is not sensitive to parameters variations on the all data sets. And the competitive performance can be easily obtained over a limited range.

5 CONCLUSION

A new Graph Convolutional Network for zero-shot node classification is proposed. Its loss function is made up of three terms, i.e. cross-entropy losses of the BiGCN from perspective of the nodes and classes, respectively, and the label consistency loss. The cross-entropy loss of the BiGCN from perspective of the nodes (classes) can constrain the learned joint representations fuse the information of the nodes (classes) and the relations between the nodes and between the classes in an united network simultaneously. The label consistency loss can constrain the dual BiGCNs work collaboratively. In addition, the bidirectional propagation mechanism is proposed to implement the new loss. Finally, the experiments on graph data sets demonstrate the effectiveness of the proposed method.

ACKNOWLEDGMENTS

This work was supported by the National Key Research and Development Program of China (2020AAA0106100) and the National Natural Science Foundation of China (Nos. U21A20473, 62022052).

REFERENCES

[1] Zeynep Akata, Florent Perronnin, Zaïd Harchaoui, and Cordelia Schmid. 2016. Label-embedding for image classification. *IEEE Transactions Pattern Analysis and Machine Intelligence* 38, 7 (2016), 1425–1438.

[2] Qi Chen, Wei Wang, Kaizhu Huang, and Frans Coenen. 2021. Zero-shot text classification via knowledge graph embedding for social media data. *IEEE Internet of Things Journal* (2021).

[3] Shiming Chen, Wenjie Wang, Beihao Xia, Qinmu Peng, Xinge You, Feng Zheng, and Ling Shao. 2021. Free: Feature refinement for generalized zero-shot learning. In *IEEE/CVF International Conference on Computer Vision, Online*. 122–131.

[4] Shiming Chen, Guo-Sen Xie, Yang Liu, Qinmu Peng, Baigui Sun, Hao Li, Xinge You, and Ling Shao. 2021. HSVA: Hierarchical Semantic-Visual Adaptation for Zero-Shot Learning. In *35th Annual Conference on Neural Information Processing Systems, virtual*. 16622–16634.

[5] Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomás Mikolov. 2013. DeViSE: A deep visual-semantic embedding model. In *27th Annual Conference on Neural Information Processing Systems. Lake Tahoe, Nevada, USA*. 2121–2129.

[6] C. Lee Giles, Kurt D. Bollacker, and Steve Lawrence. 1998. CiteSeer: An automatic citation indexing system. In *3rd ACM International Conference on Digital Libraries, Pittsburgh, PA, USA*. ACM, 89–98.

[7] Jiechao Guan, Zhiwu Lu, Tao Xiang, Aoxue Li, An Zhao, and Ji-Rong Wen. 2021. Zero and few shot learning with semantic feature synthesis and competitive learning. *IEEE Transactions Pattern Analysis and Machine Intelligence* 43, 7 (2021), 2510–2523.

[8] Jooho Kim and Makarand Hastak. 2018. Social network analysis: Characteristics of online social networks after a disaster. *International Journal of Information Management* 38, 1 (2018), 86–96.

[9] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, San Diego, CA, USA*.

[10] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, Toulon, France*.

[11] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. 2014. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions Pattern Analysis and Machine Intelligence* 36, 3 (2014), 453–465.

[12] Shaoteng Liu, Jingjing Chen, Liangming Pan, Chong-Wah Ngo, Tat-Seng Chua, and Yu-Gang Jiang. 2020. Hyperbolic visual embedding learning for zero-shot recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA*. 9270–9278.

[13] Andrew McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval* 3, 2 (2000), 127–163.

[14] Guangjin Ou, Guoxian Yu, Carlotta Domeniconi, Xuequan Lu, and Xiangliang Zhang. 2020. Multi-label zero-shot learning with graph convolutional networks. *Neural Networks* 132 (2020), 333–341.

[15] Bernardino Romera-Paredes and Philip H. S. Torr. 2015. An embarrassingly simple approach to zero-shot learning. In *32nd International Conference on Machine Learning, Lille, France*, Vol. 37. 2152–2161.

[16] Edgar Schönfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. 2019. Generalized zero- and few-Shot learning via aligned variational autoencoders. In *IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA*. 8247–8255.

[17] John Scott. 1988. Social network analysis. *Sociology* 22, 1 (1988), 109–127.

[18] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI Magazine* 29, 3 (2008), 93–106.

[19] Yu Shi, Myunghwan Kim, Shaunak Chatterjee, Mitul Tiwari, Souvik Ghosh, and Rómer Rosales. 2016. Dynamics of large multi-view social networks: synergy, cannibalization and cross-view interplay. In *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA*. 1855–1864.

[20] Bin Tong, Chao Wang, Martin Klinkigt, Yoshiyuki Kobayashi, and Yuuichi Nonaka. 2019. Hierarchical disentanglement of discriminative latent features for zero-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA*. 11467–11476.

[21] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *6th International Conference on Learning Representations, Vancouver, BC, Canada*.

[22] Ziyu Wan, Dongdong Chen, Yan Li, Xingguang Yan, Junge Zhang, Yizhou Yu, and Jing Liao. 2019. Transductive zero-shot learning with visual structure constraint. In *32th Annual Conference on Neural Information Processing Systems, Vancouver, BC, Canada*. 9972–9982.

[23] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. 2018. Zero-shot recognition via semantic embeddings and knowledge graphs. In *IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA*. 6857–6866.

[24] Zheng Wang, Jialong Wang, Yuchen Guo, and Zhiguo Gong. 2021. Zero-shot node classification with decomposed graph prototype network. In *27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore*. 1769–1779.

[25] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (2021), 4–24.

[26] Yongqin Xian, Christoph H. Lampert, Bernt Schiele, and Zeynep Akata. 2019. Zero-shot learning - A comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions Pattern Analysis and Machine Intelligence* 41, 9 (2019), 2251–2265.

[27] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. 2018. Feature generating networks for zero-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA*. 5542–5551.

[28] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. 2019. F-VAEGAN-D2: A feature generating framework for any-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA*. 10275–10284.

[29] Zhiquan Ye, Yuxia Geng, Jiaoyan Chen, Jingmin Chen, Xiaoxiao Xu, Suhang Zheng, Feng Wang, Jun Zhang, and Huajun Chen. 2020. Zero-shot text classification via reinforced self-training. In *58th Annual Meeting of the Association for Computational Linguistics, Online*. 3014–3024.

[30] Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *9th International Joint Conference on Natural Language Processing, Hong Kong, China*. 3912–3921.

[31] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open* 1 (2020), 57–81.

Table 7: The distance and renormalization setting

	TEXT-CSDs			LABEL-CSDs		
	Cora	Citeseer	C-M10M	Cora	Citeseer	C-M10M
Euclidean distance	✓	✓	✓			
Cosine distance				✓	✓	✓
Renormalization			✓	✓		

A APPENDIX

A.1 Date Sets Description

The data set Cora consists of 2708 scientific publications classified into one of 7 machine learning related classes and 5429 links. Each publication is described as 1433-dimensional vector and each dimension represents a unique word from the dictionary. And the value is 0/1-value word vector, where 0 indicates the word is absent and 1 indicates the word is present in this paper.

The data set Citeseer is a link data set built with permission from the CiteSeer Web database. It consists of 3312 scientific paper classified into one of 6 classes. And the number of the links in the citation is 4732. Each paper is described as 3703-dimensional vector and each dimension represents a unique word from the dictionary.

The data set C-M10M includes 4464 publications from 6 classes and 5804 citation links. And each publication is described in plain-text form. In the experiment, the 128-dimensional vector is used.

A.2 Data Split

In the experiment, the split of seen and unseen classes follows the literature [24]. For the class split I, the train and test classes are involved. For the class split II, the train, validation and test classes are involved. For each data set, the first few classes are the train classes, the middle ones are the validation classes. And the last ones are the test classes. The detailed partition is presented in Table 5.

Table 5: The partition of the data set

	Class split I		Class split II		
	Train	Test	Train	Validation	Test
Cora	{ 1, 2, 3 }	{ 4, 5, 6, 7 }	{ 1, 2 }	{ 3, 4 }	{ 5, 6, 7 }
Citeseer	{ 1, 2 }	{ 3, 4, 5, 6 }	{ 1, 2 }	{ 3, 4 }	{ 5, 6 }
C-M10M	{ 1, 2, 3 }	{ 4, 5, 6 }	{ 1, 2 }	{ 3, 4 }	{ 5, 6 }

A.3 Parameter Setting

Under the class split I, we adopt the grid search for the parameters of the network. And the search space of each parameter is listed in Table 6. The number of the training epoch is fixed as 10000 under the class split I. And under the class split II, the parameters of the network and the trade-off parameters are determined by the validation classes, and the number of the training epoch is fixed 600. The stop condition is that the loss on validation classes does not decrease for several consecutive epochs.

Table 6: The search space of the parameters of the DBiGCN

Parameter	the range of the value
Dropout rate	{ 0, 0.1, 0.2, 0.3 }
Learning rate	{ 1e-1, 1e-2, 1e-3, 1e-4 }
Weight decay	{ 0, 1e-4, 1e-5, 1e-6 }
α	{ 1e-2, 1e-1, 1e0, 1e1, 1e2 }
β	{ 1e-2, 1e-1, 1e0, 1e1, 1e2 }

A.4 The Renormalization Setting

In the experiment, we employ the k-nearest neighbors to construct the graph of the classes, and the computing formula is

$$s^{\mathbf{A}}_{ij} = \begin{cases} e^{-\frac{\mathrm{dis}\left(\mathbf{a}_i,\mathbf{a}_j\right)}{t}}, & \mathbf{a}_i \in \mathcal{N}\left(\mathbf{a}_j\right) \\ 0, & \text{otherwise,} \end{cases}$$

where the $\mathrm{dis}(\cdot,\cdot)$ is a distance function and Euclidean distance or cosine distance is adopted in the experiment. $\mathcal{N}\left(\mathbf{a}_j\right)$ denotes the nearest neighbors' set of the node $\mathbf{a}_j$. What's more, the renormalization trick of the graph is wildly used in graph neural network. In this experiment, we also adopt the renormalization trick on the graph based on classes. The detail experimental scheme are presented in Table 7.

A.5 Code Release

Code for reproducible experiments is available at https://github.com/warmerspring/DBiGCN.