



OPEN ACCESS

EDITED BY

Haiyong Zheng,
Ocean University of China, China

REVIEWED BY

Guoqiang Zhong,
Ocean University of China, China
Mengchu Zhou,
New Jersey Institute of Technology,
United States

*CORRESPONDENCE

Haixing Zhao
✉ h.x.zhao@163.com

RECEIVED 07 September 2023

ACCEPTED 13 February 2024

PUBLISHED 04 April 2024

CITATION

Ye Z, Li Z, Li G and Zhao H (2024) Dual-channel deep graph convolutional neural networks.

Front. Artif. Intell. 7:1290491.
doi: 10.3389/frai.2024.1290491

COPYRIGHT

© 2024 Ye, Li, Li and Zhao. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Dual-channel deep graph convolutional neural networks

Zhonglin Ye^{1,2,3,4}, Zhuoran Li^{1,2,3,4}, Gege Li^{1,2,3,4} and Haixing Zhao^{1,2,3,4*}

¹College of Computer, Qinghai Normal University, Xining, Qinghai, China, ²The State Key Laboratory of Tibetan Intelligent Information Processing and Application, Xining, Qinghai, China, ³Tibetan Information Processing and Machine Translation Key Laboratory of Qinghai Province, Xining, Qinghai, China, ⁴Key Laboratory of Tibetan Information Processing, Ministry of Education, Xining, Qinghai, China

The dual-channel graph convolutional neural networks based on hybrid features jointly model the different features of networks, so that the features can learn each other and improve the performance of various subsequent machine learning tasks. However, current dual-channel graph convolutional neural networks are limited by the number of convolution layers, which hinders the performance improvement of the models. Graph convolutional neural networks superimpose multi-layer graph convolution operations, which would occur in smoothing phenomena, resulting in performance decreasing as the increasing number of graph convolutional layers. Inspired by the success of residual connections on convolutional neural networks, this paper applies residual connections to dual-channel graph convolutional neural networks, and increases the depth of dual-channel graph convolutional neural networks. Thus, a dual-channel deep graph convolutional neural network (D2GCN) is proposed, which can effectively avoid over-smoothing and improve model performance. D2GCN is verified on CiteSeer, DBLP, and SDBLP datasets, the results show that D2GCN performs better than the comparison algorithms used in node classification tasks.

KEYWORDS

DeepGCN, D2GCN, graph neural networks, graph convolutional neural networks, GNNs

1 Introduction

The introduction of Convolutional Neural Networks (CNNs) has brought significant improvements to these areas, such as natural language processing, video processing, and image classification. However, traditional CNNs can only process European spatial data such as image (He et al., 2016), text (Hu et al., 2014), speech (Hinton et al., 2012), etc. A non-European spatial data, which is graph data, has attracted much attention for its ubiquity. Many applications in real life can be naturally represented by graph data, such as transportation networks, worldwide webs, and social networks, etc. How to define CNNs on graph data is receiving more and more attention. Drawing on the modeling ability of CNNs to local structures and the ubiquitous dependencies in graph data, Graph Convolutional Networks (GCNs) became one of the most active research fields. With the wide application of GCNs in data mining, such as recommendation system (Monti et al., 2017a; Ying et al., 2018) and point cloud segmentation (Li et al., 2019; Wang et al., 2019), researchers will pay more attention to the improvement of GCNs performance.

Another key factor behind the success of CNNs is that they can design and train deeper CNN models. However, increasing the number of convolutional layers in the graph by GCNs

may cause the gradient disappearance, which means that smoothing occurs during backpropagation, i.e., the features of all nodes in the graph converge to the same value. Thus, GCNs are generally shallow structures that contain 2–3 graph convolutional layers. Shallow structures limit the performance of the model, because they cannot mine higher-order node information. Gradient disappearance poses a challenge for the deep GCNs designing.

Gradient disappearance is also a significant factor limiting the training of deep CNNs models. ResNet introduces residual connections between convolutional layers to construct deep CNNs. Residual connections can avoid the gradient disappearance problem well by constantly reusing features. DenseNet (Huang et al., 2017) further expands ResNet, which introduces more connections between convolutional layers. However, as the convolutional layer increases, pooling can lead to more spatial information loss, but the convolution proposed in literature (Yu and Koltun, 2015) solves this problem. The above concepts have driven the rapid development of CNNs, and if they are introduced into GCNs, whether the model can get similar results to CNNs?

DeepGCNs use ResNet, DenseNet, and Void Convolution to train deep neural networks in computer vision to get success in point cloud semantic segmentation filed. However, DeepGCNs are a kind of deep neural networks based on single-feature training. Because graph convolutional networks based on single feature training cannot fully depict the relevant characteristics of the graph, and Zhao et al. (2023) consider the interaction between features and propose a GCN based on dual feature interaction. Therefore, HDGCN adds semantic features on top of structural features, which not only enriches the diversity of graph information but also enhances node features. However, HDGCN is a shallow neural network that can only demonstrate excellent performance on simple tasks, and cannot learn higher-level features for more abstract and complex data. Therefore, it is necessary to deepen the algorithm.

Drawing on the successful experience of using residual connections to construct deep GCNs in DeepGCNs, this paper successfully constructs deep dual-channel Graph Convolutional Networks Based on Hybrid Features (D2GCN) using residual connections in HDGCNs. This paper shows how residual connections can be combined with multi-layer graph convolution operations to construct D2GCN, and the effect of residual connections on the accuracy and stability of D2GCN is analyzed. This paper applies D2GCN to the task of node classification, the number of neural network layers can reach 16, and the performance of D2GCN on the three datasets is improved by about 3% compared with SOAT.

In summary, the main contributions of this paper are as follows:

- 1 Most of the existing graph neural networks are implemented based on single channel neural networks, and few achievements of dual-channel graph neural networks have been published. However, the relevant achievements of dual-channel deep graph neural networks have not been published so far. This paper is the first academic paper discussing dual-channel deep graph neural networks. The algorithm proposed in this paper is verified and introduced by many measure approaches, such as theory, experiment, comparative analysis parameter sensitivity etc.
- 2 To make full use of the feature diversity and complementarity on graph, this paper fuses the text features and structural features into hybrid features, which enriches the information diversity on graph and enhances the feature expression ability of the nodes.
- 3 Based on text features, structural features and hybrid features, three kinds of variation models based on D2GCN are proposed by using residual networks, such as $D2GCN_{(structure)}$, $D2GCN_{(semantic)}$ and $D2GCN_{(hybrid)}$. D2GCN is only the general name of dual-channel deep graph neural network, and $D2GCN_{(structure)}$, $D2GCN_{(semantic)}$ and $D2GCN_{(hybrid)}$ determine the type of graph features placed in the neural network channel.

2 Related words

Since graphs are ubiquitous in the real world, researches on graphs are receiving more and more attention from researchers. Graphs have been widely used to represent various domain information, such as recommendation system (Monti et al., 2017b), molecular graph structure (Wale et al., 2008; Zitnik and Leskovec, 2017), social network (Armeni et al., 2017), and Linguistics (Bastings et al., 2017; Marcheggiani and Titov, 2017). Graphs have also played a key role in deep learning, such as classifying the role of a protein on a bio-interaction graph, predicting the role of an author in a cooperative network, recommending new friends to users in a cooperative network, recommending new friends to users in social networks, recommending ads to users etc. However, most traditional deep learning models, such as convolutional neural networks (CNNs) and recurrent neural network (RNNs), process data limited to Euclidean space and have translational invariance and local connectivity, such as images and text. As irregular non-European data, CNNs and RNNs cannot be directly applied to the field of graph. The challenge of deep learning of graphs lies in encoding the high-dimensional, non-Euclidean information into the form of embedding and input them into subsequent analysis tasks. Graph Convolutional Neural Networks (GCNs) provide a novel direction for processing graph data, for example, graphs are used to represent individuals and the connections between individuals in social networks, and then high irregular graph data in non-European spaces are obtained. GCNs can assess the strength of individual connections in social networks, and get more accurate evaluation between individuals (Tang and Liu, 2009). GCNs have many applications in the field of computer vision, for example, graphs are used to represent semantic relationships between objects, and then objects are detected and segmented, semantic relationships between objects are predicted (Qi et al., 2017; Xu et al., 2017; Li Y. et al., 2018; Yang et al., 2018) at last. Human joints can be represented by graph and then GCNs is used to recognize the actions in video (Jain et al., 2016; Yan et al., 2018). GCNs are also the perfect approach for dealing with 3D point clouds due to its non-structural properties (Chen and Zhang, 2023; Jiang et al., 2023; Khodadad et al., 2023; Wang L. et al., 2023). Similarly, GCNs also have many applications in the field of natural language processing. In terms of sentiment analysis, they are not only applicable to unimodal sentiment analysis (Zhang et al., 2022) but also to multimodal sentiment analysis (Firdaus et al., 2023). For example, Huang et al. (2023) propose CRF-GCN, a model that utilizes conditional random fields (CRF) to extract opinion scopes of specific aspect words and

integrates their contextual information into global nodes. These global nodes are then introduced into GCNs to effectively address the issue of fluctuating model accuracy in sentences with multiple aspect words.

Current GCNs algorithms can be divided into two categories: spectral-based and spatial-based methods. Bruna proposed the Spectrum CNN based on the convolutional theorem (Bruna et al., 2014) in 2014, which imitates the characteristics of convolutional neural networks by superimposing multi-layer graph convolutions, and defines convolutional kernels and activation functions for each layer, and form graph convolutional neural networks. Due to its high spatiotemporal complexity, Defferrard subsequently proposed ChebNet (Defferrard et al., 2016) in 2016 to reduce the temporal complexity by using the Chebyshev polynomial as a convolutional kernel. Due to the high complexity of eigenvalue decomposition of Laplace matrices, David and Hammond (2011) uses K-order truncation of Chebyshev polynomials instead of convolutional kernels, converts the modeling range of convolutional kernels from the entire graph to the K-order neighbors of the nodes, and reduces the number of parameters of convolutional kernels. Kipf and Welling (2017) proposes a hierarchical propagation method using a first-order approximation ChebNet, where each graph convolutional layer aggregates only first-order neighbors, and multiple graph convolutional layers can share a convolutional kernel, which can significantly reduce the number of parameters. With the increase of the number of layers, more information can be aggregated from distant neighbors. These methods are all defined in the perspective of the spectral features, while the spatial-based method appears earlier and it is more popular at present.

The core idea of the spatial-based approaches is to iteratively aggregate the features of neighbor nodes by defining aggregation functions, and then to update the features of the current nodes. In 2009, Gori proposed GNNs (Scarselli et al., 2009) method, which uses circular recursive functions as aggregate functions, and each node updates its own embedding by aggregating neighbor node information. In 2016, DCNN (Atwood and Towsley, 2016) regarded graph convolution as a diffusion procedure, and the information between nodes spreads with a certain probability. In 2017, Hamilton proposed GraphSAGE (Hamilton et al., 2017) method, which gives three aggregation functions to update the node state, such as mean aggregation, LSTM aggregation and pooling aggregation. Gilmer find that all spatial-based graph convolutional networks aggregate neighbor's state in some form to update the state of central node, so a framework MPNN (Gilmer et al., 2017) of spatial-based graph convolution is proposed for predicting chemical molecular properties. Under the inspiration of spectral-based graph convolutional network, the spatial-based graph convolutional network quickly become popular, and begin to develop toward a unified framework.

Nowadays, many scholars have solved numerous problems based on GCN. Li et al. (2023) propose DMRGCN, a novel bidirectional mutually reinforcing GCN, which investigates the semi-supervised node classification problem under noisy labels. Wang K. et al. (2023) propose mGNN, which extends the imbalanced classification concept in the field of machine learning to graph structures and effectively improves the classification performance of graph neural networks. Zhu et al. (2023) propose RGCNU, which maps the relationship between noisy monitoring data and uncertain residual life. Hou et al. (2021) propose ST Trader, which first uses VAE to reduce the dimensions of stock related information and convert it into a graph

structure. Then, GCN-LSTM is used to effectively predict stock movements. Despite the rapid development has got for GCNs, most of the GCNs are shallow structures. At present, some researchers have begun to train deep GCNs using different methods. GraphSAGE simultaneously uses node feature and structural feature to obtain graph embeddings, which is more scalable. In 2017, Pham proposed CLN (Pham et al., 2017) for relational classification, where model performance peaks when the depth of CLN reaches 10 layers, and model performance decreases as the increasing depth of CLN. In 2018, Rahimi et al. (2018) used GCN to integrate user text with network structures to achieve a more accurate geolocation of social media users. However, the authors find that the model performance gradually decreases when the depth of the Highway GCN is 6. Xu et al. (2018) proposes Jumping Knowledge Networks, which adjusts the range of the aggregated features according to different positions and structures of each node on graph, and the model is also limited to a six-layer structure. The number of graph convolutional layers limits the performance of the above GCNs, for example, the 10-layer graph convolution is superimposed, the model performance would decrease. In 2018, Li Q. et al. (2018) found that the biggest obstacle to training deep GCNs was over-smooth, and other research results (Zhou et al., 2018; Wu et al., 2019) also proved that the convolution operation of multi-layer graphs would lead to vanishing gradient. In order to alleviate the occurrence of over-smoothing phenomena, Li proposed DeepGCNs in 2019, which adds residual/dense connections to train deep GCNs inspired by deep CNNs, such as ResNet, DenseNet, etc. Klicpera is based on the intrinsic connections between GCNs and PageRank (Page et al., 1998; Klicpera et al., 2019a), it designs a propagation scheme based on personalized PageRank. In 2020, Rong and Zhao proposed DropEdge (Rong et al., 2019) and PairNorm (Zhao and Akoglu, 2020), respectively to migrate Dropout and BatchNorm to GCNs, which can also obtain better embedding and classification effects. Zhang et al. (2023) propose DRGCN, which utilizes dynamic block initialization information and employs evolution blocks to model the residual evolution patterns between layers. This approach effectively alleviates the over-smoothing issue in deep GCNs. Yang et al. (2023) propose EM-GCN, a model that introduces the expectation-maximization algorithm and utilizes approximate inference to overcome excessive smoothing in topology optimization for any GCN.

The approaches of making graph convolutional neural networks deeper can also be achieved through some alternative methods, for instance, we can design a high-order graph convolutional neural networks. Therefore, many researchers have adopted a shallow alternative method, it is that the GCNs consider higher-order neighbors in single-layer graph neural networks, for example, k-GNNs uses high-order Weisfeiler-Lehman for designing the graph neural networks (Morris et al., 2019), the MixHop solves the mixing problem of neighboring features at different distances (Abu-El-Hajja et al., 2019), and GDC (Klicpera et al., 2019b) enhances the performance of graph neural networks using graph diffusion. These high-order neural networks can usually obtain better embedding and classification effects, however, the above research is only available for deep neural networks based on single-feature training. For the complex information on graph, the deep structure of the current graph neural network using single feature cannot completely reveal the complex information of the graph. In order to reflect the diversity of the graph and avoid the over-smoothing phenomenon caused by enlarging

receptive field of the graph convolutional networks, this paper uses residual networks to construct a dual-channel deep graph convolutional network based on hybrid features of the graph.

3 Preliminaries

3.1 Graph convolutional network

In 2017, Kipf further proposed graph convolution with K -order Laplace polynomials as follows (Kipf and Welling, 2017):

$$Ug_{\theta}(\Lambda)U^T x \approx U \left(\sum_{\ell=0}^K \theta_{\ell} \Lambda^{\ell} \right) U^T x = \left(\sum_{\ell=0}^K \theta_{\ell} L^{\ell} \right) x. \quad (1)$$

Here, $\theta \in R^{K+1}$ is a polynomial coefficient vector. If $K=1$, $\theta_0=2\theta$, and $\theta_1=-\theta$, we can get a new convolution operation $g_{\theta} * x = \theta \left(I + D^{-1/2} A D^{-1/2} \right) x$. Through the renormalization tricks, the GCN will replace the expression $I + D^{-1/2} A D^{-1/2}$ with $\tilde{P} = \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} = (D + I_n)^{-1/2} (A + I_n) (D + I_n)^{-1/2}$, and it gets a graph convolutional layer as follows:

$$H^{(\ell+1)} = \sigma \left(\tilde{P} H^{(\ell)} W^{(\ell)} \right). \quad (2)$$

The σ is the activation function ReLU. Nodes aggregate high-order neighbor node information in GNNs, it would cause that nodes become indistinguishable with other nodes, and there exists a gradient vanishing during backpropagation. In order to avoid over-smoothing and gradient vanishing, DeepGCNs use the method adopted by deep CNNs to construct the deep structure of GCNs.

3.2 DeepGCNs

In 2019, Li introduced the method of training deep CNNs as ResNet, DenseNet, and dilated convolutions to propose DeepGCNs. DeepGCNs added the residual/dense network and dilated convolutions based on GCNs (Li et al., 2019).

$$H^{(\ell+1)} = \sigma \left(\tilde{P} H^{(\ell)} W^{(\ell)} \right) + H^{(\ell)}. \quad (3)$$

DeepGCNs use the information flow of different graph convolutional layers to reuse features between graph convolutional layers by dense connections.

$$H^{(\ell+1)} = \text{Concat} \left(\begin{array}{c} \sigma \left(\tilde{P} H^{(\ell)} W^{(\ell)} \right), \sigma \left(\tilde{P} H^{(\ell-1)} W^{(\ell-1)} \right), \dots, \\ \sigma \left(\tilde{P} H^{(0)} W^{(0)} \right), H^{(0)} \end{array} \right). \quad (4)$$

DeepGCNs obtain the surrounding nodes of the target node when the dilated rate d is determined by Dilated KNN:

$$N^{(d)}(v) = u_1, u_{1+d}, \dots, u_{1+(k-1)d}. \quad (5)$$

$(u_1, u_2, \dots, u_{k \times d})$ is an ordered set of $k \times d$ nearest neighbors. When the dilated rate is d , $(u_1, u_{1+d}, \dots, u_{1+(k-1)d})$ is the neighbor node of the node v . Since GCNs and DeepGCNs are single feature graph convolutional networks, and the graph contains much complex information, so the text features are integrated based on the structural features of the graph to enhance the embedding ability, which meets the diversity of graph information.

3.3 HDGCN

In 2021, Li (Yang et al., 2018) proposes HDGCN, which uses a dual channel GCNs structure to jointly model the structure and semantic features of nodes, complementing and enhancing the features of nodes as follows.

$$H_t^{(\ell+1)} = \sigma \left(P_t H_t^{(\ell)} W_t^{(\ell)} \right). \quad (6)$$

$\forall t \in \{1, 2, \dots, T\}$, the output matrix is $H_t \in \mathbb{R}^{N_t \times d}$, parameter matrix is $W_t \in \mathbb{R}^{d \times d}$.

Because the semantic features of the nodes contain the semantic features of weakly correlated neighboring nodes, it will become noise data that affects the training effect of the model. In order to reduce the noise interference on model training and enhance the feature embedding ability, the dual-channel GCN introduces Graph Attention Network and Gateway Recurrent Unit as follows:

$$z_{GAT}^v = \alpha_v h_v, \quad (7)$$

$$z_{GRU}^v = g_v \odot h_v. \quad (8)$$

Attention coefficient is

$$\alpha_v = \frac{\exp(e_v)}{\sum_{v \in \mathcal{V}} \exp(e_v)}. \quad (9)$$

The relevance is $e_v = a^T \cdot \text{Relu}(W \cdot h_v + b)$, the gating embedding is $g_v = \text{Sigmoid}(W \cdot h_v)$, $\forall v \in \{1, 2, \dots, N\}$. The a^T , W , and b are attention embeddings, weight parameters, and biases, respectively. According to the importance of each node, we give it a corresponding weight to reflect its importance. To improve the accuracy of graph convolutional networks based on hybrid feature, we expand the receptive field and aggregate the features of the higher-order neighboring nodes. To avoid over-smoothing, we also adopt the residual networks.

Since the text features and structural features have different influences on the model during training, three kinds of dual-channel GCNs models are proposed as follows.

4 Dual channel graph convolutional neural network framework

In this paper, the D2GCN is proposed, which aims at solving the fact that current GNNs cannot fully mine the high-order features based on dual-channel GNNs. In D2GCN, hybrid features are used as the inputs, and the feature matrix of each layer is reused for residual networks to avoid gradient vanishing in backpropagation. To enhance the feature embedding ability of nodes, the upstream node features are con-catenated as the input of the fully connected layer.

4.1 Definition

Graphs G can be represented by the triplets $G = (V, \varepsilon_1, \varepsilon_2, \phi, \varphi)$, where V are set of unordered nodes, ε_1 and ε_2 represent the edge sets of semantic networks and structural networks, $\phi: \varepsilon_1 \rightarrow R_1$ is a mapping function based on the node position relationships, $\varphi: \varepsilon_2 \rightarrow R_2$ is a mapping function based on the text content between nodes, R_1 and R_2 are a collection of node relationship types. A and D are adjacency matrix and degree matrix, respectively. $X \in \mathbb{R}^{n \times d}$ is the feature matrix of the node, each of these nodes corresponds to a d dimension feature embedding X_v . Regularized Laplace matrices are semi-definite symmetric matrix, defined as $L = I_n - D^{-1/2} A D^{-1/2}$. Its eigenvalues are decomposed into $U \Lambda U^T$. Λ is a eigenvalues diagonal matrix of L , $U \in \mathbb{R}^{n \times n}$ is a unitary matrix composed of eigenvectors of L . The graph convolution operation between the input signal x and the filter $g_\gamma(\cdot) = \text{diag}(\gamma)$ is defined as $g_\gamma(L) * x = U g_\gamma(\Lambda) U^T x$, $\gamma \in \mathbb{R}^n$ is the coefficient vector corresponding to the filters.

The following three networks are the data basis of modeling in this paper, so the following explanations are given as follows:

Semantic network: semantic network refers to a network composed of texts relationships. If two nodes have the same words in their text features, a new edge will be added to these two nodes.

Structure network: structure network is the original network structure, in which no new nodes and edges are added.

Hybrid network: semantic network and structural network are combined to form a hybrid network.

4.2 Dual-channel deep graph convolutional neural network

4.2.1 D2GCN based on structural features [D2GCN_(structure)]

The framework of D2GCN based on structural features is shown in Figure 1.

In this model, the semantic network is trained by shallow neural network, and the structural network is trained by deep neural network. Finally, the two different embeddings are integrated.

The graph based on hybrid features is used as the input of the model, and the input is divided into two single feature graphs through the dual-channel GCNs: graph based on structural features and graph based on text features, and node embeddings of two different types of features is obtained. Because the graph based on text features may contain many weak correlations between nodes, it becomes noise that affects the performance of the model, and the

shallow structure of the graph convolutional networks only aggregates the neighboring node features. Therefore, the text features of the node are used as the input of the shallow structure of the graph convolutional network. In addition, the graph based on the structural features is used as the input of the deep graph convolutional networks, because the deep graph convolutional networks can aggregate high-order features. When the scale of the graph convolutional network aggregation is gradually expanded to all nodes of the graph, the features of the node will become indistinguishable, causing the gradient vanishing during the period of backpropagation. With the increase of the graph convolutional layer, the output of the graph convolutional layer is repeatedly reused using residual networks to ensure the difference, the D2GCN_(structure) can be formulated as follows:

$$H_{(structure)}^{(l+1)} = \sigma \left(\tilde{P}H_{(structure)}^{(l)} W^{(l)} \right) + H_{(structure)}^{(l)}. \quad (10)$$

According to theoretical analysis, the main reason for the complexity of the D2GCN_(structure) model is the residual connection of the structural network. The main reason for the complexity of the D2GCN_(structure) model is the residual connection of the structural network. According to formula 10, the complexity of $\tilde{P}H_{(structure)}^{(l)} W^{(l)}$ is $O(|V|^3)$, where V is the number of network nodes and the complexity of $H_{(structure)}^{(l)}$ is $O(|V|)$. Therefore, the time complexity of D2GCN_(structure) is $O(|V|^3)$.

4.2.2 D2GCN based on semantic features [D2GCN_(semantic)]

The framework of D2GCN based on semantic features is shown in Figure 2.

In this model, the structural network is trained by shallow neural network, and the semantic network is trained by deep neural network. Finally, the two different embeddings are integrated.

For large-scale sparse graphs, nodes belonging to the same type may not have a neighboring relationship or even a weak correlation. However, there is a greater probability that a node belongs to the same type as its neighbors. Thus, the features around the nodes are aggregated by the shallow structure of the graph convolutional networks. Because the graph based on text features is more dense than the graph based on structural features, many nodes of the same type without edges in the sparse graph based on structural features may establish direct/indirect connections in D2GCN_(semantic). Therefore, the graph convolution operation is used repeatedly to obtain the global structure based on the text feature graph. The local structure and global structure of the graph are fused to improve the accuracy of the downstream classification tasks. The specific formula is as follows:

$$H_{(semantic)}^{(l+1)} = \sigma \left(\tilde{P}H_{(semantic)}^{(l)} W^{(l)} \right) + H_{(semantic)}^{(l)}. \quad (11)$$

Similar to the complexity analysis of D2GCN_(structure), the time complexity of D2GCN_(semantic) is $O(|V|^3)$.

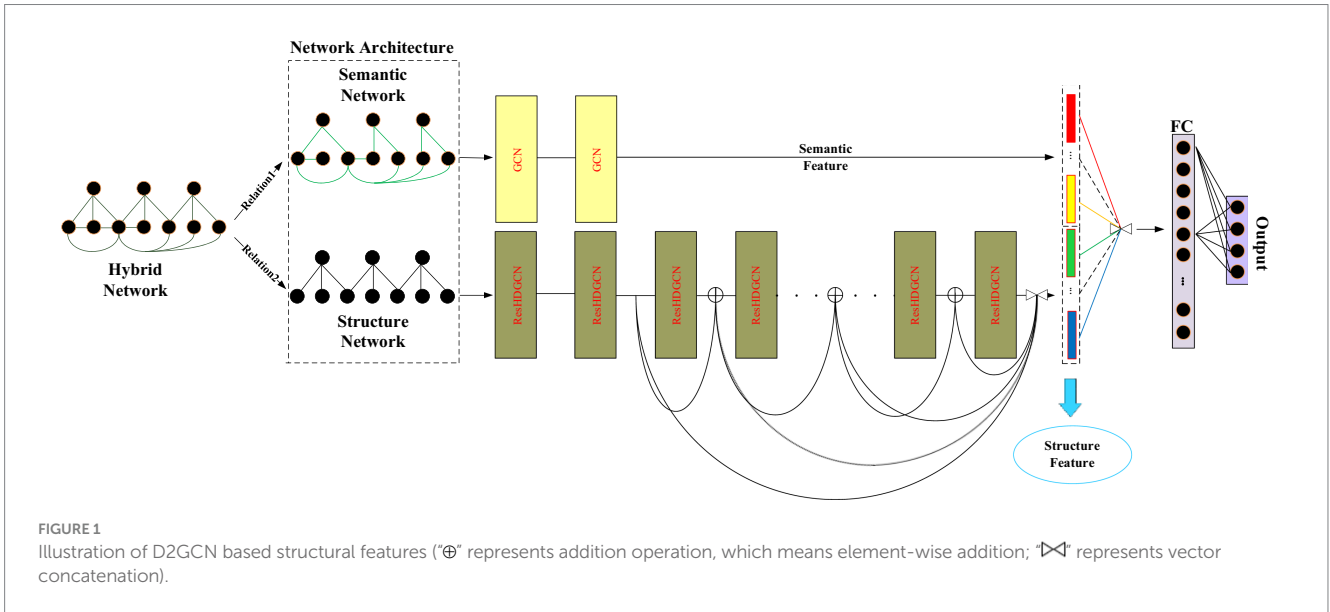


FIGURE 1 Illustration of D2GCN based structural features (“⊕” represents addition operation, which means element-wise addition; “⊔” represents vector concatenation).

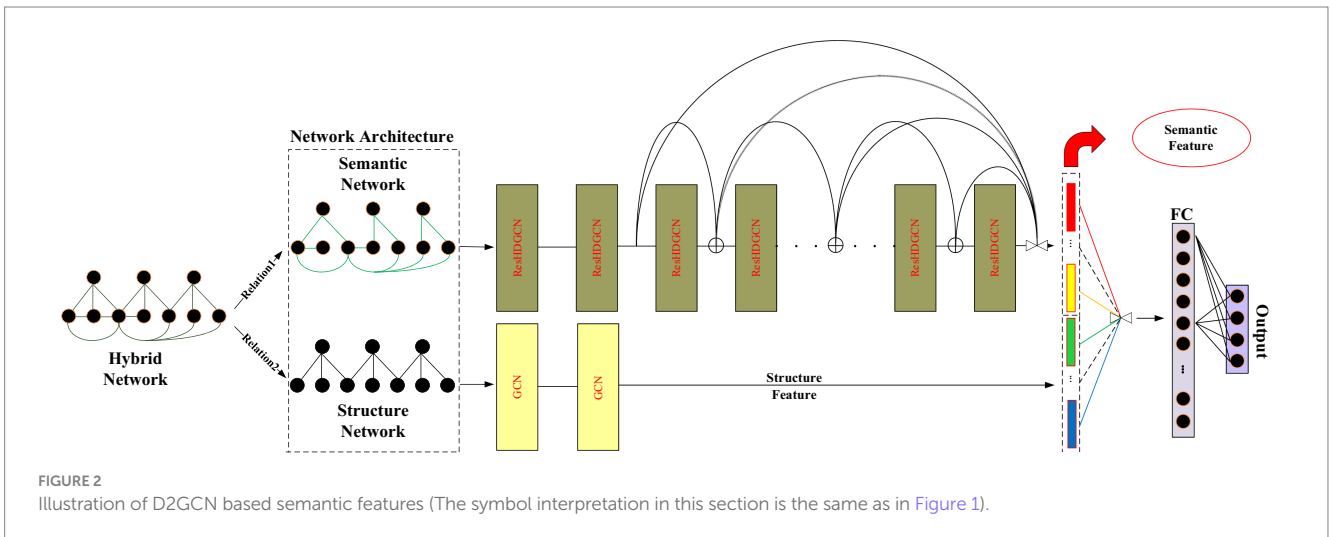


FIGURE 2 Illustration of D2GCN based semantic features (The symbol interpretation in this section is the same as in Figure 1).

4.2.3 D2GCN based on hybrid features [D2GCN (hybrid)]

The framework of D2GCN based on hybrid features is shown in Figure 3.

In this model, the structural network is trained by deep neural network, and the semantic network is also trained by deep neural network. Finally, the two different embeddings are integrated.

The probability that the interconnected nodes in the sparse graph belong to the same type decreases with the increase of distance, so the output of the shallow structure of the residual networks that reuses graph convolutional neural network is used to obtain the local graph structure based on text or structural features. The cross-reuse of text features and structural features enables features to complement each other and enhance the embedding ability of node features. The specific formula is as follows:

$$\begin{aligned}
 H_{(semantic)}^{(1)} &= \sigma \left(\tilde{P}H_{(semantic)}^{(0)}W^{(0)} \right) + H_{(semantic)}^{(0)}, H_{(structure)}^{(1)} \\
 &= \sigma \left(\tilde{P}H_{(structure)}^{(0)}W^{(0)} \right) + H_{(structure)}^{(0)}, \quad (12)
 \end{aligned}$$

$$\begin{aligned}
 H_{(semantic)}^{(2)} &= H_{(structure)}^{(1)} + H_{(semantic)}^{(2)}, H_{(structure)}^{(2)} \\
 &= H_{(semantic)}^{(1)} + H_{(structure)}^{(2)}. \quad (13)
 \end{aligned}$$

Then the deep graph convolutional network is constructed by the residual network, and the local structure of the graph obtained by the convolutional network of the previous layer of graph is used as input to obtain the global structure of the graph. Finally, the global structure of the two features is fused to obtain a probability matrix based on the hybrid features $H_{(hybrid)}^{(l+1)}$. The specific formula is as follows:

$$H_{(semantic)}^{(l+1)} = \sigma \left(\tilde{P}H_{(semantic)}^{(l)}W^{(l)} \right) + H_{(semantic)}^{(l)}, l \geq 2, \quad (14)$$

$$H_{(structure)}^{(l+1)} = \sigma \left(\tilde{P}H_{(structure)}^{(l)}W^{(l)} \right) + H_{(structure)}^{(l)}, l \geq 2, \quad (15)$$

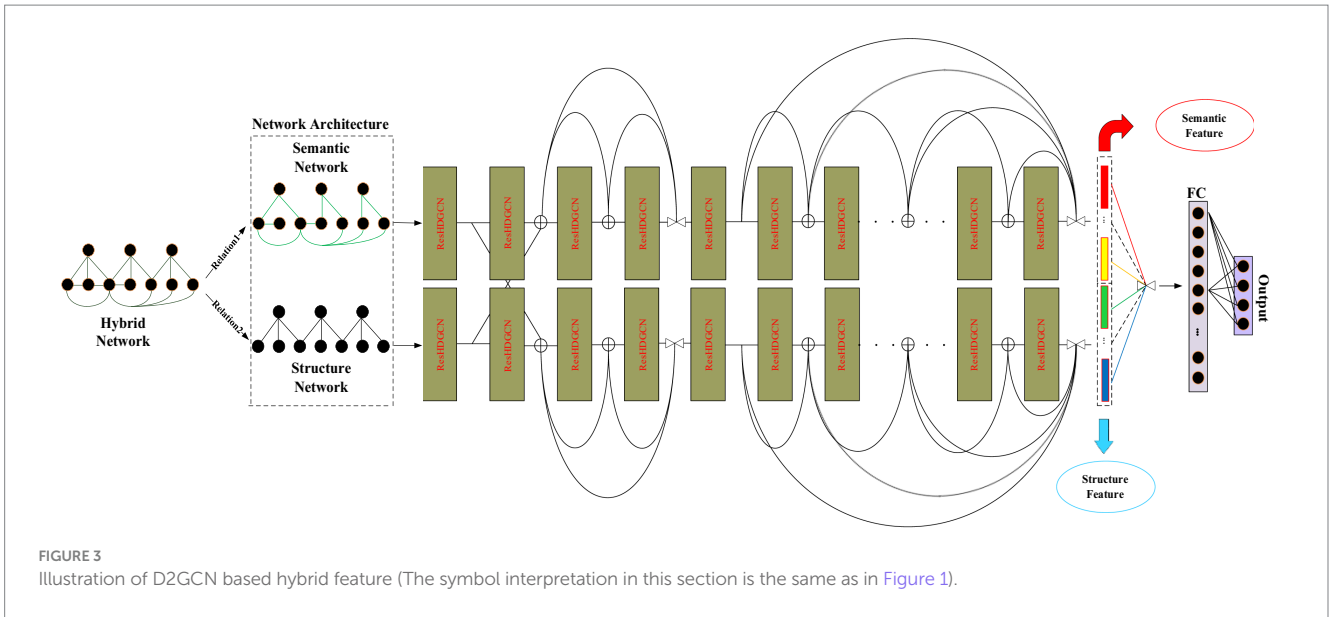


TABLE 1 Dataset description.

Dataset	CiteSeer		SDBLP		DBLP	
	Structure	Semantic	Structure	Semantic	Structure	Semantic
Node	4,610	4,610	3,119	3,119	17,725	17,725
Edge	5,923	819,346	39,516	439,182	105,781	125,360
Training	1,333	1,333	666	666	3,333	3,333
Validation	667	667	334	334	1,667	1,667
Test	2,610	2,610	2,119	2,119	12,725	12,725

$$H_{(hybrid)}^{(l+1)} = \text{concat} \left(H_{(semantic)}^{(l+1)}, H_{(structure)}^{(l+1)} \right), l \geq 2, \quad (16)$$

$$H_{(hybrid)}^{(l+1)} = \sigma \left(\tilde{P} H_{(hybrid)}^{(l)} W^{(l)} \right) + H_{(hybrid)}^{(l)}. \quad (17)$$

Due to the fusion structure of D2GCN_(hybrid), both channels are residual connected, resulting in a time complexity of $O(2|V|^3)$ and simplified to $O(|V|^3)$.

5 Experimental results and analysis

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation, as well as the experimental conclusions that can be drawn.

5.1 Datasets

To assess the effectiveness of D2GCN, this paper uses three reference network datasets, such as CiteSeer, DBLP, and SDBLP. The statistics of the datasets are shown in Table 1. Each dataset is divided

into semantic networks and structural networks. In the semantic network, the edge connection relationship between nodes is constructed according to the word co-occurrence, if the same word appears in the text of each node, there is an edge link between nodes. In the structural network, the relationship between nodes is determined according to the citation relationship between different documents. SDBLP is a simplified dataset of DBLP, in which nodes with less than 3 references are deleted, it is that nodes with node degree less than 3 will be deleted.

5.2 Baselines

This paper compares D2GCN with the following baseline methods designed to generate node embedding:

DeepGCNs_(structure): this model inputs the structural features to Current DeepGCNs.

DeepGCNs_(semantic): this model inputs the semantic features to Current DeepGCNs.

D2GCN_(JKNet): the use of initial residuals and identity mappings can solve the problem of over-smooth. In each layer, the initial residuals construct a jump connection from the input layer, while the identity map adds the identity matrix to the weight matrix. When increasing the depth of the model, these two techniques can prevent over-smooth and continuously improve the performance of the model.

D2GCN_(Drop): it proposes a random removal edge strategy using DropEdge by a certain ratio, DropEdge increases the diversity of the inputs to prevent overfit, and alleviate over-smooth.

HDGCN: it is similar to the dual-channel deep graph neural network based on hybrid features proposed in this paper, but the depth of the model is only two layers, which is a shallow dual channel graph neural network.

D2GCN_(structure): the dual-channel deep graph convolutional neural network proposed in this paper only considers to model the structural features of structure network.

D2GCN_(semantic): the dual-channel deep graph convolutional neural network proposed in this paper only considers to model the text features of semantic network.

D2GCN_(hybrid): the dual-channel deep graph convolutional neural network proposed in this paper considers to model the hybrid features integrated by the structural and semantic features.

5.3 Semi-supervised node classification

This paper compares D2GCN with the following baseline methods designed to generate node embedding.

For the semi-supervised node classification task, we randomly divide datasets into training/validation/test on CiteSeer, SDBLP, and DBLP datasets. This paper applies two Deep GNN models, which are GCNII and DropEdge, to HDGCN and GCN. We use Adam SGD as an optimizer to train D2GCN, and we set learning rate as 0.01, iterations as 200, dropout ratio as 0.5, weight decay rate as 0.0005, and embed dimension as 64. In the experiment, this paper uses D2GCN to train hyper-parameters of the model, and divides the training set into several small batches of data to update the parameters. In order to reduce the error, this paper repeats the experiment for 10 times, and the average value of the accuracy is shown in Table 2.

As shown in Table 2, the convolution operation of D2GCN_(hybrid) at 2, 4, 8, and 16 layers obtains the optimal values compared with the baseline methods on the CiteSeer dataset, and the model performance increases with the increasing number of graph convolutional layers. The model performs best when the graph convolutional layer reaches 16. The baseline methods shows that the model performance first increases and then decreases with the increasing number of convolutional layers for the dual-channel deep GCNs, resulting in little significance for the deep structure of the model based on hybrid features.

Since the SDBLP is a dense network, the probability that the central node and its surrounding nodes belong to the same type decreases as the increasing distance between the nodes. Therefore, the shallow GCN performs best in two-layer graph convolutional operation compared to D2GCN_(structure), and the text features contain many weakly correlated features, which aggregate the information of the surrounding node as noise data to interfere center node, and classify the center node and its unrelated nodes into the same category. Therefore, D2GCN_(structure) is better classified than D2GCN_(semantic). However, the GCN has over-smooth phenomenon and the model performance has gradually decreased with the increasing layer of the graph convolution. Conversely, D2GCN_(structure) aggregates the features of high-order neighboring nodes, and with the increase of convolutional layers, the accuracy of the model in the classification task is continuously improved. The other baseline methods show a trend that the accuracy first rises and then decreases with the increase

of the convolutional layers. The performance of D2GCN_(structure) is the best when the depth of graph convolution operation is 16.

On the DBLP, deep graph convolutional neural network D2GCN_(semantic) based on text feature is better than deep graph convolutional neural network D2GCN_(structure) based on structural features. Because the number of neighboring nodes of the central node is smaller, but the probability that they belong to the same type is higher. Therefore, the baseline methods are better than the proposed D2GCN when the number of convolution layer is 2. However, D2GCN also increases model performance as the increasing number of graph convolutional layers. Other baseline methods show a tendency that the accuracy first rises and then decreases in the classification task, or show an unstable phenomenon of alternating ascent and descent.

5.4 Visualization

To further illustrate the effectiveness of D2GCN, this paper conducts a set of visualization experiments. We use t-SNE to map the embeddings of nodes into 2D space on CiteSeer, the embeddings of different depths obtained through D2GCN are shown in Figure 4, and different colors represent to different node label. Through visual experiments, it can be seen that the D2GCN gets better and better as the increasing number of convolutional layers. Specifically, the internal similarity becomes higher and higher, and the boundaries between different node labels are clearer as the increasing depth of the model.

5.5 Hyperparameter analysis

This paper performs a sensitivity analysis for some main hyperparameters in D2GCN, as shown in Figure 5.

5.5.1 Final embedding dimension F

This paper first tests the effect of the final embedding dimension F , as shown in Figure 5A, when the embedding dimension is 32, the model performance increases with the increase of the graph convolutional layer. When the embedding dimension is 8, the performance decreases as the increasing number of convolutional layers. And then when the convolutional layer increases to 16 layers, the model performance improves best. Model performance of other embedding methods improves with the increase of graph convolutional layers, and then the model performance decreases when the 16-layer graph convolutional operation is performed. When the dimension is 32, the model performs best in the node classification task.

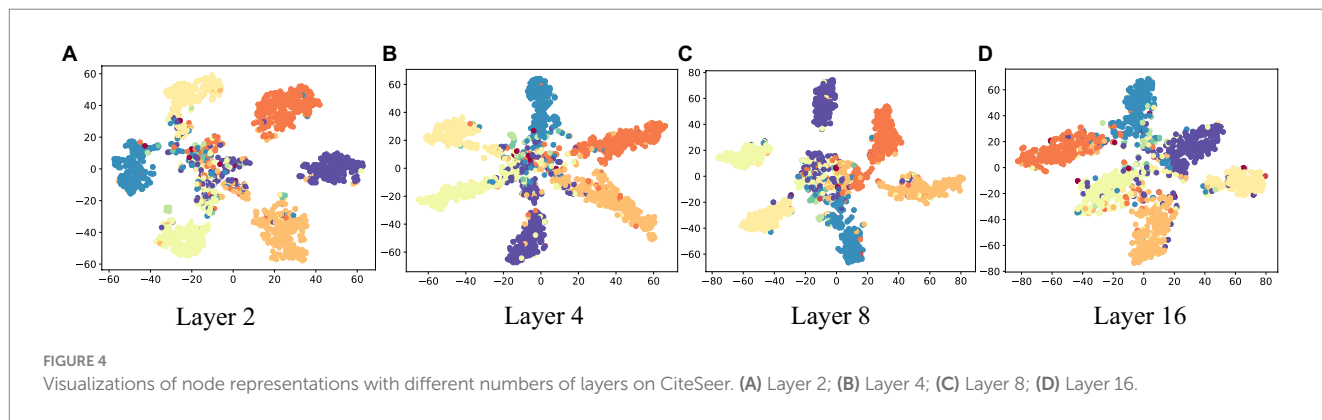
5.5.2 Convolution hyperparameter K

In this paper, the influence of the K -order approximation of the local spectral filter on the model is studied, as shown in Figure 5B, the model performs stably in the classification task with the increase of the graph convolutional layer. When the graph convolutional layer is 8, the model performance begins to decrease as the increases of K . When the graph convolutional layer reaches 16, the model performance drops significantly, and at that time, the model performance drops the fastest. However, when the hierarchical convolution operation is performed, the model performance increases with the increase of the graph convolutional layer, and when the graph convolutional layer is 16, the model performance reaches the best. Therefore, D2GCN, restricts hierarchical convolution operations.

TABLE 2 Node classification accuracy with various depts.

Dataset	Model	Layers			
		2	4	8	16
CiteSeer	Deep GCNs _(structure)	77.63	78.12	78.36	77.74
	Deep GCNs _(semantic)	67.45	67.33	66.51	18.58
	D2GCN _(JKNet)	81.99	85.13	86.63	85.71
	D2GCN _(Drop)	83.49	85.33	85.17	84.64
	HDGCN	82.30	-	-	-
	D2GCN _(structure)	84.15	84.55	85.48	83.41
	D2GCN _(semantic)	84.15	84.36	84.55	84.95
	D2GCN _(hybrid)	84.17	86.43	86.77	86.99
SDBLP	Deep GCNs _(structure)	82.41	82.21	82.51	80.15
	Deep GCNs _(semantic)	70.03	65.08	34.50	27.70
	D2GCN _(JKNet)	81.31	81.55	82.23	82.21
	D2GCN _(Drop)	81.93	82.40	81.69	82.11
	HDGCN	81.50	-	-	-
	D2GCN _(structure)	81.81	82.14	82.36	83.47
	D2GCN _(semantic)	81.78	82.36	82.47	83.11
	D2GCN _(hybrid)	81.62	81.97	82.17	82.93
DBLP	DeepGCNs _(structure)	80.70	79.91	79.12	79.80
	DeepGCNs _(semantic)	67.42	60.70	44.57	44.78
	D2GCN _(JKNet)	80.46	79.03	79.76	79.01
	D2GCN _(Drop)	80.42	80.65	80.65	80.28
	HDGCN	80.28	-	-	-
	D2GCN _(structure)	80.37	80.17	78.29	78.10
	D2GCN _(semantic)	80.09	80.33	80.61	80.88
	D2GCN _(hybrid)	80.55	80.30	80.18	80.21

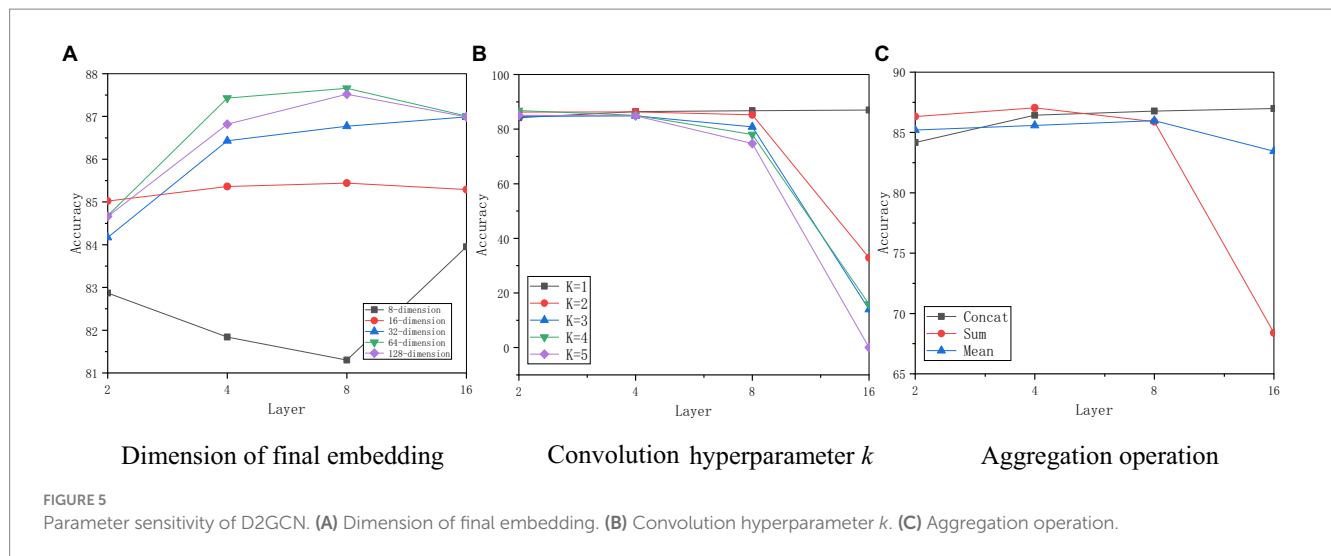
The data in bold in table represents all the results of the best model corresponding to a certain dataset.



5.5.3 Feature aggregation operation

In this paper, different feature aggregation operations are also studied, as shown in Figure 5C, the model performance is improved fast where the embeddings of semantic and structural networks are concatenated. However, the model uses the fusion approach of averaging and summing methods to lead to performance degradation as the increasing number of convolutional layers, when the number of graph convolutional layer is 16, the model

performance decreases the most, especially the summing method. The simple concatenation operation continuously improves the model performance as the increasing number of graph convolutional layers, and the model performs best when the number of graph convolutional layer reaches 8. When the number of convolutional layer reaches 16, its performance is the best. Therefore, this paper uses concatenation method to fuse semantic features and structural features.



6 Conclusion

In this paper, a dual-channel deep graph convolutional neural network (D2GCN) based on hybrid features is proposed. According to the text features, the residual connection is used to construct the deep graph convolution neural network, and for the structural features of the graph, the two-layer graph convolution neural network is used, which is the $D2GCN_{(semantic)}$. $D2GCN_{(structure)}$ trains a deep graph convolutional neural network with the structural features of the graph, in contrast, the shallow structure of the graph convolutional neural network is trained based on the text features of the graph. In this paper, $D2GCN_{(structure)}$ is constructed using residual networks, and the two features of the graph are fused by concatenating strategy. $D2GCN_{(hybrid)}$ uses the text features and structural features of the graph to simultaneously train a dual-channel deep graph convolutional neural network constructed by residual networks, in which the output of the graph convolutional neural network is cross-reused, so that the two features complement each other and improve the performance of the model in the node classification task. The experimental results in this paper demonstrate the effectiveness of D2GCN in node classification task. As an efficient way to improve model performance, it is a potential research how to incorporate pre-training into D2GCN.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: <http://www.cs.umd.edu/~sen/lbc-proj/data/cora.tgz>; <http://www.cs.umd.edu/~sen/lbc-proj/data/citeseer.tgz>.

References

- Abu-El-Hajja, S., Perozzi, B., Kapoor, A., Alipourfard, N., Lerman, K., Harutyunyan, H., et al. (2019). MixHop: higher-order graph convolutional architectures via sparsified neighborhood mixing. arXiv [Preprint]. doi: 10.48550/arXiv.1905.00067
- Armeni, I., Sax, S., Zamir, A., and Savarese, S. (2017). Joint 2D-3D-semantic data for indoor scene understanding. arXiv [Preprint]. doi: 10.48550/arXiv.1702.01105
- Atwood, J., and Towsley, D. (2016). "Diffusion-convolutional neural networks" in Proceedings of the 30th International Conference on Neural Information Processing Systems. Cambridge: MIT Press. 1993–2001.
- Bastings, J., Titov, I., Aziz, W., Marcheggiani, D., and Simaan, K. (2017). "Graph convolutional encoders for syntax-aware neural machine translation" in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 1957–1967.
- Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. (2014). Spectral networks and locally connected networks on graphs. arXiv [Preprint]. doi: 10.48550/arXiv.1312.6203
- Chen, L., and Zhang, Q. (2023). DDGCN: graph convolution network based on direction and distance for point cloud learning. *Vis. Comput.* 39, 863–873. doi: 10.1007/s00371-021-02351-8

Author contributions

ZY: Writing – original draft, Data curation, Methodology, Validation. ZL: Writing – original draft. GL: Writing – review & editing. HZ: Supervision, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work is supported by the National Key Research and Development Program of China (No. 2020YFC1523300) and Innovation Platform Construction Project of Qinghai Province (2022-ZJ-T02).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

- David, K., and Hammond, A. (2011). Wavelets on graphs via spectral graph theory. *Appl. Comput. Harmon. Anal.* 30, 129–150. doi: 10.1016/j.acha.2010.04.005
- Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). “Convolutional neural networks on graphs with fast localized spectral filtering” in *Proceedings of the 30th International Conference on Neural Information Processing Systems*. MIT Press, Cambridge. 3844–3852.
- Firdaus, M., Singh, G. V., Ekbal, A., and Bhattacharyya, P. (2023). Affect-GCN: a multimodal graph convolutional network for multi-emotion with intensity recognition and sentiment analysis in dialogues. *Multimed. Tools Appl.* 82, 43251–43272. doi: 10.1007/s11042-023-14885-1
- Gilmer, J. S., Schoenholz, S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). “Neural message passing for quantum chemistry” in *Proceedings of the 34th International Conference on Machine Learning*. ACM, New York. 1263–1272.
- Hamilton, W. L., Ying, R., and Leskovec, J. (2017). “Inductive representation learning on large graphs” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*. MIT Press, MA. 1025–1035.
- He, K. M., Zhang, X. Y., Ren, S. Q., and Sun, J. (2016). “Deep residual learning for image recognition” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, NJ. 770–778.
- Hinton, G., Deng, L., Yu, D., Dahl, G. E., and Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Process. Mag.* 29, 82–97. doi: 10.1109/MSP.2012.2205597
- Hou, X., Wang, K., Zhong, C., and Wei, Z. (2021). ST-trader: a spatial-temporal deep neural network for modeling stock market movement. *IEEE/CAA J. Automat. Sin.* 8, 1015–1024. doi: 10.1109/JAS.2021.1003976
- Hu, B. T., Lu, Z. D., Hang, L., and Chen, Q. C. (2014). “Convolutional neural network architectures for matching natural language sentences” in *Proceedings of the Advances in Neural Information Processing Systems*. NIPS, California. 2042–2050.
- Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). “Densely connected convolutional networks” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, NJ. 4700–4708.
- Huang, B., Zhang, J., Ju, J., Guo, R., Fujita, H., and Liu, J. (2023). CRF-GCN: An effective syntactic dependency model for aspect-level sentiment analysis. *Knowl.-Based Syst.* 260:110125. doi: 10.1016/j.knsys.2022.110125
- Jain, A., Zamir, A. R., Savarese, S., and Saxena, A. (2016). “Structural-RNN: Deep learning on spatio-temporal graphs” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, NJ. 5308–5317.
- Jiang, J., Lu, X., Ouyang, W., and Wang, M. (2023). Unsupervised contrastive learning with simple transformation for 3D point cloud data. *Vis. Comput.* 1–18. doi: 10.1007/s00371-023-02921-y
- Khodadad, M., Rezanejad, M., Kasmaee, A. S., Siddiqi, K., Walther, D., Mahyar, H. (2023). MLGCN: An ultra efficient graph convolution neural model for 3D point cloud analysis. arXiv [Preprint]. doi: 10.48550/arXiv.2303.17748
- Kipf, T. N., and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. arXiv [Preprint]. doi: 10.48550/arXiv.1609.02907
- Klicpera, J., Bojchevski, A., and Günnemann, S. (2019a). Predict then propagate: graph neural networks meet personalized pagerank. arXiv [Preprint]. doi: 10.48550/arXiv.1810.05997
- Klicpera, J., Weissenberger, S., and Günnemann, S. (2019b). “Diffusion improves graph learning” in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. NIPS, NY. 13366–13378.
- Li, Q., Han, Z., and Wu, X.-M. (2018). “Deeper insights into graph convolutional networks for semi-supervised learning” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI, Menlo Park.
- Li, G., Müller, M., Thabet, A., and Ghanem, B. (2019). “Deep GCNs: Can GCNs go as deep as CNNs?” in *Proceedings of the 2019 IEEE International Conference on Computer Vision*. IEEE, NJ. 9266–9275.
- Li, Y., Ouyang, W., Zhou, B., Shi, J., Zhang, C., and Wang, X. (2018). “Factorizable net: an efficient subgraph-based framework for scene graph generation” in *Proceedings of the European Conference on Computer Vision*. 335–351.
- Li, B., Wu, J., Pi, D., and Lin, Y. (2023). Dual mutual robust graph convolutional network for weakly supervised node classification in social networks of internet of people[J]. *IEEE Internet Things J.* 10, 14798–14809. doi: 10.1109/JIOT.2021.3091883
- Marcheggiani, D., and Titov, I. (2017). “Encoding sentences with graph convolutional networks for semantic role labeling” in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 1506–1515
- Monti, F., Bronstein, M., and Bresson, X. (2017a). “Geometric matrix completion with recurrent multi-graph neural networks” in *Proceedings of the Advances in Neural Information Processing Systems*. NIPS, California. 3697–3707.
- Monti, F., Bronstein, M., and Bresson, X. (2017b). “Geometric matrix completion with recurrent multi-graph neural networks” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*. ACM, NY. 3700–3710.
- Morris, C., Ritzert, M., Fey, M., Hamilton, W., Lenssen, J., Rattan, G., et al. (2019). “Weisfeiler and leman go neural: higher-order graph neural networks” in *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI, Menlo Park. 4602–4609
- Page, L., Brin, S., Motwani, R., and Winograd, T. (1998). The pagerank citation ranking: Bringing order to the web. Stanford Digital Libraries Working Paper.
- Pham, T., Tran, T., Phung, D., and Venkatesh, S. (2017). “Column networks for collective classification” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI, Menlo Park.
- Qi, X., Liao, R., Jia, J., Fidler, S., and Urtasun, R. (2017). “3D graph neural networks for RGBD semantic segmentation” in *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, NJ. 5199–5208.
- Rahimi, A., Cohn, T., and Baldwin, T. (2018). Semi-supervised user geolocation via graph convolutional networks. arXiv [Preprint]. doi: 10.48550/arXiv.1804.08049
- Rong, Y., Huang, W., Xu, T., and Huang, J. (2019). DropEdge: towards deep graph convolutional networks on node classification. arXiv [Preprint]. doi: 10.48550/arXiv.1907.10903
- Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2009). The graph neural network model. *IEEE Trans. Neural Netw.* 20, 61–80. doi: 10.1109/TNN.2008.2005605
- Tang, L., and Liu, H. (2009). “Relational learning via latent social dimensions” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. NY: ACM, 817–826.
- Wale, N., Watson, I. A., and Karypis, G. (2008). Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowl. Inf. Syst.* 14, 347–375. doi: 10.1007/s10115-007-0103-5
- Wang, K., An, J., Zhou, M., Shi, Z., Shi, X., and Kang, Q. (2023). Minority-weighted graph neural network for imbalanced node classification in social networks of internet of people[J]. *IEEE Internet Things J.* 10, 330–340. doi: 10.1109/JIOT.2022.3200964
- Wang, L., Song, Z., Zhang, X., Wang, C., Zhang, G., Zhu, L., et al. (2023). SAT-GCN: self-attention graph convolutional network-based 3D object detection for autonomous driving. *Knowl.-Based Syst.* 259:110080. doi: 10.1016/j.knsys.2022.110080
- Wang, Y., Sun, Y. B., Liu, Z. W., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. (2019). Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.* 38, 1–12. doi: 10.1145/3326362
- Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. (2019). S. A comprehensive survey on graph neural networks. arXiv [Preprint]. doi: 10.48550/arXiv.1901.00596
- Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K. I., and Jegelka, S. (2018). Representation learning on graphs with jumping knowledge networks. arXiv [Preprint]. doi: 10.48550/arXiv.1806.03536
- Xu, D., Zhu, Y., Choy, C. B., and Fei-Fei, L. (2017). “Scene graph generation by iterative message passing” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, NJ. 5410–5419.
- Yan, S., Xiong, Y., and Lin, D. (2018). “Spatial temporal graph convolutional networks for skeleton-based action recognition” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.
- Yang, R., Dai, W., Li, C., Zou, J., and Xiong, H. (2023). Tackling over-smoothing in graph convolutional networks with EM-based joint topology optimization and node classification. *IEEE Trans. Signal Info. Process. Netw.* 9, 123–139. doi: 10.1109/TSIPN.2023.3244112
- Yang, J., Lu, J., Lee, S., Batra, D., and Parikh, D. (2018). “Graph RCNN for scene graph generation” in *Proceedings of the European Conference on Computer Vision*, 670–685.
- Ying, R., He, R. N., Chen, K. F., Eksombatchai, P., Hamilton, W. L., and Leskovec, J. (2018). “Graph convolutional neural networks for web-scale recommender systems” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, NY. 974–983.
- Yu, F., and Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. arXiv [Preprint]. doi: 10.48550/arXiv.1511.07122
- Zhang, Z., Ma, Z., Cai, S., Chen, J., and Xue, Y. (2022). Knowledge-enhanced Dual-Channel GCN for aspect-based sentiment analysis. *Mathematics* 10:4273. doi: 10.3390/math10224273
- Zhang, L., Yan, X., He, J., Li, R., and Chu, W. (2023). DRGCN: dynamic evolving initial residual for deep graph convolutional networks. arXiv [Preprint]. doi: 10.48550/arXiv.2302.05083
- Zhao, L., and Akoglu, L. (2020). PairNorm: tackling oversmoothing in GNNs. arXiv [Preprint]. doi: 10.48550/arXiv.1909.12223
- Zhao, Z., Yang, Z., Li, C., Zeng, Q., Guan, W., and Zhou, M. (2023). Dual feature interaction-based graph convolutional network. *IEEE Trans. Knowl. Data Eng.* 35, 9019–9030. doi: 10.1109/TKDE.2022.3220789
- Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., and Sun, M. (2018). Graph neural networks: a review of methods and applications. arXiv [Preprint]. doi: 10.48550/arXiv.1812.08434
- Zhu, Q., Xiong, Q., Yang, Z., and Yu, Y. (2023). RGCNU: recurrent graph convolutional network with uncertainty estimation for remaining useful life prediction. *IEEE/CAA J. Automat. Sin.* 10, 1640–1642. doi: 10.1109/JAS.2023.123369
- Zitnik, M., and Leskovec, J. (2017). Predicting multicellular function through multi-layer tissue networks. *Bioinformatics* 33, i190–i198. doi: 10.1093/bioinformatics/btx252