Computer Modeling in Engineering & Sciences

Tech Science Press

https://doi.org/10.32604/cmes.2026.083131

ARTICLE

D2GSL: Self-Supervised Dual-Layer Structure-Driven Graph Structure Learning

Juncheng Zhang[1,2], **Xuhao Wei**[1,2], **Xiaolei Gu**[3], **Haixing Zhao**[4,*] **and Zhonglin Ye**[1,2,5,*]

[1]School of Computer, Qinghai Normal University, Xining, China
[2]State Key Laboratory of Tibetan Intelligent, Xining, China
[3]Qinghai Provincial Public Security Department, Xining, China
[4]College of Intelligent Science and Engineering, Qinghai Minzu University, Xining, China
[5]Graduate School of Engineering, Nagasaki Institute of Applied Science, Nagasaki, Japan

*Corresponding Authors: Haixing Zhao. Email: h.x.zhao@163.com; Zhonglin Ye. Email: yezhonglin@qhnu.edu.cn

Received: 30 March 2026; Accepted: 17 July 2026; Published: 28 August 2026

ABSTRACT: Graph structure learning depends heavily on the integrity and reliability of graph data. However, real-world graphs often contain noise, missing information, and bias, thereby limiting the expressive capacity of existing models. Single-layer structure learning methods fail to simultaneously capture local interactions and the global structure. Furthermore, they rely excessively on high-quality labeled data, leading to label scarcity issues and high annotation costs. To address these challenges, we propose a self-supervised dual-layer structure-driven graph structure learning method, termed D2GSL. Specifically, D2GSL constructs a semantic similarity channel and a spectral feature channel to model node relationships from both local semantic and global spectral views. It introduces a hyperadjacency matrix that explicitly models inter-layer node correspondences and enables joint structural reconstruction across channels. The framework further applies structural reconstruction constraints and adopts a contrastive learning mechanism to enhance structural representations in a self-supervised setting. Comprehensive experimental results demonstrate that D2GSL consistently outperforms mainstream baseline models on public benchmark datasets and exhibits remarkable efficacy under label-scarcity conditions.

KEYWORDS: Graph structure learning; dual-layer structure; hyperadjacency matrix; graph contrastive learning

1 Introduction

In recent years, graph-based learning methods have been widely applied across multiple domains, including social networks [1,2], transportation networks [3,4], and bioinformatics [5,6]. As a general and flexible data representation, graph structures effectively capture complex relational dependencies among entities and therefore serve as an important means for representing and analyzing structured data. Graph Neural Networks (GNNs) transmit information and aggregate features through adjacency relations among nodes, thereby learning high-level representations of nodes, edges, or entire graphs. They demonstrate excellent generalization performance on a variety of graph learning tasks, including node classification [7,8], link prediction [9,10], graph classification [11,12] and node clustering [13,14]. Typical GNN models adopt a message-passing mechanism, in which each layer updates node representations by aggregating features from neighboring nodes and integrating the node's own information.

Graph structures effectively characterize complex relationships among entities and are widely applied in fields such as social networks, transportation networks, and bioinformatics. GNNs transmit and aggregate features through adjacency relationships among nodes, demonstrating outstanding performance in tasks

such as node classification and link prediction. However, most GNN-based methods rely on a fundamental assumption that the original graph structure reliably reflects the true relationships among nodes. In real-world scenarios, graph data are often accompanied by noise, missing information, or structural bias [15], which significantly affects the learning performance of GNNs.

To address these issues, Graph Structure Learning (GSL) dynamically constructs or optimizes graph structures during training, enabling models to learn high-quality node representations from incomplete and noisy data. Existing GSL methods include strategies based on metric learning [16–18], probabilistic sampling [19–21], and end-to-end parameter optimization [22,23]. In recent years, several multi-view or self-supervised GSL methods, such as SUBLIME [24] and MVGRL [25], have also been proposed. Unlike existing multi-view graph structure learning methods that implicitly fuse heterogeneous views into a single adjacency representation via weighted aggregation or feature concatenation, such designs fail to explicitly preserve structural boundaries between different semantic spaces.

To address the aforementioned issues, this paper proposes a self-supervised dual-layer structure-driven graph structure learning method, termed D2GSL. D2GSL constructs two complementary channels, namely a semantic similarity channel and a spectral feature channel, to capture local semantic relationships and global structural characteristics, respectively. Instead of directly collapsing multiple views via simple fusion strategies, the two channels are modeled as a dual-layer network and interact through a hyperadjacency matrix that captures both intra-layer structures and cross-layer dependencies in a structured manner. A structure reconstruction loss is further introduced to enforce topological consistency across channels, while a contrastive learning objective enhances representation quality under a self-supervised paradigm. Overall, D2GSL learns structurally consistent representations without relying on label supervision. The main contributions of this paper are summarized as follows:

- Two structural views, namely the semantic similarity channel and the spectral feature channel, are constructed to explore multi-dimensional graph information from both local semantic and global spectral perspectives, thereby enhancing the expressive capability of node representations. . . .
- A dual-layer graph structure with a hyperadjacency matrix is proposed to model semantic and spectral channels in a structurally decoupled manner, enabling explicit inter-layer interaction beyond standard multi-view fusion strategies.
- Extensive experiments are conducted on multiple real-world datasets, and the results demonstrate that D2GSL outperforms existing baseline methods in node classification tasks.

2 Related Works

2.1 Graph Neural Network

Graph Neural Networks (GNNs) achieve outstanding performance in graph learning tasks by propagating and aggregating features through adjacency relationships among nodes. Early methods such as GCN [7] update node representations through local neighborhood feature aggregation, thereby laying the foundation for modern GNN architectures. Subsequent studies extend GNNs from multiple perspectives. For example, GAT [26] introduces an attention mechanism to assign learnable weights to different neighboring nodes, GraphSAGE [27] adopts a neighborhood sampling strategy to handle large-scale graph data, and ChebNet [28] utilizes Chebyshev polynomial approximation in the spectral domain to perform graph convolution. Detailed surveys on GNNs can be found in the literature [29–31].

2.2 Graph Structure Learning

Graph Structure Learning (GSL) aims to dynamically optimize the underlying graph structure to improve the performance of downstream tasks. Among existing methods, IDGL [16] generates graph structures through similarity measurement and jointly optimizes them with task-specific loss functions. Pro-GNN [23] exploits intrinsic graph properties such as low-rankness and sparsity to defend against adversarial attacks. SUBLIME [24] achieves unsupervised GSL through contrastive learning. PROSE [32] adopts a progressive strategy to iteratively expand graph structures. GEN [15] dynamically estimates latent graph structures to achieve collaborative parameter optimization. CoGSL [18] suppresses redundant noise by minimizing mutual information across different views. However, existing methods still exhibit several common limitations. Most of them rely heavily on supervised label information, which limits their applicability in label-scarce scenarios. Meanwhile, these methods are mainly based on single graph views or simple combinations of multiple views, making it difficult to fully characterize the complex relationships among nodes in different semantic spaces. To address these issues, this paper proposes D2GSL, which constructs two complementary channels based on semantic and spectral features and achieves cross-channel information fusion through a dual-layer structure and a hyperadjacency matrix.

2.3 Graph Contrastive Learning

Graph Contrastive Learning (GCL) has achieved significant progress in unsupervised graph representation learning in recent years. Its core idea is to guide models to learn discriminative node-level or graph-level representations by constructing positive and negative sample pairs. Among representative methods, DGI [33] enhances the capability of node embeddings to perceive global structural information by maximizing the mutual information between local node representations and global graph representations. MVGRL [25] introduces a multi-view learning strategy by contrasting the original graph with its Laplacian spectral view, thereby strengthening the learning of global structural characteristics. GRACE [34] adopts data augmentation strategies at the node level, such as feature masking and edge perturbation, and combines them with contrastive loss to optimize the similarity and discrepancy of node representations. In addition, some methods attempt to construct contrastive objectives from cross-graph or cross-channel perspectives to learn more expressive representations. Although GCL has made remarkable progress, most existing methods still rely heavily on specific data augmentation strategies and lack in-depth modeling of interactive semantics among multi-channel structures. In this paper, contrastive learning is incorporated within a self-supervised framework, and a structure reconstruction loss is further introduced to achieve cross-channel alignment.

3 Problem Definition

Given an undirected graph $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_N\}$ represents the set of nodes, $E \subseteq V \times V$ represents the set of edges, and $X \in \mathbb{R}^{N \times d}$ denotes the input features of the nodes. The graph's adjacency matrix is represented as $A \in \mathbb{R}^{N \times N}$, where $A_{ij} = 1$ if nodes v_i and v_j are connected, otherwise $A_{ij} = 0$. In GCN, node features and graph structure are jointly used for representation learning. Features are propagated and aggregated through the adjacency structure and the propagation rule can be expressed as:

$$H^{(l+1)} = \sigma\left(D^{-1/2}\tilde{A}D^{-1/2}H^{(l)} \cdot W^{(l)}\right), \tag{1}$$

where $H^{(l)}$ represents the node features at layer l, with the initial input $H^{(0)} = X$; $\tilde{A} = A + I$ represents the adjacency matrix with added self-loops; D is the degree matrix, $W^{(l)}$ is the learnable weight matrix for layer l and $\sigma(\cdot)$ is the nonlinear activation function. However, in real-world graphs, there are often issues such as noise or incomplete data, making it difficult to fully capture the true relationships between

nodes. Therefore, it is necessary to guide the construction of improved graph structures under supervision to enhance representation quality.

Definition 1 (Fiedler Value & Fiedler Vector). *In graph theory, given an undirected graph, its structure can be characterized by the Laplacian matrix. The undirected normalized Laplacian matrix of the graph is defined as:*

$$L = D - A, \tag{2}$$

where A is the adjacency matrix of the graph, D is the degree matrix. The graph Laplacian matrix L is positive semi-definite and its eigenvalues $\lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{N-1}$ are all real numbers, where $\lambda_0 = 0$. We refer to the second smallest eigenvalue λ_1 as the Fiedler value of the graph and its corresponding eigenvector is the Fiedler vector.

Definition 2 (Multilayer Network). *A multilayer network is represented as $M = (G, \varepsilon_c)$, where $G = \{G_\alpha; \alpha \in \{1, 2, \ldots, M\}\}$ is a class of directed or undirected graphs, with each $G_\alpha = (X_\alpha, E_\alpha)$. The calculation of ε_c is as follows:*

$$\varepsilon_c = \{E_{\alpha\beta} \subseteq X_\alpha \times X_\beta; \alpha, \beta \in \{1, \ldots, M\}, \alpha \neq \beta\}. \tag{3}$$

This represents the set of inter-layer connections between different layers G_α and G_β ($\alpha \neq \beta$). The elements of ε are called cross-layer connections. Each E_α is regarded as an intra-layer connection of M and each $E_{\alpha\beta}$ ($\alpha \neq \beta$) is regarded as an inter-layer connection of M.

Definition 3 (Graph Structure Learning). *Given an undirected graph $G = (V, E)$, GSL aims to optimize the objective function L to learn the adjacency matrix $A \in \{0,1\}^{N\times N}$ (or probability matrix $P(A)$) that enables effective data generation. The graph structure learning mechanism is expressed as:*

$$\hat{A} = \arg\max_A L(A; X, \Theta) + R(A), \tag{4}$$

where Θ represents the model parameters. $L(A; X, \Theta)$ is the structural loss function that quantifies the difference between the true graph structure and the predicted graph structure. $R(A)$ is the regularization term, which controls the sparsity or complexity of the adjacency matrix A.

Definition 4 (Graph Contrastive Learning). *The objective of GCL is to learn a projection function $f_\theta : \mathbb{R}^{N\times d} \to \mathbb{R}^{N\times k}$, which maps node representations from d-dimensional space to k-dimensional space. The objective function is optimized as follows:*

$$\mathcal{L}(f_\theta(G_1), f_\theta(G_2)) = -\sum_{i=1}^{N} \log \frac{\exp(\mathrm{sim}(f_\theta(v_i^{(G_1)}), f_\theta(v_i^{(G_2)})))}{\sum_{j=1}^{N} \exp(\mathrm{sim}(f_\theta(v_i^{(G_1)}), f_\theta(v_j^{(G_2)})))}. \tag{5}$$

The optimal projection function is obtained by:

$$\hat{\theta} = \arg\min_\theta \mathcal{L}(f_\theta(G_1), f_\theta(G_2)), \tag{6}$$

where $sim(z_i, z_j)$ represents the similarity between the representations of node i and node j. G_1 and G_2 represent different views of the graph.

4 Methodology

This section details our proposed self-supervised GSL framework, with the overall architecture of D2GSL shown in Fig. 1.

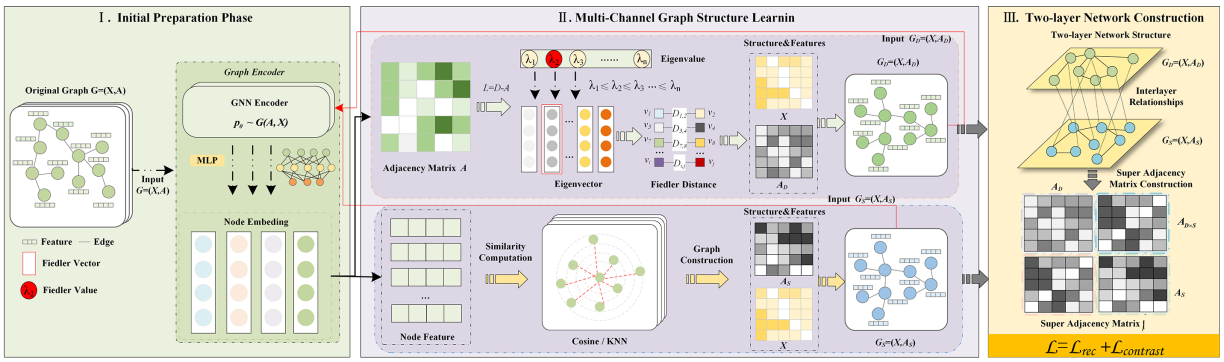

Figure 1: The overall framework of D2GSL. The model constructs a semantic similarity channel and a spectral feature channel derived from the Fiedler vector, and organizes them into a dual-layer graph structure. A hyperadjacency matrix is used to model intra- and inter-layer interactions. The framework is optimized with a structure reconstruction loss and a contrastive learning objective for self-supervised representation learning.

D2GSL explicitly models semantic and spectral information as two separated structural channels and organizes them into a dual-layer graph structure with explicit inter-layer interactions. Instead of directly collapsing multiple views into a single adjacency matrix, the model introduces a hyperadjacency matrix to represent intra-layer and inter-layer relationships in a structured manner. The framework is optimized using a structure reconstruction loss and a contrastive learning objective for self-supervised representation learning.

4.1 Multi-Channel Graph Structure Learning

D2GSL aims to achieve deep learning of graph data from both local semantic and global structural perspectives. It fully captures the multi-dimensional structural information embedded in the graph by constructing two complementary channel graphs: the Semantic Similarity Channel Graph and the Spectral Structure Channel Graph.

4.1.1 Construction of the Semantic Similarity Channel Graph

Considering that the semantic relationships among nodes can be characterized through the similarity of their embedding vectors, we first employ a graph encoder GCN to transform the original node features $X \in \mathbb{R}^{N\times d}$, thereby obtaining the node embedding matrix $H^{(D)} = f_\theta(X) \in \mathbb{R}^{N\times k}$, where $h_i^{(D)} \in \mathbb{R}^k$ denotes the embedding representation of node v_i. In this paper, two different graph construction mechanisms are designed to generate the semantic channel graph.

(1) K Nearest Neighbors

After obtaining the embedding of each node, let the number of nodes be N and the representation of each node be $h_i^{(D)} \in \mathbb{R}^k$. We use the K-nearest neighbor method to measure the Euclidean distance between the node embeddings, selecting the top k nearest neighbors for each node. The graph structure is constructed as $A_S = \{(v_i, v_j) \mid v_j \in KNN(v_i), i \in \{1, \cdots, N\}\}$, where the adjacency matrix entries are weighted by the L2 norm of the node feature differences, represented as $w_{ij} = \varnothing\left(\|h_i^{(D)} - h_j^{(D)}\|^2\right)$. Finally, DropEdge technique is applied to randomly remove edges, resulting in the redefined graph $G_S = (X, A_S)$.

(2) Cosine Similarity

To better capture the semantic relationships between nodes, another graph construction method is based on the cosine similarity of the node embeddings. The cosine similarity is computed between nodes i and j as:

$$\mathrm{sim}_{ij}^{(s)} = \frac{h_i^{(D)} h_j^{(D)}}{\|h_i^{(D)}\|_2 \cdot \|h_j^{(D)}\|_2}, \tag{7}$$

which captures the directionality of the relationship between nodes v_i and v_j. Based on this, we retain the top k nodes with the highest cosine similarity for every node, represented as $A_S = \{(v_i, v_j) \mid v_j \in \mathrm{Top}k(h_i^{(D)}), i \in \{1, \ldots, N\}\}$. We apply the same activation function $\varphi(\cdot)$ and use the DropEdge technique to obtain the final graph $G_S = (X, A_S)$.

4.1.2 Construction of the Spectral Feature Channel Graph

To further enhance the expressive capability of graph structures in the feature space, this paper introduces the Laplacian spectral decomposition method from spectral graph theory and constructs a Spectral Feature Channel Graph based on the Fiedler vector (the second eigenvector) of the graph. This method extracts low-frequency structural features from the original graph and effectively captures global connectivity information. The Fiedler vector is selected instead of higher-order spectral features because it not only characterizes global connectivity and forms a complementary relationship with the semantic channel, but also incurs relatively low computational cost, making it more suitable for self-supervised and label-scarce scenarios.

Given an original graph $G = (V, E)$, self-loops are first introduced into its adjacency matrix A to enhance the self-connection relationships of nodes, resulting in the updated adjacency matrix $\tilde{A} = A + I$. Based on this, the corresponding degree matrix $D \in \mathbb{R}^{N \times N}$ is computed, where D is a diagonal matrix whose i-th diagonal element represents the degree of node v_i, i.e., $D_{ii} = \sum_{j=1}^{N} A_{ij}$. Furthermore, the unnormalized graph Laplacian matrix is defined as $L = D - \tilde{A}$. The Laplacian matrix is symmetric positive semi-definite and possesses real non-negative eigenvalues $\lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{N-1}$, with the corresponding orthogonal eigenvectors $\{u_1, u_2, \ldots, u_N\}$. Subsequently, the eigenvectors corresponding to the first $k = 2$ smallest eigenvalues are selected. Among them, the eigenvector corresponding to the second smallest eigenvalue is referred to as the Fiedler vector, denoted as $u_2 \in \mathbb{R}^N$, and the spectral embedding representation of each node i is defined as $h_i^{(S)} = (u_2)_i$.

4.2 Dual-Layer Network Construction

This paper builds on multi-channel graph learning to explore the deep relationships and complementary features between different channel structures. It introduces a formal learning framework for dual-layer graph networks, constructing a dual-layer graph structure $M = ((G_D, G_S), \varepsilon_c)$. Here, $G_D = (X_D, E_D)$ and $G_S = (X_S, E_S)$ represent the graph structures corresponding to the two layers and ε_c represents the set of inter-layer connections between the two graphs. For each node $v_i^{(D)}$ in the G_D graph, we select the top K most similar nodes from the G_S graph and establish connections, defining the inter-layer edge set ε_c as:

$$\mathcal{E}_c = \bigcup_{i=1}^{N} \left\{ (v_i^{(D)}, v_j^{(S)}) \mid v_j^{(S)} \in \mathrm{TopK}(\mathrm{sim}(h_i^{(D)}, h_j^{(S)})) \right\}, \tag{8}$$

where $h_i^{(D)}$ and $h_j^{(S)}$ represent the node embeddings of the G_D and G_S graphs, respectively, and $\mathrm{Top}K(\cdot)$ represents the selection of the top K most similar nodes.

In the subsequent sections, we will use the notation $X_D = x_1^{(D)}, x_2^{(D)}, \ldots, x_{N_D}^{(D)}$ to represent the node set of G_D. For each layer G_D, its adjacency matrix is defined as $A^{(D)} = (a_{ij}^{(D)}) \in \mathbb{R}^{N_D \times N_D}$, as shown in Eq. (9).

$$a_{ij}^{(D)} = \begin{cases} 1, & \text{if } (x_i^{(D)}, x_j^{(D)}) \in E_D, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

For $1 \le i, j \le N_D$, the inter-layer relationships between G_D and G_S reflect the important cross-graph relationships, which are defined in the next equation. We then define the cross-layer adjacency matrix $A^{(DS)} = (a_{ij}^{(DS)}) \in \mathbb{R}^{N_D \times N_S}$ as follows:

$$a_{ij}^{(DS)} = \begin{cases} 1, & \text{if } j \in \mathrm{Top}K(\mathrm{dist}(h_i^{(D)}, h_j^{(S)})), \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Here, N_D and N_S represent the number of nodes in G_D and G_S, respectively, and $h_i^{(D)}$ and $h_j^{(S)}$ represent the node embeddings from different layers. Finally, the cross-layer hyperadjacency matrix $\bar{A}$ for the dual-layer network is defined as shown in Eq. (11):

$$\bar{A} = \begin{pmatrix} A^{(D)} & A^{(DS)} \\ (A^{(DS)})^T & A^{(S)} \end{pmatrix} \in \mathbb{R}^{(M+W)\times(M+W)}, \tag{11}$$

where $A^{(DS)}$ represents the inter-layer connections between G_D and G_S. According to the multilayer network framework defined in Definition 2, the hyperadjacency matrix can be regarded as a unified mathematical representation of the intra-layer connections $A^{(D)}$, $A^{(S)}$ and the inter-layer connections $A^{(DS)}$. This provides graph-theoretic support for modeling the dual-layer structure. The topological consistency of this matrix is further constrained through the structure reconstruction loss in Section 4.3.

4.3 Joint Optimization

To maintain consistency in the representation space and the structural space, this study introduces a joint optimization objective. The objective combines the contrastive learning loss with the dual-layer network reconstruction loss. First, to enhance the representation consistency of the model across different views, D2GSL performs encoding on the Semantic Similarity Channel Graph and the Spectral Feature Channel Graph, obtaining two node embeddings, $h_i^{(D)}$ and $h_i^{(S)}$ for node i. By maximizing the similarity between positive sample pairs and minimizing the similarity between negative sample pairs, D2GSL achieves cross-channel alignment at the semantic level. This is specifically defined by Eq. (12),

$$\mathrm{L}_{con} = -\sum_{i=1}^{N} \log \frac{\exp(\mathrm{sim}(h_i^{(D)}, h_i^{(S)})/\tau)}{\sum_{j=1}^{N} \left(\exp(\mathrm{sim}(h_i^{(D)}, h_j^{(S)})/\tau) + \exp(\mathrm{sim}(h_i^{(S)}, h_j^{(D)})/\tau) \right)}, \tag{12}$$

where $\mathrm{sim}(\cdot)$ represents the cosine similarity function and τ is the temperature coefficient. Next, to maintain topological consistency at the structural level, this paper introduces the dual-layer network reconstruction loss. Specifically, the Semantic Similarity Channel Graph and the Spectral Feature Channel Graph are first merged into a higher-order structural representation, as defined by Eq. (11). D2GSL predicts the adjacency matrices $A^{(D)}$ and $A^{(S)}$ from the embeddings of the two channels. It then calculates the structural reconstruction loss with the hyperadjacency matrix $\bar{A}$ as defined in Eq. (13),

$$\mathrm{L}_{rec}(\bar{A}, A^{(D)}) = -\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left[w_p \cdot A_{ij}^{(D)} \log(\sigma(\bar{A}_{ij})) + w_n \cdot (1 - A_{ij}^{(D)}) \log(1 - \sigma(\bar{A}_{ij}))\right], \tag{13}$$

where $\sigma(\cdot)$ represents the Sigmoid activation function, and w_p and w_n represent the weights for the positive and negative samples, respectively, which are used to balance the imbalance between edges and non-edges in sparse graphs. The structure reconstruction loss L_{rec} is used as a structural regularization term to enforce consistency between the hyperadjacency matrix and the original channel graphs. The final joint optimization objective function is defined as shown in Eq. (14).

$$\mathrm{L} = \beta \cdot \mathrm{L}_{con} + (1 - \beta) \cdot \mathrm{L}_{rec}. \tag{14}$$

Through this dual-channel joint optimization mechanism, D2GSL achieves consistency between the representation space and the structural space from both the semantic perspective and the spectral feature perspective.

5 Experiments

In this section, we evaluate the performance of the proposed D2GSL method through comparative experiments with several representative baseline methods on different tasks and datasets. These experiments validate the overall effectiveness and superiority of the method.

5.1 Experimental Settings

Datasets: To evaluate the effectiveness of D2GSL, extensive self-supervised node classification experiments are conducted on three widely used benchmark datasets. Detailed descriptions of the datasets are presented in Table 1. We evaluate our method on three benchmark datasets: Cora and CiteSeer with the standard Planetoid splits [35], and the DBLP subset preprocessed by Fu et al. [36], which are widely adopted as baselines within the GSL paradigm. Utilizing these benchmarks provides an empirical anchor, establishing a standardized baseline for direct performance cross-comparisons against foundational self-supervised models.

Table 1: Statistics of dataset properties.

Datasets	Nodes	Edges	Classes	Features
Cora	2708	5429	7	1433
Citeseer	3327	4732	6	3703
DBLP	2975	2398	4	334

Baselines: This study selects twelve representative baseline models for a rigorous comparative evaluation. Specifically, the semi-supervised learning methods include the classical GCN [7], GAT [26], and GraphSAGE [27]. Within the realm of self-supervised learning, DGI [33], IDGL [16], and GEN [15] serve as non-contrastive baselines, while GRACE [34], GraphCL [37], MVGRL [25], and ReGCL [38] are incorporated as self-supervised graph contrastive learning approaches. Finally, Pro-GNN [23] and SUBLIME [24] are employed as graph structure optimization methods.

Experimental details: The experiments for the proposed D2GSL method are conducted on a computer equipped with an NVIDIA GeForce RTX 4070 Ti GPU, using PyTorch 1.10.1 and CUDA 11.3.

The model employs a two-layer GCN as the encoder with shared parameters. The optimal experimental hyperparameters for the three datasets are shown in Table 2.

Table 2: Detailed parameter setting.

Datasets	EP	LR	β	τ	HD
Cora	300	$1e^{-2}$	0.4	0.5	512
Citeseer	400	$1e^{-3}$	0.4	0.6	512
DBLP	400	$1e^{-3}$	0.4	0.5	512

In Table 2, EP represents the number of training epochs, LR is used to control the step size for updating the model parameters during each iteration, β is the hyperparameter for the tasks, τ represents the temperature parameter and HD is the hidden layer dimension.

Time Complexity Analysis: Let N denote the number of nodes, d_f the feature dimension, d_h the hidden dimension, and E the number of edges. The main computational costs of D2GSL come from: (1) semantic graph construction with complexity $O(N^2 d_f)$; (2) spectral Fiedler vector computation with complexity $O(E)$; (3) GCN encoder with complexity $O(Ed_h)$; (4) contrastive loss with complexity $O(N^2 d_h)$; and (5) structure reconstruction loss with complexity $O(N^2)$. The overall complexity is therefore $O(N^2 d_f + N^2 d_h + Ed_h)$. In our experiments, $N \leq 3327$ and E is the number of edges in a sparse graph, so the complexity is dominated by the $O(N^2)$ terms.

5.2 Effectiveness Experiments and Analysis

In this study, we treat node classification as the downstream task. The experimental results are shown in Table 3 with the best results marked in bold.

Table 3: Node classification results on different datasets.

Method	Cora		Citeseer		DBLP	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
GCN	81.84 ± 0.57	80.60 ± 1.10	71.81 ± 0.31	68.12 ± 0.44	90.99 ± 0.28	90.01 ± 0.32
GAT	82.80 ± 0.50	81.80 ± 0.30	72.10 ± 1.00	67.10 ± 1.30	91.13 ± 0.40	90.22 ± 0.37
GraphSAGE	78.85 ± 0.14	77.83 ± 0.11	71.68 ± 0.06	**70.73 ± 0.32**	71.63 ± 0.18	70.71 ± 0.27
DGI	80.10 ± 0.02	75.00 ± 0.07	71.89 ± 0.54	67.74 ± 0.49	83.00 ± 0.03	78.21 ± 0.01
IDGL	84.00 ± 0.44	82.65 ± 0.45	72.10 ± 0.46	69.05 ± 0.26	90.61 ± 0.56	89.65 ± 0.60
GEN	81.36 ± 0.49	80.10 ± 0.46	72.00 ± 0.87	68.63 ± 0.72	89.77 ± 0.67	87.54 ± 0.32
Pro-GNN	81.26 ± 0.53	80.49 ± 0.47	70.98 ± 0.94	67.43 ± 0.10	84.83 ± 1.36	83.13 ± 1.56
SUBLIME	83.82 ± 0.57	82.82 ± 0.66	72.72 ± 0.49	68.19 ± 0.61	91.82 ± 0.27	90.89 ± 0.37
GRACE	83.32 ± 0.13	76.78 ± 0.24	64.16 ± 0.52	60.13 ± 0.39	84.22 ± 0.16	83.35 ± 0.39
GraphCL	81.91 ± 0.69	76.25 ± 1.07	65.16 ± 0.81	59.58 ± 0.65	80.21 ± 0.32	79.12 ± 0.21
ReGCL	82.94 ± 0.18	79.28 ± 0.28	72.53 ± 0.47	69.57 ± 0.12	**92.72 ± 0.39**	**91.54 ± 0.47**
MVGRL	82.72 ± 0.56	78.08 ± 0.32	67.41 ± 0.04	62.10 ± 0.40	90.12 ± 0.53	89.13 ± 0.62
D2GSL	**85.12 ± 0.83**	**83.54 ± 0.71**	**73.45 ± 0.42**	68.74 ± 0.21	91.76 ± 0.30	89.32 ± 0.45

Note: Bold indicates optimal.

It can be observed from Table 3 that D2GSL achieves superior performance over all baseline methods on both metrics of the Cora dataset and obtains the best performance on the Micro-F1 metric of the Citeseer dataset. On the Macro-F1 metric of Citeseer and the DBLP dataset, D2GSL also demonstrates strong competitiveness, with relatively small gaps compared to strong baseline methods such as SUBLIME and ReGCL. Specifically, on the Cora dataset, D2GSL improves performance by approximately 1.12% over the second-best method IDGL. On the Citeseer dataset, D2GSL outperforms the second-best method SUBLIME by approximately 0.73% in terms of Micro-F1. It should be noted that this paper adopts a fully self-supervised setting, whereas some baseline methods are originally evaluated under semi-supervised settings in their respective works. Under this setting, the above results further demonstrate the effectiveness of D2GSL in self-supervised graph structure learning tasks. Therefore, these performance gains can be attributed to the following key factors:

(1) Multichannel structure modeling enhances feature representation: D2GSL constructs semantic similarity channels and spectral feature channels, which capture local semantic and global structural information, respectively. The two channels are complementary, enabling the model to strike a balance between local discriminability and global perception.

(2) Dual-layer structure and hyperadjacency matrix enhance information fusion: D2GSL constructs a hyperadjacency matrix through inter-layer interactions. This design enables deep integration of local and global structures and improves the diversity and stability of node representations.

(3) Structure reconstruction and contrastive learning improve representation consistency: Structure reconstruction maintains consistency across multichannel structures, while contrastive learning strengthens the aggregation of semantically similar nodes, thereby enhancing the discriminability of self-supervised representations.

5.3 Ablation Study

In this section, we conduct ablation experiments on the D2GSL model to examine the contribution of each core component. We construct two model variants on the Cora and Citeseer datasets and compare their performance by progressively removing key modules. The experimental results are summarized in Table 4.

Table 4: D2GSL and its variants' performance.

Method	Cora		Citeseer	
	Micro-F1	**Macro-F1**	**Micro-F1**	**Macro-F1**
Only Semantic	83.47 ± 0.43	82.08 ± 0.31	71.96 ± 0.37	67.73 ± 0.28
Only Spectral	80.18 ± 0.61	78.53 ± 0.48	69.42 ± 0.35	65.31 ± 0.39
D2GSL w/o L_{rec}	84.12 ± 0.11	82.11 ± 0.23	72.91 ± 0.01	67.22 ± 0.10
D2GSL w/o L_{con}	83.80 ± 0.23	82.27 ± 0.10	72.10 ± 0.22	66.87 ± 0.56
D2GSL	**85.12 ± 0.83**	**83.54 ± 0.71**	**73.45 ± 0.42**	**68.74 ± 0.21**

Note: Bold indicates optimal.

As shown in Table 4, the model performance degrades on both Cora and Citeseer when key components are removed. Specifically, using only the semantic similarity channel (Only Semantic) substantially outperforms using only the spectral feature channel (Only Spectral), which is expected since semantic information typically provides stronger discriminative power than pure spectral structure in citation networks. However, both single-channel variants perform significantly worse than any dual-channel variant, indicating that a

single view cannot simultaneously capture local semantics and global structure, thus validating the necessity of the dual-channel design. Furthermore, removing either L_{rec} or L_{con} leads to a performance drop compared to the full model, with the removal of L_{con} having a more pronounced negative impact, suggesting that the contrastive loss is crucial for learning discriminative features. The full model achieves the best results across all metrics, fully demonstrating the effectiveness of the dual-channel design, dual-layer structure, and joint optimization mechanism.

5.4 Hyperparameter Sensitivity

To comprehensively analyze the impact of hyperparameters on D2GSL, we conduct experiments on the Citeseer dataset.

Fig. 2 shows the impact of different hyperparameters, such as Beta and learning rate, on the Micro F1 score for the CiteSeer dataset. When the learning rate is in the lower range of [0.005, 0.01], the model achieves a higher Micro F1 score. However, as the learning rate increases, it negatively affects the optimization process, leading to a significant drop in performance. Regarding the hyperparameter Beta, higher values cause the D2GSL model to gradually degrade to the performance level of the baseline methods. Keeping Beta within the range of [0.4, 0.5] yields better performance. In summary, Beta and the learning rate need to be coordinated to achieve better Micro F1 performance on the CiteSeer dataset. Both parameters need relatively low values to obtain stable results.

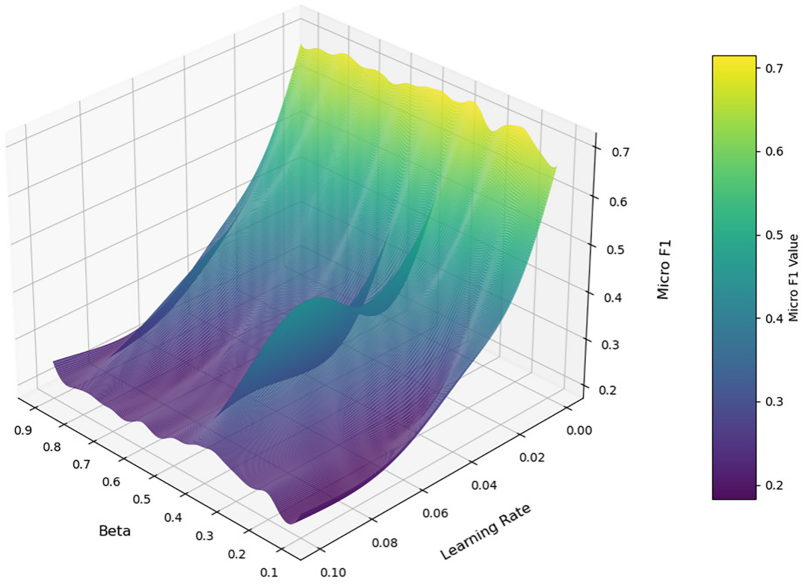

Figure 2: Hyperparameter sensitivity analysis on Citeseer.

6 Conclusion

We propose D2GSL, a self-supervised dual-layer structure-driven graph structure learning method, to mitigate structural noise, missing information, and inherent bias in graph data. In a self-supervised manner, D2GSL constructs dual channels based on semantic similarity and spectral features, seamlessly fuses cross-channel information via a hyperadjacency matrix over a dual-layer network, and introduces joint structural reconstruction along with contrastive learning to enhance consistency and complementarity. The key innovation lies in explicitly modeling structural interactions through the hyperadjacency formulation and leveraging a dedicated reconstruction loss to overcome the limitations of heuristic view fusion. Experimental

results demonstrate that D2GSL achieves superior performance on tasks such as node classification. Future work will focus on: (1) improving scalability on large-scale and dynamic graphs through graph sampling or sparsification techniques; (2) systematically evaluating model robustness against missing edges, structural noise, and adversarial attacks; and (3) incorporating higher-order spectral information to enhance global structure capture.

Acknowledgement: None

Funding Statement: This work is supported by the Scientific Research Innovation Capability Support Project for Young Faculty (SRICSPYF-BS2025007), Construction Project of State Key Laboratory of Tibetan Intelligent (2025-ZJ-J08), National Natural Science Foundation of China (62566050).

Author Contributions: In this study, Juncheng Zhang primarily contributed to conceptualization, methodology development and manuscript writing. Xuhao Wei and Xiaolei Gu were involved in data curation and analysis. Zhonglin Ye and Haixing Zhao provided overall research guidance, supervision and manuscript revision. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The authors confirm that the data supporting the findings of this study are available within the article.

Ethics Approval: Not applicable.

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

References

1. Kumar S, Mallik A, Panda BS. Influence maximization in social networks using transfer learning via graph-based LSTM. Expert Syst Appl. 2023;212(9):118770. doi:10.1016/j.eswa.2022.118770.
2. Wei X, Liu Y, Sun J, Jiang Y, Tang Q, Yuan K. Dual subgraph-based graph neural network for friendship prediction in location-based social networks. ACM Trans Knowl Discov Data. 2023;17(3):42. doi:10.1145/3554981.
3. Chen C, Liu Y, Chen L, Zhang C. Bidirectional spatial-temporal adaptive transformer for urban traffic flow forecasting. IEEE Trans Neural Netw Learn Syst. 2022;34(10):6913–25. doi:10.1109/TNNLS.2022.3183903.
4. Jiang W, Luo J. Graph neural network for traffic forecasting: a survey. Expert Syst Appl. 2022;207(7):117921. doi:10.1016/j.eswa.2022.117921.
5. Li R, Yuan X, Radfar M, Marendy P, Ni W, O'Brien TJ, et al. Graph signal processing, graph neural network and graph learning on biological data: a systematic review. IEEE Rev Biomed Eng. 2021;16(5980):109–35. doi:10.1109/RBME.2021.3122522.
6. Gao Z, Jiang C, Zhang J, Jiang X, Li L, Zhao P, et al. Hierarchical graph learning for protein–protein interaction. Nat Commun. 2023;14(1):1093. doi:10.1038/s41467-023-36736-1.
7. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv:1609.02907. 2016. doi:10.48550/arxiv.1609.02907.
8. Zou M, Gan Z, Cao R, Guan C, Leng S. Similarity-navigated graph neural networks for node classification. Inf Sci. 2023;633(9):41–69. doi:10.1016/j.ins.2023.03.057.
9. Kipf TN, Welling M. Variational graph auto-encoders. arXiv:1611.07308. 2016. doi:10.48550/arXiv.1611.07308.
10. Peng Z, Huang W, Luo M, Zheng Q, Rong Y, Xu T, et al. Graph representation learning via graphical mutual information maximization. In: Proceedings of the Web Conference 2020; 2020 Apr 20–24; Taipei, Taiwan. p. 259–70. doi:10.1145/3366423.3380112.
11. Zeng J, Xie P. Contrastive self-supervised learning for graph classification. Proc AAAI Conf Artif Intell. 2021;35(12):10824–32. doi:10.1609/aaai.v35i12.17293.

12. Du J, Wang S, Miao H, Zhang J. Multi-channel pooling graph neural networks. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence; 2021 Aug 19–27; Montreal, QC, Canada. p. 1442–8. doi:10.24963/ijcai.2021/199.
13. Tsitsulin A, Palowitch J, Perozzi B, Müller E. Graph clustering with graph neural networks. J Mach Learn Res. 2023;24(127):1–21. doi:10.1145/3580305.3599177.
14. Bhowmick A, Kosan M, Huang Z, Singh A, Medya S. DGCLUSTER: a neural framework for attributed graph clustering via modularity maximization. Proc AAAI Conf Artif Intell. 2024;38(10):11069–77. doi:10.1609/aaai.v38i10.28983.
15. Wang R, Mou S, Wang X, Xiao W, Ju Q, Shi C, et al. Graph structure estimation neural networks. In: Proceedings of the Web Conference 2021; 2021 Apr 19–23; Ljubljana Slovenia. p. 342–53. doi:10.1145/3442381.3449952.
16. Chen Y, Wu L, Zaki M. Iterative deep graph learning for graph neural networks: better and robust node embeddings. In: Proceedings of the 34th International Conference on Neural Information Processing Systems; 2020 Dec 6–12; Vancouver, BC, Canada. p. 19314–26.
17. Fatemi B, El Asri L, Kazemi SM. SLAPS: self-supervision improves structure learning for graph neural networks. In: Proceedings of the 35th International Conference on Neural Information Processing Systems; 2021 Dec 6–14; Virtual. p. 22667–81. doi:10.48550/arxiv.2102.05034.
18. Liu N, Wang X, Wu L, Chen Y, Guo X, Shi C. Compact graph structure learning via mutual information compression. In: Proceedings of the ACM Web Conference 2022; 2022 Apr 25–29; Lyon, France. p. 1601–10. doi:10.1145/3485447.3512206.
19. Franceschi L, Niepert M, Pontil M, He X. Learning discrete structures for graph neural networks. In: Proceedings of the Proceedings of the 36th International Conference on Machine Learning; 2019 Jun 9–15; Long Beach, CA, USA. p. 1972–82.
20. Sun Q, Li J, Peng H, Wu J, Fu X, Ji C, et al. Graph structure learning with variational information bottleneck. PProc AAAI Conf Artif Intell. 2022;36(4):4165–74. doi:10.1609/aaai.v36i4.20335.
21. Zhao J, Wen Q, Ju M, Zhang C, Ye Y. Self-supervised graph structure refinement for graph neural networks. In: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining; 2023 Feb 27–Mar 3; Singapore. p. 159–67. doi:10.1145/3539597.3570455.
22. Wu Q, Zhao W, Li Z, Wipf DP, Yan J. NodeFormer: a scalable graph structure learning transformer for node classification. In: Proceedings of the 36th International Conference on Neural Information Processing Systems; 2022 Nov 28–Dec 9; New Orleans, LA, USA. p. 27387–401.
23. Jin W, Ma Y, Liu X, Tang X, Wang S, Tang J. Graph structure learning for robust graph neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2020 Jul 6–10; Virtual. p. 66–74. doi:10.1145/3394486.3403049.
24. Liu Y, Zheng Y, Zhang D, Chen H, Peng H, Pan S. Towards unsupervised deep graph structure learning. In: Proceedings of the ACM Web Conference 2022; 2022 Apr 25–29; Lyon, France. p. 1392–403. doi:10.1145/3485447.3512186.
25. Hassani K, Khasahmadi AH. Contrastive multi-view representation learning on graphs. In: Proceedings of the 37th International Conference on Machine Learning; 2020 Jul 13–18; Virtual. p. 4116–26.
26. Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. In: Proceedings of the 6th International Conference on Learning Representations; 2018 Apr 30–May 3; Vancouver, BC, Canada.
27. Hamilton WL, Ying Z, Leskovec J. Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems; 2017 Dec 4–9; Long Beach, CA, USA. p. 1025–35.
28. Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. In: Proceedings of the 30th International Conference on Neural Information Processing System; 2016 Dec 5–10; Barcelona, Spain. p. 3844–52.
29. Jin M, Koh HY, Wen Q, Zambon D, Alippi C, Webb GI, et al. A survey on graph neural networks for time series: forecasting, classification, imputation, and anomaly detection. IEEE Trans Pattern Anal Mach Intell. 2023;46(12):10466–85. doi:10.1109/TPAMI.2024.3443141.

30. Dai E, Zhao T, Zhu H, Xu J, Guo Z, Liu H, et al. A comprehensive survey on trustworthy graph neural networks: privacy, robustness, fairness, and explainability. Mach Intell Res. 2024;21(6):1011–61. doi:10.1007/s11633-024-1510-8.
31. Lu H, Wang L, Ma X, Cheng J, Zhou M. A survey of graph neural networks and their industrial applications. Neurocomputing. 2025;614(1):128761. doi:10.1016/j.neucom.2024.128761.
32. Wang H, Fu Y, Yu T, Hu L, Jiang W, Pu S. PROSE: graph structure learning via progressive strategy. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; 2023 Aug 6–10; Long Beach, CA, USA. p. 2337–48. doi:10.1145/3580305.3599476.
33. Veličković P, Fedus W, Hamilton WL, Liò P, Bengio Y, Hjelm RD. Deep graph infomax. arXiv:1809.10341. 2018. doi:10.48550/arXiv.1809.10341.
34. Zhu Y, Xu Y, Yu F, Liu Q, Wu S, Wang L. Deep graph contrastive representation learning. arXiv:2006.04131. 2021. doi:10.48550/arXiv.2006.04131.
35. Sen P, Namata G, Bilgic M, Getoor L, Gallagher B, Eliassi-Rad T. Collective classification in network data. AI Mag. 2008;29(3):93–106. doi:10.1609/aimag.v29i3.2157.
36. Fu X, Zhang J, Meng Z, King I. MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding. In: Proceedings of the Web Conference 2020; 2020 Apr 20–24; Taipei, Taiwan. p. 2331–41. doi:10.1145/3366423.3380297.
37. You Y, Chen T, Sui Y, Chen T, Wang Z, Shen Y. Graph contrastive learning with augmentations. In: Proceedings of the 34th International Conference on Neural Information Processing Systems; 2020 Dec 6–12; Vancouver, BC, Canada. p. 5812–23.
38. Ji C, Huang Z, Sun Q, Peng H, Fu X, Li Q, et al. ReGCL: rethinking message passing in graph contrastive learning. Proc AAAI Conf Artif Intell. 2024;38(8):8544–52. doi:10.1609/aaai.v38i8.28698.