Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 34(3): 1626–1652.
DOI: 10.3934/era.2026074
Received: 15 December 2025
Revised: 17 January 2026
Accepted: 03 February 2026
Published: 27 February 2026

Research article

Semi-supervised teaching graph contrastive network for node classification

Chenbin Shen[1], **Jingjing Song**[1,*], **Qiguo Sun**[1], **Qihang Guo**[1] **and Eric C.C. Tsang**[2]

[1] School of Computer, Jiangsu University of Science and Technology, Zhenjiang 212100, China
[2] School of Computer Science and Engineering, Macau University of Science and Technology, Macau 999078, China

* **Correspondence:** Email: songjingjing108@163.com.

Abstract: Graph contrastive learning methods have recently emerged as a promising solution to tackle the problem of label scarcity in real-world scenarios. However, most of the existing methods are still flawed due to the lack of guided objectives. Moreover, they fail to effectively utilize complementary structural information from different graphs. To address these limitations, we propose a novel semi-supervised teaching graph contrastive network (STGCN) for node classification. Based on the teaching network architecture, STGCN establishes multi-level contrastive objectives, ensuring rich and detailed supervision for graph encoders. Specifically, after carefully analyzing the intrinsic correlation between different augmented views, we send a diffusion graph and two augmented views together into the novel teaching network, which owns one teacher encoder to guide two shared student encoders. Furthermore, we introduce a random sampling mixing module that extracts complementary information from multiple graphs, along with a label propagation technique to fully exploit limited labeled data. Finally, our method incorporates supervised contrastive loss and node similarity regularization to ensure coherent alignment between labeled and unlabeled nodes. Extensive experiments on five real-world node classification datasets demonstrate a maximum of 2.60% higher improvement than other models.

Keywords: graph neural networks; supervised graph contrastive learning; node classification; few label

1. Introduction

Graph neural networks (GNNs) have gained significant popularity due to their remarkable capability in processing graph-structured data. They are widely used in tasks such as community detection [1–3], node classification [4–6], and recommendation systems [7–9]. GNNs incorporate both node attributes and graph structural information through recursive neighborhood aggregation. They are usually based on supervised learning mechanisms and require labeled data for training. However, in the real world,

the label information of data is generally limited and difficult to obtain.

To address the problem of label scarcity, graph contrastive learning (GCL) has become a powerful paradigm. GCL encourages models to maximize the similarity between positive pairs while minimizing the similarity between negative pairs, enabling the extraction of discriminative and robust features without supervision. However, most existing methods rely solely on unsupervised contrastive loss, which only distinguishes between different views and does not utilize any explicit labels or semantic information to guide the learning process. This design often leads to suboptimal representations: The reliance on numerous negative samples introduces sampling bias and false negatives, while the absence of supervised or semantic constraints prevents the model from capturing class-level relationships. Although methods like bootstrapped graph latents (BGRL) [10] and bootstrap your own latent (BYOL) [11] mitigate the dependence on negative samples through prediction-based consistency learning, they still lack guidance, making representation learning unstable or semantically inconsistent. Therefore, designing guided and semantically consistent objectives remains an open problem in GCL.

Another challenge of GCL lies in the lack of mechanisms to effectively capture and integrate complementary structural information across multiple graph views. BGRL [10] does not require the design of complex positive and negative case pairs, but rather predicts the same data from different perspectives, enabling the model to generate consistent representations for different versions of the same data. BYOL [11] involves a teaching architecture where one network generates stable targets while another one learns meaningful representations by matching its outputs to those targets. Graph with Few Labels (GraFN) [12] incorporates a teaching network paradigm and minimizes the discrepancy between two class distributions, which are non-parametrically derived based on anchor and support similarity from two differently augmented views. Supervised GCL (SGCL) [13] introduces augmentation techniques driven by a novel centrality function, aimed at enhancing the topological structures of the graph. However, they tend to overlook the complementary structural information from different graphs and the hierarchical relationships shared across these graphs. This misalignment can hinder the ability of graph encoders to capture structured and informative representations.

In order to address the above constraints, we propose a semi-supervised teaching graph contrastive network (STGCN) for node classification. STGCN introduces a teacher-guided network and label propagation mechanism to generate a contrastive objective through semantic supervision. Therefore, it enables more consistent and meaningful cross-view representation alignment. Specifically, STGCN constructs two augmented views, incorporating a novel centrality-based edge dropout and feature masking approach to optimize augmentation. In addition, a diffusion graph is used to smooth the structural information and builds an intermediate bridge between the two distributions. Together, they form a specialized teaching network, consisting of three GNN encoders, that utilizes complementary structural information from different graphs. A module called random mixing is designed to further explore the underutilized information in labeled nodes by sampling labeled nodes from two augmented views. Then, it generates new samples for contrast with the original graph. It also facilitates label propagation, maximizing the utility of labeled information. STGCN designs a contrastive loss based on normalized temperature-scaled cross-entropy, comparing augmented views with the original graph. Three GNN encoders compute mutual information between augmented representations to enhance learning ability. The designed contrastive target is used to ensure

consistency in each node embedding from different perspectives, and maintain sufficient discrimination from other node embeddings. This alleviates the limitations of contrastive targets in the model.

This paper is structured as follows: Section 2 discusses related work. Section 3 presents the proposed STGCN framework. Section 4 provides experimental results and evaluations. And finally, Section 5 concludes the whole paper.

2. Related work

2.1. Semi-supervised learning

Semi-supervised learning methods leverage both labeled and unlabeled data for model training. In the graph domain, the graph contrastive network (GCN) [14] is considered as a classical semi-supervised model. It not only relies on labeled node data for supervised training but also can learn through unlabeled node data. The graph attention network (GAT) [15] aggregates the features of neighboring nodes through adaptive attention weights, freeing itself from the limitations of equal weight aggregation of neighbor information in GCN. The graph sample and aggregate (GraphSAGE) method [16] selects only a subset of neighboring nodes for information aggregation during each training session by sampling them, effectively improving computational efficiency. Contrastive and generative GCN (CG^3) [17] introduces a graph sampling mechanism to capture different local and global features by generating multiple sampled images of the same graph. The main idea of CG^3 is to enhance the representation ability of GNNs through graph sampling and contrastive learning. Self-training [18] utilizes Parwalks [19] to provide reliable pseudo-labels to train GNNs. The label set is augmented through the utilization of pseudo-labels derived from a pre-trained GNN. Multi-stage self-supervised learning (M3S) [20] employs clustering techniques to eliminate those that are incompatible with the clustering objective to enhance the precision of pseudo-labels. Confidence and label propagation GCN (CLP) [21] proposes a framework that uses label confidence scores and simultaneously propagates pseudo-labels to improve model accuracy. Hierarchal GCN (H-GCN) [22] repeatedly aggregates nodes with similar structures into hyper nodes, and then refines the coarsened graph back to the original graph to restore the representation of each node. Its coarsening can capture more global information.

These semi-supervised learning methods can effectively learn low-dimensional nodes or graph embeddings from high-dimensional complex graph-structured data. They can accurately classify the learned effective node representations, which is the node classification task and the goal of this paper. But when they encounter limited label scenarios, their performance will significantly decrease.

2.2. Graph contrastive learning

GCL has become a powerful self-supervised paradigm for cross-domain representation learning, capable of effective learning without the need for labeled data. By constructing positive and negative sample pairs, contrastive learning trains models to produce discriminative embeddings. Adaptive GCL (AdaGCL) [23] introduces additional high-quality training signals for collaborative filtering through data augmentation using two adaptive contrastive view generators, which help alleviate data sparsity and noise issues. Adversarial GCL (AD-GCL) [24] optimizes the adversarial graph

augmentation strategy used in GCL to allow GNNs to mitigate the learning of redundant representations throughout the training process. Deep graph infomax (DGI) [25] compares local representations with global representations, using graph perturbations or feature masking to ensure consistency. Graph contrastive learning with augmentations (GraphCL) [26] employs diverse graph augmentation techniques to generate multi-view representations, aligning representations across augmented views for more robust embeddings. Contrastive multi-view representation learning (MVGRL) [27] leverages a local-global contrastive learning paradigm. It maximizes mutual information between local node features and global graph representations, and incorporates subgraph embeddings to capture multi-scale graph structures. Graph contrastive coding (GCC) [28] focuses on multi-level contrasts, aligning nodes with their corresponding subgraphs and global graphs to capture structural relationships at various scales. Graph representation embedding enhanced (GRE^2-MDCL) [29] enhances the discriminability of node representations by constructing contrastive objectives across multiple representation dimensions. Deep GCL (GRACE) [30] learns node representations by leveraging two augmented graph views, ensuring that representations of the same nodes are pulled closer while those of different nodes are pushed apart. Self-supervised contrastive learning (SCL) [31] proposes a contrastive learning mechanism that focuses on the representation of points of interest to capture visiting uncertainties in itinerary generation. Keywords-enhanced contrastive learning model (KCLM) [32] employs keywords as semantic information to introduce a contrastive model for learning the representations of users and travel items.

Despite the progress achieved by existing GCL methods, two challenges remain. First, most methods, due to the lack of guided objectives, still have shortcomings, which hinder their effective integration of supervision in the representation learning process. Second, existing methods often neglect the complementary structural information and hierarchical relationships shared across augmented views. These challenges motivate the need for more guided, effectively utilized complementary structural information, and structure-aware GCL paradigms.

3. Method

3.1. Semi-supervised teaching graph contrastive network for node classification

3.1.1. Augmentation graph view

Compared to common data augmentation techniques used in text and image domains [26], graph structure augmentation is significantly more complex. This complexity poses challenges in defining effective augmentation strategies for contrastive learning. For instance, simple augmentation techniques struggle to create diverse node neighborhoods, making it difficult to optimize contrastive objectives effectively. Furthermore, the functions of nodes and edges in graph augmentation are fundamentally distinct, which adds additional complexity. Currently, there are two methods for generating graph augmentation. The first augmentation method [26] involves random graph transformation, where we generate one augmented graph view by randomly masking node features and removing a subset of edges. The second augmentation method [33] adopts a probability-based edge perturbation strategy, where we construct the augmented graph view by assigning higher removal probabilities to unimportant edges. Both of these augmentation methods involve the strategy of discarding edges and features. The strategy of discarding edges alone may result in the loss of

important graph structural information, affecting the model's understanding of global and local relationships in the graph. The strategy of discarding edges excessively may lead to graph fragmentation and limit information transmission between nodes. Meanwhile, discarding key node features separately may lead to random dropout, resulting in the model being unable to fully utilize important features. Simply discarding features cannot utilize the topological structure of the graph, which can affect the overall performance.

To address these issues, we propose a novel graph augmentation strategy that relieves these drawbacks and simultaneously possesses the advantages of discarding edges and node features.

In GCL with adaptive augmentation (GCA) [33], node centrality is determined using metrics that assess the similarity between nodes. Specifically, the cosine similarity between the feature vectors of nodes u and v to quantify their similarity. This similarity measure is based directly on the nodes' feature vectors and can be expressed as follows [33]:

$$Cos_{uv} = \frac{X_u \cdot X_v}{||X_u|| \cdot ||X_v||}, \tag{3.1}$$

where Cos_{uv} represents the cosine similarity metric, which quantifies the degree of similarity between the feature vectors X_u and X_v.

Notably, SGCL [13] introduces a centrality measure of an edge connecting a pair of nodes, which is mathematically formulated as [13]:

$$\phi(u, v) = \begin{cases} Deg(u) + ncn(uv), if\ Cos_{uv} > R, \\ 0.001, otherwise, \end{cases} \tag{3.2}$$

where $Deg(u)$ serves as the degree of node u, $ncn(uv)$ represents the count of shared adjacent vertices for any two given nodes in a network, normalized by their individual degrees, $R = \beta + \frac{1}{Deg(u)+1}$, and the hyperparameter $\beta \in (0, 1)$ is a smoothing factor.

GCA introduces $M_{uv} = \log \phi(u, v)$ to alleviate the dominance of high-degree nodes in the learning process. Inspired by GCA, we propose the edge drop probability, which is defined as:

$$P_{ed} = min(\frac{M_{max} - M_{uv}}{M_{max} - \mu_1} \cdot P_e, P_\tau), \tag{3.3}$$

where M_{max} and μ_1 respectively serve as the maximum and average value of M_{uv}, P_e is the overall edge removal probability, and P_τ denotes a threshold probability designed to prevent excessive disruption of the graph structure.

Accordingly, the centrality score for each node is calculated based on the following formula:

$$\phi(u) = \frac{\sum_{k \in \zeta(u)} \phi(u, k)}{Deg(u)}, \tag{3.4}$$

where $Deg(u)$ denotes the count of adjacent nodes connected to vertex u, $\zeta(u)$ is the set of neighbors of node u, and $\phi(u)$ denotes the node centrality measure designed to quantify node importance.

Similar to edge drop probability, node feature probability can be calculated as follows:

$$P_{nf} = min(\frac{M_{max} - M_f}{M_{max} - \mu_2} \cdot P_f, P_\tau), \tag{3.5}$$

where $M_f = \log \sum_{u \in V} X_u \cdot \phi(u)$, and M_{max} and μ_2 respectively represent both the maximum and the average values of M_f.

At last, according to Eqs (3.3)–(3.5), we leverage the calculated edge drop probability P_{ed} and node feature probability P_{nf} to selectively and strategically discard edges and node features. In this way, the model can be more robust during the training process.

It is worth noting that we employ a new diffusion graph as our third augmented view. The diffusion mechanism has been widely applied in various fields. Learning degradation representations in diffusion models (LLDiffusion) [34] employs a diffusion mechanism to specify a degradation representation, where it learns the process of reversing degradation through a diffusion model for low-light image enhancement. All-in-oneweather-degraded image restoration (ADSM) [35] integrates the acquired degradation aware prompts into the temporal embedding of the diffusion model to improve degradation perception. Zhang et al. [36] provides a comprehensive and timely overview of recently published deep learning-based image deblurring methods, and suggests applying diffusion models to deblurring methods as a future direction. To construct the diffusion graph, we first add self-loops and apply symmetric normalization starting from the original adjacency matrix. We then apply a diffusion transform based on the personalized pagerank (PPR) kernel [37], which models a random walk with restart to aggregate multi-hop neighborhood information. This approach captures high-order structural dependencies by propagating information through multi-hop neighborhoods. Specifically, the process of constructing the diffusion graph structure can be formulated as follows [37]:

$$Q_{PPR} = \eta(I_N - (1-\eta)\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}})^{-1}, \tag{3.6}$$

where η is the teleport probability in a random walk, which controls the balance between local and global structural information, and I_N is a size-corresponding identity matrix.

The diffusion-based graph, as the third augmented view in our model, serves as a semantic guidance view, providing a stable reference for representation learning. By encoding the diffusion graph, the teaching network generates node representations with consistent semantics. These representations act as intermediate anchors, guiding the alignment between the two augmented views ($G1$ and $G2$) processed by the student network. This design helps alleviate the structural and semantic inconsistency issues brought about by random graph augmentation.

Furthermore, the inherent relationship among the various augmented views lies in the fact that they all stem from the original graph. Each graph focuses on different information that can be extracted, in a comprehensive and hierarchical manner.

3.1.2. Teacher-student teaching network

Based on previous research, we propose a novel teaching network to perceive local structural differences and global topological smoothness. It establishes multi-level contrastive objectives, ensuring rich and detailed supervision for graph encoders. The overall architecture of STGCN is illustrated in Figure 1.

The teaching network, highlighted in the middle, forms the core of our framework. This network comprises three graph encoders: two GNN-based student encoders and one GNN-based teacher encoder. Each encoder processes node representations from the original graph and its augmented

views, expressed as $Z_1 = f_g(X_1, A)$, $Z_2 = f_g(X, A_2)$, and $Z_0 = f_g(X, A_0)$. The primary innovation lies in its parameter update mechanism, which is governed by the interaction between teacher and student networks. The teacher encoder's parameters are not updated via gradients but rather through exponential moving average (EMA) [11] derived from the student networks. In the context of curriculum learning, we compare the original graph to textbooks. The teacher encoder distributes the textbook to student encoders and conducts teaching. After teaching guidance, the teacher encoder will be improved based on the performance of the student encoders. As shown in Figure 2, this guided update mechanism ensures that the teacher continuously refines its knowledge representation as the students improve, fostering a mutually supervised and iterative learning process. It maintains stability and provides a reference vector that enables smooth updates and reduces the fluctuation in the loss caused by node inconsistency between different views.

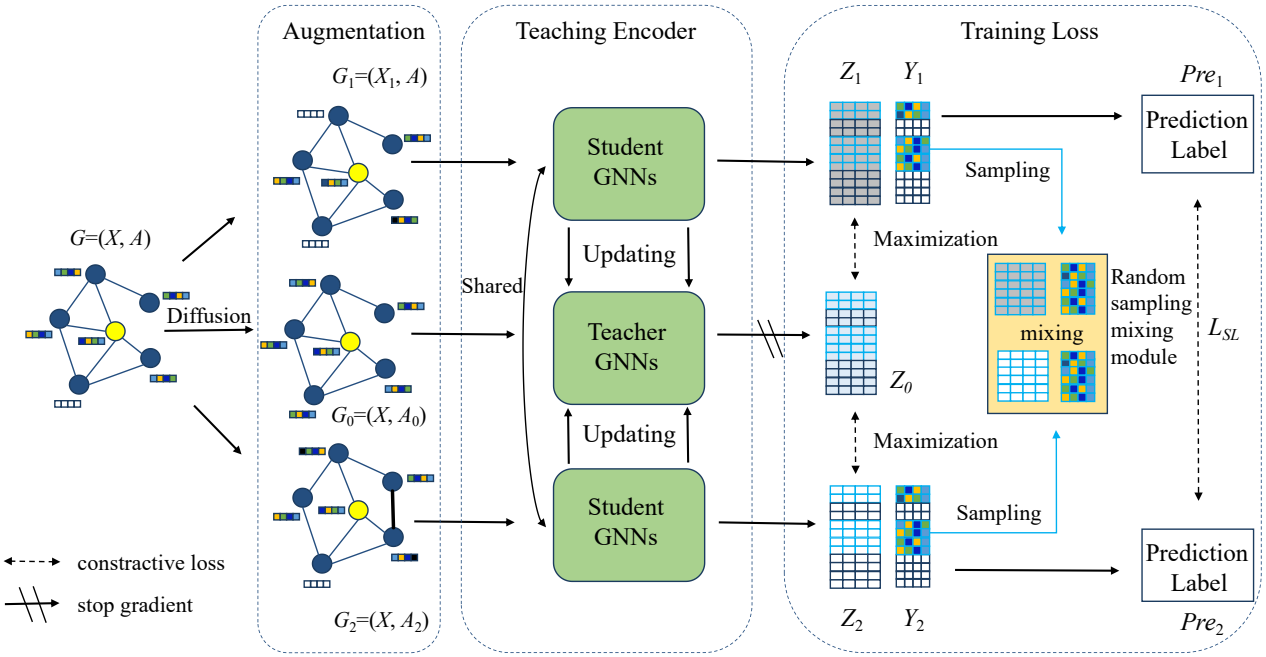


Figure 1. The proposed architecture of STGCN. Consider a graph G characterized by its node attribute matrix X, adjacency matrix A, and the corresponding node labels Y. We generate a pair of distinct augmented views G_1 and G_2. Both augmented views are processed by a shared encoder, while the original graph is passed through a separate GNN encoder. Two shared GNN encoders are used as students, and the other GNN encoder is used as a teacher, forming a novel teaching network together. Then, we obtain three node-level representations Z_1, Z_2, and Z_0, and STGCN evaluates the difference between the prediction labels by using cross-entropy loss (L_{SL}).

Given the variability in the number of neighbors for graph nodes, GNNs are particularly appropriate for this task as they aggregate neighborhood information to capture structural features; the GNN encoder such as GCN [14], GAT [15], GraphSAGE [16], How powerful are graph neural

networks (GIN) [38], and Modularity-based siamese simple GCN (MS-SGC) [39] can be integrated into our framework. Meanwhile, the original graph undergoes encoding via a GNN to produce lower-dimensional node representations, denoted as $f_g \in R^{N\times D}$. The original graph representation is $Z_0 = f_g(X_0, A_0)$. Projector network typically refers to a network structure used in deep learning models to map high-dimensional features to a low-dimensional space. It is mainly used for dimensionality reduction and feature projection, helping the model learn more concise representations in a smaller feature space. Inspired by BGRL [10], we find that omitting the projector network not only simplifies the architecture but also maintains or even enhances performance, and our framework eliminates the projector network. Additionally, labeled nodes in our framework play a critical role in preventing the collapse of representations, ensuring effective learning.

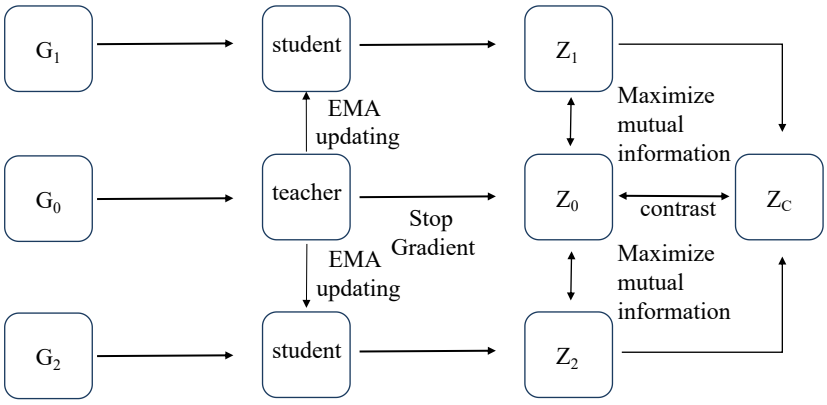


Figure 2. Simplified architecture of the proposed framework with three graph views.

3.1.3. Multi-view maximum mutual information

This section presents a set of contrastive loss functions based on the teaching network, designed to measure the mutual information between three views: original view Z_0 and its augmented views Z_1 and Z_2.

The node-level representations learned from the original view and its augmented views should demonstrate high consistency. To achieve this goal, we employ a contrastive loss function based on the normalized temperature-scaled cross-entropy loss [40]. This loss function encourages the node-level representations of the original view and its augmented views to be similar while pushing apart the representations of augmented views from other views in the mini-batch. For instance, in the mini-batch, taking the ith graph G_i and its first augmented view T_1 (G_i) as an example, the specific formula of contrastive loss function is as follows:

$$L_1 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(sim(Z_i^0, Z_i^1)/\xi)}{\sum_{j=1}^{N}\exp(sim(Z_i^0, Z_j^1)/\xi)}, \tag{3.7}$$

where N is the number of original graphs in a mini-batch, ξ is the temperature hyperparameter, and $sim(Z_i^0, Z_i^1) = (Z_i^0)^T \cdot Z_i^1/(\|Z_i^0\| \cdot \|Z_i^1\|)$ is the cosine similarity function.

For graph G_i and its second augmented view $T_2(G_i)$, we derive a contrastive loss function aimed at

maximizing mutual information between these two graphs, expressed as follows:

$$L_2 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(sim(Z_i^0, Z_i^2)/\xi)}{\sum_{j=1}^{N}\exp(sim(Z_i^0, Z_j^2)/\xi)}. \tag{3.8}$$

Finally, we can combine the above loss functions, and gain a new contrastive loss function of our framework as follows:

$$L_{GL} = \alpha L_1 + (1-\alpha)L_2, \tag{3.9}$$

where α is a non-negative tuning parameter.

The loss L_{GL} encourages graph representations to maximize mutual information across augmented views while maintaining sufficient discrimination among different views.

In Eq (3.9), it is expected that each augmented view maintains a positive relationship with the original graph.

3.1.4. Random sampling mixing

Although the self-supervised loss function is effective, the node representations may fail to distinguish between distinct classes due to the absence of label information. To overcome this limitation, we propose a label propagation technique. Specifically, we demonstrate that it is possible to organize unlabeled nodes into distinct groups corresponding to their classes by enforcing consistent proximity to labeled nodes of the same class across different augmented views. To achieve this goal, we randomly select b labeled nodes (b is determined by the class with the fewest samples among all classes) from each class to construct the randomly sampled support set S, and denote Z^S as support node representations. Then, by computing the similarity between the support set S of labeled nodes and all nodes in the augmented views, we strengthen the alignment of class assignments through similarity-based augmentation. The objective is to predict the class distribution, which is formally defined as follows:

$$pre_i^1 = \sum_{(Z_p^1, Y_p)\in Z^S(i)}\frac{\exp(sim(Z_i^1, Z_p^1)/\tau)\cdot Y_p}{\sum_{k\in Z^S(i)} exp(sim(Z_i^1, Z_k^1)/\tau)}, \tag{3.10}$$

$$pre_i^2 = \sum_{(Z_p^2, Y_p)\in Z^S(i)}\frac{\exp(sim(Z_i^2, Z_p^2)/\tau)\cdot Y_p}{\sum_{k\in Z^S(i)} exp(sim(Z_i^2, Z_k^2)/\tau)}, \tag{3.11}$$

where sim(. , .) calculates the cosine similarity between two vectors, Z_i^1 is the learned representation for the first view and similarly, Z_i^2 is the latent representation for the second view, pre_i can be considered as the predicted class distributions for each node $v_i \in V$, and τ denotes a temperature hyperparameter.

After calculating two predicted class distributions pre_i^1 and pre_i^2 , the supervised contrastive loss function is defined:

$$L_{SL} = \sum_{i\in V} H(pre_i^1, pre_i^2), \tag{3.12}$$

where, H denotes the cross-entropy between two distinct distributions pre_i^1 and pre_i^2 for each node $v_i \in V$.

In addition, there is another issue worth discussing: The contrastive information learned in each round is only utilized once. To better leverage underutilized information from different perspectives, we employ a random sampling mixing module to create new samples to fill in the gaps and actively participate in the contrastive framework. The presence of filling-gap samples allows for the reemphasis of information that has been underexplored from various perspectives. Based on the previously defined randomly sampled support set S, we extract anchor support set representation Z^A from G_1 and positive sample support set representation Z^B from G_2. $\forall \beta_i \sim U(0,1)$, where U denotes a uniform distribution, and we can generate a new mixing node representation Z_i^C based on the corresponding nodes Z_i^A and Z_i^B. This process can be described as:

$$Z_i^C = \beta_i \cdot Z_i^A + (1 - \beta_i) \cdot Z_i^B. \tag{3.13}$$

This process involves generating a mixed representation based on the embedding of node i. Upon the completion of mixed representation generation for all nodes, a new graph view, Z_i^C, is generated. This procedure can be repeated to generate multiple views as required to facilitate the presentation of our model, but we only generate one Z_i^C to mine underutilized information. Then, we extract the original representation Z_i^D from the original graph on the basis of support set S. To capture the consistency between the filling-gap and original views, the contrastive loss function is defined as:

$$L_{CD} = \sum_{i=1}^{N} -\log \frac{\exp(sim(Z_i^C, Z_i^D))}{\sum_{j=1}^{N} \exp(sim(Z_i^C, Z_j^D))}. \tag{3.14}$$

It is important to note that Z_i^D serves as a fixed teacher embedding: Its parameters are not updated through this loss function, and it does not act as a learnable anchor or negative sample. In this way, Z_i^C is encouraged to align with the stable diffusion-based semantics, while still incorporating complementary information from multiple augmented views.

At last, the loss function of this supervised graph contrastive learning can be described as:

$$L_{SC} = (L_{SL} + L_{CD})/2. \tag{3.15}$$

The supervised graph contrastive loss L_{SC} integrates two complementary objectives, the supervised label-consistent contrastive loss and the filling-gap consistency loss. It balances semantic consistency, representation smoothness, and cross-view robustness, leading to more reliable node representations under limited supervision.

3.1.5. Node-based consistency regularization

In our framework, we utilize a GCL strategy to improve node consistency under different graph views. By comparing the contrastive loss functions, we ensure that the embeddings of each node remain consistent across these perspectives while remaining distinguishable from the embeddings of other nodes. To learn node representations, we minimize the cosine distance of the same node in two augmented views to ensure the robustness and consistency of node representations. The corresponding formulation is as follows:

$$L_{NC} = -\frac{1}{N} \sum_{i=1}^{N} \frac{Z_i^1 \cdot Z_i^2}{\|Z_i^1\| \cdot \|Z_i^2\|}, \tag{3.16}$$

where, Z_i^1 and Z_i^2 denote the representations corresponding to the first and second views.

The loss L_{NC} is designed to enforce node-level representation consistency across different augmented graph views. Unlike pairwise contrastive objectives that rely on negative samples, L_{NC} focuses solely on positive pairs of the same node, which simplifies optimization and avoids potential instability caused by negative sampling.

Notably, the aforementioned loss function differs from that of BGRL [10]. In BGRL, two independent encoders are employed: One is updated by minimizing the semantic inconsistency between node representations across different augmented views, while the other employs an EMA to adjust its parameters, preventing the node representations from collapsing into trivial solutions. In contrast, our approach utilizes one shared encoder and introduces monitoring signals within the framework to ensure that node representations do not converge to overly simplistic solutions.

Finally, we formulate the overall loss of our framework by combining several distinct loss terms L_{GL}, L_{SC}, and L_{NC}, each with respective coefficients λ_1, λ_2, and λ_3. Additionally, we add the cross-entropy loss function L_{sup}, which is defined over a set of labeled nodes. So, the final objective loss function is as follows:

$$L_{Training} = L_{sup} + \lambda_1 L_{GL} + \lambda_2 L_{SC} + \lambda_3 L_{NC}. \tag{3.17}$$

The overall pipeline of STGCN shown in Algorithm 1.

Algorithm 1 The learning process of STGCN.

Require: Feature matrix X; Adjacency matrix A; True label Y; Maximum number of iterations T; Encoder f_g by random initialization.

While not reaching T do
  Generate augmented view $G_1 = (X_1, A)$, mask node features and drop partial edges based on edge centrality.
  Generate augmented view $G_2 = (X, A_2)$, mask node features and drop partial edges based on edge centrality.
  Generate diffusion graph $G_0 = (X, A_0)$ based on PPR kernel.
  Get node embeddings through GNNs as encoder: $Z_1 = f_g(X_1, A)$, $Z_2 = f_g(X, A_2)$, $Z_0 = f_g(X, A_0)$.
  Calculate model predictions as $pre_1 = p(Y|Z_1)$ and $pre_2 = p(Y|Z_2)$.
  Calculate graph contrastive loss L_{GL} based on Eq.(9).
  Calculate graph contrastive loss L_{SC} based on Eq.(15).
  Calculate graph contrastive loss L_{NC} based on Eq.(16).
  Calculate cross-entropy loss L_{sup} .
  Update g, m by gradient descent according to the overall loss function $L_{Training}$ in Eq.(17).
end while
return model f_g.

4. Experiments

In Section 4, we evaluate the effectiveness of the proposed framework introduced in Section 3.

4.1. Baselines

We compare STGCN with 17 baseline methods:

1) Two traditional baseline methods: Multi-layer perceptron (MLP), GCN [14].
2) Four graph convolution based methods: GAT [15], SGC [41], graph random neural networks (GRAND) [42] and predict then propagate (APPNP) [37]. SGC simplifies the original GCN model through the removal of repeated feature transformations and nonlinearity operations. GRAND and APPNP alleviate the limited receptive field problem inherent in existing message passing models.
3) Two self-supervised baseline methods: BGRL [10] and GRACE [30]. Both methods aim to maximize the consistency of node representations across two augmented views of the same graph.
4) Three label-efficient baseline methods: GLP [21], IGCN [21] and contrastive graph poisson networks (CGPN) [43]. GLP and IGCN utilize low-pass graph filters during message propagation to enhance label efficiency. CGPN efficiently propagates limited labels across the entire graph, while using contrastive loss to extract information from unlabeled nodes.
5) Six pseudo-label-based baseline methods: self-training [18], co-training [18], GraFN [12], M3S [20], SGCL [13] and a pseudo-labeling approach (PLD-FSNC) [44]. PLD-FSNC leverages knowledge distillation to improve pseudo-label quality and enhance model generalization in complex network analysis.

4.2. Datesets

To verify the efficacy of our proposed framework, we evaluate it on five widely used node classification datasets, namely Cora [45], CiteSeer [45], PubMed [46], Amazon Photos, and Amazon Computers [47]. Table 1 shows the details of these datasets. A brief introduction to the datasets used is presented below:

1) Cora, CiteSeer, and PubMed: They are widely used benchmark datasets in the GNN field, mainly for node classification tasks. These datasets have a similar structure to citation networks, where nodes represent research papers and edges represent citation relationships between them. Each node is associated with a feature vector and is typically constructed using bag-of-words representation and assigned a category label for supervised learning tasks.

2) Amazon Photo and Amazon Computers: For the sake of convenience, we abbreviate them as Am. Photos and Am. Comp, respectively. Both datasets are derived from the same type of item network of Amazon products, where nodes represent products, and edges represent co-purchasing relationships between products (i.e., items frequently purchased together by users).

Table 1. Basic information of the datasets.

Dataset	Nodes	Edges	Features	Classes
Cora	2708	5429	1433	7
CiteSeer	3327	4732	3703	6
PubMed	19,717	44,338	500	3
Am. Photos	7650	119,081	745	8
Am. Comp	13,752	245,861	767	10

4.3. Experimental settings and parameter study

To evaluate the performance of the proposed STGCN framework on node classification under limited label scenarios, we conduct experiments using 20 random data splits and report the average results across these splits for each dataset, and the average performance across these splits is reported for each dataset. Specifically, on the Cora and CiteSeer datasets, we use training label rates of 0.50%, 1.00%, and 2.00%, and for PubMed, the training sizes are set to 0.03%, 0.06%, and 0.10%. In the co-purchase datasets, we evaluate the framework using label rates of 0.15%, 0.20%, and 0.25%. Our STGCN framework is implemented with a two-layer GCN architecture, each layer consisting of [128, 128] units. The activation function used for all layers is PReLU, providing flexibility in modeling non-linear relationships. For training, dropout with a rate of 0.60 is applied on the CiteSeer, Cora, and PubMed datasets to prevent overfitting. The model is trained for up to 1000 training epochs, and the Adam optimizer [48] is used for gradient-based optimization. The training process includes an early stopping mechanism, where training is halted if the validation accuracy fails to improve for a specified number of epochs. This helps ensure that the model is not trained longer than necessary. Then, the precision of the framework is validated through the evaluation on the testing dataset, and the best performance is highlighted in bold.

Table 2. Optimal hyperparameter configurations for each dataset.

Hyperparameter	Cora	CiteSeer	PubMed	Am. Photos	Am. Comp
lr	1×10^{-3}	1×10^{-3}	1×10^{-2}	1×10^{-2}	5×10^{-3}
wd	1×10^{-3}	1×10^{-3}	1×10^{-3}	1×10^{-2}	1×10^{-2}
τ	0.80	0.90	0.90	0.90	0.90
P_e^1	0.50	0.50	0.50	0.30	0.30
P_f^1	0.50	0.50	0.50	0.10	0.10
P_e^2	0.30	0.30	0.30	0.50	0.50
P_f^2	0.10	0.10	0.10	0.10	0.10
λ_1	0.01	1.00	0.50	0.10	0.50
λ_2	1.00	0.10	0.10	0.50	0.50
λ_3	0.50	2.00	0.50	2.00	0.50

In Table 2, the hyperparameters of the STGCN framework are determined through a small-scale grid search, encompassing all necessary parameters across the five benchmark datasets. This process identifies the optimal configuration for each task through systematic exploration of the parameter space. The following hyperparameter ranges are considered during the grid search: For STGCN, we search for weight decay and learning rate in $\{1 \times 10^{-2}, 1 \times 10^{-3}, 1 \times 10^{-4}, 5 \times 10^{-4}\}$ and $\{1 \times 10^{-1}, 1 \times 10^{-2}, 5 \times 10^{-2}, 1 \times 10^{-3}, 5 \times 10^{-3}\}$, respectively. These ranges are selected based on observational experience and past work to ensure that the model is flexible enough to achieve optimal performance without excessive adjustments. According to the method described in Section 3.3, our approach incorporates dual augmentation processes. For both the primary and secondary perspectives, the corresponding probability parameters, denoted as P_e^1 and P_f^1 for the first view, and P_e^2 and P_f^2 for the second view, are configured within the range of 0.00 to 0.50. This range ensures that the graph's original structure is preserved and provides sufficient diversity for learning. In our experiments, the temperature hyperparameter P_τ is set to 0.10. The hyperparameter τ is selected from the set

$\{0.80, 0.90\}$, and the balance hyperparameters λ_1, λ_2, and λ_3 are tuned within the range $\{0.01, 0.10, 0.50, 1.00, 2.00\}$. Setting the beta hyperparameter to 0.01, we conduct a complete grid search on all other parameters to find the best combination for optimal performance. This final search configuration ensures that the model is trained by a balanced set of hyperparameters to maximize its performance in the dataset.

4.4. Performance analysis

Table 3 illustrates the node classification performance of different methods under different labeling rates. Part of the data is directly sourced from reference [12]. Our analysis shows that at different labeling rates, STGCN outperforms various complex GNN encoders, such as APPNP and GRAND, in node classification accuracy. Additionally, it surpasses highly label-efficient GNNs, like IGCN and GLP, as well as contrastive loss-based approaches, such as CGPN, demonstrating its effectiveness even when labeled data is limited. It is worth noting that when compared to self-supervised methods, such as GRACE and BGRL, STGCN shows superior performance. This is achieved through a unified GNN encoder combined with a simplified self-supervised loss, offering a more efficient and effective learning process.

Table 3. Performance of semi-supervised node classification.

Datasets	Cora			CiteSeer			PubMed		
Label rate	0.50%	1.00%	2.00%	0.50%	1.00%	2.00%	0.03%	0.06%	0.10%
MLP	32.25	38.75	45.54	33.08	44.08	47.12	53.51	56.80	62.24
GCN	57.00	67.37	73.38	45.65	55.62	61.58	60.29	65.03	74.75
GAT	59.58	68.76	73.74	49.70	59.77	63.72	64.16	65.12	74.18
SGC	50.17	64.62	70.55	45.03	56.87	64.63	59.52	63.50	72.94
GRAND	55.52	71.94	74.91	47.78	58.41	65.34	55.87	61.23	72.43
APPNP	62.03	71.46	76.88	41.78	54.70	62.84	63.16	64.12	73.17
GLP	56.95	68.27	72.98	41.51	54.83	63.07	56.72	60.84	73.46
IGCN	58.81	70.10	74.34	43.28	57.00	64.62	57.50	62.06	73.13
CGPN	64.21	70.54	72.97	53.90	63.70	65.15	64.55	67.58	71.42
BGRL	61.75	68.76	73.66	54.68	63.76	67.76	65.75	68.87	75.92
GRACE	60.96	68.68	74.67	52.02	58.00	63.77	64.87	68.36	75.93
Self-training	57.29	70.74	75.41	46.27	60.37	66.48	57.35	65.14	72.87
Co-training	62.76	68.73	74.06	43.77	54.76	61.14	63.02	68.16	74.25
GraFN	66.74	72.51	77.20	57.49	66.48	69.89	65.91	68.41	75.74
M3S	64.48	72.92	76.41	55.08	65.75	67.65	61.52	64.61	73.19
SGCL	66.90	73.11	78.04	58.45	66.90	70.19	68.40	70.96	76.87
PLD-FSNC	66.95	73.50	78.22	60.55	67.32	70.41	69.86	71.38	77.25
STGCN	**67.70**	**73.72**	**78.30**	**62.50**	**68.60**	**70.53**	**72.82**	**73.16**	**79.30**

As demonstrated in Table 3, STGCN demonstrates clear performance improvements across five datasets, consistently outperforming other methods at various label rates. For example, on the PubMed dataset, STGCN improves upon M3S by 18.36%, 13.23%, and 8.34% for label rates of

0.03%, 0.06%, and 0.10%, respectively. Overall, STGCN outperforms methods like CGPN [43], BGRL, and GCA on all datasets, emphasizing the benefits of integrating self-supervised learning techniques with semi-supervised methodologies in machine learning applications. It exceeds both semi-supervised approaches and self-supervised approaches, showing its broader applicability and effectiveness.

However, there are certain differences in the experimental results on the Cora, Citeseer, and PubMed datasets, which are attributed to the variations in structural density, feature sparsity, and semantic consistency among different graph datasets. Three datasets under the condition of the lowest label rate all yield different and varying results. On Cora, this dataset achieves a classification accuracy of 67.70%. This dataset possesses a medium-density graph structure and abundant node features, enabling the diffusion graph to effectively capture high-order structural information. However, excessive propagation may also introduce certain oversmoothing issues. On CiteSeer, the accuracy is 62.50%. Due to the sparse connections and significant feature noise in this dataset, diffusion may lead to noise propagation, thereby limiting the improvement of model performance. On PubMed, the accuracy is 72.82%. Our model achieves the best performance compared to the other two, indicating that the diffusion graph and teacher-student teaching network exhibit more advantages on graphs with stable structures and strong semantic consistency.

Table 4. Performance of node classification results on Am. Comp and Am. Photos.

Datasets	Am. Comp			Am. Photos		
Label rate	0.15%	0.20%	0.25%	0.15%	0.20%	0.25%
MLP	41.32	43.20	49.99	30.75	32.65	39.56
GCN	63.72	67.83	72.76	67.70	71.73	76.75
GAT	67.18	71.17	73.83	74.25	75.44	81.13
SGC	60.68	65.25	69.28	56.96	62.65	70.67
GRAND	68.03	72.72	75.78	73.80	75.84	82.32
APPNP	68.54	72.48	74.28	75.55	78.47	82.76
GLP	62.98	68.57	70.71	63.19	67.97	75.17
IGCN	65.49	70.06	71.04	71.26	73.27	77.94
CGPN	65.38	67.94	70.73	74.15	76.88	81.54
BGRL	68.81	73.05	75.12	74.25	78.27	83.12
GRACE	65.27	67.78	71.79	70.16	71.88	77.32
Self-training	61.34	65.95	68.67	61.92	65.27	71.33
Co-training	67.06	71.62	71.34	72.85	74.65	79.92
GraFN	71.73	74.26	77.37	79.25	80.87	85.36
M3S	61.53	66.32	68.12	63.95	67.67	73.38
SGCL	72.05	74.99	78.19	79.99	81.25	85.20
PLD-FSNC	72.32	75.01	78.74	80.22	81.73	85.44
STGCN	**72.64**	**75.16**	**79.24**	**81.85**	**82.23**	**86.18**

Table 4 presents a comprehensive comparison of node classification performance on the Am.

Comp and Am. Photos datasets under varying label rates. The results demonstrate that on both Am. Comp and Am. Photos, STGCN achieves the highest node classification accuracy at almost every label rate. Furthermore, Table 4 also presents a comparative analysis between STGCN and traditional methods such as MLP, GNNs including GCN and GAT, as well as contrastive learning-based approaches such as GRACE and BGRL. The results highlight STGCN's ability to exploit both labeled and unlabeled information effectively and significantly enhance its learning ability by effectively integrating label information with graph structure through advanced strategies. In addition, unlike pseudo-label strategies such as co-training, self-training, or M3S, which either focus primarily on labeled data or exploit unlabeled data in a two-stage manner, STGCN jointly optimizes a unified encoder to leverage labels directly within a contrastive framework, thereby extracting richer structural information from unlabeled nodes. The experimental outcomes highlight the efficacy of our proposed approach and indicate that STGCN achieves competitive performance in semi-supervised node classification tasks.

Since Am. Comp and Am. Photos are recognized as medium-sized graphs, the experimental results demonstrate that our model maintains stable performance on these datasets. It indicates that the proposed framework can be effectively extended to graphs with larger scales and stronger connectivity. Although our method integrates diffusion-based graph construction and teacher-student teaching network, these components are efficiently incorporated into the training process. The diffusion-based graph only needs to be constructed once and can be reused across multiple training epochs, while the teaching network is updated by using EMA. They eliminate the need to introduce additional backpropagation costs. Therefore, the overall computational complexity is comparable to existing GCL methods, and the model can be trained on large-scale graphs without incurring excessive memory or runtime overhead. So, our model can be extended to large-scale graphs. Furthermore, the effectiveness of the proposed STGCN framework is evaluated through semi-supervised node classification, with the average classification accuracy and corresponding standard deviation provided over 10 independent runs, as shown in Table 5. Our analysis reveals that although popular self-supervised methods like MVGRL, DGI, GCL via graphical mutual information maximization (GMI) [49], GRACE, and sub-graph contrast (Subg-Con) [50] have demonstrated effectiveness in learning graph representations, STGCN outperforms them in both accuracy and efficiency. By more deeply constructing the complex dependencies and relationships among unlabeled nodes, STGCN is able to capture structural information and surpass these traditional approaches. In addition, unlike methods such as augmentation-free GCL (AF-GCL) [51] and Neighbor contrastive learning (NCLA) [52], STGCN is designed to function within a semi-supervised learning framework. It makes use of a small number of labeled nodes to directly guide the contrastive learning process, enabling more effective supervision during representation learning. While methods such as label-guided GCL (LGGCL) [53] and CG^3 also incorporate labeled nodes, they lack mechanisms to propagate label information to large-scale unlabeled nodes of the graph. STGCN introduces a label augmentation module that injects supervisory signals into the contrastive loss, significantly improving representation quality and learning effectiveness. As shown in Table 5, STGCN consistently achieves top results across various datasets. The reported average accuracies and standard deviations over repeated runs confirm the statistical significance and robustness of its performance. Additionally, STGCN exhibits remarkable generalization capabilities, maintaining superior performance across diverse benchmark datasets. This stability highlights the framework's

robustness to different graph structures, label distributions, and levels of supervision. STGCN achieves notable accuracy improvements of 3.10%, 2.60%, and 2.50% on three standard citation datasets in comparison with the highest-performing unsupervised approach Subg-Con. These results demonstrate the framework's ability to effectively utilize label information, leading to the creation of robust node representations. Remarkably, STGCN outperforms previous approaches, surpassing CG^3 by 2.70% on the CiteSeer dataset, HCP by 1.70% on Cora, and NCLA by 1.40% on PubMed, further solidifying its superior performance across different datasets.

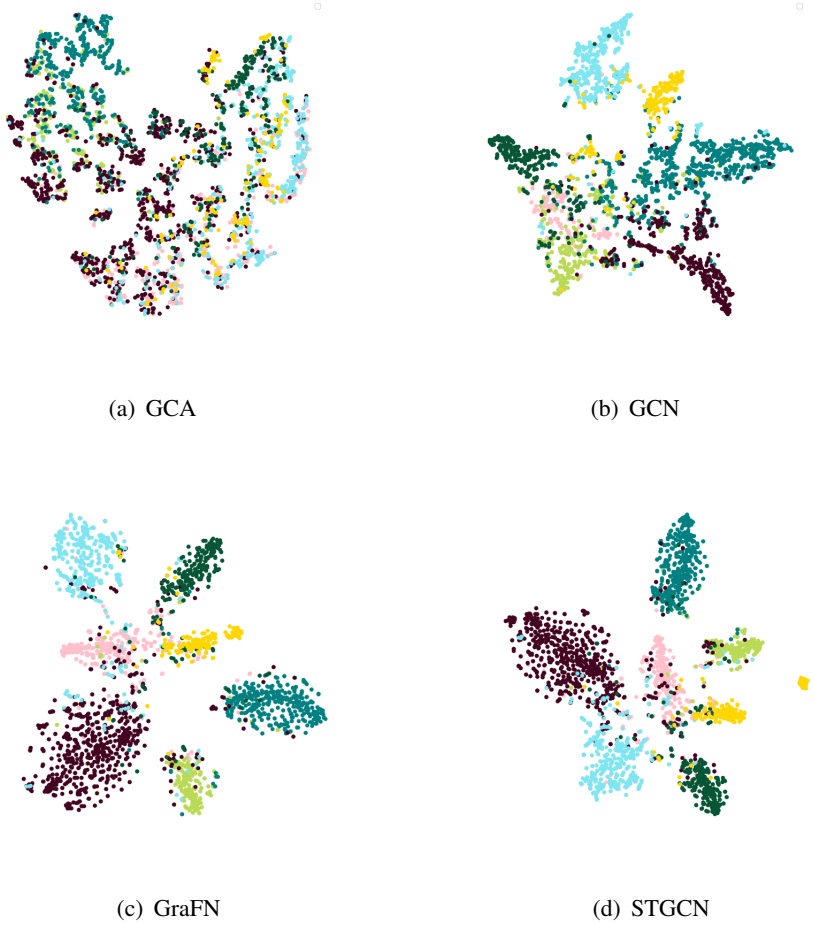
Table 5. The performance of node classification in comparison with contemporary approaches.

Methods	CiteSeer	Cora	PubMed
GCN	70.5 ± 0.4	81.5 ± 0.2	79.0 ± 0.5
GAT	71.0 ± 0.5	83.1 ± 0.3	79.0 ± 0.4
APPNP	71.8 ± 0.6	83.7 ± 0.5	80.4 ± 0.2
SGC	71.9 ± 0.1	81.0 ± 0.0	78.9 ± 0.0
DGI	71.5 ± 0.7	81.7 ± 0.6	77.3 ± 0.6
Subg-Con	73.3 ± 0.2	83.6 ± 0.5	81.1 ± 0.1
GRACE	71.8 ± 0.6	80.1 ± 0.4	79.6 ± 1.1
MVGRL	72.6 ± 0.7	83.2 ± 0.5	79.4 ± 0.3
GMI	73.2 ± 0.3	82.8 ± 0.2	80.1 ± 0.3
NCLA	71.7 ± 0.9	82.2 ± 1.6	82.0 ± 1.4
AF-GCL	72.1 ± 0.4	83.3 ± 0.1	79.1 ± 0.7
CG^3	73.6 ± 0.8	83.4 ± 0.7	80.2 ± 0.8
HCP	74.5 ± 0.7	84.3 ± 0.5	82.3 ± 0.7
LGGCL	74.6 ± 0.5	84.7 ± 0.5	82.8 ± 0.1
GRE^2-MDCL	74.2 ± 0.5	84.9 ± 0.5	82.8 ± 0.5
SGCL	75.1 ± 0.6	85.2 ± 0.5	83.0 ± 0.5
STGCN-GAGE	**72.2 ± 0.5**	**84.8 ± 0.5**	**80.2 ± 0.5**
STGCN-GAT	**73.8 ± 0.5**	**84.3 ± 0.5**	**81.4 ± 0.5**
STGCN	**75.6 ± 0.5**	**85.8 ± 0.5**	**83.2 ± 0.5**

To evaluate the adaptability of the proposed framework to different GNN encoders, we further replaced the default GCN encoder with alternative designs, including GAT and GraphSAGE, while keeping all other settings unchanged. The experimental results, as shown in Table 5, indicate that the proposed method is not restricted to a specific GNN architecture. Although the performance varies depending on the encoder capacity, the relative gains remain stable, demonstrating the general adaptability of our framework.

In addition, as illustrated in Figure 3, we employ T-distributed stochastic neighbor embedding (T-SNE) to embed node embeddings into a two-dimensional space to intuitively demonstrate the classification performance of our framework. The visualization compares STGCN with multiple important benchmark models, including GCN, GCA, and GraFN. Although GraFN performs equally

well in classification tasks, our framework demonstrates the ability to generate more compact cluster node embeddings for most categories. The results reveal that STGCN exhibits significantly clearer separation effects in node embeddings compared to GCN and GCA. This indicates that the node embeddings extracted by STGCN are more advantageous for downstream tasks. Overall, the classification performance demonstrated by STGCN proves the effectiveness of our proposed method.



(a) GCA (b) GCN

(c) GraFN (d) STGCN

Figure 3. T-SNE visualization of node embeddings on the Cora dataset.

Although STGCN is primarily evaluated on node classification in our work, the framework structure proposed in our work is not limited to this specific task. STGCN can also accomplish downstream tasks such as link prediction or graph classification. For link prediction, STGCN enhances structural consistency across multiple views. Its generated embeddings can capture relational patterns that are beneficial for predicting missing or future links. For graph classification, STGCN can be extended by applying graph-level pooling operations over node representations, followed by a task-specific classifier. The multi-view design of STGCN allows it to produce node embeddings that can be effectively aggregated for graph-level inference. We will conduct an empirical evaluation of these tasks in future work, and we believe the current results have fully demonstrated the powerful potential of STGCN as a general framework for graph representation learning.

4.5. Additional experiments

4.5.1. Comparison study

As shown in Figure 1, we propose a diffusion graph as our augmentation method based on the PPR kernel for the third channel. This approach helps to gradually propagate node information from labeled nodes, effectively capturing the overall structural properties of the original graph. Although other methods have achieved similar goals, we compare our diffusion graph with the following three methods on the Cora dataset:

Feature graph [30]: Constructs a graph structure based on feature similarity and update features through sparse matrix multiplication.

Smooth graph [41]: Enhances global smoothness and suppresses noise by constructing a smoothing matrix that combines the probability transition matrix and a uniform distribution matrix.

Self-attention graph [15]: Computes attention weights for edges, enabling weighted aggregation and feature updates for nodes.

As shown in Table 6, our proposed method outperforms other approaches under different label rates on the Cora dataset. This superior performance is attributed to our method's ability to effectively balance local and global information and simultaneously enhance the smoothness and robustness of the graph.

Table 6. Comparison of node classification results between different augmented views on Cora dataset.

Datasets	Cora		
Label rate	0.15%	0.2%	0.25%
Feature graph	67.45	73.46	77.63
Smooth graph	67.49	73.58	77.68
Self-attention graph	66.35	73.08	77.68
Diffusion graph	**67.70**	**73.72**	**78.30**

4.5.2. Hyperparameter analysis

In Figure 4, we investigate the influence of key hyperparameters on STGCN. The coefficients λ_1, λ_2, and λ_3 play distinct roles in our framework's loss function, and their values significantly impact the framework's overall performance. To gain deeper insight into their impact, we perform a set of ablation experiments, varying each coefficient individually from the set {0.00, 0.01, 0.10, 0.50, 1.00, 2.00} while holding the others constant. As shown in Figure 4, we conduct comprehensive experiments across five benchmark datasets and report the resulting performance. The coefficient λ_1 modulates the loss function of mutual information, and we notice that the model achieves optimal performance when it is usually set between 0.01 and 1.00. For λ_2, which is used to adjust the loss of this supervised graph contrastive learning, we notice that when λ_2 is set to 0.50, our framework's performance is usually the best, and the Am. Photos dataset's parameter setting to 2.00 is optimal. The last coefficient λ_3, is associated with the loss of node representations. We usually set it to 0.50, but there are some minor differences on different datasets. Our experimental results indicate that on datasets such as Cora, Am. Comp, and PubMed, the influence of hyperparameters λ_1, λ_2, and λ_3 remains consistent, indicating

strong model stability. However, on datasets like CiteSeer and Am. Photos, the framework exhibits higher sensitivity to hyperparameters, with noticeable performance fluctuations. This result indicates that although the model has good robustness and generalization on most datasets, it still requires fine-tuning of hyperparameters to adapt to its unique structure and distribution characteristics on certain datasets. This comparison of consistency and variability further emphasizes the robustness of the framework, as well as its dependence on hyperparameter selection for different datasets.

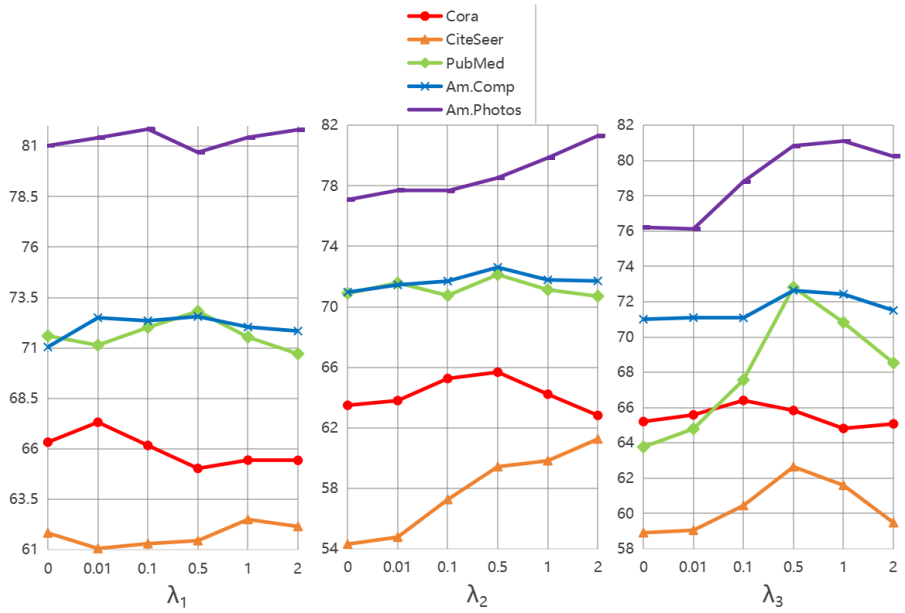


Figure 4. The performance of STGCN under values of different coefficients λ_1, λ_2, and λ_3.

4.5.3. Ablation study

Through ablation experiments, we can better understand the internal mechanisms of the model, the roles of each component, and their applicability to different datasets, providing important references for optimizing the model and improving its performance.

To demonstrate the importance of each part of our framework, ablation studies are conducted on three reference datasets, Cora, CiteSeer, and PubMed, at label rate settings of 1.00%, 0.50%, and 0.03%. As shown in Figure 5, relying on a single subset of loss functions, such as supervised loss and mutual information loss, may lead to insufficient supervisory signals and higher error rates. However, the complete integration of the four loss components is crucial for fully utilizing available label information and achieving optimal performance. This is mainly due to the complementarity of different losses, collectively improving learning ability and model universality.

In addition, to illustrate the contribution of each module in our model, we selectively remove key modules while keeping other settings unchanged. Specifically, we disable the update of the teaching network EMA, replace the diffusion map with the original adjacency matrix, and disable the random sampling mixing module. As shown in Figure 6, when any module is removed, the performance of the model decreases significantly, indicating that each module makes a positive contribution to the model. We observe that the performance on each dataset decreases when diffusion graphs are not used. It fully demonstrates that diffusion graphs help gradually propagate node information from labeled nodes,

effectively capturing the overall structural characteristics of the original graph to improve performance.

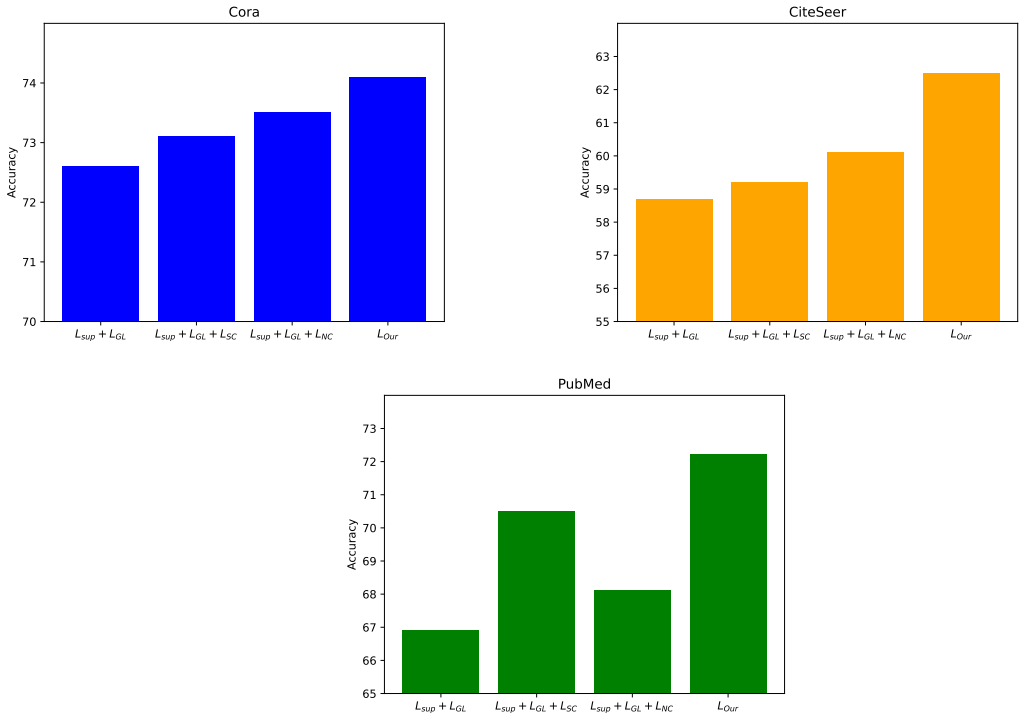


Figure 5. The results of the ablation study on STGCN framework.

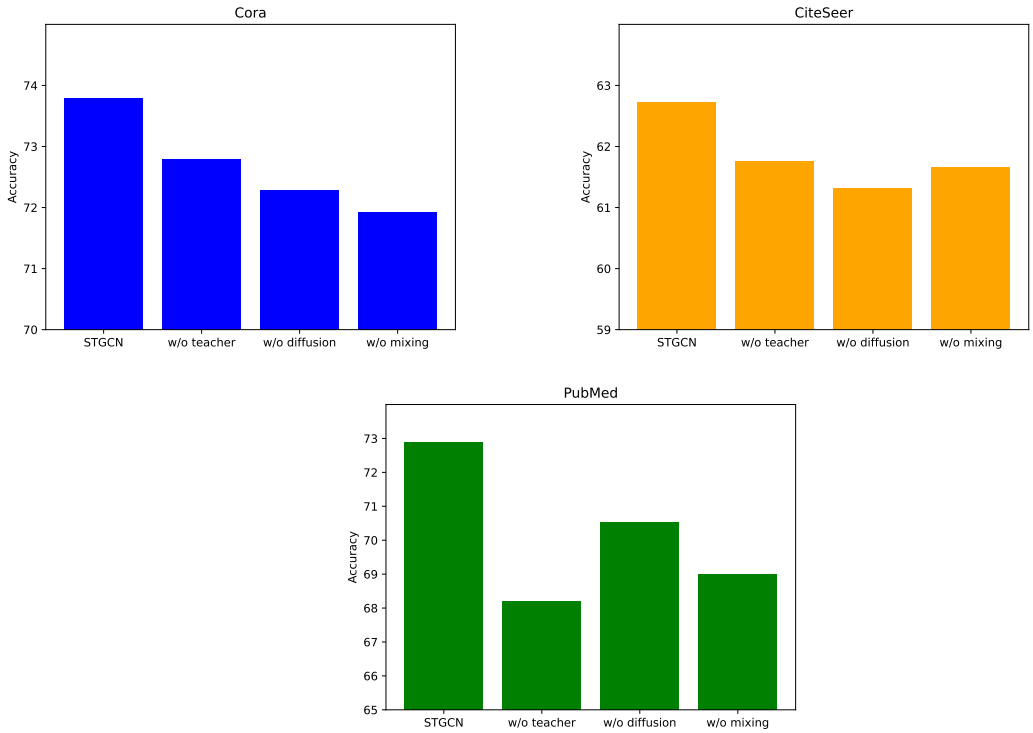


Figure 6. The results of the ablation study on STGCN framework.

4.5.4. Complexity analysis

In this subsection, we analyze the complexity of the model in terms of training time, and the GPU and CPU memory required for operation.

As shown in Table 7, the complexity of our method is basically within the normal range. Specifically, our model has lower GPU and CPU memory requirements. Compared with some GCL methods, our model possesses certain advantages. However, since the proposed framework incorporates a diffusion-based graph construction and a teacher student teaching network into the overall contrastive learning process, there is no obvious advantage in training time. This is primarily because the diffusion graph introduces additional message-passing operations, while the teaching network requires maintaining an extra encoder during training.

Table 7. Analysis of the complexity of various models on the Cora dataset.

Model	Training Time (Epoch/Batch)	CPU Memory	GPU Memory
GCN	0.01s	11.4GB	1.8GB
GAT	0.04s	11.5GB	3.0GB
GRACE	0.02s	12.6GB	1.4GB
BGRL	0.86s	13.1GB	3.1GB
GraFN	0.80s	12.7GB	3.6GB
STGCN	0.65s	11.4GB	2.3GB

5. Conclusions

This paper proposes a new framework for semi-supervised GCL, STGCN, to address the key challenge of node classification with few labels. Using an innovative centrality function, STGCN introduces a new graph augmentation strategy that combines edge dropping and feature masking techniques. In addition, a diffusion graph is used to improve the effectiveness of smoothing out structural information and creating an intermediate bridge between the two distributions. Our framework introduces a new teaching network that can effectively integrate and utilize complementary structural information from different graphs, as well as a random sampling mixing module that offers new perspectives on GCL. STGCN combines supervised contrastive learning with node consistency regularization. It contrasts a diffusion graph with two augmentation graph views to learn node representations that are simultaneously discriminative to structural perturbations. The experiments on five real-world benchmarks demonstrate that STGCN performs better than current state-of-the-art methods. The results validate the effectiveness of our method in utilizing limited labels to guide the contrastive learning process, resulting in good node classification performance. Despite its advantages, it still relies on manually designed augmentation methods, and this architecture increases computational costs. Future work can explore adaptive augmentation strategies and more downstream tasks, including link prediction or graph classification tasks.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This paper is supported by the Macao Science and Technology Development Fund (0036/2023/RIA1, 0037/2024/RIA1) and Jiangsu Kaite Auto Parts Co., Ltd. (MUST Code: 1330).

Conflict of interest

The authors declare there is no conflict of interest.

Code Availability

The implementation of this paper is available at https://github.com/qwertyuiopqwe123/STGCN.

References

1. Y. Wang, X. Li, Y. Du, D. Huang, X. Chen, Z. Dong, et al., Graph convolutional network reconstruction with high-order node information for community detection, *Eng. Appl. Artif. Intell.*, **163** (2026), 112957. https://doi.org/10.1016/j.engappai.2025.112957
2. R. Cavoretto, C. Comoglio, A. De Rossi, Community detection methods for GBF-PUM signal approximation on graphs, *Appl. Math. Comput.*, **510** (2026), 129702. https://doi.org/10.1016/j.amc.2025.129702
3. M. Teng, C. Gao, X. Li, Z. Wang, K. Fan, V. Nekorkin, Multi-scale graph contrastive learning for community detection in dynamic graphs, *Inf. Process. Manag.*, **63** (2026), 104410. https://doi.org/10.1016/j.ipm.2025.104410
4. M. A. S. Sejan, M. H. Rahman, M. A. Aziz, R. Tabassum, I. Hameed, N. Nasser, et al., Graph neural network enhanced internet of things node classification with different node connections, *J. Network Comput. Appl.*, **244** (2025), 104363. https://doi.org/10.1016/j.jnca.2025.104363
5. J. Wang, Q. Guan, L. Deng, M. Zhou, Z. Gong, GraphST: Class-imbalanced node classification with semantic relation transfer, *Pattern Recognit.*, **172** (2026), 112626. https://doi.org/10.1016/j.patcog.2025.112626
6. R. Song, P. Cao, G. Wen, L. Li, W. Liang, W. Li, et al., CGMAE: Self-supervised masked auto-encoder with cross-graph node alignment for node classification, *Eng. Appl. Artif. Intell.*, **163** (2026), 112910. https://doi.org/10.1016/j.engappai.2025.112910
7. Y. Su, Y. Zhao, S. Erfani, J. Gan, R. Zhang, Detecting arbitrary order beneficial feature interactions for recommender systems, in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, (2022), 1676–1686. https://doi.org/10.1145/3534678.3539238
8. Y. Hou, S. Mu, W. X. Zhao, Y. Li, B. Ding, J. Wen, Towards universal sequence representation learning for recommender systems, in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, (2022), 585–593. https://doi.org/10.1145/3534678.3539381

9. Y. Ma, Y. He, A. Zhang, X. Wang, T. Chua, CrossCBR: Cross-view contrastive learning for bundle recommendation, in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, (2022), 1233–1241. https://doi.org/10.1145/3534678.3539229
10. S. Thakoor, C. Tallec, M. G. Azar, M. Azabou, E. L. Dyer, R. Munos, et al., Large-scale representation learning on graphs via bootstrapping, preprint, arXiv:2102.06514.
11. J. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, et al., Bootstrap your own latent-a new approach to self-supervised learning, preprint, arXiv:2006.07733.
12. J. Lee, Y. Oh, Y. In, N. Lee, D. Hyun, C. Park, GraFN: Semi-supervised node classification on graph with few labels via non-parametric distribution assignment, in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, (2022), 2243–2248. https://doi.org/10.1145/3477495.3531838
13. M. Ghayekhloo, A. Nickabadi, Supervised contrastive learning for graph representation enhancement, *Neurocomputing*, **588** (2024), 127710. https://doi.org/10.1016/j.neucom.2024.127710
14. T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, preprint, arXiv:1609.02907.
15. P. Veličković, A. Casanova, P. Liò, G. Cucurull, A. Romero, Y. Bengio, Graph attention networks, in *ICLR 2018 The Sixth International Conference on Learning Representations*, (2018), 1–12. https://doi.org/10.17863/CAM.48429
16. J. Liu, G. P. Ong, X. Chen, GraphSAGE-based traffic speed forecasting for segment network with sparse data, *IEEE Trans. Intell. Transp. Syst.*, **23** (2022), 1755–1766. https://doi.org/10.1109/TITS.2020.3026025
17. S. Wan, S. Pan, J. Yang, C. Gong, Contrastive and generative graph convolutional networks for graph-based semi-supervised learning, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **35** (2021), 10049–10057. https://doi.org/10.1609/aaai.v35i11.17206
18. Q. Li, Z. Han, X. Wu, Deeper insights into graph convolutional networks for semi-supervised learning, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **32** (2018), 3538–3545. https://doi.org/10.1609/aaai.v32i1.11604
19. X. Wu, Z. Li, A. M. So, J. Wright, S. Chang, Learning with partially absorbing random walks, in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, **2** (2012), 3077–3085.
20. K. Sun, Z. Lin, Z. Zhu, Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes, in *Proceedings of the AAAI conference on artificial intelligence*, **34** (2020), 5892–5899. https://doi.org/10.1609/aaai.v34i04.6048
21. M. Ghayekhloo, A. Nickabadi, CLP-GCN: Confidence and label propagation applied to graph convolutional networks, *Appl. Soft Comput.*, **132** (2023), 109850. https://doi.org/10.1016/j.asoc.2022.109850
22. F. Hu, Y. Zhu, S. Wu, L. Wang, T. Tan, Hierarchical graph convolutional networks for semi-supervised node classification, in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, (2019), 4532–4539.

23. Y. Jiang, C. Huang, L. Huang, Adaptive graph contrastive learning for recommendation, in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, (2023), 4252–4261. https://doi.org/10.1145/3580305.3599768
24. S. Suresh, P. Li, C. Hao, J. Neville, Adversarial graph augmentation to improve graph contrastive learning, in *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, (2021), 1–14.
25. P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, R. D. Hjelm, Deep graph infomax, preprint, arXiv:1809.10341.
26. Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, Y. Shen, Graph contrastive learning with augmentations, in *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, (2020), 1–12.
27. K. Hassani, A. H. Khasahmadi, Contrastive multi-view representation learning on graphs, in *Proceedings of the 37th International Conference on Machine Learning*, (2020), 1–11.
28. J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, et al., GCC: Graph contrastive coding for graph neural network pre-training, in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, (2020), 1150–1160. https://doi.org/10.1145/3394486.3403168
29. Q. Li, W. Li, X. Zheng, J. Zhou, W. Zhong, X. Chen, et al., GRE2-MDC: Graph representation embedding enhanced via multidimensional contrastive learning, *IEEE Access*, **13** (2025), 61312–61321. https://doi.org/10.1109/ACCESS.2025.3553862
30. Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, L. Wang, Deep graph contrastive representation learning, preprint, arXiv:2006.04131.
31. L. Chen, G. Zhu, Self-supervised contrastive learning for itinerary recommendation, *Expert Syst. Appl.*, **268** (2025), 126246. https://doi.org/10.1016/j.eswa.2024.126246
32. L. Chen, G. Zhu, W. Liang, J. Cao, Y. Chen, Keywords-enhanced contrastive learning model for travel recommendation, *Inf. Process. Manage.*, **61** (2024), 103874. https://doi.org/10.1016/j.ipm.2024.103874
33. Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, L. Wang, Graph contrastive learning with adaptive augmentation, in *Proceedings of the Web Conference 2021*, (2021), 2069–2080. https://doi.org/10.1145/3442381.3449802
34. T. Wang, K. Zhang, Y. Zhang, W. Luo, B. Stenger, T. Lu, et al., LLDiffusion: Learning degradation representations in diffusion models for low-light image enhancement, *Pattern Recognit.*, **166** (2025), 111628. https://doi.org/10.1016/j.patcog.2025.111628
35. Y. Wen, T. Gao, Z. Li, J. Zhang, K. Zhang, T. Chen, All-in-one weather-degraded image restoration via adaptive degradation-aware self-prompting model, *IEEE Trans. Multimedia*, **27** (2025), 3343–3355. https://doi.org/10.1109/TMM.2025.3535316
36. K. Zhang, W. Ren, W. Luo, W. Lai, B. Stenger, M. Yang, et al., Deep image deblurring: A survey, *Int. J. Comput. Vision*, **130** (2022), 2103–2130. https://doi.org/10.1007/s11263-022-01633-5
37. J. Gasteiger, A. Bojchevski, S. Günnemann, Predict then propagate: Graph neural networks meet personalized pagerank, preprint, arXiv:1810.05997.

38. K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks? preprint, arXiv:1810.00826.

39. X. Yao, H. Zhu, M. Gu, Brain-inspired GCN:modularity-based siamese simple graph convolutional networks, *Inf. Sci.*, **657** (2024), 119971. https://doi.org/10.1016/j.ins.2023.119971

40. T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in *Proceedings of the 37th International Conference on Machine Learning*, (2020), 1–11.

41. F. Wu, T. Zhang, A. H. de Souza, C. Fifty, T. Yu, K. Q. Weinberger, Simplifying graph convolutional networks, in *Proceedings of the 36th International Conference on Machine Learning*, (2019), 1–11.

42. W. Feng, J. Zhang, Y. Dong, Y. Han, H. Luan, Q. Xu, et al., Graph random neural networks for semi-supervised learning on graphs, in *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, (2020), 1–12.

43. S. Wan, Y. Zhan, L. Liu, B. Yu, S. Pan, C. Gong, Contrastive graph poisson networks: Semi-supervised learning with extremely limited labels, in *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, (2021), 1–12.

44. Z. Wu, P. Zhou, G. Wen, X. Zhu, A pseudo-labeling approach based on knowledge distillation for graph few-shot learning, *Inf. Process. Manage.*, **62** (2025), 104268, https://doi.org/10.1016/j.ipm.2025.104268

45. P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, T. Eliassi-Rad, Collective classification in network data, *AI Mag.*, **29** (2008), 93–106. https://doi.org/10.1609/aimag.v29i3.2157

46. G. Namata, B. London, L. Getoor, B. Huang, Query-driven active surveying for collective classification, in *Proceedings of the Workshop on Mining and Learn ing with Graphs (MLG-2012)*, (2012), 1–8.

47. O. Shchur, M. Mumme, A. Bojchevski, S. Günnemann, Pitfalls of graph neural network evaluation, preprint, arXiv:1811.05868.

48. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, preprint, arXiv:1412.6980.

49. Z. Peng, W. Huang, M. Luo, Q. Zheng, Y. Rong, T. Xu, et al., Graph representation learning via graphical mutual information maximization, in *Proceedings of The Web Conference 2020*, (2020), 259–270. https://doi.org/10.1145/3366423.3380112

50. Y. Jiao, Y. Xiong, J. Zhang, Y. Zhang, T. Zhang, Y. Zhu, Sub-graph contrast for scalable self-supervised graph representation learning, in *2020 IEEE International Conference on Data Mining (ICDM)*, (2020), 222–231. https://doi.org/10.1109/ICDM50108.2020.00031

51. H. Wang, J. Zhang, Q. Zhu, W. Huang, Augmentation-free graph contrastive learning with performance guarantee, preprint, arXiv:2204.04874.

52. X. Shen, D. Sun, S. Pan, X. Zhou, L. T. Yang, Neighbor contrastive learning on learnable graph augmentation, in *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, **37** (2023), 9782–9791. https://doi.org/10.1609/aaai.v37i8.26168

53. M. Peng, X. Juan, Z. Li, Label-guided graph contrastive learning for semi-supervised node classification, *Expert Syst. Appl.*, **239** (2024), 122385. https://doi.org/10.1016/j.eswa.2023.122385

AIMS Press