Graph Contrastive Learning with Joint Spectral Augmentation of Attribute and Topology

Liang Yang[1], **Zhenna Li**[1], **Jiaming Zhuo**[1*], **Jing Liu**[1], **Ziyi Ma**[1],
Chuan Wang[2], **Zhen Wang**[3], **Xiaochun Cao**[4]
[1]Hebei Province Key Laboratory of Big Data Calculation,
School of Artificial Intelligence, Hebei University of Technology, Tianjin, China,
[2]School of Computer Science and Technology, Beijing JiaoTong University, Beijing, China,
[3]School of Artificial Intelligence, OPtics and ElectroNics (iOPEN),
School of Cybersecurity, Northwestern Polytechnical University, Xi'an, China,
[4]School of Cyber Science and Technology, Shenzhen Campus of Sun Yat-sen University, Shenzhen, China
yangliang@vip.qq.com, 202222802031@stu.hebut.edu.cn, jiaming.zhuo@outlook.com, liujing@scse.hebut.edu.cn
zyma@hebut.edu.cn, wangchuan@iie.ac.cn, w-zhen@nwpu.edu.cn, caoxiaochun@mail.sysu.edu.cn

Abstract

As an essential technique for Graph Contrastive Learning (GCL), Graph Augmentation (GA) improves the generalization capability of the GCLs by introducing different forms of the same graph. To ensure information integrity, existing GA strategies have been designed to simultaneously process the two types of information available in graphs: node attributes and graph topology. Nonetheless, these strategies tend to augment the two types of graph information separately, ignoring their correlation, resulting in limited representation ability. To overcome this drawback, this paper proposes a novel GCL framework with a Joint spectrAl augMentation, named GCL-JAM. Motivated the equivalence between the graph learning objective on an attribute graph and the spectral clustering objective on the attribute-interpolated graph, the node attributes are first abstracted as another type of node to harmonize the node attributes and graph topology. The newly constructed graph is then utilized to perform spectral augmentation to capture the correlation during augmentation. Theoretically, the proposed joint spectral augmentation is proved to perturb more inter-class edges and noise attributes compared to separate augmentation methods. Extensive experiments on homophily and heterophily graphs validate the effectiveness and universality of GCL-JAM.

Introduction

Graphs are ubiquitous data structures in the real world and have a wide range of applications in fields such as social networks (Kipf and Welling 2017) and citation networks (Sen et al. 2008). To effectively extract useful information in the graphs, Graph Neural Networks (GNNs), including GCN (Kipf and Welling 2017) and GAT (Veličković et al. 2018), have been proposed and trained in semi-supervised learning scenarios. The reliance on label information limits the usability of these models to many tasks where collecting labels is expensive. To solve this drawback, Graph Contrastive Learning (GCL) (Hassani and Khasahmadi 2020; Zhuo et al. 2024a,c), a representative self-supervised learning architecture, has been developed. Vanilla GCLs includes three key components: graph augmentation that increases the number of training samples, graph encoder that represents these nodes, and contrastive loss that guides the update direction.

According to how the information is processed, existing graph augmentation strategies in GCLs are mainly classified into two categories: random augmentation and prior-based augmentation. The former performs random perturbations on both graph topology (e.g., adding and removing edges) (Zhu et al. 2020; Thakoor et al. 2021) and node attributes (e.g., masking attributes and adding noise) (Mo et al. 2022; Zhu et al. 2020). Although simple and somewhat successful, this strategy inevitably incurs the corruption of semantic information, resulting in sub-optimal model performance (Chang et al. 2021). The latter, hence, focuses on the integrity of the semantic information by incorporating a priori knowledge to guide the augmentation process (Zhuo et al. 2024b). The representative works include: graph diffusion-based (Hassani and Khasahmadi 2020), graph sample-based (You et al. 2020; Qiu et al. 2020) and spectrum perturbation-based topology augmentation (Lin, Chen, and Wang 2023). Unfortunately, most graph augmentations neglect to consider the correlation of the two types of available information in GCLs: node attributes and graph topology. Intuitively, there are correlations between them. The attributes of a node can determine its connectivity pattern in the graph, thereby affecting the entire graph topology (Li, Huang, and Zitnik 2021; Sen et al. 2008). In turn, the topology of the graph shapes or changes node attributes by impacting information flow and interactions between nodes (Kipf and Welling 2017). Therefore, it is essential to model such correlations in graph augmentation improve the model performance.

To address the aforementioned issue, this paper seeks to propose a joint augmentation of node attributes and graph topology for GCL. The major challenges are, on one hand, to identify a unified view to align the node attributes and graph topology, on the other hand, to devise an effective augmentation strategy based on this view. As a solution to

[*]corresponding author

these hurdles, a novel GCL framework, named Joint spectrAl augMentation (GCL-JAM), is introduced. The idea is to transform the original graph into an attribute-interpolated graph (as defined in Definition 1) and subsequently perform spectral augmentation on this newly constructed graph.

The creation of attribute-interpolated graphs offers a manner to harmonize node attributes with graph topology. Firstly, the attribute-interpolated graph derived from the input graph is a heterogeneous graph with graph topology but without node attributes. In such a graph, node attributes are regarded as another type of nodes (i.e., attribute node) besides the original nodes; the node-to-attribute inclusion relationships are represented by another type of edges beside the original edges (as illustrated in Figure 1). Theoretically, the objective function of spectral clustering (Von Luxburg 2007) on the attribute-interpolated graph is equivalent to the objective function of graph learning on the input graph, which has been proven to unify many GNNs (graph encoders) (Zhu et al. 2021a; Yang et al. 2021), as demonstrated in Theorem 1. This spectral interpretation not only underscores the capacity of the attribute-interpolated graph to uniformly represent both attributes and topology but also motivates the joint augmentation of these two types of graph information via spectral augmentation. Thus, the second phase of GCL-JAM involves selectively perturbing the edges that cause the largest change in the Laplacian matrix's eigenvalues (i.e., graph spectrum) for the attribute-interpolated graph, that is, inter-cluster edges.

The introduction of attribute nodes notably enhances the topological connectivity between nodes of the same class in the graph (especially for heterophily graphs), as depicted in Figure 1. As a result, the proposed GCL-JAM is adept at preserving edges that link nodes from the cluster while strategically perturbing those that connect nodes across different clusters within the attribute-interpolated graphs. From the input graph perspective, the proposed joint spectral augmentation achieves a more effective perturbation of both node attributes and graph topology over the existing augmentation, capitalizing on the implicit correlation between them, as detailed in Theorem 2.

In summary, this paper has the following contributions:

- We justify the equivalence relation between GNNs and spectral clustering on an attribute-interpolated graph containing attribute nodes and original nodes, providing the theoretical basis for unifying attributes and topology.
- We propose a novel GCL framework GCL-JAM, which can jointly augment attributes and topology on the attribute-interpolated graph. And we further provide theoretical analysis to justify the effectiveness of GCL-JAM.
- We conduct extensive experiments on twelve well-known benchmark datasets with various homophily degrees to demonstrate the superior performance of GCL-JAM.

Related Work

Graph Contrastive Learning (GCL) aims to build effective representations by comparing sample pairs (Zhuo et al. 2024a). To increase the diversity of samples and thus improve the robustness of the model, GCL generally utilizes various augmentation strategies to perturb topology or attributes. Specifically, GRACE (Zhu et al. 2020) generates augmented graphs by randomly dropping edges or nodes and maximizes the consistency of the node representations from the two views; GCA (Zhu et al. 2021b) uses an adaptive augmentation method to perturb the unimportant information; MVGRL (Hassani and Khasahmadi 2020) employs graph diffusion to generate graph views and learns both node-level and graph-level representations; GREET (Liu et al. 2023) introduces a discriminator to assess the homophily of edges and employs random augmentation.

All of the above augmentation methods are designed from the spatial domain. And there are also some models designed from the spectral domain. SPAN-GCL (Lin, Chen, and Wang 2023) selects the edges which have the greatest change on the graph spectrum for perturbation; GASSER (Yang et al. 2023) proposes tailored perturbation on the specific frequencies of graph structure; SpCo (Liu et al. 2022) proposes to preserve the low-frequency components and perturb the high-frequency components.

Preliminaries

Notations

Let $G(\mathbf{A}, \mathbf{X})$ denotes an attribute graph, where $\mathbf{X} \in \mathbb{R}^{N\times F}$represents the node attributes with the number of nodes N and attribute dimension F. $\mathbf{A}$ describes the adjacency matrix, typically reflecting the graph topology. If there is an edge between node i and node j, $\mathbf{A}_{ij} = 1$. $\mathbf{D}$ denotes the dignal degree matrix with $\mathbf{D}_{ii} = \sum_{j=1}^{N} \mathbf{A}_{ij}$. $\mathbf{L} = \mathbf{D} - \mathbf{A}$ terms the Laplacian matrix of the graph. And its normalized Laplacian matrix is further defined as $\tilde{\mathbf{L}} = \mathbf{I} - \hat{\tilde{\mathbf{A}}} = \mathbf{I} - \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$, where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\tilde{\mathbf{D}} = \mathbf{D} + \mathbf{I}$ denote the adjacency matrix and degree matrix with added self-loop, respectively. Eigenvalue decomposition can be performed: $\tilde{\mathbf{L}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\top}$. From the spectral perspective, the eigenvalues $\mathbf{\Lambda}$ correspond to the notions of frequency and the eigenvectors $\mathbf{U}$ as the spectral bases (Shuman et al. 2013).

Graph Learning Objective

Graph representation learning aims to train an encoder which can produce node representations or graph representations for downstream tasks. The node representation learning generally follows two objectives: the node representations are similar with their original attributes; connected nodes have similar representations (Zhu et al. 2021a; Yang et al. 2021). One of the most commonly utilized objective functions can be formulated as:

$$\mathcal{O} = \min_{\mathbf{H}} \{\|\mathbf{H} - \mathbf{X}\|_F^2 + \lambda tr(\mathbf{H}^{\top}\mathbf{L}\mathbf{H})\}, \quad (1)$$

where $\mathbf{H}$ denotes node representations. Graph Convolutional Network (GCN) as a method for node representation learning can be derived by optimizing Eq. 1. Specifically, with $\lambda = 1$ and $\tilde{\mathbf{L}}$, the convolution operation in GCN can be induced by setting derivative of Eq. 1 with respect to $\mathbf{H}$ to zero:

$$\mathbf{H} = (\mathbf{I} + \tilde{\mathbf{L}})^{-1}\mathbf{X} \approx (\mathbf{I} - \tilde{\mathbf{L}})\mathbf{X} = \hat{\tilde{\mathbf{A}}}\mathbf{X}. \quad (2)$$

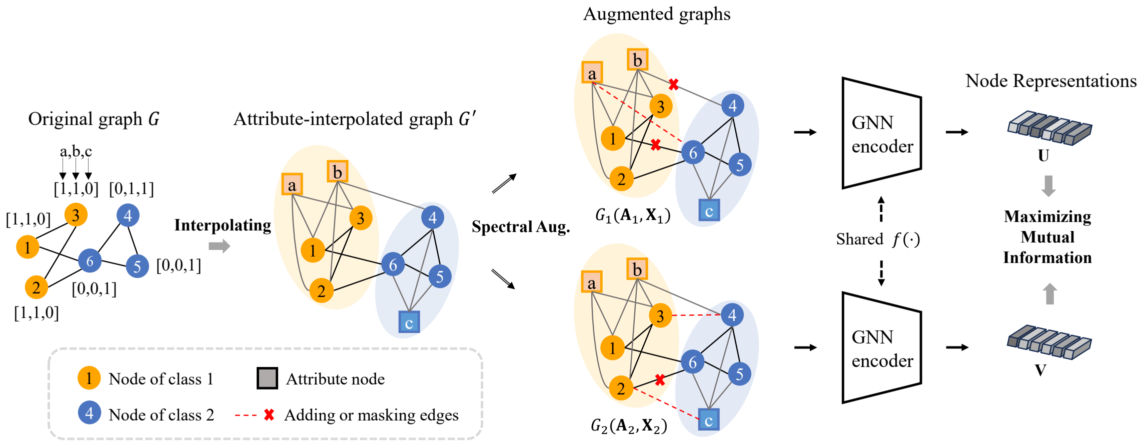


Figure 1: The overview of the proposed GCL-JAM. Firstly, the original graph G is transformed into an attribute-interpolated graph G' (defined in Definition 1) by treating attributes as nodes. Secondly, the attributes and topology of the original graph are jointly augmented via spectral augmentation on the attribute-interpolated graph. Finally, the augmented graphs are utilized in the subsequent modules, i.e., encoder and contrastive loss.

Besides, several graph neural networks (including SGC (Wu et al. 2019), JKNet (Xu et al. 2018) and APPNP (Gasteiger, Bojchevski, and Günnemann 2019)) can be derived from corresponding variants of this framework.

Graph Contrastive Learning

Inspired by the design of contrastive learning in CV (Chen et al. 2020) and NLP (Gao, Yao, and Chen 2021), Graph Contrastive Learning (GCL) typically includes three components: (1) *Graph Augmentation.* Graph augmentation is introduced to perturb the original graph, thus effectively expanding the diversity of samples. Given the input graph $G(\mathbf{A}, \mathbf{X})$, two augmented views can be generated through attribute perturbation or edge perturbation, formulated as $G_i(\mathbf{A}_i, \mathbf{X}_i) = t_i(G(\mathbf{A}, \mathbf{X}))$. (2) *Graph Encoder.* Graph neural networks are generally employed to encode the augmented views. By mapping the augmented views into a low-dimensional vector space, the latent graph information can be captured. (3) *Contrastive Loss.* Various contrastive losses are designed to guide the training of graph encoder, such as local-local (Zhu et al. 2020), local-global (Hassani and Khasahmadi 2020) and global-global contrastive loss (You et al. 2020). Take the local-local loss adopted by GCL-JAM as an example, the same nodes from two views are positive pairs, and all other nodes are negative samples. Discriminative node representations can be learned by pulling positive samples closer in the embedding space while pushing negative samples apart.

Methodology

This section proposes a novel graph contrastive learning model with joint spectral augmentation (GCL-JAM), as illustrated in Figure 1. It begins by providing an attribute-interpolated graph. Next, building upon the attribute-interpolated graph, a spectral augmentation is employed for graph perturbation. Finally, the validity of the proposed GCL-JAM is proved theoretically.

Attribute as Node

To align node attributes and graph topology, an attribute-interpolated graph is first introduced by abstracting attributes as nodes. The definition of the attribute-interpolated graph is as follows.

Definition 1. *Given the attribute graph $G(\mathbf{A}, \mathbf{X})$, its node attributes can be regarded as a special type of nodes, i.e., attribute nodes. The attribute values of G determine the existence of edges between the attribute node and the original node. Then we have an attribute-interpolated graph G' with the adjacency matrix:*

$$\mathbf{A}' = \begin{bmatrix} \mathbf{A} & \mathbf{X} \\ \mathbf{X}^\top & \mathbf{0} \end{bmatrix}, \tag{3}$$

where $\mathbf{0}$ is an all-zero matrix denoting that edges between attribute nodes are not considered.

The attribute-interpolated graph is a heterogeneous graph with two types of edges, representing: 1) the connection relationship between the original nodes; 2) the inclusion relationship between the attribute node and the original node.

To illustrate the significance of constructing the attribute-interpolated graph, the relationship between G' and Graph Neural Network (GNN) is derived from the Theorem 1. The unnormalized Laplacian matrix of G' is first given as:

$$\mathbf{L}' = \begin{bmatrix} \mathbf{D} + \mathbf{E} - \mathbf{A} & -\mathbf{X} \\ -\mathbf{X}^\top & \mathbf{F} \end{bmatrix}, \tag{4}$$

where $\mathbf{F}$ and $\mathbf{E}$ are diagonal matrices, denoting the degree of the attribute nodes and the increased degree of the original nodes, respectively.

Theorem 1. *The optimization objective of spectral clustering on the attribute-interpolated graph G' is equal to the optimization objective of GNN:*

$$\min_{\mathbf{H}}\{tr(\mathbf{H'}^{\top}\mathbf{L'}\mathbf{H'})\} = \min_{\mathbf{H}}\{\|\mathbf{H}-\mathbf{X}\|_F^2 + tr(\mathbf{H}^{\top}\mathbf{L}\mathbf{H})\},$$

where $\mathbf{H}$ ($\mathbf{H'}$) denotes the node representations.

Theorem 1 emphasizes the significance of the attribute-interpolated graph in unifying the attributes and topology of the original graph. In the attribute-interpolated graph, nodes within the same class that share similar attributes are more closely connected due to edges that establish an inclusion relation. This leads to distinguishable cluster structures of intra-class nodes. Besides, Theorem 1 stimulates the realization of joint augmentation from a spectral perspective, effectively preserving the cluster properties.

Joint Spectral Augmentation

After the attribute-interpolated graph is obtained, the spectral augmentation on its topology is utilized in GCL-JAM. Specifically, the edges resulting in greatest spectral effect will be selected for perturbation (Lin, Chen, and Wang 2023), which is measured by the difference of the eigenvalues of normalized Laplacian matrix.

An edge perturbation matrix $\mathbf{E} \in \{0,1\}^{(N+F)\times(N+F)}$ can be obtained from the probability matrix $\mathbf{P}$ by Bernoulli sampling, $\mathbf{E}_{ij} \sim \mathcal{B}(\mathbf{P}_{ij})$. If $\mathbf{E}_{ij} = 1$, the edge between node i to node j will be flipped. To make the operations of adding and removing edges effective, a matrix $\mathbf{C}$ is introduced:

$$\mathbf{C}_{ij} = \begin{cases} 1, & \text{if } \mathbf{A'}_{ij} = 0, \\ -1, & \text{Otherwise} \end{cases}. \tag{5}$$

The perturbed $\mathbf{A'}$ can be denoted as $\mathbf{A'}{+}\mathbf{E}{\odot}\mathbf{C}$, where $\odot$ denotes Hadamard product. Next, the objective function with the largest eigenvalue change is defined to guide the spectral augmentation process. Concretely, with the adjacency matrix $\mathbf{A'}$, its normalized Laplacian matrix $\tilde{\mathbf{L}}'$ can be obtained. And the Laplacian eigenvalues can be computed by matrix decomposition, $\mathbf{\Lambda} = \{\lambda_i\}_{i=1}^{N+F}$ with $\lambda_1 \leq \lambda_2 \ldots \leq \lambda_{N+F}$. To make the objective function derivable, the model is trained with $\mathbf{P}$ instead of $\mathbf{E}$. The augmentation objective function is given as follows:

$$\max_{\mathbf{P}}\|\mathbf{\Lambda}_s - \mathbf{\Lambda}_t\|_2^2,\ s.t.\ \|\mathbf{P}\|_1 \leq \epsilon, \tag{6}$$

where $\mathbf{\Lambda}_s$ corresponds to the original attribute-interpolated graph G' and $\mathbf{\Lambda}_t$ corresponds to the perturbed attribute-interpolated graph. ϵ is used to control the perturbation strength. There are two optimal cases for Eq.6: the maximum positive value and the minimum negative value, which generate two probability matrices $\mathbf{P}_1$ and $\mathbf{P}_2$. Corresponding augmented attribute-interpolated graphs G'_1 and G'_2 can be obtained with the perturbed adjacency matrices:

$$\mathbf{A}'_1 = \begin{bmatrix} \mathbf{A}_1 & \mathbf{X}_1 \\ \mathbf{X}_1^{\top} & \mathbf{0} \end{bmatrix}, \quad \mathbf{A}'_2 = \begin{bmatrix} \mathbf{A}_2 & \mathbf{X}_2 \\ \mathbf{X}_2^{\top} & \mathbf{0} \end{bmatrix}. \tag{7}$$

Thus, the augmented attribute graphs are denoted as $G_1 = (\mathbf{A}_1, \mathbf{X}_1)$ and $G_2 = (\mathbf{A}_2, \mathbf{X}_2)$. Finally, the node representations of G_1 and G_2 are learned by a graph encoder. Since GCL-JAM focuses on node-level tasks, a widely-used local-local contrastive objective is adopted, which is consistent with GRACE (Zhu et al. 2020).

Theoretical Analysis

This section aims to provide the theoretical analysis of GCL-JAM. Firstly, the changes of topology after treating attributes as nodes are analyzed using modularity measure (Newman and Girvan 2004). Next, combined with the fact that the edges causing the largest difference in eigenvalues are inter-cluster edges, it can be concluded that GCL-JAM can perturb more inter-class edges and noise attributes.

Definition 2. *Consider a graph G with k classes of nodes. Let us define a $k \times k$ symmetric matrix $\boldsymbol{e}$ whose element e_{ij} is the fraction of all edges that link nodes in class i to nodes in class j. The row (or column) sums $a_i = \sum_j e_{ij}$. Then the modularity measure can be defined by:*

$$\mathcal{Q} = \sum_i (e_{ii} - a_i^2). \tag{8}$$

Theorem 2. *Consider a heterophilic graph with the edge homophily $h < 0.5$. There are N nodes with degree d and attribute dimension F. Treating attributes as nodes can make intra-class nodes more tightly connected:*

$$\mathcal{Q}' = \mathcal{Q} + \frac{NF(\overline{p} - h)}{Nd + NF}, \tag{9}$$

where $\mathcal{Q}'$ and $\mathcal{Q}$ correspond to the attribute-interpolated graph and the original graph, respectively. $\overline{p} > 0.5$ denotes the average probability that the attribute belongs to the corresponding class.

For homophilic graphs, the modularity measure will increase when $\overline{p} > h$. The interplay of spectral change used in GCL-JAM and spatial change has been derived in (Bojchevski and Günnemann 2019) as follows:

$$\triangle\lambda_m = \sum_{i=1}^{N+F}\sum_{j=1}^{N+F} \triangle w_{ij}((u_{mi} - u_{mj})^2 - \lambda_m(u_{mi}^2 + u_{mj}^2)), \tag{10}$$

where $\triangle w_{ij}$ denotes an edge flip between node i and node j. u_m is the m-th eigenvector corresponding to the eigenvalue λ_m. When the distance between u_{mi} and u_{mj} is large, these two nodes should belong to different clusters (Ng, Jordan, and Weiss 2001). Combined with Theorem 2, intra-class nodes in the attribute-interpolated graph are more likely to form clusters. Thus, it can be demonstrated that GCL-JAM will perturb inter-class edges and noise attributes, which benefits the retention of semantic information.

Time Complexity Analysis

The complexity of augmentation in GCL-JAM is $O(T(N + F)^3)$, where T denotes the time of iterations. Specifically, eigenvalue decomposition is required to compute the augmentation probability matrix, which incurs a complexity of $O(T(N+F)^3)$. For large-scale graphs, the time complexity can be reduced to $O(TK(N+F)^2)$ by appealing to selective eigen-decomposition on K lowest- and highest-eigenvalues via the Lanczos Algorithm (Parlett and Scott 1979).

Methods	Cora	CiteSeer	PubMed	Wiki-CS	Computers	Photo
GCN	82.17±0.59	71.46±0.97	84.16±0.23	76.89±0.37	86.34±0.48	92.35±0.25
GAT	83.46±0.78	72.59±0.82	84.95±0.48	77.42±0.19	87.06±0.35	92.64±0.42
DeepWalk	76.43±0.57	59.73±0.25	79.36±0.57	74.35±0.06	85.68±0.06	89.44±0.11
Node2Vec	79.13±0.88	60.64±0.59	80.19±0.84	71.79±0.05	84.39±0.08	89.67±0.12
DGI	82.46±0.30	71.60±0.21	85.61±0.14	75.73±0.13	84.09±0.39	91.49±0.25
GMI	82.36±0.97	71.64±0.49	84.29±0.90	75.06±0.13	81.76±0.52	90.72±0.33
MVGRL	83.01±0.42	72.76±0.53	85.13±0.38	77.97±0.18	87.09±0.27	92.01±0.13
GRACE	83.44±0.39	71.52±0.36	**86.02±0.34**	79.16±0.36	87.21±0.44	92.65±0.32
GCA	82.79±0.53	71.19±0.22	$\underline{85.64\pm0.75}$	79.35±0.12	87.84±0.27	92.78±0.17
BGRL	82.67±0.78	71.68±0.52	84.13±0.17	78.74±0.22	88.92±0.33	93.24±0.29
GraphMAE	84.01±0.37	72.75±0.41	84.55±0.23	78.82±0.24	89.68±0.35	93.37±0.20
SpCo	84.08±0.76	72.78±0.45	84.98±0.31	79.43±0.36	89.15±0.57	$\underline{93.46\pm0.18}$
GCL-SPAN	$\underline{85.01\pm0.89}$	$\underline{72.79\pm0.61}$	85.23±0.20	$\underline{81.36\pm0.13}$	$\underline{90.07\pm0.39}$	93.31±0.24
GCL-JAM	**85.54±0.53**	**73.29±0.39**	85.46±0.46	**82.09±0.42**	**90.69±0.28**	**94.35±0.21**

Table 1: Node classification performance on homophilic graphs. The metric is mean accuracy (%) and standard deviation. The best and the second best results are highlighted with bold and underline, respectively.

Experiments

In this section, the effectiveness and universality of the proposed GCL-JAM are evaluated by comparing it with several graph learning methods. The experiments include node classification tasks, effectiveness study, and ablation study.

Experiment Setup

Datasets. To conduct an extensive evaluation, twelve well-known benchmark datasets with various homophily degrees are used in experiments. These datasets can be broadly divided into two categories: six homophilic graphs and six heterophilic graphs. **Cora**, **Citeseer**, and **PubMed** (Sen et al. 2008) are citation networks. **Wiki-CS** (Mernyei and Cangea 2020) is a hyperlink network. **Computers** and **Photo** (Shchur et al. 2018) are co-purchase networks. **Chameleon** and **Squirrel** (Pei et al. 2020) are page-page networks. **Actor** (Pei et al. 2020) is an actor co-occurrence network. **Cornell**, **Texas** and **Wisconsin** (Pei et al. 2020) are webpage datasets. For homophilic graphs, 1:1:8 train/validation/test random splits are employed. For heterophilic graphs, the proportion of nodes utilized for training, validation, and testing is 48%, 32%, and 20%.

Baselines. To verify the effectiveness and superiority of our proposed model, we compare GCL-JAM with several graph learning methods. These methods fall into three categories: (1) semi-supervised GNN models for node classification task, including vanilla GCN (Kipf and Welling 2017) and GAT (Veličković et al. 2018); (2) unsupervised graph learning methods, including DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) and Node2Vec (Grover and Leskovec 2016); (3) self-supervised graph learning methods, including GRACE (Zhu et al. 2020), BGRL (Thakoor et al. 2022), MVGRL (Hassani and Khasahmadi 2020), GCA (Zhu et al. 2021b), GMI (Peng et al. 2020), DGI (Veličković et al. 2019), GraphMAE (Hou et al. 2022), SpCo (Liu et al. 2022) and GCL-SPAN (Lin, Chen, and Wang 2023).

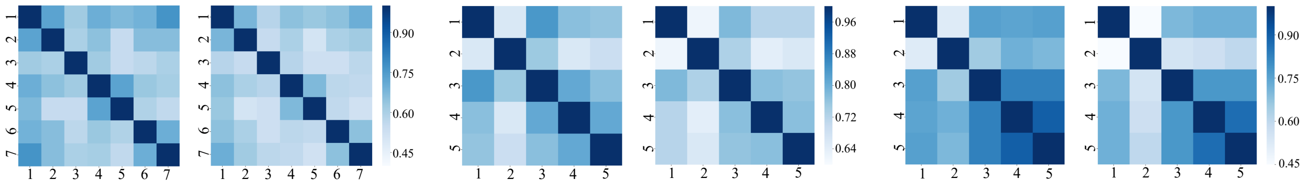
Experimental Details. For reproducibility, the detailed settings of the experiments are described below. The experiments are performed on Nvidia GeForce RTX 3090 (24GB) GPU cards. We use a 2-layer GCN with a hidden size of 512 as the graph encoder. For datasets with large initial attribute dimensions, such as Chameleon and Squirrel, the hidden size is set to 1024. The training epoch is 500 with full batch training, and the augmentation training epoch is 50. For hyperparameter settings, we tune the learning rate of learning the probability matrix $\mathbf{P}$ from {0.1, 0.05, 0.01, 0.005, 0.001}. Besides, we tune the perturbation rate from {0.1, 0.2, ... , 0.9}. Based on the representations learned by the encoder, we train a Logistic classifier to perform downstream tasks. In all the experiments, we use the Adam optimizer. The learning rate is tuned from {0.1, 0.05, 0.01, 0.005, 0.001} and weight decay is tuned from {0.0, 0.001, 0.005, 0.01, 0.1}. We run all models ten times on each dataset, and the mean and standard deviation of accuracy are used as the evaluation metric.

Experiment Results

Homophilic Graphs. The comparison of accuracy between GCL-JAM and the baselines on six homophilic graphs is shown in Table 1. First, it can be observed that GCL-JAM achieves the optimal performance on five of the six datasets, which illustrates the superiority of GCL-JAM for processing homophilic graphs. In particular, compared to the Graph Contrastive Learning (GCL) baselines that have the same network architecture and contrastive loss, GCL-JAM and the baselines with spectral augmentations (i.e., GCL-SPAN and SpCo) achieve consistent performance advantages across five homophilic benchmark datasets, which demonstrates the superior ability of spectral augmentation compared to spatial augmentation (especially random spatial augmentation) in capturing self-supervision information. Furthermore, compared to GCL-SPAN, which only considers spectral augmentations on graph topology, GCL-JAM

Methods	Chameleon	Squirrel	Actor	Cornell	Texas	Wisconsin
GCN	59.63±2.32	36.28±1.52	30.83±0.77	57.03±3.30	60.00±4.80	56.47±6.55
GAT	56.38±2.19	32.09±3.27	28.06±1.48	59.46±3.63	61.62±3.78	54.71±6.87
DeepWalk	47.74±2.05	32.93±1.58	22.78±0.64	39.18±5.57	46.49±6.49	33.53±4.92
Node2Vec	41.93±3.29	22.84±0.72	28.28±1.27	42.94±7.46	41.92±7.76	37.45±7.09
DGI	39.95±1.75	31.80±0.77	29.82±0.69	63.35±4.61	60.59±7.56	55.41±5.96
GMI	46.97±3.43	30.11±1.92	27.82±0.90	54.76±5.06	50.49±2.21	45.98±2.76
MVGRL	51.07±2.68	35.47±1.29	30.02±0.70	**64.30±5.43**	62.38±5.61	62.37±4.32
GRACE	48.05±1.81	31.33±1.22	29.01±0.78	54.86±6.95	57.57±5.68	50.00±5.83
GCA	49.80±1.81	35.50±0.91	29.65±1.47	55.41±4.56	59.46±6.16	50.78±4.06
BGRL	47.46±2.74	32.64±0.78	29.86±0.75	57.30±5.51	59.19±5.85	52.35±4.12
GraphMAE	59.02±1.93	37.08±1.01	29.61±0.60	51.08±4.78	52.33±5.06	52.11±6.48
SpCo	43.23±2.03	31.66±0.99	28.96±1.03	51.67±2.67	54.67±4.53	49.68±5.67
GCL-SPAN	46.36±1.40	35.16±1.04	28.63±0.93	52.37±4.48	58.95±6.05	55.10±4.31
GCL-JAM	**66.37±2.37**	**49.84±0.93**	**31.01±0.46**	61.08±2.43	**65.83±3.61**	**64.71±4.08**

Table 2: Node classification performance on heterophilic graphs. The metric is mean accuracy (%) and standard deviation. The best and the second best results are highlighted with bold and underline, respectively.



(a) Cora (b) Wisconsin (c) Texas

Figure 2: Cross-class attribute similarity. The left and right sides correspond to the original and the perturbed graphs.

achieves great performance improvement. This is mainly due to the coupling between the topology and attribute information, thus benefiting each other in spectral augmentation.

Heterophilic Graphs. Table 2 shows the experimental results on heterophilic graphs. It can be observed that GCL-JAM outperforms the baselines on five of the six datasets. Specifically, GCL-JAM outperforms the second-highest model (i.e., GCN and GraphMAE) by 6.74% and 12.76% on the Chameleon and Squirrel datasets, respectively. This is because that they contain many groups of nodes with the same neighborhoods and labels, making intra-class nodes more tightly connected in the attribute-interpolated graph. For Cornell, it has a higher homophily than other heterophilic graphs. Thus, the role of attribute nodes to enhance the homophilic edge density may be reduced. Nonetheless, it is evident that the superiority of GCL-JAM in processing heterophilic graphs. More importantly, it can be seen that the improvement of GCL-JAM over GCL-SPAN is more significant than that on homophilic graphs. It is because intra-class nodes have largely formed clusters on homophilic graphs. But on heterophilic graphs, GCL-SPAN perturbs inter-cluster edges solely upon topology making it more likely to connect intra-class nodes. In contrast, GCL-JAM reduces the proportion of inter-cluster edges between intra-class nodes by introducing attribute nodes which improves model performance significantly.

Effectiveness Study

Change of Attribute. To show the effect of GCL-JAM on attributes, we evaluate the changes about the similarity of the node attributes, as shown in Figure 2. To be more intuitive, the attribute similarity is normalized. Firstly, it can be observed that attribute similarity between intra-class nodes is higher than it between inter-class nodes. This is consistent with the assumptions of our previous theoretical analysis. Then we can observe that GCL-JAM can significantly increase the attribute similarity gap between intra-class nodes and inter-class nodes. It demonstrates that GCL-JAM can perturb noise attributes. Thus, more important attribute information is retained compared to the random perturbation.

Change of Topology. To intuitively understand the performance improvement due to GCL-JAM, a comparison about the change of topology is provided. For spatial domain, since the encoder used in GCL-JAM is GCN, the propagation between inter-class nodes should be reduced. We count the proportion of inter-class edges among the perturbed edges, as shown in Table 3. It can be observed that compared to GCL-SPAN, GCL-JAM can reach a higher proportion. For spectral domain, we compare the impact between GCL-JAM and uniformly random edge augmentation as shown in Figure 3. We divide the Laplacian decomposed components into 10 groups and measure the distance of Laplacians caused by graph augmentations in different frequency bands.

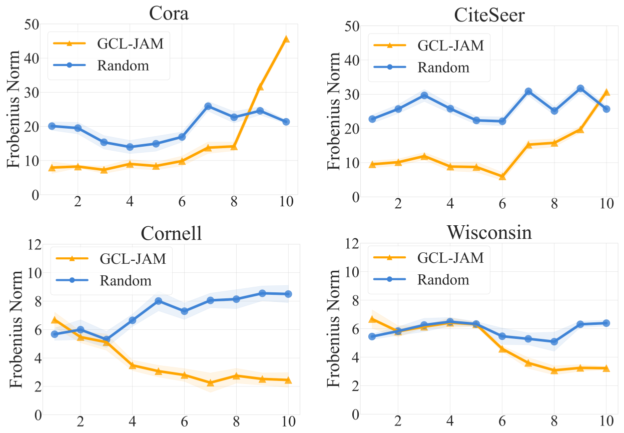


Figure 3: Frobenius distance of components of normalized Laplacian matrixes between original and augmented graph.

Dataset	Cora	CiteSeer	Cornell	Wisconsin
GCL-SPAN	0.80	0.81	0.64	0.76
GCL-JAM	**0.82**	**0.84**	**0.72**	**0.85**

Table 3: The proportion of inter-class edges among all perturbed edges.

The normalized m-th component Laplacian is defined as $\tilde{\mathbf{L}}^m = (\sum_{i\in[\frac{(m-1)N}{10},\frac{mN}{10})} u_i\lambda_i u_i^\top)/max(\lambda_i)$. The smaller m is, the lower the band frequency $\tilde{\mathbf{L}}^m$ indicates. We employ the Frobenius norm to measure their distance, $\|\tilde{\mathbf{L}}^m - \tilde{\mathbf{L}}_1^m\|_F$, where $\tilde{\mathbf{L}}_1^m$ corresponds to the augmented graph (Wang et al. 2022). As shown in the Figure 3, GCL-JAM is more impactful in the high frequency bands of the homophilic graphs and in the low frequency bands of the heterophilic graphs. Comparatively, uniform augmentation has a more even impact across the frequency bands or is the opposite of GCL-JAM. Thus, it can also be concluded that GCL-JAM is more effective from spectral domain.

Ablation Study

An ablation study is conducted to delve into the impact on the performance of introducing two types of information, i.e., topology (A) and attribute (X). The comparison results are shown in Table 4, where ✓ means the corresponding information is considered during spectral augmentation. First, it can be observed that compared to random spatial augmentation (in the first row), which has been proven to improve robustness, spectral ones (in the last three rows) achieve performance advantages in most cases, which illustrates their effectiveness and adaptability. This is mainly due to the fact that topology and attribute induced by spectral augmentation can fulfill the requirements of the employed GCN encoder, thus the discriminative node representations are obtained after the encoding. In addition, an important observation is that the proposed augmentation strategy, which spectral augmenting topology and attribute simultaneously, outperforms all spectral variants, which emphasizes the necessity of the co-augmentation and the validity of the proposed scheme.

A	**X**	CiteSeer	Wiki-CS	Chameleon	Squirrel
		71.06±0.45	78.61±0.23	49.65±1.79	32.46±1.18
✓		72.79±0.61	81.36±0.13	51.26±1.40	35.23±1.09
	✓	71.80±0.75	78.84±0.29	55.80±2.15	34.96±1.27
✓	✓	**73.29±0.39**	**82.09±0.42**	**66.37±2.37**	**49.84±0.93**

Table 4: Consideration of different components when performing spectral augmentation.

Dataset	Cora	Photo	Chameleon	Squirrel
GRACE	83.44±0.39	92.65±0.32	48.05±1.71	31.33±1.22
+JAM	**85.54±0.53**	**94.35±0.21**	**66.37±2.37**	**49.84±0.93**
MVGRL	83.01±0.42	92.01±0.13	51.07±2.68	35.47±1.29
+JAM	**84.26±0.55**	**94.26±0.19**	**64.04±1.90**	**47.18±0.68**
BGRL	82.67±0.78	93.24±0.29	47.46±2.74	32.64±0.78
+JAM	**84.19±0.47**	**93.81±0.10**	**69.05±1.18**	**48.09±0.89**

Table 5: Plug the JAM augmentation to different GCL frameworks, denoted by +JAM.

Since the work focuses on the augmentation method, we evaluate the effectiveness of GCL-JAM with different GCL frameworks by an ablation study. We choose three well-used contrastive learning models, including GRACE (local-local), MVGRL (local-global) and BGRL (bootstrapping). Specifically, the augmentation of these models is replaced with joint spectral augmentation. The results are shown in Table 5. We can observe that the proposed augmentation strategy makes all three contrastive learning frameworks improved significantly on four datasets with various homophily degrees. It is shown that GCL-JAM combining topology and attribute retains more important information than random uniform augmentation, and therefore more discriminative representations can be learned by GCL-JAM. In addition, these results illustrate that the superiority of GCL-JAM does not depend on the choice of contrastive learning framework, but rather because of the effectiveness of the method itself.

Conclusions

Both node attributes and graph topology are important information in graph and there are correlations between them. Unfortunately, existing Graph Contrastive Learning (GCL) models augment attributes and topology separately, ignoring their correlations and resulting in information loss. To address the above issue, we propose a novel GCL model with joint spectral augmentation of attribute and topology (GCL-JAM). The main idea is to generate an attribute-interpolated graph by regarding node attributes as attribute nodes and then perform spectral augmentation. We have shown experimentally and theoretically that GCL-JAM can perturb inter-class edges and noise attributes. Therefore, more distinguishable node representations can be obtained. Extensive experiments on both homophilic graphs and heterophilic graphs demonstrate the superior performance of GCL-JAM.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. U22B2036, 62376088, 62276187, 62102413, 62272020), in part by the Hebei Natural Science Foundation (No. F2024202047, F2024202068), in part by the National Science Fund for Distinguished Young Scholarship (No. 62025602), in part by the Science Research Project of Hebei Education Department (BJK2024172), in part by the National Key R&D Program of China (No. 2024YFB3311901), in part by the Guangxi Key Laboratory of Machine Vision and Intelligent Control (2023B03), in part by the Hebei Yanzhao Golden Platform Talent Gathering Programme Core Talent Project (Education Platform) (HJZD202509), and in part by the XPLORER PRIZE.

References

Bojchevski, A.; and Günnemann, S. 2019. Adversarial attacks on node embeddings via graph poisoning. In *ICML*, 695–704. PMLR.

Chang, H.; Rong, Y.; Xu, T.; Bian, Y.; Zhou, S.; Wang, X.; Huang, J.; and Zhu, W. 2021. Not all low-pass filters are robust in graph convolutional networks. In *NeurIPS*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *ICML*, 1597–1607. PMLR.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *EMNLP 2021-2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*.

Gasteiger, J.; Bojchevski, A.; and Günnemann, S. 2019. Combining Neural Networks with Personalized PageRank for Classification on Graphs. In *ICLR*.

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *SIGKDD*, 855–864.

Hassani, K.; and Khasahmadi, A. H. 2020. Contrastive multi-view representation learning on graphs. In *ICML*, 4116–4126. PMLR.

Hou, Z.; Liu, X.; Cen, Y.; Dong, Y.; Yang, H.; Wang, C.; and Tang, J. 2022. Graphmae: Self-supervised masked graph autoencoders. In *SIGKDD*, 594–604.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

Li, M. M.; Huang, K.; and Zitnik, M. 2021. Representation learning for networks in biology and medicine: advancements, challenges, and opportunities. *arXiv preprint arXiv:2104.04883*, 8.

Lin, L.; Chen, J.; and Wang, H. 2023. Spectral Augmentation for Self-Supervised Learning on Graphs. In *ICLR*.

Liu, N.; Wang, X.; Bo, D.; Shi, C.; and Pei, J. 2022. Revisiting graph contrastive learning from the perspective of graph spectrum. In *NeurIPS*.

Liu, Y.; Zheng, Y.; Zhang, D.; Lee, V. C.; and Pan, S. 2023. Beyond smoothing: Unsupervised graph representation learning with edge heterophily discriminating. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 4516–4524.

Mernyei, P.; and Cangea, C. 2020. Wiki-cs: A wikipedia-based benchmark for graph neural networks. *arXiv preprint arXiv:2007.02901*.

Mo, Y.; Peng, L.; Xu, J.; Shi, X.; and Zhu, X. 2022. Simple unsupervised graph representation learning. In *AAAI*, volume 36, 7797–7805.

Newman, M. E.; and Girvan, M. 2004. Finding and evaluating community structure in networks. *Physical review E*, 69(2): 026113.

Ng, A.; Jordan, M.; and Weiss, Y. 2001. On spectral clustering: Analysis and an algorithm. In *NeurIPS*.

Parlett, B. N.; and Scott, D. S. 1979. The Lanczos algorithm with selective orthogonalization. *Mathematics of computation*, 33(145): 217–238.

Pei, H.; Wei, B.; Chang, K. C.-C.; Lei, Y.; and Yang, B. 2020. Geom-GCN: Geometric Graph Convolutional Networks. In *ICLR*.

Peng, Z.; Huang, W.; Luo, M.; Zheng, Q.; Rong, Y.; Xu, T.; and Huang, J. 2020. Graph representation learning via graphical mutual information maximization. In *WWW*, 259–270.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *SIGKDD*, 701–710.

Qiu, J.; Chen, Q.; Dong, Y.; Zhang, J.; Yang, H.; Ding, M.; Wang, K.; and Tang, J. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *SIGKDD*, 1150–1160.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29(3): 93–93.

Shchur, O.; Mumme, M.; Bojchevski, A.; and Günnemann, S. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*.

Shuman, D.; Narang, S.; Frossard, P.; Ortega, A.; and Vanderghenyst, P. 2013. The Emmerging Field of Signal Processing on Graphs. *IEEE Signal Proc. Magazine*.

Thakoor, S.; Tallec, C.; Azar, M. G.; Azabou, M.; Dyer, E. L.; Munos, R.; Veličković, P.; and Valko, M. 2022. Large-Scale Representation Learning on Graphs via Bootstrapping. In *ICLR*.

Thakoor, S.; Tallec, C.; Azar, M. G.; Munos, R.; Veličković, P.; and Valko, M. 2021. Bootstrapped representation learning on graphs. In *ICLR*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *ICLR*.

Veličković, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep Graph Infomax. In *ICLR*.

Von Luxburg, U. 2007. A tutorial on spectral clustering. *Statistics and computing*, 17: 395–416.

Wang, H.; Zhang, J.; Zhu, Q.; and Huang, W. 2022. Augmentation-free graph contrastive learning with performance guarantee. *arXiv preprint arXiv:2204.04874*.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019. Simplifying graph convolutional networks. In *ICML*, 6861–6871. PMLR.

Xu, K.; Li, C.; Tian, Y.; Sonobe, T.; Kawarabayashi, K.-i.; and Jegelka, S. 2018. Representation learning on graphs with jumping knowledge networks. In *ICML*, 5453–5462. PMLR.

Yang, K.; Han, H.; Jin, W.; and Liu, H. 2023. Augment with Care: Enhancing Graph Contrastive Learning with Selective Spectrum Perturbation. *arXiv preprint arXiv:2310.13845*.

Yang, L.; Wang, C.; Gu, J.; Cao, X.; and Niu, B. 2021. Why do attributes propagate in graph convolutional neural networks? In *AAAI*, volume 35, 4590–4598.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph contrastive learning with augmentations. In *NeurIPS*.

Zhu, M.; Wang, X.; Shi, C.; Ji, H.; and Cui, P. 2021a. Interpreting and unifying graph neural networks with an optimization framework. In *WWW*, 1215–1226.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2020. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2021b. Graph contrastive learning with adaptive augmentation. In *WWW*, 2069–2080.

Zhuo, J.; Cui, C.; Fu, K.; Niu, B.; He, D.; Wang, C.; Guo, Y.; Wang, Z.; Cao, X.; and Yang, L. 2024a. Graph contrastive learning reimagined: Exploring universality. In *ACM MM*, 641–651.

Zhuo, J.; Lu, Y.; Ning, H.; Fu, K.; Niu, B.; He, D.; Wang, C.; Guo, Y.; Wang, Z.; Cao, X.; et al. 2024b. Unified Graph Augmentations for Generalized Contrastive Learning on Graphs. In *NeurIPS*.

Zhuo, J.; Qin, F.; Cui, C.; Fu, K.; Niu, B.; Wang, M.; Guo, Y.; Wang, C.; Wang, Z.; Cao, X.; et al. 2024c. Improving Graph Contrastive Learning via Adaptive Positive Sampling. In *CVPR*, 23179–23187.