1. APPENDIX

1.1. Appendix A:Preliminaries and Related Work

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X)$ be a graph, where $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$ is the node set, $E \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set, $|E|$ is edge number of origin graph, $X \in \mathbb{R}^{N \times F}$ is the attribute matrix of all nodes where $x_i \in \mathbb{R}^F$ is the feature vector of v_i. We use $A \in \{0,1\}^{N \times N}$ to represent the adjacency matrix of $\mathcal{G}$, and $A_{ij} = 1$ if$(v_i, v_j) \in E$. $\hat{A}$ stands for the adjacency matrix for a graph with added self-loops I. D and $\hat{D}$ denote the diagonal degree matrix of A and $\hat{A}$. Hence, an attributed graph can be described as $\mathcal{G} = (X, A)$ for simplicity.

1.1.1. *Spectral Decomposition*

The spectral decomposition of the normalized Laplacian matrix L_{norm} is defined as $\mathrm{L}_{\text{norm}} = \mathrm{Lap}(A) = U\Lambda U^\top$, where the diagonal matrix $\Lambda = \mathrm{eig}(\mathrm{Lap}(A)) = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ consists of real eigenvalues known as the graph spectrum, and $U = [u_1\, u_2\, \ldots\, u_n] \in \mathbb{R}^{n \times n}$ are the corresponding orthonormal eigenvectors known as the spectral bases.

1.1.2. *Graph Neural Networks*

Graph Neural Networks (GNNs) are a class of neural networks designed to work with graph-structured data. One popular variant of GNNs is the Graph Convolutional Network (GCN). In a GCN, each node aggregates features from its neighbors, incorporating both its own features and those of its neighbors to update its representation. This process is formulated as follows:

$$\mathbf{H}^{(l+1)} = \sigma\left(\mathbf{D}^{-\frac{1}{2}}\hat{\mathbf{A}}\mathbf{D}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}\right), \tag{1}$$

1.1.3. *Graph Contrastive Learning*

Graph contrastive learning approaches usually design different views and aim to pre-train a graph encoder by maximizing agreement between representations of views. Generally, given an origin graph $\mathcal{G} = (X, A)$, two augmented views are denoted as $\mathcal{G}_1 = (X_1, A_1)$ and $\mathcal{G}_2 = (X_2, A_2)$, by applying the data augmentation function $t(\cdot)$. The representations of all nodes in augmented views are denoted as $H^1 = f_\theta(X^1, A^1)$ and $H^2 = f_\theta(X^2, A^2)$, where $f_\theta(\cdot)$ is an GNN encoder. The agreement between the node representations is commonly measured through Mutual Information (MI). Thus, the contrastive objective can be generally formulated as:

$$\max_\theta \mathcal{MI}(\mathbf{H}^1, \mathbf{H}^2) \tag{2}$$

1.2. Appendix B:Method and Theoretical Analysis

1.2.1. *Detailed Description of Method-Structure Awareness*

In this paper, we propose a novel GCL framework, named GCL-LSAA. Our proposed framework consists of two main components: (i) Learnable Structure Awareness Augmentation. (ii)multi-view contrastive learning strategy. We have presented the overall framework in Figure 2.To generate augmented graph views that are both structurally meaningful and semantically rich, we design a structure-aware connection probability matrix that effectively fuses topological relationships with node attribute similarities. This construction involves four key steps: local similarity estimation, normalization, node-pair adaptive fusion, and final probability formulation.

First, at the attribute level, we measure the semantic similarity between nodes using the cosine similarity of their feature vectors. Given the node feature matrix $X \in \mathbb{R}^{N \times F}$, the similarity between nodes v_i and v_j is defined as:

$$\mathrm{sim}(x_i, x_j) = \frac{x_i \cdot x_j}{\|x_i\|\|x_j\|}. \tag{3}$$

Relying solely on attribute similarity, however, overlooks important structural information. Computing all-pairs shortest paths has a prohibitive complexity of $O(N^3)$, making it infeasible for large graphs. To address this, we approximate structural relations by performing a Breadth-First Search (BFS) from each node up to a maximum depth K, thereby focusing on local neighborhoods. Let $N_K(v_i)$ denote the set of nodes within distance K from v_i, and $N_k(v_i)$ the nodes exactly at distance k:

$$N_K(v_i) = \{v_j \in V \mid \mathrm{dist}(v_i, v_j) \le K\} \tag{4}$$

$$N_k(v_i) = \{v_j \in V \mid \mathrm{dist}(v_i, v_j) = k\}. \tag{5}$$

Using this, we define the structural similarity matrix $P \in \mathbb{R}^{N \times N}$ as:

$$P_{i,j} = \begin{cases} \frac{1}{k}, & \text{if } v_j \in N_k(v_i), \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Similarly, to capture local attribute similarity without dense computations, we restrict attribute similarity calculations to BFS-reachable neighbors:

$$S_{i,j} = \begin{cases} \mathrm{sim}(x_i, x_j), & \text{if } v_j \in N_k(v_i), \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Next, to convert both P and S into valid probability distributions, we apply row-wise normalization:

$$P_{i,j}^{\text{nor}} = \frac{P_{i,j}}{\sum_{k \in N_K(v_i)} P_{i,k} + \epsilon}, \quad S_{i,j}^{\text{nor}} = \frac{S_{i,j}}{\sum_{k \in N_K(v_i)} S_{i,k} + \epsilon}, \tag{8}$$

where ϵ is a small constant for numerical stability.

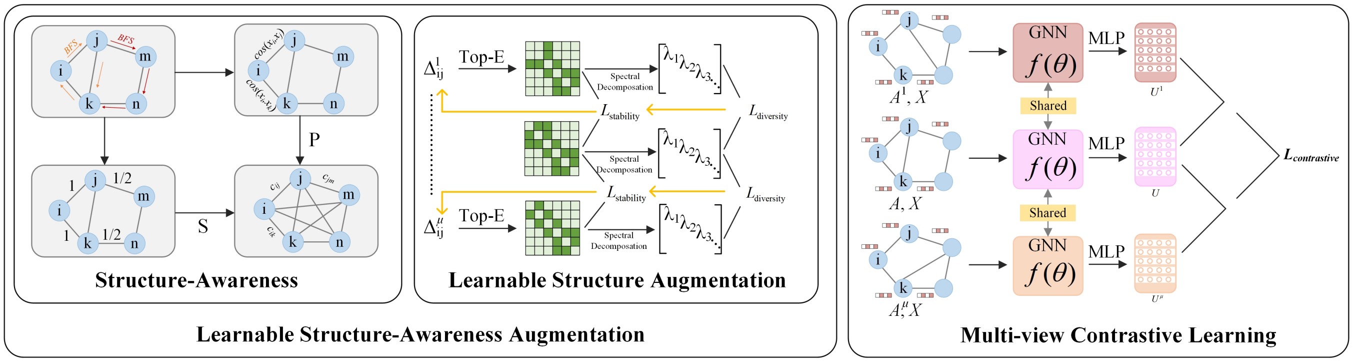


Fig. 1: GCL-LSAA model.

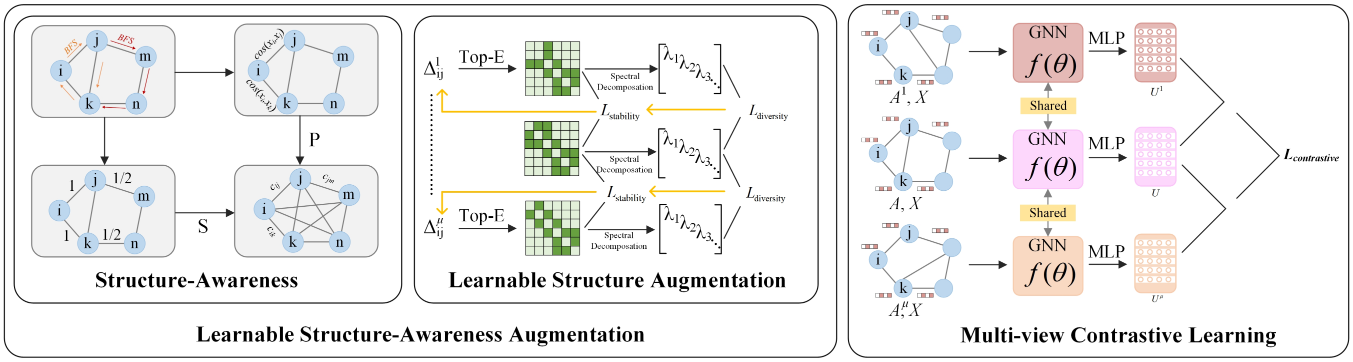


Fig. 2: GCL-LSAA model.

To adaptively fuse the structural and attribute similarities for each node pair, we introduce a fusion weight $\omega_{i,j}$, defined as:

$$\omega_{i,j} = \sigma\left(\alpha \cdot \mathrm{sim}(x_i, x_j) + \beta \cdot \left(1 - \frac{d(i,j)}{K}\right)\right), \qquad (9)$$

where $\sigma(\cdot)$ is the sigmoid function, $d(i,j)$ denotes the BFS distance between v_i and v_j, and α, β are balance coefficients. We empirically set $\alpha = \beta = 1$ in all experiments. This formulation assigns higher fusion weights to node pairs that are both semantically similar and topologically close, and lower weights otherwise.

Finally, the structure-aware connection probability matrix C is computed as a weighted combination of the normalized similarities:

$$C_{i,j} = \omega_{i,j} \cdot S^{\mathrm{nor}}_{i,j} + (1 - \omega_{i,j}) \cdot P^{\mathrm{nor}}_{i,j}. \qquad (10)$$

To ensure valid probability distributions, we normalize C row-wise:

$$C^{\mathrm{nor}}_{i,j} = \frac{C_{i,j}}{\sum_{j=1}^{N} C_{i,j} + \epsilon}. \qquad (11)$$

1.2.2. Theoretical Analysis

Structure Expressiveness of LSAA Augmentation

We first show that the LSAA mechanism, composed of a neural perturbation controller and the Gumbel–Sinkhorn operator, can approximate any symmetric, bounded, and sparse adjacency matrix.

Let $A^* \in \mathbb{R}^{N\times N}$ be any symmetric and bounded target adjacency matrix such that $A^* = (A^*)^\top$ and $A^*_{ij} \in [0,1]$. For any $\varepsilon > 0$, there exists a learnable perturbation network

$$\psi^{(\mu)} : \mathbb{R}^{N\times d} \to \mathbb{R}^{N\times N} \qquad (12)$$

and a temperature $\tau > 0$ such that the generated LSAA view

$$\Delta^{(\mu)} = S_\tau\left(\log C + \epsilon^{(\mu)} + \psi^{(\mu)}(X)\right), \qquad (13)$$

$$A^{(\mu)} = \frac{1}{2}\left(\Delta^{(\mu)} + (\Delta^{(\mu)})^\top\right) \qquad (14)$$

satisfies

$$\|A^{(\mu)} - A^*\|_F < \varepsilon. \qquad (15)$$

Proof: First, by the Universal Approximation Theorem, for any target matrix $B^* \in \mathbb{R}^{N\times N}$ and any $\delta > 0$, there exists a neural network $\psi^{(\mu)}$ such that

$$\|\psi^{(\mu)}(X) - B^*\|_F < \delta. \qquad (16)$$

Thus, we can approximate any structural deviation matrix arbitrarily well.

Next, from , the Gumbel–Sinkhorn operator $S_\tau(\cdot)$ is continuous with respect to its input logits. As the temperature $\tau \to 0$, it converges to a permutation matrix. In particular, for any doubly-stochastic matrix D^*, one can choose $\tau > 0$ sufficiently small so that

$$\|S_\tau(\Theta) - D^*\|_F < \frac{\varepsilon}{2}. \tag{17}$$

Finally, we combine these two observations. Define

$$\Delta^{(\mu)} = S_\tau\left(\log C + \epsilon^{(\mu)} + \psi^{(\mu)}(X)\right), \tag{18}$$

$$A^{(\mu)} = \frac{1}{2}\left(\Delta^{(\mu)} + (\Delta^{(\mu)})^\top\right). \tag{19}$$

Since $\psi^{(\mu)}(X)$ can approximate any desired bias matrix and $S_\tau(\cdot)$ can approximate any doubly-stochastic matrix closely, we ensure

$$\|\Delta^{(\mu)} - A^*\|_F < \varepsilon. \tag{20}$$

Because

$$\|A^{(\mu)} - A^*\|_F = \left\|\frac{1}{2}\left(\Delta^{(\mu)} + (\Delta^{(\mu)})^\top\right) - A^*\right\|_F \le \|\Delta^{(\mu)} - A^*\|_F < \varepsilon, \tag{21}$$

the proof is complete.

This result ensures that our augmentation module can flexibly produce structurally meaningful and diverse graph views in the LSAA framework.

Theorem 1 (Unified Structure Preservation Bounds Representation Shift). Let $f(A, X) : \mathbb{R}^{N\times N} \times \mathbb{R}^{N\times d} \to \mathbb{R}^{N\times d'}$ be an encoder that is L-Lipschitz continuous with respect to the adjacency matrix A, i.e., for any graphs A_1, A_2,

$$\|f(A_1, X) - f(A_2, X)\|_F \le L \cdot \|A_1 - A_2\|_F. \tag{22}$$

Define the representation shift between an augmented view $A^{(\mu)}$ and the original graph A as

$$D^{(\mu)} := \|f(A^{(\mu)}, X) - f(A, X)\|_F^2. \tag{23}$$

Let the unified structure preservation loss be be defined as in Equation **??**. Then, $D^{(\mu)}$ satisfies the following bounds:

$$D^{(\mu)} \le L^2 \cdot \|A^{(\mu)} - A\|_F^2 \quad \text{(Upper bound)} \tag{24}$$

$$D^{(\mu)} \ge \kappa \cdot \sum_{i=1}^{k}\left(\lambda_i^{(\mu)} - \lambda_i^{\text{orig}}\right)^2 \quad \text{(Lower bound)}, \tag{25}$$

where $\kappa > 0$ depends on the encoder's spectral sensitivity.

Proof: (1) Upper Bound via Lipschitz Continuity.

By assumption,

$$\|f(A^{(\mu)}, X) - f(A, X)\|_F \le L \cdot \|A^{(\mu)} - A\|_F, \tag{26}$$

and squaring both sides yields

$$D^{(\mu)} \le L^2 \cdot \|A^{(\mu)} - A\|_F^2. \tag{27}$$

(2) Lower Bound via Spectral Stability.

Let $\mathcal{L} = D - A = U\Lambda U^\top$ and $\mathcal{L}^{(\mu)} = D^{(\mu)} - A^{(\mu)} = U^{(\mu)}\Lambda^{(\mu)}U^{(\mu)\top}$ be the Laplacian eigendecompositions. For spectral GNNs with filtering function $g(\cdot)$, the outputs are:

$$f(A, X) = U \cdot g(\Lambda) \cdot U^\top X, \quad f(A^{(\mu)}, X) = U^{(\mu)} \cdot g(\Lambda^{(\mu)}) \cdot U^{(\mu)\top} X. \tag{28}$$

Assuming g is Lipschitz and smooth on low-frequency eigenvalues $\lambda_1, \ldots, \lambda_k$, then for some $\kappa > 0$,

$$\|f(A^{(\mu)}, X) - f(A, X)\|_F^2 \ge \kappa \cdot \sum_{i=1}^{k}\left(\lambda_i^{(\mu)} - \lambda_i^{\text{orig}}\right)^2. \tag{29}$$

(3) Edge Term Controls Frobenius Perturbation.

The sparsity constraint

$$\left|\sum_{i,j} A_{ij}^{(\mu)} - E_{\text{target}}\right| \tag{30}$$

encourages the total edge weight of $A^{(\mu)}$ to remain close to that of A, which implies:

$$\sum_{i,j} A_{ij}^{(\mu)} \approx \sum_{i,j} A_{ij} \Rightarrow \|A^{(\mu)} - A\|_F^2 \text{ is constrained.} \tag{31}$$

Therefore, minimizing the edge term indirectly tightens the upper bound on $D^{(\mu)}$ via Frobenius norm control.

■

Corollary.

Minimizing the unified loss $\mathcal{L}_{\text{preserve}}$ simultaneously enforces: (i) local structure proximity, (ii) global spectral alignment, and (iii) structural sparsity, thereby bounding representation shifts from both above and below.

Theorem 2 (Spectral Diversity Enhances Inter-view Discriminability).

Let $f(A, X) : \mathbb{R}^{N\times N} \times \mathbb{R}^{N\times d} \to \mathbb{R}^{N\times d'}$ be a graph encoder that is spectrally sensitive to high-frequency Laplacian components. Given two augmented graphs $A^{(\mu)}$ and $A^{(\nu)}$, let their representations be:

$$Z^{(\mu)} = f(A^{(\mu)}, X), \quad Z^{(\nu)} = f(A^{(\nu)}, X) \tag{32}$$

and define the inter-view representation discrepancy:

$$D^{(\mu,\nu)} := \|Z^{(\mu)} - Z^{(\nu)}\|_F^2. \tag{33}$$

Assume that the encoder satisfies the following spectral sensitivity condition:

$$D^{(\mu,\nu)} \ge \eta \cdot \sum_{i=k+1}^{N}\left(\lambda_i^{(\mu)} - \lambda_i^{(\nu)}\right)^2 \tag{34}$$

for some $\eta > 0$, where $\lambda_i^{(\mu)}$ denotes the i-th Laplacian eigenvalue of $A^{(\mu)}$, and similarly for $A^{(\nu)}$. Then, maximizing the high-frequency spectral diversity loss

$$\mathcal{L}_{\text{div}-\text{high}} = \sum_{\mu\neq\nu}\sum_{i=k+1}^{N}\left(\lambda_i^{(\mu)} - \lambda_i^{(\nu)}\right)^2 \tag{35}$$

increases $D^{(\mu,\nu)}$, thereby improving the separability of representations among augmented views.

Proof: (1) *Spectral Filtering Encoder Model.*

Assume $f(A, X)$ operates in the spectral domain via graph Laplacian decomposition:

$$\mathcal{L}^{(\mu)} = D^{(\mu)} - A^{(\mu)} = U^{(\mu)}\Lambda^{(\mu)}U^{(\mu)\top}, \tag{36}$$

$$f(A^{(\mu)}, X) = U^{(\mu)}g(\Lambda^{(\mu)})U^{(\mu)\top}X, \tag{37}$$

where $g(\cdot)$ is a spectral filter function. For GCN-like filters that emphasize higher frequencies, the main signal difference lies in the components associated with eigenvalues $\lambda_{k+1}, \ldots, \lambda_N$.

(2) Representation Discrepancy Bound.

Under the encoder's spectral sensitivity, we assume there exists $\eta > 0$ such that:

$$D^{(\mu,\nu)} = \|f(A^{(\mu)}, X) - f(A^{(\nu)}, X)\|_F^2 \geq \eta \cdot \sum_{i=k+1}^{N} \left(\lambda_i^{(\mu)} - \lambda_i^{(\nu)}\right)^2. \tag{38}$$

(3) Conclusion.

Therefore, maximizing the high-frequency spectral diversity term

$$\mathcal{L}_{\text{div}-\text{high}} = \sum_{\mu \neq \nu} \sum_{i=k+1}^{N} \left(\lambda_i^{(\mu)} - \lambda_i^{(\nu)}\right)^2 \tag{39}$$

leads to a proportional increase in $D^{(\mu,\nu)}$, enhancing the contrastive signal and improving the discriminability among views.

Corollary: The spectral diversity loss acts as a lower bound controller on the inter-view distance, complementing the structural preservation loss that controls upper bounds. Together, they form a frequency-aware representation constraint that balances stability and diversity.

1.3. Supplemental Experiment

2. EXPERIMENT SETUP DETAILS

2.1. Hardware Specification and Environment

We implement our proposed framework in PyTorch and optimize it with Adam. For all the baseline methods, we also use Adam as the optimizer. All experiments are conducted on a Nvidia Tesla v100 16GB GPU.

2.2. Details on Datasets

2.3. Cora

The Cora[1] dataset is a widely used benchmark in graph-based machine learning, comprising a citation network of academic papers. It contains 2708 nodes, each representing a paper, and 10556 edges indicating citations between these papers. Each node is described by a 1,433-dimensional sparse bag-of-words feature vector derived from the text of the papers. The dataset is categorized into 7 classes, making it a popular choice for evaluating node classification and graph neural network models. You can access the Cora dataset.

Table 1: Detailed information of the datasets.

Datasets	Nodes	Edges	Features	Labels
Cora	2708	10556	1433	7
Citeseer	3327	9104	3703	6
PubMed	19717	88648	500	3
Wikics	11701	431726	300	10
Amazon-Photo	7650	238162	745	8
Amazon-Computers	13752	491722	767	10

2.4. CiteSeer

The CiteSeer[1] dataset is a widely used benchmark in graph-based machine learning, comprising a citation network of academic papers. It contains 3327 nodes, each representing a paper, and 9104 edges indicating citations between these papers. Each node is described by a 3703-dimensional binary bag-of-words feature vector derived from the text of the papers. The dataset is categorized into 6 classes, making it a popular choice for evaluating node classification and graph neural network models.

2.5. PubMed

The PubMed[1] dataset is a benchmark in graph-based machine learning, featuring a citation network of academic papers. It consists of 19,717 nodes, each representing a paper, and 88648 edges indicating citations between these papers. Each node is described by a 500-dimensional binary bag-of-words feature vector derived from the text of the papers. The dataset is categorized into 3 classes, making it a valuable resource for evaluating node classification and graph neural network models.

2.6. WikiCS

The WikiCS[2] dataset is a benchmark dataset used for graph learning, comprising a citation network of academic papers. It includes 11,701 nodes, each representing a paper, and 431726 edges indicating citations between these papers. Each node is described by a 2,078-dimensional binary bag-of-words feature vector derived from the text of the papers. The dataset is categorized into 10 classes, making it suitable for evaluating node classification and graph neural network models.

2.7. Amazon-Photo

The Amazon-Photo[3] dataset is a benchmark dataset used for graph learning and recommendation systems. It consists

Table 2: The Hyperparameters for Learnable Structure Awareness Augmentation.

Datasets	μ	λ	K	ω	α
Cora	5	0.4	4	5	0.1
Citeseer	5	0.2	5	5	0.2
PubMed	2	0.3	3	3	0.3
Wikics	3	0.3	4	1	0.2
Amazon-Photo	5	0.3	4	5	0.3
Amazon-Computers	4	0.2	2	3	0.2

of a photo-to-photo co-purchase network where each node represents a photo, and edges represent co-purchase relationships between photos. The dataset includes 7,650 nodes and 238162 edges. Each node is described by a 745-dimensional feature vector, which includes attributes such as photo tags and descriptions. The dataset is often used to evaluate models in the context of graph-based recommendation and classification tasks.

2.8. Amazon-Computers

The Amazon-Computers[3] dataset is a standard dataset used for graph learning and recommendation systems. It includes a co-purchase network of computer products, where each node represents a computer product, and edges indicate co-purchase relationships between products. The dataset comprises 13752 nodes and 491722 edges. Each node is described by a 767-dimensional feature vector, including product category labels and descriptions. This dataset is commonly used for evaluating models in graph-based recommendation and classification tasks.

2.9. Detailed Model Configurations: Learnable Structure Awareness Augmentation and Contrastive Learning

The hyperparameters used in the model are mainly divided into two parts: one part is related to the Learnable Structure Awareness Augmentation (LSAA) and the other part is related to the contrastive learning.

2.9.1. Learnable Structure Awareness Augmentation

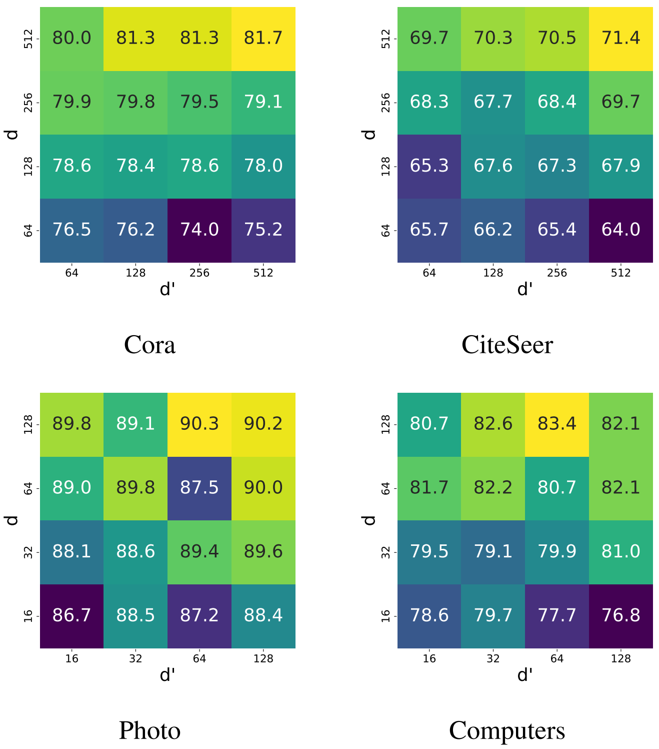
For each dataset, the parameters we selected are shown in Table 2. In LSAA, the following hyperparameters play a crucial role:

- μ: This hyperparameter controls the number of augmented views generated for each graph. Increasing μ allows the model to learn from multiple perspectives of the graph, potentially improving the representation learning. It is crucial for capturing diverse information across different views.
- λ: This parameter is used to balance the trade-off between maintaining structural stability and introducing diversity in the augmented views. A higher value of λ emphasizes the spectral changes in the graph, enhancing the diversity of the generated views. This balance is key to ensuring that the augmented graph maintains its structural integrity while improving its representational power.
- K: In the Breadth-First Search (BFS) process, K represents the maximum depth of the search. It determines the extent of the neighborhood considered when capturing the structural features of the graph. Choosing an appropriate K value helps balance the local and global structural information captured from the graph. In our experiments, we evaluated the performance of the model across different datasets while varying the K value. The results are shown in Figure 4. For different datasets, there is an optimal range of K values that maximizes performance. For the Cora and CiteSeer datasets, performance typically peaks within the range of $K = 5$ to $K = 6$, where the balance between captured local and global structural information is best. In contrast, for the Computers dataset, the best performance is observed at $K = 2$, where the graph features are captured effectively.
- ω: This is the temperature parameter used in the Gumbel-Softmax to control the approximation of discrete distributions. A smaller ω results in a distribution closer to the discrete categorical distribution, while a larger ω provides a smoother approximation. Proper setting of ω is essential for effective gradient-based optimization.
- E: This parameter defines the number of top edges selected from the Gumbel-Softmax distribution to form the symmetric adjacency matrix. For each dataset, we choose E to be the same as the original graph.
- α: This threshold parameter is used to control the penalty term for diversity loss. It helps in regulating the trade-off between structural similarity and spectral changes. By adjusting α, one can fine-tune the balance between preserving the original graph structure and introducing diversity in the augmented views.

2.9.2. Contrastive Learning

Table 3 presents the hyperparameters used for contrastive learning across different datasets. Each dataset's configuration includes the number of encoder layers, output dimension, contrastive learning temperature parameter τ, and learning rate. Specifically:

- Encoder Layers: The number of layers in the encoder. All datasets use 2 encoder layers.



Cora

CiteSeer

Photo

Computers

Fig. 3: Impact of the output dimension of GNN encoder d and dimension of the projection head d' on the classification results.

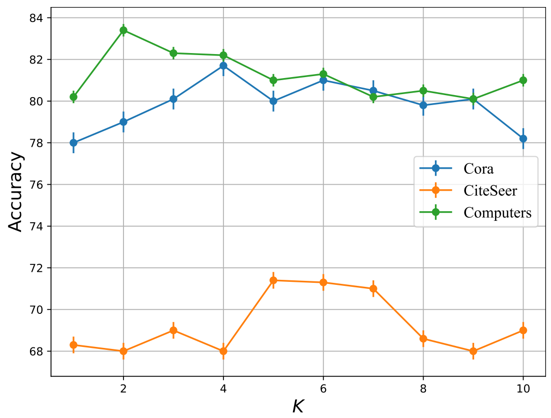


Fig. 4: Impact of K on Model Performance Across Datasets

Table 3: The Hyperparameters for Contrastive Learning.

Datasets	Encoder Layers	Output Dimension	τ	μ	Learning Rate
Cora	2	512	0.5	5	1e-05
Citeseer	2	512	0.5	5	1e-03
PubMed	2	128	0.3	3	1e-03
Wikics	2	128	0.4	1	1e-03
Amazon-Photo	2	128	0.4	5	1e-02
Amazon-Computers	2	128	0.3	3	1e-03

- Output Dimension: The dimension of the output for each layer. Cora and Citeseer have an output dimension of 512, while PubMed, Wikics, Amazon-Photo, and Amazon-Computers have 128. We conducted experiments with different output dimensions and projection dimensions. The results are illustrated in Figure 3.
- τ: The temperature parameter in contrastive learning, which controls the distance between positive and negative samples. Values vary across datasets, such as 0.5 for Cora and Citeseer, and 0.3 for PubMed.
- Learning Rate: The rate at which the model updates during training. It varies across datasets, with values such as 1e-05 for Cora and 1e-02 for Amazon-Photo.

2.10. Hyperparameter Analysis

2.10.1. The augmented view count μ

Figure **??** shows the sensitivity analysis on the hyperparameters μ of five various graph datasets. We observed that with the same temperature coefficient, as the number of views μ increases, the accuracy also increases. This indicates that increasing the number of views brings more information and helps improve performance. However, with further increase in the number of views, we observed a decreasing trend in performance on the Pubmed dataset and the Computers dataset. We speculate that this may be due to information redundancy, where the additional views no longer contribute positively to the performance improvement of downstream tasks.

2.10.2. The temperature coefficient τ

Figure **??** shows the sensitivity analysis on the hyperparameters τ of five various graph datasets. Overall, with the same number of augmented views, as the temperature coefficient increases, the accuracy shows a trend of initially increasing and then decreasing.

2.10.3. Trade-off coefficient λ and diversity loss threshold α

Figure 5 shows the impact of the trade-off coefficient λ and diversity loss threshold α on node classification performance

Table 4: Results of the ablation study on the node classification task.

Method	Cora	CiteSeer	Computers	Photo
w/o P	80.7(±1.4) $\downarrow$ 1.0	68.3(±1.3) $\downarrow$ 3.1	80.2(±1.5) $\downarrow$ 3.2	87.3(±0.7) $\downarrow$ 3.0
w/o S	80.3(±1.5) $\downarrow$ 1.4	69.0(±1.2) $\downarrow$ 2.4	81.8(±2.1) $\downarrow$ 1.6	88.3(±0.8) $\downarrow$ 2.0
w/o stability	78.3(±2.0) $\downarrow$ 3.4	69.2(±1.2) $\downarrow$ 2.2	80.7(±1.4) $\downarrow$ 2.7	89.3(±0.6) $\downarrow$ 1.0
w/o diversity	80.6(±1.1) $\downarrow$ 1.1	70.4(±1.2) $\downarrow$ 1.0	82.2(±1.4) $\downarrow$ 1.2	88.3(±0.7) $\downarrow$ 2.0
GCL-LSAA	**81.7**(±1.0)	**71.4**(±1.0)	**83.4**(±0.9)	**90.3**(±0.7)

across Cora, CiteSeer, and Computers datasets. The experiments reveal that on both Cora and Computers datasets, accuracy initially increases and then decreases as λ rises. For instance, on Cora, accuracy improves from 80.6 to 81.7 as λ increases from 0 to 0.2, but drops to 79.3 when λ reaches 1.0. This indicates that a moderate λ value effectively balances stability and diversity, enhancing model performance, while excessively high λ values may degrade performance by compromising stability.

The diversity loss threshold α also influences performance. On Cora, accuracy peaks at α=0.1, with higher α values leading to decreased performance. For CiteSeer, α=0.2 yields the best results, balancing diversity and stability. On Computers, α values of 0.1 and 0.2 perform best, with higher values reducing accuracy.

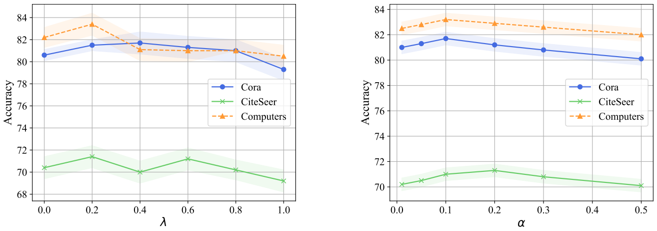


Fig. 5: Node Classification results of GCL-LSAA with different Trade-off coefficients and diversity loss threshold.

3. ABLATION STUDY

To verify the benefits of each component of GCL-LSAA, we conduct some ablation studies on four datasets. Based on the experimental results shown in Table 4, removing the P matrix (i.e., using S as C) results in a significant drop in accuracy on the CiteSeer dataset, from 71.4 to 68.3, indicating that the P matrix is crucial for performance. Similarly, omitting the S matrix (i.e., using P as C) leads to decreased performance on both the Cora and CiteSeer datasets, with a notable drop in CiteSeer from 71.4 to 69.0, demonstrating the importance of the S matrix. The removal of the stability loss results in a general decline in performance across all datasets, with the most significant drop on the Cora dataset, where accuracy decreases from 81.7 to 78.3, highlighting the critical role of stability loss in maintaining model performance. Although removing the diversity loss also leads to a decrease in performance, the reduction is relatively small, with some improvement observed on the Computers dataset, indicating that the impact of diversity loss is less pronounced. The complete model (GCL-LSAA) outperforms all other configurations across all datasets, achieving the best results particularly on Cora and Computers, thereby validating the effectiveness of the comprehensive strategy in node classification tasks.

4. REFERENCES

[1] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.

[2] Péter Mernyei and Cătălina Cangea, "Wiki-cs: A wikipedia-based benchmark for graph neural networks," *arXiv preprint arXiv:2007.02901*, 2020.

[3] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann, "Pitfalls of graph neural network evaluation," *arXiv preprint arXiv:1811.05868*, 2018.