Graph Contrastive Learning Reimagined: Exploring Universality

Jiaming Zhuo
Can Cui
jiaming.zhuo@outlook.com
594021820@qq.com
School of Artificial Intelligence
Hebei University of Technology
Tianjin, China

Kun Fu
Bingxin Niu
fukun@hebut.edu.cn
niubingxin666@163.com
School of Artificial Intelligence
Hebei University of Technology
Tianjin, China

Dongxiao He
hedongxiao@tju.edu.cn
College of Intelligence and Computing
Tianjin University
Tianjin, China

Chuan Wang*
wangchuan@iie.ac.cn
State Key Laboratory of Information Security
Institute of Information Engineering
Chinese Academy of Sciences
Beijing, China

Yuanfang Guo
andyguo@buaa.edu.cn
School of Computer Science and Engineering
Beihang University
Beijing, China

Zhen Wang
w-zhen@nwpu.edu.cn
OPtics and ElectroNics (iOPEN),
School of Cybersecurity
Northwestern Polytechnical University
Xi'an, China

Xiaochun Cao
caoxiaochun@mail.sysu.edu.cn
School of Cyber Science and Technology, Shenzhen Campus
Sun Yat-sen University
Shenzhen, China

Liang Yang*
yangliang@vip.qq.com
School of Artificial Intelligence
Hebei University of Technology
Tianjin, China

ABSTRACT

Real-world graphs exhibit diverse structures, including homophilic and heterophilic patterns, necessitating the development of a universal Graph Contrastive Learning (GCL) framework. Nonetheless, the existing GCLs, especially those with a local focus, lack universality due to the mismatch between the input graph structure and the homophily assumption for two primary components of GCLs. Firstly, the encoder, commonly Graph Convolution Network (GCN), operates as a low-pass filter, which assumes the input graph to be homophilic. This makes it challenging to aggregate features from neighbor nodes of the same class on heterophilic graphs. Secondly, the local positive sampling regards neighbor nodes as positive samples, which is inspired by the homophily assumption. This results in feature similarity amplification for the samples from the different classes (i.e., FALSE positive samples). Therefore, it is crucial to feed the encoder and positive sampling of GCLs with homophilic graph structures. This paper presents a novel GCL framework, named gRaph cOntraStive Exploring uNiversality (ROSEN), designed to achieve this objective. Specifically, ROSEN equips a local graph structure inference module, utilizing the Block Diagonal Property (BDP) of the affinity matrix extracted from node ego networks. This module can generate the homophilic graph structure by selectively removing disassortative edges. Extensive evaluations validate the effectiveness and universality of ROSEN across node classification and node clustering tasks.

CCS CONCEPTS

• **Computing methodologies** → **Unsupervised learning**; • **Networks** → **Network algorithms**.

KEYWORDS

Graph Neural Networks; Graph Self-Supervised Learning; Graph Contrastive Learning

ACM Reference Format:
Jiaming Zhuo, Can Cui, Kun Fu, Bingxin Niu, Dongxiao He, Chuan Wang, Yuanfang Guo, Zhen Wang, Xiaochun Cao, and Liang Yang. 2024. Graph Contrastive Learning Reimagined: Exploring Universality. In *Proceedings of the ACM Web Conference 2024 (WWW '24), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3589334.3645480

1 INTRODUCTION

Graph self-supervised learning (GSSL) stands out as a prominent technique in unsupervised learning on graphs [37], focusing on training models by extracting implicit self-supervised signals from graphs. As a representative GSSL, graph contrastive learning (GCL) devises a basic architecture, which learns invariant node representations via maximizing the agreement between embedding vectors

*Corresponding authors.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WWW '24, May 13–17, 2024, Singapore, Singapore
© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0171-9/24/05...$15.00
https://doi.org/10.1145/3589334.3645480

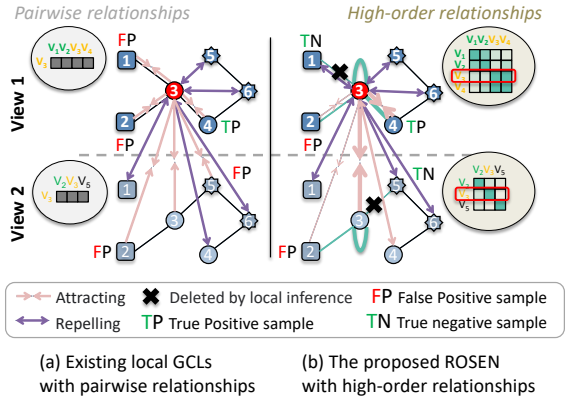


Figure 1: Comparison of existing local GCL with pairwise relationships and the proposed ROSEN with higher-order relationships. The thickness of the line indicates the weight.

from different perturbations of the same graph [18, 30, 42, 43]. Inheriting the success design and of contrastive learning in compute vision (CV) [4], most GCLs not only employ GNN encoders but mainly regard the same instance (node) in two augmented views (graphs) as the positive samples of each other (i.e., **Pairwise GCL**), and achieves outstanding performances on downstream tasks [30, 42]. Recently, GCLs have developed a graph-specific design based on homophily assumption, namely that the connected nodes tend to be the same class [1, 21]. To be specific, they propose to treat the neighbor nodes as the positive samples of target nodes (i.e., **Local GCL**), as shown in Figure 1 (a). This design has been verified empirically to boost the performance of baseline models on homophilic graphs [10, 18, 40].

Real-world graphs display diversity. They not only contain the mentioned homophilic graphs but also include heterophilic graphs such as the marriage network, where the connected nodes tend to be the different classes [39]. Therefore, such a diversity of graphs necessitates the development of universal GCLs. Unfortunately, the majority of GCLs, particularly local ones, fall short of meeting the above requirements for universality [1]. This shortfall could be attributed to a mismatch between the properties of graph structures and the homophily requirements for two key components of GCLs, namely the encoder and the positive term in contrastive loss.

On the one hand, the GNN encoder of GCLs, usually GCN [17], operates under the homophily assumption [1, 21]. This assumption implies that connected nodes are more likely to have similar features or belong to the same class. Therefore, these encoders presuppose that the input graph is homophilic. However, as topology augmentation is typically implemented through random strategies, such as random edge dropping [18, 42], the augmented graph structure hardly satisfies the above requirements. This discrepancy can lead to incorrect feature propagation during the encoding process. In addressing this challenge, several supervised GNNs with label-guided feature propagation have been introduced, such as CPGNN [41] and BM-GCN [9]. CPGNN propagates soft labels under the guidance of a compatibility matrix, while BM-GCN propagates features over a block matrix constructed using soft labels. However, the challenge remains in self-supervised settings. On the other hand, the optimization of the positive term in the contrastive loss (e.g., InfoNCE loss [31]), which involves maximizing the feature similarity of positive samples [36], also adheres to the homophily assumption. The local positive sampling regards all neighbor nodes of a given node as the positive samples for that node, which does not apply to heterophilic graphs. Consequently, node representations may lose their discriminative capacity.

To remedy the two drawbacks, this paper aims to devise a graph structure inference module to provide the homophilic graph structure for encoding and positive sampling. The intuitive idea is to selectively remove edges that connect the nodes from different classes in the input graph (i.e., disassortative edges). The primary challenge is to determine the criteria for selection without node labels. To address this challenge, an analysis is conducted on the properties of the affinity matrix extracted from the node ego network, which contains the node and its one-hop neighborhoods. Ideally, within the ego network of each node, the affinity matrix describing the relationships among these nodes is expected to adhere to the Block Diagonal Property (BDP). This implies that diagonal blocks, representing relationships between nodes from the same class, should exhibit non-zero values, while off-diagonal elements, denoting relationships between nodes from different classes, should be zero. The local BDP serves as a basis for selectively removing disassortative edges.

In light of the analysis, a universal GCL framework with a local inference module of graph structures, named gRaph cOntraStive Exploring uNiversality (ROSEN), is proposed. To be specific, inspired by the Block Diagonal Representation (BDR) [19], the local inference module is designed to calculate the local affinity matrices on the ego networks by optimizing a self-expressive learning objective with soft diagonal block regularization. Furthermore, to facilitate the low-noise node features for graph structure inference and the robust graph structures for local contrasting learning, the local self-expressive learning objective and contrastive learning objective are optimized via the alternative optimization strategy. Theoretically, it is proven that ROSEN can be formulated as an Expectation Maximization (EM) based algorithm. The iterative steps of structure inferencing and contrastive learning can be elucidated as approximating and maximizing the log-likelihood function.

The main contributions of this study are summarized as follows.

- We investigate the local block diagonal properties of the affinity matrix on node ego networks.
- We introduce a universal GCL framework with a local graph structure inference module, named gRaph cOntraStive Exploring uNiversality (ROSEN).
- We theoretically prove that the proposed ROSEN follows the EM algorithm.
- We extensively evaluate the effectiveness and universality of ROSEN on node classification and node clustering tasks.

2 PRELIMINARIES

This section first introduces the notations used in this paper. Next, it elucidates the basic concept of graph contrastive learning (GCL).

[1]Universality means that models are applied to both homophilic and heterophilic graphs.

2.1 Notations

Let $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{X})$ represents an undirected attribute graph, where $\mathcal{V}$ denotes the node set and $\mathcal{E}$ terms the edge set. $\mathbf{X} \in \mathbb{R}^{N \times F}$ stands for the node attribute matrix, where N and F are the numbers of nodes and attributes, respectively. The adjacency matrix is denoted by $\mathbf{A} \in \mathbb{R}^{N \times N}$. The normalization form of the adjacency matrix is commonly utilized [17, 38], namely $\tilde{\mathbf{A}} = (\mathbf{D}+\mathbf{I})^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})(\mathbf{D}+\mathbf{I})^{-\frac{1}{2}}$, where $\mathbf{D}$ denotes the diagonal degree matrix. According to the edge set $\mathcal{V}$, the neighbor node set of each node can be obtained (e.g., $N(v)$ of node v). The node labels are denoted by $\mathbf{Y} \in \mathbb{R}^{N \times C}$, which are exclusively employed in fine-tuning the classifier parameters on downstream tasks, such as node classification.

2.2 Graph Contrastive Learning

GNN Encoder. To produce informative node representations, the raw attribute is projected by a GNN encoder (typically Graph Convolution Network (GCN) [17]). For graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{X})$, the encoding process can be formulated as

$$\mathrm{GCN}(\tilde{\mathbf{A}}, \mathbf{H}^{(l)}) : \mathbf{H}^{(l+1)} = \sigma(\tilde{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}), \tag{1}$$

where $\mathbf{H}^{(0)} = \mathbf{X}$ denotes the initial node representation, and $\sigma(\cdot)$ terms the nonlinear activation function (e.g., $\mathrm{ReLU}(\cdot) = \max(0, \cdot)$), and $\mathbf{W}^{(l)}$ represents the parameter matrix at layer l. As a result, by encoding nodes via GNNs, node representations would capture the structural relationships and local patterns on graphs.

Pairwise and Local Contrastive Losses. Based on the defined positive and negative sample sets, contrastive loss is implemented as the distance minimization between the positive pairs alongside the distance maximization between the negative pairs. As exemplified by a node-level GCL scheme, a widely used InfoNCE [31] loss $\mathcal{L}_{info}$ can be expressed as

$$\begin{aligned} \mathcal{L}_{info} &= -\frac{1}{|V|}\sum_{v \in V} log \frac{\mathrm{pos}(v)}{\mathrm{pos}(v) + \mathrm{neg}(v)}, \\ \mathrm{pos}(v) &= \sum_{v^+ \in \mathcal{P}_v} e^{\theta(\mathbf{h}_v, \mathbf{h}_{v^+})/\tau}, \quad \mathrm{neg}(v) = \sum_{v^- \in \mathcal{N}_v} e^{\theta(\mathbf{h}_v, \mathbf{h}_{v^-})/\tau}, \end{aligned} \tag{2}$$

where θ represents the cosine similarity. τ is a temperature coefficient, and a smaller one helps form a more uniform representation space. $\mathcal{P}_v$ and $\mathcal{N}_v$ denote the positive and negative sample set of node v, respectively. Generally, the positive sampling strategies of GCLs are two categories: pairwise and local positive sampling. The former strategy leverages the same node in another graph (view) [42], while the latter involves appending nodes with similar semantics from node neighborhoods [10, 18]. These strategies can be formulated as

$$Pairwise \quad \mathcal{P}_v = \{u | u = \tilde{v}\}, \tag{3}$$

$$Local \quad \mathcal{P}_v = \{u | u \in N(v) \ and \ u \in \{N(\tilde{v}) \cup \tilde{v}\}\}, \tag{4}$$

where $\tilde{v}$ represents the corresponding node of v in another graph. Typically, the negative sample set consists of all remaining nodes, i.e., $\mathcal{N}_v = \{\{V \cup \tilde{V}\} \setminus \{\mathcal{P}_v \cup v\}\}$. Compared to the pairwise GCLs, which relies on pariwise positive sampling, the local GCLs enhance the depiction of homophily [1, 21]. Homophily can be viewed a reliable indicator of encoding and positive sampling in self-supervised scenarios [10].

3 METHODOLOGY

This section starts by elucidating the motivation behind graph structure inference on node neighborhoods. Subsequently, a novel Graph Contrastive Learning (GCL) framework with a local inference module of graph structures, named ROSEN, is introduced. Finally, an analysis of the efficiency of ROSEN is presented.

3.1 Analysis and Motivation

Existing GCLs, particularly local ones, tend to be not universal for homophilic and heterophilic graphs [18, 40]. This failure could result from the mismatch between the properties of graph structures and the requirement for two key components of local GCLs, i.e., encoder and positive term in contrastive loss.

On the one hand, most GNN encoders of GCLs, especially the widely used GCN [17], follow homophily assumption [1, 21]. Thus, they require the input graph to be homophilic. However, since the topology augmentation is often implemented according to random strategies, such as random edge dropping [18, 42], the augmented graph structure hardly satisfies the above requirements. On the other hand, the positive term in contrastive loss, which is formulated in Equation 2, also follows the homophily assumption, as analyzed in the introduction. Based on the input graph, the local positive sampling regards all neighbor nodes of a node as the positive samples of this node, as described in Equation 4, which again does not meet the above requirement. Therefore, node representations will lose themself discrimination.

A solution to this problem of local GCLs is to provide the model with a fully homophilic graph structure by dropping edges that connect the nodes of different classes (i.e., disassortative edges). So that GCLs can perform feature propagation and positive sampling among nodes of the same class in node neighborhoods. However, this presents a challenging issue, as the node labels are unknown in self-supervised learning scenarios. To address this challenge, the paper explores the characteristics of affinity matrices within node neighborhoods to selectively drop the disassortative edges.

Definition 3.1. **Block Diagonal Property (BDP)** [11]. Given a square matrix, if it can be decomposed into small block matrices, where each block is denoted by non-zero elements on the principal diagonal while non-diagonal elements are zero, it obeys the BDP.

On the ego network of each node, which comprises the node and its one-hop neighbors, the affinity matrix describing the relationships among these nodes should obey the BDP. That is, diagonal blocks that denote the relationship between nodes of the same class is nonzero while the off-diagonal elements that represent the relationship between nodes from different classes is zero. Therefore, the homophilic graph structure can be obtained by pursuing the block diagonal affinity matrix on node neighborhoods.

3.2 ROSEN Framework

According to the analysis presented previously, this paper proposes a novel universal GCL framework with a local inference module of graph structures, called gRaph cOntraStive Exploring uNiversality (ROSEN). The idea of the local inference module is to generate homophilic graphs for GCLs by removing edges connecting nodes of different classes in the input graphs, as depicted in Figure 2.

This local inference module consists of three primary components: ego network extraction, affinity matrix calculation, and graph reconstruction. Once reconstructed graphs are obtained, they are applied to the GNN encoder and localized contrastive loss to facilitate learning discriminative node representations.

3.2.1 Local Graph Structure Inference Module. Inspired by the classic Block Diagonal Representation (BDR) [19], which is widely employed in subspace clustering, this module first calculates the local affinity matrices by optimizing a self-expressive learning objective with soft diagonal block regularization on local feature space. This can be formulated as

$$\begin{aligned} &\min_{\mathbf{B}_v} \frac{1}{2} \|\mathbf{H}_v - \mathbf{B}_v\mathbf{H}_v\|^2 + \gamma \|\mathbf{B}_v\|_{\underline{k}}, \\ &\text{s.t. } \operatorname{diag}(\mathbf{B}_v) = 0,\ \mathbf{B}_v \geq 0,\ \mathbf{B}_v = \mathbf{B}_v^T, \end{aligned} \tag{5}$$

where $\mathbf{H}_v \in \mathbb{R}^{(|N(v)|+1)\times D}$ represents features for the ego network of node v and $N(v)$ denotes the number of one-hop neighbor nodes and D terms the dimension of features. $\mathbf{B}_v$ stands for the local affinity matrix, and $\|\mathbf{B}_v\|_{\underline{k}} = \sum_{i=0}^{k-1} \lambda_i \left(\mathbf{L}_{\mathbf{B}_v}\right)$ terms the sum of the smallest k eigenvalue of $\mathbf{L}_{\mathbf{B}_v}$, where $\mathbf{L}_{\mathbf{B}_v}$ denotes the laplacian matrix of $\mathbf{B}_v$ and the eigenvalues are listed in ascending order. γ is a hyperparameter for a tradeoff between two terms. Based on spectral graph theory [5], the two terms of Equation 5 guarantee that matrix $\mathbf{B}_v$ is self-expressive and has k connected subgraphs (i.e., blocks). In addition, three constraints of the matrix are considered: no self-loop, nonnegative and symmetric.

However, directly applying the three constraints on $\mathbf{B}_v$ limits its expressive power. To alleviate this issue, the module seeks to introduce an approximation term. Thus, Equation 5 can be rewritten as

$$\begin{aligned} &\min_{\mathbf{Z}_v, \mathbf{B}_v} \frac{1}{2} \|\mathbf{H}_v - \mathbf{Z}_v\mathbf{H}_v\|^2 + \frac{\lambda}{2} \|\mathbf{Z}_v - \mathbf{B}_v\|^2 + \gamma \|\mathbf{B}_v\|_{\underline{k}}, \\ &\text{s.t. } \operatorname{diag}(\mathbf{B}_v) = 0,\ \mathbf{B}_v \geq 0,\ \mathbf{B}_v = \mathbf{B}_v^T, \end{aligned} \tag{6}$$

where $\mathbf{Z}_v$ stands for the affinity matrix that approximates $\mathbf{B}_v$ and λ denotes a hyperparameter.

Solution. The above objective function can be optimized via alternating minimization solver [35], namely, updating one while fixing the other. After the solver converges, the generated affinity matrices are applied to the GNN encoder for feature propagation and to the localized contrastive loss for positive sampling. For ease of presentation, $\mathbf{Z}_v$ and $\mathbf{B}_v$ are abbreviated as $\mathbf{Z}$ and $\mathbf{B}$. Note that due to $\|\mathbf{B}\|_{\underline{k}}$ is nonconvex for which the optimization is NP-hard, it needs to be converted into a convex program [2] according to

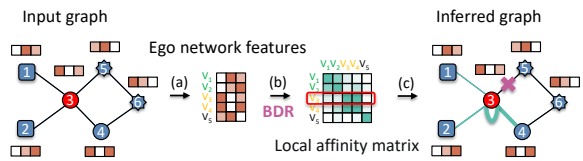
$$\|\mathbf{B}\|_{\underline{k}} = \min_{\mathbf{W}} \langle \mathbf{L}_{\mathbf{B}}, \mathbf{W} \rangle, \text{ s.t. } 0 \preceq \mathbf{W} \preceq \mathbf{I}, \operatorname{Tr}(\mathbf{W}) = k, \tag{7}$$

where $\mathbf{W} \in \mathbb{R}^{(|N(v)|+1)\times(|N(v)|+1)}$ denotes a newly added variable block. Therefore, the overall objective of the proposed local graph structure inference module can be expressed as

$$\begin{aligned} &\min_{\mathbf{Z},\mathbf{B},\mathbf{W}} \frac{1}{2}\|\mathbf{H}_v - \mathbf{Z}\mathbf{H}_v\|^2 + \frac{\lambda}{2}\|\mathbf{Z} - \mathbf{B}\|^2 + \gamma\langle \mathbf{L}_{\mathbf{B}}, \mathbf{W}\rangle, \\ &\text{s.t. } \operatorname{diag}(\mathbf{B}) = 0, \mathbf{B} \geq 0, \mathbf{B} = \mathbf{B}^\top, 0 \preceq \mathbf{W} \preceq \mathbf{I}, \operatorname{Tr}(\mathbf{W}) = k, \end{aligned} \tag{8}$$

The objective function is split into three problems and solved independently. Please refer to the appendix for the solving process.

After the affinity matrix of each node is obtained, the row vector corresponding to this node is selected as the edge weights between the node and its neighbor nodes, as shown in Figure 2. Based on it, the weights of all edges in the input graph can be obtained. In this process, some edges that connects nodes from different subspaces could be removed since the affinity values being 0.



(a) Extracting ego network. (b) Calculating affinity matrix. (c) Reconstructing graph.

Figure 2: Workflow of the proposed local graph structure inference module. Nodes are sorted by classes. Edges that connect the nodes of the same class tend to be kept.

Moreover, to ensure that the relationship between the node pairs are reciprocal, it is necessary to make the adjacency matrix of the graph nonnegative and symmetric. The operations on the adjacency matrix $\mathbf{S}$ can be formulated as

$$\mathbf{S} = \left(|\mathbf{A}_{\mathbf{B}}| + |\mathbf{A}_{\mathbf{B}}{}^\top|\right)/2 \quad \text{or} \quad \mathbf{S} = \left(|\mathbf{A}_{\mathbf{Z}}| + |\mathbf{A}_{\mathbf{Z}}{}^\top|\right)/2, \tag{9}$$

where $\mathbf{A}_{\mathbf{B}}$ and $\mathbf{A}_{\mathbf{Z}}$ denote the adjacency matrices corresponding to $\mathbf{B}$ and $\mathbf{Z}$, respectively. Furthermore, to ease the computational burden and preserve the confident relationship, sparsification operation is applied on $\mathbf{S}$, i.e., zeroing out values less than the threshold β. This can be expressed as

$$\mathbf{S}_{ij} = \begin{cases} 0, & \text{if } \mathbf{S}_{i,j} < \beta, \\ \mathbf{S}_{i,j}, & \text{otherwise.} \end{cases} \tag{10}$$

Benefits. The proposed local structure inference module provides several advantages to GCLs. (1) **Robust topology augmentation**. Firstly, the structure inference is equivalent to edge deletion, which is widely utilized as topology augmentation. As described in Figure 2, some edges in the initial graph may be removed through affinity matrix estimation and matrix sparsification. Secondly, compared to random edge deletion, the edge deletion resulting from the structure inference can improve the robustness of GCLs. Note that the inference process, which is based on feature self-expressive assumption, always keeps edges that connect two nodes of the same class. (2) **Higher-order information**. Given that the inferred edge weights actually characterize higher-order relationships between local neighbors, GCLs can benefit from exploring macro community structure rather than fragile pairwise relationships. Besides, this module performs parallel training on nodes, which guarantees its scalability.

3.2.2 Contrastive Learning. After the graph structure is modified, the discussion follows on how to improve the universality of GCLs using this graph structure. Based on the previous analysis, the quality of node representation can be enhanced by performing the encoding and local positive sampling using this graph structure. Note that, the primary module of ROSEN is orthogonal to GCL work on graph augmentation and encoder designs, thus ROSEN is extensible to most GCLs. For convenience, the implementation of ROSEN refers to the configuration of GRACE [42] in this paper.

Firstly, the modified graph structure is employed in a two-layer GCN encoder to guide the feature propagation, namely

$$\mathbf{H} = \sigma\left((\mathbf{S} \odot \tilde{\mathbf{A}}) \cdot \sigma\left((\mathbf{S} \odot \tilde{\mathbf{A}}) \cdot \mathbf{X} \cdot \mathbf{W}^{(0)}\right) \cdot \mathbf{W}^{(1)}\right), \tag{11}$$

Table 1: Accuracy in percentage (mean ± std) of node classification for six homophilic graphs. The Best and Runner-up of unsupervised models are bolded and underlined, respectively. The second column presents the data considered in training.

Model	Training Data	Cora	CiteSeer	PubMed	Wiki-CS	Computers	Photo
GCN	A, X, Y	85.77 ± 0.25	73.68 ± 0.31	88.13 ± 0.28	76.89 ± 0.37	86.34 ± 0.48	92.35 ± 0.25
GAT	A, X, Y	86.37 ± 0.30	74.32 ± 0.27	87.62 ± 0.26	77.42 ± 0.19	87.06 ± 0.35	92.64 ± 0.42
JKNet	A, X, Y	85.93 ± 1.35	74.37 ± 1.53	87.68 ± 0.30	79.52 ± 0.21	85.28 ± 0.72	92.68 ± 0.13
DeepWalk	A	73.96 ± 0.12	61.91 ± 0.42	74.79 ± 0.98	74.35 ± 0.06	85.68 ± 0.06	89.44 ± 0.11
Node2Vec	A	75.87 ± 0.22	62.54 ± 0.13	76.49 ± 0.32	71.79 ± 0.05	84.39 ± 0.08	89.67 ± 0.12
GAE	A, X	76.83 ± 1.22	65.43 ± 1.13	76.52 ± 0.33	70.15 ± 0.01	85.27 ± 0.19	91.62 ± 0.13
VGAE	A, X	79.36 ± 0.83	69.18 ± 0.27	79.17 ± 0.44	76.63 ± 0.19	86.37 ± 0.21	92.20 ± 0.11
GraphMAE	A, X	87.31 ± 1.01	73.47 ± 0.32	84.83 ± 0.53	79.49 ± 0.11	88.83 ± 0.25	93.07 ± 0.55
DGI	A, X	85.90 ± 0.57	72.57 ± 0.23	83.52 ± 1.24	75.73 ± 0.13	84.09 ± 0.39	91.49 ± 0.25
MVGRL	A, X	86.77 ± 0.33	73.71 ± 0.48	84.63 ± 0.73	77.97 ± 0.18	87.09 ± 0.27	92.01 ± 0.13
GRACE	A, X	84.79 ± 0.64	72.94 ± 0.72	84.51 ± 0.68	79.16 ± 0.36	87.21 ± 0.44	92.65 ± 0.32
GCA	A, X	85.16 ± 0.51	72.73 ± 0.45	85.22 ± 0.73	79.35 ± 0.12	87.84 ± 0.27	92.78 ± 0.17
BGRL	A, X	85.37 ± 0.74	73.45 ± 0.83	84.61 ± 0.32	78.74 ± 0.22	88.92 ± 0.33	93.24 ± 0.29
LOCAL-GCL	A, X	86.27 ± 0.91	73.27 ± 0.18	85.01 ± 0.48	79.18 ± 0.53	88.72 ± 0.42	93.15 ± 0.47
HGRL	A, X	85.85 ± 0.73	72.06 ± 0.75	84.44 ± 0.63	78.97 ± 0.83	87.59 ± 0.47	92.33 ± 0.82
SP-GCL	A, X	86.07 ± 0.83	73.39 ± 0.64	84.17 ± 0.81	79.21 ± 0.94	88.25 ± 0.33	92.57 ± 0.93
HomoGCL	A, X	85.02 ± 0.68	73.67 ± 0.78	82.33 ± 0.49	77.47 ± 0.45	87.84 ± 0.28	93.59 ± 0.27
ROSEN	A, X	**87.72 ± 1.00**	**74.13 ± 0.68**	**85.30 ± 0.72**	**80.17 ± 1.28**	**89.03 ± 0.41**	**93.90 ± 1.10**

where $\odot$ stands for Hadamard product.

Next, ROSEN proposes to leverage the modified graph structure to guide the positive sampling on node neighborhoods. To be specific, given the node features $\mathbf{H}$ and graph structure $\mathbf{S}$, ROSEN devises the objective function for node v as

$$\begin{aligned} \mathcal{L}_{rosen} &= -\frac{1}{|V|}\sum_{v\in V} log\frac{\text{pos}(v)}{\text{pos}(v)+\text{neg}(v)}, \\ \text{pos}(v) &= \sum_{v^+\in N^S(v)} \mathbf{S}_{v,v^+} * e^{\theta(\mathbf{h}_v,\mathbf{h}_{v^+})/\tau}, \\ \text{neg}(v) &= \sum_{v^-\in V\setminus N^S(v)} e^{\theta(\mathbf{h}_v,\mathbf{h}_{v^-})/\tau}, \end{aligned} \tag{12}$$

where $N^S(v)$ denotes the neighbor node set of node v on matrix $\mathbf{S}$. In contrast to existing local GCLs which blindly regard all neighbor nodes as positive samples [40], ROSEN elaborately selects the reliable neighbor nodes as positive samples through the local structure inference module.

Considering that GCN essentially can serve as a denoising encoder [20], which satisfies the needs of low-noise features for the proposed local inference module, performing on the feature space is a logical solution. To provide expressive encoded features, ROSEN presents the alternating optimization strategy of encoder parameters and graph structure, as described in Algorithm 1.

Theorem 3.2. *Let Θ, k and $\mathbb{1}\mathbf{G}$ denote the parameters of the GNN encoder, the number of subspaces (blocks), and the subspace indicator, respectively, the proposed ROSEN follows Expectation-Maximization (EM) algorithm, where the structure inference and the maximization of the lower bound on the mutual information of representations for the contrastive pairs are equivalent to E-step and M-step, respectively.*

Please refer to the appendix for the proof.

3.3 Complexity Analysis

This section analyzes the time complexity of the proposed ROSEN, which is based on the baseline GRACE. It is worth noting that ROSEN introduces light computational overhead over GRACE.

The overall complexity of ROSEN is $O(|\mathcal{E}|(F+D)+N^2D+N(PF+P^3))$, which comes from three components: encoding, loss calculation, and graph structure inference. P terms the average size of ego networks. The complexity of the first two components is the same as that of GRACE. Specifically, since the backbone for encoding is a two-layer GCN, the complexities of the two-layer calculations is $O(EF+ED)$. Moreover, before calculating loss, node features are dimensionally reduced through the projection head, which is a two-layer MLP. The complexity of the two-step projection is $O(ND^2)$. Besides, the calculation of contrastive loss takes $O(N^2D)$ time due to a quadratic all-pairs contrast at each update step. Thus, the complexity of GRACE is $O(|\mathcal{E}|(F+D)+N^2D)$.

The additional complexity of ROSEN arises from the inference of graph structure. Specifically, for each node, ROSEN extracts an ego network, optimizes the objective function outlined in Equation 8, and constructs a new edge set for the entire graph. For each node, the complexities of ego network extraction, optimization, and edge set construction are $O(PF)$, $O(P^3)$, $O(P^3)$, and $O(P^3)$, and $O(P)$, respectively. Thus, the complexity for each ego network is $O(PF+P^3)$. As a result, the overall complexity across all nodes is $O(N(PF+P^3))$. It is noteworthy that the number of iterations is ignored since the maximum value is set. As the additional complexity is linear with the network size, ROSEN incurs light computational overhead over GRACE.

4 EXPERIMENTS

This section begins by introducing the fundamental setup of the experiment, including datasets, baseline models, and configurations.

Table 2: Accuracy in percentage (mean ± std) of node classification for six heterophilic graphs. The Best and Runner-up of unsupervised models are bolded and underlined, respectively. The second column presents the data considered in training.

Model	Training Data	Cornell	Texas	Wisconsin	Chameleon	Squirrel	Actor
GCN	A, X, Y	55.14 ± 7.57	55.68 ± 9.61	58.42 ± 5.10	59.82 ± 2.58	36.89 ± 1.34	30.64 ± 1.49
GAT	A, X, Y	58.92 ± 3.32	58.38 ± 4.45	55.29 ± 8.71	60.26 ± 2.50	40.72 ± 1.55	27.44 ± 0.89
JKNet	A, X, Y	56.49 ± 3.22	65.35 ± 4.86	51.37 ± 3.21	60.31 ± 2.76	44.24 ± 2.11	36.47 ± 0.51
DeepWalk	A	39.18 ± 5.57	46.49 ± 6.49	33.53 ± 4.92	47.74 ± 2.05	32.93 ± 1.58	22.78 ± 0.64
Node2Vec	A	42.94 ± 7.46	41.92 ± 7.76	37.45 ± 7.09	41.93 ± 3.29	22.84 ± 0.72	28.28 ± 1.27
GAE	A, X	58.85 ± 3.21	58.64 ± 4.53	52.55 ± 3.80	33.84 ± 2.77	28.03 ± 1.61	28.03 ± 1.18
VGAE	A, X	59.19 ± 4.09	59.20 ± 4.26	56.67 ± 5.51	35.22 ± 2.71	29.48 ± 1.48	26.99 ± 1.56
GraphMAE	A, X	59.32 ± 4.15	60.17 ± 5.32	56.45 ± 5.33	50.13 ± 2.11	38.03 ± 1.23	29.88 ± 1.05
DGI	A, X	63.35 ± 4.61	60.59 ± 7.56	55.41 ± 5.96	39.95 ± 1.75	31.80 ± 0.77	29.82 ± 0.69
MVGRL	A, X	64.30 ± 5.43	62.38 ± 5.61	62.37 ± 4.32	51.07 ± 2.68	35.47 ± 1.29	30.02 ± 0.70
GRACE	A, X	54.86 ± 6.95	57.57 ± 5.68	50.00 ± 5.83	48.05 ± 1.81	31.33 ± 1.22	29.01 ± 0.78
GCA	A, X	55.41 ± 4.56	59.46 ± 6.16	50.78 ± 4.06	49.80 ± 1.81	35.50 ± 0.91	29.65 ± 1.47
BGRL	A, X	57.30 ± 5.51	59.19 ± 5.85	52.35 ± 4.12	47.46 ± 2.74	32.64 ± 0.78	29.86 ± 0.75
LOCAL-GCL	A, X	53.31 ± 1.87	62.19 ± 2.38	64.98 ± 1.32	**59.27 ± 3.37**	**49.31 ± 2.81**	32.39 ± 1.48
HGRL	A, X	74.34 ± 5.13	73.66 ± 6.92	77.16 ± 4.62	48.58 ± 2.46	37.81 ± 1.54	32.87 ± 0.98
SP-GCL	A, X	74.96 ± 5.19	73.07 ± 5.28	76.87 ± 5.47	51.17 ± 1.97	38.79 ± 1.57	31.87 ± 0.92
HomoGCL	A, X	48.64 ± 2.59	54.05 ± 2.32	39.21 ± 5.75	48.68 ± 1.16	38.71 ± 0.85	28.81 ± 0.78
ROSEN	A, X	**76.49 ± 6.84**	**74.86 ± 6.29**	**78.63 ± 4.68**	49.25 ± 2.33	39.13 ± 1.36	**33.19 ± 0.81**

Table 3: Overall performance of node clustering measured by ACC, NMI, and ARI scores in percentage. The best results are in bold, and the second-best results are underlined.

	Cora			CiteSeer		
	ACC	NMI	ARI	ACC	NMI	ARI
K-Means	35.78	16.88	8.30	44.47	21.35	17.43
GRACE	64.02	36.17	22.69	53.65	27.62	25.14
BGRL	61.84	40.39	24.29	52.52	15.4	14.17
MVGRL	72.45	55.05	41.55	64.14	39.12	38.93
ROSEN	**76.08**	**58.53**	**46.50**	**66.22**	**40.34**	**40.17**

Then, it comprehensively assesses the effectiveness of the proposed ROSEN by experimentally validating the performances on two downstream tasks, i.e., node classification and clustering. Finally, it performs several additional experiments to provide an intuitive understanding of the performance improvements, including the effectiveness study, ablation study, hyperparameter study, robustness study, and scalability study.

4.1 Experimental Setup

4.1.1 Datasets. In the experiments, fifteen publicly available graph datasets are employed in the experiment, which consists of twelve small graphs and three large graphs. Firstly, according to whether the Edge Homophily [23] is more significant than 0.5, the samll graph datasets can be divided into two categories: homophilic graphs (i.e., Cora, CiteSeer, PubMed, Wiki-CS, Computers, and Photo) and heterophilic graphs (i.e., Cornell, Texas, Wisconsin, Chameleon, Squirrel, and Actor). Secondly, three large graphs are Ogbn-Arxiv, Ogbn-Products, MAG-Scholar-F. The statistics of these datasets are shown in Table 5 and Table 6 in the Appendix.

For a fair comparison, the graph datasets are split according to broadly adopted schemes. Specifically, for three homophilic graphs (Cora, CiteSeer, and Pubmed) and all six heterophilic graphs, the partition provided by Geom-GCN [23] is used, where the nodes for training, validation, and testing are 48%, 32%, and 20% of all nodes, respectively. In addition, for the remaining three homophilic graphs (Wiki-CS, Computers, and Photo), we randomly split the training, validation, and test sets into 10%, 10%, and 80% of all nodes [30]. Moreover, we employ the official splits in [14] for Ogbn-Arxiv and Ogbn-Products. And, as for MAG-Scholar-F, we randomly split 5%/5%/40% nodes for training/validation/testing, respectively [12].

4.1.2 Baseline Models. The baseline models include the following four categories: three semi-supervised GNN (including GCN [17], GAT [32], and JKNet [38]), three unsupervised models (including K-Means [7], DeepWalk [24], and Node2Vec [6]), three graph generative models (GAE [16], VGAE [16], and GraphMAE [13]), and nine graph contrastive models (including DGI [33], MVGRL [8], GRACE [42], GCA [43], BGRL [30], LOCAL-GCL [40], HGRL [3], SP-GCL [34], and HomoGCL [18]). Please refer to appendix for the introduction of these models.

4.1.3 Configurations. The experiment is performed on a Linux machine with GeForce RTX4090 24GB GPU and a Linux machine with four NVIDIA A800 80GB GPUs. The results are reported as an average of ten random runs. To make a fair comparison, the experiment works with the configuration reported in the original paper for all baseline models except for the embedding dimension is set to 64. Note that thanks to the open PyTorch libraries: PyG[1] and PyGCL[2], the reproduction of all baseline models is facilitated. The proposed ROSEN follows the baseline model GRACE [42], where the GNN

[1]https://www.pyg.org/
[2]https://github.com/PyGCL

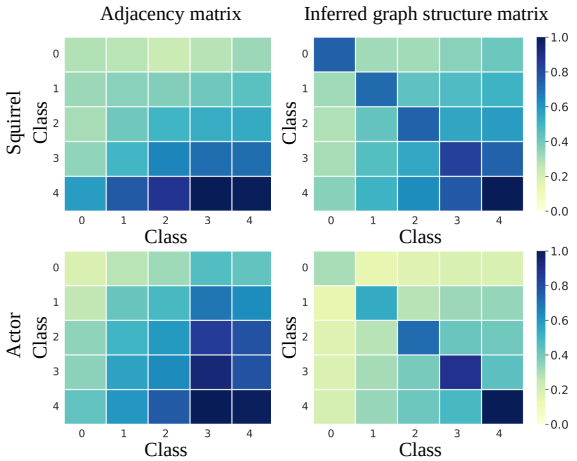


Figure 3: Visualization of the adjacency matrix and inferred structure matrices. The block value denotes the normalized edge weights. The inferred structure matrix is more similar to the block diagonal matrix.

encoder is a two-layer GCN [17] with the dimension is 64. Moreover, the graph augmentations involve attribute masking and edge dropping, which have the ratio $\{0.2, 0.4, 0.6, 0.8\}$. The temperature coefficient of the contrastive loss is taken from $\{0.1, 0.3, 0.5, 0.7, 1, 2\}$. In addition, the network optimizer used in network training is Adam optimizer [15], the learning rate is taken out of $\{0.001, 0.01\}$ and the weight decay rate is chosen in $\{0, 1 \times 10^{-3}, 1 \times 10^{-4}, 1 \times 10^{-5}\}$. For the hyperparameters introduced by ROSEN, λ, γ, and tol are chosen among $\{40, 50, 60\}$,$\{0.5, 1, 2\}$, and $\{1 \times 10^{-3}, 1 \times 10^{-4}\}$, respectively, and the number of subspaces (blocks) is picked from a range no more significant than the number of classes, and its impact on the performance is shown in Section 4.5.

4.2 Results and Analysis

Homophilic Graphs. Table 1 presents the node classification performance of all models on the homophilous graph. First, the proposed ROSEN framework achieves the best classification performance on all datasets compared to all unsupervised baselines. For example, ROSEN outperforms the second-ranked GraphMAE on Cora by 0.41%, which demonstrates the superiority of ROSEN in exploiting self-supervised information. Second, even in comparison to the supervised baseline methods, the proposed ROSEN framework obtains the highest classification accuracy on five datasets except CiteSeer and PubMed. In particular, ROSEN outperforms its backbone GCN by 3.28% on Wiki-CS and by 2.69% on Computers. This illustrates the potential of ROSEN as an unsupervised learning scheme for the base model. Third, compared to the baseline GRACE, which employs the same configurations (i.e., encoder and augmentation), ROSEN achieves 2.93% and 1.82% performance improvement on Cora and Computers, respectively. This can be attributed to the proposed graph structure inference module, through which more TRUE positive samples can be obtained. **Heterophilic Graphs.** It is observed from Figure 2 that as compared to all unsupervised baselines, ROSEN achieves the best performance on four heterophilous graphs. Specifically, ROSEN outperforms the second-place SP-GCL by 1.53% on Cornell and the second-place HGRL by 1.47% on Wisconsin, respectively, which demonstrates its superiority on heterophilous graphs. Moreover, ROSEN has consistent performance gains over the baseline GARCE on all datasets, which illustrates the effectiveness of the proposed structure inference module to extract TRUE positive samples. Furthermore, the proposed ROSEN remarkably achieves performance improvement over the backbone GCN. The only exception is on Chameleon, where ROSEN slightly underperforms GCN yet still achieves a comparable result. It should be noted that unsupervised models generally fail to perform well on Chameleon and Squirrel datasets. One possible reason is that the uniform representations induced from the optimization for the negative sample pairs do not match the neighborhood overlap property of the two graphs [25].

Table 4: Impact of the inferenced graph structure used in different components.

	Cora	CiteSeer	Texas	Actor
ROSEN	$87.72_{\pm 1.00}$	$74.13_{\pm 0.68}$	$74.68_{\pm 6.29}$	$33.19_{\pm 0.81}$
w/o in encoder	$85.77_{\pm 1.49}$	$73.94_{\pm 0.55}$	$63.24_{\pm 4.67}$	$29.93_{\pm 0.69}$
w/o in contrastive loss	$85.63_{\pm 1.57}$	$73.79_{\pm 0.47}$	$61.72_{\pm 4.97}$	$29.61_{\pm 0.52}$
local GCL	$84.65_{\pm 1.18}$	$73.68_{\pm 0.63}$	$59.27_{\pm 5.91}$	$29.02_{\pm 0.70}$

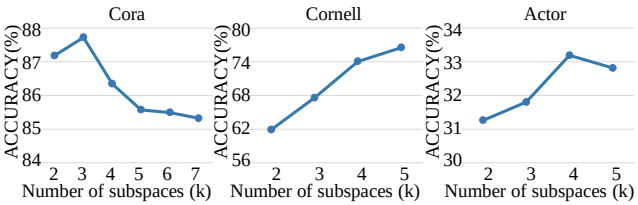


Figure 4: Impact of the number of subspaces (blocks) on the node classification performance.

Node Clustering. To assess the discrimination of the obtained node embeddings, this section performs the node clustering using these embeddings on Cora and CiteSeer datasets. The node clustering task is conducted with the help of K-Means. It can be observed from Table 3 that, compared to the unsupervised models K-Means, which does not employ graph structure, the unsupervised models with graph structure show performance advantages on all datasets. This demonstrates that considering graph structure in the model design enables GCL methods to generate more discriminative representations. What's more, compared to baseline GCL models, ROSEN generates clustering-friendly embeddings. For example, ROSEN outperforms the runner-up MVGRL by at least 1% on all datasets, which highlights the representation ability of ROSEN.

4.3 Effectiveness Analysis

As discussed in the previous section, this paper aims to create homophilic graphs by constraining the corresponding matrices to satisfy the BDP. To intuitive explain the effectiveness of the proposed local graph structure inference module, the edge distributions of the input adjacency matrices and the inferred graph structure matrices of Squirrel and Actor datasets are visualized, as depicted in Figure 3. The inferred graph structures are selected from $\mathbf{A_Z}$ and $\mathbf{A_B}$ according to the reported results in Table 2.

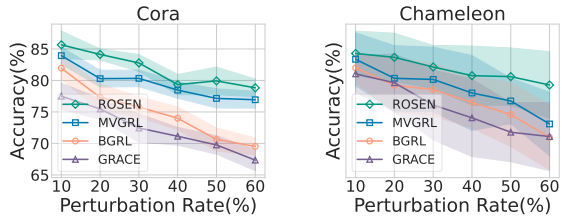


Figure 5: Performance variation of GCL models on graph data with topology noise.

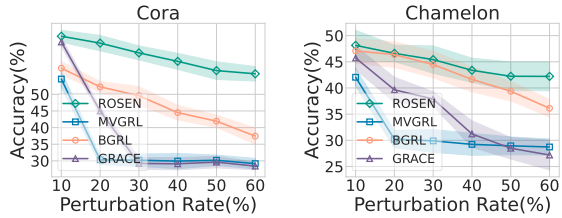


Figure 6: Performance variation of GCL models on graph data with attribute noise.

Observing Figure 3 we can notice the inferred structure matrices are more similar to the block diagonal matrices than the input adjacency matrices. In particular, as exemplified on the Squirrel dataset, many elements of the non-diagonal (stand for the edges that connect the nodes of different classes) are deleted or the weights are reduced, which illustrates the outstanding ability of the proposed graph structure inference module. Besides that, this phenomenon is more obvious on the Actor dataset, which underlines the validity of the proposed inference module.

4.4 Ablation Study

To verify the contributions of the inferred graph structure in different components (i.e., encoder and contrastive loss) of the proposed ROSEN, this section conducts several ablation experiments, as exhibited in Table 4. The "local GCL" stands for the GCL variants which employ the adjacency matrix for both encoding and contrasting. First, ROSEN presents the best classification performance across all datasets compared to the variants, which underlines the rationality of its design. Second, it is observed that compared to Localized GCL, other variants harvest performance gains by employing the inferred graph structures, especially on heterophilic Texas. This demonstrates the effectiveness and feasibility of the proposed graph structure inference module. Third, note that utilizing the inferred graph structure in the contrasting brings more benefits than in the encoding as the accuracy is enhanced more. This may be because the contrast in the outer layer can better control parameter update than the encoding. Overall, the results confirm that the superior performance of ROSEN comes from the design rather than any individual contribution.

4.5 Hyperparameter Study

To provide valuable insights on selecting the number of subspaces (diagonal blocks), this section experimentally analyzes the impact of this hyperparameter on node classification performance. To preserve sufficient neighbor nodes for each node, this hyperparameter is chosen from the range of less than the number of classes.

As can be observed from Figure 4, ROSEN shows steady performance improvements over a range of parameters, which are $\{2, 3\}$, $\{2, 3, 4, 5\}$ and $\{4, 5\}$ for the Cora, Cornell, and Actor, respectively. This illustrates the insensitivity to the number of blocks k. Furthermore, combined with the results in Table 1 and Table 2, it is apparent that ROSEN with a small hyperparameter value (e.g., $k = 2$) still outperforms most baseline models. Compared to the baseline GCLs which treat the neighbor nodes as positive samples, even if ROSEN may incorrectly preserve FALSE positive samples during structure inference, it benefits from excluding part of FALSE positive samples. The results highlight the effectiveness of ROSEN.

4.6 Robustness Analysis

Figure 5 and Figure 6 show the performance variation of GCL models under the topology attack (adding noisy edges) and the attribute attack (flipping attributes), respectively. First, while the baselines manifest the adaptability to minor topology noise, their performance degrades incrementally as the perturbation increases. For example, the accuracy of the baseline GRACE, which employs the same configurations, drops to approximately 67% on Cora when the perturbation rate is 60%. By contrast, ROSEN exhibits stability in preserving its predictive performance under topology perturbation. Under the perturbation rate is 60%, ROSEN achieves an accuracy of approximately 80%. This outstanding performance is attributed to the fact that the inferred graph structure always keeps a high proportion of intra-class edges and thus is insensitive to the local feature distribution. Second, ROSEN is also superior to the baselines under attribute perturbation, which illustrates its robustness to attribute noise. This can be attributed to the utilization of matrix constraints to capture higher-order relationships. The graph structure inference module is stable against attribute attacks and thereby promotes the denoising ability of the models.

5 CONCLUSIONS

This work explores challenges in applying Graph Contrastive Learning (GCL) to both homophilic and heterophilic graphs, presenting a solution named ROSEN. The framework focuses on reconstructing the homophilic graph structure for encoding and positive sampling. It involves selectively removing the disassortative edges through a local inference module, which optimizes a self-expressive learning objective with soft diagonal block regularization on ego networks. The proposed local graph structure inference module significantly enhances the performance of baseline models on several graph-specific tasks. The potential future research directions include designing the beyond local positive sampling and self-supervised criteria for multimodal data.

6 ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (No. 62376088, 61972442, 62272020, 62276187, 62102413, U22B2036, and U1936208), in part by the National Science Fund for Distinguished Young Scholars (No. 62025602), in part by the National Social Science Fund of China under Grant 22VMG037, in part by the Natural Science Foundation of Hebei Province of China under Grant F2020202040, and in part by the Tencent Foundation and XPLORER PRIZE.

REFERENCES

[1] Kristen M Altenburger and Johan Ugander. 2018. Monophily in social networks introduces similarity among friends-of-friends. *Nature human behaviour* 2, 4 (2018), 284–290.

[2] Stephen P. Boyd and Lieven Vandenberghe. 2014. *Convex Optimization*.

[3] Jingfan Chen, Guanghui Zhu, Yifan Qi, Chunfeng Yuan, and Yihua Huang. 2022. Towards Self-supervised Learning on Graphs with Heterophily. In *CIKM*. ACM, 201–211.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*. 1597–1607.

[5] Fan RK Chung and Fan Chung Graham. 1997. *Spectral graph theory*. Number 92.

[6] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *SIGKDD*. 855–864.

[7] John A Hartigan and Manchek A Wong. 1979. Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)* 28, 1 (1979), 100–108.

[8] Kaveh Hassani and Amir Hosein Khas Ahmadi. 2020. Contrastive Multi-View Representation Learning on Graphs. In *ICML*, 4116–4126.

[9] Dongxiao He, Chundong Liang, Huixin Liu, Mingxiang Wen, Pengfei Jiao, and Zhiyong Feng. 2022. Block Modeling-Guided Graph Convolutional Neural Networks. In *AAAI*.4022–4029.

[10] Dongxiao He, Jitao Zhao, Rui Guo, Zhiyong Feng, Di Jin, Yuxiao Huang, Zhen Wang, and Weixiong Zhang. 2023. Contrastive Learning Meets Homophily: Two Birds with One Stone. In *ICML*, 12775–12789.

[11] Roger A Horn and Charles R Johnson. 2012. *Matrix analysis*. Cambridge university press.

[12] Zhenyu Hou, Yufei He, Yukuo Cen, Xiao Liu, Yuxiao Dong, Evgeny Kharlamov, and Jie Tang. 2023. GraphMAE2: A Decoding-Enhanced Masked Self-Supervised Graph Learner. In *WWW*, 737–746.

[13] Zhenyu Hou, Yufei He, Yukuo Cen, Xiao Liu, Yuxiao Dong, Evgeny Kharlamov, and Jie Tang. 2023. GraphMAE2: A Decoding-Enhanced Masked Self-Supervised Graph Learner. In *WWW*, 737–746.

[14] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open Graph Benchmark: Datasets for Machine Learning on Graphs. In *NeurIPS*

[15] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

[16] Thomas N. Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. *CoRR* abs/1611.07308 (2016).

[17] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[18] Wen-Zhi Li, Chang-Dong Wang, Hui Xiong, and Jian-Huang Lai. 2023. HomoGCL: Rethinking Homophily in Graph Contrastive Learning. In *SIGKDD*. 1341–1352.

[19] Canyi Lu, Jiashi Feng, Zhouchen Lin, Tao Mei, and Shuicheng Yan. 2019. Subspace Clustering by Block Diagonal Representation. *IEEE Trans. Pattern Anal. Mach. Intell.* 41, 2 (2019), 487–501.

[20] Yao Ma, Xiaorui Liu, Tong Zhao, Yozen Liu, Jiliang Tang, and Neil Shah. 2020. A Unified View on Graph Neural Networks as Graph Signal Denoising. arXiv:2010.01777

[21] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001), 415–444.

[22] Péter Mernyei and Catalina Cangea. 2020. Wiki-CS: A Wikipedia-Based Benchmark for Graph Neural Networks. *CoRR* (2020).

[23] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. 2020. Geom-GCN: Geometric Graph Convolutional Networks. In *ICLR*.

[24] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: online learning of social representations. In *SIGKDD*. 701–710.

[25] Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. 2023. A critical look at the evaluation of GNNs under heterophily: Are we really making progress?. In *ICLR*.

[26] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. 2021. Multi-Scale attributed node embedding. *J. Complex Networks* (2021).

[27] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. 2008. Collective Classification in Network Data. *AI Mag.* (2008), 93–106.

[28] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Pitfalls of Graph Neural Network Evaluation. *CoRR* (2018).

[29] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. 2009. Social influence analysis in large-scale networks. In *KDD*. 807–816.

[30] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. 2021. Bootstrapped representation learning on graphs. (2021).

[31] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *CoRR* abs/1807.03748 (2018).

[32] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.

[33] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2019. Deep Graph Infomax. In *ICLR*.

[34] Haonan Wang, Jieyu Zhang, Qi Zhu, and Wei Huang. 2022. Can Single-Pass Contrastive Learning Work for Both Homophilic and Heterophilic Graph? *CoRR* arXiv:2211.10890

[35] Yu Wang, Wotao Yin, and Jinshan Zeng. 2019. Global convergence of ADMM in nonconvex nonsmooth optimization. *Journal of Scientific Computing* 78 (2019), 29–63.

[36] Yifei Wang, Qi Zhang, Tianqi Du, Jiansheng Yang, Zhouchen Lin, and Yisen Wang. 2023. A Message Passing Perspective on Learning Dynamics of Contrastive Learning. In *ICLR*.

[37] Yaochen Xie, Zhao Xu, Jingtun Zhang, Zhengyang Wang, and Shuiwang Ji. 2022. Self-supervised learning of graph neural networks: A unified review. *IEEE transactions on pattern analysis and machine intelligence* 45, 2 (2022), 2412–2429.

[38] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation Learning on Graphs with Jumping Knowledge Networks. In *ICML*. 5449–5458.

[39] Yujun Yan, Milad Hashemi, Kevin Swersky, Yaoqing Yang, and Danai Koutra. 2021. Two Sides of the Same Coin: Heterophily and Oversmoothing in Graph Convolutional Neural Networks. arXiv:2102.06462

[40] Hengrui Zhang, Qitian Wu, Yu Wang, Shaofeng Zhang, Junchi Yan, and Philip S. Yu. 2022. Localized Contrastive Learning on Graphs. *CoRR* abs/2212.04604 (2022).

[41] Jiong Zhu, Ryan A. Rossi, Anup Rao, Tung Mai, Nedim Lipka, Nesreen K. Ahmed, and Danai Koutra. 2021. Graph Neural Networks with Heterophily. In *AAAI*. 11168–11176.

[42] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep Graph Contrastive Representation Learning. *CoRR* abs/2006.04131 (2020). arXiv:2006.04131

[43] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph Contrastive Learning with Adaptive Augmentation. In *WWW*.

A APPENDIX

A.1 Datasets

The statistics of twelve graph datasets used in this experiment are shown in Table 5.

Table 5: Statistics of twelve graph datasets. The abbreviation *#EdgeHom* denotes the edge homophily elucidated in [23].

Dataset	Nodes	Edges	Features	Classes	*#EdgeHom*
Cora [27]	2,708	5,278	1,433	7	0.81
CiteSeer [27]	3,327	4,552	3,703	6	0.74
PubMed [27]	19,717	44,324	500	3	0.80
Wiki-CS [22]	11,701	216,123	300	10	0.65
Computers [28]	13,752	245,861	767	10	0.78
Photo [28]	7,650	238,163	745	8	0.83
Cornell [23]	183	295	1,703	5	0.13
Texas [23]	183	309	1,703	5	0.11
Wisconsin [23]	251	499	1,703	5	0.20
Chameleon [26]	2,277	36,101	2,325	5	0.23
Squirrel [26]	5,201	217,073	2,089	5	0.22
Actor [29]	7,600	33,544	931	5	0.22

A.2 Introduction of Baseline Models

A.2.1 Semi-supervised Graph Neural Networks (GNNs). GCN [17], GAT [32], and JKNet [38] are widely-used semi-supervised Graph Neural Network (GNN).
(1) GCN : a widely-used semi-supervised Graph Neural Network (GNN), acquires node representations by placing emphasis on the similarity of neighboring nodes.
(2) GAT: a spatial variant derived from GCN, introduces a self-attention mechanism to enable flexible capturing of inter-node relationships.
(3) JKNet: an innovative approach, adeptly integrates node representations across different layers, culminating in final representations that surpass the flexibility and effectiveness of traditional stacked GCN architectures. This novel integration enhances JKNet's capacity to capture nuanced information and fosters improved performance in diverse graph-based learning tasks.

A.2.2 Network Embedding Models. (1) Network Embedding models. DeepWalk [24] and Node2Vec [6] are network embedding models that employ Random Walk strategies to establish the proximity information in graph structures. Furthermore, DeepWalk emphasizes the analogy of random walk paths to sentences to leverage word2vec algorithm, while Node2Vec enhances the flexibility of the random walk process through breadth-first search (BFS) and depth-first search (DFS). (1) DeepWalk: (2) Node2Vec:

A.2.3 Graph Generative Models. (2) Graph Generative models. GAE [16] and VGAE [16] are both graph-generative models. The former encodes graph structures into a compact latent space via an autoencoder for efficient graph reconstruction, while the latter introduces a variational approach with a Gaussian distribution to capture the uncertainty in the graph structure.
(3) GraphMAE [13].

A.2.4 Graph Contrastive Models. (3) Graph Contrastive Learning models. DGI [33], MVGRL [8], GRACE [42], GCA [43], BGRL [30], and
LOCAL-GCL [40],
HGRL [3],
SP-GCL [34].
HomoGCL [18] are common baseline GCL models. Specifically, DGI brings in a local-global contrastive learning framework by maximizing the mutual information between the node-level and the graph-level representations. MVGRL extends the DGI framework by introducing graph diffusion techniques and a multi-view mechanism. Inspired by the two-branch contrastive learning (CL) framework, which is widely employed in computer vision, GRACE extends it to the graph domain via graph encoders and graph augmentations such as edge deletion and feature masking. Moreover, GCA innovates by incorporating adaptive augmentation strategies tailored to the graph structure. Besides, BGRL enhances GCL models through the incorporation of the bootstrapped strategy, namely bootstrapping the output of a delayed version of the encoder. In addition, HomoGCL capitalizes on graph homophily to expand the positive set using the neighbor nodes that possess specific significances.

A.3 Algorithm Description

To jointly modify the graph structure $\mathbf{S}$ and train the GCL parameters Θ, we alternately update one while fixing the other, and the details are shown in Algorithm 1. And we show in Alogrithm 2 that the structure of graph is inferred through the node embedding.

A.4 Theoretical Analysis

Theorem A.1. *Let Θ, k and $\mathbb{1}\mathbf{G}$ denote the parameters of the GNN encoder, the number of subspaces (blocks), and the subspace indicator, respectively, the proposed ROSEN follows the Expectation-Maximization (EM) algorithm, in which the structure inference and the maximization of the lower bound on the mutual information of the contrastive pairs' representation are equivalent to the E step and the M step, respectively.*

Proof. In the localized GCL framework with the variable positive sample set, the optimal parameters of the GNN encoder is learned by maximizating the function $\mathcal{L}_(\Theta, \Omega)$. This can be formulated as

$$\Theta^* = \arg\max_{\Theta} \sum_{v \in V} \log \sum_{u \in N(v,\Omega)} p(\mathbf{h}_v, \mathbf{h}_u | \Theta) \tag{13}$$

However, since the latent variables, the directly computation of Equation 13 is difficult. With the help of Amortized Variational Inference [35], this problem can be alleviated by introducing the approximated posterior $p(\mathbf{h}_u|\mathbf{h}_v, \Theta)$. Therefore, the log-likelihood function $\mathcal{L}_(\Theta, \Omega)$ in Equation 13 can be reformulated as

$$\begin{aligned} \mathcal{L}_(\Theta, \Omega) &= \sum_{v \in V} \log \sum_{u \in N(v,\Omega)} p(\mathbf{h}_u|\mathbf{h}_v, \Theta) \frac{p(\mathbf{h}_v, \mathbf{h}_u|\Theta)}{p(\mathbf{h}_u|\mathbf{h}_v, \Theta)} \\ &\geq \sum_{v \in V} \sum_{u \in N(v,\Omega)} p(\mathbf{h}_u|\mathbf{h}_v, \Theta) \log p(\mathbf{h}_v, \mathbf{h}_u|\Theta) \\ &\quad - p(\mathbf{h}_u|\mathbf{h}_v, \Theta) \log p(\mathbf{h}_u|\mathbf{h}_v, \Theta) \end{aligned} \tag{14}$$

Algorithm 1 ROSEN

Data: Graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{X})$.
Input: GNN encoder f_Θ, the number of subspaces (blocks) k, and the maximum number of iterations t.
Output: Trained GNN encoder f_Θ, and embeddings $\mathbf{H}_1$ and $\mathbf{H}_2$.

Initialize GNN encoder f_Θ.
for v *in* $\hat{\mathcal{V}}_i$ **do**
  $\hat{\mathcal{G}}_1(\hat{\mathcal{V}}_1, \hat{\mathcal{E}}_1, \hat{\mathbf{X}}_1) \leftarrow \mathrm{AUG}_1(\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{X}))$
  $\hat{\mathcal{G}}_2(\hat{\mathcal{V}}_2, \hat{\mathcal{E}}_2, \hat{\mathbf{X}}_2) \leftarrow \mathrm{AUG}_2(\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{X}))$

  /* E-step */
  for $\hat{\mathcal{G}}_i$ *in* $\{\hat{\mathcal{G}}_1, \hat{\mathcal{G}}_2\}$ **do**
    for v *in* $\hat{\mathcal{V}}_i$ **do**
      $\mathbf{H}_v^{Ego} \leftarrow \mathrm{CAT}\{\mathbf{H}_u | u \in N(v)\}$
      $\mathbf{Z}_v^{Ego}, \mathbf{B}_v^{Ego} \leftarrow \mathrm{StructureInference}(\mathbf{H}_v^{Ego}, k, t)$
      $\mathbf{Z}_v, \mathbf{B}_v \leftarrow [\mathbf{Z}_v^{Ego}]_v, [\mathbf{B}_v^{Ego}]_v$
    end
  end
  Generate $\mathbf{S}_1$ and $\mathbf{S}_2$ via Eq.9 and Eq.10
  /* M-step */
  while *not MaxEpoch* **do**
    $\mathbf{H}_1 \leftarrow f_\Theta(\mathbf{S}_1, \mathbf{H}_1),\ \mathbf{H}_2 \leftarrow f_\Theta(\mathbf{S}_2, \mathbf{H}_2)$
    $\mathcal{L}_{contrast}(\mathbf{H}_1, \mathbf{H}_2, \mathbf{S}_1, \mathbf{S}_2)$
    $\Theta \leftarrow \mathrm{Adam}(\mathcal{L}_{contrast}, \Theta)$
  end
end
return f_Θ and $\mathbf{H}_1, \mathbf{H}_2$

where the inequality holds due to the Jensen's inequality. It is worth noting that $-p(\mathbf{h}_u|\mathbf{h}_v, \Theta) log p(\mathbf{h}_u|\mathbf{h}_v, \Theta)$ is the entropy operator, which does not affect the update of the parameter Θ. Overall, the log-likelihood function can be formulated as

$$l = \sum_{v \in V} \sum_{u \in N(v,\Omega)} \log p(\mathbf{h}_u|\mathbf{h}_v, \Theta) p(\mathbf{h}_v, \mathbf{h}_u|\Theta) \tag{15}$$

Accordingly, the EM algorithm for optimizating this function can be described as inferencing structure in E step and maximizating the lower bound on the mutual information in M step.

E step. To infer the approximated posterior probability $p(\mathbf{h}_u|\mathbf{h}_v, \Theta)$, the proposed structure inference module introduce the bloack diagonal constraint. Thus, this probability can be formulated as

$$p(\mathbf{h}_v|\mathbf{h}_u, \Theta) = \sum_{t=1}^{k} p(\mathbf{h}_u|\mathbf{h}_v, t, \Theta) p(t|\mathbf{h}_v, \Theta)$$

It can be obtained from the local block diagonal graph structure inference in Section 3.2.1, namely $p(\mathbf{h}_v|\mathbf{h}_u, \Theta) = \mathbb{1}\mathbf{G}_{v,u}$, which assumes that the neighbour nodes which belong to the same subspace are of the same class ($\mathbb{1}\mathbf{G}_{v,u} = 1$), i.e., TRUE positive samples.

M step. Based on the E step, the M step focuses on maximizating

Algorithm 2 Local Structure Inference (LSI)

Input: Ego embeddings of node $\mathbf{H}_v^{Ego}$, number of subspaces k, and MaxIteration t.
Output: The affinity matrices $\mathbf{Z}$, $\mathbf{B}$.

Initialize $\mathbf{Z} = \mathbf{B} = \mathbf{W} = \mathbf{0}, \epsilon = 1e-8$
while *not MaxIteration t* **do**
  for v *in* $\hat{\mathcal{V}}_i$ **do**
    $\mathbf{H}_v^{Ego} \leftarrow \mathrm{CAT}\{\mathbf{H}_u | u \in N(v)\}$
    $\mathbf{Z}_v^{Ego}, \mathbf{B}_v^{Ego} \leftarrow \mathrm{StructureInference}(\mathbf{H}_v^{Ego}, k, t)$
    $\mathbf{Z}_v, \mathbf{B}_v \leftarrow [\mathbf{Z}_v^{Ego}]_v, [\mathbf{B}_v^{Ego}]_v$
  end
  Update $\mathbf{Z}^{t+1}$ by
  $\mathbf{Z}^{t+1} = \underset{\mathbf{Z}}{\mathrm{argmin}} \frac{1}{2}\|\mathbf{H}^{Ego} - \mathbf{Z}\mathbf{H}^{Ego}\|^2 + \frac{\lambda}{2}\|\mathbf{Z} - \mathbf{B}\|^2.$
  Update $\mathbf{B}^{t+1}$ by
  $\mathbf{B}^{t+1} = \underset{\mathbf{B}}{\mathrm{argmin}} \frac{\lambda}{2}\|\mathbf{Z} - \mathbf{B}\|^2 + \gamma\langle \mathrm{Diag}(\mathbf{B1}) - \mathbf{B}, \mathbf{W}\rangle,$
  s.t. $\mathrm{diag}(\mathbf{B}) = 0, \mathbf{B} \geq 0, \mathbf{B} = \mathbf{B}^\top.$
  Update $\mathbf{W}^{t+1}$ by $\mathbf{W}^{t+1} = \underset{\mathbf{W}}{\mathrm{argmin}} \langle \mathrm{Diag}(\mathbf{B1}) - \mathbf{B}, \mathbf{W}\rangle,$
  s.t. $0 \leq \mathbf{W} \leq \mathbf{I}, \mathrm{Tr}(\mathbf{W}) = k.$
  Check the convergence conditions:
  $\|\mathbf{Z}^{t+1} - \mathbf{Z}^t\|_\infty \leq \epsilon, \|\mathbf{B}^{t+1} - \mathbf{B}^t\|_\infty \leq \epsilon.$
end
return *The coefficient matrices* $\mathbf{Z}$, $\mathbf{B}$

the lower-bound of Equation 15. In particular, there are

$$l = \sum_{v \in V} \sum_{u \in N(v,\Omega)} p(\mathbf{h}_u|\mathbf{h}_v, \Theta) \log p(\mathbf{h}_v, \mathbf{h}_u|\Theta) \tag{16}$$

$$= \sum_{v \in V} \sum_{u \in N(v,\Omega)} \mathbb{1}\mathbf{G}_{v,u} \log p(\mathbf{h}_v, \mathbf{h}_u|\Theta) \tag{17}$$

and $p(\mathbf{h}_v, \mathbf{h}_u|\Theta) = \frac{1}{|N(v,\Omega)|} p(\mathbf{h}_u|\mathbf{h}_u, \Theta)$. Since we consider that the prior obeys a uniform distribution, and describe the distribution of each sample in the feature space with isotropic Gaussian, thus there is

$$p(\mathbf{h}_v|\mathbf{h}_u, \Theta) = \frac{1}{2\sigma_i^2} \exp\left(-\frac{1}{2\sigma_1^2} (\mathbf{h}_v - \mathbf{h}_u)^T \cdot (\mathbf{h}_v - \mathbf{h}_u)\right) \tag{18}$$

$$= \frac{1}{2\sigma_i^2} \exp\left(-\frac{\left(\mathbf{h}_v^T \cdot \mathbf{h}_u - 1\right)}{2\sigma_1^2}\right) \tag{19}$$

where the last equivalence due to the L2 normalization for the features $\mathbf{H}$. Setting $\tau = \sigma^2$ as the hyperparameter for all terms, ignoring the constant term, and taking Equation 19 into Equation 17, it can be obtained as

$$\sum_{v \in V} log \frac{\sum_{v^+ \in N_v^S} \mathbb{1}\mathbf{G}_{v,v_+} * e^{\theta(\mathbf{h}_v, \mathbf{h}_{v^+})/\tau}}{\sum_{v^+ \in N_v^S} \mathbb{1}\mathbf{G}_{v,v_+} * e^{\theta(\mathbf{h}_v, \mathbf{h}_{v^+})/\tau} + \sum_{v^- \in \{V \setminus \mathcal{N}_v^S\}} e^{\theta(\mathbf{h}_v, \mathbf{h}_{v^-})/\tau}}$$

It can be discovered that when considering weights which reflect local higher-order relationships, the objective is equivalent to the ROSEN. □

Table 6: Statistics of three large graph datasets and the performance comparison of GCL models on these large graphs.

Datasets	Ogbn-Arxiv	Ogbn-Products	MAG-Scholar-F
#Nodes	169,343	2,449,029	1,939,743
#Edges	1,166,243	61,859,140	358,010,024
#Features	128	100	128
DGI	68.49 ± 0.02	78.36 ± 0.04	55.38 ± 0.03
MVGRL	70.47 ± 0.14	78.32 ± 0.04	56.82 ± 0.02
GRACE	69.75 ± 0.01	79.89 ± 0.22	55.47 ± 0.04
BGRL	70.27 ± 0.03	79.02 ± 0.01	56.59 ± 0.02
HomoGCL	71.23 ± 0.02	80.79 ± 0.09	57.89 ± 0.01
ROSEN	**72.97 ± 0.22**	**82.34 ± 0.13**	**59.12 ± 0.03**

A.5 Scalability Study

The results in Table 6 show the performance advantage of the proposed ROSEN,which demonstrates the effectiveness and scalability of ROSEN.

Received 13 October 2023; revised 15 December 2023; accepted 23 January 2024