

# Dual Low-Rank Graph Autoencoder for Semantic and Topological Networks

Zhaoliang Chen<sup>1,2</sup>, Zhihao Wu<sup>1,2</sup>, Shiping Wang<sup>1,2</sup>, Wenzhong Guo<sup>1,2\*</sup>

<sup>1</sup> College of Computer and Data Science, Fuzhou University, Fuzhou, China

<sup>2</sup> Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou University, Fuzhou, China

chenzl23@outlook.com, zhihaowu1999@gmail.com, shipingwangphd@163.com, guowenzhong@fzu.edu.cn

## Abstract

Due to the powerful capability to gather the information of neighborhood nodes, Graph Convolutional Network (GCN) has become a widely explored hotspot in recent years. As a well-established extension, Graph AutoEncoder (GAE) succeeds in mining underlying node representations via evaluating the quality of adjacency matrix reconstruction from learned features. However, limited works on GAE were devoted to leveraging both semantic and topological graphs, and they only indirectly extracted the relationships between graphs via weights shared by features. To better capture the connections between nodes from these two types of graphs, this paper proposes a graph neural network dubbed Dual Low-Rank Graph AutoEncoder (DLR-GAE), which takes both semantic and topological homophily into consideration. Differing from prior works that share common weights between GCNs, the presented DLR-GAE conducts sustained exploration of low-rank information between two distinct graphs, and reconstructs adjacency matrices from learned latent factors and embeddings. In order to obtain valid adjacency matrices that meet certain conditions, we design some surrogates and projections to restrict the learned factor matrix. We compare the proposed model with state-of-the-art methods on several datasets, which demonstrates the superior accuracy of DLR-GAE in semi-supervised classification.

## Introduction

Graph Convolutional Network (GCN) (Kipf and Welling 2017) is a crucial technique of graph learning and has been universally applied to a multitude of machine learning tasks (Kang et al. 2020; Bian et al. 2020; Liu et al. 2022; Feng et al. 2021), including node classification, computer vision and social analysis. Originated from GCN, Graph AutoEncoder (GAE) aims to explore underlying node representations from topological networks. It learns low-dimensional node embeddings via graph convolutional layers, and supervises the learned latent features via link connections stored in the adjacency matrix. A multitude of researchers have put emphases on the study and applications of GAE (Sánchez-Martín, Rateike, and Valera 2022; Jing, Xu, and Li 2022; Liu et al. 2019).

\*Corresponding author.

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

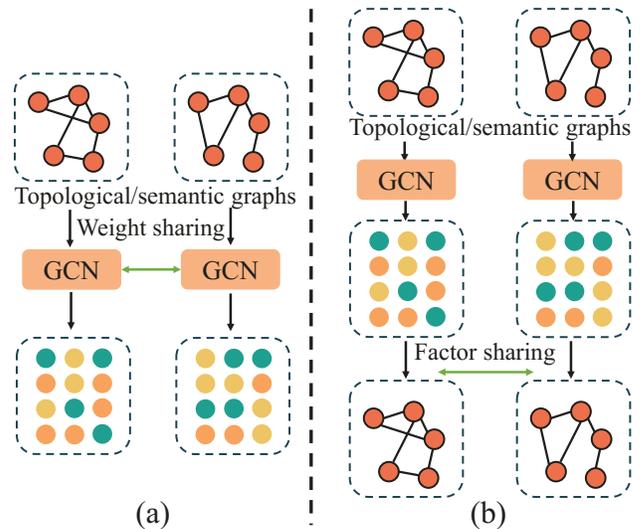


Figure 1: Comparison of GCN exploring semantic and topological graphs. Subfigure (a) depicts prior works that built parallel GCNs with weight sharing to learn two distinct node embeddings. Subfigure (b) briefly elaborates on the proposed model. Instead, it considers the factor sharing of reconstructed adjacency matrices to explore a common and direct correlation between two types of graphs.

As a supplement to the natural topological structure of nodes, extracting semantic node connections from the feature space is beneficial to more dexterous embedding exploration, which has been investigated by some state-of-the-art works (Wang et al. 2020; Zhang et al. 2021). The utilization of feature graphs is helpful to capturing additional homophily between vertices and latent semantics in nodes. As described in Figure 1 (a), most of the prior frameworks built parallel networks to manipulate topological and semantic node relationships simultaneously. These models often fed different adjacency matrices to two networks respectively, and shared convolutional weights between them. Eventually, the unified node representation is attained via a linear weighted combination or an attention mechanism.

Nevertheless, these models only explore the latent consistency of learned features through shared convolutional

weights. As a matter of fact, compared to indirect relationships between learned features, the correlation between two types of adjacency matrices is often stronger. This is because that semantic node connections extracted from features not only complement the missing links in topological graphs, but also preserve similar characteristics to the topological structure. It motivates us to develop parallel networks that learn underlying common factors between two types of graphs, as shown in Figure 1 (b). We attempt to study the shared latent factors instead of shared weights for GCN, expecting that more dexterous and effective features can be extracted by two networks, and the consistency of two graphs can be represented by a shared latent factor matrix. Accordingly, in this paper, we come up with a Dual Low-Rank Graph AutoEncoder (DLR-GAE) framework, which explores common low-rank factors between reconstructed semantic adjacency matrix and topological adjacency matrix in a factorization-based method. Derived from a Schatten- $p$ -norm-based low-rank matrix factorization optimization problem, a differentiable loss function is proposed to approximately minimize the optimization objective. For the pursuit of achieving valid adjacency matrix reconstruction, we add constraints to the learned latent factor matrix, and develop some surrogates and projections to ensure that the matrix satisfies certain properties such as symmetry and non-negativity. In short, the main contributions of this paper are listed as below:

- 1) A dual low-rank GAE addressing semantic and topological graphs is constructed, whose decoders conduct adjacency matrix reconstruction via optimizing the surrogate of Schatten- $p$  norm w.r.t. a shared latent factor matrix.
- 2) For the sake of theoretical strictness, several surrogates and projections are designed to guarantee the validity of the factor matrix, which enables DLR-GAE to reconstruct valid adjacency matrices and promotes the learning performance.
- 3) Comprehensive experiments on several graph datasets are conducted to verify the superiority of the proposed DLR-GAE, which reveal that the proposed model outperforms state-of-the-art graph-based models.

## Related Works

### Graph Autoencoder

We first review the basic architecture of a GAE, which is formulated as

$$\mathbf{Z} = \mathcal{F}_{GCN} \left( \mathbf{X}, \tilde{\mathbf{A}} \mid \left\{ \mathbf{W}^{(l)} \right\}_{l=1}^L \right) \quad (1)$$

$$\hat{\mathbf{A}} = \text{sigmoid}(\mathbf{Z}\mathbf{Z}^T), \quad (2)$$

where  $\mathbf{X}$  is the node feature. In particular, each layer of GCN is formulated as

$$\mathbf{H}^{(l)} = \sigma \left( \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{H}^{(l-1)} \mathbf{W}^{(l)} \right), \quad (3)$$

where  $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$  is the self-connected adjacency matrix,  $\sigma(\cdot)$  is the activation function, and  $[\tilde{\mathbf{D}}]_{ii} = \sum_j [\mathbf{A}]_{ij}$ . Different from a common autoencoder, GAE encodes node features to low-dimensional embeddings via GCN, and decodes the learned representation via an inner-product. GAE-based

models have been extensively applied to learning underlying node representations. For example, a variational GAE was utilized to approximate the interventional and counterfactual distributions on various structural causal models (Kipf and Welling 2016; Sánchez-Martín, Rateike, and Valera 2022). A graph-based autoencoder framework was proposed to perform the semi-supervised classification tasks, which trained a matrix completion target and a classifier simultaneously (Kang et al. 2020). A graph masked autoencoder was presented to reduce the dependence on supervision information via a self-supervised pre-training strategy with untrained architectures (Jing, Xu, and Li 2022).

### Learning from Semantic and Topological Graphs

Several recent studies have attempted to learn a unified node representation from both semantic and topological graphs. An adaptive multi-channel GCN was proposed for learning common features from feature space and topology space simultaneously, which used the attention mechanism to obtain a weighted embedding (Wang et al. 2020). (Wu et al. 2019) proved that topological connections could be regarded as the low-pass filtering on node features when the node embedding propagated over the topology network. (Gao, Pei, and Huang 2019) presented a conditional random field layer to preserve the connective between various vertices. The node homophily learned from both structural and feature similarities was exploited to explore potential neighbor connections among nodes (Zhang et al. 2021). Some of these existing works constructed parallel graph neural networks to manipulate two graphs simultaneously, sharing the weights of GCNs. This may be restrictive because it is an indirect exploration of relationships between feature space, rather than a direct extraction of the common information between graphs. Instead, this paper considers mining underlying correlations between two kinds of graphs via a factor matrix, on the basis of which we reconstruct low-rank adjacency matrices in a matrix factorization way to implicitly study the shared latent information.

## The Proposed Method

In this section, we elaborate on the proposed model. Given a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with  $n$  nodes, the natural topological node relationships are described by a binary adjacency matrix  $\mathbf{A}_{\mathcal{T}} \in \mathbb{R}^{n \times n}$ , and the semantic node connections obtained from the feature space are stored in the adjacency matrix  $\mathbf{A}_{\mathcal{S}} \in \mathbb{R}^{n \times n}$ . Node features are represented by  $\mathbf{X} \in \mathbb{R}^{n \times m}$ . The proposed framework aims to explore a unified node embedding from topological and feature space via low-rank-factor-shared GAEs. A detailed illustration of the proposed DLR-GAE is shown in Figure 2.

### GAE with Low-Rank Reconstruction

In pursuit of supervising the learned node embeddings, we reconstruct the adjacency matrix via a low-rank matrix factorization target. The low-rank property is beneficial to the adjacency matrix reconstruction, because nodes often have similar connective patterns and the information in the similarity matrix is generally redundant. In order to optimize

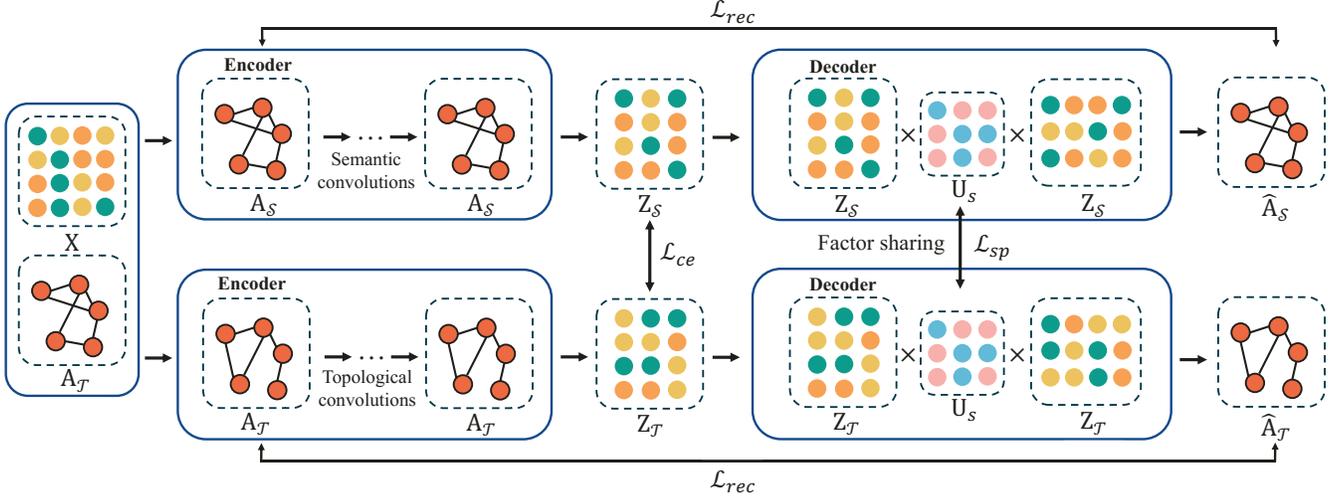


Figure 2: The framework of the proposed DLR-GAE, which is a parallel graph neural network learning node embeddings from semantic and topological graphs simultaneously.

the low-rank reconstruction with node embeddings, we first review a PCA-like optimization problem, defined as:

$$\min_{\mathbf{Z}, \mathbf{U}} \|\mathbf{A} - \mathbf{Z}\mathbf{U}^T\|_F^2 + g(\mathbf{U}), \quad (4)$$

where  $\mathbf{A}$  is a known adjacency matrix,  $\mathbf{Z}$  is the node embedding matrix and  $\mathbf{U}$  is the low-dimensional latent factor matrix.  $g(\mathbf{U})$  is a low-rank regularization w.r.t.  $\mathbf{U}$ . The objective defined in Eq. (4) is restrictive, because the adjacency matrix reconstruction  $\hat{\mathbf{A}} = \mathbf{Z}\mathbf{U}^T$  is asymmetric, which is problematic for a graph convolutional operation. In addition, the low-dimensional mapping  $\mathbf{Z}\mathbf{U}^T$  ignores the fact that the same row and column of  $\mathbf{A}$  depicts the node relationships of the same vertex.

In pursuit of addressing the aforementioned issues, we first introduce an asymmetric reconstruction of the adjacency matrix, whose objective can be written as:

$$\min_{\mathbf{Z}, \mathbf{U}} \|\mathbf{A} - \mathbf{Z}\mathbf{U}\mathbf{Z}^T\|_F^2 + g(\mathbf{U}), \quad (5)$$

where  $\mathbf{Z}\mathbf{U}\mathbf{Z}^T$  is the asymmetric matrix. The factor model defined in Eq. (5) maps the node features onto the  $\mathbb{R}^d$  space and encodes the incoming and outgoing connectivity of each node. Namely, the reconstruction  $\hat{\mathbf{A}} = \mathbf{Z}\mathbf{U}\mathbf{Z}^T$  also considers the connections among distinct nodes. Because  $\mathbf{A}$  is generally binary and sparse, we rewrite the objective as

$$\min_{\mathbf{Z}, \mathbf{U}} \mathcal{L}_{rec}(\mathbf{A}, \mathbf{Z}, \mathbf{U}) + g(\mathbf{U}), \quad (6)$$

where  $\mathcal{L}_{rec}(\mathbf{A}, \mathbf{Z}, \mathbf{U})$  is the log likelihood reconstruction error defined as

$$\mathcal{L}_{rec}(\mathbf{A}, \mathbf{Z}, \mathbf{U}) = -\frac{1}{\kappa} \sum_{i,j} \mathbf{A}_{ij} \log(\hat{\mathbf{A}}_{ij}), \quad (7)$$

where  $\hat{\mathbf{A}}_{ij} = [\mathbf{Z}\mathbf{U}\mathbf{Z}^T]_{ij}$  and  $\kappa$  is the number of non-zero entries in  $\mathbf{A}$ . It is noted that Eq. (7) only evaluates the reconstruction performance of existing node connections in  $\mathbf{A}$ . In

this paper, we further consider a multi-factor model for the adjacency matrix reconstruction. Factorizing  $\mathbf{U}$  into numerous low-rank matrices  $\mathbf{U}_1 \in \mathbb{R}^{d \times d_1}$ ,  $\mathbf{U}_i \in \mathbb{R}^{d_{i-1} \times d_i}$ ,  $i = 2, \dots, I-1$ ,  $\mathbf{U}_I \in \mathbb{R}^{d_{I-1} \times d}$ , we have

$$\min_{\mathbf{Z}, \{\mathbf{U}_i\}_{i=1}^I} \mathcal{L}_{rec}(\mathbf{A}, \mathbf{Z}, \mathbf{U}) + g\left(\prod_{i=1}^I \mathbf{U}_i\right). \quad (8)$$

Because  $d \ll n$ ,  $\mathbf{Z} \prod_{i=1}^I \mathbf{U}_i \mathbf{Z}^T$  can be regarded as a low-rank reconstruction of  $\mathbf{A}$ . Owing to the low-rank constraint, we adopt Schatten- $p$  norm to regularize  $\mathbf{U} = \prod_{i=1}^I \mathbf{U}_i$ , i.e.,

$$\|\mathbf{U}\|_{S_p} = \left( \sum_{i=1}^{\min\{m,n\}} \sigma_i(\mathbf{U})^p \right)^{\frac{1}{p}} = \left( \text{Tr}((\mathbf{U}^T \mathbf{U})^{\frac{p}{2}}) \right)^{\frac{1}{p}}, \quad (9)$$

where the indicator  $p$  satisfies  $0 < p < \infty$ . Schatten- $p$  norm is an extensively used unitarily invariant norm. Particularly, it becomes the nuclear norm or trace norm when  $p = 1$ . Exactly, it is the rank norm when  $p = 0$ . For a matrix-factorization-based Schatten- $p$  norm, we have the following lemma (Xu, Lin, and Zha 2017; Chen et al. 2023).

**Lemma 1.** Given  $I$  factors  $\mathbf{U}_i$ ,  $i = 1, \dots, I$ , where  $\mathbf{U}_1 \in \mathbb{R}^{d \times d_1}$ ,  $\mathbf{U}_i \in \mathbb{R}^{d_{i-1} \times d_i}$ ,  $i = 2, \dots, I-1$ ,  $\mathbf{U}_I \in \mathbb{R}^{d_{I-1} \times d}$  with  $\text{rank}(\mathbf{U}) = r \leq \min\{d_i, i = 1, \dots, I\}$ , we have

$$\frac{1}{p} \|\mathbf{U}\|_{S_p}^p = \min_{\mathbf{U}_i: \mathbf{U} = \prod_{i=1}^I \mathbf{U}_i} \sum_{i=1}^I \frac{1}{p_i} \|\mathbf{U}_i\|_{S_{p_i}}^{p_i}, \quad (10)$$

where any  $p_i > 0$  satisfies  $\frac{1}{p} = \sum_{i=1}^I \frac{1}{p_i}$ .

Consequently, we first transform Eq. (8) into

$$\min_{\mathbf{Z}, \{\mathbf{U}_i\}_{i=1}^I} \mathcal{L}_{rec}(\mathbf{A}, \mathbf{Z}, \mathbf{U}) + \frac{1}{p} \sum_{i=1}^I \|\mathbf{U}_i\|_{S_{p_i}}^{p_i}. \quad (11)$$

In particular, setting  $p_i = 2$  for  $i = 1, \dots, I$ , Eq. (11) becomes an objective with Frobenius norm, i.e.,

$$\min_{\mathbf{Z}, \{\mathbf{U}_i\}_{i=1}^I} \mathcal{L}_{rec}(\mathbf{A}, \mathbf{Z}, \mathbf{U}) + \frac{I}{2} \sum_{i=1}^I \|\mathbf{U}_i\|_F^2. \quad (12)$$

According to Lemma 1, it is equivalent to

$$\min_{\mathbf{Z}, \mathbf{U}_i: \mathbf{U} = \prod_{i=1}^I \mathbf{U}_i} \mathcal{L}_{rec}(\mathbf{A}, \mathbf{Z}, \mathbf{U}) + \frac{I}{2} \|\mathbf{U}\|_{S_p}^p, \quad (13)$$

which indicates that  $p = \frac{2}{I}$  for the low-rank regularization w.r.t.  $\mathbf{U} = \prod_{i=1}^I \mathbf{U}_i$ . Therefore, when  $I \rightarrow \infty$ ,  $\frac{I}{2} \|\mathbf{U}\|_{S_p}^p$  is close to rank norm owing to  $p \rightarrow 0$ . In order to obtain a symmetric reconstruction of the adjacency matrix  $\mathbf{A}$ , we use  $\mathbf{U}_s = \mathbf{U}\mathbf{U}^T$  which is symmetric to replace  $\mathbf{U}$  in Eq. (13). Namely, Eq. (13) is transformed into

$$\min_{\mathbf{Z}, \mathbf{U}_i: \mathbf{U} = \prod_{i=1}^I \mathbf{U}_i} \mathcal{L}_{rec}(\mathbf{A}, \mathbf{Z}, \mathbf{U}_s) + \mathcal{L}_{sp}(\mathbf{U}_s), \quad (14)$$

where  $\mathbf{U}_s = \mathbf{U}\mathbf{U}^T$ . We define the Schatten- $p$  norm surrogate w.r.t.  $\mathbf{U}_s$  as

$$\mathcal{L}_{sp}(\mathbf{U}_s) = \frac{I}{2} \sum_{i=1}^I \|\mathbf{U}_i\|_F^2 \quad (15)$$

with  $p = \frac{2}{I}$ . It is exactly  $\frac{I}{2} \|\mathbf{U}\|_{S_p}^p$ , which is equivalent to  $\frac{I}{2} \|\mathbf{U}_s\|_{S_p}^p$ . We have the following theorem for this surrogate.

**Theorem 1.** *For any factor matrix  $\mathbf{U} \in \mathbb{R}^{d \times d}$ , the optimization of  $\frac{I}{2} \|\mathbf{U}\|_{S_p}^p$  is equivalent to that of  $\frac{I}{2} \|\mathbf{U}_s\|_{S_p}^p$  when  $p = \frac{2}{I}$ , where  $\mathbf{U}_s = \mathbf{U}\mathbf{U}^T$ .*

*Proof.* Due to Lemma 1, for any  $\mathbf{U} \in \mathbb{R}^{d \times d}$ , we have

$$\min_{\mathbf{U}} \frac{1}{p} \|\mathbf{U}\mathbf{U}^T\|_{S_p}^p = \min_{\mathbf{U}} \left\{ \frac{1}{p_1} \|\mathbf{U}\|_{S_{p_1}}^{p_1} + \frac{1}{p_2} \|\mathbf{U}^T\|_{S_{p_2}}^{p_2} \right\}. \quad (16)$$

When  $p_1 = p_2 = \hat{p}$ , it leads to

$$\min_{\mathbf{U}} \frac{1}{\hat{p}} \|\mathbf{U}\mathbf{U}^T\|_{S_{\hat{p}}}^{\hat{p}} = \min_{\mathbf{U}} \frac{2}{\hat{p}} \|\mathbf{U}\|_{S_{\hat{p}}}^{\hat{p}}, \quad (17)$$

attributed to  $\|\mathbf{U}\|_{S_{\hat{p}}}^{\hat{p}} = \|\mathbf{U}^T\|_{S_{\hat{p}}}^{\hat{p}}$ . Letting  $\frac{1}{\hat{p}} = \frac{2}{\hat{p}}$ , we have  $\hat{p} = 2p$ . Therefore, the equation

$$\min_{\mathbf{U}} \frac{I}{2} \|\mathbf{U}\mathbf{U}^T\|_{S_p}^p = \min_{\mathbf{U}} \frac{I}{2} \|\mathbf{U}\|_{S_p}^p \quad (18)$$

holds when  $\hat{p} = 2p = \frac{4}{I}$ . That is,

$$\min_{\mathbf{U}: \mathbf{U}_s = \mathbf{U}\mathbf{U}^T} \frac{I}{2} \|\mathbf{U}_s\|_{S_p}^p = \min_{\mathbf{U}} \frac{I}{2} \|\mathbf{U}\|_{S_p}^p, \quad (19)$$

when  $p = \frac{2}{I}$ . This completes the proof.  $\square$

Consequently, we can adopt the surrogate  $\mathbf{U}_s = \mathbf{U}\mathbf{U}^T$  to guarantee the symmetry of the estimated  $\hat{\mathbf{A}} = \mathbf{Z}\mathbf{U}_s\mathbf{Z}^T$ . Eventually, we define the formulation of a low-rank GAE  $\mathcal{F}_{LRGAE}(\mathbf{X}, \tilde{\mathbf{A}}|\mathbf{U}_s, \{\mathbf{W}^{(l)}\}_{l=1}^L)$  as:

$$\mathbf{Z} = \mathcal{F}_{GCN}(\mathbf{X}, \tilde{\mathbf{A}}|\{\mathbf{W}^{(l)}\}_{l=1}^L), \quad (20)$$

$$\mathbf{U} = \prod_{i=1}^I \mathbf{U}_i, \quad \mathbf{U}_s = \mathbf{U}\mathbf{U}^T, \quad (21)$$

$$\hat{\mathbf{A}} = \text{sigmoid}(\hat{\mathbf{Z}}\mathbf{U}_s\hat{\mathbf{Z}}^T), \quad (22)$$

where the renormalized  $\hat{\mathbf{Z}} = \text{softmax}(\mathbf{Z})$ . Actually, if  $\mathbf{U}_s = \mathbf{I}$ , it is a similar structure of GAE.

## Dual Low-Rank GAE

In light of previous analysis, in this subsection, we further propose a dual low-rank GAE to explore semantic and topological node relationships simultaneously. The topological graph is obtained from the existing node connections, while the semantic graph can be estimated by measuring feature distances among nodes. Specifically, we can use the cosine similarity to measure the correlations among nodes and generate a semantic graph through the KNN method that keeps the most important node connections.

In real-world datasets, semantic graphs often have tight connections to topological graphs, because they describe the connections within the same node set. In light of this, the semantic and topological networks should have similar semantic information, which is stored in the latent factor matrix factorized from adjacency matrices. Accordingly, the proposed framework consists of two parallel GAEs, sharing the same factor loading matrix  $\mathbf{U}_s$ . Namely,

$$\mathbf{Z}_{\mathcal{T}}, \hat{\mathbf{A}}_{\mathcal{T}} = \mathcal{F}_{LRGAE}(\mathbf{X}, \tilde{\mathbf{A}}_{\mathcal{T}}|\mathbf{U}_s, \{\mathbf{W}_{\mathcal{T}}^{(l)}\}_{l=1}^L), \quad (23)$$

$$\mathbf{Z}_{\mathcal{S}}, \hat{\mathbf{A}}_{\mathcal{S}} = \mathcal{F}_{LRGAE}(\mathbf{X}, \tilde{\mathbf{A}}_{\mathcal{S}}|\mathbf{U}_s, \{\mathbf{W}_{\mathcal{S}}^{(l)}\}_{l=1}^L). \quad (24)$$

For the sake of better interpretability, we add some constraints to  $\mathbf{U}_s$ . Accordingly, the minimization objective of DLR-GAE becomes

$$\begin{aligned} & \mathcal{L}(\mathbf{Z}_{\mathcal{T}}, \mathbf{Z}_{\mathcal{S}}, \mathbf{A}, \mathbf{U}_s, \mathbf{Y}) \\ &= \mathcal{L}_{ce}(\mathbf{Z}_{\mathcal{T}}, \mathbf{Z}_{\mathcal{S}}, \mathbf{Y}) \\ &+ \alpha \mathcal{L}_{rec}(\mathbf{A}, \mathbf{Z}_{\mathcal{T}}, \mathbf{U}_s) + (1 - \alpha) \mathcal{L}_{rec}(\mathbf{A}, \mathbf{Z}_{\mathcal{S}}, \mathbf{U}_s) \\ &+ \gamma \mathcal{L}_{sp}(\mathbf{U}_s), \end{aligned} \quad (25)$$

s.t.  $\mathbf{U}_s \mathbf{1} = \mathbf{1}$ ,  $\mathbf{U}_s^T \mathbf{1} = \mathbf{1}$ ,  $[\mathbf{U}_s]_{ij} > 0$ ,

where  $\gamma$  is a hyperparameter to adjust the influence of Schatten- $p$  norm. It is reasonable to put nonnegative constraint to  $\prod_{i=1}^m \mathbf{U}_i$ , because entries in  $\mathbf{A}$  is generally non-negative. We require that  $\mathbf{U}_s \mathbf{1} = \mathbf{1}$  and  $\mathbf{U}_s^T \mathbf{1} = \mathbf{1}$  for symmetry reconstruction and weighted node representations. Herein, the semi-supervised cross-entropy error  $\mathcal{L}_{ce}(\mathbf{Z}_{\mathcal{T}}, \mathbf{Z}_{\mathcal{S}}, \mathbf{Y})$  is defined as

$$\begin{aligned} & \mathcal{L}_{ce}(\mathbf{Z}_{\mathcal{T}}, \mathbf{Z}_{\mathcal{S}}, \mathbf{Y}) = \\ & - \sum_{i \in \Omega} \sum_{j=1}^c \mathbf{Y}_{ij} \ln [\alpha \mathbf{Z}_{\mathcal{T}} + (1 - \alpha) \mathbf{Z}_{\mathcal{S}}]_{ij}, \end{aligned} \quad (26)$$

where  $\Omega$  is the set containing nodes with semi-supervised information, and  $\alpha$  is the trade-off parameter balancing the impact of semantic and topological graphs. In pursuit of satisfying constraints in Eq. (25), we can apply the differentiable Dykstra's projection algorithm (Zeng et al. 2019) to map  $\mathbf{U}_s$  onto three feasible sets individually, i.e.,

$$\mathcal{P}_1(\mathbf{U}_s) = \text{ReLU}(\mathbf{U}_s), \quad (27)$$

$$\mathcal{P}_2(\mathbf{U}_s) = \mathbf{U}_s - \frac{1}{d} (\mathbf{U}_s \mathbf{1} - \mathbf{1}) \mathbf{1}^T, \quad (28)$$

$$\mathcal{P}_3(\mathbf{U}_s) = \mathbf{U}_s - \frac{1}{d} \mathbf{1} (\mathbf{1}^T \mathbf{U}_s - \mathbf{1}^T). \quad (29)$$

---

**Algorithm 1: Dual Low-Rank Graph AutoEncoder**

---

**Input:** Node features  $\mathbf{X} \in \mathbb{R}^{n \times m}$ , topological adjacency matrix  $\mathbf{A}_{\mathcal{T}} \in \mathbb{R}^{n \times n}$ , semantic adjacency matrix  $\mathbf{A}_{\mathcal{S}} \in \mathbb{R}^{n \times n}$ , numbers of factor matrices  $I$ , ground truth  $\mathbf{Y} \in \mathbb{R}^{n \times c}$ , hyperparameters  $\alpha$  and  $\gamma$ .

**Output:** Node embedding  $\mathbf{Z}$ .

- 1: Initialize trainable weights  $\{\mathbf{W}_{\mathcal{T}}^{(l)}\}_{l=1}^L, \{\mathbf{W}_{\mathcal{S}}^{(l)}\}_{l=1}^L$ , factor matrices  $\{\mathbf{U}_i\}_{i=1}^I$ ;
  - 2: **while**  $\mathcal{L}(\mathbf{Z}_{\mathcal{T}}, \mathbf{Z}_{\mathcal{S}}, \mathbf{A}, \mathbf{U}_s, \mathbf{Y})$  does not converge or the model does not meet the early-stop condition **do**
  - 3:   Compute the outputs  $\mathbf{Z}_{\mathcal{T}}, \hat{\mathbf{A}}_{\mathcal{T}}, \mathbf{Z}_{\mathcal{S}}, \hat{\mathbf{A}}_{\mathcal{S}}, \mathbf{U}_s$  of DLR-GAE via Eqs. (20) - (22);
  - 4:   Conduct the softmax operation on  $\mathbf{U}_s$  via Eq. (30);
  - 5:   Iteratively project the low-rank latent factor matrix  $\mathbf{U}_s$  onto the feasible set via Eqs. (27) - (29);
  - 6:   Calculate the loss value  $\mathcal{L}(\mathbf{Z}_{\mathcal{T}}, \mathbf{Z}_{\mathcal{S}}, \mathbf{A}, \mathbf{U}_s, \mathbf{Y})$  via Eqs. (7), (15), (26) and (25);
  - 7:   Update trainable weights  $\{\mathbf{W}_{\mathcal{T}}^{(l)}\}_{l=1}^L, \{\mathbf{W}_{\mathcal{S}}^{(l)}\}_{l=1}^L$  and factor matrices  $\{\mathbf{U}_i\}_{i=1}^I$  with back propagation;
  - 8: **end while**
  - 9: Obtain the unified embedding  $\mathbf{Z} = \alpha \mathbf{Z}_{\mathcal{T}} + (1 - \alpha) \mathbf{Z}_{\mathcal{S}}$ .
  - 10: **return** Node embedding  $\mathbf{Z}$ .
- 

Nevertheless, Dykstra’s projection is an iterative algorithm that stops when the current solution meets the condition, requiring a large number of iterations for convergence. This is not acceptable during network training, especially when  $d$  is large. In order to accelerate the convergence speed of Eqs. (27) - (29), a softmax initialization is adopted to obtain a  $\mathbf{U}_s$  that approximately satisfies constraints. Namely,

$$\mathcal{P}_0(\mathbf{U}_s) = \frac{\text{softmax}_{dim=0}(\mathbf{U}_s) + \text{softmax}_{dim=1}(\mathbf{U}_s)}{2}, \quad (30)$$

which makes the cumulative sum of entries in each column or row is close to 1. With Eq. (30), we only need to conduct the three-step projection several times at each training epoch to get the desired  $\mathbf{U}_s$ .

## Training Algorithm

The training progress of DLR-GAE is illustrated in Algorithm 1. The whole framework first performs the forward propagation of DLR-GAE, and then projects the learned  $\mathbf{U}_s$  via the iterative projection method. Eventually, we calculate the loss values and update all trainable weights with back propagation. Supposing that a 2-layer GCN with trainable weights  $\mathbf{W}^{(1)} \in \mathbb{R}^{m \times d}$  and  $\mathbf{W}^{(2)} \in \mathbb{R}^{d \times c}$  is adopted, the computational complexity of encoders is linearly related to the number of edges  $|\mathcal{E}|$ , i.e.,  $\mathcal{O}(|\mathcal{E}|mdc)$ , where  $c$  is the number of classes. The matrix reconstruction procedure takes about  $\mathcal{O}(n^2c)$ . Consequently, the overall computational complexity of DLR-GAE is  $\mathcal{O}(|\mathcal{E}|mdc + n^2c)$ .

Datasets	# Nodes	# Edges	# Features	# Classes
Citeseer	3,327	4,732	3,703	6
CoraFull	19,793	63,421	8,710	70
BlogCatalog	5,196	171,743	8,189	6
ACM	3,025	13,128	1,870	3
Flickr	7,575	239,738	12,047	9
UAI	3,067	28,311	4,973	19

Table 1: A brief statistics of adopted graph datasets.

## Experimental Analysis

### Datasets and Compared Methods

In order to validate the effectiveness of DLR-GAE, we utilize several widely used graph datasets for the performance evaluation, including Citeseer, CoraFull, BlogCatalog, ACM, Flickr and UAI. These datasets describe distinct types of node connections, e.g., paper citations, social relationships and web linkages. A statistical summary of these datasets is illustrated in Table 1.

We compare the proposed DLR-GAE with various graph-based methods. Apart from classical baselines (MLP and Chebyshev (Defferrard, Bresson, and Vandergheynst 2016)), other state-of-the-art methods are: GCN (Kipf and Welling 2017), GAE (Kipf and Welling 2016), GraphSAGE (Hamilton, Ying, and Leskovec 2017), GAT (Velickovic et al. 2018), JK-Net (Xu et al. 2018), SGC (Wu et al. 2019), APPNP (Klicpera, Bojchevski, and Günnemann 2019), ClusterGCN (Chiang et al. 2019), ScatteringGCN (Min, Wenkel, and Wolf 2020), SSGC (Zhu and Koniusz 2021), AdaGCN (Sun, Zhu, and Lin 2021) and AMGCN (Wang et al. 2020). In particular, AMGCN also considers both semantic and topological node relationships.

### Experimental Settings

In order to prevent undesired influence raised by the data distribution, in the following experiments, we shuffle datasets and randomly select 20 labeled samples per class for training, 500 samples for validation and 1,000 samples for testing. In particular, because some categories of CoraFull do not have enough samples, we randomly select a fixed ratio of samples from each category, resulting in about 1,400 training samples from 70 classes. In pursuit of providing a fair test bed, we list some common hyperparameters used in experiments. Learning rates of all compared frameworks are fixed as 0.01. For all GNN-based methods, the number of hidden units at each layer is fixed as 16 and a 2-layer GCN is adopted. As for DLR-GAE, we keep consistent with compared GCN-based models, setting the number of hidden units as 16 and applying parallel two-layer GCNs. The learning rate of DLR-GAE is also fixed as 0.01 and the weight decay is  $5 \times 10^{-4}$ . The choice of  $k$  ranges in  $[5, 10, \dots, 50]$  when constructing semantic graphs via KNN algorithm. The number of latent factors  $I$  is fixed as 5.

### Experimental Results

**Performance Comparison.** The semi-supervised classification comparison between DLR-GAE and compared state-

Methods / Datasets	Citeseer	CoraFull	BlogCatalog	ACM	Flickr	UAI
MLP	0.366	0.051	0.646	0.812	0.431	0.188
Chebyshev	0.693	0.534	0.357	0.829	0.304	0.215
GCN	0.697	0.567	0.697	0.875	0.414	0.498
GAE	0.719	0.576	0.715	0.899	0.477	0.387
GraphSAGE	0.620	0.521	0.525	0.872	0.286	0.483
GAT	0.683	0.571	0.681	0.889	0.429	0.597
JK-Net	0.684	0.568	0.725	0.892	0.547	0.494
SGC	0.697	0.583	0.716	0.887	0.410	0.571
APNP	0.698	0.576	0.813	0.885	0.477	0.538
ClusterGCN	0.681	0.576	0.731	0.893	0.483	0.525
ScatteringGCN	0.679	0.519	0.690	0.890	0.419	0.364
SSGC	0.673	0.572	0.760	0.889	0.478	0.523
AdaGCN	0.663	0.552	0.800	0.894	0.552	0.588
AMGCN	0.719	0.584	0.858	0.899	0.756	0.643
<b>DLR-GAE</b>	<b>0.724</b>	<b>0.590</b>	<b>0.901</b>	<b>0.924</b>	<b>0.810</b>	<b>0.658</b>

Table 2: Semi-supervised classification accuracy comparison on six graph datasets.

Methods / Datasets	Flickr	UAI
GAE (topological graph)	0.477	0.497
GAE (semantic graph)	0.757	0.614
DLR-GAE w/o $\mathbf{U}_s$	0.741	0.630
DLR-GAE w/o $\mathcal{L}_{sp}(\mathbf{U}_s)$	0.783	0.636
DLR-GAE w/o constraints in Eq. (25)	0.562	0.632
<b>DLR-GAE</b>	<b>0.810</b>	<b>0.658</b>

Table 3: Ablation study (accuracy) of DLR-GAE.

of-the-art models is shown in Table 2. DLR-GAE gains the optimal classification accuracy on all datasets. It not only achieves favorable performance compared with baseline GCN methods (e.g., GCN and GAE), but also obtains higher accuracy than other latest works. This can be attributed to the more dexterous information propagation over two types of graphs. In particular, DLR-GAE outperforms AMGCN which also endeavors to explore node embeddings from topology and feature spaces. This indicates that the strategy of learning shared latent factor for reconstruction adjacency matrix attains remarkable performance improvement, compared with that learns shared weights of GCN.

**Ablation Study.** To verify the effectiveness of each model component, the ablation study of DLR-GAE is conducted, as demonstrated in Table 3. We investigate the performance of GAE with a topological or a semantic graph, and also check the classification accuracy of DLR-GAE without some essential components. From this table, we can draw the following conclusions. First of all, DLR-GAE gains the optimal performance on these datasets, which is more significant on Flickr. Second, DLR-GAE that utilizes both topological and semantic graphs obtains encouraging accuracy increment, compared with GAE that only adopts either topological graph or semantic graph. This validates that the co-utilization of topological relationship and feature homophily

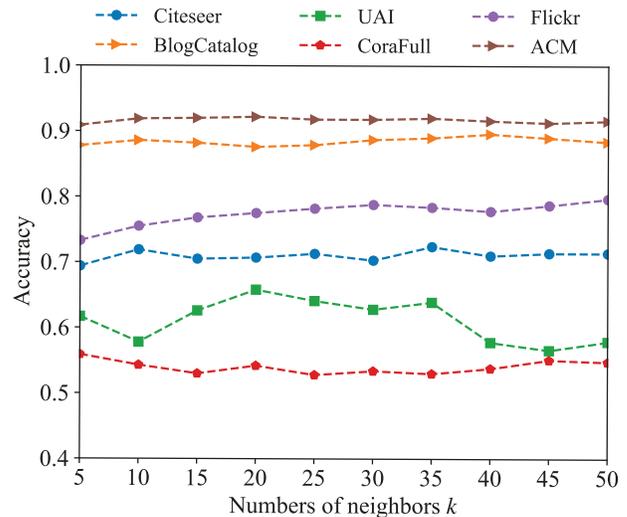


Figure 3: Impact of neighbor number  $k$  when constructing semantic graphs from node features.

facilitates the node embedding learning. Third, we test the performance of DLR-GAE without some components, finding that the learned latent factor matrix  $\mathbf{U}_s$ , low-rank loss  $\mathcal{L}_{sp}(\mathbf{U}_s)$ , and the constraints w.r.t.  $\mathbf{U}_s$  promote the accuracy of the model. It is noted that DLR-GAE without constraints in Eq. (25) has poor performance on Flickr, which indicates that the validity of  $\mathbf{U}_s$  is essential to DLR-GAE.

**Parameter Sensitivity.** In this subsection, we investigate the impact of hyperparameters ( $k$ ,  $\alpha$  and  $\gamma$ ) used in the proposed DLR-GAE. First, Figure 3 presents the influence of neighbor number  $k$  when we construct semantic graphs from node features. In general, a small  $k$  value (e.g., 5) leads to satisfactory accuracy and the performance fluctuates marginally as  $k$  changes. On most datasets, a larger  $k$  may result in higher accuracy, indicating that more neigh-

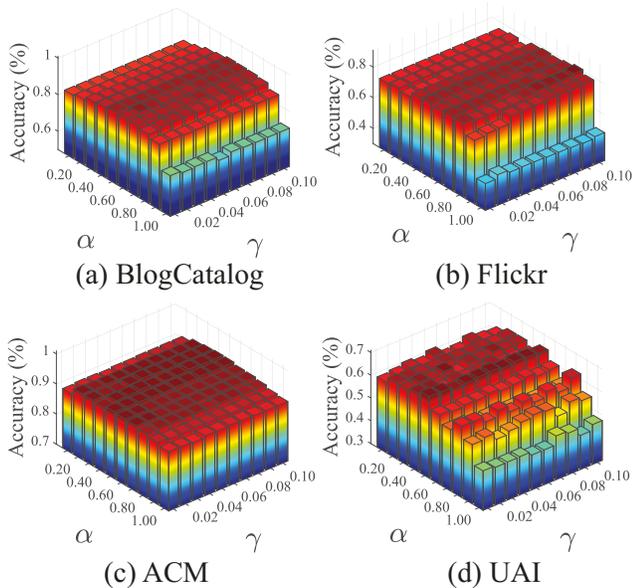


Figure 4: Parameter sensitivity of DLR-GAE w.r.t.  $\alpha$  and  $\gamma$ .

bors discovered from the feature homophily facilitate the model to mine more favorable information. Besides, it can be observed that excessive neighbors do not contribute to performance improvement and sometimes even lead to accuracy decline. This is because that a dense graph may contain more noises that imperil the relationships between nodes.

Next, we analyze the impact of hyperparameters  $\alpha$  and  $\gamma$  in the loss function  $\mathcal{L}(\mathbf{Z}_{\mathcal{T}}, \mathbf{Z}_{\mathcal{S}}, \mathbf{A}, \mathbf{U}_s, \mathbf{Y})$ . It can be seen that the accuracy fluctuates slightly on most datasets when  $\gamma$  is changing. As for the impact of  $\alpha$ , experimental results reveal that DLR-GAE achieves poor performance when  $\alpha = 1$ . Namely, DLR-GAE encounters significant accuracy decline when no feature graph is adopted, especially on BlogCatalog, Flickr and UAI. This phenomenon indicates that the utilization of feature graphs facilitates the accuracy remarkably. In detail, the proposed DLR-GAE generally gains better performance when a proper percentage of node embeddings yielded by semantic graphs is applied.

**Visualization of Latent Factor Matrix.** In pursuit of verifying the learned shared latent factor matrix  $\mathbf{U}_s$ , we visualize it in Figure 5<sup>1</sup>. The figure suggests that the learned  $\mathbf{U}_s$  is symmetric and nonnegative. The cumulative sum of each row or column of  $\mathbf{U}_s$  is 1. These observations prove that the proposed DLR-GAE ensures that all constraints of  $\mathbf{U}_s$  are satisfied and the projection operations succeed in mapping  $\mathbf{U}_s$  onto the feasible set during the network training, which facilitates the model to gain more favorable representations of shared latent factors.

**Convergence Analysis.** Finally, we examine the convergence of the proposed model, as exhibited in Figure 6. Experimental results show that the loss values plunge and eventually converge on all tested datasets. The accuracy of the

<sup>1</sup>Zero values shown in the latent factor matrices are elements with absolute values smaller than  $10^{-2}$ .

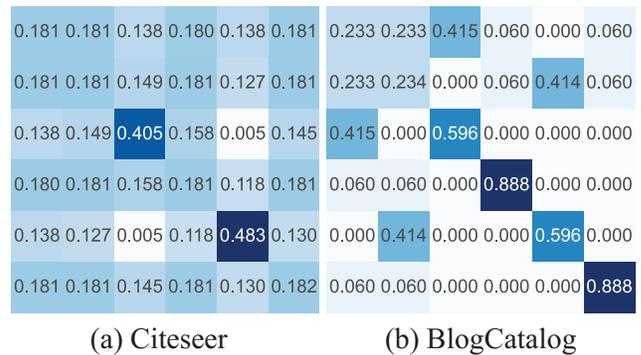


Figure 5: Visualization of learned shared factor matrices  $\mathbf{U}_s$ .

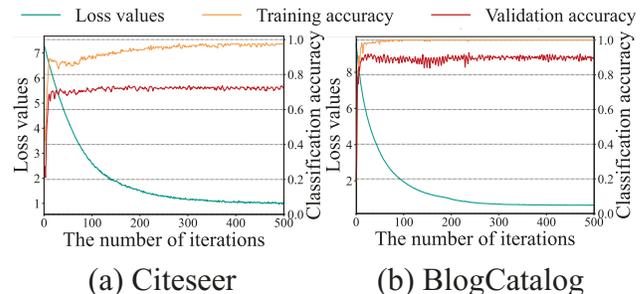


Figure 6: Loss and accuracy curves of DLR-GAE.

training set and the validation set also rises as the number of iterations increases, and begins to fluctuate later in the training. In order to obtain the optimal model with better generalization and robustness, we select the model with the highest validation accuracy for the model test.

## Conclusion

In this paper, we proposed a graph convolutional network dubbed Dual Low-Rank Graph AutoEncoders (DLR-GAE), which explored distinctive node embeddings from topology and feature space. DLR-GAE encoded node embeddings simultaneously via a parallel low-rank autoencoder. The decoders conducted low-rank adjacency matrix reconstruction via a surrogate objective of optimization w.r.t. Schatten- $p$  norm, which shared the same low-rank latent factor matrix. To ensure the validity of this matrix, several constraints were added and we developed relevant solutions to them. The proposed model was utilized to carry out semi-supervised classification tasks, and substantial experimental results revealed the favorable performance of DLR-GAE. In the future study, we will devote ourselves to further investigation of multi-channel graph neural networks.

## Acknowledgments

This work is in part supported by the National Natural Science Foundation of China (Grant Nos. U21A20472 and 62276065), the Natural Science Foundation of Fujian Province (Grant No. 2020J01130193).

## References

- Bian, T.; Xiao, X.; Xu, T.; Zhao, P.; Huang, W.; Rong, Y.; and Huang, J. 2020. Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 549–556.
- Chen, Z.; Yao, J.; Xiao, G.; and Wang, S. 2023. Efficient and Differentiable Low-rank Matrix Completion with Back Propagation. *IEEE Transactions on Multimedia*, 25: 228–242.
- Chiang, W.; Liu, X.; Si, S.; Li, Y.; Bengio, S.; and Hsieh, C. 2019. Cluster-GCN: An Efficient Algorithm for Training Deep and Large Graph Convolutional Networks. In *Proceedings of the 35th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 257–266.
- Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in Neural Information Processing Systems*, volume 29, 3837–3845.
- Feng, F.; He, X.; Zhang, H.; and Chua, T.-S. 2021. Cross-GCN: Enhancing Graph Convolutional Network with k-Order Feature Interactions. *IEEE Transactions on Knowledge and Data Engineering*. Doi: 10.1109/TKDE.2021.3077524.
- Gao, H.; Pei, J.; and Huang, H. 2019. Conditional Random Field Enhanced Graph Convolutional Neural Networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 276–284.
- Hamilton, W. L.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems*, volume 30, 1024–1034.
- Jing, K.; Xu, J.; and Li, P. 2022. Graph Masked Autoencoder Enhanced Predictor for Neural Architecture Search. In Raedt, L. D., ed., *Proceedings of the 31st International Joint Conference on Artificial Intelligence*, 3114–3120.
- Kang, M.; Lee, K.; Lee, Y. H.; and Suh, C. 2020. Autoencoder-Based Graph Construction for Semi-supervised Learning. In *Proceedings of the 16th European Conference on Computer Vision*, volume 12369, 500–517.
- Kipf, T. N.; and Welling, M. 2016. Variational Graph Auto-Encoders. *ArXiv preprint*, abs/1611.07308.
- Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*.
- Klicpera, J.; Bojchevski, A.; and Günnemann, S. 2019. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *Proceedings of the 7th International Conference on Learning Representations*.
- Liu, X.; Ji, Z.; Pang, Y.; Han, J.; and Li, X. 2022. DGIG-Net: Dynamic Graph-in-Graph Networks for Few-Shot Human-Object Interaction. *IEEE Transactions on Cybernetics*, 52(8): 7852–7864.
- Liu, Y.; Xie, D.; Gao, Q.; Han, J.; Wang, S.; and Gao, X. 2019. Graph and Autoencoder Based Feature Extraction for Zero-shot Learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 3038–3044.
- Min, Y.; Wenkel, F.; and Wolf, G. 2020. Scattering GCN: Overcoming Oversmoothness in Graph Convolutional Networks. In *Advances in Neural Information Processing Systems*, volume 33, 14498–14508.
- Sánchez-Martín, P.; Rateike, M.; and Valera, I. 2022. VACA: Designing Variational Graph Autoencoders for Causal Queries. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, 8159–8168.
- Sun, K.; Zhu, Z.; and Lin, Z. 2021. AdaGCN: Adaboosting Graph Convolutional Networks into Deep Models. In *Proceedings of the 9th International Conference on Learning Representations*.
- Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *Proceedings of the 6th International Conference on Learning Representations*.
- Wang, X.; Zhu, M.; Bo, D.; Cui, P.; Shi, C.; and Pei, J. 2020. AM-GCN: Adaptive Multi-channel Graph Convolutional Networks. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1243–1253.
- Wu, F.; Jr., A. H. S.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. Q. 2019. Simplifying Graph Convolutional Networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, 6861–6871.
- Xu, C.; Lin, Z.; and Zha, H. 2017. A unified convex surrogate for the Schatten- $p$  norm. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 926–932.
- Xu, K.; Li, C.; Tian, Y.; Sonobe, T.; Kawarabayashi, K.; and Jegelka, S. 2018. Representation Learning on Graphs with Jumping Knowledge Networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, 5449–5458.
- Zeng, X.; Liao, R.; Gu, L.; Xiong, Y.; Fidler, S.; and Urtasun, R. 2019. DMM-Net: Differentiable Mask-Matching Network for Video Object Segmentation. In *2019 IEEE/CVF International Conference on Computer Vision*, 3928–3937.
- Zhang, Z.; Chen, C.; Chang, Y.; Hu, W.; Xing, X.; Zhou, Y.; and Zheng, Z. 2021. SHNE: Semantics and Homophily Preserving Network Embedding. *IEEE Transactions on Neural Networks and Learning Systems*. Doi: 10.1109/TNNLS.2021.3116936.
- Zhu, H.; and Koniusz, P. 2021. Simple Spectral Graph Convolution. In *Proceedings of the 9th International Conference on Learning Representations*.