Graph embedding clustering

Xu, Huiling; Xia, Wei; Gao, Quanxue; Han, Jungong; Gao, Xinbo

Published in:
Neural Networks

DOI:
10.1016/j.neunet.2021.05.008

Publication date:
2021

Citation for published version (APA):
Xu, H., Xia, W., Gao, Q., Han, J., & Gao, X. (2021). Graph embedding clustering: Graph attention auto-encoder with cluster-specificity distribution. *Neural Networks*, *142*, 221-230. https://doi.org/10.1016/j.neunet.2021.05.008

Neural Networks

Graph Embedding Clustering: Graph Attention Auto-encoder With Cluster-Specificity Distribution

--Manuscript Draft--

Manuscript Number:	NEUNET-D-20-01307R2
Article Type:	Article
Section/Category:	Learning Systems
Keywords:	Nodes clustering; graph neural networks; cluster-specificity distribution.
Corresponding Author:	quanxue Gao State Key Laboratory of Integrated Services Networks Xi'an, Shaan xi CHINA
First Author:	Huiling Xu
Order of Authors:	Huiling Xu
	Wei Xia
	quanxue Gao
	Jungong Han
	Xinbo Gao
Abstract:	Towards exploring the topological structure of data, numerous graph embedding clustering methods have been developed in recent years, none of them takes into account the cluster-specificity distribution of the nodes representations, resulting in suboptimal clustering performance. Moreover, most existing graph embedding clustering methods execute the nodes representations learning and clustering in two separated steps, which increases the instability of its original performance. Additionally, rare of them simultaneously takes node attributes reconstruction and graph structure reconstruction into account, resulting in degrading the capability of graph learning. In this work, we integrate the nodes representations learning and clustering into a unified framework, and propose a new deep graph attention auto-encoder for nodes clustering that attempts to learn more favorable nodes representations by leveraging self-attention mechanism and node attributes reconstruction. Meanwhile, a cluster-specificity distribution constraint, which is measured by \ell_{1,2}-norm, is employed to make the nodes representations within the same cluster end up with a common distribution in the dimension space while representations with different clusters have different distributions in the intrinsic dimensions. Extensive experiment results reveal that our proposed method is superior to several state-of-the-arts in terms of performance.orm, is employed to make the nodes representations within the same cluster end up with a common distribution in the dimension space while representations with different clusters have different distributions in the intrinsic dimensions. Extensive experiment results reveal that our proposed method is superior to several state-of-the-arts in terms of performance.

Response to 'NEUNET-D-20-01307R1'

We would like to thank the associate editor and anonymous reviewers for their constructive suggestions. It is these constructive suggestions that remarkably improve our understanding and the quality of this manuscript. According to the suggestion of AE, we revised our paper and resubmit it. According to these constructive suggestions, we have carefully revised our paper. Below is our detailed response to the reviewers.

Response to AE:

This paper can be accepted for publication after correcting English mistakes in the text.

Response: We sincerely thank AE and the reviewer for these constructive suggestions which help to remarkably improve the quality of the paper. In our revised paper, we double checked the paper and tried our best to correct some grammar errors and typos **(See Section 1, Section 2, Section 3 and Section 4 in our revised paper)**.

Response to Reviewers #1:

1. All of the previous questions have been addressed in the current document. Therefore, I just recommend a final text review. See below some examples:

- Line 20: should read "They still have" instead of "they still has";
- Line 175: should read "indicates" instead of "indicts";.

Response: We sincerely thank the reviewer for these constructive suggestions which help to remarkably improve the quality of the paper. According to your advice, we revised these grammar errors. Meanwhile, we double checked our manuscript, and revised some other grammar errors and typos **(See Section 1, Section 2, Section 3 and Section 4 in our revised paper)**. Again, thank you very much for your appreciation for our work.

Response to Reviewers #2:

I would like to thank the authors for their point-by-point response to the reviewers' comments.

The quality of the manuscript has considerably improved after the revision.

Response: We sincerely thank the reviewer for this constructive advice which help to remarkably improve the quality of the paper. Again, thank you very much for your appreciation for our work.

Title Page (With all author details listed)

Graph Embedding Clustering: Graph Attention Auto-encoder With Cluster-Specificity Distribution

Huiling Xu[a], Wei Xia[a,*], Quanxue Gao[a,*], Jungong Han[c] , Xinbo Gao[b, d]

[a] State Key Laboratory of Integrated Services Networks, Xidian University, Shaanxi 710071, China.

[b] School of Electronic Engineering, Xidian University, Shaanxi 710071, China.

[c] Computer Science Department, Aberystwyth University, SY23 3FL, United Kingdom.

[d] Chongqing Key Laboratory of Image Cognition, Chongqing University of Posts and Telecommunications, Chongqing 400065, China.

[*]Corresponding author: Quanxue Gao (Email: qxgao@xidian.edu.cn), xdweixia@gmail.com (Wei Xia)

Abstract

Towards exploring the topological structure of data, numerous graph embedding clustering methods have been developed in recent years, none of them takes into account the cluster-specificity distribution of the nodes representations, resulting in suboptimal clustering performance. Moreover, most existing graph embedding clustering methods execute the nodes representations learning and clustering in two separated steps, which increases the instability of its original performance. Additionally, rare of them simultaneously takes node attributes reconstruction and graph structure reconstruction into account, resulting in degrading the capability of graph learning. In this work, we integrate the nodes representations learning and clustering into a unified framework, and propose a new deep graph attention auto-encoder for nodes clustering that attempts to learn more favorable nodes representations by leveraging self-attention mechanism and node attributes reconstruction. Meanwhile, a cluster-specificity distribution constraint, which is measured by $\ell_{1,2}$-norm, is employed to make the nodes representations within the same cluster end up with a common distribution in the dimension space while representations with different clusters have different distributions in the intrinsic dimensions. Extensive experiment results reveal that our proposed method is superior to several state-of-the-arts in terms of performance.

Keywords: Nodes clustering; graph neural networks; cluster-specificity distribution.

Biography of the authors

Huiling Xu received the B.E degree in communication engineering from Wuhan University of Technology, Wuhan, China, in 2019. She is currently pursuing the master's degree in information and communication engineering under the supervision of Prof. Q. Gao in Xidian University. Her research interests include machine learning and pattern recognition.

Wei Xia received the B.Eng. degree in Communication Engineering from Lanzhou University of Technology, Lanzhou, China, in 2018. He is currently pursuing the Ph.D. degree in communication and information system in Xidian University, Xi'an, China. His research interests include pattern recognition, machine learning and deep learning.

Quanxue Gao received the B. Eng. degree from Xi'an Highway University, Xi'an, China, in 1998, the M.S. degree from the Gansu University of Technology, Lanzhou, China, in 2001, and the Ph.D. degree from Northwestern Polytechnical University, Xi'an China, in 2005. He was an associate research with the Biometrics Center, The Hong Kong Polytechnic University, Hong Kong from 2006 to 2007. From 2015 to 2016, he was a visiting scholar with the department of computer science, The University of Texas at Arlington, Arlington USA. He is currently a professor with the School of Telecommunications Engineering, Xidian University, and also a key member of State Key Laboratory of Integrated Services Networks. He has authored 60 technical articles in refereed journals and proceedings, including IEEE Transactions on Image Processing, IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Cybernetics, CVPR, AAAI, and IJCAI. His

current research interests include pattern recognition and machine learning.

Jungong Han is Professor of Computer Science Department at Aberystwyth University, UK. In the past 15 years, he has been continuously conducting research in the fields of video analysis, computer vision and machine learning, and has published over 150 articles in leading journals and prestigious conferences, in which one of the first authored papers has been cited for more than 1000 times. Dr. Han is the member of the editorial board of several international journals, such as Elsevier Neurocomputing, Springer Multimedia Tools and Applications and IET Computer Vision, and has been (lead) Guest Editors for IEEE Transactions on Neural Networks and Learning Systems and IEEE Transactions on Cybernetics.

Xinbo Gao received the B.Eng., M.Sc. and Ph.D. degrees in electronic engineering, signal and information processing from Xidian University, Xi'an, China, in 1994, 1997, and 1999, respectively. From 1997 to 1998, he was a research fellow at the Department of Computer Science, Shizuoka University, Shizuoka, Japan. From 2000 to 2001, he was a post-doctoral research fellow at the Department of Information Engineering, the Chinese University of Hong Kong, Hong Kong. Since 2001, he has been at the School of Electronic Engineering, Xidian University. He is currently a Cheung Kong Professor of Ministry of Education of P. R. China, a Professor of Pattern Recognition and Intelligent System of Xidian University and a Professor of Computer Science and Technology of Chongqing University of Posts and Telecommunications. His current research interests include Image processing, computer

vision, multimedia analysis, machine learning and pattern recognition. He has published six books and around 300 technical articles in refereed journals and proceedings. Prof. Gao is on the Editorial Boards of several journals, including Signal Processing (Elsevier) and Neurocomputing (Elsevier). He served as the General Chair/Co-Chair, Program Committee Chair/Co-Chair, or PC Member for around 30 major international conferences. He is a Fellow of the Institute of Engineering and Technology and a Fellow of the Chinese Institute of Electronics.

Graph Embedding Clustering: Graph Attention Auto-encoder With Cluster-Specificity Distribution

Huiling Xu[a], Wei Xia[a,*], Quanxue Gao[a,*], Jungong Han[c], Xinbo Gao[b,d]

[a]*State Key Laboratory of Integrated Services Networks, Xidian University, Shaanxi 710071, China.*
[b]*School of Electronic Engineering, Xidian University, Shaanxi 710071, China.*
[c]*Computer Science Department, Aberystwyth University, SY23 3FL, United Kingdom.*
[d]*Chongqing Key Laboratory of Image Cognition, Chongqing University of Posts and Telecommunications, Chongqing 400065, China.*

Abstract

Towards exploring the topological structure of data, numerous graph embedding clustering methods have been developed in recent years, none of them takes into account the cluster-specificity distribution of the nodes representations, resulting in suboptimal clustering performance. Moreover, most existing graph embedding clustering methods execute the nodes representations learning and clustering in two separated steps, which increases the instability of its original performance. Additionally, rare of them simultaneously takes node attributes reconstruction and graph structure reconstruction into account, resulting in degrading the capability of graph learning. In this work, we integrate the nodes representations learning and clustering into a unified framework, and propose a new deep graph attention auto-encoder for nodes clustering that attempts to learn more favorable nodes representations by leveraging self-attention mechanism and node attributes reconstruction. Meanwhile, a cluster-specificity distribution constraint, which is measured by $\ell_{1,2}$-norm, is employed to make the nodes representations within the same cluster end up with a common distribution in the dimension space while representations with different clusters have different distributions in the intrinsic dimensions. Extensive experiment results reveal that our proposed method is superior to several state-of-the-arts in terms of performance.

Keywords: Nodes clustering; graph neural networks; cluster-specificity distribution.

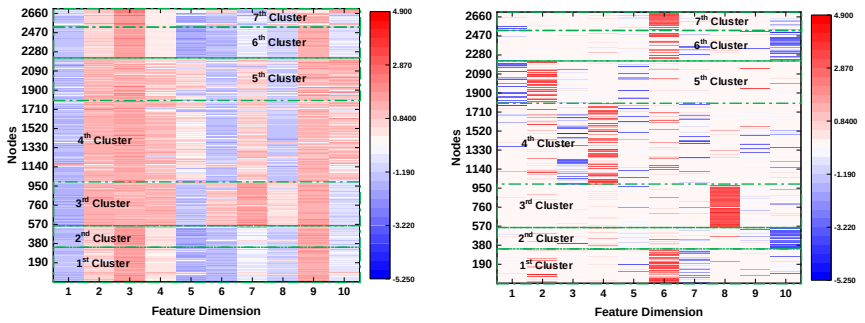
*Corresponding author
Email addresses: xdweixia@gmail.com (Wei Xia), qxgao@xidian.edu.cn (Quanxue Gao)

1. Introduction

As we enter the era of Internet data, the graph-structured data is ubiquitous, *e.g.*, the data comes from social media and citation networks. Most of the data that people usually obtain is unlabeled. However, numerous intelligent methods require labeled data to train deep neural networks, such as classification and prediction tasks. To fill this gap, clustering has emerged. The goal of clustering is to divide data into some disjoint groups such that the data in the common group are similar to each other, while data in different groups have low similarity. Compared with the conventional learning methods [1, 2, 3, 4, 5, 6, 7] which mainly investigate Euclidean structure data, such as face data, handwritten digit and object, graph convolutional networks (GCNs) [8, 9, 10, 11, 12, 13, 14] can better handle such graph-structured data, *i.e.,* non-Euclidean structure data, this is because that GCNs can provide powerful node representations via preserving the topological structure [15, 16, 17, 18]. In this paper, we aim to present a new graph convolutional solution to the nodes clustering task, which help label graph structure data, thereby facilitating the development of deep neural network.

To date, several graph convolutional auto-encoder based clustering models have been proposed [10, 19, 20], at the core of which is to learn the low-dimensional, compact and continuous representations, then they implement classical clustering methods, *e.g.*, K-Means [21], on the learned representations to obtain clustering labels. Despite the impressive clustering performance, they still have the following limitations.

1. They all neglect the cluster-specificity distribution (CSD) of nodes representations. In fact, different clusters are distributed in different dimensions of the feature dimension, *i.e.*, the values of features with the same cluster are large in the corresponding dimensions of the cluster, and the values in other dimensions, which correspond to other clusters, are small or zero. As shown in **Fig**. 1, this significant property, which is called CSD, is very important to learn more robust nodes representations. ***To the best of our knowledge, similar investigations for nodes clustering have been found lacking so far, which is one of the motiva-***



(a) Without CSD constraint (b) With CSD constraint

Figure 1: Illustration of cluster-specificity distribution on **Cora** dataset. (a) When without CSD constraint, the nodes'features are distributed in almost 10 dimensions and they are very messy. Even in most dimensions, the features of the nodes are remarkably similar, which may make the algorithm to cluster them all into the same class. By contrast, in (b), the nodes'features are more discriminative under CSD constraint. For example, the samples, which belong to the 1-st cluster, are mainly distributed in the 6-th and 7-th dimensions. Thus, CSD constraint helps the algorithm to divide the samples.

tions behind this work. Thus, to well exploit the cluster structure, we should take CSD constraint into account, while existing methods do not.

2. Most of them fail to simultaneously reconstruct node attributes and graph structure, resulting in suboptimal nodes representations.

3. Most of them execute the nodes representations learning and clustering in two separated steps, thus the networks cannot be trained for nodes clustering via an end to end manner, which limits its performance.

To well exploit cluster structure, inspired by these insight analysis, we propose a nodes clustering method, namely, **G**raph **E**mbedding **C**lustering: Graph Attention Auto-encoder With **C**luster-**S**pecificity **D**istribution (GEC-CSD). Specifically, to make the decoder part learnable, node attributes reconstruction term is introduced. Moreover, to eliminate the uncertainty of the post-processing clustering operation, we introduce the cluster activation function which help to integrate the nodes representations learning and node clustering in a unified framework. Meanwhile, we employ $\ell_{1,2}$-norm penalty to exploit the CSD of nodes representations. We integrate the afore-

mentioned concerns into a unified optimization framework. Extensive experiments on three datasets are conducted to demonstrate the superiority of our proposed GEC-CSD over state-of-the-art methods. The main contributions of this paper are summarized as follows:

1. We propose a novel deep graph convolutional embedding clustering model based on graph attention auto-encoder which joints nodes representations learning and clustering into a unified framework. Thus, the learned node representations not only well encode data information but also well characterize cluster structure.

2. We find that $\ell_{1,2}$-norm has an important role of characterizing the CSD of graph structured data in dimension space, and then apply it to learn nodes representations, which well characterizes cluster structure and consequently boosts clustering results.

3. Both node attributes reconstruction and node neighbours importance are considered in clustering, which can help to well embed graph structure, thereby learning more favorable nodes representations.

2. Related Work

Nodes clustering is one of the fundamental and active topics in unsupervised learning. Over the years, studies have proposed a large number of nodes clustering method, which can be roughly divided into three classes: probabilistic models, matrix factorization-based methods, and deep learning-based methods. To be specific, probabilistic models target at learning graph embedding via extracting different patterns or walks from the graph. For example, Perozzi *et al.* [22] proposed the DeepWalk model, in which the captured walks include global structural equivalence and local neighborhood connectivity. Matrix factorization-based methods aims to obtain low dimension embedding by decomposing the adjacency matrix, *e.g.*, Wang *et al.* [23] modularized nonnegative matrix factorization (M-NMF), Yang *et al.* [24] text-associated DeepWalk (TADW).

To well exploit deep nonlinear representation, deep learning-based methods are proposed [25, 26, 19, 20]. One of the most representative methods is graph auto-

encoder (GAE) [20]. It encodes graph structure and node attribute to a node representation, on which a decoder is trained to reconstruct the graph structure. To improve the robustness of node representation, Pan *et al.* proposed adversarial regularized graph auto-encoder (ARGAE). However, in above methods, the neighbors of each node carry the same weight without considering the existence of noise in the graph structure. To better mine the relevance of nodes and their neighbors, Velickovic *et al.* [27] proposed graph attention networks (GATs), however, their method is designed to reconstruct graph structure instead of node attributes, in which the graph structure can not be used at all in the decoder part, resulting in degrading the capability of graph learning. To remedy this situation, Salehi *et al.* [28] proposed an improved graph-based encoder with the self-attention mechanism (GATE) integrated, which reconstructs graph structured inputs, including both node attributes and the graph structure.

Although aforementioned methods work well in most cases, they require post-processing operation to obtain clustering labels. To this end, Wang *et al.* [9] proposed deep attentional graph embedding clustering approach (DAEGC). Tao *et al.* [29] proposed another graph embedding clustering methods. Despite achieving remarkable progress, the decoder part is not learnable. To solve this problem, Park *et al.* [11] proposed a symmetric graph convolutional auto-encoder to node clustering (GALA). Pan *et al.* [12] proposed the improved ARVGAE with simultaneous reconstructing the graph structure and node attribute (ARVGA-AX). Kou *et al.* [30] proposed the self-supervised node clustering model via preserving latent distribution.

However, all of them fail to take CSD constraint into account. To well exploit cluster structure, motivated by the great success of the $\ell_{1,2}$-norm [31] on feature selection [32, 33, 34], classification [35] and unsupervised learning [36], we study the graph auto-encoder and propose a novel graph attention auto-encoder with the cluster-specificity (GEC-CSD) constraint. Our proposed GEC-CSD integrates nodes representations learning and clustering into an end-to-end framework. Meanwhile, we make the decoder part learnable via introducing node attribute reconstruction. Thus, the learned nodes representations well capture the cluster structure and are more suitable for downstream clustering task.

Notations. For convenience, the adjacency matrix of a graph is represented by

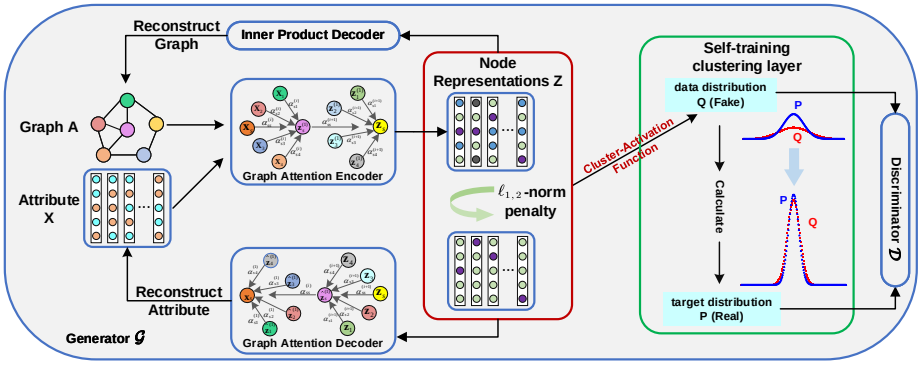


Figure 2: The overall framework of our proposed GEC-CSD. For the given node attribute $\mathbf{X}$ and the corresponding graph structure $\mathbf{A}$, GEC-CSD jointly learns node representations and performs nodes clustering in two successive steps, one is a graph attention auto-encoder with cluster-specificity distribution constraint which simultaneously maps $\mathbf{X}$ and $\mathbf{A}$ into a latent space to get node representations $\mathbf{Z}$, and the other is a self-training clustering layer with discriminator $\mathcal{D}$ which minimizes the mismatch between different distributions of $\mathbf{Z}$ w.r.t. cluster centers. In GEC-CSD, we use the graph attention auto-encoder and the self-training clustering layer together as a generator $\mathcal{G}$ to produce different distributions of $\mathbf{Z}$ w.r.t. cluster centers.

$\mathbf{A} \in \mathbb{R}^{N\times N}$, in which $\mathbf{A}_{Sj} = 1$ indicates there is an edge between the S-th node and j-th node ($S = j$ is present). Suppose that we aim to cluster N input nodes $\{\mathbf{x}_S^{1\times d_1} \in \mathbf{X}\}_{S=1}^N$ into $\mathcal{C}$ clusters. Then the graph attention encoder of generator $\mathcal{G}$ maps raw nodes attributes and graph structure to their corresponding latent representations $\{\mathbf{z}_S^{1\times d_3} \in \mathbf{Z}\}_{S=1}^N$, where d_i denotes the dimension of i-th layer. The reconstructed node content is represented by $\hat{\mathbf{X}}$. The m-th row of a matrix $\mathbf{B} \in \mathbb{R}^{d\times k}$ is represented by $\mathbf{b}_m$. The Frobenius norm of $\mathbf{B}$ is $\|\mathbf{B}\|_F = \sqrt{\sum_{m=1}^{d}\sum_{j=1}^{k} b_{mj}^2}$. The $\ell_{1,2}$-norm of $\mathbf{B}$ is $\|\mathbf{B}\|_{1,2}^2 = \sum_{m=1}^{d}\|\mathbf{b}_m\|_1^2 = \sum_{m=1}^{d}\left(\sum_{j=1}^{k}|b_{mj}|\right)^2$. $\mu_t(t = 1, \cdots, \mathcal{C})$ represents the centroid of t-th cluster.

3. The Proposed Method

Now, we detail our proposed GEC-CSD. Firstly, we explain the network formulation, then present each component, and finally we describe the implementation process.

3.1. Formulation

As aforementioned, graph embedding clustering methods [10, 29] use learnable graph embedding process to divide input nodes. Although they seamlessly integrate graph structure and node attributes, they fail to take the importance of neighbor nodes into account, thus giving rise to poor performance in the existence of noise. To remedy this situation, GATs [27] adopts a learnable attention mechanism to enable specifying different weights to different nodes in a neighborhood. However, above methods are designed to reconstruct the graph structure $\mathbf{A}$ instead of node attributes $\mathbf{X}$. In this case, the decoder part cannot be learnable, resulting in degrading the capability of graph learning. Recently, GATE [28] model was proposed, which is able to reconstruct both node attributes and graph structure. Unfortunately, GATE executes the nodes representations and nodes clustering in two separated steps, which limits its performance.

To tackle the above vital issues simultaneously within a unified framework, we propose a novel graph embedding clustering model via unsupervised discriminant feature extraction, which learns more discriminative representations for node clustering by graph attention auto-encoder with the $\ell_{1,2}$-norm penalty, and we introduce adversarial regularizer to complement node representations distribution for better node clustering. Fig. 2 illustrates the overall architecture of our proposed GEC-CSD, which consists of a node latent representations (and self-training clustering) generator $\mathcal{G}$ and a discriminator $\mathcal{D}$.

Specifically, our proposed GEC-CSD utilizes a two-layer non-linear graph attention encoder of generator $\mathcal{G}$ to map the raw node attribute matrix $\mathbf{X}$ and graph structure $\mathbf{A}$ to get the latent node representations $\mathbf{Z}$. To ensure the learned representations $\mathbf{Z}$ maintain the local structure of both node attributes and graph structure, we leverage an inner product decoder and a learnable graph attention decoder to reconstruct both the graph structure $\mathbf{A}$ and node attribute $\mathbf{X}$, respectively.

In order to make the learned node latent representations $\mathbf{Z}$ more meaningful so that it can help the downstream node clustering task, we introduce the combination of t-SNE algorithm [37] and adversarial learning as a novel and complementary unsupervised graph embedding clustering solution. Concretely, let distribution $\mathbf{P}$ be the target (ideal) distribution of the node representations $\mathbf{Z}$. When the obtained cluster μ_t is not accurate

due to the complex data structure, we will learn an actual data distribution $\mathbf{Q}$ (obtained via Eq. (6), a special cluster activation function) of $\mathbf{Z}$. According to the thought of t-SNE, if the data in the two spaces are similar, then the corresponding distribution should be the same. In other words, if the current cluster μ_t is inappropriate, there will be a gap between $\mathbf{Q}$ and $\mathbf{P}$.

The target of our proposed GEC-CSD is to eliminate such difference between the actual data distribution $\mathbf{Q}$ (generated from noisy μ_t and node representations $\mathbf{Z}$) and ideal target distribution $\mathbf{P}$. Consequently, we use the squared F-norm based constraint as clustering loss to minimize this difference. Although the error between the two distributions is small, the magnitude of the elements values of the two distributions may differ significantly in terms of important features. This is ***scale issue*** [38]. However, due to the scale issue of $\mathbf{Q}$, the single clustering loss term may result in inferior clustering results. To make sure the diversity of two distributions, a discriminator $\mathcal{D}$ is adopted to complement the clustering loss (It will be explained in more details in Sec. 3.3).

The discriminator $\mathcal{D}$ in our proposed GEC-CSD is alternately trained with generator $\mathcal{G}$. The $\mathcal{D}$ tries its best to distinguish the "*real*" samples from target distribution $\mathbf{P}$ and "*fake*" samples from actual data distribution $\mathbf{Q}$. If the clustering performance is satisfactory, the actual data distribution $\mathbf{Q}$ and ideal target distribution $\mathbf{P}$ should be the same. By feeding back such supervision information to $\mathcal{G}$, the $\mathcal{G}$ will update its parameters to produce more powerful node representations $\mathbf{Z}$ and cluster μ_t to help cluster. By iterative learning, the $\mathcal{G}$ will produce outstanding and effective representations $\mathbf{Z}$ for clustering, the actual data distribution and the target distribution will be basically the same under this condition.

To make the node representations $\mathbf{Z}$ well characterize cluster structure, in other words, making the latent representations $\mathbf{Z}$ more discriminative, the $\ell_{1,2}$-norm penalty on $\mathbf{Z}$ is adopted. We can enforce the graph attention encoder to capture the difference of latent space in different cluster by this constraint.

The network structure of $\mathcal{G}$ and $\mathcal{D}$, objective function and the implementation details of our proposed GEC-CSD are elaborated in detail in the following subsections.

3.2. Generator

By characterizing the geometric difference embedding in different cluster spaces, the generator $\mathcal{G}$ of our proposed GEC-CSD learns discriminative node representations that are favorable for node clustering. In particular, it learns to map the raw node attributes and graph structure to a latent representation space, where nodes can be better represented, then produces "*real*" and "*fake*" samples via latent representation and centroid of each cluster. Finally, the clustering results can be obtained via actual distribution $\mathbf{Q}$.

As shown in Fig. 2, the generator $\mathcal{G}$ utilizes a graph convolutional auto-encoder with self-attention mechanism [28, 27] to non-linearly map the node attribute matrix $\mathbf{X}$ and the corresponding graph structure to latent representations $\mathbf{Z}$. Following [28], in $\mathcal{G}$, the output representations of i-th encoder layer of node S are defined as

$$\mathbf{z}_S^{(i)} = \sum_{j \in \mathcal{N}_S} \alpha_{ij}^{(i)} \sigma\left(\mathbf{W}^{(i)} \mathbf{z}_j^{(i-1)}\right), \tag{1}$$

where $\mathcal{N}_S$ represents the neighbourhood of node S (including itself). $\mathbf{z}_S^{(0)}$ is the raw node attribute $\mathbf{x}_S$ of node S. $\alpha_{Sj}^{(i)}$ is set to make the relevance coefficients $r_{Sj}^{(i)}$ of the node neighbours comparable of the S-th, which is defined as

$$\alpha_{Sj}^{(i)} = \frac{\exp\left(r_{Sj}^{(i)}\right)}{\sum\limits_{l \in \mathcal{N}_S} \exp\left(r_{Sl}^{(i)}\right)}, \tag{2}$$

where $r_{Sj}^{(i)}$ indicates the relevance coefficient between neighbor j-th node and S-th node, it can be calculated by

$$r_{Sj}^{(i)} = \phi\left(\left(\mathbf{v}_s^{(i)}\right)^T \sigma\left(\mathbf{h}_S^{(i)}\right) + \left(\mathbf{v}_r^{(i)}\right)^T \sigma\left(\mathbf{h}_j^{(i)}\right)\right), \tag{3}$$

where $\mathbf{h}_\Theta^{(i)} = \mathbf{W}^{(i)} \mathbf{z}_\Theta^{(i-1)}$. $\mathbf{W}^{(i)} \in \mathbb{R}^{d_i \times d_{i-1}}$, $\mathbf{v}_s^{(i)}, \mathbf{v}_r^{(i)} \in \mathbb{R}^{d_i}$ are the trainable parameters of the i-th encoder layer. $\phi(\cdot)$ denotes the *sigmoid* activation function, and $\sigma(\cdot)$ denotes the activation function of i-th encoder layer.

Following [28], the decoder part is constructed with the same number of layers as the encoder. Each decoder layer tries to reverse the process of its corresponding

encoder layer. By utilizing the output of the encoder as the input of the decoder, *i.e.*, the i_{th} decoder layer reconstructs the representation of node S in layer i-1, we have

$$\hat{\mathbf{z}}_S^{(i-1)} = \sum_{j \in N_S} \hat{\alpha}_{Sj}^{(i)} \sigma\left(\hat{\mathbf{W}}^{(i)} \hat{\mathbf{z}}_j^{(i)}\right), \tag{4}$$

where $\hat{\mathbf{z}}_S^{(0)}$ represents the output of the last layer in decoder, which is the reconstructed feature content $\hat{\mathbf{x}}_S$ of node S.

According to Eqs. (1, 4), to make sure the learned node representations $\mathbf{Z}$ can preserve the sufficient information of both node attributes and graph structure, the reconstruction loss is defined as

$$\mathcal{L}_{\mathcal{R}} = \frac{1}{N} \sum_{S=1}^{N} \left[\|\mathbf{x}_S - \hat{\mathbf{x}}_S\|_2^2 - \xi_r \sum_{j \in \mathcal{N}_S} \phi\left(-\mathbf{z}_S^T \mathbf{z}_j\right) \right], \tag{5}$$

where the first term is node attribute reconstruction, and the second term is the corresponding graph structure reconstruction, which is implemented by an inner product decoder [28]. Now, the node representations $\mathbf{Z}$ are obtained.

The proposed method aims to cluster with the node representations $\mathbf{Z}$. In order to characterize the relationship between representation $\mathbf{z}_S$ of node S and cluster centroid μ_t, we herein define a distinctive activation function $\mathbb{A}(\cdot, \cdot)$, which is related to the cluster centroid. Hence, we have

$$\mathbb{A}(\mathbf{z}_S, \mu_t) = \frac{\left(1 + ||\mathbf{z}_S - \mu_t||^2\right)^{-1}}{\sum_{t'} \left(1 + ||\mathbf{z}_S - \mu_{t'}||^2\right)^{-1}}. \tag{6}$$

We define the result obtained by Eq. (6) as the actual data distribution $\mathbf{Q} \in \mathbb{R}^{N \times \mathcal{C}}$. So, we have $q_{St} = \mathbb{A}(\mathbf{z}_S, \mu_t)$, where $q_{St} \in \mathbf{Q}$ represents the probability that clusters node S into t-th cluster. The clustering result of node S can be calculated from the last optimized $\mathbf{Q}$ by $l_S = \max \mathbf{Index}(\mathbf{q}_S)$, where $\max \mathbf{Index}(\cdot)$ is set to find the index of max probability value in S-th row of $\mathbf{Q}$. The centroid μ_t of each cluster is defined as the trainable variable. The centroids calculated by K-Means is utilized to initialize μ_t.

As shown in Fig. 3, suppose we map node representations $\mathbf{Z}$ to a low-dimensional Ψ. According to *t-SNE*, we need to find a reasonable distribution $\mathbf{P}$ of Ψ, if representations $\mathbf{Z}$ are similar to low-dimensional Ψ, the actual distribution $\mathbf{Q}$ should be the

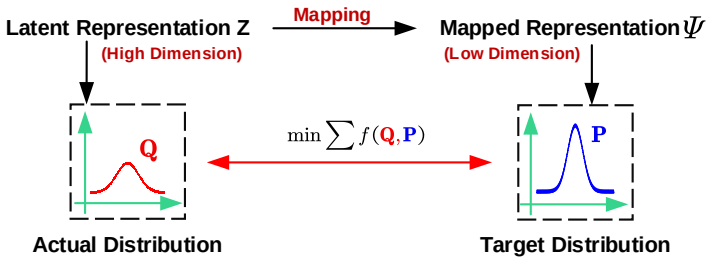


Figure 3: The illustration shows the idea to calculate target distribution $\mathbf{P}$.

same as distribution $\mathbf{P}$. Hence, a clustering loss term $\min \sum f(\mathbf{Q}, \mathbf{P})$ is introduced to describe the difference between the two distributions, which is defined by

$$\mathcal{L}_{\mathcal{C}} = \sum f(\mathbf{Q}, \mathbf{P}) = \xi_{\mathcal{C}} \|\mathbf{Q} - \mathbf{P}\|_F^2, \tag{7}$$

where $\xi_{\mathcal{C}}$ is a coefficient to control the balance between $\mathcal{L}_{\mathcal{C}}$ and $\mathcal{L}_{\mathcal{R}}$. Due to the scale issue of $\mathbf{Q}$, only adopting the clustering loss term may degrade the diversity of distribution $\mathbf{Q}$. Considering that the adversarial learning can well characterize the differences between two different distributions [39], we set data distribution $\mathbf{Q}$ to the "*fake*" samples, and naturally distribution $\mathbf{P}$ is the "*real*" and samples (See Sec.3.3). Thus, as a complement to the clustering loss term, the distribution $\mathbf{Q}$ and $\mathbf{P}$ can match each other via adversarial learning.

It's crucial to calculate a reasonable distribution $\mathbf{P}$. Because actual distribution $q_{Sj} \in \mathbf{Q}$ refers to the probability of clustering node S to cluster j. We hope target distribution $\mathbf{P}$ has the following properties: 1) it can further emphasize more on the nodes assigned with high confidence, 2) it can strengthen predictions, 3) it can prevent large clusters from distorting the latent representations of the nodes. In our proposed GEC-CSD, we consider the data distribution $\mathbf{Q}$ as low-dimensional map Ψ. Hence, motivated by [40], $\mathbf{P}$ is defined as:

$$p_{St} = \frac{q_{St}^2 / \sum_S q_{St}}{\sum_{t'} q_{St'}^2 / \sum_S q_{St'}}, \tag{8}$$

where $\sum_S q_{St}$ is the soft cluster frequency. In fact, the target distribution intends to further enhance the actual distribution, and it concentrates more on the assigned data with high confidence, so the distribution $\mathbf{P}$ is called ideal target distribution.

In terms of node representations discriminability, the existing methods do not consider the CSD in latent node space, resulting in inferior clustering performance. To obtain the intrinsic feature distribution of different clusters, we herein employ $\ell_{1,2}$-norm to measure the CSD. Here, the $\ell_{1,2}$-norm penalty on $\mathbf{Z}$ is adopted as CSD constraint, so that the CSD is fully considered in the latent space. Hence, it can be defined as

$$\mathcal{L}_{\text{CSD}} = \beta \|\mathbf{Z}\|_{1,2}^2 = \beta \sum_{S=1}^{N} \|\mathbf{z}_S\|_1^2, \tag{9}$$

where β is a tradeoff parameter. By minimizing Eq. (9), different elements in squared ℓ_1-norm of S-th row $\mathbf{z}_S$ are competing with each other to survive, and at least one element in row $\mathbf{z}_S$ survives (remaining nonzero). By doing so, some discriminative features are survived for each cluster to provide certain flexibility in the learned nodes representations, *i.e.*, making $\mathbf{Z}$ well exploit the CSD property.

3.3. Discriminator

We build the discriminator to distinguish whether the ideal target distribution $\mathbf{P}$ ("*real*" data) and actual data distribution $\mathbf{Q}$ ("*fake*" data) are consistent. For the discriminator $\mathcal{D}$, it only needs to learn an effective discriminant model to make sure the actual data distribution $\mathbf{Q}$ can get close to the ideal target distribution $\mathbf{P}$. Inspired by t-SNE, the distribution $\mathbf{P}$ and $\mathbf{Q}$ should be the same if the data in two different spaces are similar. Hence, the discriminator $\mathcal{D}$ evaluates current clustering performance by feeding back the differences of target distribution $\mathbf{P}$ and data distribution $\mathbf{Q}$ to generator $\mathcal{G}$. In our proposed GEC-CSD, the $\mathcal{D}$ is trained by minimizing the following discriminative loss $\mathcal{L}_{\mathcal{D}}$:

$$\mathcal{L}_{\mathcal{D}} = \min_{\mathcal{D}} \gamma_1 \sum_{S=1}^{N} [\log(\mathbf{p}_S) + \log(1 - \mathbf{q}_S)], \tag{10}$$

where γ_1 is a trade-off parameter. For all datasets, we employ a three-layer fully connected neural network $d(\mathbf{p}; \vartheta)$ as discriminator $\mathcal{D}$.

In training step, $\mathcal{D}$ is trained to distinguish whether the input distribution is from ideal target distribution $\mathbf{P}$ or actual data distribution $\mathbf{Q}$, while $\mathcal{G}$ is trained to fool the discriminator $\mathcal{D}$ to think that the data distribution $\mathbf{Q}$ is ideal distribution $\mathbf{P}$. By iterating

alternately, finally, these two distributions $\mathbf{Q}$, $\mathbf{P}$ will be the same, which also means the generator $\mathcal{G}$ generating more excellent $\mathbf{Z}$.

3.4. Training and Clustering

Now we can define the total loss of the generator $\mathcal{G}$. Following the idea of adversarial learning, generator $\mathcal{G}$ is trained to fool the discriminator $\mathcal{D}$ to think that the data distribution $\mathbf{Q}$ is ideal distribution $\mathbf{P}$. Therefore, we constrain generator $\mathcal{G}$ to minimize $\mathcal{L}_a = \gamma_2 \sum_{S=1}^{N} [-\log(\mathbf{q}_S)]$, *i.e.*, to encourage the generated data distribution $\mathbf{Q}$ ("*fake*" data) to be close to the ideal target distribution $\mathbf{P}$ ("*real*" data) indicating more satisfactory clustering results through tuning the representation learning and clustering performance from $\mathcal{G}$. Combining this adversarial loss $\mathcal{L}_a$ of generator $\mathcal{G}$ with the reconstruction loss $\mathcal{L}_{\mathcal{R}}$, the clustering loss $\mathcal{L}_{\mathcal{C}}$ and CSD constraint, we have the final training objective of function $\mathcal{G}$:

$$\mathcal{L}_{\mathcal{G}} = \min_{\mathcal{G}} \gamma_2 \mathcal{L}_a + \mathcal{L}_{\mathcal{R}} + \mathcal{L}_{\mathcal{C}} + \mathcal{L}_{\text{CSD}}, \tag{11}$$

where γ_2 is a trade-off parameter.

We train the whole network structure as follows. We initialize the network parameters of generator $\mathcal{G}$ and discriminator $\mathcal{D}$ (See Sec. 4 for the specific initialization methods). We can gain the node representations $\mathbf{Z}$ via inputting the node attribute $\mathbf{X}$ and their corresponding graph structure $\mathbf{A}$ into the generator $\mathcal{G}$. We apply K-Means on the representations $\mathbf{Z}$ to obtain the initial cluster centroid μ_t of each cluster. Then we utilize representations $\mathbf{Z}$ and cluster centroids μ to compute actual data distribution $\mathbf{Q}$, and we further calculate the target distribution $\mathbf{P}$. Next we jointly train the $\mathcal{D}$ and $\mathcal{G}$.

Since the network structure of discriminator $\mathcal{D}$ is shallower than the generator's. In order to train the model more stable, we alternately update $\mathcal{D}$ and $\mathcal{G}$ for 3 times and 1 time within each epoch respectively. Both discriminator $\mathcal{D}$ and generator $\mathcal{G}$ adopt the Adam algorithm [41] to optimize, and the proposed method does not require pre-training step. The learning rate of $\mathcal{D}$ and $\mathcal{G}$ are 3×10^{-3} and 3×10^{-5} respectively.

At the end of the adversarial training, we can predict the clustering label of each node from actual data distribution $\mathbf{Q}$. For node $\mathbf{x}_S$, its clustering label can be calculated by $\mathbf{q}_S$, in which the index with the highest probability value is $\mathbf{x}_S$'s cluster.

Table 1: Descriptions of datasets, where # means the number of.

Dataset	The size of attributes	# Classes	# Nodes	# Links
Cora	$1,433$	7	$2,708$	$5,429$
Citeseer	$3,703$	6	$3,327$	$4,732$
Pubmed	500	3	$19,717$	$44,438$

4. Experiments

We evaluate the node clustering performance of our proposed method in three citation network datasets (Cora, Citeseer and Pubmed) [42] with three frequently-used evaluation metrics: accuracy (ACC), normalized mutual information [43] (NMI) and average rand index (ARI) as in [11], and the higher values imply more correct results. Brief statistics of three datasets are shown in Table 1.

4.1. Comparing methods

We compare the proposed method with several traditional clustering methods and state-of-the-art graph clustering methods. According to the input of different algorithms, the compared methods can be roughly categorized into three groups as described below:

• **Using node attribute only:** K-Means [21], as the baseline clustering method, is compared in our experiments.

• **Using node graph structure only:** Spectral clustering (SC) [44], graph encoder [26], DeepWalk [22], denoising auto-encoder for graph embedding (DNGR) [25] and M-NMF [23].

• **Using both node graph structure and node attribute:** Robust multi-view SC (RMSC) [45], TADW [24], GAE and variational GAE (VGAE) [20], marginalized graph auto-encoder (MGAE) [8], ARGAE and variational ARGAE (ARVGAE) [10], DAEGA [9], GATE [28], GALA [11] and ARVGA-AX [12].

Table 2: The dimensions of the different network layers on Cora, Citeseer and Pubmed datasets.

Dataset	encoder-1/decoder-2	encoder-2/decoder-1
Cora	512	512
Citeseer	$2,000$	500
Pubmed	$2,000$	348

4.2. Experimental Parameters Setting

In our experiments, for all datasets, we fix the parameters of the proposed method as $\gamma_1 = \gamma_2 = 0.5$, $\xi_r = 0.5$, $\xi_{\mathcal{C}} = 10$. As for the parameter $\beta = 1 \times 10^{-5}$ is adopted on Cora and Citeseer dataset, $\beta = 1 \times 10^{-3}$ is set on Pubmed dataset. The reason for setting the ϱ very small is to maintain balance in Eq. (11), we find the loss value of the CSD term in Eq. (11) is much larger than the other three items. The graph encoder of generator $\mathcal{G}$ in our proposed GEC-CSD has two layers, their node representation dimensions are given in Table 2, the decoder has a symmetrical structure to the encoder. The activation functions $\sigma(\cdot)$ of all layers in $\mathcal{G}$ are set to the identity function, empirically resulting in better performance compared to other activation functions [28]. The network dimensions of $\mathcal{D}$ is set as 2000-2000-1. We use ReLU [46] as the non-linear activations of first two layers in $\mathcal{D}$, and the Sigmoid activation is set to the output layer in $\mathcal{D}$. Unlike other methods [29, 9], the proposed method does not require pre-training. For fair comparison, we employ the same dataset provided by [28, 9], and the processing method is also set the same. Our hyper-parameters in numbers are almost the same as the competitor GATE [28] and DAEGC [9]. We utilize TensorFlow $1.13.1$ platform to implement our proposed GEC-CSD.

4.3. Node Clustering Results

We evaluate our proposed GEC-CSD on the Cora, Citeseer, and Pubmed datasets for node clustering task. Table 3 summarizes the clustering performance on three datasets. One can observe that the proposed method consistently shows superior performance to the other baselines for all three metrics. On the Cora and Pubmed datasets,

Table 3: Clustering results of various methods on three datasets. Best results are highlighted in **bold**. *Info.* means the input information of different methods: $\mathbf{A}$ denotes the node graph structure, $\mathbf{X}$ represents the node feature content.

Dataset	*Info.*	Cora			Citeseer			Pubmed		
Metric		ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
K-means [21]	$\mathbf{X}$	0.500	0.547	0.501	0.544	0.312	0.285	0.580	0.278	0.246
SC [44]	$\mathbf{A}$	0.398	0.297	0.174	0.308	0.090	0.082	0.496	0.147	0.098
Graph Encoder [26]	$\mathbf{A}$	0.301	0.059	0.046	0.293	0.057	0.043	0.531	0.210	0.184
Deep Walk [22]	$\mathbf{A}$	0.529	0.384	0.291	0.390	0.131	0.137	0.647	0.238	0.255
DNGR [25]	$\mathbf{A}$	0.419	0.318	0.142	0.326	0.180	0.043	0.468	0.153	0.059
M-NMF [23]	$\mathbf{A}$	0.423	0.256	0.161	0.336	0.099	0.070	0.470	0.084	0.058
RMSC [45]	$\mathbf{A\&X}$	0.466	0.320	0.203	0.516	0.308	0.266	0.629	0.273	0.247
TADW [24]	$\mathbf{A\&X}$	0.536	0.366	0.240	0.529	0.320	0.286	0.565	0.224	0.177
GAE [20]	$\mathbf{A\&X}$	0.530	0.397	0.293	0.380	0.174	0.141	0.632	0.249	0.246
VGAE [20]	$\mathbf{A\&X}$	0.592	0.408	0.347	0.392	0.163	0.101	0.619	0.216	0.201
MGAE [8]	$\mathbf{A\&X}$	0.684	0.511	0.448	0.661	0.412	0.414	0.593	0.282	0.248
ARGE [10]	$\mathbf{A\&X}$	0.640	0.449	0.352	0.573	0350	0.341	0.681	0.276	0.291
ARVGE [10]	$\mathbf{A\&X}$	0.638	0.450	0.374	0.544	0.261	0.245	0.513	0.117	0.078
DAEGC [9]	$\mathbf{A\&X}$	0.704	0.528	0.496	0.672	0.397	0.410	0.671	0.266	0.278
GATE [28]	$\mathbf{A\&X}$	0.658	0.527	0.451	0.616	0.401	0.381	0.673	0.322	0.299
GALA [11]	$\mathbf{A\&X}$	0.746	0.578	0.532	0.693	0.441	0.446	0.684	0.327	0.321
ARVGA-AX [12]	$\mathbf{A\&X}$	0.711	0.526	0.495	0.581	0.338	0.301	0.640	0.239	0.226
GEC-CSD	$\mathbf{A\&X}$	**0.755**	**0.607**	**0.540**	**0.718**	**0.445**	**0.473**	**0.699**	**0.340**	**0.330**

our proposed GEC-CSD respectively improves by 2.9% and 1.3% over the second best GALA on NMI. For both ACC and ARI metrics on Citeseer dataset, the proposed method also brings 2.5% improvement and improves by 2.7% over the state-of-the-arts. Moreover, our proposed GEC-CSD achieves much better clustering performance rather than several shallow graph clustering or feature content clustering methods, *e.g.,* RMSC, K-Means and spectral clustering. This is because the proposed method utilizes a multi-layer graph convolutional attention auto-encoder to learn node representations. So our proposed GEC-CSD can learn deeper, more powerful latent features. Furthermore, the algorithms with self-attention mechanism, *e.g.,* GATE and DAEGC, out-

Table 4: Comparison of the execution times (in seconds) of different GCNs based methods on three datasets.

Dataset	Cora	Citeseer	Pubmed
ARGE [10]	27.727	**47.077**	1933.370
ARVGE [10]	173.730	145.216	16942.421
DAEGC [9]	29.340	69.533	207.526
ARVGA-AX [12]	188.526	159.033	16531.420
GEC-CSD	**27.433**	74.689	**199.041**

perform the methods without attention mechanism, *e.g.,* ARGE and GAE, since self-attention mechanism determines the significance between nodes and their neighbours. Especially, our proposed GEC-CSD outperforms GATE [28] and ARVGA-AX [12], which adopts the same graph auto-encoder architecture as our generator $\mathcal{G}$. The better performance of our proposed GEC-CSD owes to the more discriminate node representations, which indicates the adversarial learning and $\ell_{1,2}$-norm penalty in the proposed method are effective at benefiting the node representation learning and clustering.

Comparison of the execution times. We also compare our proposed GEC-CSD with several representative GCNs based methods. Table 4 shows the comparison results of the execution times (in seconds). As can be seen, the time consumed by our proposed GEC-CSD is significantly less than that of other algorithms when dealing large scale dataset. The same amount of time is consumed when dealing with small scale datasets. These results well demonstrate the effectiveness of our proposed GEC-CSD from another aspect.

Table 5: Ablation studies on three datasets.

$\ell_{1,2}$-norm	Adversarial	Cora			Citeseer			Pubmed		
		ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
✗	✗	0.721	0.565	0.500	0.678	0.415	0.428	0.677	0.313	0.306
✗	✓	0.742	0.589	0.523	0.697	0.442	0.451	0.691	0.333	0.320
✓	✗	0.730	0.574	0.511	0.704	0.425	0.450	0.688	0.325	0.321
✓	✓	**0.755**	**0.607**	**0.540**	**0.718**	**0.445**	**0.473**	**0.699**	**0.340**	**0.330**

4.4. Ablation studies

To better illustrate the performance of our proposed GEC-CSD, we validate the effectiveness of the $\ell_{1,2}$-norm penalty and adversarial learning by clustering task on the above three datasets. We verify the results of discarding or reserving two terms, corresponding four configurations as shown in Table 5. For fairness, we utilize the same hyper-parameters corresponding to the three datasets reported in Sec. 4.2.

• **Case1: We discard both the $\ell_{1,2}$-norm penalty and adversarial learning**. It means the objective function of the network only contains reconstruction loss $\mathcal{L}_{\mathcal{R}}$ in Eq. (5) and clustering loss $\mathcal{L}_{\mathcal{C}}$ in Eq. (7). Compared with the baseline GATE [28] in Table 3, it can be clearly noticed that the clustering loss in Eq. (5) are helpful to find the latent representations which can improve clustering performance.

• **Case2: We discard either the $\ell_{1,2}$-norm penalty or adversarial learning**. It indicates the objective function in Eqs. (10, 11) of $\mathcal{D}$ and $\mathcal{G}$ will remove adversarial loss or cluster-specific constraint term. Compared with **Case1** mentioned above, it can be clearly noticed that both $\ell_{1,2}$-norm penalty and adversarial learning are helpful to learn better latent node discriminative representations for node clustering task. In addition, one can observe that the method, on some evaluation metrics, outperforms state-of-the-arts in Table 3. This is because compared with some state-of-the-arts, *e.g.,* MGAE, ARGE, DAEGC and GALA, we directly impose CSD constraint on latent features, which can characterize the geometric distribution characteristics of data in different cluster spaces. Meanwhile, when we reserve adversarial learning between generator and discriminator, this can guild generator to correct current clustering error so that the generator can generate more powerful representations.

• **Case3: We reserve both the $\ell_{1,2}$-norm penalty and adversarial learning**. This can clearly demonstrate the clustering effectiveness of the proposed method. This is because both the $\ell_{1,2}$ penalty and adversarial learning between generator and discriminator in our proposed GEC-CSD are effective at benefiting the representation learning and clustering.

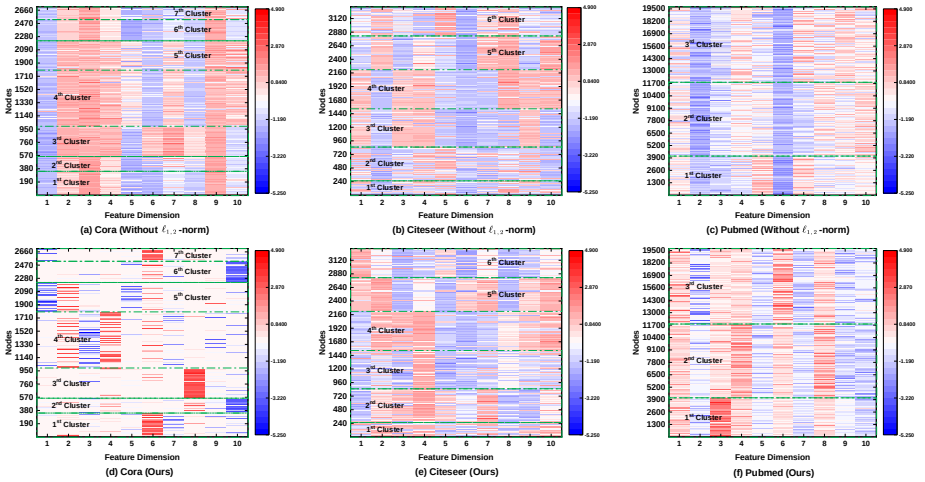


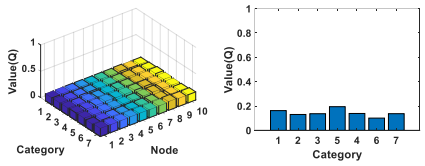
Figure 4: Visualizations of the representations $\mathbf{Z}$ on three datasets. In order to easily observe the discriminability of $\mathbf{Z}$, we arrange the nodes according to the correct cluster, each row is the latent representation of a node.

4.5. Effect of CSD constraint

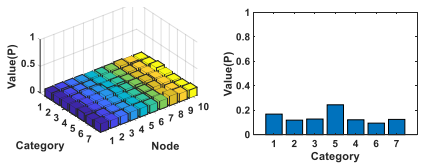
One of the key ideas of the proposed GEC-CSD is the cluster-specificity distribution, which is measured by $\ell_{1,2}$-norm penalty. To verify its effectiveness on a visual level, we visualize the learned latent representations on three datasets in Fig. 4. For better explanation, the dimension of a node's latent representation is set to 10, (a-c) mean the learned representations without $\ell_{1,2}$-norm penalty, (d-f) are just the opposite. Each row is a node's latent feature, we put together samples of the same cluster. From the visualizations (d-f), it is clearly observed that the nodes of different clusters are distributed in different dimensions of the feature dimension, and the nodes of same clusters have a common distribution in the intrinsic feature dimension, especially on the Cora and Pubmed datasets. This well demonstrates that the $\ell_{1,2}$-norm help characterize the cluster-specificity distribution of data in different cluster spaces in unsupervised learning. Although the representations in Fig. 4 (b) also look very discriminative without the $\ell_{1,2}$-norm penalty, they also confirm the effectiveness of adversarial learning from the other side.

Table 6: The effectiveness of discriminator.

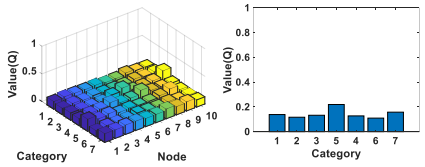
Method	Cora			Citeseer			Pubmed		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
GEC-CSD with KLD	0.738	0.586	0.520	0.695	0.421	0.438	0.683	0.306	0.300
GEC-CSD	**0.755**	**0.607**	**0.540**	**0.718**	**0.445**	**0.473**	**0.699**	**0.340**	**0.330**



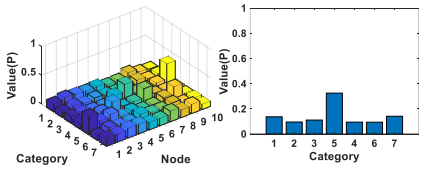
(a) Distribution $\mathbf{Q}$ in Epoch 1



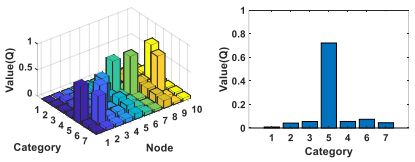
(b) Distribution $\mathbf{P}$ in Epoch 1



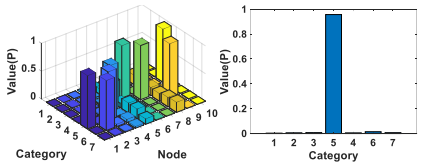
(c) Distribution $\mathbf{Q}$ in Epoch 20



(d) Distribution $\mathbf{P}$ in Epoch 20



(e) Distribution $\mathbf{Q}$ in Epoch 200



(f) Distribution $\mathbf{P}$ in Epoch 200

Figure 5: Visualizations of distributions $\mathbf{Q}$ and $\mathbf{P}$ on **Cora** dataset. We randomly selected 10 nodes to show the distribution relationship vs. training epoch, where (a, c, e) are the visualizations of distribution $\mathbf{Q}$, (b, d, f) are the visualizations of distribution $\mathbf{P}$.

4.6. Effect of adversarial learning

To demonstrate the effectiveness of adversarial learning in distribution measurement, we verify the node clustering performance of GEC-CSD under Kullback-Leibler divergence (KLD) measure. Specifically, we first get rid of the discriminator, then the original clustering objective function $\mathcal{L}_c$ is replaced by Kullback-Leibler divergence

(KLD), thus we have

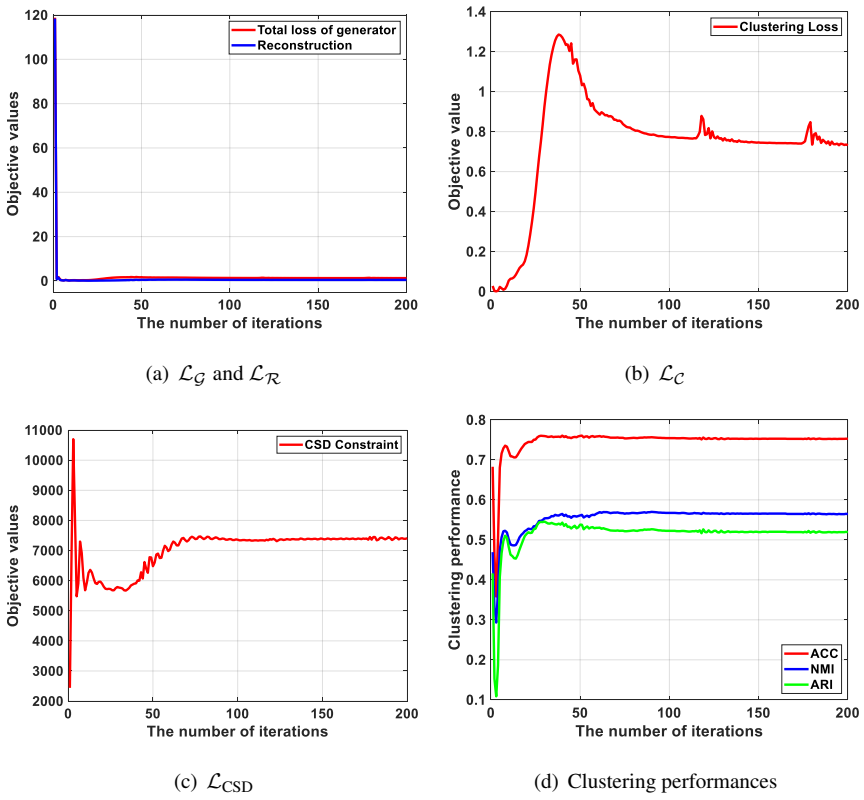
$$\mathcal{L}_{\mathcal{C}} = \mathrm{KLD}(\mathbf{P} \,\|\mathbf{Q}) = \sum_{S=1}^{N} \sum_{j=1}^{\mathcal{C}} p_{Sj} \log \left(\frac{p_{Sj}}{q_{Sj}} \right), \tag{12}$$

where N and $\mathcal{C}$ are the number of nodes and clusters, respectively. Table 6 shows the clustering results of considering KLD measure. As can be seen, our proposed GEC-CSD is still have advantages. The reason maybe that the introduced discriminator help tackle scale issue, thereby further eliminating gaps between two distributions. Meanwhile, the distributions visualizations also demonstrate the effectiveness of this strategy.

Distributions Visualization. As reported in Fig. 5, we visualize the target distribution and data distribution of a random 10 nodes of Cora dataset vs. training epoch, where the left color *3-D* bar shows the 10 nodes data distribution $\mathbf{Q}$ and target distribution $\mathbf{P}$. The right *2-D* bar clearly shows the 1-*st* node's distribution $\mathbf{Q}$ and $\mathbf{P}$. With training the $\mathcal{G}$ and $\mathcal{D}$ alternately, one can be observed is that the distribution difference of $\mathbf{Q}$ and $\mathbf{P}$ becomes smaller and smaller. As aforementioned, we hope the data distribution can get close to target distribution, which means the clustering performance is excellent. Comparing with each training epoch, we find our proposed GEC-CSD performs well. Hence, the $\mathbf{P}, \mathbf{Q}$ adversarial learning in Eqs. (10, 11) is advantageous in the process of learning latent node representations.

4.7. Convergence Behaviors

To verify the convergence of the proposed GEC-CSD, as shown in Fig. 6, we record the objective values and clustering performances of GEC-CSD vs. iterations. As observed, although the objective values do monotonically decrease at each iteration, the overall convergence can be reached within approximately 50 steps of iterations. Moreover, we observe that clustering results gradually increase to a maximum and generally maintains it up to slight variation. These results demonstrate that our proposed GEC-CSD can quickly converge.



(a) $\mathcal{L}_{\mathcal{G}}$ and $\mathcal{L}_{\mathcal{R}}$ (b) $\mathcal{L}_{\mathcal{C}}$

(c) $\mathcal{L}_{\text{CSD}}$ (d) Clustering performances

Figure 6: The objective values and clustering performances of GEC-CSD vs. iterations on Cora dataset.

5. Conclusion

We proposed a novel graph embedding clustering model based on graph attention auto-encoder, which joints node representations learning and clustering into a unified framework. Different from previous works, we simultaneously take node attributes reconstruction and graph structure reconstruction into account to boost the capability of representations learning. Meanwhile, the $\ell_{1,2}$-norm penalty on node representations is introduced to enforce the learned representations more cluster-specific and consequently improve clustering performance. Moreover, a reasonable adversarial learning is adopted to complement the diversity of node representations distributions. Experimental results on the citation datasets demonstrate the validity of our proposed GEC-CSD, and our proposed GEC-CSD performs superior advantages over state-of-

the-arts. Our proposed method can also work with other types of datasets, such as image datasets *e.g.*, MNIST, COIL20 and YALE, *etc*. However, when dealing with such image datasets, it needs to construct corresponding graph structure. Choosing a proper graph construction approach is a challenging task, we will continue to study this in the future.

Acknowledgements

The authors would like to thank the anonymous reviewers and AE for their constructive comments and suggestions. This work is supported by the National Natural Science Foundation of China under Grants 61773302, Natural Science Basic Research Plan in Shaanxi Province under Grant 2020JZ-19 and 2020JQ-327, the Initiative Postdocs Supporting Program BX20190262, the Fundamental Research Funds for the Central Universities and the Innovation Fund of Xidian University.

References

[1] J. Yang, D. Parikh, D. Batra, Joint unsupervised learning of deep representations and image clusters, in: Proc. IEEE CVPR, 2016, pp. 5147–5156.

[2] P. Ji, T. Zhang, H. Li, M. Salzmann, I. D. Reid, Deep subspace clustering networks, in: Proc. NeurIPS, 2017, pp. 24–33.

[3] J. Chang, L. Wang, G. Meng, S. Xiang, C. Pan, Deep adaptive image clustering, in: Proc. IEEE ICCV, 2017, pp. 5880–5888.

[4] P. Zhou, Y. Hou, J. Feng, Deep adversarial subspace clustering, in: Proc. IEEE CVPR, 2018, pp. 1596–1604.

[5] J. Wen, Y. Xu, H. Liu, Incomplete multiview spectral clustering with adaptive graph learning, IEEE Trans. Cybern. 50 (4) (2020) 1418–1429.

[6] Q. Gao, W. Xia, Z. Wan, D. Xie, P. Zhang, Tensor-svd based graph learning for multi-view subspace clustering, in: AAAI, 2020, pp. 3930–3937.

[7] W. Xia, X. Zhang, Q. Gao, X. Shu, J. Han, X. Gao, Multi-view subspace clustering by an enhanced tensor nuclear norm, IEEE Trans. Cybern. doi: 10.1109/TCYB.2021.3052352.

[8] C. Wang, S. Pan, G. Long, X. Zhu, J. Jiang, Mgae: Marginalized graph autoencoder for graph clustering, in: Proc. ACM CIKM, 2017, pp. 889–898.

[9] C. Wang, S. Pan, R. Hu, G. Long, J. Jiang, C. Zhang, Attributed graph clustering: A deep attentional embedding approach, in: Proc. IJCAI, 2019, pp. 3670–3676.

[10] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, C. Zhang, Adversarially regularized graph autoencoder for graph embedding, in: Proc. IJCAI, 2018, pp. 2609–2615.

[11] J. Park, M. Lee, H. J. Chang, K. Lee, J. Y. Choi, Symmetric graph convolutional autoencoder for unsupervised graph representation learning, in: Proc. IEEE ICCV, 2019, pp. 6518–6527.

[12] S. Pan, R. Hu, S. Fung, G. Long, J. Jiang, C. Zhang, Learning graph embedding with adversarial training methods, IEEE Trans. Cybern. 50 (6) (2020) 2475–2487.

[13] Y. Xie, Y. Zhang, M. Gong, Z. Tang, C. Han, MGAT: multi-view graph attention networks, Neural Networks 132 (2020) 180–189.

[14] G. Ou, G. Yu, C. Domeniconi, X. Lu, X. Zhang, Multi-label zero-shot learning with graph convolutional networks, Neural Networks 132 (2020) 333–341.

[15] X. Zhou, F. Shen, L. Liu, W. Liu, L. Nie, Y. Yang, H. T. Shen, Graph convolutional network hashing, IEEE Trans. Cybern. 50 (4) (2020) 1460–1472.

[16] M. T. Kejani, F. Dornaika, H. Talebi, Graph convolution networks with manifold regularization for semi-supervised learning, Neural Networks 127 (2020) 160–167.

[17] X. Shen, F. Chung, Deep network embedding for graph representation learning in signed networks, IEEE Trans. Cybern. 50 (4) (2020) 1556–1568.

[18] G. Nikolentzos, G. Dasoulas, M. Vazirgiannis, k-hop graph neural networks, Neural Networks 130 (2020) 195–205.

[19] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: Proc. ICLR, 2017.

[20] T. N. Kipf, M. Welling, Variational graph auto-encoders, in: Proc. NeurIPS Workshop on Bayesian Deep Learning, 2016.

[21] J. MacQueen, et al., Some methods for classification and analysis of multivariate observations, in: Proc. Mathematical Statistics and Probability, 1967, pp. 281–297.

[22] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: online learning of social representations, in: Proc. ACM SIGKDD, 2014, pp. 701–710.

[23] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, S. Yang, Community preserving network embedding, in: Proce. AAAI, 2017, pp. 203–209.

[24] C. Yang, Z. Liu, D. Zhao, M. Sun, E. Y. Chang, Network representation learning with rich text information, in: Proc. IJCAI, 2015, pp. 2111–2117.

[25] S. Cao, W. Lu, Q. Xu, Deep neural networks for learning graph representations, in: Proc. AAAI, 2016, pp. 1145–1152.

[26] F. Tian, B. Gao, Q. Cui, E. Chen, T. Liu, Learning deep representations for graph clustering, in: Proc. AAAI, 2014, pp. 1293–1299.

[27] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: Proc. ICLR, 2018.

[28] A. Salehi, H. Davulcu, Graph attention auto-encoders, CoRR abs/1905.10715.

[29] Z. Tao, H. Liu, J. Li, Z. Wang, Y. Fu, Adversarial graph embedding for ensemble clustering, in: Proc. IJCAI, 2019, pp. 3562–3568.

[30] S. Kou, W. Xia, X. Zhang, Q. Gao, X. Gao, Self-supervised graph convolutional clustering by preserving latent distribution, Neurocomputing 437 (2021) 218–226.

[31] P. Zhao, G. Rocha, B. Yu, The composite absolute penalties family for grouped and hierarchical variable selection, Ann. Statist. 37 (6A) (2009) 3468–3497.

[32] Y. Zhou, R. Jin, S. C. Hoi, Exclusive lasso for multi-task feature selection, in: Proc. AISTATS, 2010, pp. 988–995.

[33] F. Campbell, G. I. Allen, Within group variable selection through the exclusive lasso, Electron. J. Statist. 11 (2) (2017) 4220–4257.

[34] D. Ming, C. Ding, Robust flexible feature selection via exclusive L21 regularization, in: Proc. IJCAI, 2019, pp. 3158–3164.

[35] S. Deng, W. Xia, Q. Gao, X. Gao, Cross-view classification by joint adversarial learning and class-specificity distribution, Pattern Recognit. 110. doi:10.1016/j.patcog.2020.107633.

[36] W. Xia, X. Zhang, Q. Gao, X. Gao, Adversarial self-supervised clustering with cluster-specificity distribution, Neurocomputing 449 (2021) 38–47. doi:https://doi.org/10.1016/j.neucom.2021.03.108.

[37] L. van der Maaten, G. Hinton, Visualizing data using t-sne, Journal of Machine Learning Research 9 (86) (2008) 2579–2605.

[38] W. Xia, Q. Gao, Q. Wang, X. Gao, Regression-based clustering network via combining prior information, Neurocomputing 448 (2021) 324–332.

[39] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proc. NeurIPS, 2014, pp. 2672–2680.

[40] J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, in: Proc. ICML, 2016, pp. 478–487.

[41] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proc. ICLR, 2015.

[42] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, T. Eliassi-Rad, Collective classification in network data, AI Magazine 29 (3) (2008) 93–106.

[43] Y. Luo, T. TIAN, J. Shi, J. Zhu, B. Zhang, Semi-crowdsourced clustering with deep generative models, in: Proc. NeurIPS, 2018, pp. 3212–3222.

[44] A. Y. Ng, M. I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: Proc. NeurIPS, 2001, pp. 849–856.

[45] R. Xia, Y. Pan, L. Du, J. Yin, Robust multi-view spectral clustering via low-rank and sparse decomposition, in: Proc. AAAI, 2014, pp. 2149–2155.

[46] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proc. NeurIPS, 2012, pp. 1106–1114.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:
