

# Unveiling community structures in static networks through graph variational Bayes with evolution information

Junwei Cheng<sup>a</sup>, Chaobo He<sup>a</sup>, Kunlin Han<sup>b</sup>, Gangbin Chen<sup>a</sup>, Wanying Liang<sup>a</sup>, Yong Tang<sup>a,\*</sup>

<sup>a</sup> School of Computer Science, South China Normal University, Guangzhou, China

<sup>b</sup> Computer Science Department, University of Southern California, Los Angeles, USA

## ARTICLE INFO

Communicated by Z. Tao

### Keywords:

Variational graph autoencoder  
Community detection  
Evolution information  
Mixture of Gaussian distributions  
K-core decomposition

## ABSTRACT

With the rapid development of graph variational Bayes theory, some representative community detection methods have been proposed. Although these methods are well designed, they are limited by the inherent constraints of static networks. Therefore, how to re-examine the information in static networks from a new perspective becomes a challenge. To this end, in this paper, we propose EG-VGAE and its variant EGC-VGAE, going beyond the constraints of static networks from a new perspective. Specifically, we first demonstrate that the evolution relations on static networks can be simulated reasonably. And then, our EG-VGAE method combines evolution information with static network information to realize a fine-grained propagation of local low-frequency signals to global low-frequency signals, thereby improving the accuracy of community assignment for each node. Building on this progress, our method EGC-VGAE imposes the smoothness constraint on adjacent slices, significantly enhancing the sensitivity of the method to evolution information and mitigating the impact of network noise. The comprehensive experimental results on real static networks well validate that our methods outperform state-of-the-art methods in most cases. The code is available at <https://github.com/GDM-SCNU/EG-VGAE>.

## 1. Introduction

Static networks, as the basic carrier of graph data mining tasks, have received extensive attention for a long time, because what we are familiar with, such as social relations [1], citation relations [2], etc., can all be modeled as static networks. As one of the most important tasks of graph data mining, community detection (also known as graph clustering) is of great significance for downstream applications, such as anomaly detection [3], recommendation [4], etc. Therefore, how to support the downstream applications better, that is, to find the optimal community structure with dense intra-cluster and sparse inter-cluster characteristics in static networks [5], has become an urgent problem to be solved.

In recent years, with the rapid development of shallow [6] and deep methods [7], many works have been proposed to detect communities in static networks, including Non-negative Matrix Factorization-based methods [8,9], Autoencoder-based methods [10,11] and Convolutional Networks-based methods [12–14], etc. Typically, these methods project static network information into a low-dimensional space and explore the community assignment of each node. Specifically, Graph Convolution Network (GCN) [15] extends the convolution operation to the

graph domain, serving as a prominent method within Graph Neural Networks in the spectral domain, and have achieved remarkable achievements in representation learning. Building on this progress, Variational Graph Auto-encoder (VGAE) [16] is able to encode node representations by exploiting the ability of GCN to capture global graph information and low-frequency node signals. As a deep auto-encoder method, VGAE exhibits superior scalability, generalization and generative capabilities. Thereby, it is spurring the development of numerous community detection methods [17,18], which are built upon it, in recent years.

Despite the clever design and effectiveness of these methods, their capabilities are restricted by the inherent constraints of static networks. Specifically, most methods are only capable of leveraging a limited amount of static network information, such as topology and attribute, or their higher-order counterparts, to infer the community assignment for nodes. Thereby, this constraint hinders the ability to gain new insights into such networks beyond their static network information. Given the limitations of current methods, therefore, it is natural to ask whether there exists other hidden valuable information in static networks, that can be used to gain new insights into these networks.

\* Corresponding author.

E-mail addresses: [jung@m.scnu.edu.cn](mailto:jung@m.scnu.edu.cn) (J. Cheng), [ytang@m.scnu.edu.cn](mailto:ytang@m.scnu.edu.cn) (Y. Tang).

<https://doi.org/10.1016/j.neucom.2024.127349>

Received 1 August 2023; Received in revised form 1 November 2023; Accepted 28 January 2024

Available online 30 January 2024

0925-2312/© 2024 Elsevier B.V. All rights reserved.

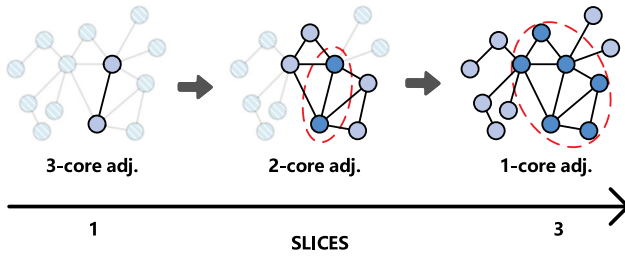


Fig. 1. A toy static network with evolution relations. Light-colored transparent nodes indicate that they do not exist in the current slice.

As we mentioned before, static networks are widely used to represent the relations that exist in the real world, and there are two kinds of relations, i.e., birth and growth [19]. For example, a company (i.e., a community) is started by a small number of core members, and as long as those members remain stable, it will grow steadily over time [20]. Obviously, the evolution relations are important information, because we are able to infer the community structure associated with them when we identify some stable core members. To this end, our goal is to verify the feasibility of the following aspects: (1) It is possible to simulate the evolution relations on static networks, which implies that static networks hold valuable information concerning community evolution. And further, (2) the fine-grained low-frequency signal is propagated from the local network to the global network with the assistance of evolution relations, which helps to improve the accuracy of node community assignment.

We investigate a lot of static networks to conduct empirical observation (refer to Section 2), and conclude that simulated the evolution relations held by static networks is possible if we consider the edge nodes in the distribution data as core members. After validating the first goal, in this paper, we propose two novel methods based on the VGAE framework, namely EG-VGAE and its variant EGC-VGAE. Comparing with other designs, our vision is no longer limited to the information on the surface of the static networks. Instead, we direct our attention to evolution information on static networks that few researchers have focused on. Specifically, we partition the static network into different slices using the  $k$ -core decomposition algorithm [21] to simulate the evolution relations. As the serial number of slices increases, the degree distribution of nodes approximately follows a gaussian distribution according to Limov-Laplace central limit theorem [22]. Nodes, falling on both sides of the quantile point, are considered as core members. This formation process of the distribution precisely represents the evolution process on static networks. For example, we divide a static network into 3 slices shown in Fig. 1, where we regard the nodes in slice-1 network (left) as core members. With the development of the network, new nodes will establish relations with core members, and sometimes expand outward from themselves, i.e., nodes outside the red circle in slice-2 (middle) and slice-3 (right) network. For our proposed method, EG-VGAE, we assume that the latent representation of nodes follows a mixture of gaussian distributions. In each gaussian distribution, EG-VGAE retains and learns the low-frequency distribution signal of nodes associated with the current slice. We make the assumption that the simulated evolution process follows a continuous pattern, which enables the low-frequency distribution signal from the current slice to be captured by the corresponding node in the next slice. In the next slice, the captured signal is updated based on newly established relations. At the same time, the new low-frequency distribution signals are treated in the same manner. This evolution process from local to global helps to learn node representations to approximate the true distribution, while it is cooperating with static network information to alleviate the impact of graph noise on performance of method. Compared with EG-VGAE, EGC-VGAE additionally emphasizes the smoothness constraints between adjacent slices, making it more sensitive to the evolution information in each slice. The contributions of this paper are:

- By combining the results of empirical experiments with the evolution theory of communities, we find that it is possible to simulate evolution relations on static networks, if the formation process of the node degree distribution is regarded as an evolution process.
- We propose two novel methods, EG-VGAE and its variant EGC-VGAE, inspired by the evolution on static networks. For EG-VGAE, it propagates the low-frequency distribution signal of nodes from local to global, which helps to learn node representations to approximate the true distribution. Furthermore, this method can cooperate with the static network information to reduce the impact of noise on our method. For our variant method EGC-VGAE, it imposes a smoothness constraint on adjacent slices, which significantly enhances the sensitivity of the method to evolution information and mitigates the impact of network noise.
- The extensive experiments on real-world static networks show that our methods, i.e., EG-VGAE and EGC-VGAE, and at least one of them achieves SOTA or on par performance with existing methods.

## 2. Empirical observation

We conduct empirical researches to get the general rule of node degree distribution. We first observe real static networks such as Citeseer, Cora, Pubmed and Blogcatalog, and then present our empirical results.

**Degree distributions.** Evolution theory of community shows that the evolution of communities often exhibits a characteristic life-cycle, which comprises two distinct phases, namely birth and growth. Generally, the birth of a community is usually initiated by a small stable core members. And the process that new nodes (relations) increase the size of the community as long as these members remain relatively stable in the community is called the growth of community. The motivation for exploring the evolution relations is to utilize evolution information in conjunction with static network information to infer communities. However, static networks, as snapshots at specific moments, are static and complex, making it difficult to identify the evolution relations of communities.

**Observation.** To address the challenges posted by the static feature of networks in identifying evolution relations, we design a systematic observation process, as shown in Fig. 2. Specifically, we conduct a study on the degree distribution of Citeseer, Cora, Pubmed and Blogcatalog, respectively. Assuming that the distribution satisfies the i.i.d. assumption, and the probability of establishing a relation between nodes follows a binomial distribution with parameters  $n$  and  $p$ . According to the Limov-Laplace central limit theorem, as the number of nodes gradually increases, the binomial distribution approximates a gaussian distribution, as follows:

$$\lim_{n \rightarrow \infty} P\left\{ \frac{X_n - np}{\sqrt{np(1-p)}} \leq x \right\} = \Phi(x) \quad (1)$$

where  $X_n$  and  $x$  are a random variable and any real number, respectively.

Therefore, our analysis reveals that the distribution of degrees in these networks can be approximated by a gaussian distribution [23]. For instance, we use Citeseer as an example and observe that a large proportion of nodes has a degree within the range of 0 to 15, which is indicated by the scatter plot in the black box. And the points of these nodes are concentrated around the red line. Beyond this range, the degree distribution is relatively sparse. This phenomenon is abstractly characterized as the relation between edge and center of data distribution, i.e., evolution relations of edge distribution towards center distribution. That is what we mention before, as long as core members (at the edge of the distribution) remain stable, new nodes (concentrated in the center of the distribution) increase the size of community. Thereby, the above observations lead us to consider whether the evolution relations can be inferred from degree distributions.

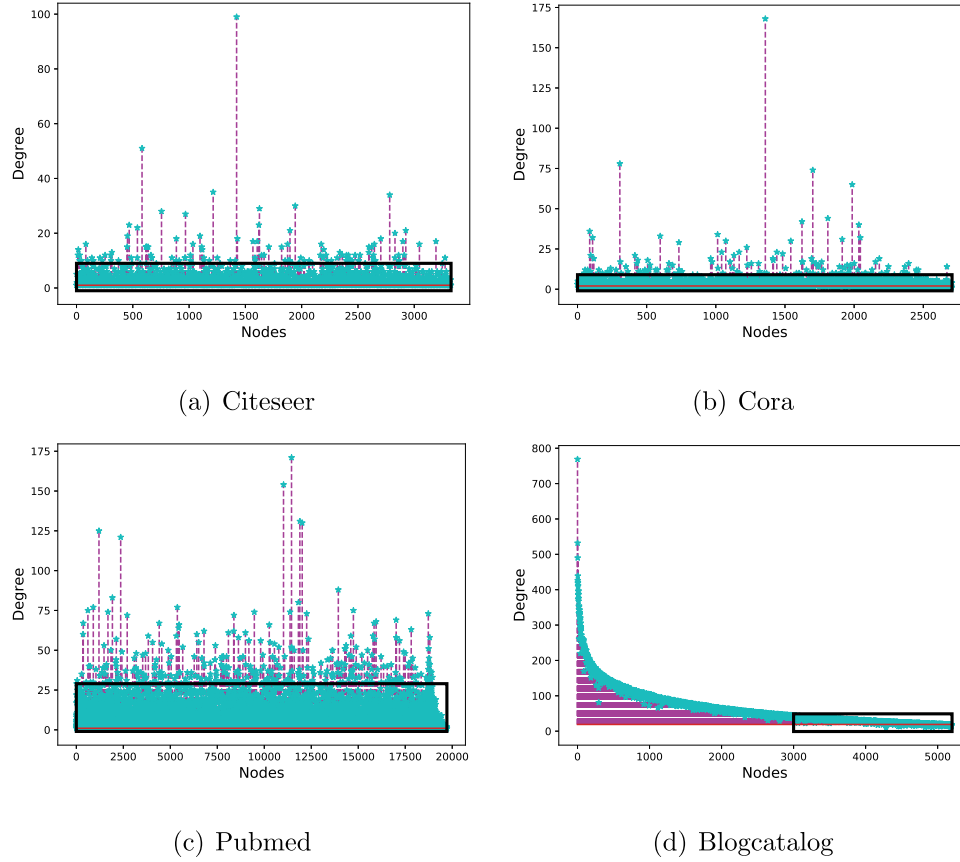


Fig. 2. Distribution of degrees for Citeseer, Cora, Pubmed and Blogcatalog.

**Discussion.** A naive idea is to decompose different sub-graphs through different degree distributions. However, this idea presents two challenges. Firstly, if the sub-graph is decomposed based on the degree categories, it may lose its cluster characteristics. Secondly, the more degree categories there are, the more sub-graphs there will be, which can seriously affect the generalization ability of the method.

To avoid these issues and gain insight into the evolution relations on static networks, a simple yet effective approach is to utilize the  $k$ -core decomposition algorithm to sort out the associations among different degree distributions. As explained in literatures [24,25], the  $k$ -core decomposition algorithm provides a core theory that enables hierarchical division of the network. Specifically, as  $k$  increases, nodes with higher coreness (referred to as core nodes) are retained in the sub-graph, and unimportant nodes are filtered. Theoretically, the sub-graph corresponding to the maximum  $k$  contains the most important nodes in the network. Consequently, the community, which these nodes belong to, is considered as the initial community, signifying the birth of the community, i.e., the first appearance of a new community composed of any number of core members in networks [26]. Conversely, as  $k$  decreases, more nodes emerge in the sub-graph, all of which meet the constraints of  $k$ -core decomposition. Subsequently, more node relations are established, leading to an expansion in the size of each community. This is considered as community growth, wherein new nodes contribute to the enlargement of community [26].

Building on the core theory, we form a case study with Citeseer, which partitions graph into sub-graphs. The sub-graph with the highest  $k$  is slice-1 while the sub-graph, slice-2, is with the second highest  $k$ , and so on. We apply the VGAE method to perform community detection on each sub-graph. The experimental results are shown in Fig. 3.

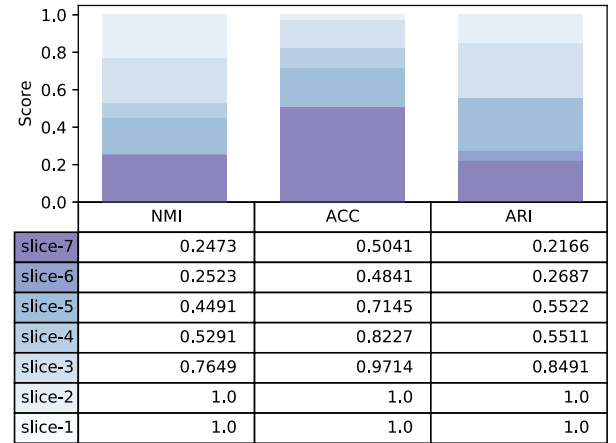


Fig. 3. Empirical experiment by using the VGAE method to analyze evolution relations on Citeseer.

Notably, the decreasing trend of metrics can be explained by the complexity of the network, which is caused by the evolution of community. Specifically, when the serial numbers of the slices are small, VGAE can effectively detect communities of core nodes, since these nodes have significant cluster characteristics. However, as the serial number of slices increases, new nodes (relations) emerge, which further complicates the sub-graphs. Consequently, the performance of the method gradually decreases, as shown in the results from slice-5 to slice-7.

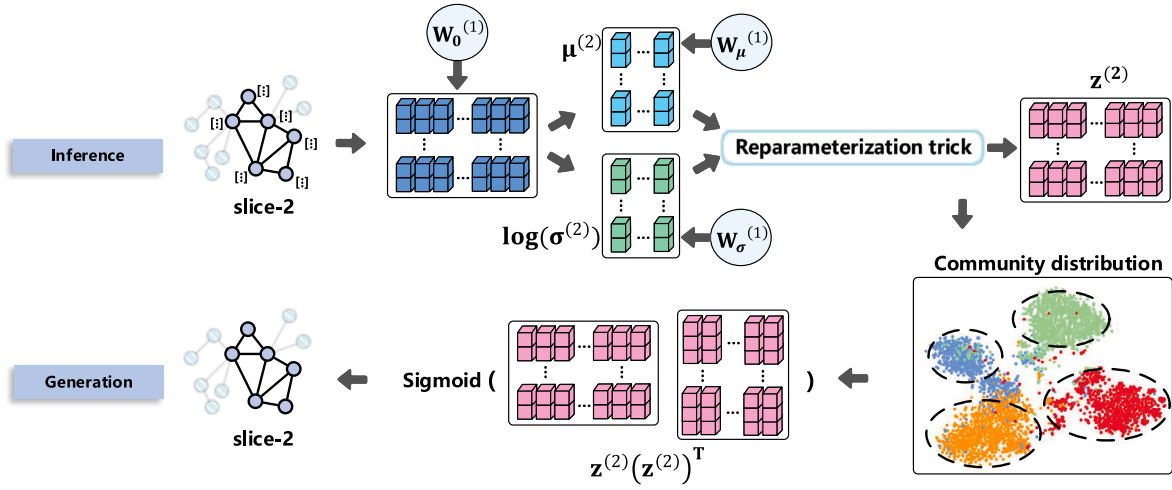


Fig. 4. The framework of EG-VGAE and EGC-VGAE.

To this end, how to improve the accuracy of node community assignment with the aid of evolution relations is the focus of this paper.

### 3. Methodology

In this section, two new methods called EG-VGAE and its variant EGC-VGAE are proposed. These methods are designed to infer communities in an end-to-end manner by incorporating both evolution information and static network information. To start with, some concepts and notations that will be used later are defined. Subsequently, we introduce our proposed methods and derive new objective functions within the variational Bayes framework. For EG-VGAE, we present the Evolution ELBO, while for EGC-VGAE, we propose the Conditional Evolution ELBO.

#### 3.1. Preliminaries

**Static Networks.** A static network can be defined as a non-directed graph  $\mathcal{G} = \{\mathbf{A}, \mathbf{X}\}$ , where  $\mathbf{A} \in \mathbb{R}^{N \times N}$  denotes the adjacency matrix with  $N$  nodes, and if there is a relation between  $u$  and  $v$ ,  $\mathbf{A}_{u,v} = 1$ , otherwise, 0.  $\mathbf{X} \in \mathbb{R}^{N \times M}$  indicates the attribute matrix with  $M$  attributes.

**Network Partition.** To identify the evolution relations on static networks, we given the definition of  $k$ -core [27]:

**Definition 1.** Consider a graph  $\mathcal{G}$  and a sub-graph  $\mathcal{G}'$ .  $\mathcal{G}'$  is a  $k$ -core of  $\mathcal{G}$  if  $\mathcal{G}'$  is a maximal sub-graph of  $\mathcal{G}$  in which all nodes have a degree of at least  $k$ . They satisfy the following nested chain:

$$\underbrace{\mathcal{G}^{(k_{\max})}}_{\text{slice-1}} \subseteq \underbrace{\mathcal{G}^{(k_{\max}-1)}}_{\text{slice-2}} \subseteq \dots \subseteq \mathcal{G}^{(i)} \subseteq \dots \subseteq \mathcal{G}^{(1)} = \mathcal{G} \quad (2)$$

where  $k_{\max}$  is the max  $k$ -core number of  $\mathcal{G}$ . For slice- $i$ , all elements different from other slices are marked with superscript  $i$ , such as  $\mathbf{A}^{(i)}$ .

**Variational Bayes Framework.** The variational Bayes framework is summarized in three components: inference, generation and evidence lower bound (ELBO). For a sample  $\omega$ , methods based on this framework hope to learn its latent representation  $z$  via probability density function of its true distribution. However, this is a challenging problem as we cannot directly determine the true distribution  $p(z | x)$ . Therefore, classical methods such as VaDE [28] and VAE [29] aim to approximate the unknown true distribution by introducing a known distribution  $q(z | x)$  and deriving and optimizing the objective function of the likelihood, as follows:

$$\begin{aligned} \log p(\omega) &= \mathbb{E}_{z \sim q(z|x)} \log p(x) \\ &= \int_z q(z | x) \log p(x | z) dz - KL(q(z | x) \| p(z)) \\ &\quad + KL(q(z | x) \| p(z | x)) \end{aligned} \quad (3)$$

among them, since Kullback–Leibler Divergence ( $KL$ ) [30]  $> 0$ , so the optimization of the above objective function is adjusted to optimize the ELBO of the maximum likelihood, as follows:

$$\log p(\omega) > \mathbb{E}_{z \sim q(z|x)} \log p(x) = \int_z q(z | x) \log p(x | z) dz - KL(q(z | x) \| p(z)) \quad (4)$$

**Community Detection.** Given  $N$  nodes, community detection methods aim to divide these nodes into  $K$  disjoint communities  $\{c_1, c_2, \dots, c_K\}$  under unsupervised learning in an end-to-end manner.

#### 3.2. Inference of EG-VGAE with evolution relations

Inspired by the connection between  $k$ -core theory and community evolution theory, we partition the static network into different slices via  $k$ -core decomposition algorithm. The upper side of Fig. 4 shows the inference component of our EG-VGAE method. Consider a sample  $\omega$  randomly sampled from sample space  $\Omega^{(i)}$ , which consists of all nodes in slice- $i$ . The inference of EG-VGAE is defined as follows based on mean field distribution assumption:

$$q(z^{(i)}, c | a^{(i)}, x) = q(z^{(i)} | a^{(i)}, x) q(c | a^{(i)}, x) \quad (5)$$

where  $q(\cdot)$  is the variational prior we introduce to approximate the unknown posterior  $p(\cdot)$ .  $z^{(i)}$ ,  $a^{(i)}$  and  $x$  are denotes vectors corresponding to the matrix associated with the sample  $\omega$ , such as  $a^{(i)} \in \mathbf{A}^{(i)}$ .  $c$  is the community assignment of the sample  $\omega$ , where  $c$  can take on any of the values  $\{c_1, c_2, \dots, c_K\}$  to describe a non-specific community.

The first term is probabilization by a two-layer GCN, defined as:

$$q(z^{(i)} | a^{(i)}, x) = \mathcal{N}(\mu^{(i)}, \text{diag}[(\sigma^{(i)})^2]) \quad (6)$$

where  $\mu^{(i)}$  and  $\sigma^{(i)}$  represent the mean and standard deviation vector in slice- $i$  respectively, defined as:

$$\mu^{(i)} = \tilde{a}^{(i)} \text{ReLU}(\tilde{a}^{(i)} x w_0) w_{\mu,1} \quad (7)$$

$$\log \sigma^{(i)} = \tilde{a}^{(i)} \text{ReLU}(\tilde{a}^{(i)} x w_0) w_{\sigma,1} \quad (8)$$

where  $w_{\mu,1}$ ,  $w_{\sigma,1}$  and  $w_0$  are trainable weight vectors, and are shared across all slices.  $\tilde{a}^{(i)} = (d^{(i)})^{-\frac{1}{2}} a^{(i)} (d^{(i)})^{-\frac{1}{2}}$  is the symmetric normalized adjacency vector in slice- $i$  and  $d^{(i)}$  is the degree corresponding to  $a^{(i)}$ .

Notably, the trainable weight vectors in GCN are shared across all slices. This special detail incorporates evolution information into the learning process of node representations. Based on the mathematical theory of Fourier operator and Laplace transform, the propagation and aggregation process of GCN can be regarded as preserving the low-frequency signals of nodes while filtering the high-frequency signals

of nodes. Within a specific slice, the trainable weight matrices, constrained by the current adjacency matrix  $\mathbf{A}^{(i)}$ , is only associated with nodes in the current slice. In simpler terms, the transfer and aggregation of low-frequency information among local nodes is not affected by global graph information. Consequently, without the interference of complex global graph information, these weight matrices can simultaneously contain accurate local community evolution information and static network information, and be passed as additional signal to the learning of the next slice. In addition, in order to maintain the continuity and rationality of the simulated evolution process, the training of our method is carried out according to the sequence number of slices from small to large.

Finally, the latent representation  $z^{(i)}$ , to allow gradient backpropagation, is computed by the reparameterization trick:

$$z^{(i)} = \mu^{(i)} + \sigma^{(i)} \times \theta^{(i)} \quad (9)$$

where  $\theta^{(i)} \sim \mathcal{N}(0, 1)$  is a gaussian noise vector.

For another term  $p(c | a^{(i)}, x)$ , which involves the probability of community assignment, its probability density function is determined by the mixture of gaussian distributions, because a single gaussian distribution cannot distinguish which community a node should belong to. Consequently, we extend and regroup the Eq. (4) by introducing static network information, as follows:

$$\begin{aligned} & \mathbb{E}_{(z^{(i)}, c) \sim q(z^{(i)}, c | x, a^{(i)})} \log \frac{p(z^{(i)}, c, a^{(i)})}{q(z^{(i)}, c | x, a^{(i)})} \\ &= \int_{z^{(i)}} q(z^{(i)} | x, a^{(i)}) \log \frac{p(a^{(i)} | z^{(i)}) p(z^{(i)})}{q(z^{(i)} | x, a^{(i)})} dz \\ & - \int_{z^{(i)}} q(z^{(i)} | x, a^{(i)}) KL(q(c | x, a^{(i)} || p(c | z^{(i)}))) dz \end{aligned} \quad (10)$$

where our goal is to optimize the ELBO of maximum likelihood only and if only the  $KL(\cdot || \cdot) \equiv 0$ , i.e.,  $q(c | x, a^{(i)}) = p(c | z^{(i)})$ . From that, we follow [28] to define density function  $p(c | z^{(i)})$  as:

$$p(c | z^{(i)}) = \frac{p(z^{(i)} | c) p(c)}{\sum_{k=1}^K p(z^{(i)} | c_k) p(c_k)} \quad (11)$$

where  $p(z^{(i)} | c) \sim \mathcal{N}(z^{(i)} | \mu_c, \text{diag}[\sigma_c^2])$ , and  $p(c) \sim \text{Cat}(\pi)$  is categorical distribution with prior probability  $\pi$ , where  $\pi$  meets  $\sum_{k=1}^K \pi_k = 1$ .

### 3.3. Generation of EG-VGAE

The generation of EG-VGAE is shown on the bottom side of Fig. 4. For each pair of samples  $\omega$  and  $\varphi \in \Omega^{(i)}$ , their connection state are generated by following the Bernoulli distribution. In our method, we follow the commonly used inner product decoder in the variational Bayes framework. More precisely, the generation of EG-VGAE is defined as follows:

$$p(\tilde{\mathbf{A}}_{\omega, \varphi}^{(i)} | \mathbf{Z}_{\omega}^{(i)}, \mathbf{Z}_{\varphi}^{(i)}) = \text{Sigmoid}(\mathbf{Z}_{\omega}^{(i)} [\mathbf{Z}_{\varphi}^{(i)}]^T) \quad (12)$$

### 3.4. Evolution evidence lower bound for EG-VGAE

Our method is to optimize the variational objective such that the likelihood function is maximized. Specifically, given sample  $\omega$  sampled from sample space  $\Omega^{(i)}$ , the optimization objective is to maximize the ELBO, which is defined as follows:

$$\log p(\omega) \geq \mathcal{L}_{ELBO}(\omega) = \mathbb{E}_{(z^{(i)}, c) \sim q(z^{(i)}, c | x, a^{(i)})} \log \frac{p(z^{(i)}, c, a^{(i)})}{q(z^{(i)}, c | x, a^{(i)})} \quad (13)$$

Since  $a^{(i)}$  and  $c$  are conditionally independent given  $z^{(i)}$ , the joint probability  $p(z^{(i)}, c, a^{(i)})$  is decomposed as follows:

$$p(z^{(i)}, c, a^{(i)}) = p(a^{(i)} | z^{(i)}) p(z^{(i)} | c) p(c) \quad (14)$$

By combining the previously mentioned mean field distribution assumption with Eq. (14), we apply the identity transformation to

Eq. (13), as follows:

$$\begin{aligned} \mathcal{L}_{ELBO} &= \mathbb{E}_{\sim q(\cdot)} \log p(a^{(i)} | z^{(i)}) + \mathbb{E}_{\sim q(\cdot)} \log p(z^{(i)} | c) \\ &+ \mathbb{E}_{\sim q(\cdot)} \log p(c) - \mathbb{E}_{\sim q(\cdot)} \log q(z^{(i)} | x, a^{(i)}) - \mathbb{E}_{\sim q(\cdot)} \log q(c | x, a^{(i)}) \end{aligned} \quad (15)$$

where  $\mathbb{E}_{(z^{(i)}, c) \sim q(z^{(i)}, c | x, a^{(i)})}$  is simplified to  $\mathbb{E}_{\sim q(\cdot)}$  for convenience.

For brevity, we treat all variables as continuous random variables, and expand Eq. (15) using double integrals, as follows:

$$\begin{aligned} \mathcal{L}_{ELBO} &= \underbrace{\int_{z^{(i)}} \int_c \phi(z^{(i)}, c) \log p(a^{(i)} | z^{(i)}) dz dc}_{\textcircled{1}} \\ &+ \underbrace{\int_{z^{(i)}} \int_c \phi(z^{(i)}, c) \log p(z^{(i)} | c) dz dc}_{\textcircled{2}} \\ &+ \underbrace{\int_{z^{(i)}} \int_c \phi(z^{(i)}, c) \log p(c) dz dc}_{\textcircled{3}} \\ &- \underbrace{\int_{z^{(i)}} \int_c \phi(z^{(i)}, c) \log q(z^{(i)} | x, a^{(i)}) dz dc}_{\textcircled{4}} \\ &- \underbrace{\int_{z^{(i)}} \int_c \phi(z^{(i)}, c) \log q(c | x, a^{(i)}) dz dc}_{\textcircled{5}} \end{aligned} \quad (16)$$

where  $\phi(z^{(i)}, c) = q(z^{(i)} | x, a^{(i)}) q(c | x, a^{(i)})$  for convenience.

Given that the distribution  $p(z^{(i)} | c)$  in the Eq. (16)-② is known, it can be further expanded as:

$$\begin{aligned} & \int_{z^{(i)}} q(z^{(i)} | x, a^{(i)}) dz \int_c q(c | x, a^{(i)}) [\log \frac{1}{\sqrt{2\pi}\sigma_c} - \frac{(Z^{(i)} - \mu_c)^2}{2\sigma_c^2}] dc \\ &= \sum_{c=1}^K \gamma_c^{(i)} \log \frac{1}{\sqrt{2\pi}\sigma_c} - \frac{1}{2} \sum_{c=1}^K \gamma_c^{(i)} [\frac{[\sigma^{(i)}]^2}{\sigma_c^2} + \frac{(\mu^{(i)} - \mu_c)^2}{\sigma_c^2}] \end{aligned} \quad (17)$$

where  $\gamma_c^{(i)} = q(c | x, a^{(i)}) = p(c | z^{(i)})$ .

Similar to Eq. (17), we expand the Eq. (16)-③, Eq. (16)-④ and Eq. (16)-⑤ in sequence, as follows:

$$\int_{z^{(i)}} q(z^{(i)} | x, a^{(i)}) dz \int_c q(c | x, a^{(i)}) \log \pi_c dc = \sum_{c=1}^K \gamma_c^{(i)} \log \pi_c \quad (18)$$

$$\int_{z^{(i)}} q(z^{(i)} | x, a^{(i)}) \log \frac{1}{\sqrt{2\pi}\sigma^{(i)}} e^{-\frac{(z^{(i)} - \mu^{(i)})^2}{2\sigma^{(i)2}}} dz \int_c q(c | x, a^{(i)}) dc = \log \frac{1}{\sqrt{2\pi}\sigma^{(i)}} - \frac{1}{2} \quad (19)$$

$$\int_{z^{(i)}} q(z^{(i)} | x, a^{(i)}) dz \int_c q(c | x, a^{(i)}) \log q(c | x, a^{(i)}) dc = \sum_{c=1}^K \gamma_c^{(i)} \log \gamma_c^{(i)} \quad (20)$$

Substituting the terms from Eq. (17) to Eq. (20), the  $\mathcal{L}_{ELBO}$  can be rewritten as:

$$\begin{aligned} \mathcal{L}_{ELBO} &= \mathcal{L}_{rec} + \sum_{c=1}^K \gamma_c^{(i)} \log \frac{\pi_c}{\gamma_c^{(i)}} + \frac{1}{2} (\log [\sigma^{(i)}]^2 + 1) \\ &- \frac{1}{2} \sum_{c=1}^K \gamma_c^{(i)} (\log \sigma_c^2 + \frac{[\sigma^{(i)}]^2}{\sigma_c^2} + \frac{(\mu^{(i)} - \mu_c)^2}{\sigma_c^2}) \end{aligned} \quad (21)$$

where Eq. (16)-① is denoted as  $\mathcal{L}_{rec}$ , which represents the reconstruction loss of slice- $i$  and can be expanded by the Bernoulli distribution as:

$$\mathcal{L}_{rec} = \frac{1}{N} \sum_{j=1}^N a_j^{(i)} \log(\tilde{a}_j^{(i)}) + (1 - a_j^{(i)}) \log(1 - \tilde{a}_j^{(i)}) \quad (22)$$



where  $\tilde{a}_j^{(i)}$  is an element in probability distribution of Eq. (12).

### 3.5. Inference of EGC-VGAE with smoothness constraint

Our proposed method, EG-VGAE, effectively integrates evolution information and static network information to infer the community assignment of nodes. However, we find that the community assignment of nodes in the current slice should be influenced by its previous states. To this end, we further consider the smoothness of evolution relations between adjacent slices, and propose a variant of the EG-VGAE method, which we call EGC-VGAE. This method utilizes the same generation component as EG-VGAE, but with a distinct inference component.

Given the community assignment of samples in the previous slice- $(i-1)$ , we construct pairwise constraints to impose a smoothness constraint on adjacent slices, as follows:

$$\mathbf{H}_{\omega,\varphi}^{(i-1)} \begin{cases} > 0, \text{ if there is a pairwise relationship} \\ & \text{between } \omega \text{ and } \varphi \text{ under probability } \rho^{(i-1)} \\ = 0, \text{ otherwise} \end{cases} \quad (23)$$

where smoothness constraint is not applied to slice-1, as it corresponds to the birth of community, and the probability  $\rho^{(i-1)}$  is determined based on the Normalized Mutual Information calculated within slice- $(i-1)$ .

Given the pairwise constraint matrix  $\mathbf{H}^{(i-1)} \in \mathbb{R}^{N \times N}$ , the community assignment is no longer independent and identically distributed, that is,  $p(c | \mathbf{H}^{(i-1)}) \neq p(c)$ . Here, we define the smoothness constraint probability  $p(c | \mathbf{H}^{(i-1)})$  as follows:

$$p(c | \mathbf{H}^{(i-1)}) = \eta_c^{(i-1)} := \frac{\pi_c \prod_{\omega, \varphi \in \Omega} e^{\mathbf{H}_{\omega, \varphi}^{(i-1)} \delta_{\omega, \varphi}(c)}}{\sum_{k=1}^K \pi_k \prod_{\omega, \varphi \in \Omega} e^{\mathbf{H}_{\omega, \varphi}^{(i-1)} \delta_{\omega, \varphi}(k)}} \quad (24)$$

where  $c(\cdot)$  represents the community assignment of the corresponding sample.  $\delta_{\omega, \varphi}(c)$  is Kronecker function about sample  $\omega$  and  $\varphi$ . And if these two sample belong to the same community,  $\delta = 1$ , otherwise, 0. We can observe that an increase in  $\mathbf{H}_{\omega, \varphi} \rightarrow +\infty$  is considered to elevate the probability of a node being assigned to the corresponding community, vice versa.

### 3.6. Conditional evolution evidence lower bound for EGC-VGAE

As the smoothness constraint is imposed on our EGC-VGAE method, we modify the calculation objective of ELBO as follows:

$$\log p(\omega) \geq \mathcal{L}_{ELBO}(\omega) = \mathbb{E}_{(z^{(i)}, c) \sim q(z^{(i)}, c | x, a^{(i)})} \log \frac{p(z^{(i)}, c, a^{(i)} | \mathbf{H}^{(i-1)})}{q(z^{(i)}, c | x, a^{(i)})} \quad (25)$$

where the equation holds when  $i > 1$ , otherwise the ELBO is described by Eq. (13).

It is worth noting that, except for Eq. (16)-③, the rest of the double integral equations are the same, so we give the modified equation, as follows:

$$\begin{aligned} & \int_{z^{(i)}} \int_c q(z^{(i)} | x, a^{(i)}) q(c | x, a^{(i)}) \log p(c | \mathbf{H}^{(i-1)}) dz dc \\ &= \int_{z^{(i)}} q(z^{(i)} | x, a^{(i)}) dz \int_c q(c | x, a^{(i)}) \eta_c^{(i-1)} dc = \sum_{c=1}^K \gamma_c^{(i)} \log \eta_c^{(i-1)} \end{aligned} \quad (26)$$

By replacing the mathematical description about Eq. (18) in Eq. (21) with Eq. (26), we obtain a modified optimization objective that imposes a smoothness constraint into our method EGC-VGAE, as follows:

$$\begin{aligned} \mathcal{L}_{ELBO} &= \mathcal{L}_{rec} + \sum_{c=1}^K \gamma_c^{(i)} \log \frac{\eta_c^{(i-1)}}{\gamma_c^{(i)}} + \frac{1}{2} (\log [\sigma^{(i)}]^2 + 1) \\ &\quad - \frac{1}{2} \sum_{c=1}^K \gamma_c^{(i)} (\log \sigma_c^2 + \frac{[\sigma^{(i)}]^2}{\sigma_c^2} + \frac{(\mu^{(i)} - \mu_c)^2}{\sigma_c^2}) \end{aligned} \quad (27)$$

### Algorithm 1: EG-VGAE

---

**Input:**  
A non-directed graph  $\mathcal{G}$  and iteration number  $L$ ;

**Output:**  
embedding vector  $z^{(k_{max})}$  and community assignment  $c$  of sample  $\omega$ ;

- 1 Divide the original network  $\mathcal{G}$  into different slices through the  $k$ -core decomposition algorithm;
- 2 Training the parameters involving the gaussian mixture model via Eq. (22);
- 3 **for each iteration**  $l \in L$  **or method has not converged do**
- 4   **for start from slice-1 until end do**
- 5     **for**  $n = 1, \dots, N$  **do**
- 6       **► Inference**
- 7       Calculate gaussian parameters  $\mu^{(i)}$  and  $\sigma^{(i)}$  via Eq. (7) and Eq. (8);
- 8       Calculate embedding vector  $z^{(i)}$  via Eq. (9);
- 9       **► Community sampling**
- 10       Sampling a community  $c \sim \text{Cat}(\pi)$ ;
- 11       Calculate community assignment  $c$  via Eq. (11);
- 12       **► Generation**
- 13       Reconstruct  $\tilde{a}^{(i)}$  via Eq. (12);
- 14       **► ELBO**
- 15       Calculate loss  $\mathcal{L}_{ELBO}$  via Eq. (21) and backpropagate gradients;
- 16 **return**  $z^{(k_{max})}$  and  $c$ .

---

### Algorithm 2: EGC-VGAE

---

**► Construct pairwise constraint**  
Construct  $\mathbf{H}^{(i-1)}$  under probability  $\rho^{(i-1)}$  via Eq. (23);

**► Community sampling**  
Calculate community assignment  $c$  under smoothness constraint via Eq. (24);

**► ELBO**  
Calculate loss  $\mathcal{L}_{ELBO}$  via Eq. (27) and backpropagate gradients.

---

**Table 1**

The statistics of the datasets.

Datasets	# Node	# Attributes	# Edges	K
Cora	2,708	1,433	5,278	7
Citeseer	3,327	3,703	4,552	6
Pubmed	19,717	500	44,324	3
Blogcatalog	5,196	8,189	171,743	6

### 3.7. Time complexity

The complete procedures of EG-VGAE is outlined in Algorithm 1, while the difference between EGC-VGAE and EG-VGAE is outlined in Algorithm 2. The computational costs of EG-VGAE and EGC-VGAE are equal, mainly divided into four parts: inference model, community sampling, generation model, and ELBO. The computational cost for inference model is of order  $O(\sum_{i=1}^{k_{max}} 2\|\mathbf{A}^{(i)}\|M + 2NM^2)$ , where  $\|\mathbf{A}^{(i)}\|$  is number of nonzeros in the adjacency  $\mathbf{A}^{(i)}$ . Similarly, the computational cost for the remaining three parts are of order  $O(Nd)$ ,  $O(N^2d)$  and  $O(N^2)$ , respectively. In all, the computational cost of EG-VGAE is of order  $O(\sum_{i=1}^{k_{max}} [2\|\mathbf{A}^{(i)}\|M + 2NM^2 + N^2d])$ . In fact, many matrices in experiments exhibit sparsity, which implies that the actual computational cost may be lower than the theoretical estimate.

**Table 2**

Community detection result. (Bold: best; “\_”: runner-up).

Datasets	Cora			Citeseer			Pubmed			Blogcatalog		
Metrics	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
VGAE	0.4107	0.5347	0.3147	0.2410	0.4575	0.1948	0.2344	0.6654	0.2644	0.2278	0.3430	0.1625
GMM-VGAE	0.4892	0.6854	0.4877	0.4087	0.6728	0.4233	0.2929	0.7073	0.3209	0.2135	0.3480	0.1086
VGAER	0.4470	0.4664	0.3088	0.2105	0.3730	0.1657	0.2078	0.3590	0.1713	0.2191	0.2171	0.0818
R-GMM-VGAE	<b>0.5550</b>	<b>0.7467</b>	<b>0.5400</b>	0.4057	0.6755	0.4243	0.3004	0.7145	0.3319	0.2333	0.3400	<u>0.1398</u>
ARVGE	0.4688	0.5953	0.3809	0.2737	0.5269	0.2231	0.0817	0.4874	0.0498	0.1985	0.3549	0.1103
ARGE	0.4655	0.6651	0.4081	0.3620	0.6366	0.3676	0.2473	0.6388	0.2340	0.2499	<u>0.3723</u>	<b>0.1498</b>
VGAECD	0.4445	0.5770	0.3349	0.2539	0.5547	0.2935	0.1824	0.5456	0.1365	0.1784	0.3639	0.1295
GM-VGAE	0.5087	0.6766	0.5016	0.4095	0.6338	0.4167	0.2948	0.6937	0.3203	0.1767	0.2769	0.1227
EG-VGAE	<b>0.5618</b>	<u>0.7404</u>	<u>0.5306</u>	<b>0.4213</b>	<b>0.6877</b>	<b>0.4414</b>	<u>0.3018</u>	<u>0.7175</u>	<u>0.3351</u>	<u>0.2585</u>	0.3697	0.1194
EGC-VGAE	0.5422	<u>0.7308</u>	0.5078	<u>0.4206</u>	<u>0.6868</u>	<u>0.4403</u>	<b>0.3202</b>	<b>0.7335</b>	<b>0.3612</b>	<b>0.2646</b>	<b>0.3764</b>	0.1349

## 4. Experiments

In this section, we conduct experiments to answer the following questions: How significant evolution information is in the community detection of static networks? Furthermore, is combining evolution information with static network information to detect community a good idea?

### 4.1. Experiment setup

(1) **Datasets.** We use the real-world networks, i.e., Cora, Citeseer, Pubmed and Blogcatalog,<sup>1</sup> as the benchmark datasets to conduct our experiments. Table 1 shows the details of these datasets.

(2) **Baselines.** To answer the question posted at the beginning, we compare EG-VGAE and EGC-VGAE with many representative methods based on variational framework. They are: VGAE, GMM-VGAE [31], VGAER, R-GMM-VGAE [32], ARVGE & ARGE [33], VGAECD [34] and GM-VGAE [35].

(3) **Parameter settings and Metrics.** For EG-VGAE and EGC-VGAE, we use the Adam optimizer and set the learning rate to 0.002. For all datasets, the dimension of hidden layers in GCN are set to 32 and 16, respectively. For the EGC-VGAE method, we set  $\mathbf{H}_{w,q} = 1$  if these two samples have a pairwise relationship. For all benchmark datasets, our two methods are trained with 200 iterations, and we report the best result. To train the parameters involving the gaussian mixture model, we adopt  $\mathcal{L}_{rec}$  as loss function and train it. Other baseline methods adopt the settings described in their papers. We employ three widely used metrics to measure performance of all methods: Normalized Mutual Information (NMI), Accuracy (ACC) and Adjusted Rand Index (ARI).

### 4.2. Community detection performance

The community detection results are reported in Table 2. We have the following observations and analysis:

(1) Compared with all baselines, our proposed method EG-VGAE and EGC-VGAE respectively achieves the best performance on most benchmark datasets, where EG-VGAE outperforms other baseline methods on all metrics for Citeseer, while EGC-VGAE achieves the best performance on all metrics for Pubmed. Notably, for NMI, ACC and ARI, EG-VGAE achieves maximum relative improvements of 22.01%, 35.85% and 28.53% on Pubmed respectively, and certain boosts of 1.18%, 1.22% and 1.17% on Citeseer over contemporaneous methods. The variant method EGC-VGAE has also achieved significant performance improvements. Analogously, for NMI, ACC and ARI, EGC-VGAE also achieves maximum relative improvements of 23.85%, 37.45% and 31.14% on Pubmed respectively. The above experimental data shows that evolution information needs to be paid attention to, and the combination of evolution information and static network information noticeably enhances the accuracy of community detection.

(2) In order to explore the impact of number of slices obtained by  $k$ -core decomposition on the performance of our methods, we depict the performance change trends of EG-VGAE (dashed line) and EGC-VGAE (solid line) on all slices for each dataset. This visualization is shown in Fig. 5, and we have the following observations: On the Cora and Pubmed datasets, the ACC polyline exhibits relatively little variation, whereas NMI and ARI display more pronounced fluctuations in some adjacent slices, especially in the case of Pubmed. In contrast, the ACC polyline exhibits a degree of irregularity on Citeseer and Blogcatalog. We analyze the possible reasons from the perspective of community evolution as follows: As the slice serial number increases, the complexity of the network also rises significantly. Consequently, the performance of most methods tends to exhibit a slow or cliff-like decline. However, the introduction of evolution information seems to be the highlight of slowing down performance degradation or slightly improving performance. Specifically, for Cora and Pubmed, the increase in the total number of slices makes a smaller variation range in node degrees and a relatively slower network evolution. In this scenario, there are no significant differences in the network structure, ensuring the continued reliability of previously obtained evolutionary information. Building on this foundation, EGC-VGAE can also acquire reliable pairwise constraints to further enhance the accuracy of community detection. The above still holds in Citeseer and Blogcatalog.

(3) We focus on the solid line segments in these four sub-graphs and observe that the performance trend of EGC-VGAE is similar to that of EG-VGAE. However, we only consider the effect of network noise on the performance of EGC-VGAE, and find that it affects the training of parameters  $w_{\mu,1}$ ,  $w_{\sigma,1}$  and  $w_0$  and leads to a bias in the distribution  $q(z^{(i)} | a^{(i)}, x)$ . As we assumed earlier, the probability  $p(c | a^{(i)}, x)$  follows the mixture of gaussian distributions and can be derived from the density function  $p(c | z^{(i)})$ . Consequently,  $p(c | a^{(i)}, x)$  is also a biased distribution. Furthermore, it can be easily inferred that  $\mathbf{H}$  is also biased because it is affected by the inaccurate community distribution. As a result, under the cumulative biases of  $p(c | z^{(i)})$ ,  $q(z^{(i)} | a^{(i)}, x)$  and  $\mathbf{H}$ , EGC-VGAE does not perform as well as contemporary methods on Cora. The primary negative impact is particularly evident during the transition from slice-1 to slice-2, as shown in Fig. 5(a).

(4) We compare EG-VGAE with the variant EGC-VGAE and find that the former performs better on Cora and Citeseer. Specifically, on the Cora, EG-VGAE demonstrates relative improvements of 1.96% in NMI, 0.96% in F1 and 2.28% in ARI. On the contrary, EGC-VGAE performs better on Pubmed and Blogcatalog. From the perspective of GCN training, we analyze the possible reasons for this experimental result as follows: As mentioned before, in networks of similar sizes, having fewer total slices obtained through  $k$ -core decomposition indicates more significant changes in the topology structure of the sub-graph between adjacent slices. Consequently, when GCN requires training with shared parameters, significant changes in topology structure can easily lead to the partial failure of previously trained parameters. Simultaneously, the confidence in pairwise constraints between nodes may decrease. As a result, EG-VGAE performs slightly better than EGC-VGAE on Cora

<sup>1</sup> <https://github.com/GDM-SCNU/Datasets>

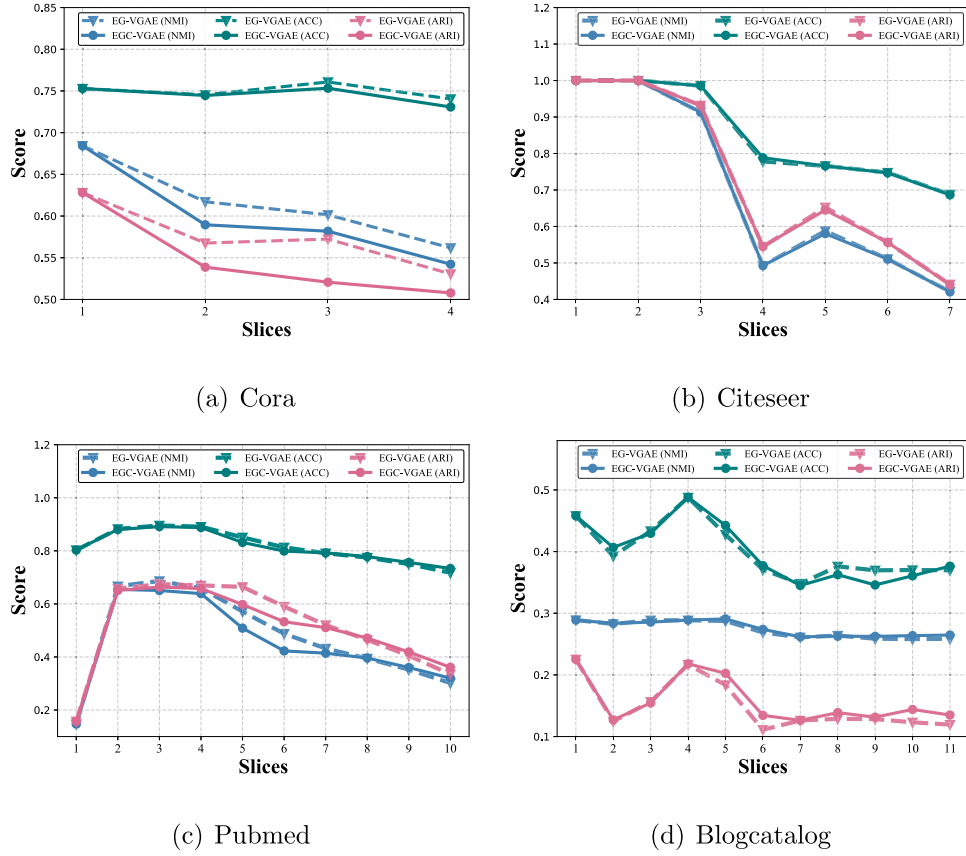


Fig. 5. On all datasets, the performance of EG-VGAE and EGC-VGAE in each slice.

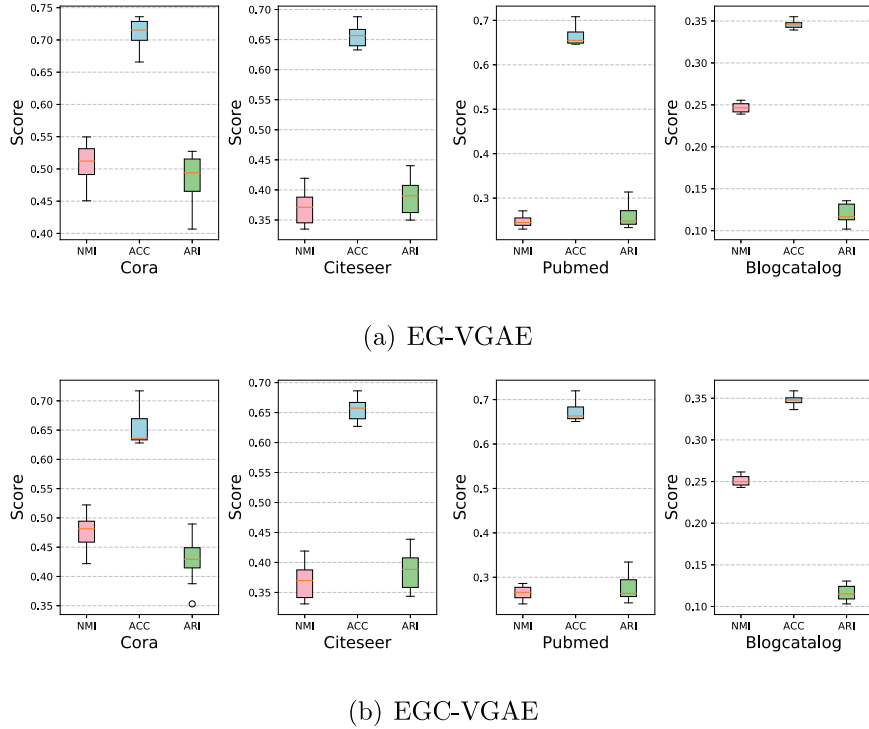


Fig. 6. Robustness experiments of EG-VGAE and EGC-VGAE on Cora, Citeseer, Pubmed and Blogcatalog.

and Citeseer, particularly in the former network. In contrast, EGC-VGAE exhibits better performance on Pubmed and Blogcatalog. This improvement can be attributed to the stable changes in the topology

structure, which enhance the accuracy of community detection of slice- $(i-1)$  and ultimately increase the confidence in pairwise constraints of slice- $i$ .



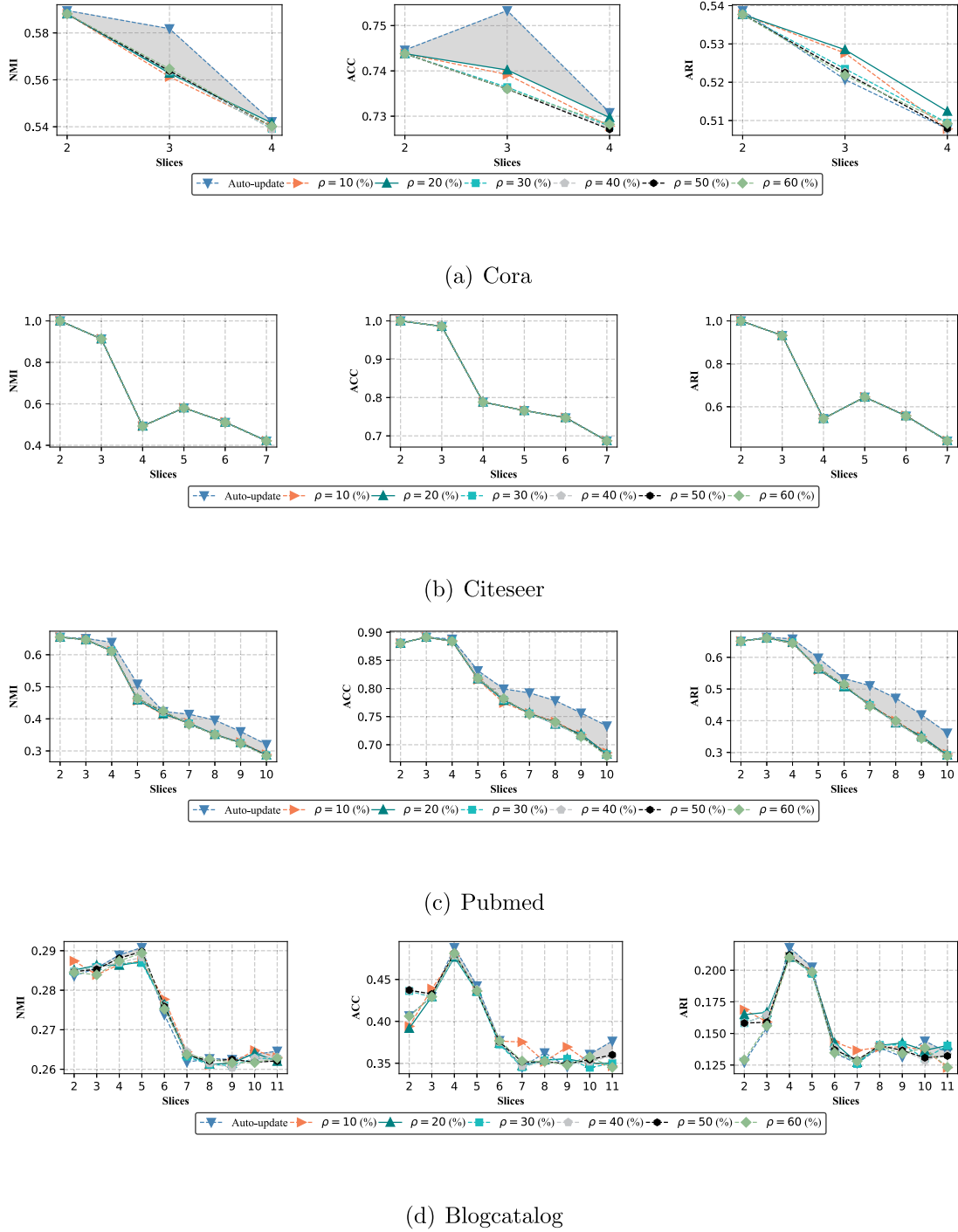


Fig. 7. Effect of different probabilities  $\rho$  on the performance of EGC-VGAE.

#### 4.3. Robustness experiment of EG-VGAE and EGC-VGAE

We conduct robustness experiments to evaluate the ability of our methods, EG-VGAE and EGC-VGAE, to resist attribute noise. In the experimental setup, we randomly permute the attributes of nodes according to ratios of 5%, 10%, 15%, 20%, 25%, 30%, 35% and 40% to conduct 8 groups of experiments. The experimental results are shown in Fig. 6. We can observe that the spread range of performance on Pubmed and Blogcatalog is significantly smaller for both EG-VGAE and EGC-VGAE compared to Cora and Citeseer. In contrast, the spread range of performance on Cora and Citeseer appears to be wider, but

mostly no more than 8%. In fact, relevant literature [36] has confirmed that the effectiveness of GCN is attributed to the homogeneity assumption. Therefore, when there are large differences in the node attributes within the same community, the performance of methods tend to decrease. However, experimental results demonstrate that as attribute noise increases in networks, the performance of our methods experiences only a decline slowly, rather than in a cliff-like manner. We analyze the reasons behind this experimental result and find that the robustness of our models can be attributed to the help of evolution information. Specifically, when our methods are sensitive to the evolution information on the network, the spread range of performance is small.

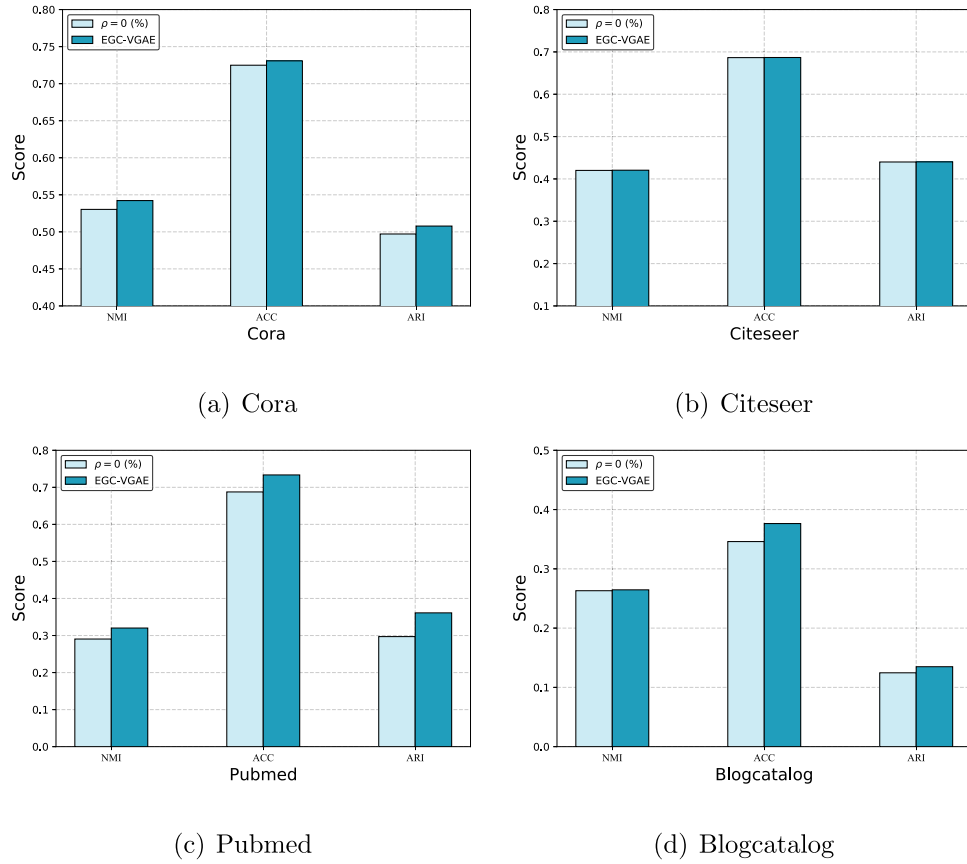


Fig. 8. Ablation experiment: Influence of whether  $\rho$  is zero or not on performance of EGC-VGAE.

Conversely, the spread range of performance is slightly larger when sensitivity of methods to evolution information is relatively lower.

#### 4.4. The influence of probability $\rho$ in EGC-VGAE

We argue that network noise can easily affect the accuracy of community detection results. To mitigate the bias of smoothness constraint caused by this result, we introduce a probability constraint denoted  $\rho$  in the construction of  $\mathbf{H}$ , as described in Eq. (23). In this sub-section, we will demonstrate that imposing probability constraint  $\rho$  on  $\mathbf{H}$  is reasonable. Subsequently, we will determine a probability value based on a predefined ratio to verify the effectiveness of our setting for  $\rho$ .

(1) **Whether  $\rho$  is zero or not on the performance of EGC-VGAE.** Fig. 8 shows the effect of the presence or absence of probability constraint  $\rho$  on the performance of our method. We can observe that, for all datasets, the control group with constraints all outperformed the experimental group without constraints, especially in Pubmed and Blogcatalog. Therefore, we have reason to believe that imposing probability constraint  $\rho$  on  $\mathbf{H}$  is reasonably efficient. We believe that factors, such as network noise and the insensitivity of our method to partial network evolution information, can result in anomalies during the construction of the smoothness constraint  $\mathbf{H}$ . These anomalies accumulate and are propagated across each slice, eventually impacting the performance of our method. Therefore, to mitigate the influence of these outliers on performance, a straightforward yet effective approach is to discard a certain ratio of the smoothness constraint information.

(2) **Effect of different  $\rho$  on the performance of EGC-VGAE.** For all datasets, we conduct six groups of experiments by setting the probability constraint  $\rho$  to 10%, 20%, 30%, 40%, 50% and 60%. We compare these experiments with our proposed method, using NMI to calculate the adaptation ratio for updates. The results are shown in Fig. 7, where the shaded areas represent the difference range. We can

draw the following conclusions. If our method exhibits sensitivity to the evolution information of the specific dataset, it shows almost insensitivity to the probability constraint. Conversely, the adaptive update strategy enables EGC-VGAE to perform better in each slice. Therefore, our ratio setting is both reasonable and effective. Furthermore, this approach eliminates the need to introduce additional hyperparameters, thus avoiding an increase in the complexity of our method.

#### 4.5. Effect of different $\mathbf{H}$ values on the performance of EGC-VGAE

In our experimental setup, we assign a constant value of 1 to  $\mathbf{H}_{\omega,\varphi}$  if sample  $\omega$  and  $\varphi$  have a pairwise relationship. In this section, we explore the performance impact of different constant assignments on EGC-VGAE, where the constants are set to [0.7, 0.8, 0.9, 2, 3, 4, 5], respectively. The results are shown in Fig. 9, and we have the following findings: While different constant assignments lead to some degree of performance variation, we believe that small-scale performance changes are acceptable compared to the time spent on parameter adjustments, and our settings can already achieve sub-optimal. Specifically, on Cora, which exhibits the most significant performance change, EGC-VGAE exhibits performance variances of  $\pm 0.004$ ,  $\pm 0.025$  and  $\pm 0.017$  on NMI, ACC and ARI, respectively. Overall, similar to the setting of [37], we recommend directly assigning the constant 1 to  $\mathbf{H}_{\omega,\varphi}$  when the constraints are met, to avoid cumbersome parameter adjustments.

## 5. Related work

As far as our knowledge extends, variational autoencoders (VAEs) can be traced back to the literature [29], which is a class of autoencoder method based on Variational Bayesian theory. The central challenge for this type of method lies in effectively utilizing continuous random

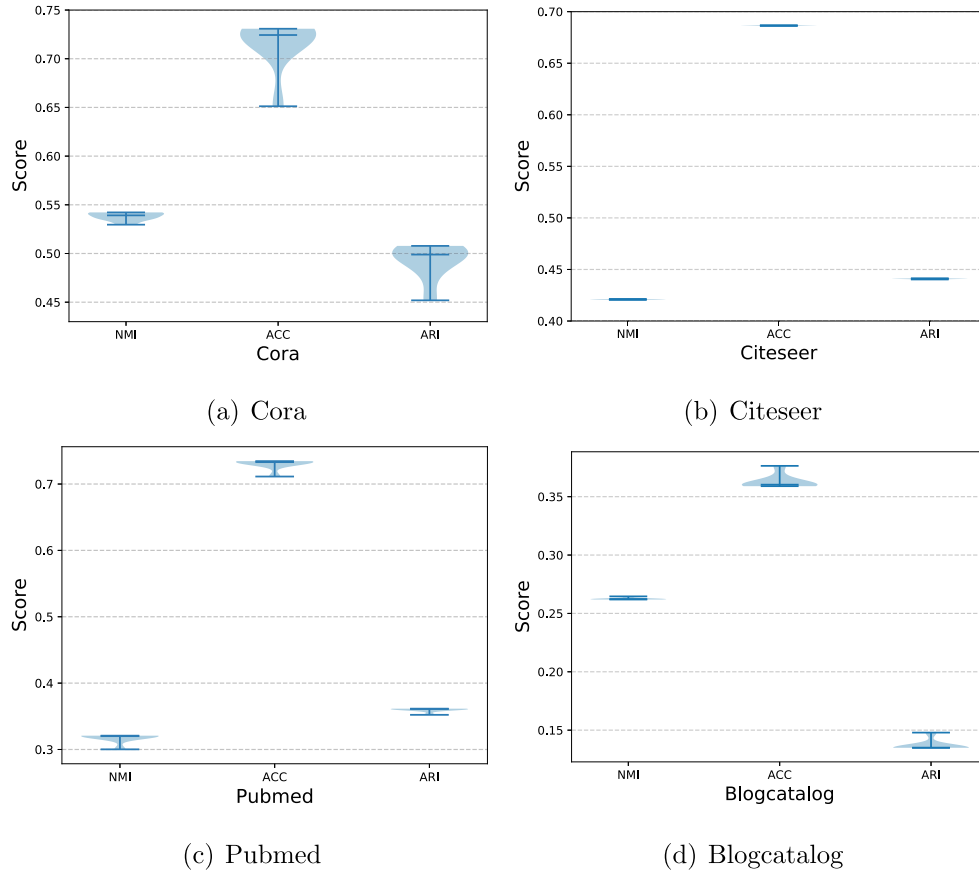


Fig. 9. Effect of different  $H$  values on the performance of EGC-VGAE.

variables and/or parameters to approximate inference and learning on complex posterior distributions. In the non-Euclidean space, many methods based on VAEs for clustering have been proposed, including VaDE [28], DGG [38] and DC-GMM [37], etc.

With the remarkable impact of the GNNs in the graph domain, VAEs are also extended to the Euclidean space and are called VGAEs, which can be traced back to the literature [16]. After that, Qiu et al. [17] proposed the VGAE method, which combines high-order modularity information and node attribute information, emphasizing that the posterior distribution should include the community information of the node. Pan et al. [33] proposed adversarial graph embedding framework, which the posterior distribution enforce to match prior distribution via an adversarial training scheme. In addition, for the task of community detection, Choong et al. [34] made the assumption that the community assignment relationship of nodes can be represented by the mixture of gaussian distributions. Each gaussian distribution has its own mean and variance. When a node has the highest probability or weight associated with a particular Gaussian distribution, it will be assigned to the corresponding community. However, it is unfortunate that their paper does not provide details on how to train the parameters of the gaussian mixture model. Building on this progress, Hui et al. [31] investigated the use of unsupervised learning to enhance the performance of semi-supervised learning methods, and proposed CGCN for community detection in an end-to-end manner. At the same time, their method also inspires us how to simply yet effectively train the parameters of the gaussian mixture model under unsupervised learning. Hu et al. [39] introduced a hierarchical structure with three layers for nodes, attributes and values. They proposed a new method based on the Bayesian framework, which offers both flexibility and interpretability. Additionally, this research has provided inspiration for us in designing a multi-level variational framework. In additional, there are also many

methods such as those introduced in literature [18,40,41], which are important parts of promoting the development of VGAE in the graph domain. More types of community detection methods, such as fuzzy-based methods [42,43] and non-negative matrix factorization-based methods [44,45], can be found in survey [46].

## 6. Conclusion and future work

In this paper, we adopt a new vision, i.e., evolution on static networks, to re-examine the information of networks, and propose two novel methods, EG-VGAE and its variant EGC-VGAE. For these two methods, they realize the propagation of fine-grained low-frequency signals from the local to the global network with the assistance of evolution relations, which helps improve the accuracy of node community assignment. Furthermore, we introduce smoothness constraints in EGC-VGAE, enhancing its sensitivity to evolution information in adjacent slice compared to EG-VGAE.

Although our methods show effectiveness on most benchmarks, a larger number of slices can pose challenges during training. To this end, we have two considerations for future work. On the one hand, under the premise of not destroying the evolution relationship, we plan to keep only slices with rich information in order to reduce the time overhead during training. For example, some continuous networks are usually divided into discrete networks based on certain rules, and these rules can be introduced or improved for merging slices. On the other hand, we are considering whether the evolution information can be used as a pre-training as part, so that valuable information can be integrated into the pairwise constraints matrix.

## CRediT authorship contribution statement

**Junwei Cheng:** Conceptualization, Methodology, Software, Investigation, Writing – original draft, Writing – review & editing. **Chaobo He:** Data curation, Investigation, Methodology. **Kunlin Han:** Writing – review & editing. **Gangbin Chen:** Data curation. **Wanying Liang:** Data curation. **Yong Tang:** Supervision.

## Declaration of competing interest

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.

## Data availability

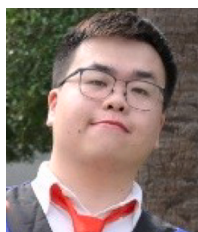
Data will be made available on request.

## Acknowledgments

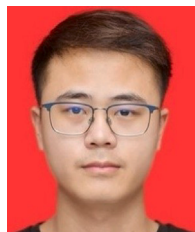
This work was supported in part by the National Natural Science Foundation of China under Grant 62077045, and Grant U1811263.

## References

- [1] N.D. Soares, R.M.M. Braga, J.M.N. David, K.B. Siqueira, V.S.A. Menezes, Data analysis in social networks for agribusiness: A systematic review, *IEEE Access* 11 (2023) 8422–8432, <http://dx.doi.org/10.1109/ACCESS.2023.3237984>.
- [2] S. Razaq, A.K. Malik, B. Raza, H.A. Khattak, G.W.M. Zegar, Y.D. Zelada, Research collaboration influence analysis using dynamic co-authorship and citation networks, *Int. J. Interact. Multim. Artif. Intell.* 7 (3) (2022) 103, <http://dx.doi.org/10.9781/ijimai.2022.03.001>.
- [3] R. Francisquini, A.C. Lorena, M.C. Nascimento, Community-based anomaly detection using spectral graph filtering, *Appl. Soft Comput.* 118 (2022) 108489, <http://dx.doi.org/10.1016/j.asoc.2022.108489>.
- [4] C. Choudhary, I. Singh, M. Kumar, Community detection algorithms for recommendation systems: Techniques and metrics, *Computing* 105 (2) (2023) 417–453, <http://dx.doi.org/10.1007/s00607-022-01131-z>.
- [5] F. Liu, S. Xue, J. Wu, C. Zhou, W. Hu, C. Paris, S. Nepal, J. Yang, P.S. Yu, Deep learning for community detection: Progress, challenges and opportunities, in: *IJCAI 2020*, *ijcai.org*, 2020, pp. 4981–4987, <http://dx.doi.org/10.24963/ijcai.2020/693>.
- [6] C. He, X. Fei, Q. Cheng, H. Li, Z. Hu, Y. Tang, A survey of community detection in complex networks using nonnegative matrix factorization, *IEEE Trans. Comput. Soc. Syst.* 9 (2) (2022) 440–457, <http://dx.doi.org/10.1109/TCSS.2021.3114419>.
- [7] D. Jin, Z. Yu, P. Jiao, S. Pan, D. He, J. Wu, P.S. Yu, W. Zhang, A survey of community detection approaches: From statistical modeling to deep learning, *IEEE Trans. Knowl. Data Eng.* 35 (2) (2023) 1149–1170, <http://dx.doi.org/10.1109/TKDE.2021.3104155>.
- [8] J. Chavoshinejad, S.A. Seyedi, F. Akhlaghian Tab, N. Salahan, Self-supervised semi-supervised nonnegative matrix factorization for data clustering, *Pattern Recognit.* 137 (2023) 109282, <http://dx.doi.org/10.1016/j.patcog.2022.109282>.
- [9] K. Berahmand, M. Mohammadi, F. Saberi-Movahed, Y. Li, Y. Xu, Graph regularized nonnegative matrix factorization for community detection in attributed networks, *IEEE Trans. Netw. Sci. Eng.* 10 (1) (2023) 372–385, <http://dx.doi.org/10.1109/TNSE.2022.3210233>.
- [10] X. Zhou, L. Su, X. Li, Z. Zhao, C. Li, Community detection based on unsupervised attributed network embedding, *Expert Syst. Appl.* 213 (Part) (2023) 118937, <http://dx.doi.org/10.1016/j.eswa.2022.118937>.
- [11] L. Guo, Q. Dai, Graph clustering via variational graph embedding, *Pattern Recognit.* 122 (2022) 108334, <http://dx.doi.org/10.1016/j.patcog.2021.108334>.
- [12] Z. Zhao, H. Zhou, L. Qi, L. Chang, M. Zhou, Inductive representation learning via CNN for partially-unseen attributed networks, *IEEE Trans. Netw. Sci. Eng.* 8 (1) (2021) 695–706, <http://dx.doi.org/10.1109/TNSE.2020.3048902>.
- [13] B. Jing, C. Park, H. Tong, HDML: High-order deep multiplex infomax, in: *WWW*, 2021, pp. 2414–2424, <http://dx.doi.org/10.1145/3442381.3449971>.
- [14] C. Fattal, L. Labiod, M. Nadif, Efficient graph convolution for joint node representation learning and clustering, in: *WSDM*, *ACM*, 2022, pp. 289–297, <http://dx.doi.org/10.1145/3488560.3498533>.
- [15] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: *International Conference on Learning Representations*, 2017.
- [16] T.N. Kipf, M. Welling, Variational graph auto-encoders, 2016, *CoRR* abs/1611.07308, [arXiv:1611.07308](http://arxiv.org/abs/1611.07308).
- [17] C. Qiu, Z. Huang, W. Xu, H. Li, VGER: Graph neural network reconstruction based community detection, 2022, *CoRR* abs/2201.04066.
- [18] N. Mrabah, M. Bouguessa, R. Ksantini, Escaping feature twist: A variational graph auto-encoder for node clustering, in: *IJCAI*, 2022, pp. 3351–3357, <http://dx.doi.org/10.24963/ijcai.2022/465>, Main Track.
- [19] G. Rossetti, R. Cazabet, Community discovery in dynamic networks: A survey, *ACM Comput. Surv.* 51 (2) (2018) 35:1–35:37, <http://dx.doi.org/10.1145/3172867>.
- [20] G. Palla, A. Barabási, T. Vicsek, Quantifying social group evolution, *Nature* 446 (2007) 664–667, <http://dx.doi.org/10.1038/nature05670>.
- [21] Y. Kong, G. Shi, R. Wu, Y. Zhang, K-core: Theories and applications, *Phys. Rep.* 832 (2019) 1–32, <http://dx.doi.org/10.1016/j.physrep.2019.10.004>.
- [22] E. Lukacs, Analytical methods in probability theory, in: *Symposium on Probability Methods in Analysis*, Springer, 1967, pp. 208–238.
- [23] A. Clauset, C.R. Shalizi, M.E.J. Newman, Power-law distributions in empirical data, *SIAM Rev.* 51 (4) (2009) 661–703, <http://dx.doi.org/10.1137/070710111>.
- [24] K. Shin, T. Eliassi-Rad, C. Faloutsos, CoreScope: Graph mining using k-core analysis - patterns, anomalies and algorithms, in: *ICDM*, *IEEE Computer Society*, 2016, pp. 469–478, <http://dx.doi.org/10.1109/ICDM.2016.0058>.
- [25] G. Mei, J. Tu, L. Xiao, F. Piccialli, An efficient graph clustering algorithm by exploiting k-core decomposition and motifs, *Comput. Electr. Eng.* 96 (Part) (2021) 107564, <http://dx.doi.org/10.1016/j.compeleceng.2021.107564>.
- [26] G. Palla, A.L. Barabási, T. Vicsek, Quantifying social group evolution, *Nature* 446 (7136) (2007) 664–667.
- [27] G. Nikolentzos, P. Meladianos, S. Limnios, M. Vazirgiannis, A degeneracy framework for graph similarity, in: *IJCAI*, *ijcai.org*, 2018, pp. 2595–2601, <http://dx.doi.org/10.24963/ijcai.2018/360>.
- [28] Z. Jiang, Y. Zheng, H. Tan, B. Tang, H. Zhou, Variational deep embedding: An unsupervised and generative approach to clustering, in: *IJCAI*, *ijcai.org*, 2017, pp. 1965–1972.
- [29] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, in: *ICLR*, 2014.
- [30] S. Kullback, R.A. Leibler, On information and sufficiency, *Ann. Math. Stat.* 22 (1) (1951) 79–86.
- [31] B. Hui, P. Zhu, Q. Hu, Collaborative graph convolutional networks: Unsupervised learning meets semi-supervised learning, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, (no. 04) 2020, pp. 4215–4222.
- [32] N. Mrabah, M. Bouguessa, M.F. Touati, R. Ksantini, Rethinking graph auto-encoder models for attributed graph clustering, *IEEE Trans. Knowl. Data Eng.* 35 (9) (2023) 9037–9053, <http://dx.doi.org/10.1109/TKDE.2022.3220948>.
- [33] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, C. Zhang, Adversarially regularized graph autoencoder for graph embedding, in: *IJCAI*, *ijcai.org*, 2018, pp. 2609–2615, <http://dx.doi.org/10.24963/ijcai.2018/362>.
- [34] J.J. Choong, X. Liu, T. Murata, Learning community structure with variational autoencoder, in: *ICDM*, *IEEE*, 2018, pp. 69–78.
- [35] G. Niknam, S. Molaei, H. Zare, D. Clifton, S. Pan, Graph representation learning based on deep generative gaussian mixture models, *Neurocomputing* 523 (2023) 157–169, <http://dx.doi.org/10.1016/j.neucom.2022.11.087>.
- [36] H. N.T., T. Maehara, Revisiting graph neural networks: All we have is low-pass filters, 2019, *CoRR* abs/1905.09550, [arXiv:1905.09550](http://arxiv.org/abs/1905.09550).
- [37] L. Manduchi, K. Chin-Cheong, H. Michel, S. Wellmann, J.E. Vogt, Deep conditional Gaussian mixture model for constrained clustering, in: *NeurIPS*, 2021, pp. 11303–11314.
- [38] L. Yang, N. Cheung, J. Li, J. Fang, Deep clustering by Gaussian mixture variational autoencoders with graph embedding, in: *ICCV*, *IEEE*, 2019, pp. 6439–6448, <http://dx.doi.org/10.1109/ICCV.2019.00654>.
- [39] L. Hu, K.C.C. Chan, X. Yuan, S. Xiong, A variational Bayesian framework for cluster analysis in a complex network, *IEEE Trans. Knowl. Data Eng.* 32 (11) (2020) 2115–2128, <http://dx.doi.org/10.1109/TKDE.2019.2914200>.
- [40] S.J. Ahn, M. Kim, Variational graph normalized autoencoders, in: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 2827–2831.
- [41] S. Yang, S. Verma, B. Cai, J. Jiang, K. Yu, F. Chen, S. Yu, Variational co-embedding learning for attributed network clustering, *Knowl.-Based Syst.* 270 (2023) 110530.
- [42] L. Hu, Y. Yang, Z. Tang, Y. He, X. Luo, FCAN-MOPSO: An improved fuzzy-based graph clustering algorithm for complex networks with multiobjective particle swarm optimization, *IEEE Trans. Fuzzy Syst.* 31 (10) (2023) 3470–3484, <http://dx.doi.org/10.1109/TFUZZ.2023.3259726>.
- [43] L. Hu, X. Pan, Z. Tang, X. Luo, A fast fuzzy clustering algorithm for complex networks via a generalized momentum method, *IEEE Trans. Fuzzy Syst.* 30 (9) (2022) 3473–3485, <http://dx.doi.org/10.1109/TFUZZ.2021.3117442>.
- [44] J. Cheng, Y. Tang, C. He, K. Han, Y. Li, J. Wei, Community detection in attributed networks via adaptive deep nonnegative matrix factorization, *Neural Comput. Appl.* (2023) 1–16.
- [45] S. Su, J. Guan, B. Chen, X. Huang, Nonnegative matrix factorization based on node centrality for community detection, *ACM Trans. Knowl. Discov. Data* 17 (6) (2023) <http://dx.doi.org/10.1145/3578520>.
- [46] X. Su, S. Xue, F. Liu, J. Wu, J. Yang, C. Zhou, W. Hu, C. Paris, S. Nepal, D. Jin, Q.Z. Sheng, P.S. Yu, A comprehensive survey on community detection with deep learning, *IEEE Trans. Neural Netw. Learn. Syst.* (2022) 1–21, <http://dx.doi.org/10.1109/TNNLS.2021.3137396>.



**Junwei Cheng** received his B.S. degree from School of Computer Science, University of Electronic Science and Technology of China, Zhongshan Institute, in 2020. He is currently a Ph.D. student in School of Computer Science, South China Normal University, China. His research interest is graph data mining.



**Gangbin Chen** received his B.S. degree from School of Computer Science, South China Normal University, China. Also, he is currently a postgraduate student in School of Computer Science, South China Normal University, China. His research interest is graph representation learning.



**Chaobo He** received his Ph.D., M.S. and B.S. degree from South China Normal University, China in 2014, 2007 and 2004, respectively. He is currently a professor with the School of Computer Science, South China Normal University, and is also a visiting scholar in School of Data and Computer Science, Sun Yat-sen University, China. His research interests are machine learning and social computing. He has published over 20 papers on international journals and conferences.



**Wanying Liang** received her B.S. degree and currently a postgraduate student from School of Computer Science, South China Normal University, China. Her research interests are social network community detection and multi-modal learning.



**Kunlin Han** received the B.S. degree from the School of Computer Science, South China Normal University, China, in 2021. He is currently working towards the M.Sc. degree with the Department of Computer Science, University of Southern California, United States. His research interests are machine learning and computer architecture.



**Yong Tang** received the BS and MS degrees from Wuhan University, China, in 1985 and 1990, respectively, and the Ph. D. degree from the University of Science and Technology of China, China, in 2001, all in computer science. He is currently a professor of the School of Computer Science, South China Normal University, China. His research interests are big data and collaborative computing. He has published more than 100 papers on international journals and conferences.