VGGM: Variational Graph Gaussian Mixture Model for Unsupervised Change Point Detection in Dynamic Networks

Xinxun Zhang, Pengfei Jiao, *Member, IEEE*, Mengzhou Gao, Tianpeng Li, Yiming Wu, Huaming Wu, *Senior Member, IEEE*, and Zhidong Zhao, *Member, IEEE*

***Abstract*— Change point detection in dynamic networks aims to detect the points of sudden change or abnormal events within the network. It has garnered substantial interest from researchers due to its potential to enhance the stability and reliability of real-world networks. Most change point detection methods are based on statistical characteristics and phased training, and some methods are required to set the percent of change points. Meanwhile, existing methods for change point detection suffer from two limitations. On one hand, they struggle to extract snapshot features that are crucial for accurate change point detection, thereby limiting their overall effectiveness. On the other hand, they are typically tailored for specific network types and lack the versatility to adapt to networks of varying scales. To solve these issues, we propose a novel unified end-to-end framework called Variational Graph Gaussian Mixture model (VGGM) for change point detection in dynamic networks. Specifically, VGGM combines Variational Graph Auto-Encoder (VGAE) and Gaussian Mixture Model (GMM) through joint training, incorporating a Mixture-of-Gaussians prior to model dynamic networks. This approach yields highly effective snapshot embeddings via VGAE and a dedicated readout function, while automating change point detection through GMM. The experimental results, conducted on both real-world and synthetic datasets, clearly demonstrate the superiority of our model in comparison to the current state-of-the-art methods for change point detection.**

***Index Terms*— Change point detection, abnormal events, unsupervised learning, graph neural networks, dynamic networks.**

I. Introduction

DYNAMIC networks are ubiquitous in real world. Social networks [1], [2], transcriptional regulation networks [3], financial transactions [4], and trade networks [5] can all be characterized by dynamic networks. Modeling dynamic networks is critical as it describes the way networks evolve and interact over time, which helps us understand the evolutionary patterns and predict future behavior of real-world systems [6], [7]. Anomaly detection in dynamic networks is challenging but of high importance since it can effectively safeguard real-world networks from the threats caused by system faults and attacks [8], [9]. Change point detection is an essential task of anomaly detection in dynamic networks, which aims to discover sudden change points or events that deviate from normal patterns and helps to secure real-world networks against sabotage [10]. Change point detection plays a crucial role in enhancing the understanding and security of dynamic systems and has numerous practical applications across multiple domains. For instance, attack detection in Cyber-Physical Systems [11], behavioral detection in cyber security [12], finance fraud detection [13], pathological detection in medical images [14] and so on.

Existing change point detection methods are mainly divided into two categories [15], i.e., generative methods and feature-based methods. For instance, Peel and Clauset [16] developed the GHRG model, which simulates the dynamic network structure by hierarchical random graphs and detects deviations from normal behavior by sliding windows. LAD [17] used singular value decomposition of the graph Laplacian operator as the low-dimensional representation of the snapshot. CICPD [15] applied PageRank [18] to calculate the PageRank value of each snapshot as the low-dimensional representation. However, these methods require manually constructing the snapshot feature vectors as the low-dimensional representation, and some need to set the prior information of change points. These limitations hinder the generalizability of traditional detection methods, confining them to specific dynamic networks and impeding their scalability to networks of different characteristics and scales. This becomes especially evident in larger-scale dynamic networks, where the performance of these methods deteriorates significantly.

In recent years, deep learning-based methods [19], [20], [21] have been widely applied to anomaly detection in various domains. Compared to traditional methods, deep

Manuscript received 6 October 2023; revised 27 February 2024; accepted 11 March 2024. Date of publication 18 March 2024; date of current version 6 May 2024. This work was supported in part by Zhejiang Provincial Natural Science Foundation of China under Grant LDT23F01012F01 and Grant LDT23F01015F01; in part by the Fundamental Research Funds for Provincial Universities of Zhejiang under Grant GK229909299001-008; in part by the National Natural Science Foundation of China under Grant 62372146, Grant 62003120, and Grant 6207132; and in part by Zhejiang Laboratory Open Research Project under Grant K2022QA0AB01. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Z. Berkay Celik. *(Corresponding author: Pengfei Jiao.)*

Xinxun Zhang, Pengfei Jiao, Mengzhou Gao, Yiming Wu, and Zhidong Zhao are with the School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China (e-mail: xxzhang@hdu.edu.cn; pjiao@hdu.edu.cn; mzgao@hdu.edu.cn; ymwu@hdu.edu.cn; zhaozd@hdu.edu.cn).

Tianpeng Li is with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China (e-mail: ltpnimeia@tju.edu.cn).

Huaming Wu is with the Center for Applied Mathematics, Tianjin University, Tianjin 300072, China (e-mail: whming@tju.edu.cn).

Digital Object Identifier 10.1109/TIFS.2024.3377548

learning-based detection methods automatically extract features by neural networks [22] and have demonstrated impressive performance on larger-scale datasets. For example, Netwalk [23] transposes vertex encoding into a low-dimensional vector representation using clique embeddings to detect anomalies in dynamic networks. However, it is a multi-stage detection method, and its embedding generation process is relatively complex. Some methods [24], [25] view change point detection as a time series anomaly detection task and use deep sequential models for supervised or semi-supervised learning. However, due to the limited number of network snapshots and the significant changes in network patterns before and after change points, the performance is often unsatisfactory. In addition, due to some events in the real world not being recorded, the manual label may hinder supervised learning methods from deeply exploring hidden evolving patterns in dynamic networks, making it challenging to generalize change point detection in dynamic networks. Therefore, it is more suitable to detect change points in dynamic networks through unsupervised learning methods. Consequently, we view this task as an unsupervised clustering problem in this paper.

Graph Neural Networks (GNNs) have shown powerful feature extraction ability, particularly when dealing with data structured as graphs [26], [27], [28], [29], [30]. Dynamic networks inherently exhibit temporal evolution in their graph structure [31], [32], [33], [34], making GNN-based approaches well-suited for the task of change point detection within dynamic networks. To the best of our knowledge, only one previous work [35] has explored the utilization of GNNs for change point detection in dynamic networks, and it employs a supervised learning methodology. However, based on our previous analysis, unsupervised detection methods are more appropriate for detecting change points in dynamic networks. In summary, change point detection still faces some challenges as follows: i) Most existing change point detection methods are conducted in a staged manner, lacking a complete end-to-end framework that can adapt to dynamic networks with varying scales. ii) The inherent nature of change point detection suggests a preference for unsupervised learning methods. However, the absence of sufficient supervised information makes it more challenging to learn effective embeddings and discover change points in an unsupervised manner.

Building upon the aforementioned challenges, we propose a novel Variational Graph Gaussian Mixture model (VGGM) for change point detection in dynamic networks. VGGM is an end-to-end unsupervised model achieved through jointly iterative training Variational Graph Auto-Encoder (VGAE) [36] and Gaussian Mixture Model (GMM) [37], which performs well on both real-world and synthetic networks with different scales. VGGM automatically discovers change points by GMM without setting additional parameters. And the mechanism of joint training guarantees the effectiveness of snapshot embedding. Specifically, We obtain node embeddings by extracting the network's topological structure information through VGAE. Then, we obtain snapshot embeddings that preserve the information of network structural changes through a specific readout function. Next, we update the prior distribution in VGAE by Gaussian mixture distribution obtained from GMM and assign clustering labels for each snapshot. Finally, we discover change points according to the clustering labels and the temporal information. To summarize, the main contributions of our work are:

- We propose a novel unified framework for change point detection in dynamic networks, called VGGM, which is capable of handling dynamic networks of varying scales. To the best of our knowledge, we are the first to propose an end-to-end unsupervised approach for change point detection in dynamic networks.
- We jointly optimize VGAE and GMM through a way of iterative training mechanism. The snapshot embedding obtained under this constraint of joint iterative training will capture more essential network information and be more effective for change point detection.
- We conduct experiments on three real-world dynamic networks and two synthetic networks with ground truth change points. Experiment results validate the superiority and excellent detection ability of our model.

The remainder of this paper is structured as follows: In Section II, we provide a brief overview of prior research on change point detection in dynamic networks. Section III outlines the problem definition and provides an in-depth description of our model. Section IV presents the experimental results along with additional analysis. Finally, Section V concludes and initiates a discussion regarding our work.

II. Related Work

A. Generative Methods

Generative methods explore change point detection in dynamic networks with a specific probabilistic architecture, which seeks to discover potential change points in the latent space. The difference between generative methods in adopting various probabilistic generative models.

Peel and Clauset [16] introduced GHRG to pay more attention to hierarchical network structures. To accurately and quantitatively detect the change points, a generalized hierarchical random graph model is combined with the Bayesian hypothesis test to estimate model parameters in CHRG. Wang et al. [38] proposed EdgeMonitering, which is a more general generative model for accommodating another snapshot model. EdgeMonitering captures the temporal dependency while computing the similarity between the snapshots with the first-order Markov process. Miller and Mokryn [39] adopted a size-agnostic model to detect change points in the dynamic network, namely SizeCPD. SizeCPD doesn't have any professional and prior knowledge about the network, even the temporal dependency relations. It computes the degree distribution of snapshots with sliding windows and then conducts hypothesis testing to discover change points. Jiao et al. [40] proposed a new generative evolutionary model that takes into account the interaction of anomalies at different levels called GEABS. Specifically, GEABS discovers the correlation of network macroscopic, mesoscopic, and micro-scale with the Stochastic Block Model (SBM). Although these methods have made promising results in dynamic networks, parameter

estimation is required for these generative methods, which usually bring high computational complexity and memory cost.

B. Feature-Based Methods

The main idea of feature-based methods is to directly extract feature vectors that contain the network's essential structure information from snapshots. Compared to generative methods, these approaches have lower computational complexity and memory cost. However, the quality of the extracted features plays a crucial role in determining the detection performance for change point detection in dynamic networks. Therefore, how to extract features that comprehensively represent the entire network is the key challenge for feature-based methods. One such feature-based method is DeltaCon [41], which leverages the similarity between two snapshots to detect change points in dynamic networks. DeltaCon utilizes affinity matrices of pairwise vertices as the feature vectors for each snapshot. Affinity matrices consider more neighboring nodes, allowing them to capture richer network information. Another method, proposed by Zhu et al. [15], computes the PageRank value of every snapshot using the PageRank algorithm [18] as the feature vectors. Subsequently, it constructs a graph by computing the similarity between the features of each snapshot, transforming the change point detection problem into a community detection problem that can be dealt with through spectral methods. Huang et al. [17] introduced a graph Laplacian-based detection framework. This approach utilizes Singular Value Decomposition to assign low-dimension graph representation for each snapshot. It captures both gradual and abrupt changes by stimulating the long-term and short-term dependencies in dynamic evolution by two sliding windows.

Some feature methods based on deep learning have also emerged. For instance, Netwalk [23] transposes vertex encoding into a low-dimensional vector representation by using clique embeddings that minimize pairwise distances between vertices in each random walk, and then utilizes a dynamic clustering algorithm to discover change points. Sulem et al. [35] proposed a graph similarity learning model based on a Siamese graph neural network. it's capable of handling any available node attributes and detecting local and global changes in dynamic networks. Although these methods can effectively detect change points in dynamic networks, some limitations hinder their generalizability, such as the demand for manual feature engineering, additional parameters related to change points, and staged training, among other issues.

III. Methodology

In this section, we first formalize the formal problem definition for change point detection in dynamic networks and then introduce the key concepts and symbols used in this paper. In addition, we elaborate on the concrete details of the proposed model. Table I summarizes the major notations used in this paper.

A. Problem Definition

We represent a dynamic network as a series of snapshots, denoted by $G = \{G_0, G_2, \ldots, G_{T-1}\}$, where T is the total number of snapshots. G_t represents the snapshot at time t, characterized by a node-set V_t and an edge-set E_t, such that $G_t = \{V_t, E_t, \mathbf{A}_t\}$. Here, $\mathbf{A}_t$ is the adjacency matrix of G_t. In dynamic networks, nodes and edges may emerge or vanish over time as the network evolves. We use $\mathbf{X}_t \in \mathbb{R}^{N\times D}$ to represent the feature matrix of snapshot G_t, with D indicating the dimensionality of the feature matrix. Our model assigns clustering labels to all snapshots, denoted as $K = \{K_0, K_2, \ldots, K_{M-1}\}$, where M is the number of clusters. Our goal is to identify those snapshots whose clustering labels differ from those of their preceding snapshots. These identified snapshots serve as change points within the dynamic network.

TABLE I
Major Notations in This Paper

Symbol	Description
G, G_t	A dynamic network and snapshot at timestamp t
N	The number of nodes of dynamic network G
$\mathbf{A}_t$	The adjacency matrix of G_t
V_t	The nodes of the snapshot G_t
E_t	The edges of the snapshot
$\mathbf{X}_t$	The feature matrix of the snapshot G_t
$\boldsymbol{\mu}_t^{enc}$	The mean vectors of G_t in VGAE
$\boldsymbol{\sigma}_t^{enc}$	The variance vectors of G_t in VGAE
$\hat{\mathbf{D}}$	Degree matrix
$\mathbf{I}$	Identity matrix
$\mathbf{Z}_t$	The node embedding of G_t
$\boldsymbol{z}_t$	The embedding representation of snapshot G_t
$\mathcal{R}$	The Readout function
$\boldsymbol{\mu}_t^{GMM}$	The mean vector of G_t in GMM
$\boldsymbol{\Sigma}_t^{GMM}$	The covariance matrix of G_t in GMM
w_t^i	The mixture weight of component i in G_t
K^i	The i-th cluster is obtained through GMM.

B. Proposed Method

The overall structure of VGGM is depicted in Fig. 1. Our model consists of five main parts: VGAE, Readout Function, GMM, Joint Iterative Training Mechanism, and the Change Point Detection Algorithm. In the following sections, we will offer an in-depth explanation of the operation and functionality of each component.

1) VGAE: VGAE is a probabilistic generative model that adopts a two-layer GNN encoder and an inner product decoder. VGAE combines ideas from both graph autoencoders and variational autoencoders to capture the topology structure information within a given graph. It allows for the extraction of meaningful and informative node embeddings while accounting for uncertainty, making it a valuable tool in graph representation learning and generative modeling. VGAE learns the node embedding of graph data by optimizing the reconstruction loss and KL divergence between the approximate posterior distribution and a single Gaussian prior distribution. Due to the simplicity and effectiveness of VGAE, we employ VGAE to extract node embedding. In this work, we adopt GCN [42] as the GNN encoder in VGAE. Specifically, for a snapshot G_t, we feed $\mathbf{X}_t$ and $\mathbf{A}_t$ of G_t into the GCN encoder

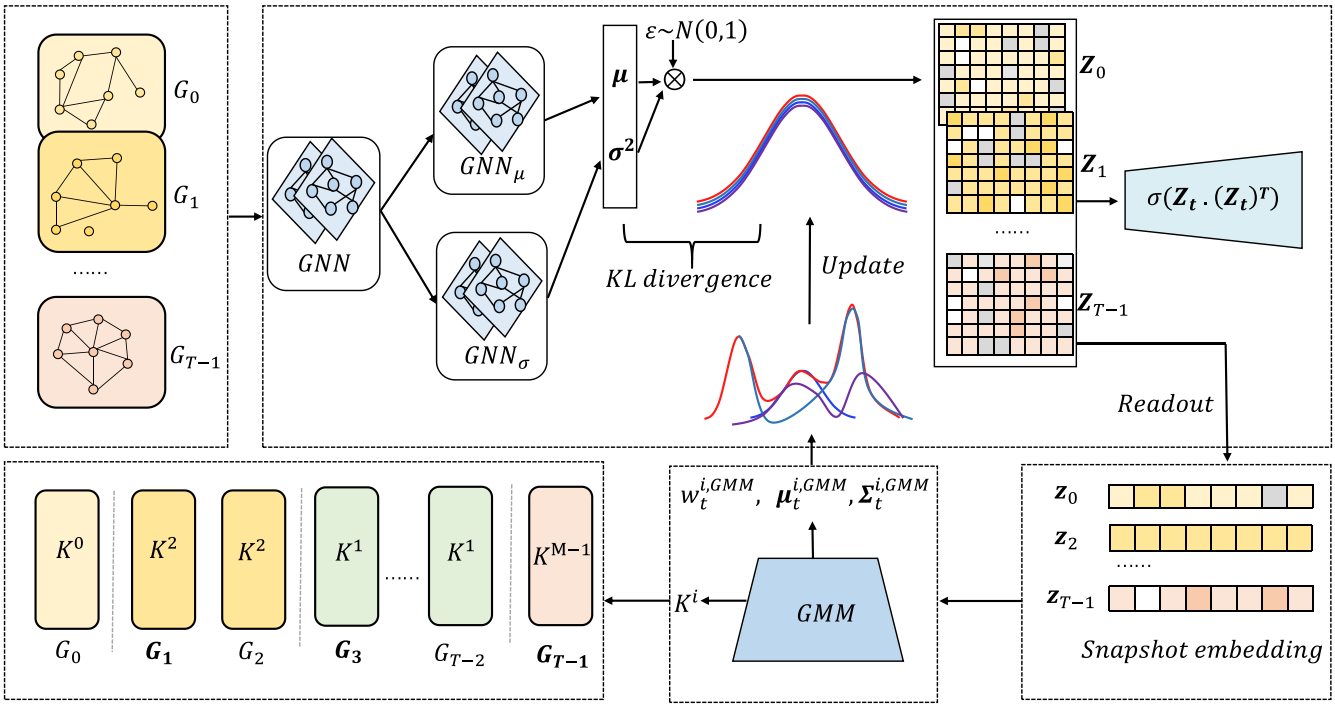


Fig. 1. Overview of VGGM framework, which detects change points in dynamic networks through the collaboration of five parts. First, node embedding is obtained by VGAE. And then we get snapshot embedding with a special Readout function. Next, we get the Gaussian mixture distribution by GMM to update the single Gaussian prior in VGAE. At last, we discover the change points with clustering results and temporal information of the dynamic network.

to obtain the latent variable $\mathbf{H}$.

$$\mathbf{H} = \delta(\hat{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}_t\hat{\mathbf{D}}^{-1/2}\mathbf{X}_t\mathbf{W}), \tag{1}$$

where $\widetilde{\mathbf{A}}_t = \mathbf{A}_t + \mathbf{I}$ denotes the adjacency matrix with inserted self-loops. $\mathbf{X}_t$ is the feature matrix, $\hat{\mathbf{D}}$ is the degree matrix, and the $\mathbf{W}$ is the parameters matrix that can be learned. δ is the activation function.

Then, we utilize two single-layer GCNs to encode the latent variable, yielding the mean vector $\boldsymbol{\mu}_t^{enc}$ and the variance vector $\boldsymbol{\sigma}_t^{enc}$ as follows:

$$\boldsymbol{\mu}_t^{enc} = \delta(\hat{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}_t\hat{\mathbf{D}}^{-1/2}\mathbf{H}\mathbf{W}_\mathbf{1}), \tag{2}$$

$$\boldsymbol{\sigma}_t^{enc} = \delta(\hat{\mathbf{D}}^{-1/2}\widetilde{\mathbf{A}}_t\hat{\mathbf{D}}^{-1/2}\mathbf{H}\mathbf{W}_\mathbf{2}). \tag{3}$$

Node embeddings $\mathbf{Z}_t$ can be sampled from posterior distribution $\mathcal{N} \sim (\boldsymbol{\mu}_t^{enc}, \boldsymbol{\sigma}_t^{enc})$. It is important to note that the sampling operation cannot be backpropagated through gradient descent because it is non-differentiable. Therefore, the reparameterization operation [43] is introduced to address this issue. Specifically, a variable $\boldsymbol{\varepsilon}$ that follows a standard normal distribution is introduced to transform the sampling operation into a linear operation, i.e., $\mathbf{Z}_t = \boldsymbol{\varepsilon} * \boldsymbol{\sigma}_t^{enc} + \boldsymbol{\mu}_t^{enc}$, where $*$ is the denotes the dot product. This design ensures that the model can perform backpropagation.

Based on the above statements, we can formalize the variational inference process achieved through a two-layer GCN encoder as follows:

$$q(\mathbf{Z}_t|\mathbf{X}_t, \mathbf{A}_t) = \prod_{i=1}^{n} q(\mathbf{Z}_t^i|\mathbf{X}_t, \mathbf{A}_t), \tag{4}$$

$$q(\mathbf{Z}_t^i|\mathbf{X}_t, \mathbf{A}_t) = \mathcal{N}(\mathbf{Z}_t^i|\boldsymbol{\mu}_t^{i,enc}, \mathrm{diag}((\boldsymbol{\sigma}_t^{i,enc})^2)), \tag{5}$$

where $\mathbf{Z}_t^i$ is the i-th elements of $\mathbf{Z}_t$, $\boldsymbol{\mu}_t^{i,enc}$ and $\boldsymbol{\sigma}_t^{i,enc}$ are the i-th row of $\boldsymbol{\mu}_t^{enc}$ and $\boldsymbol{\sigma}_t^{enc}$, respectively.

An inner product between latent variables is adopted as the decoder to reconstruct the input graph's topological structure. More specifically,

$$\hat{\mathbf{A}}_t = \delta(\mathbf{Z}_t(\mathbf{Z}_t)^T), \tag{6}$$

where $\hat{\mathbf{A}}_t$ is the reconstructed adjacency matrix. δ is the logistic sigmoid function. We define $\mathbf{A}_t^{ij}$ are the elements of $\mathbf{A}_t$. The generative process can be formalized as,

$$p(\mathbf{A}_t|\mathbf{Z}_t) = \prod_{i=1}^{n}\prod_{j=1}^{n} p(\mathbf{A}_t^{ij}|\mathbf{Z}_t^i, \mathbf{Z}_t^j), \tag{7}$$

$$p(A_t^{ij} = 1|\mathbf{Z}_t^i, \mathbf{Z}_t^j) = \delta((\mathbf{Z}_t^i)^T\mathbf{Z}_t^j). \tag{8}$$

The initial variational lower bound of VGAE in our model consists of reconstruction loss and the KL divergence between the approximate posterior distribution $q(\mathbf{Z}_t|\mathbf{X}_t, \mathbf{A}_t)$ and the initialized prior distribution $P(\mathbf{Z}_t)$ i.e., single Gaussian distribution.

$$\mathcal{L}_{(1)} = \mathbb{E}_{q(\mathbf{Z}_t|\mathbf{X}_t,\mathbf{A}_t)}[p(\mathbf{A}_t|\mathbf{Z}_t)] - \mathrm{KL}[q(\mathbf{Z}_t|\mathbf{X}_t, \mathbf{A}_t)||p(\mathbf{Z}_t)]. \tag{9}$$

where $\mathrm{KL}[\cdot||\cdot]$ denotes the Kullback-Leibler (KL) divergence. The calculation of KL divergence will be updated in the subsequent joint iterative optimization. This will be introduced in detail in the joint iterative training subsection.

2) Readout Function: We acquire node embeddings $\mathbf{Z}_t$ through the training of VGAE. Subsequently, we employ a permutation-invariant readout function (i.e., pooling operation) $\mathcal{R}(\cdot)$ to obtain each snapshot embedding z_t.

Algorithm 1 Training Algorithm of VGGM

Input: Snapshots sequence $G_{t=0}^{T-1}$, feature matrix and adjacency matrix of each snapshot $\mathbf{X}_t, \mathbf{A}_t$, joint iteration training times r, single Gaussian distribution $p(\mathbf{Z}_t)$
Output: Clustering results $K_i (i = 0, 1, \ldots, M-1)$

```
Get node embedding with VGAE optimized by Eq. 9
Get snapshot representation with Readout function R(·) by Eq. 10
Compute mixture weights w_t^i, μ_t^{i,GMM} and covariance matrix Σ_t^{i,GMM} of GMM for each snapshot
Choose a cluster c from cat(w_t), each cluster meets N ~ (μ_t^{c,GMM} Σ_t^{c,GMM}) and denote it as p_m(Z_t)
Replace p(Z_t) with p_m(Z_t)
for i = 0 To r − 1 do
    Get node embedding with VGAE optimized by Eq. 14
    Repeat steps 2-5
end for
Get the clustering results K_i (i = 0, 1, ..., M − 1) by Eq. 13
return K_i (i = 0, 1, ..., M − 1)
```

Algorithm 2 Algorithm of CPD for Clustering Results

Input: Clustering results $K^i (i = 0, 1, 2, \ldots, M-1)$, $K_{(G_t)}$ is denoted as the clustering label of G_t
Output: Change points set T_{cpd}

```
initialize tag = K_(G_0), T_cpd = ∅, arrange clustering results by time
for t = 0 To T − 1 do
    if K_(G_t) = tag then
        G_t isn't a change point
        continue
    else
        T_cpd = T_cpd ∪ t
        tag = K_(G_t)
    end if
end for
return T_cpd
```

Different types of pooling operations can significantly affect the performance of change point detection. In the majority of cases, the number of nodes responsible for inducing changes in the network structure tends to be relatively small. In general, the alteration of a small subset of nodes can potentially trigger a substantial transformation in the entire network structure. Some studies analyzed the characteristics of anomalous nodes from a spectral domain perspective and found that a small portion of anomalous nodes exhibit higher signal strength in the spectral domain, while the majority of normal nodes have low-frequency signals [44], [45]. This suggests that the majority of nodes share a similar feature representation, while only a few abnormal nodes possess distinct feature representation. Consequently, when employing mean pooling or sum pooling functions, the representations of these abnormal nodes will be overshadowed by a multitude of normal representations. Consequently, the resulting graph features obtained from such pooling operations will exhibit similarity and prove unsuitable for change point detection. Therefore, we apply the max pooling to realize the pooling operation as follows:

$$z_t = \mathcal{R}_{max}(\mathbf{Z}_t^i | i \in N). \tag{10}$$

3) GMM: GMM is a parameter probability density function, expressed as a weighted sum of Gaussian component densities. GMM describes data distribution by employing multiple Gaussian distributions, which is a widely used clustering algorithm both in academic research and industry due to its capability of representing a class of data distributions. Thus, we leverage GMM to divide the representation of all snapshots z_t into different clusters. The formalization of GMM for dynamic networks can be described as follows:

$$p(\mathbf{x} \mid \lambda) = \sum_{i=1}^{M} w_t^i g\left(\mathbf{x} \mid \boldsymbol{\mu}_t^{i,GMM}, \boldsymbol{\Sigma}_t^{i,GMM}\right), \tag{11}$$

where $\mathbf{x}$ is the input feature vectors of data, w_t^i is the mixture weights for snapshot G_t, $w_t^i \geq 0$ and $\sum_{i=1}^{M} w_t^i = 1$. $\boldsymbol{\mu}_t^{i,GMM}$ and $\boldsymbol{\Sigma}_t^{i,GMM}$ is the mean vector and covariance matrix separately of component i at snapshot G_t in the GMM, respectively. $\lambda_M = (\boldsymbol{\mu}_t^{i,GMM}, \boldsymbol{\Sigma}_t^{i,GMM})$, M is the number of Gaussian component. $g\left(\mathbf{x} \mid \boldsymbol{\mu}_t^{i,GMM}, \boldsymbol{\Sigma}_t^{i,GMM}\right)$ is the component Gaussian distribution density, each component is a D-variate Gaussian function.

$$\begin{aligned} & g\left(\mathbf{x} \mid \boldsymbol{\mu}_t^{i,GMM}, \boldsymbol{\Sigma}_t^{i,GMM}\right) \\ &= \frac{1}{(2\pi)^{D/2} \left|\Sigma_t^{i,GMM}\right|^{1/2}} \\ &\quad \times \exp\left\{-\frac{1}{2}\left(\mathbf{x}-\boldsymbol{\mu}_t^{i,GMM}\right)' (\boldsymbol{\Sigma}_t^{i,GMM})^{-1} \left(\mathbf{x}-\boldsymbol{\mu}_t^{i,GMM}\right)\right\}. \end{aligned} \tag{12}$$

Moreover, to estimate the parameters of the GMM, we employ the Expectation-Maximization algorithm [46]. The GMM yields M clusters, denoted as $K^i, i \in \{0, 1, \ldots, M-1\}$, where each cluster represents a set of snapshots exhibiting similar structural and attribute characteristics.

$$K^i = GMM(z), \quad i \in \{0, 1, \ldots, M-1\}, \tag{13}$$

where z denotes the set of all snapshot embeddings.

A GMM can be accurately represented by a series of parameters, including mean vectors $\boldsymbol{\mu}_t^{i,GMM}$, covariance matrix $\boldsymbol{\Sigma}_t^{i,GMM}$ and weights w_t^i of all components. We will update the prior distribution of VGAE with the parameters of GMM to achieve joint iterative training. This will be introduced in detail in the joint iterative training subsection.

4) Joint Iterative Training: VGAE employs a single Gaussian distribution prior to node embedding. However, the single Gaussian distribution cannot sufficiently describe the data distribution sometimes. Especially in the task of change point detection in dynamic networks, the interactions between nodes are more complex, and the topology structure undergoes sudden changes as well. A simple Gaussian prior cannot provide effective constraints to learn effective embedding in

this case. Therefore, We adopt joint iterative training to replace the single Gaussian distribution in VGAE with the Gaussian mixture distribution obtained by GMM. There are some existing studies that try to replace the Gaussian prior in the variational framework with the Gaussian mixture distribution. VaDE [47] combines the GMM with VAE to solve problems of image clustering. Hui et al. [48] focuses on node clustering by combining VGAE and GMM. Here we expand the idea to detect change points in the dynamic network through joint iterative training of VGAE and GMM.

With the help of joint iterative training, the generative process can be described as follows:

- First, we select a cluster c for G_t from the M clusters, employing the GMM to model the choice of c. This selection is governed by a categorical distribution $Cat(w_t)$, where w_t represents the mixture weights of the GMM.
- Each cluster c is characterized by a distribution parameterized by mean vectors $\boldsymbol{\mu}_t^{c,GMM}$ and covariance matrix $\boldsymbol{\Sigma}_t^{c,GMM}$, with the latter defined as $\boldsymbol{\Sigma}_t^{c,GMM} = (\boldsymbol{\sigma}_t^{c,GMM})^2 I$, where I is the identity matrix and $\boldsymbol{\sigma}_t^{c,GMM}$ represents the variance.
- Subsequently, we update the single Gaussian distribution of the data with the Gaussian mixture distribution $N \sim (\boldsymbol{\mu}_t^{c,GMM}, \boldsymbol{\Sigma}_t^{c,GMM})$. This updated distribution serves as the prior of VGAE in the next joint iteration training phase.
- Next, we infer the posterior distribution to extract node embeddings $\mathbf{Z}_t$, all under the constraints of this new prior.
- Finally, we employ an inner product decoder to generate the final generative sample.

The final variational lower bound of VGAE consists of reconstruction loss and the KL divergence between an approximate posterior distribution and Gaussian mixture distribution.

$$\mathcal{L}_{(2)} = \mathbb{E}_{q(\mathbf{Z}_t|\mathbf{X}_t,\mathbf{A}_t)}[p(\mathbf{A}_t|\mathbf{Z}_t)] - \mathrm{KL}[q(\mathbf{Z}_t|\mathbf{X}_t,\mathbf{A}_t)||p_m(\mathbf{Z}_t)]. \tag{14}$$

where $\mathbb{E}_{q(\mathbf{Z}_t|\mathbf{X}_t,\mathbf{A}_t)}[p(\mathbf{A}_t|\mathbf{Z}_t)]$ is the graph reconstruction loss, $q(\mathbf{Z}_t|\mathbf{X}_t,\mathbf{A}_t)$ is the approximate distribution, $p_m(\mathbf{Z}_t)$ is the Mixture-of-Gaussians prior obtained by GMM. Since the Mixture-of-Gaussians prior more accurately describes the characteristics of dynamic networks, the snapshot embedding obtained under this distribution constraint can capture more network structure information. Algorithm 1 describes the details of our model training.

5) Change Point Detection: In our clustering process through GMM, there's a potential issue where snapshots may be grouped into the same cluster even if they aren't temporally adjacent. This clustering behavior isn't suitable for identifying potential change points in dynamic networks. To address this concern, we incorporate temporal dependence information into the clustering results to identify potential change points. Specifically, we obtain the clusters through GMM, resulting in M clusters, each containing snapshots with identical cluster labels. To discover potential change points, we organize these snapshots based on their cluster labels and arrange them in temporal order. We iterate through the arranged snapshots and compare the cluster label of each snapshot with the current value of *tag*. If the cluster label of a snapshot at time t differs from the current value of *tag*, we flag the snapshot as a potential change point and update the value of *tag* with the cluster label of the current snapshot. This process continues until we have examined all snapshots, and at the end, we will identify all the change points. This procedure is further detailed in Algorithm 2, which provides a step-by-step guide on how we identify change points by incorporating temporal information into the clustering results.

TABLE II
CHARACTERS OF DATASETS, T IS THE NUMBEROF SNAPSHOTS, N_{max} IS THE MAXIMUM NUMBER OF NODES IN THE DYNAMIC NETWORK, E_{mean}, E_{max}, AND E_{min} ARE THE MEAN, MAXIMUM, AND MINIMUM NUMBER OF EDGES IN THE DYNAMIC NETWORK, RESPECTIVELY

Datasets	N_{max}	E_{mean}	E_{max}	E_{min}	T
Mit Proximity Network	94	504	1,229	4	45
ENRON Email Network	147	103	306	2	44
World Trade Network	196	10,926	17,954	3,321	53
Synthetic Network-1000	1,000	53,889	74,921	34,705	16
Synthetic Network-5000	5,000	408,678	687,417	236,900	16

C. Complexity Analysis

The time complexity of our model primarily arises from VGAE. VGAE learns node representations through the reconstruction graph loss, and for snapshot G_t, its complexity is $\mathcal{O}(|V_t|^2)$, where $|V_t|$ is the number of nodes in G_t. To reduce complexity and accelerate training, we employ negative sample sampling to compute the reconstruction loss. This involves treating all edges within a snapshot as positive samples and sampling an equal number of edges as negative samples. The optimization is then performed by minimizing the cross-entropy loss. The complexity for G_t is $\mathcal{O}(|E_t|)$, where $|E_t|$ is the number of edge in G_t. For the dynamic network G with T snapshots, the overall complexity is $\mathcal{O}(T|E_t|)$.

IV. EXPERIMENTS

In this section, we first introduce our five datasets, including three real-world networks and two synthetic networks. Then we introduce all the compared methods with our model and what kind of metric we adopt to evaluate. At last, we show our results on five datasets and make further analysis.

A. Datasets and Baselines

1) Datasets: We conduct experiments using three real-world networks and two synthetic networks of varying scales to assess VGGM's change point detection capabilities. A detailed overview of the characteristics of these five datasets is provided in Table II. The ground truth information is provided in Fig. 2, Fig. 7, and Table III.

- **MIT Proximity Network** [49]: MIT Proximity Network is a dynamic network that describes the social interactions between students and staff in MIT Media Laboratory. This network is composed of 94 individuals (the number of nodes) interactions between August 31, 2000, and July 10, 2005. We divide the MIT Proximity Network by

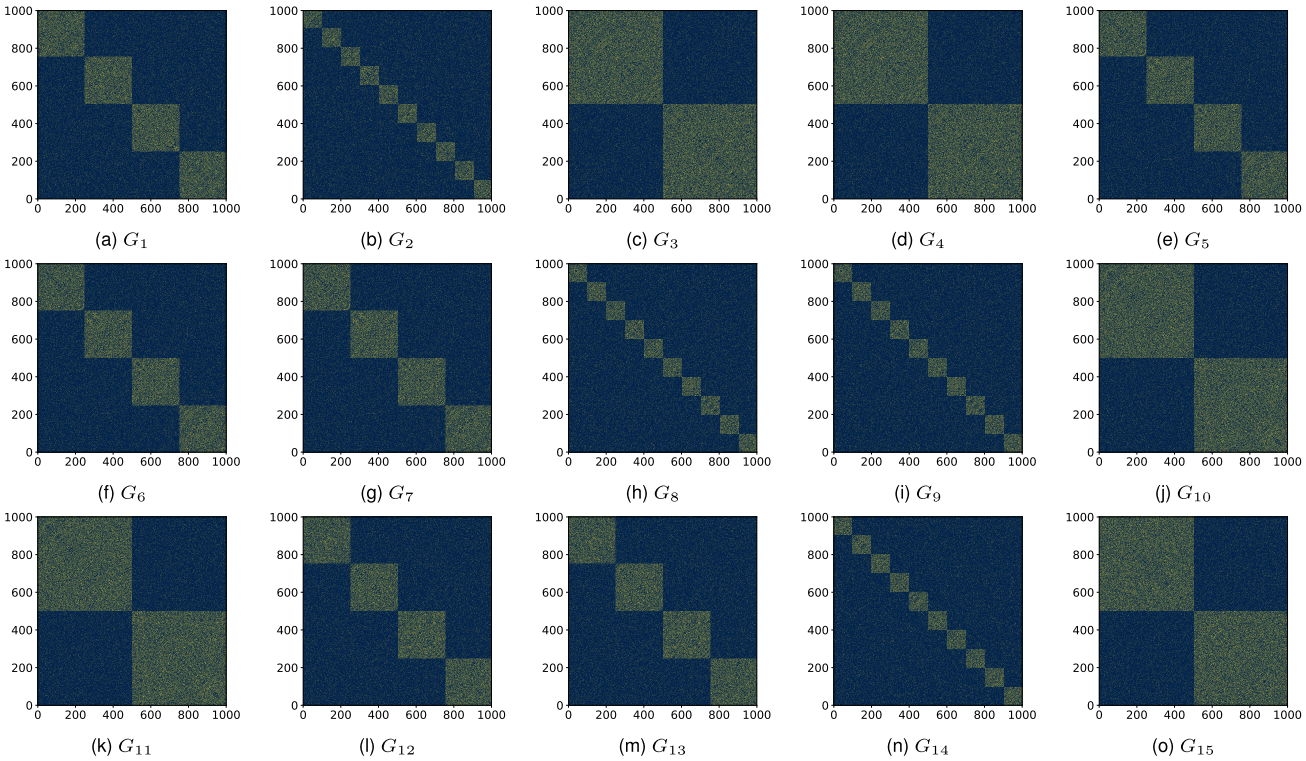


Fig. 2. Visualization of adjacency matrix in Synthetic Network-1000. The coordinate axis represents the indices of nodes, where an element is marked as 1 if there exists an edge between a pair of nodes, and 0 if there is no edge. In this dynamic network, we initialize G_0 with four communities (the structure of G_0 is the same as G_1). When a snapshot exhibits a significant difference in network structure compared to the snapshot of the previous moment, we label this snapshot as a change point. For example, G_2 has 10 communities while G_1 has 4 communities, G_2 is considered a change point. It can be observed that we introduce a total of eight change points into the dynamic network.

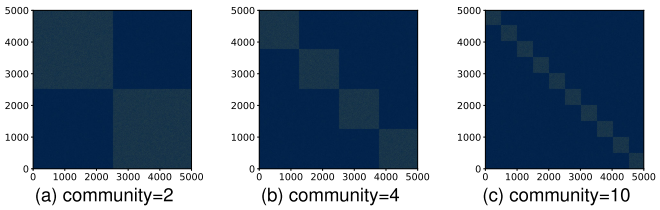


Fig. 3. Three network structures with different numbers of communities in Synthetic Network-5000.

one week, and finally, get a dynamic network composed of 45 snapshots from August 30, 2004, to July 10, 2005. Each snapshot is an undirected graph describing the relation between individuals. If two nodes interact with each other, we can establish an edge for them. The number of change points in the MIT Proximity Network is 20.

- **ENRON Email Network** [50]: Enron Email Network is built by email communications between different individuals of Enron energy company. We construct a dynamic network with 45 snapshots and 147 nodes (each node represents a senior employee of the company.), taking one month as a scale between November 1998 to June 2002. If two senior employees have direct email communication, we establish an edge between them. Each snapshot of the dynamic network describes the communication behavior of the company's senior employees in the corresponding month. We identify 18 incidents that affect the Enron Mail Network structure as the change points.
- **World Trade Network** [51]: World Trade Network consists of trade records which include annual import and export amounts from 196 countries between 1948 to 2000. Therefore, we use one year as the time scale to construct a dynamic network, where each edge shows that two countries conducted trade. The World Trade Network has 53 snapshots and 196 nodes. On average, each snapshot has 10926 edges. In our work, 21 points that have a significant influence on the network structure are labeled as change points.
- **Synthetic Network** [52]: We synthesize two dynamic networks with different scales by the Stochastic Block Model (SBM). SBM is a probabilistic generative model describing the structure of a graph. It partitions the graph into several communities by maintaining different connection probabilities for nodes within the same community and nodes in different communities. It can be used for community detection, graph clustering, and other tasks. Both synthetic networks consist of 16 snapshots. For Synthetic Network-1000, the probability of edges within the same community is 0.25, while the probability of connecting nodes between different communities is 0.05. For Synthetic Network-5000, the within-community connection probability is 0.1, and the probability of connecting nodes between communities is 0.01. In our setting, the intra-community and inter-community connection probabilities remain constant. We only vary the number of communities to introduce change points in the synthetic networks. We show the network structure

TABLE III
EVENTS OF ENRON EMAIL NETWORK AND WORLD TRADE NETWORK

Data Name	Time	Events
Enron Email	1999.11	Lunch of Enron online
	2000.01	Launch of EBS
	2000.07	EBS-blockbusters partnership
	2000.08	Stock all-time high
	2000.10	Enron attorney discusses Belden's strategies
	2000.11	FERC exonerates Enron
	2000.12	EBS $ 53m profit
	2001.01	Blackouts in CA
	2001.03	Enron schedules conference call to boost stock
	2001.04	Conference calls with arguments
	2001.05	Schwarzeneagger.Lay. Milken Meeting
	2001.07	Quarterly conference call
	2001.08	Watkins raises accounting irregularities
	2001.11	Dynegy agrees to buy Enron
	2001.12	Enron files for bankrupcy
	2002.01	Stephen Cooper takes over as Enron CEO
	2002.02	Lay cancels Senate committee appearance
	2002.03	Arthur Anderen LLP indicted
World Trade	1950	The third round of the GATT round was held in Turkey
	1955	The next round of trade was completed in May 1956
	1957	Establishment of the European Community
	1959	Establishment of the International Maritime Organization
	1960	African independence year
	1964	The Kennedy Round, named in honor of the late President of the United States, achieved tariff reductions in world trade worth $40 billion.
	1967	Establishment of the European Community
	1968	Two important ISO recommendations for global standardization and containerization
	1971	The collapse of the Bretton Woods system
	1973	The first oil crisis broke out
	1980	The United States grants China most-favoured-nation treatment
	1981	Japan became the first country in Asia to have a large trade surplus
	1983	With the gradual thawing of Sino-Soviet relations, the border trade between China and Central Asia resumed
	1987	Economic crisis in the United States
	1989	Establishment of the Asia-Pacific Economic Cooperation Organization
	1991	Collapse of the Soviet Union
	1993	Establishment of the European Community
	1994	The North American Free Trade Area was officially established
	1995	The WTO was officially established
	1997	Southeast Asia breaks out economic crisis
	1999	Birth of Euro

and change points of Synthetic Network-1000 in Fig. 2. Eight change points largely affect the network structure in 16 snapshots in Synthetic Network-1000. The evolution process of Synthetic Network-5000 is the same as Synthetic Network-1000. Fig. 3 shows the network structure of Synthetic Network-5000. The change points are identical in both synthetic networks.

2) Baselines: To ascertain the effectiveness of our model, we conduct a comprehensive performance comparison with seven advanced methods in change point detection, utilizing five distinct dynamic network datasets as our evaluation benchmarks. These selected approaches served as our baselines for the comparative analysis.

- SCOUT [53]: This method first discovers potential change points through a search and then performs clustering on the same segment in the dynamic network.
- GHRG [16]: It combines a generalized hierarchical random graph model with a Bayesian hypothesis test to estimate model parameters and calculates the deviation score of each snapshot whether it is a change point.
- CICPD [15]: This method is a change point detection method based on community detection. It extracts features for each snapshot with PageRank and finds change points with spectral clustering.
- LAD [17]: This method obtains the low-dimension graph representation of each snapshot by computing the Singular Value Decomposition of graph Laplacian. and distinguishes change points by computing the anomalous score.
- MultiLAD [54]: MultiLAD is a method based on LAD and extends LAD to the multi-view setting. It identifies the most informative singular values from each view.
- Netwalk [23]: It learns vector representations based on the walk and detects change points based on a dynamic clustering algorithm. Change points are detected by learning graph representations which can be updated steadily as networks evolve.
- MICPD [55]: It is a multi-view feature interpretable framework, which encodes high-dimensional data information into a low-dimensional representation by a vector autoregressive model, and discovers change points with interactions of multiple entities across time.
- GEABS [40] This method introduces a generation model, which takes into account the interaction of anomalies at different levels (node, community, and network) to capture all types of anomalies of dynamic networks.
- s-GNN [35] It adopts a siamese graph neural network to learn the graph similarity of the dynamic network. This method can online detect change points and adapt to the specific network domain and localize changes.

3) Evaluation Metric: The change point detection problem can also be viewed as a binary classification problem. Thus, we employ three widely used indicators in classification tasks, namely, $Precision$, $Recall$ and F_1, as our evaluation metrics. which can be computed as follows:

- **Precision**: It indicates the proportion of the exact number of searches in the predictions.

$$Precision = \frac{TP}{TP + FP}, \tag{15}$$

TABLE IV
PRECISION, RECALL, F1 OF OUR METHOD AND OTHER BASELINES

Method	MIT			Enron			World Trade			Synthetic-1000			Synthetic-5000		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
SCOUT	0.2667	0.2000	0.2286	0.2727	0.1500	0.1935	0.4000	0.1818	0.2500	0.4286	0.3000	0.3529	0.3333	0.2500	0.2857
GHRG	0.4194	0.6500	0.5098	0.4284	0.1667	0.2400	0.3500	0.6364	0.4516	0.2857	0.2000	0.2353	0.4011	0.2410	0.3041
LAD	0.5000	0.5500	0.5238	0.4500	0.5000	0.4737	0.4231	0.5000	0.4583	0.5000	0.5000	0.5000	0.3750	0.3750	0.3750
MutiLAD	0.5200	0.6500	0.5778	0.4545	0.5556	0.5000	0.4194	0.5909	0.4906	0.5000	0.6250	0.5556	0.4166	0.6250	0.5000
CICPD	0.5161	0.8000	0.6275	0.5217	0.6667	0.5854	**0.6682**	0.3182	0.4309	0.3818	0.5000	0.4628	0.4444	0.5000	0.4706
Netwalk	0.5191	0.6253	0.5669	**0.5875**	0.5250	0.5302	0.4444	0.6657	0.5333	0.5963	0.7750	0.6738	0.5307	0.7515	0.6191
MICPD	0.4853	0.6111	0.5238	0.5328	0.6944	0.6024	0.5217	0.6667	0.5854	0.5603	0.7778	0.6412	0.5210	0.7813	0.6246
GEABS	0.5166	0.7000	0.5927	0.5069	0.8333	0.6243	0.5192	0.7112	0.5989	0.6065	0.8333	0.7071	—	—	—
s-GNN	0.4662	0.7833	0.5776	0.5317	0.6944	0.6022	0.4870	0.6136	0.5384	**0.6843**	0.7916	0.7309	0.6273	0.8333	0.7115
VGGM	**0.5414**	**0.9444**	**0.6826**	0.5360	**0.8636**	**0.6556**	0.5235	**0.9215**	**0.6648**	0.6568	**0.9625**	**0.7773**	**0.6454**	**0.9250**	**0.7523**

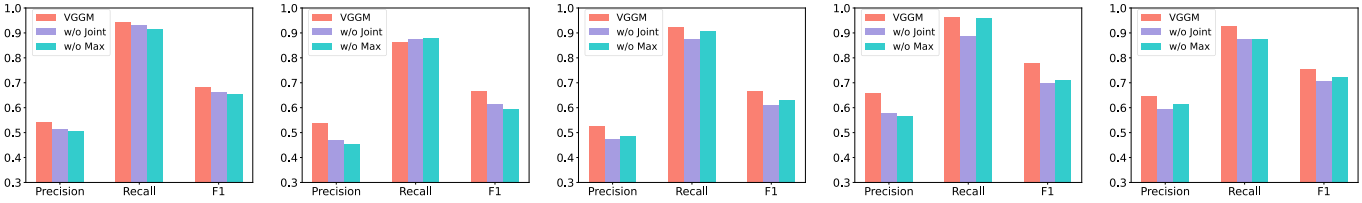


Fig. 4. The results of ablation experiment on five datasets. The performance of VGGM and other variants on the Mit Proximity Network, Enron Email Network, World Trade Network, and two Synthetic Networks are listed from left to right. We conduct all variants of VGGM on each dataset.

where TP is the number of positive class predictions that belong to ground truth, and FP represents that model classifies true negative as positive.

- **Recall**: It indicates the proportion of positive instance predictions by the model in the whole dataset.

$$Recall = \frac{TP}{TP + FN}, \tag{16}$$

where FN represents the number of data instances that the model considers to be negative but positive.

- **F1-score**: We employ the F1-score as the final performance metric for evaluation.

$$F_1 = \frac{2 * TP}{2 * TP + FN + FP}, \tag{17}$$

where F_1 is a comprehensive indicator, representing the harmonic mean of both $Precision$ value and $recall$ value.

B. Experiment Settings and Results

1) Implement Details: We conduct our method and all baselines in an environment featuring Ubuntu 22.04.1 LTS, an Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz, and a GeForce GTX 3090 Graphics Card. The software versions utilized included CUDA 11.3 and PyTorch 1.12.0. To represent the feature matrix for each dynamic network, we employed one-hot encoding across all datasets. We conduct all experiments with a fixed number of epochs and iterations for each dataset. For constructing graph neural networks and handling mini-batch data, we utilized the PyTorch Geometric Library version 2.3.0. In addition, all methods are conducted 10 times, and the final results are obtained as the mean values. Our model is trained using the Adam optimizer with a learning rate of 0.01, and a weight decay of 1e-5 is applied. The embedding dimension for snapshot representation is set to 8, while the GNN in VGAE featured 32 and 16 channels. For training GMM, we utilize a diagonal covariance matrix and determined the number of components based on the network structure of each dataset, with values of 2, 3, and 4 being chosen accordingly. For instance, we set the number of components to 3 for two synthetic networks.

2) Experiment Results: We show the change point detection results of our method and other compared methods in Table IV. The best performance is highlighted in black, and the second-best performance is underlined. Due to the high time complexity of GEABS, it results in an "out of time" issue on the Synthetic-5000 dataset with millions of edges. Therefore, our experimental results don't report the performance of GEABS on this dataset. According to the experimental results, we can draw the following conclusions:

- It can be seen that our model VGGM outperforms other methods in most cases, in terms of measures $Recall$ and F_1, which implies that our model can discover more change points in a dynamic network. We can find that the $recall$ rate is close to 1 in five datasets, which shows that almost all the change points in the dynamic network can be found.
- We also observe that VGGM performs exceptionally well on two synthetic networks, showcasing significantly higher precision than other methods. However, its performance on three real-world datasets is not notably remarkable. This could be attributed to the possibility of incomplete recording of certain events in real-world datasets, leading to imperfect ground truth labels. Consequently, false negatives may not necessarily indicate true model inaccuracies. In the case of synthetic datasets, where the ground truth is entirely accurate, the model exhibits higher precision compared to the real-world datasets.

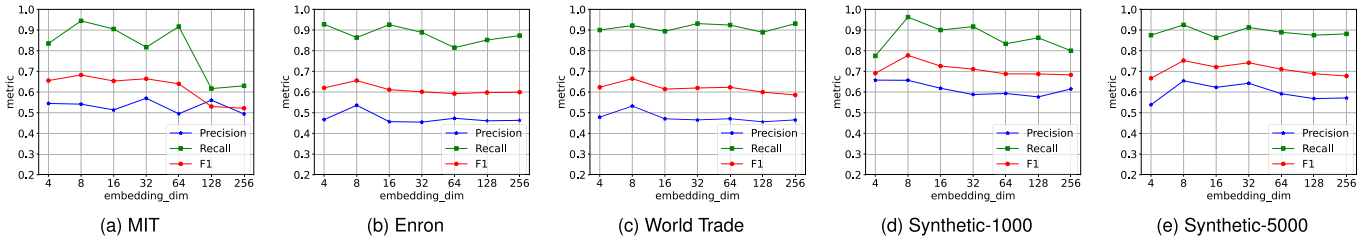


Fig. 5. The performance of our model with different graph embedding dimensions on five datasets. It reflects the robustness of VGGM and the effect of embedding dimensions as graph embedding dimensions changed. The F_1 curve is generally stable on each dataset when the embedding dimensions are 4 to 256.

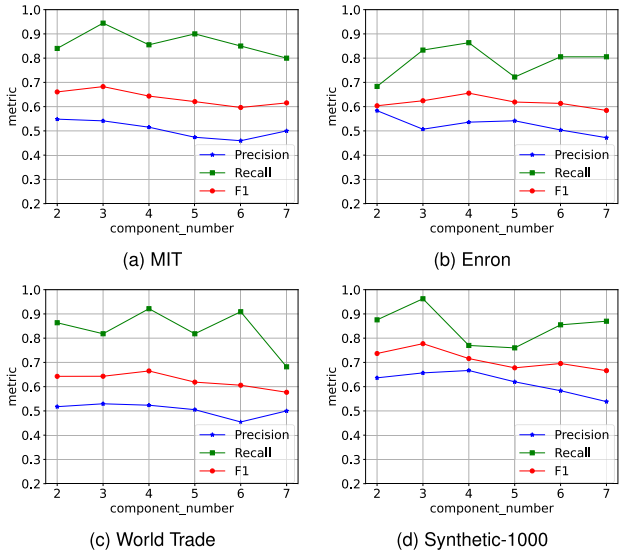


Fig. 6. The hyper-parameter sensitivity experiments on the number of components of GMM.

- Lastly, our evaluation reveals interesting variations in the performance of different methods across datasets. CICPD demonstrates strong performance on the MIT and Enron datasets but struggles on synthetic datasets. LAD and MutilLAD exhibit similar performance on the first four datasets but perform poorly on larger-scale datasets. Netwalk displays relatively good performance on synthetic datasets but yields mixed outcomes on real-world datasets. In contrast, MICPD doesn't perform well on the MIT dataset. GEABS exhibits better performance on three small-scale networks compared to synthetic networks. s-GNN performs well on two synthetic networks but does not perform well on three real-world networks. In comparison, VGGM consistently delivers excellent performance across all datasets, demonstrating its versatility in detecting change points within dynamic networks of varying scales, including larger-scale networks, while maintaining excellent overall performance. This suggests that VGGM is more capable of generalizing dynamic networks of different scales and characteristics and mining the evolution patterns of the dynamic network.

Based on the aforementioned experimental results, we have confidence in the effectiveness of our model. By jointly iteratively optimizing the GMM and VGAE, we approximate the Mixture-of-Gaussians prior, which we believe provides a more accurate characterization of the dynamic networks compared to the distribution assumed in VGAE. Consequently, the embeddings of each node capture more comprehensive and precise network information. Additionally, the incorporation of the max-pooling mechanism successfully preserves the anomaly information across the entire network snapshot. We attribute the effectiveness of our model to these factors.

C. Ablation Experiments

To verify the effectiveness of the different parts in VGGM, we conduct a set of ablation studies.

- **w/o Joint**. In order to investigate the impact of joint iterative training, we conduct experiments using a variant referred to as "w/o Joint", where VGAE and GMM are trained separately. In this variant, VGAE is employed with a single Gaussian prior to obtaining node embeddings.
- **w/o Max**. To examine the significance of different pooling methods, we train another variant referred to as "w/o Max". In this variant, we replace the max pooling operation with mean pooling to obtain snapshot embeddings.

All presented results are depicted in Fig. 4, from which we can derive the following conclusions:

- In all five datasets, the performance metrics measured by F_1 experience degradation when the joint iterative training mechanism is removed. This deterioration is particularly pronounced in the World Trade and Synthetic datasets. This observation underscores the positive impact of joint iterative training, which enhances performance by enabling the Gaussian mixture distribution to provide a more precise representation of the data. This effect is especially significant in the context of large-scale dynamic networks.
- Replacing the max pooling operation with mean pooling as the readout function results in underperformance compared to the original model. This alternative approach approximately matches the performance of the model without the joint iterative training mechanism across the five datasets. This finding suggests that the selection of pooling functions can significantly influence the model's detection capabilities. In particular, max pooling has proven to be effective in capturing abnormal information within dynamic networks.

D. Hyper-Parameter Sensitivity

To explore the effect of hyper-parameter, we conduct a series of hyper-parameter sensitivity experiments. Given the limited number of hyper-parameters in our model, we specifically focused on investigating the impact of varying the

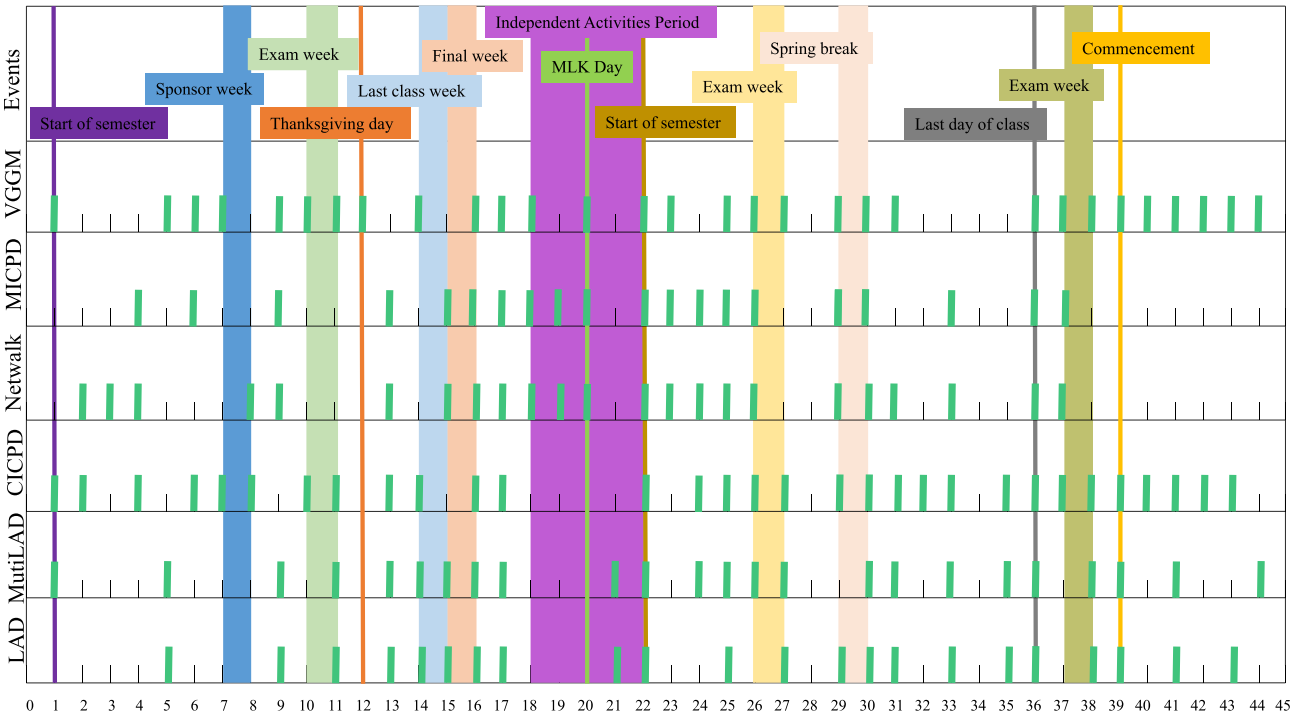


Fig. 7. A case study of change point detection within the MIT proximity network. Real change points are marked with colored vertical bars that span the entire height of the figure. The colored vertical bars represent the occurrence of continuous events, where the start and end of the events are also considered change points. Detected change points are marked using small green vertical bars.

embedding dimensions of the graph representation vector and the component number of GMM. We systematically vary the embedding dimensions to 4, 8, 16, 32, 64, 128, and 256, monitoring the resulting changes in performance across five datasets. The results are presented in Fig. 5. Additionally, We set the component number of GMM to 2, 3, 4, 5, 6, and 7 to observe its impact on the model's performance across four datasets (The impact of the number of GMM components is consistent for the two synthetic datasets.). Fig. 6 shows the results.

1) Embedding Dimension: Notably, for the MIT Proximity Network dataset, the optimal performance is achieved when the embedding dimension is set to 8. Furthermore, high performance is maintained when the embedding dimension is chosen as 32. Even in the case of the highest dimensionality (256), VGGM still manages to maintain an F1 score exceeding 0.5. Interestingly, for the remaining datasets, the best performance consistently corresponds to an embedding dimension of 8. The performance remains relatively stable when different embedding dimensions are employed. This valuable observation suggests that our model exhibits a high degree of resilience to variations in hyperparameters, highlighting its robustness in practical applications.

2) GMM Component: It can be observed that all datasets achieve the best performance when the number of components of GMM is set to 3 or 4. Relatively good performance is maintained when the number is 2 as well. As the number of components increases, the performance tends to decrease. The optimal number of components for each dataset is related to its intrinsic structural characteristics. For example, in the Synthetic-1000 dataset, which has three different network structures (community=2, 4, 10), the best performance is achieved when the number of components is set to 3.

E. Case Study

To further demonstrate the change point detection capability of our model, we conducted an in-depth case study using the MIT Proximity Network dataset. The objective is to highlight the specific points successfully detected by our model, which eluded detection by other methods. The results, as illustrated in Fig. 7, illuminate several noteworthy observations:

- Our model exhibits an exceptional recall rate, missing only two points in the dataset, contributing to its high overall recall performance.
- Points labeled as 1, 7, 10, 18, and 20, which remained undetected by most other methods, were accurately identified by our approach. This suggests that our model possesses superior sensitivity to changes in network structure, enabling it to excel in capturing change points within dynamic networks.
- It is noteworthy that our model's precision is slightly compromised due to its identification of the last five points as change points. In reality, the dataset for the MIT ACADEMIC CALENDAR 2004-2005 solely includes "SUMMER SESSION (incl. Exam Period)" to document the period from June 6 (Mon) to August 16 (Tues) [15]. Consequently, some events during this period aren't recorded. Therefore, our model identifying these points as change points cannot be considered incorrect.

This case study effectively demonstrates that our model is more sensitive to changes in network structure compared to other methods, showcasing its superior capabilities in capturing change points within dynamic networks.

V. Conclusion and Future Work

In this paper, we present a novel unsupervised model named VGGM for change point detection in dynamic networks. VGGM achieves seamless end-to-end training by means of a joint iterative training process involving VGAE and GMM, thus addressing the issue of staged training. Meanwhile, under the constraint of joint iterative training, VGAE is capable of acquiring more effective snapshot embeddings, while VGGM exhibits the ability to accurately find change points without additional parameters. Experimental results show that our model not only works well on various real-world networks but also can deal with larger-scale synthetic networks. We conduct ablation experiments to verify the importance of different parts in our model. Additionally, we demonstrate the resilience of our model through a hyper-parameter sensitivity experiment. Finally, we conduct a case study to thoroughly explore the detection capabilities of VGGM. Our paper has some limitations as we only study dynamic networks with snapshots at the same scale. However, dynamic networks with snapshots at different scales are more challenging and crucial for real systems. We plan to address this issue in future work.

References

[1] J. C. Mitchell, "Social networks," *Annu. Rev. Anthropol.*, vol. 3, no. 1, pp. 279–299, 1974.

[2] M. E. J. Newman and J. Park, "Why social networks are different from other types of networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 68, no. 3, Sep. 2003, Art. no. 036122.

[3] R. Silva-Rocha and V. de Lorenzo, "Mining logic gates in prokaryotic transcriptional regulation networks," *FEBS Lett.*, vol. 582, no. 8, pp. 1237–1244, Apr. 2008.

[4] J. Hemer, "A snapshot on crowdfunding," Arbeitspapiere Unternehmen und Region, Fraunhofer-Institut für Syst. Innovationsforschung ISI, Karlsruhe, Germany, Tech. Rep. R2/2011, 2011.

[5] R. C. Y. Cheung, A. Aue, S. Hwang, and T. C. M. Lee, "Simultaneous detection of multiple change points and community structures in time series of networks," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 580–591, 2020.

[6] M. Yang, M. Zhou, M. Kalander, Z. Huang, and I. King, "Discrete-time temporal network embedding via implicit hierarchical learning in hyperbolic space," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 1975–1985.

[7] C. D. T. Barros, M. R. F. Mendonça, A. B. Vieira, and A. Ziviani, "A survey on embedding dynamic graphs," *ACM Comput. Surv.*, vol. 55, no. 1, pp. 1–37, Jan. 2023.

[8] G. Duan, H. Lv, H. Wang, and G. Feng, "Application of a dynamic line graph neural network for intrusion detection with semisupervised learning," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 699–714, 2023.

[9] Z. Yuan, M. Shao, and Q. Yan, "Motif-level anomaly detection in dynamic graphs," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 2870–2882, 2023.

[10] L. Kendrick, K. Musial, and B. Gabrys, "Change point detection in social networks—Critical review with experiments," *Comput. Sci. Rev.*, vol. 29, pp. 1–13, Aug. 2018.

[11] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, Feb. 2021, pp. 4027–4035.

[12] C. Wang and H. Zhu, "Wrongdoing monitor: A graph-based behavioral anomaly detection in cyber security," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 2703–2718, 2022.

[13] M. Ahmed, A. N. Mahmood, and M. R. Islam, "A survey of anomaly detection techniques in financial domain," *Future Gener. Comput. Syst.*, vol. 55, pp. 278–288, Feb. 2016.

[14] C. Baur, B. Wiestler, S. Albarqouni, and N. Navab, "Deep autoencoding models for unsupervised anomaly segmentation in brain MR images," in *Proc. Int. MICCAI Brainlesion Workshop*, Granada, Spain. Cham, Switzerland: Springer, 2018, pp. 161–169.

[15] T. Zhu, P. Li, L. Yu, K. Chen, and Y. Chen, "Change point detection in dynamic networks based on community identification," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 3, pp. 2067–2077, Jul. 2020.

[16] L. Peel and A. Clauset, "Detecting change points in the large-scale structure of evolving networks," in *Proc. AAAI Conf. Artif. Intell.*, 2015, vol. 29, no. 1, pp. 2914–2920.

[17] S. Huang, Y. Hitti, G. Rabusseau, and R. Rabbany, "Laplacian change point detection for dynamic graphs," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 349–358.

[18] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Stanford InfoLab, Stanford, CA, USA, Tech. Rep. 1999-66, 1999.

[19] T. Liu, C. Zhang, K.-M. Lam, and J. Kong, "Decouple and resolve: Transformer-based models for online anomaly detection from weakly labeled videos," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 15–28, 2023.

[20] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, 2021.

[21] O. Theodosiadou et al., "Change point detection in terrorism-related online content using deep learning derived indicators," *Information*, vol. 12, no. 7, p. 274, Jul. 2021.

[22] S. Pouyanfar et al., "A survey on deep learning: Algorithms, techniques, and applications," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–36, Sep. 2018.

[23] W. Yu, W. Cheng, C. C. Aggarwal, K. Zhang, H. Chen, and W. Wang, "NetWalk: A flexible deep embedding approach for anomaly detection in dynamic networks," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2672–2681.

[24] S. Deldari, D. V. Smith, H. Xue, and F. D. Salim, "Time series change point detection with self-supervised contrastive predictive coding," in *Proc. Web Conf.*, Apr. 2021, pp. 3124–3135.

[25] R. Zhang, Y. Hao, D. Yu, W.-C. Chang, G. Lai, and Y. Yang, "Correlation-aware unsupervised change-point detection via graph neural networks," 2020, *arXiv:2004.11934*.

[26] M. Li, L. Zhang, L. Cui, L. Bai, Z. Li, and X. Wu, "BLoG: Bootstrapped graph representation learning with local and global regularization for recommendation," *Pattern Recognit.*, vol. 144, Dec. 2023, Art. no. 109874.

[27] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *Proc. 7th Int. Conf. Learn. Represent.*, New Orleans, LA, USA, 2019, pp. 1–17.

[28] Y. Zhou, H. Zheng, X. Huang, S. Hao, D. Li, and J. Zhao, "Graph neural networks: Taxonomy, advances, and trends," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 1, pp. 1–54, Feb. 2022.

[29] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Mar. 2020.

[30] L. Bai, L. Cui, Y. Wang, M. Li, and E. R. Hancock, "HAQJSK: Hierarchical-aligned quantum Jensen–Shannon kernels for graph classification," 2022, *arXiv:2211.02904*.

[31] L. Cai et al., "Structural temporal graph neural networks for anomaly detection in dynamic graphs," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 3747–3756.

[32] Y. Han et al., "Dynamic neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7436–7456, Oct. 2021.

[33] G. Xue, M. Zhong, J. Li, J. Chen, C. Zhai, and R. Kong, "Dynamic network embedding survey," *Neurocomputing*, vol. 472, pp. 212–223, Feb. 2022.

[34] H. Chen, P. Jiao, H. Tang, and H. Wu, "Temporal graph representation learning with adaptive augmentation contrastive," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Cham, Switzerland: Springer, 2023, pp. 683–699.

[35] D. Sulem, H. Kenlay, M. Cucuringu, and X. Dong, "Graph similarity learning for change-point detection in dynamic networks," *Mach. Learn.*, vol. 113, no. 1, pp. 1–44, Jan. 2024.

[36] T. N. Kipf and M. Welling, "Variational graph auto-encoders," 2016, *arXiv:1611.07308*.

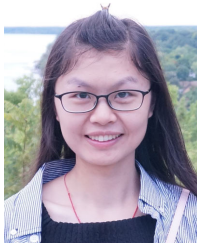
[37] D. A. Reynolds et al., "Gaussian mixture models," *Encyclopedia Biometrics*, vol. 741, nos. 659–663, 2009.

[38] Y. Wang, A. Chakrabarti, D. Sivakoff, and S. Parthasarathy, "Fast change point detection on dynamic social networks," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2992–2998.

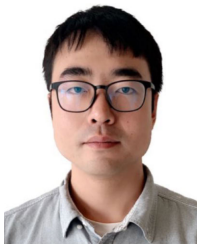
[39] H. Miller and O. Mokryn, "Size agnostic change point detection framework for evolving networks," *PLoS ONE*, vol. 15, no. 4, Apr. 2020, Art. no. e0231035.
[40] P. Jiao et al., "Generative evolutionary anomaly detection in dynamic networks," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12234–12248, 2023.
[41] D. Koutra, N. Shah, J. T. Vogelstein, B. Gallagher, and C. Faloutsos, "DeltaCon principled massive-graph similarity function with attribution," *ACM Trans. Knowl. Discovery Data*, vol. 10, no. 3, pp. 1–43, Feb. 2016.
[42] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent.*, Toulon, France, 2017, pp. 1–14.
[43] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *Stat*, vol. 1050, p. 1, Jan. 2014.
[44] Z. Chai et al., "Can abnormality be detected by graph neural networks?" in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Vienna, Austria, Jul. 2022, pp. 23–29.
[45] J. Tang, J. Li, Z. Gao, and J. Li, "Rethinking graph neural networks for anomaly detection," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 21076–21089.
[46] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, Nov. 1996.
[47] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: An unsupervised and generative approach to clustering," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1965–1972.
[48] B. Hui, P. Zhu, and Q. Hu, "Collaborative graph convolutional networks: Unsupervised learning meets semi-supervised learning," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 4215–4222.
[49] A. Pentland, N. Eagle, and D. Lazer, "Inferring social network structure using mobile phone data," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 36, pp. 15274–15278, 2009.
[50] G. J. Benston and A. L. Hartgraves, "Enron: What happened and what we can learn from it," *J. Accounting Public Policy*, vol. 21, no. 2, pp. 105–127, Jun. 2002.
[51] K. S. Gleditsch, "Expanded trade and GDP data," *J. Conflict Resolution*, vol. 46, no. 5, pp. 712–724, Oct. 2002.
[52] C. Lee and D. J. Wilkinson, "A review of stochastic block models and extensions for graph clustering," *Appl. Netw. Sci.*, vol. 4, no. 1, pp. 1–50, Dec. 2019.
[53] Y. Hulovatyy and T. Milenković, "SCOUT: Simultaneous time segmentation and community detection in dynamic networks," *Sci. Rep.*, vol. 6, no. 1, p. 37557, Nov. 2016.
[54] S. Huang, S. Coulombe, Y. Hitti, R. Rabbany, and G. Rabusseau, "Laplacian change point detection for single and multi-view dynamic graphs," 2023, *arXiv:2302.01204*.
[55] Y. Xie, W. Wang, M. Shao, T. Li, and Y. Yu, "Multi-view change point detection in dynamic networks," *Inf. Sci.*, vol. 629, pp. 344–357, Jun. 2023.

Xinxun Zhang received the bachelor's degree from Zhengzhou University, Zhengzhou, China, in 2021. He is currently pursuing the master's degree with the School of Cyberspace, Hangzhou Dianzi University, Hangzhou, China. His current research interests include complex network analysis and network embedding.

Pengfei Jiao (Member, IEEE) received the Ph.D. degree in computer science from Tianjin University, Tianjin, China, in 2018. From 2018 to 2021, he was a Lecturer with the Center of Biosafety Research and Strategy, Tianjin University. He is currently a Professor with the School of Cyberspace, Hangzhou Dianzi University, Hangzhou, China. His current research interests include complex network analysis and its applications.



Mengzhou Gao received the B.E. degree in mechanical design, manufacturing, and automation from Harbin Institute of Technology, China, in 2012, and the Ph.D. degree in control science and engineering from Zhejiang University, China, in 2017. Since 2017, she has been an Assistant Professor with the School of Cyberspace, Hangzhou Dianzi University, China. Her current research interests include cyber-physical systems security, secure control, and complex networks.



Tianpeng Li is currently pursuing the Ph.D. degree with the College of Intelligence and Computing, Tianjin University, China. His current research interests include complex network analysis, role discovery, network representation learning, and percolation models.

Yiming Wu received the B.Eng. degree in automation and the Ph.D. degree in control science and engineering from Zhejiang University of Technology, Hangzhou, China, in 2010 and 2016, respectively. He held a visiting position with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, from 2012 to 2014. Since July 2016, he has been with Hangzhou Dianzi University, Hangzhou, where he is currently an Associate Professor with the School of Cyberspace. His main research interests include multi-agent systems, security and privacy theory, iterative learning control, and applications in intelligent transportation systems and sensor networks.

Huaming Wu (Senior Member, IEEE) received the B.E. and M.S. degrees in electrical engineering from Harbin Institute of Technology, China, in 2009 and 2011, respectively, and the Ph.D. degree (Hons.) in computer science from Freie Universität Berlin, Germany, in 2015. He is currently a Professor with the Center for Applied Mathematics, Tianjin University, China. His research interests include wireless networks, mobile edge computing, the Internet of Things, and complex networks.

Zhidong Zhao (Member, IEEE) received the B.S. and M.S. degrees in mechanical engineering from Nanjing University of Science and Technology, Nanjing, China, in 1998 and 2001, respectively, and the Ph.D. degree in biomedical engineering from Zhejiang University, Hangzhou, China, in 2004. He is currently a Full Professor with Hangzhou Dianzi University, Hangzhou. His research interests include biomedical signal processing, wireless sensor networks, biometrics, and machine learning.