Neurocomputing 626 (2025) 129548

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A contrastive learning strategy for optimizing node non-alignment in dynamic community detection

Xiaohong Li *, Wanyao Shi, Qixuan Peng, Hongyan Ran

Northwest Normal University, College of Computer Science and Engineering, Lanzhou, 730070, China

ARTICLE INFO

Communicated by J. Yu

Keywords:
Dynamic community detection
Graph contrastive learning
Node non-alignment
Spectral graph theory

ABSTRACT

Dynamic community detection, which focuses on tracking local topological variation with time, is crucial for understanding the changing affiliations of nodes to communities in complex networks. Existing researches fell short of expectations primarily due to their heavy reliance on clustering methods or evolutionary algorithms. The emergence of graph contrastive learning offers us a novel perspective and inspiration, which performed well in recognizing pattern at both the node-node and node-graph levels. However, there are still the following limitations in practice: (i) conventional data augmentations may undermine task-relevant information by bring in invalid views or false positive samples, leading the model toward weak discriminative representations. (ii) the non-alignment of nodes caused by dynamic changes also limits the expressive ability of GCL. In this paper, we propose a **C**ontrastive **L**earning strategy for **O**ptimizing **N**ode non-alignment in **D**ynamic Community Detection (**CL-OND**). Initially, we confirm the viability of utilizing dynamic adjacent snapshots as monitoring signals through graph spectral experiments, which eliminates the dependence of contrastive learning on traditional data augmentations. Subsequently, we construct an end-to-end dynamic community detection model and introduce a non-aligned neighbor contrastive loss to capture temporal properties and inherent structure of evolutionary graphs by constructing positive and negative samples. Furthermore, extensive experimental results demonstrate that our approach consistently outperforms others in terms of performance.

1. Introduction

Community structure is one of the most common and fundamental topological properties of complex networks, characterized by dense intra-cluster connectivity and sparse inter-cluster connectivity [1]. The world we live in can be seen as a vast network composed of a series of communities. For example, communities in social networks represent groups of individuals with similar interests, while communities in biological networks correspond to genes with similar biological functions. Extensive research has been conducted on community detection across networks of various scales, and an increasing number of methods have been proposed to analyze community structures, understand network functions, and predict network behaviors. However, most prior research on community detection has been based on a stringent assumption: that the real world can be represented by static networks, *i.e.*, mathematical objects "frozen" in time [2]. Unfortunately, this simplified scenario rarely aligns with the dynamic nature of the real world. Dynamic community detection goal aims to accurately identify the dynamic community at each time point, describe the life cycle and evolution of communities, and reveal patterns of formation, dissolution, merging, and splitting [3], as shown in Fig. 1. Therefore, the research of dynamic community detection technology has important theoretical significance and wide application prospects, but it also faces great challenges.

Currently, most of the solutions for dynamic community detection are still rooted in traditional approaches such as stochastic block models, multi-objective evolutionary algorithms and genetic algorithms [4]. These methods are often constrained by computational complexity and struggle to handle large-scale networks. In addition, while some frameworks utilize deep learning techniques, most simply produce basic clustering results in downstream tasks through node embeddings or graph embeddings without analyzing the evolution of communities over time. Compared to solutions using traditional methods, deep learning models excel at capturing intricate patterns and dependencies between nodes and communities. Recent advancements in Graph Contrastive Learning (GCL) have demonstrated significant potential in the field of dynamic network research [5]. Most existing GCL methods mainly explore its application on static graphs from the two perspectives of graph augmentations and contrastive objectives.

* Corresponding author.
E-mail addresses: xiaohongli@nwnu.edu.cn (X. Li), 2023222205@nwnu.edu.cn (W. Shi), 2021222181@nwnu.edu.cn (Q. Peng), hongyran@nwnu.edu.cn (H. Ran).

https://doi.org/10.1016/j.neucom.2025.129548
Received 25 July 2024; Received in revised form 23 December 2024; Accepted 25 January 2025
Available online 4 February 2025
0925-2312/

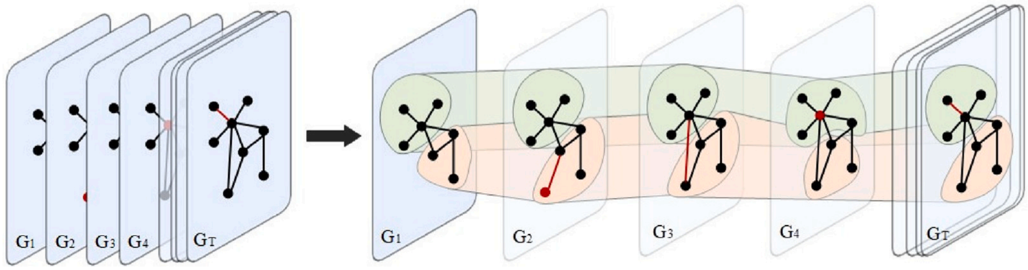


Fig. 1. The General Process of Dynamic Community Detection.

Firstly, as the key to the self-supervised learning paradigm, graph augmentation can help models mine deeper semantic information about graphs. Commonly employed graph augmentations include node dropping [6], edge perturbation, attribute masking [7], subgraph [8], and graph diffusion [9]. Specifically, node dropping simulates node absence by randomly removing nodes, while edge perturbation simulates uncertainty by randomly modifying edge connections [6]. Both techniques aim to enable the model to accurately identify graph features in the presence of missing information or changing connectivity. On the other hand, attribute masking simulates information missing by covering the attribute information of nodes or edges, thereby enhancing the model's robustness [7]. Subgraph sampling involves sampling smaller subgraphs from the original graph to increase data diversity [8]. Graph diffusion simulates information propagation, generating diverse topological views and enhancing the model's understanding and generalization capabilities of graph structures [9]. In general, these methods collectively contribute to improve the performance of graph contrastive learning models. To achieve the best model performance, GCL models need to manually select the augmentations for each dataset by trial and error, or introduce expensive domain-specific knowledge as a guide [10]. However, this is not a permanent solution. Semantics and labels may change during augmentation, leading to significant performance degradation on downstream tasks. For instance, the addition of an edge may introduce noise, while the removal of an edge could result in the loss of a node's most crucial edges and neighbors. Furthermore, current methods fail to fully utilize temporal information when constructing contrastive pairs, thereby limiting the effectiveness of dynamic graphs. Hence, the first challenge is to choose an efficient graph augmentation method that retains semantic information while still making full use of temporal information.

Secondly, the loss function is a critical component in contrastive learning, as it facilitates the acquisition of valuable representations by maximizing similarity between similar instances and minimizing similarity between dissimilar instances. Commonly employed contrastive losses include InfoNCE, normalized temperature-scaled cross-entropy (NT-Xent), Jensen–Shannon Divergence (JSD), and Triplet loss [11]. However, it is important to note that these contrastive losses are originally derived from the computer vision (CV) and natural language processing (NLP) domains, and are subsequently applied to graph data. This is rude and ignores the essential difference between the two data types [12]. Additionally, the graph homogeneity assumption posits that, within a graph structure, adjacent nodes have similar attributes, functions, or roles to some extent, implying an intrinsic relationship and consistency between them. However, InfoNCE and NT-Xent violate this assumption when handling node-level contrastive tasks. Specifically, in their mechanism for constructing contrastive pairs, the positive samples focus on the similarity between different augmented views of a node, while neglecting the inherent similarity that should exist between nodes and their neighbors, based on the graph's structure and attributes. In terms of negative samples processing, they simply regard anchor nodes and other unrelated nodes as negative sample pairs, without distinguishing potential homogeneity between these nodes and anchor nodes. Particularly in node-level tasks such as community detection, previous work on node-node graph contrastive learning has been implemented in aligned scenarios, which may hamper the flexibility and variability of sampled views and limit the expressive power of contrastive learning [13]. For example, in dynamic networks, nodes may appear/disappear over time, which makes node-node contrasting in non-aligned scenarios particularly important [14]. Therefore, the second challenge is to design a suitable non-aligned neighbor contrastive loss that maintains the homogeneity rules of the graph while still handling dynamically changing node information.

To tackle the challenges discussed above, we propose a contrastive learning strategy for optimizing node non-alignment in dynamic community detection (**CL-OND**) to explore the process and outcome of node and community evolution over time in a concise, fast, and efficient manner. Firstly, we have conducted numerous experiments to demonstrate that the natural variation between adjacent snapshots of dynamic networks conforms to the general graph augmentation rules of contrastive learning from the graph spectral perspective. Hence, we feed data from adjacent snapshots into the encoder as different data augmentation forms in the contrastive learning strategy. Secondly, we address the issue of node non-alignment in the temporal contrastive scenario by constructing the non-aligned neighbor contrastive loss functions, which facilitates the model to acquire a more precise community structure, thereby enabling a profound exploration of the community's evolution over time.

- We propose a contrastive learning strategy, and our graph spectral experimental exploration confirms the feasibility of utilizing adjacent snapshots of the dynamic network as a monitoring signal for contrastive learning. It can avoid the deviation of node semantics caused by traditional augmentation methods and obtain more accurate node representation.
- We propose an end-to-end dynamic community detection model and introduce a neighbor contrastive loss for node non-aligned scenarios. This approach not only maintains the homogeneity of the graph and effectively addresses the problem of node non-alignment caused by dynamic changes but also enhances the accuracy of community detection.
- We evaluate the model performance using different datasets and find that the introduction of contrastive learning techniques can not only capture more stable community structures in the initial snapshots but also help to achieve smooth transitions between adjacent snapshots.

2. Related works

Dynamic community detection is a rapidly evolving research field that focuses on identifying and tracking communities or groups of nodes in dynamic networks. In this section, we briefly review the studies relevant to our work, specifically in the domains of dynamic community detection and contrastive learning.

2.1. Dynamic community detection

Traditional methods primarily include instant-optimal, incremental, and evolutionary clustering methods, which we introduce from the following three aspects.

2.1.1. Instant-optimal clustering methods

The instant-optimal method is carried out in two stages. Initially, the static method is employed to extract the community structure for each snapshot of the dynamic network. Subsequently, similarity measurement is utilized to match communities between adjacent snapshots to capture the dynamic evolution of these communities. These methods typically deal with dynamic networks by extending static methods, which may ignore historical information from previous snapshots [15].

2.1.2. Incremental clustering methods

The incremental clustering method computes the community structure of the current time network based on the previous time network's community structure and the increment of the adjacent time network without the network mutation. This means that by comparing and analyzing network snapshots at adjacent points in time, only the units with data changes and their neighbor nodes are analyzed, thereby identifying the evolution and changes of the community and improving the efficiency of the algorithm. The DyCID [16] is a typical incremental clustering algorithm for overlapping community detection in dynamic networks. The algorithm initially employs an efficient and straightforward seed selection strategy to rapidly identify core nodes, followed by the utilization of a cascade information diffusion model to simulate the evolutionary process of communities. The IncNSA [17] algorithm is an incremental clustering approach that leverages node similarity to identify and adapt to changes in a limited subset of nodes. It assesses the activity of nodes by monitoring variations in their edge connections, enabling efficient tracking of evolving network structures. The efficacy of these methodologies in snapshot analysis is predominantly contingent upon the community structure derived from preceding snapshots. Consequently, as the dynamic network evolves, errors aggregate across each snapshot, resulting in a diminution of algorithmic precision.

2.1.3. Evolutionary clustering methods

Instead of focusing on individual snapshots of a dynamic network, evolutionary clustering methods identify communities by considering the continuous process encompassing all the snapshots. Evolutionary clustering produces a series of clusters by processing the time step process, and each cluster corresponds to a time step of the system. These methods are based on the concept of time smoothing, which assumes that abrupt changes are unlikely to occur in a short period, and thus accurately discern dynamic community structure. The most classic approach is Folino and Pizzuti's DYNMOGA method [18], which formulates the dynamic community detection task as a multi-objective optimization problem to maximize snapshot quality while minimizing the temporal cost of dynamic networks. In recent years, machine learning and deep learning methodologies have been utilized for dynamic community detection to leverage their capacity for capturing intricate patterns and acquiring representations from dynamic network data. Considering the dynamic nature of networks, evolutionary clustering is a more viable approach for solving dynamic community detection problems and can be easily integrated with deep learning methods [19,20]. These methods typically utilize recurrent neural networks (RNNS), graph neural networks (GNN), or depth-generating models to simulate time dependence and capture changing community structures. sE-Autoencoder [21] is an innovative semi-supervised learning method. Its core idea is to use time matrix and regularization terms to effectively deal with the challenge of low-dimensional representation of nonlinear features. By constructing a time matrix, the algorithm successfully extended the nonlinear reconstruction model to the dynamic network environment to capture the dynamic evolution characteristics of network data. In the domain of dynamic graph representation learning, DGCN [22] introduces an innovative methodology that integrates GCN and incorporates the memory mechanism of LSTM. By dynamically updating the weight parameters, it captures the global structural evolution information of the dynamic graph across various time steps. Recent studies have also explored how to improve the performance of the model by introducing external signals, such methods not only consider the internal structural changes of the network, but also include external factors such as user behavior and time series data [23].

2.2. Contrastive learning

Contrastive learning has become a highly influential self-supervised learning paradigm in the field of machine learning. Its core idea is to guide models to learn to distinguish between similar and dissimilar instances by constructing contrastive sample pairs, thereby capturing key features of the data. Through carefully designed contrastive mechanisms, models can automatically extract discriminative feature representations from data without relying on extensive manually labeled datasets. In recent years, this approach has achieved remarkable progress across multiple domains, particularly in computer vision and natural language processing, offering new insights and methodologies for network science tasks such as dynamic community detection.

In the field of computer vision, the MoCo framework proposed by He et al. effectively learns image feature representations by constructing dynamic dictionaries and employing a momentum update mechanism. This method achieves performance comparable to supervised learning methods in tasks such as image classification and object detection [24]. Similarly, SwAV [25] introduces the concept of "clustering contrast", enhancing stability by predicting the cluster to which a sample belongs. Chen et al.'s SimCLR leverages simple yet effective data augmentation strategies and contrastive loss function designs to further improve the efficacy of contrastive learning in visual tasks [26]. In natural language processing, Sentence-BERT focuses on sentence-level semantic representation learning by mapping sentences into a low-dimensional vector space and comparing distances between semantic meanings. This facilitates efficient sentence similarity calculations and tasks like text classification [27]. Meanwhile, XLNet combines autoregressive language models with contrastive learning, enhancing its ability to understand text sequences and demonstrating strong performance in both language generation and understanding tasks [28].

As research progresses, increasing attention has been directed toward more complex data types, such as graph data. In the study of graph data, contrastive learning methods can be categorized into three levels based on the learning objective: graph-level, subgraph-level, and node-level contrastive learning. Graph-level contrastive learning focuses on capturing the overall structural information of a graph. By constructing positive and negative sample pairs for global views, these methods enhance global representation capabilities. For instance, Deep Graph Infomax (DGI) maximizes the mutual information between global graph representations and local node embeddings, effectively capturing the global characteristics of a graph [29]. Similarly, GraphCL introduces a variety of data augmentation techniques, generating different views for contrastive learning, which significantly improves the model's robustness across varying data distributions [6]. However, these methods are less effective in handling dynamic scenarios or capturing localized community features. In contrast, subgraph-level contrastive learning focuses on modeling local structures, making it more suitable for community-related tasks. For example, MVGRL employs multi-view contrastive learning, combining local subgraph embeddings with global graph structures to enhance the model's community-awareness [30]. This method effectively bridges the gap between global and local structural information, making it particularly valuable for tasks that require community-level insights. Node-level

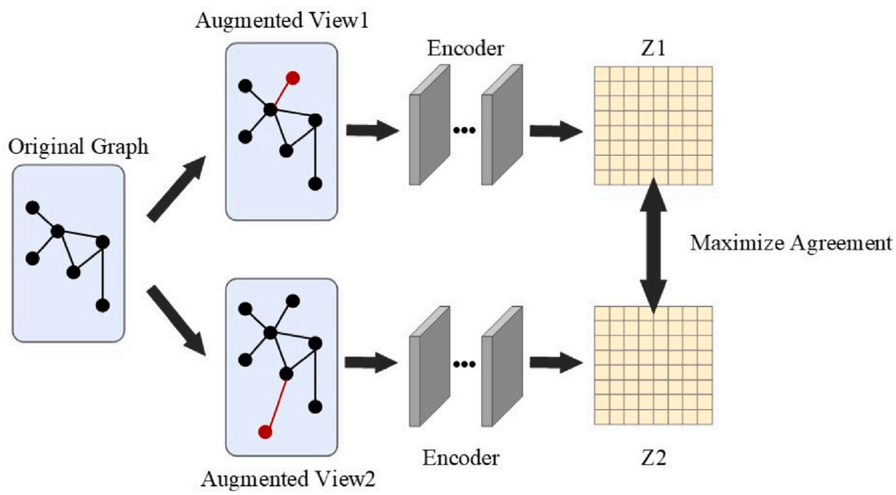


Fig. 2. Graph Contrastive Learning Model.

contrastive learning further refines the modeling of fine-grained features. CLDG [31] puts forward a framework for constructing contrastive pairs among views with different time spans by leveraging the time translation invariance of dynamic graphs. It learns the representations of dynamic graphs in an unsupervised manner, significantly reducing the number of model parameters and training time. jNCDC [32] combines non-negative matrix factorization and graph contrastive learning. By constructing evolutionary graphs and utilizing vertex roles to select positive and negative samples, it proposes a new algorithm to track dynamic communities in temporal networks, which improves the detection accuracy.

3. Preliminaries

In this section, we present some basics. It mainly includes the definition of dynamic community detection and the traditional framework of graph contrastive learning.

3.1. Graph contrastive learning

Contrastive learning (CL) has recently empowered unsupervised computer vision models to achieve performance comparable to that of supervised models. The Visual Contrastive Learning (VCL) framework aims to maximize the similarity between different views of the input (positive views) while minimizing the similarity to other samples in the batch (negative views) [33]. Graph Contrastive Learning (GCL) also inherits this core idea as an application in graph data. The general process can be described as shown in Fig. 2: for a given original graph $G = \{V, E\}$, let G_1 and G_2 be differently augmented views derived from G, and then the information in G_1 and G_2 is passed through the shared encoder to obtain the final node representation $\boldsymbol{Z_1}$ and $\boldsymbol{Z_2}$. Mutual information between the two representations is maximized. By comparing the similarities and differences among various datasets, contrastive learning can assist the model in identifying more distinctive features. Consequently, this process leads to improved accuracy and robustness of the model.

3.2. Problem definition

Before delving into well-established methodologies pertinent to our research, it is essential to establish fundamental definitions of dynamic networks. A dynamic network G, consisting of a sequence of snapshots, can be represented as the sets $G = \{G^1, G^2, \dots, G^T\}$, where T represents the total time step in the entire time series. These snapshots reflect the state and evolution of the network at different points in time. More specifically, $G^t = \{V^t, E^t, \boldsymbol{X}^t\}$ represents a network snapshot captured at time step t (where $1 \le t \le T$). In this snapshot, V^t denotes the set of active nodes at the current time step, which belongs to a larger shared node set $V = \{V_1, V_2, \dots, V_N\}$; E^t depicts the connection relationships among these active nodes at time step t, forming the edge set of the network; and $\boldsymbol{X}^t$, as a feature matrix, records in detail the attributes or state information of all nodes at time step t. This series of snapshots arranged in chronological order jointly depicts the detailed evolution process of the dynamic network G. Existing studies generally assume that all snapshot networks are undirected and unweighted. Dynamic community detection aims to identify and track the evolving communities or clusters within a dynamic network over time. The community structure, denoted by $\{C_i^t\}_{i=1}^k$, corresponds to a network partition characterized by a higher density of internal connections compared to external connections. C_i^t represents the ith community at time t. It is important to note that there is no intersection between any two independent communities, meaning that $C_i^t \cap C_j^t = \emptyset$ if $i \neq j$.

4. Methodology

In this section, we present the motivation and details of the CL-OND model. Specifically, we verify that the real variation of the dynamic network itself conforms to GCL's general graph augmentation (GAME) [34] rules through experimental and theoretical analysis. Then, we use two snapshots of adjacent time steps as augmented views in the contrastive learning framework to achieve accurate dynamic community detection.

4.1. Investigation: Dynamic changes instead of data augmentations

Data augmentation techniques are essential in graph contrastive learning to improve the diversity and quality of the training data. Multiple studies have emphasized their importance [35,36]. However, according to Trivedi's research [10], these augmentation techniques may inadvertently destroy task-relevant information, thereby compromising the model's capacity to learn discriminative representations. Although these DAGA techniques are effective in a large number of models, it is still necessary to manually select augmentations per dataset by trial and error. Given that the different perspectives within a dynamic snapshot network represent inherently multi-viewed data, we propose utilizing two consecutive time-step snapshots instead of manual forms to replace the augmented views. It can eliminate the complicated data augmentation process and accurately retain the original semantic information in the network.

To verify the validity of this idea, we conducted several empirical studies to explore the feasibility and effectiveness of using dynamic

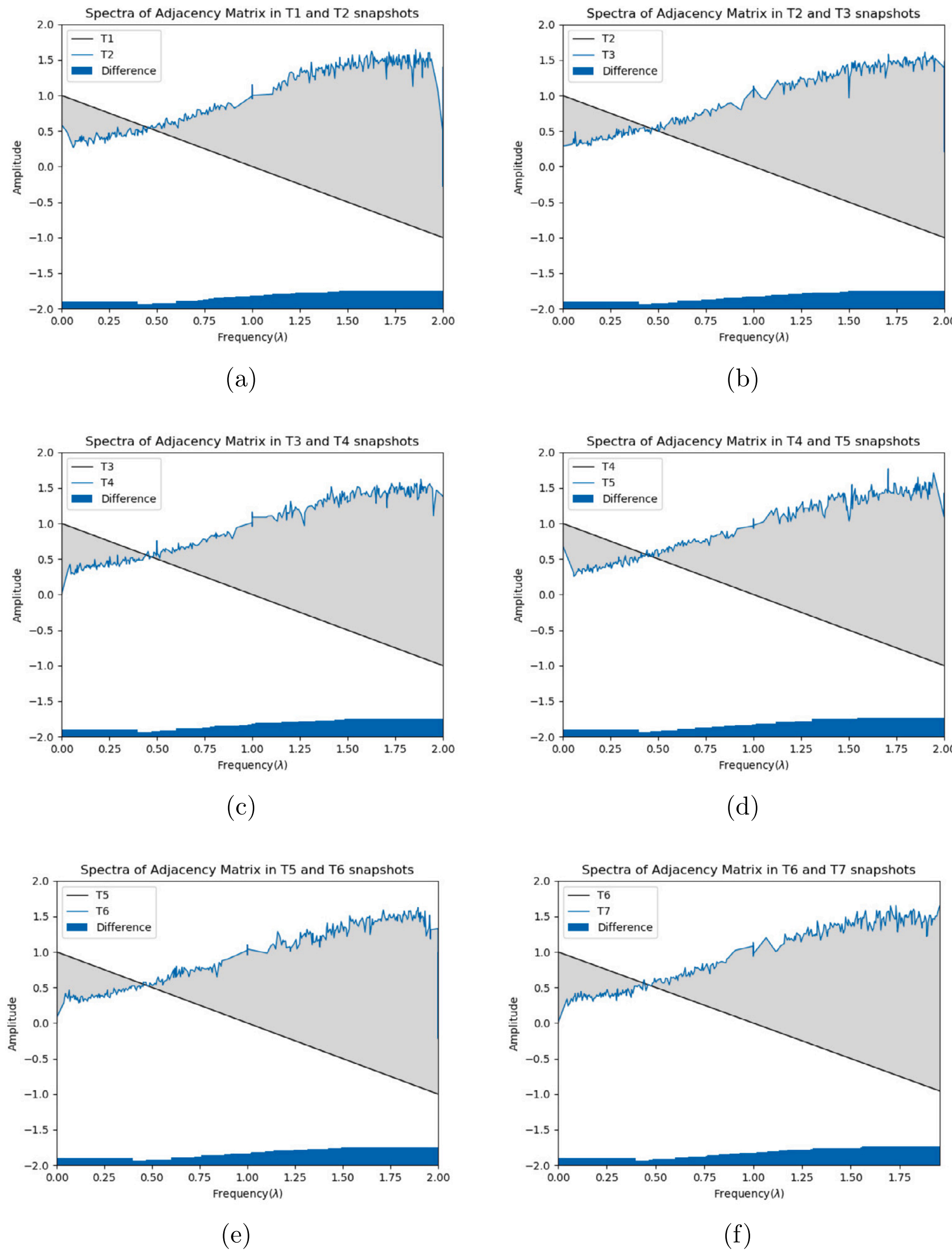


Fig. 3. Spectra of Adjacency Matrix between Adjacent Snapshots in Cellphone Calls.

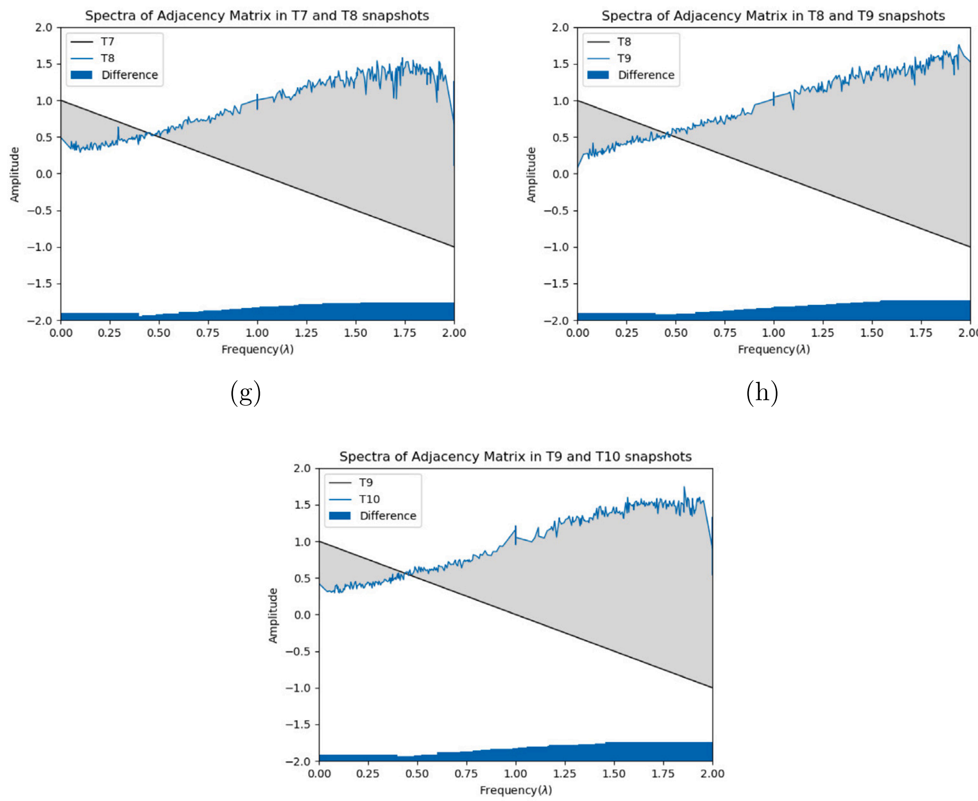
changes instead of data augmentations in the contrastive learning framework. Previous research has demonstrated that graph augmentations preserve the low-frequency components while perturbing the middle and high-frequency components of the graph [37]. Shichuan et al. further summarized the GAME rules of GCL through spectral experimental studies [34]. Specifically, they found that the difference between the high-frequency components of two augmented graphs should be greater than that of the low-frequency components. This finding is important for understanding the behavior of GCL in graph augmentation processes. To verify the feasibility of adjacent snapshots as augmented views, we used the same experimental setup as SpCo [34]. to explore the relationship between the eigenvalues of two adjacent snapshots. Since most augmentations come from the raw adjacency matrix $\boldsymbol{A}$, it is natural to fix one view as $\boldsymbol{A}^t$ and the other as $\boldsymbol{A}^{t+1}$ in dynamic networks [38]. The specific process is as follows:

First, the normalized Laplacian matrix $\boldsymbol{L}^t$ of snapshot G^t is decomposed to obtain the eigenvector $\boldsymbol{u}^t$ and the eigenvalue λ^t. Then, the eigenvalues are arranged in ascending order, and the eigenvectors also get a new order with the change of eigenvalues. Next, we turn to matrix perturbation theory [39] to calculate the changing eigenvalues

$$\Delta\lambda_i^t = \lambda_i^{t+1} - \lambda_i^t = (\boldsymbol{u}_i^t)^T \Delta\boldsymbol{A}^t \boldsymbol{u}_i^t - \lambda_i^t (\boldsymbol{u}_i^t)^T \Delta\boldsymbol{D}^t \boldsymbol{u}_i^t + \mathrm{O}(\|\Delta\boldsymbol{A}^t\|) \quad (1)$$

where $\Delta\boldsymbol{A}^t = \boldsymbol{L}^{t+1} - \boldsymbol{L}^t$, $\Delta\boldsymbol{D}^t = \boldsymbol{D}_{\boldsymbol{L}^{t+1}} - \boldsymbol{D}_{\boldsymbol{L}^t}$ ($\boldsymbol{D}_{\boldsymbol{L}^{t+1}}$ and $\boldsymbol{D}_{\boldsymbol{L}^t}$ are diagonal matrix, the diagonal elements are the sum of the rows of the matrix), we will get $\Delta\lambda^t$. According to Eq. (1), we could get $\lambda^{t+1} = \lambda^t + \Delta\lambda^t$.

Fig. 3 shows the spectra of the adjacency matrix between adjacent snapshots in the Cellphone Calls dataset. The gray part represents the difference in the spectrum between the adjacent matrix, and the blue bar chart at the bottom is a visualization of the difference. The result shows that the dynamic changes of adjacent snapshots fully conform to the GAME rules of GCL, which further proves that it is feasible to replace data augmentations with dynamic changes. The detailed operation of the experiment and the result analysis of other datasets are listed in Section 5.2.



(g) (h)

(i)

Fig. 3. (*continued*).

4.2. A dynamic community detection framework

Based on the above empirical research, we choose adjacent snapshots as the augmented views and propose a novel contrastive learning strategy for optimizing node non-alignment in dynamic community detection. The proposed model modifies the traditional graph contrastive learning framework to achieve end-to-end learning of node-community affiliations while obtaining node embeddings. The following sections outline the implementation steps for this model.

At each epoch, we sequentially sample a dynamic network with T snapshots in the form of $\{G^t, G^{t+1}\}$, and construct $|T|-1$ group of comparison combinations, defined as $C = \{\{G^1, G^2\}, \{G^2, G^3\}, \ldots \ldots, \{G^{t-1}, G^t\}\}$ and $G^t = \{\boldsymbol{A}^t, \boldsymbol{X}^t\}$, where $\boldsymbol{X}^t$ and $\boldsymbol{A}^t$ are the feature matrix and adjacency matrix of the Tth snapshot. As shown in Fig. 4, each contrastive pair $\{G^t, G^{t+1}\}$, in turn is fed into the parameter-sharing encoder. In this paper, we simplify the research process and focus on using GCN as the basic encoder to model structural dependencies. The specific equation is as follows:

$$\mathbf{Z}^{(l+1)} = \sigma\left(\widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{Z}^{(l)}\mathbf{W}^{(l)}\right) \tag{2}$$

where for each snapshot G^t, $\widetilde{\boldsymbol{A}} = \boldsymbol{A}^t + \boldsymbol{I}$, $\boldsymbol{I}$ represents the identity matrix, $\widetilde{\boldsymbol{D}}$ is the degree matrix of $\widetilde{\boldsymbol{A}}$, $\boldsymbol{Z}^{(0)} = \boldsymbol{X}^t$, and $\boldsymbol{W}^{(l)}$ is the trainable transformation matrix of a specific layer. For each contrastive framework, a group of node embedding matrices $\boldsymbol{Z}^t = f_\theta(\boldsymbol{X}^t, \boldsymbol{A}^t)$ and $\boldsymbol{Z}^{t+1} = f_\theta(\boldsymbol{X}^{t+1}, \boldsymbol{A}^{t+1})$ can be obtained, where N indicates the number of nodes and D indicates the embedding dimension.

In addition, different from the contrastive framework in Fig. 2 which focuses on node/graph representation learning, we introduce a randomly initialized community center matrix $\boldsymbol{C}$ and a community affiliation matrix $\boldsymbol{R}$ to realize community detection in an end-to-end manner. The specific operation process is defined as follows:

$$\mathbf{R} = normalize(\delta(\mathbf{Z}, \mathbf{C})) \tag{3}$$

where $\delta()$ represents the cosine similarity function, which is utilized for computing a similarity score between $\boldsymbol{Z}$ and $\boldsymbol{C}$ by constructing abstract community seed node information. The goal is to identify the most appropriate community center for each node in the graph. In particular, we establish a community affiliation matrix $\boldsymbol{R}$, in which each $\boldsymbol{R}[i,:]$ represents a normalized similarity vector that assesses the distance between the ith node and all community centers. Next, we get $\boldsymbol{R}^t$ and $\boldsymbol{R}^{t+1}$ in each contrastive framework. Ultimately, each snapshot yields two outcomes (except the initial and concluding snapshot networks), from which one of the most outstanding community affiliation matrices $\boldsymbol{R}$ is chosen as the outcome.

4.3. Construction of the objective function

In reviewing the development of dynamic community detection, previous approaches for dealing with the task of community detection on snapshot networks have been based on the assumption of time smoothing. This assumption suggests that the network is unlikely to change drastically within a short period, allowing for the discovery of dynamic communities and simultaneous tracking of their dynamic changes. This framework provides time smoothing constraints [2], and the objective function is constructed by snapshot cost (**SC**) and time cost (**TC**) to better evaluate the quality and efficiency of different community detection algorithms on the snapshot network. This approach aims to optimize a quality function in the following form:

$$COST = \alpha \cdot \boldsymbol{SC} + (1-\alpha) \cdot \boldsymbol{TC} \tag{4}$$

where the quality of the community structure at the current time step is estimated by **SC**, while **TC** measures the similarity between the community structure at the current time step and the previous time step. The parameter $\alpha \in (0, 1)$ serves as a correction factor that controls the relative importance of **SC** and **TC**. When $\alpha = 1$, the algorithm

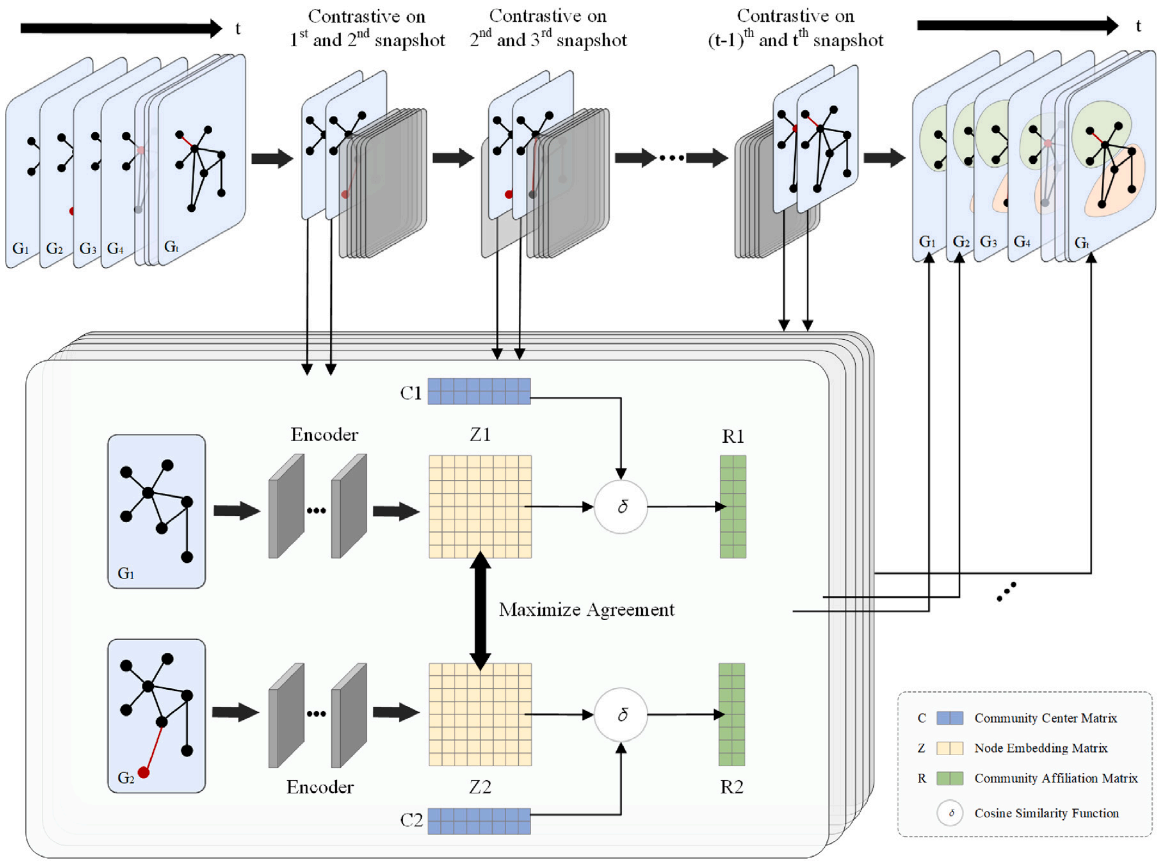


Fig. 4. The Model of CL-OND. Specifically, we feed the snapshot network of adjacent time into different encoders as an augmented view, obtain the node embedding matrix $\boldsymbol{Z}$, and compute the cosine similarity with the initialized community center matrix $\boldsymbol{C}$, and finally obtain the affiliations $\boldsymbol{R}$ between the community and the node through the normalization function. And so forth, we can obtain the optimal community partitioning outcomes for each snapshot and observe how the nodes evolve.

captures a partition of the current network without considering temporal smoothness from the previous network. On the other hand, when $\alpha = 0$, it returns the clustering of the previous network without taking into account snapshot cost. This function simultaneously maximizes the clustering quality at the current time step and minimizes the clustering bias between two consecutive time steps. This ensures that the partition identified at time $t+1$ naturally evolves from the partition identified at time t.

To identify dynamic communities more accurately, this section details and improves the two objective functions **SC** and **TC** for the dynamic community detection task, respectively. Among them, the **SC** aims to identify the optimal community structure in the current snapshot network. It is based on two objective functions (intra-community density and inter-community density) to facilitate community segmentation training, as defined below:

$$D_{intra} = \frac{1}{N}\sum_{i,j}\sum_{k}[\boldsymbol{A}[i,j] - d(k)]\boldsymbol{R}[i,k]\boldsymbol{R}[j,k] \tag{5}$$

$$D_{inter} = \frac{1}{N(N-1)}\sum_{i,j}\sum_{k_1\neq k_2}\boldsymbol{A}[i,j]\boldsymbol{R}[i,k_1]\boldsymbol{R}[j,k_2] \tag{6}$$

In Eqs. (5) and (6), $A[i,j]-d(k)$ and $A[i,j]-0$ represent the difference between the actual local density ($A[i,j]$) and the expected density ($d(k)$ within the community, 0 between the communities). These two objectives are derived from the work of Bolian Li [40], which focuses on training community division by assessing the impact of each edge on the community edge density, as opposed to using the local edge density employed in modularity. By minimizing the common goals, the community center matrix $\boldsymbol{T}$ will be updated to achieve reasonable community zoning. The specific loss function is as follows

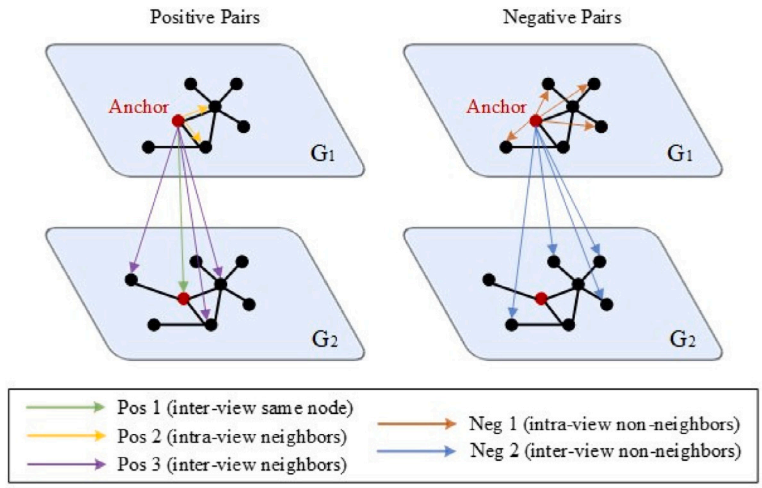
$$L(\boldsymbol{R}) = \lambda D_{inter} - D_{intra} \tag{7}$$

where λ is the co-efficient. Ultimately, we amalgamate the objectives for both graph views in the following manner

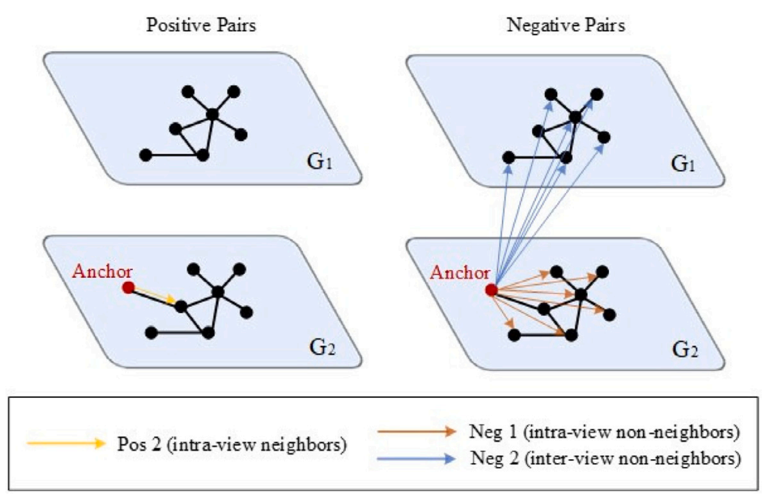
$$L_{SC} = \frac{1}{2}\left[L(\boldsymbol{R}^{t}) + L(\boldsymbol{R}^{t+1})\right] \tag{8}$$

In most of the previous methods, **TC** is measured by the consistency of node embeddings between adjacent snapshots, which is consistent with the learning goal of node-node level graph contrastive learning. This further demonstrates the applicability of graph contrastive learning and dynamic community detection tasks. It is important to note that both InfoNCE and NT-Xent adhere to a fundamental principle: each anchor point is matched to only one pair of positive numbers. Specifically, embeddings representing the same node in two different views are considered positive pairs, while all embeddings representing different nodes are considered negative pairs. In this context, even neighbors of an anchor are classified as negative pairs and pushed away during optimization to emphasize the similarities between positive pairs and the differences between negative pairs. In addition, the practice of replacing augmented views with adjacent snapshots also leads to the problem of non-aligned views, which further increases the difficulty of this task.

To achieve this objective, we introduce a non-aligned neighbor contrastive loss (NA-NCL). The delineation of positive and negative pairs in NA-NCL is illustrated in Fig. 5. It builds on the research of Xiao et al. [41] and further extends it to the scenarios where nodes are not aligned. Choose the ith node z_i^t in snapshot t as the anchor point, the positive pairs come from three disjoint sources: (1) Inter-view same node; (2) Intra-view neighbors; (3) Inter-view neighbors. Negative pairs come from two disjoint sources: (1) intra-view non-neighbor; (2) inter-view non-neighbor. Fig. 5 summarizes the various contrastive scenarios in the non-aligned view. Fig. 5(a) shows a contrastive view where



(a) Common Node in Different Contrastive Views.



(b) Independent Node in Different Contrastive Views.

Fig. 5. Comparison of Positive and Negative Pairs Defined in NA-NCE Loss.

the anchor points are public nodes. The red node represents the same anchor point in view G^1 and the identical node in view G^2. The lines with differently colored arrows indicate positive and negative pairs formed with the anchor. Fig. 5(b) shows the contrastive view where the anchor point is an independent node. Details are the same as in Fig. 5(a). Finally, the non-aligned neighbor contrastive loss between view 1 and view 2 associated with the anchor z_i^t is formulated as follows:

$$L(z_i^t) = -log \frac{\left(e^{\theta\left(z_i^t, z_i^{t+1}\right)/\tau} + \Sigma_{v_j \in N_i}\left(e^{\theta\left(z_i^t, z_j^t\right)/\tau} + e^{\theta\left(z_i^t, z_j^{t+1}\right)/\tau}\right)\right)/N^*}{e^{\theta\left(z_i^t, z_i^{t+1}\right)/\tau} + \Sigma_{j\neq i}\left(e^{\theta\left(z_i^t, z_j^t\right)/\tau} + e^{\theta\left(z_i^t, z_j^{t+1}\right)/\tau}\right)} \tag{9}$$

where N^* denotes the number of positive pairs. As shown in Fig. 5(a), when the anchor is a common node in the contrastive view, $N^* = |N_i^t + N_i^{t+1} + 1|$. As shown in Fig. 5(b), when the anchor is an independent node that cannot be aligned, $N^* \leq |N_i^{t+1}|$. The denominator details can be further elaborated as follows:

$$\sum_{j\neq i} e^{\theta\left(z_i^t, z_j^t\right)/\tau} = \sum_{v_j \in \mathcal{N}_i} e^{\theta\left(z_i^t, z_j^t\right)/\tau} + \sum_{v_j \notin \mathcal{N}_i} e^{\theta\left(z_i^t, z_j^t\right)/\tau} \tag{10}$$

Investigation

$$\sum_{j\neq i} e^{\theta\left(z_i^t, z_j^{t+1}\right)/\tau} = \sum_{v_j \in \mathcal{N}_i} e^{\theta\left(z_i^t, z_j^{t+1}\right)/\tau} + \sum_{v_j \notin \mathcal{N}_i} e^{\theta\left(z_i^t, z_j^{t+1}\right)/\tau} \tag{11}$$

Similarly, considering the embedding in view 2 as the anchor point, the non-aligned neighbor contrastive loss can be equivalently defined following Eq. (9). The ultimate loss for views 1 and view 2 is determined as an average across all nodes:

$$L_{TC} = L(Z^t, Z^{t+1}) = \frac{1}{2}\sum_{i=1}^{N}[L(\mathbf{z}_i^t) + L(\mathbf{z}_i^{t+1})] \tag{12}$$

Hence, the ultimate loss function may be articulated as follows:

$$L = L_{SC} + L_{TC} \tag{13}$$

5. Experiments

In this section, we elaborate on the experimental setup and the corresponding results. To demonstrate the efficacy of our proposed model, we conduct a comparison with robust baseline models for community detection tasks. We commence with four research questions (**RQ**) to guide the experiments and subsequent discussions.

- **RQ1**: From the perspective of spectral theory, can the dynamic changes of the adjacent snapshot network replace the data augmentation? Does it conform to general graph augmentation rules?
- **RQ2**: Does CL-OND achieve state-of-the-art performance on dynamic community detection tasks compared to the baseline model?
- **RQ3**: What are the advantages of introducing contrastive learning techniques for dynamic community detection tasks?
- **RQ4**: What are the advantages of non-aligned neighbor contrastive loss functions? Is it widely applicable?

5.1. Experiments setup

This section provides a detailed introduction to the basic setup of the experiment, including the dataset, comparison method, evaluation index, and parameter setting.

5.1.1. Datasets

In this segment, we introduce a selection of synthetic networks and real-world dynamic networks to assess the effectiveness of the CL-OND model in the task of dynamic community detection. To begin with, for the synthetic networks, we employ the Dynamic Benchmark Network Generator developed by Greene et al. [42] to create a series of unweighted and undirected time-evolving networks that contain authentic communities within them. In the SYN-Events dataset, there are a total of 1000 nodes and 10 snapshots. It examines four types of evolutionary events that occur during the development of synthetic networks: community births and deaths, expansions and contractions, intermittent communities, and mergers and splits. These events are integrated into the synthetic network by the software in the following manner:

- **Births and Deaths**: At each time step, five new communities come into being as a result of detaching nodes from the existing ones, and simultaneously, five other existing communities are randomly removed from the network.
- **Expansion and Contraction**: In each time step, 5 communities were randomly picked and then either enlarged or reduced by 25 percent of their original size.
- **Intermittent Communities**: Hiding 10% of the community in the snapshot network at the moment t=1.
- **Merging and splitting**: In each time step, 5 communities are divided, and another 5 communities are chosen. Subsequently, each pair of the selected communities is combined.

Secondly, to assess the applicability and effectiveness of the model, we have chosen two real-world dynamic networks[1]: Cellphone Calls and High School. Both of these networks have produced genuine community partitioning results.

- **Cellphone Calls**: The dataset comprises records of mobile phone calls among members of the fictional Paraiso movement, spanning ten days of data in June 2006. It includes 400 nodes and is divided into 10 snapshots.
- **High School**: The dataset contains time-network contact data between students at a high school in Marseille, France, with 327 nodes. It is divided into 9 snapshots.

5.1.2. Baselines and evaluation metrics

In this section, we choose seven dynamic community detection algorithms: **DYNMOGA** [18], **DyCID** [16], **IncNSA** [17], **sE-Autoencoder** [21], **DGCN** [22], **CLDG** [31] and **jNCDC** [32] to compare with the CL-OND model. It includes the classical evolutionary clustering method, incremental clustering method, dynamic embedding method, dynamic community detection method using deep learning and contrastive learning, etc, which can evaluate the effectiveness of the model in this paper.

In terms of the selection of evaluation metrics, Shchur et al. [43] pointed out that some commonly used metrics to quantify the consistency between real communities and detected communities, such as Jaccard and F1 scores, have limitations. These metrics may mistakenly assign high scores to communities that do not provide valid information. Thus, to more accurately assess community discovery quality, this paper adopts Normalized Mutual Information (**NMI**) as the primary evaluation criterion, as it better captures the similarity between real and detected communities. Further, to conduct a more thorough evaluation of the effectiveness of community detection models, we have selected **ERROR** [21] as a supplementary evaluation metric, intending to quantify the deviations between the community partitions outputted by the model and the true community structures.

5.1.3. Parameter setting

In this section, each composite network consists of 10 snapshots, with each snapshot containing 1000 nodes. The average degree is 20, the maximum degree is 50, and the mixing parameter μ=0.2. Specifically, the value of the hybrid parameter μcontrols the proportion of edges per node connected to other community nodes, which is the default value in this generator. In the model in this chapter, a graph encoder using a 2-layer GCN as the baseline for each deep learning is adopted, and an Adam optimizer [44] is used to optimize each model. It is important to note that while we have chosen GCN as the encoder in this study, the CL-OND model allows for the use of any encoder network architecture without limitations. Before implementing the SC target, both graph views undergo a shard projection module (a 2-layer MLP) with identical input, hidden layer, and output dimensions to enhance their representation. The number of communities is set according to the number of classes in each diagram. For parameter Settings, the learning rate is set to 0.01, λ set to 0.5. Regarding parameters α, we followed the settings of most previous works and set it to 0.5. In specific experiments, $\alpha = 0.5$ indeed achieved the best performance, aiming to balance the snapshot cost and time cost effectively [18,45,46].

5.2. Feasibility analysis of adjacent snapshots instead of data augmentation

In response to **RQ1**, this section conducts an empirical study on two real-world datasets, aiming to explore whether the scheme of dynamic changes instead of data augmentation is consistent with the general graph augmentation rules from the perspective of graph spectral theory. Fig. 3 illustrates the spectral of the adjacency matrix between adjacent snapshots in the Cellphone Calls dataset, which is divided into a network composed of 10 snapshots. According to the combination rule of $\{T^t, T^{t+1}\}$, it can be formed into 9 different groups of comparison schemes, and the experimental results of each group of comparison schemes correspond to the 9 spectral comparison diagrams in Fig. 3 respectively. Consider the spectrum plots of the adjacency matrices in snapshots T^1 and T^2 in Fig. 3(a). By fixing the eigenvalues and eigenvectors of the snapshot network at time T^1, we use Eq. (1) to calculate $\Delta \boldsymbol{A}$ and get the eigenvalues and eigenvectors of the snapshot network at time T^2. Then, plot the spectrum of the two sets of signals and note the absolute difference between the two (shown as the blue bar at the bottom of the horizontal axis). As can be seen from the figure, the low-frequency difference between the two signals is smaller than the high-frequency difference. Throughout the 9 groups of contrastive

[1] http://www.cs.umd.edu/hcil/VASTchallenge08/download/Download.htm.

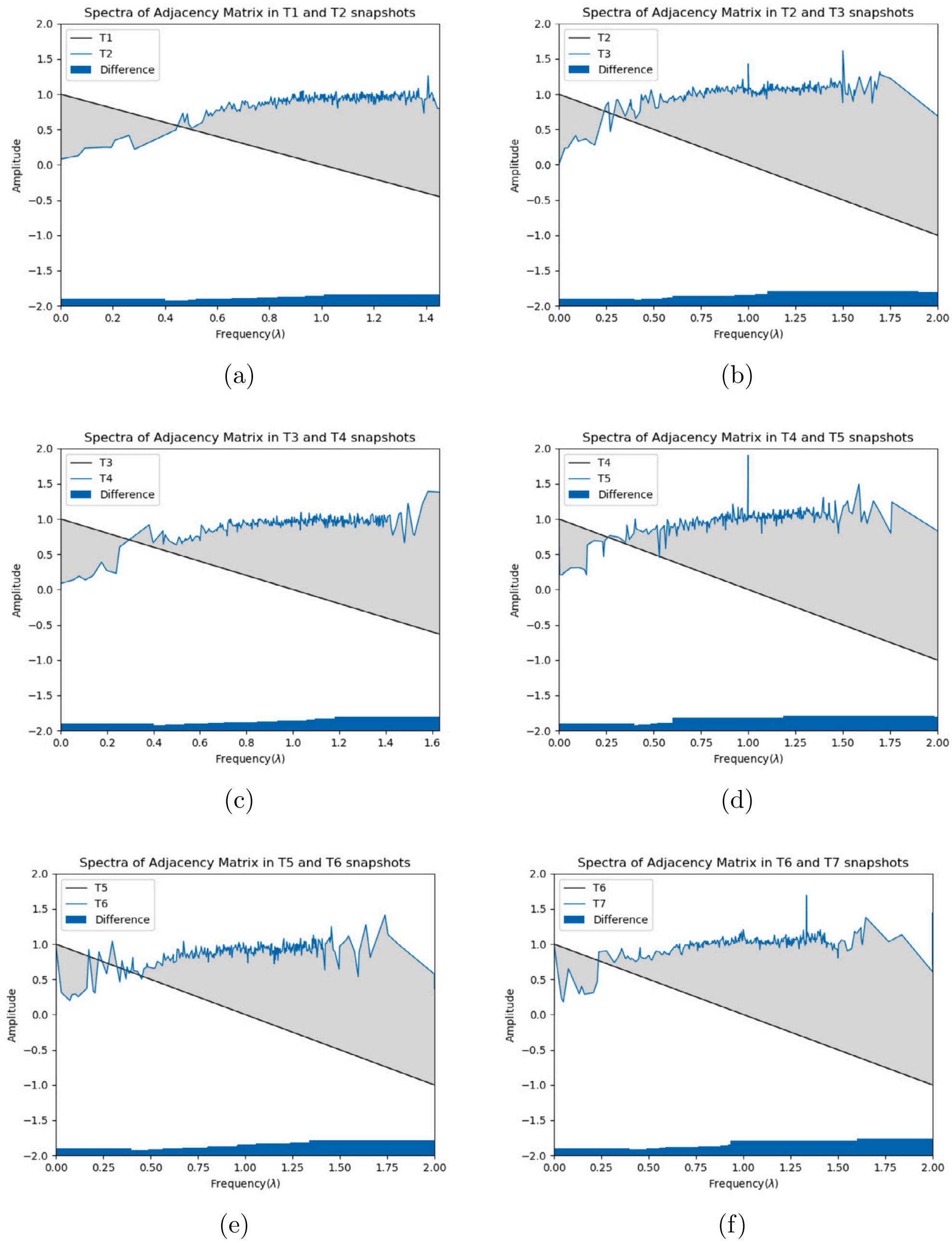


Fig. 6. Spectra of Adjacency Matrix between Adjacent Snapshots in High School.

scenarios in the Cellphone Calls dataset, all of them satisfy the GAME rules.

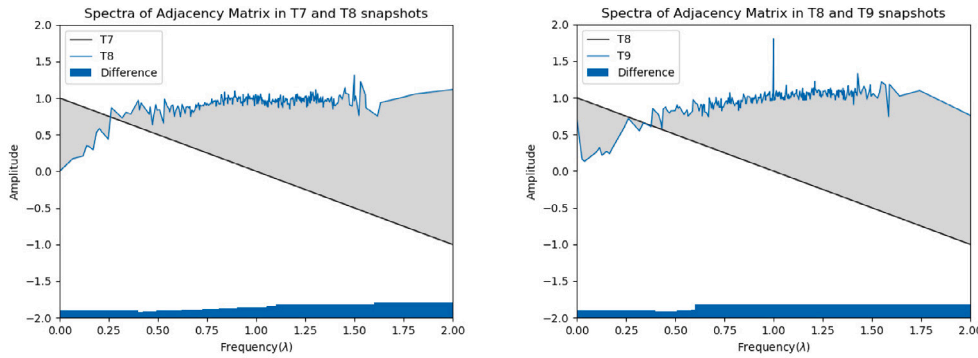
Fig. 6 illustrates the spectral of the adjacency matrix between adjacent snapshots in the High School dataset, which is divided into a network composed of 9 snapshots. According to the combination rule of $\{T^t, T^{t+1}\}$, it can be formed into 8 different groups of contrastive schemes, and the experimental results of each group of contrastive schemes correspond to the 8 spectral contrastive diagrams in Fig. 6 respectively. Throughout the 8 groups of contrastive scenarios in the High School dataset, all of them satisfy the GAME rules.

Based on the above experimental exploration of two real network datasets, further proves the effectiveness of using dynamically changing adjacent snapshots instead of data augmentation schemes and provides a solid foundation for subsequent dynamic community detection tasks. In the future, more contrastive combinations can be further explored to obtain the most suitable ones for downstream tasks, thus reducing the data preprocessing process and achieving better community segmentation.

5.3. Analysis of the results of comparative experiments

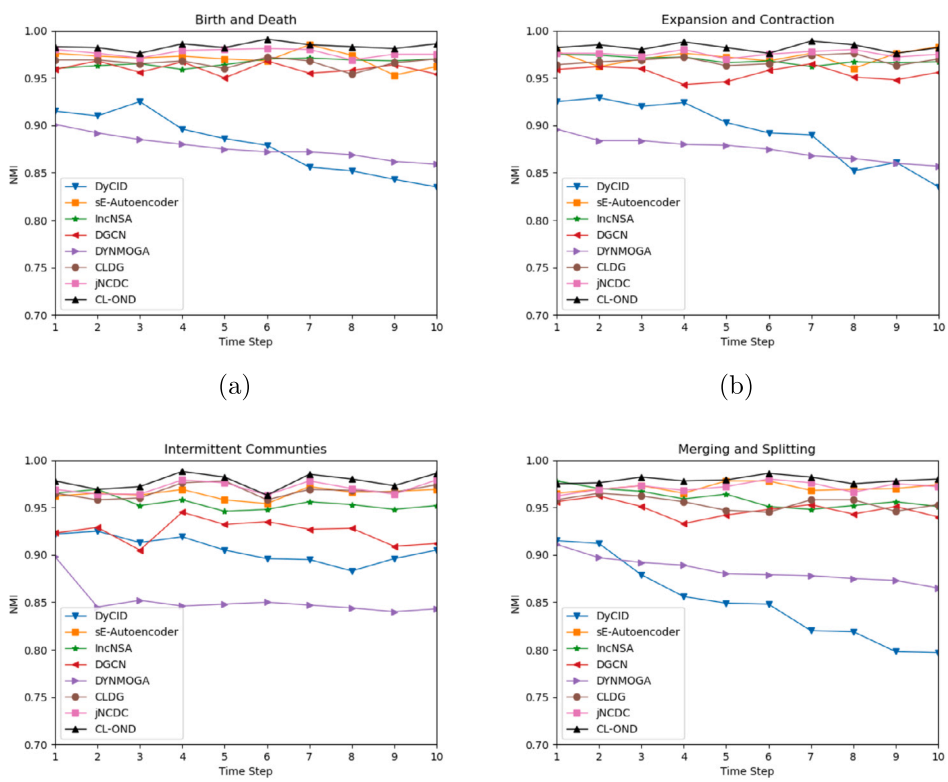
In response to **RQ2**, this section will focus on elucidating the practical performance of the CL-OND model in executing dynamic community detection tasks on both synthetic and real-world network datasets. To this end, we have meticulously selected seven baseline models for comparative analysis. These models broadly encompass both classical and novel dynamic community detection algorithms, ensuring a comprehensive scope of comparison.

On the SYN-EVENTS synthetic dataset, we meticulously design four different types of data embeddings and calculate the NMI values based



(g) (h)

Fig. 6. (*continued*).



(a) (b) (c) (d)

Fig. 7. Comparative Experimental Results in Four Events in SYN-EVENTs.

on them, using this as a quantitative indicator to assess the performance of models in dynamic community detection tasks. Fig. 7 shows our CL-OND model achieves good performance across all networks. Fig. 7(a) shows the results of the network considering community births and deaths, where CL-OND identifies the same results as true communities in almost all snapshots, successfully revealing community births and deaths. Fig. 7(b) shows the network results considering community expansion and contraction, CL-OND also achieves good results, successfully revealing the expansion and contraction of communities. Among them, sE-Autoencoder, CLDG and jNCDC all perform well on this series of networks, while DyCID and DYNMOGA show less satisfactory results. In particular, the performance of the DyCID model decreases gradually with the extension of the time step. Fig. 7(c) shows the results considering intermittent community networks, CL-OND also achieves good results, but the DGCN method is not good at dealing with such networks. For dynamic networks reflecting community merging and splitting, the detection results are shown in Fig. 7(d). In this series of networks, the NMI value of each method decreases over time. This trend suggests that community structures have become more chaotic and harder to detect over time. The CL-OND model detects all NMI values greater than 0.95 from each snapshot, reflecting greater stability.

Table 1 presents a detailed list of the NMI values for various snapshot networks within the Cellphone Calls dataset. The results indicate that the average NMI value of the CL-OND model reaches 69.0%, which outperforms all other compared methods, adequately demonstrating the excellent clustering performance of the CL-OND model when dealing with mobile phone call datasets. Further analysis reveals that in most test instances (including T1, T2, T3, T4, T6, T7, T8, and T9), the NMI values of the CL-OND model are close to or exceed the threshold of 68%, which not only reflects the good stability of the model across different datasets but also showcases its broad applicability. Although the CL-OND model did not achieve the best performance in the test

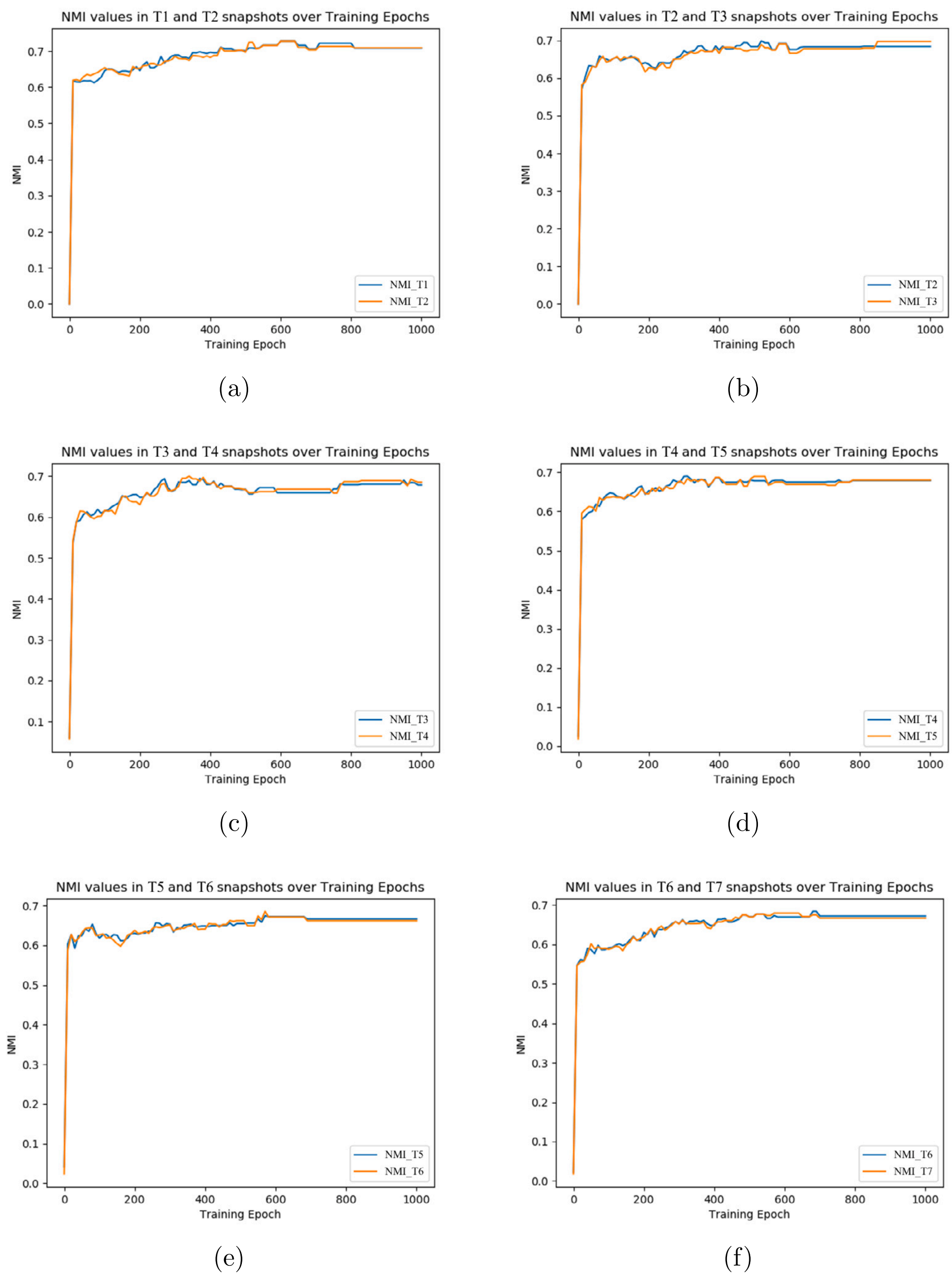


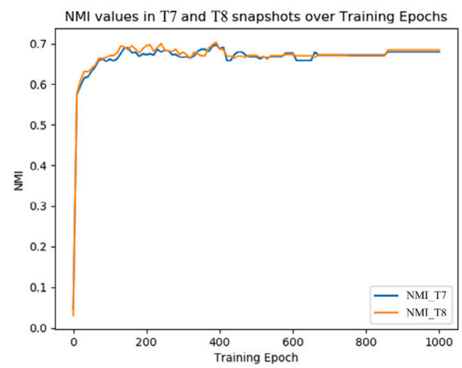
Fig. 8. NMI Values in Adjacency Snapshots over Training Epochs in Cellphone Calls.

instances T5 and T10, the gap between its results and the optimal ones is relatively small, which once again verifies the excellent and stable overall performance of the CL-OND model. Especially in the first two snapshots (T^1 and T^2), CL-OND can obtain an efficient and stable community structure at the initial time compared with the traditional models based on evolutionary clustering. Table 2 shows the NMI values of each snapshot network in the High School dataset, and the results are basically consistent with the above conclusions, among which the CL-OND model also performs well on most snapshots.

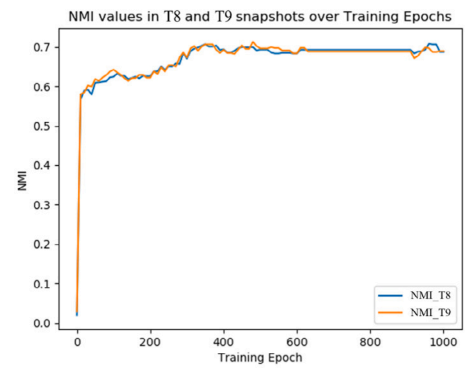
As a result, both in synthetic and real networks, the CL-OND model not only obtains efficient community detection results but also achieves better accuracy and higher stability in the initial few snapshots. In future work, the limits of contrastive learning techniques can be further explored.

5.4. Advantages of introducing contrastive learning

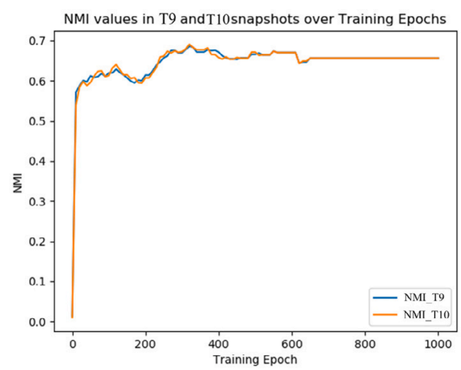
In response to **RQ3**, this section will focus on the advantages of introducing contrastive learning in dynamic community detection tasks. The introduction of contrastive learning is not out of nowhere. It is based on the core idea of node-node level GCL task, that is, to seek consistency of the same node representation in different contrastive views. Coincidentally, the core idea of time cost (**TC**) in dynamic community detection tasks is also to seek the consistency of the same node representation in the adjacent snapshot network to ensure a smooth transition. Based on the consistency of the above two different task ideas, introducing contrastive learning does not increase the number of objective functions. Moreover, it draws on the rich information of adjacent views to improve the consistency and accuracy of our model.



(g)



(h)



(i)

Fig. 8. (*continued*).

Table 1
Comparative results in cellphone calls.

	DYN	DyCID	**sE-AE**	IncNSA	DGCN	CLDG	**jNCDC**	CL-OND
T1	47.3	57.8	67.5	63.5	63.2	68.4	70.1	**71.2**
T2	46.5	58.6	70.4	66.9	66.5	67.9	69.2	**70.9**
T3	43.6	56.4	68.0	67.2	70.6	67.2	68.5	**69.5**
T4	45.1	56.1	67.2	68.0	68.4	67.5	68.2	**68.6**
T5	44.0	54.3	**68.9**	66.4	67.6	66.3	67.7	67.4
T6	44.8	53.0	66.7	67.2	66.5	68.0	67.2	**68.6**
T7	45.3	54.9	67.3	66.7	62.9	68.2	66.8	**68.8**
T8	46.4	56.2	69.1	68.1	65.4	68.5	68.5	**69.2**
T9	46.5	57.8	68.7	67.5	66.1	67.4	68.2	**69.5**
T10	42.9	57.0	64.9	65.8	66.5	65.6	**67.4**	66.7
AVG	45.2	56.2	67.9	66.7	66.4	67.5	68.2	**69.0**

Best performer in bold (NMI(%)).

Table 2
Comparative Results in High School.

	DYNM	DyCID	**sE-AE**	IncNSA	DGCN	CLDG	**jNCDC**	CL-OND
T1	65.4	72.6	82.6	79.6	80.2	82.6	83.5	**84.8**
T2	68.9	71.5	83.0	75.9	68.4	79.1	81.2	**83.1**
T3	66.8	70.9	**75.6**	70.1	70.6	71.2	73.8	74.0
T4	67.2	68.5	71.2	71.8	**73.5**	69.0	72.1	72.0
T5	67.6	64.3	67.6	66.4	67.8	65.4	67.0	**68.9**
T6	68.0	66.0	68.4	67.6	61.6	66.2	67.7	**69.5**
T7	66.3	67.2	68.9	65.9	66.3	67.7	68.4	**69.2**
T8	67.5	67.3	69.2	69.4	70.0	72.6	68.9	**70.3**
T9	66.7	66.1	67.8	67.0	65.4	65.4	67.5	**68.5**
AVG	67.2	68.2	72.7	70.7	69.3	71.2	72.3	**73.4**

Best performer in bold (NMI(%)).

Fig. 8 shows the change of mutual information value between adjacent snapshots within 1000 epochs in the Cellphone Calls. The dynamic network is divided into 10 snapshot networks, which can form 9 different sets of comparison schemes according to the combination rule $\{T^t, T^{t+1}\}$. The experimental results of each group of comparison schemes correspond to the 9 graphs in Fig. 8 respectively. Taking the contrastive view composed of T^1 and T^2 snapshots in Fig. 8(a) as an example, the NMI value between the two sets of signals gradually converges and becomes stable with the increase of training rounds. All the 9 comparison groups showed good performance. Fig. 9 shows the change of mutual information value between adjacent snapshots within 800 epochs in the High School. The dynamic network is divided into 9 snapshot networks, which can form 8 different sets of comparison schemes according to the combination rule $\{T^t, T^{t+1}\}$. The experimental results of each group of comparison schemes correspond to the 8 graphs in Fig. 9 respectively. All of the 8 comparison groups showed good performance, and the results obtained are consistent with the results of the Cellphone Calls dataset.

Table 3
Ablation experiments in cellphone calls dataset.

	DYNMOGA	**sE-Autoencoder**	**jNCDC**	CL-OND
SC+TC$_{InfoNCE}$	39.4	61.9	63.5	64.8
SC+TC$_{NT-Xent}$	42.6	64.5	65.8	67.2
SC+TC$_{NA-NCE}$	45.2	67.9	68.2	**69.0**

Best performer in bold (NMI(%)).

5.5. Ablation experiment

In response to **RQ4**, this section discusses the advantages of non-aligned neighbor contrastive loss (NA-NCE) over traditional contrastive loss. Table 3 shows an ablation experiment using a different contrastive loss function to replace the cost time (**TC**) in the Cellphone Calls

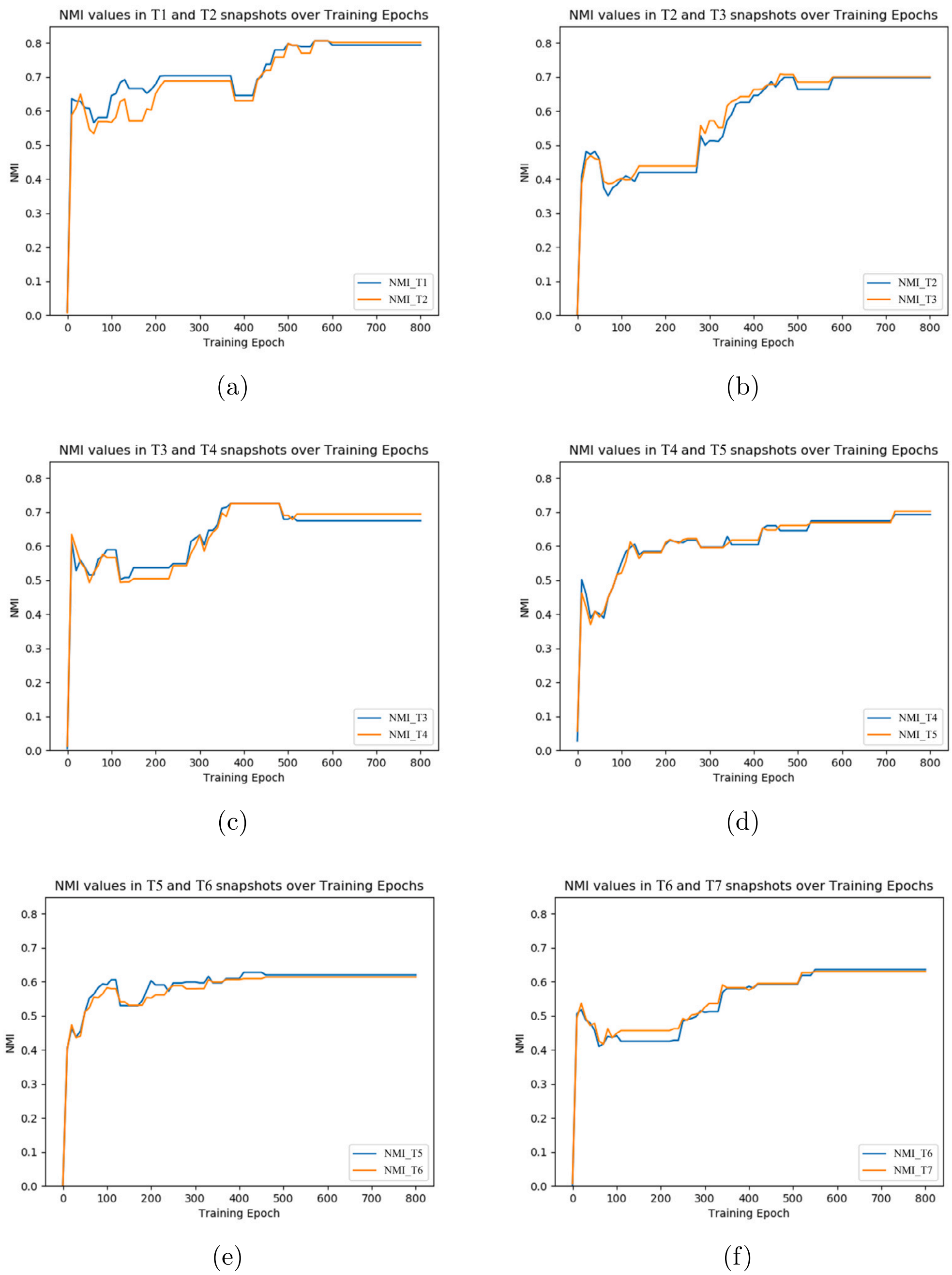


Fig. 9. NMI Values in Adjacency Snapshots over Training Epochs in High School.

Table 4
Ablation experiments in high school dataset.

	DYNMOGA	**sE-Autoencoder**	**jNCDC**	CL-OND
SC+TC$_{InfoNCE}$	61.4	67.0	68.1	68.9
SC+TC$_{NT-Xent}$	63.5	69.2	69.8	70.1
SC+TC$_{NA-NCE}$	67.2	72.7	72.3	**73.4**

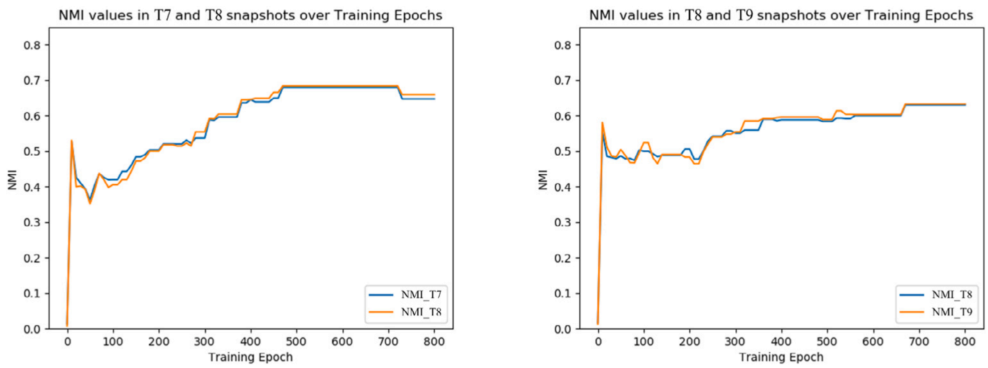
Best performer in bold (NMI(%)).

dataset. The experimental results show that the NA-NCE loss is more suitable for dealing with complex and changeable scenes in dynamic networks, and achieves better community division results. Table 4 shows an ablation experiment using a different contrastive loss function to replace the cost time (**TC**) in the High School dataset, which also achieved the same effect as the Cellphone Calls dataset. In conclusion, the proposed non-aligned neighbor contrastive loss makes it more suitable for dealing with graph fields and non-aligned task scenarios and also provides a foundation for more complex tasks in the future.

5.6. *Error analysis and computational efficiency*

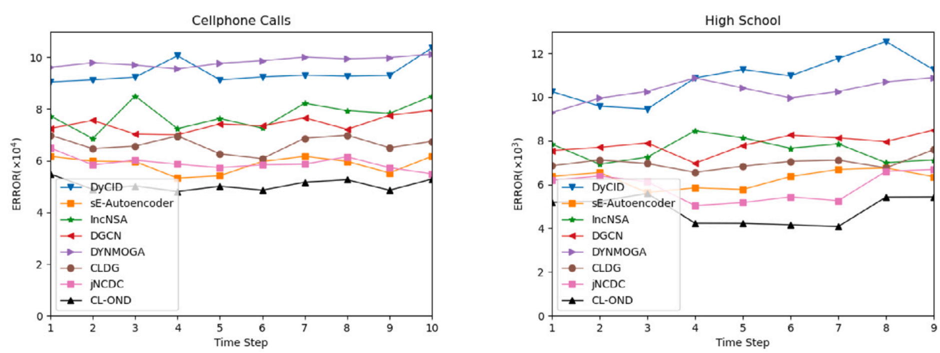
In this section, we evaluate the error performance and computational efficiency of our model on two real-world datasets: Cellphone Calls and High School.

To assess the accuracy, we calculated the **ERROR** metric — which quantifies the deviations between the detected communities and the true community structures — for both datasets. As illustrated in Fig. 10(a), on the Cellphone Calls dataset, the CL-OND method consistently achieves the lowest error at each time step compared to other methods. Similarly, Fig. 10(b) shows that on the High School dataset, the CL-OND method also attains the lowest error across all time steps.



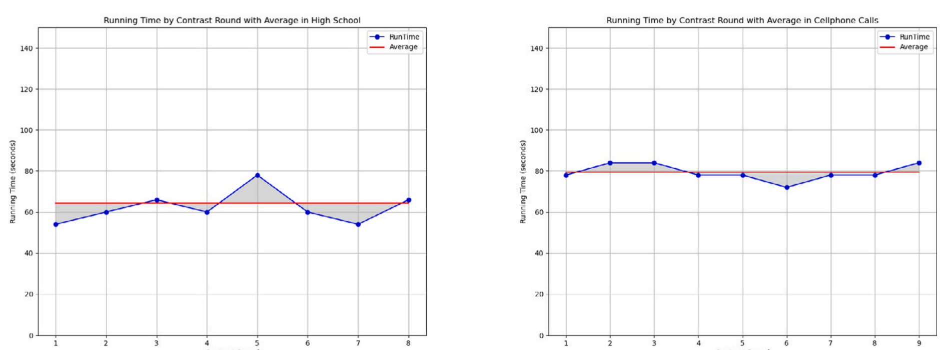
(g) (h)

Fig. 9. (*continued*).



(a) (b)

Fig. 10. Error Analysis using the ERROR metric.



(a) (b)

Fig. 11. Computational Time Analysis with Varying Contrastive Rounds.

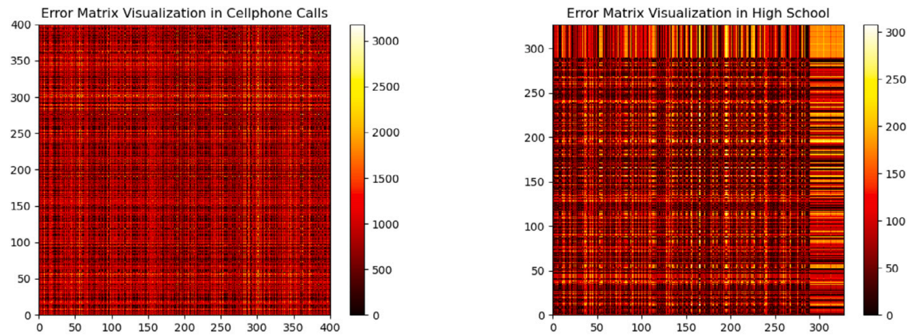
Regarding computational efficiency, we measured the runtime of our model on both datasets, including the actual runtime and average runtime under different numbers of contrastive rounds. As shown in Fig. 11, our model maintains reasonable computational times across different settings. Since our approach does not require additional data augmentation to generate contrastive views, it simplifies the preprocessing steps and reduces computational overhead. This efficiency makes our method practical for real-world dynamic networks.

5.7. Visualization of error matrices and community evolution

As shown in Fig. 12, we present the heatmaps of the error matrices for both datasets. In these heatmaps, a redder color indicates a smaller error value. Our model achieves very low errors, as evidenced by the prominent red regions in the heatmaps, demonstrating the high accuracy of our community detection results.

To further understand the performance of our model, we visualize the community evolution process on the Cellphone Calls dataset, addressing practical challenges such as difficulty in community tracking and subtle effects in dynamic networks. We adopt the approach proposed by Vehlow et al. [47], combining community structure perspectives with alluvial representations.

As depicted in Fig. 13, a custom sorting algorithm is employed for the communities and vertices at each time step to minimize crossings, along with an automatic color assignment method to enhance



(a) (b)

Fig. 12. Heatmaps Illustrating the Error Distribution.

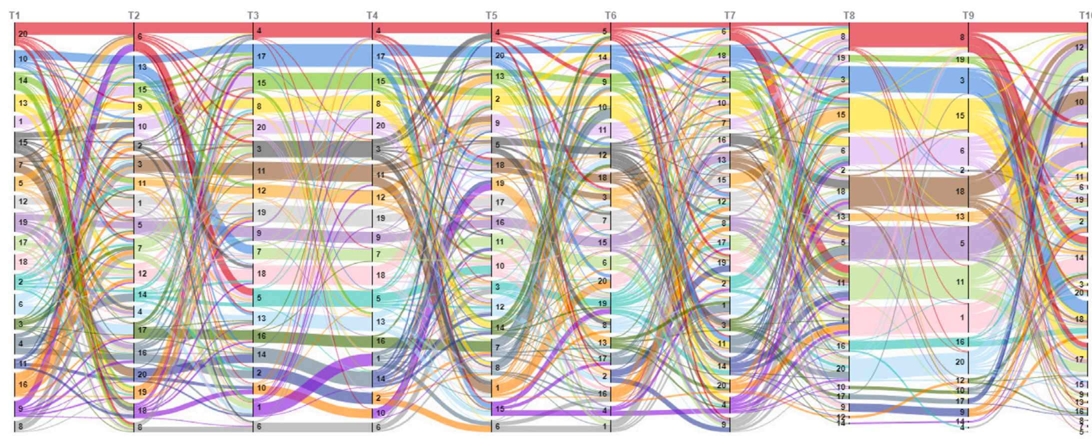


Fig. 13. Visualization of Community Evolution Process in the Cellphone Calls Dataset.

readability. This visualization leverages the advantages of alluvial diagrams to intuitively understand the flow trends of each community node at different time states from a local perspective. From a global perspective, we can intuitively grasp the overall flow dynamics and migration directions of nodes at each moment, which facilitates detailed observations of special nodes or communities in subsequent analyses.

6. Conclusion

This paper proposes a dynamic community detection model based on contrastive learning. The core idea of this model is to cleverly use adjacent snapshots as a contrastive view of the GCL framework. By constructing positive and negative sample pairs, the temporal dynamic characteristics and internal structure of the evolution graph are effectively captured, the discriminative feature learning ability of the model is improved, and the accuracy of community detection is guaranteed. In addition, through a series of in-depth experiments on synthetic networks and real networks, it is found that the model shows excellent performance. Among them, the feasibility analysis of adjacent snapshots rather than data augmentation further confirms this point. More importantly, by avoiding complex and expensive data augmentations, the model greatly reduces the pre-processing process of model data. This finding provides a new perspective and solution for the study of dynamic community detection. In the end, we look forward to further exploring the connection between contrastive learning and dynamic networks to better address the complexity and dynamics of real-world network environments and to provide more efficient and accurate solutions for community detection tasks.

CRediT authorship contribution statement

Xiaohong Li: Writing – review & editing, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Wanyao Shi:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Qixuan Peng:** Writing – review & editing, Visualization, Validation, Supervision, Software, Resources, Investigation, Formal analysis, Data curation. **Hongyan Ran:** Writing – review & editing, Project administration, Funding acquisition, Data curation.

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.

Acknowledgments

This work was supported in part by the Gansu Province Youth Science and Technology Fund 24JRRA148 and Backbone Project of Northwest Normal University NWNU-LKQN2024-22.

Data availability

Data will be made available on request.

References

[1] J. Skarding, B. Gabrys, K. Musial, Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey, IEEE Access 9 (2021) 79143–79168.

[2] G. Rossetti, R. Cazabet, Community discovery in dynamic networks: a survey, ACM Comput. Surv. 51 (2) (2018) 1–37.

[3] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, Y. Wang, Dynamic neural networks: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 44 (11) (2021) 7436–7456.

[4] Q. Wang, Y. Ma, K. Zhao, Y. Tian, A comprehensive survey of loss functions in machine learning, Ann. Data Sci. 9 (2022).

[5] X. Li, X. Zhen, X. Qi, H. Han, L. Zhang, Z. Han, Dynamic community detection based on graph convolutional networks and contrastive learning, Chaos Solitons Fractals 176 (2023) 114157.

[6] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, Y. Shen, Graph contrastive learning with augmentations, Adv. Neural Inf. Process. Syst. 33 (2020) 5812–5823.

[7] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, L. Wang, Graph contrastive learning with adaptive augmentation, in: Proceedings of the Web Conference 2021, 2021, pp. 2069–2080.

[8] H. Yang, H. Chen, S. Pan, L. Li, P.S. Yu, G. Xu, Dual space graph contrastive learning, in: Proceedings of the ACM Web Conference 2022, 2022, pp. 1238–1247.

[9] K. Hassani, A.H. Khasahmadi, Contrastive multi-view representation learning on graphs, in: International Conference on Machine Learning, PMLR, 2020, pp. 4116–4126.

[10] P. Trivedi, E.S. Lubana, Y. Yan, Y. Yang, D. Koutra, Augmentations in graph contrastive learning: Current methodological flaws & towards better practices, in: Proceedings of the ACM Web Conference 2022, pp. 1538–1549.

[11] X. Su, S. Xue, F. Liu, J. Wu, J. Yang, C. Zhou, W. Hu, C. Paris, S. Nepal, D. Jin, A comprehensive survey on community detection with deep learning, IEEE Trans. Neural Netw. Learn. Syst. pp (2022).

[12] X. Shen, D. Sun, S. Pan, X. Zhou, L.T. Yang, Neighbor contrastive learning on learnable graph augmentation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, No. 8, 2023, pp. 9782–9791.

[13] J. Duan, S. Wang, P. Zhang, E. Zhu, J. Hu, H. Jin, Y. Liu, Z. Dong, Graph anomaly detection via multi-scale contrastive learning networks with augmented view, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, No. 6, 2023, pp. 7459–7467.

[14] Y. Zhu, J. Guo, F. Wu, S. Tang, RoSA: A robust self-aligned framework for node-node graph contrastive learning, in: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, 2022.

[15] G. Palla, A.-L. Barabási, T. Vicsek, Quantifying social group evolution, Nature 446 (7136) (2007) 664–667.

[16] L. He, W. Guo, Y. Chen, K. Guo, Q. Zhuang, Discovering overlapping communities in dynamic networks based on cascade information diffusion, IEEE Trans. Comput. Soc. Syst. 9 (3) (2021) 794–806.

[17] X. Su, J. Cheng, H. Yang, M. Leng, W. Zhang, X. Chen, IncNSA: Detecting communities incrementally from time-evolving networks based on node similarity, Internat. J. Modern Phys. C 31 (07) (2020) 2050094.

[18] F. Folino, C. Pizzuti, An evolutionary multiobjective approach for community discovery in dynamic networks, IEEE Trans. Knowl. Data Eng. 26 (8) (2013) 1838–1852.

[19] F. Zhang, J. Zhu, Z. Luo, Z. Wang, L. Tao, C. Gao, Community detection in dynamic networks: A novel deep learning method, in: International Conference on Knowledge Science, Engineering and Management, Springer, 2021, pp. 115–127.

[20] L. Wu, Y. Ouyang, C. Shi, C.-H. Chen, Deep learning-based dynamic community discovery, in: Database Systems for Advanced Applications. DASFAA 2021 International, Springer, 2021, pp. 237–248.

[21] Z. Wang, C. Wang, C. Gao, X. Li, X. Li, An evolutionary autoencoder for dynamic community detection, Sci. China Inf. Sci. 63 (11) (2020) 212205.

[22] C. Gao, J. Zhu, F. Zhang, Z. Wang, X. Li, A novel representation learning for dynamic graphs based on graph convolutional networks, IEEE Trans. Cybern. (2022).

[23] J. Yu, D. Tao, M. Wang, Y. Rui, Learning to rank using user clicks and visual features for image retrieval, IEEE Trans. Cybern. 45 (4) (2014) 767–779.

[24] A. Khan, A. Sohail, M. Fiaz, e.a. Hassan, A survey of the self supervised learning mechanisms for vision transformers, 2024, arXiv preprint arXiv:2408.17059.

[25] E. Fini, P. Astolfi, Alahari, Semi-supervised learning made simple with self-supervised clustering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 3187–3197.

[26] C. Zhou, Q. Li, C. Li, J. Yu, Y. Liu, G. Wang, K. Zhang, C. Ji, Q. Yan, L. He, et al., A comprehensive survey on pretrained foundation models: A history from bert to chatgpt, Int. J. Mach. Learn. Cybern. (2024) 1–65.

[27] X. Shen, Z. Chen, M. Backes, Y. Shen, Y. Zhang, "Do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models, in: Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, 2024, pp. 1671–1685.

[28] YangJingfeng, JinHongye, TangRuixiang, HanXiaotian, et al., Harnessing the power of LLMs in practice: A survey on ChatGPT and beyond, ACM Trans. Knowl. Discov. Data (2024).

[29] A. Mehrish, N. Majumder, R. Bhardwaj, S. Poria, A review of deep learning techniques for speech processing, Inf. Fusion (2023).

[30] K. Hassani, A.H. Khasahmadi, Contrastive multi-view representation learning on graphs, in: International Conference on Machine Learning, PMLR, 2020, pp. 4116–4126.

[31] Y. Xu, B. Shi, T. Ma, B. Dong, H. Zhou, Q. Zheng, CLDG: Contrastive learning on dynamic graphs, in: 2023 IEEE 39th International Conference on Data Engineering, ICDE, IEEE, 2023, pp. 696–707.

[32] Y. Ai, X. Xie, X. Ma, Graph contrastive learning for tracking dynamic communities in temporal networks, IEEE Trans. Emerg. Top. Comput. Intell. (2024).

[33] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning, PMLR, 2020, pp. 1597–1607.

[34] N. Liu, X. Wang, D. Bo, C. Shi, J. Pei, Revisiting graph contrastive learning from the perspective of graph spectrum, Adv. Neural Inf. Process. Syst. 35 (2022) 2972–2983.

[35] H. Li, J. Cao, J. Zhu, Q. Luo, S. He, X. Wang, Augmentation-free graph contrastive learning of invariant-discriminative representations, IEEE Trans. Neural Netw. Learn. Syst. (2023) 1–11.

[36] X. Gong, C. Yang, C. Shi, Ma-gcl: Model augmentation tricks for graph contrastive learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, No. 4, 2023, pp. 4284–4292.

[37] R. Wang, X. Wang, C. Shi, L. Song, Uncovering the structural fairness in graph contrastive learning, Adv. Neural Inf. Process. Syst. 35 (2022) 32465–32473.

[38] X. Li, X. Zhen, X. Qi, H. Han, L. Zhang, Z. Han, Dynamic community detection based on graph convolutional networks and contrastive learning, Chaos Solitons Fractals 176 (2023) 114157.

[39] R.-C. Li, Matrix perturbation theory, in: Handbook of Linear Algebra, Chapman and Hall/CRC, 2006, pp. 1–15.

[40] B. Li, B. Jing, H. Tong, Graph communal contrastive learning, in: Proceedings of the ACM Web Conference 2022, pp. 1203–1213.

[41] X. Shen, D. Sun, S. Pan, X. Zhou, L.T. Yang, Neighbor contrastive learning on learnable graph augmentation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, No. 8, 2023, pp. 9782–9791.

[42] D. Greene, D. Doyle, P. Cunningham, Tracking the evolution of communities in dynamic social networks, in: 2010 International Conference on Advances in Social Networks Analysis and Mining, IEEE, 2010, pp. 176–183.

[43] O. Shchur, S. Günnemann, Overlapping community detection with graph neural networks, in: Deep Learning on Graphs Workshop, KDD, 2019.

[44] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proceedings of the ICLR 2014, 2014.

[45] G.B. Giannakis, Y. Shen, G.V. Karanikolas, Topology identification and learning over graphs: Accounting for nonlinearities and dynamics, Proc. IEEE (2018) 787–807.

[46] Y. Yin, Y. Zhao, H. Li, X. Dong, Multi-objective evolutionary clustering for large-scale dynamic community detection, Inform. Sci. 549 (3C5) (2020).

[47] C. Vehlow, F. Beck, P. Auwärter, D. Weiskopf, Visualizing the evolution of communities in dynamic graphs, in: Computer Graphics Forum, Vol. 34, No. 1, Wiley Online Library, 2015, pp. 277–288.

Xiaohong Li received her Master's degree from the Lanzhou University of Technology. She is an associate professor in the Northwest Normal University. Her current research interests include machine learning and intelligent information processing, with research areas focusing on community discovery, recommendation systems, and text mining.

Wanyao Shi is a graduate student at the School of Computer Science and Engineering, Northwest Normal University. Her research interests include community detection.

Qixuan Peng is a graduate student at the School of Computer Science and Engineering, Northwest Normal University. Her research interests include community detection.

Hongyan Ran, Ph.D., is employed at the College of Computer Science and Engineering, Northwest Normal University. Her main research interests include fake news detection, transfer learning, and multi-task learning.