Graph Contrastive Learning for Tracking Dynamic Communities in Temporal Networks

Yun Ai, Xianghua Xie and Xiaoke Ma*

Abstract—Temporal networks are ubiquitous because complex systems in nature and society are evolving, and tracking dynamic communities is critical for revealing the mechanism of systems. Moreover, current algorithms utilize temporal smoothness framework to balance clustering accuracy at current time and clustering drift at historical time, which are criticized for failing to characterize the temporalty of networks and determine its importance. To overcome these problems, we propose a novel algorithm by joining Non-negative matrix factorization and Contrastive learning for Dynamic Community detection (jNCDC). Specifically, jNCDC learns the features of vertices by projecting successive snapshots into a shared subspace to learn the low-dimensional representation of vertices with matrix factorization. Subsequently, it constructs an evolution graph to explicitly measure relations of vertices by representing vertices at current time with features at historical time, paving a way to characterize the dynamics of networks at the vertex-level. Finally, graph contrastive learning utilizes the roles of vertices to select positive and negative samples to further improve the quality of features. These procedures are seamlessly integrated into an overall objective function, and optimization rules are deduced. To the best of our knowledge, jNCDC is the first graph contrastive learning for dynamic community detection, that provides an alternative for the current temporal smoothness framework. Experimental results demonstrate that jNCDC is superior to the state-of-the-art approaches in terms of accuracy.

Index Terms—Dynamic community, Temporal networks, Non-negative matrix factorization, Self-representation, Graph contrastive clustering

I. Introduction

NETWORKS effectively describe, model, and analyze many complex systems from various disciplines, such as social [1], [2], [3], ecological [4], and cancer networks [5], [6], where each entity is denoted as a vertex, and an interaction is represented by an edge. The ultimate goal of network analysis is to extract potential and interesting graph patterns, that facilitate the understanding of structure and functions of the underlying systems. For example, the critical abundance thresholds in ecological networks proves that species that are most likely to be extinctive are determined by another species rather than morality rate, thereby providing solid principles for environmental protection [4].

Clusters, also called modules and communities, are a typical graph pattern (i.e., groups of vertices with the same or similar features). Great evidence demonstrate that numerous networks also present module structures (i.e., clusters are ubiquitous, which correspond to dense subgraphs). For example, persons of organizations with the same or similar aspiration are more likely to establish partnership than those with opposite ideas [7]. Clusters of networks dramatically reduce the complexity of networks because the structure and functions of whole systems can be approximately inferred from clusters. For instance, clusters in gene networks are critical pathways, that serve as bio-markers for cancer therapy [5]. Therefore, the detection of communities in networks is a prominent task in network analysis.

Community detection corresponds to the classic graph clustering problem, which attempts to identify groups of vertices with strong connectivity [8], [9], [10], [11], [12], [13], [14], [15], [16]. Based on the strategies of algorithms, current community detection methods are divided into two categories, namely, topological structure optimization [17], [9], [18], [19], [20] and feature learning based approaches [12], [21]. The former first predefines topological indexes, such as graph cut [17] and modularity [9], to quantify the connectivity of clusters. Subsequently, these indexes are optimized to determine communities. These algorithms are criticized for their sensitivity to network perturbation, whereas feature learning-based methods are devoted to obtain the low-dimensional features of vertices. Typical algorithms include matrix factorization and graph representation learning [12], [21].

However, these algorithms are designed for static networks, suggesting that the topological structure is irrelative to time. Actually, complex systems in the real world are dynamic, where the topology structure evolves time [22], [23], [24]. For example, airlines systematically schedule their flights according to the weather condition that is highly related to time, thereby resulting in temporal flight networks, which are essential for management [22]. Gene regulation networks are also evolving as cancer progresses from initial to deleterious stages. In detail, signal transduction signals are dysfunctional at the initial stages, whereas pathways (clusters of genes) fail to execute their functions at the late stages, resulting in gene deletion and recruitment during cancer progression [5], [6]. Temporal networks pave the way to track evolving

Y. Ai and X. Ma are with the School of Computer Science and Technology, Xidian University, Xi'an, Shaanxi, 710071. Y. Ai and X. Ma are also with Key Laboratory of Smart Human-Computer Interaction and Wearable Technology of Shaanxi Province, Xidian University, No.2 South Taibai Road, Xi'an Shaanxi, 710071, China. X. Xie is with Department of Computer Science, Swansea University, Swansea, UK
E-mail: xkma@xidian.edu.cn (X. Ma).

This work was supported by the National Natural Science Foundation of China (62272361), and Shaanxi Natural Science Funds for Distinguished Young Scholar (Program No. 2022JC-38).

graph patterns, particularly dynamic clusters, which are of great importance for revealing the mechanisms of systems because these patterns are much more accurate to depict structure of networks than static ones [23], [24].

Intuitively, temporal networks consist of a sequence of snapshots, where topology structure evolves, implying graph patterns simultaneously by considering the structure within each snapshot and the dynamics of subsequent snapshots. Therefore, tracking dynamic communities in temporal networks is considerably more challenging than detecting static communities, thereby presenting a significant obstacle in algorithm design [25]. Furthermore, the most critical technique is the characterization and quantification of dynamics of temporal networks, which are the foundation for tracking evolving communities. Balancing the connectivity of each snapshot (also called clustering accuracy) and the dynamics (also called clustering drift) of subsequent ones is also an urgent issue.

Thus, current algorithms address these two issues with various strategies, which are the greatest difference of methods [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38]. According to the principle for balancing clustering accuracy and drift, the existing dynamic community detection methods can be divided roughly into three categories, namely, coupling graph-, two-stage-, evolutionary clustering-based methods. Coupling graph-based methods [39] construct a single one by preserving the dynamics of the original temporal networks and performing clustering on the constructed graph. These algorithms are criticized for losing important information during preservation, resulting in low accuracy. In essence, these algorithms replace dynamic networks with static one, thereby ignoring the temporality of networks.

To allow for temporality of networks, two-stage-based methods independently address clustering accuracy and drift, where they first perform graph clustering for each snapshot and then reach the final clusters with consensus clustering [40], [41], [42]. Compared with coupling graph-based methods, these algorithms not only avoid destroying the structure of temporal networks but also facilitate the identification of dynamic clusters because any conventional graph clustering can be directly applied. However, independence of clustering accuracy and drift separates these procedures, where clusters of two subsequent snapshots are very unlikely to capture dynamics because connectivity is priori to temporalty. Hence, these algorithms have received criticism for their poor performance.

Intrinsically, the coupling graph- and two-stage-based methods fail in terms of smooth clustering accuracy and drift, whereas evolutionary clustering [26] overcomes this limitation with temporal smoothing framework (TSF). More specifically, TSF avoids separating clustering accuracy and drift by combining these two items with a weighted linear function that serves as the quantification function of dynamic communities. Many algorithms are developed under the TSF model [43], [30], [31], [44]. For example, DYNMOGA [30] sets clustering accuracy and drift as two parallel objectives and performs multi-objective optimization to identify dynamic communities. PisCES [43] utilizes matrix factorization to identify dynamic communities, where all snapshots are jointly integrated to capture the temporality of networks. In comparison to non-smoothness algorithms, evolutionary clustering greatly enhances the performance of methods for dynamic community detection, implying that it is a good balance between connectivity and dynamics.

A. Motivation and Contribution

However, many problems for dynamic community detection are unsolved. First, evolutionary clustering utilizes a linear combination of clustering accuracy and drift by assuming that dynamics are always the same. Actually, this assumption deviates from the reality because the dynamics of various time differ greatly. Thus, alternatives for TSF are needed critically. Second, the available algorithms characterize and measure the dynamics of clusters at the network or sub-network level, ignoring the vertex-level dynamics and failing to fully depict the temporality of networks. Recently, contrastive learning [45], [46], [47], [48] remarkably improves quality of features with self-supervision priori by exploiting positive and negative samples. The use of contrastive learning to measure dynamics of temporal networks is not attempted, which is one of the major motivation of this study.

To tackle these problems, a novel joint learning algorithm called jNCDC for tracking dynamic community in temporal networks is proposed. This algorithm integrates feature learning, evolution graph construction, and graph contrastive learning (Fig. 2). jNCDC utilizes non-negative matrix factorization (NMF) to learn the features of vertices for successive snapshots within a window. To avoid to balance clustering accuracy and drift, jNCDC employs self-representation learning to construct an evolution graph for each time, where vertices at current time are represented by features at historical time. In this case, the dynamics of networks are characterized at the vertex-level, providing a more precise way to quantify the temporality of networks. Graph contrastive learning enhances the quality of features by selecting positive and negative samples from the constructed evolution graph, thereby improving the performance of algorithms for the clustering of temporal networks. Experimental results show that jNCDC outperforms state-of-the-art algorithms, indicating that graph contrastive learning is also promising for tracking dynamic communities.

In summary, the main contributions of this study can be summarized as follows:

- A novel strategy that represents vertices at current time by using features at the previous time to characterize and quantify temporality of networks is proposed. The evolution graph for each time is constructed, paving a way to quantify the dynamics of networks at the vertex-level.
- Graph contrastive learning for temporal networks is proposed. It improves the quality of features by

discriminating dynamic and static vertices. As far as we know, it is the first attempt to measure the dynamics of temporal networks with contrastive learning, serving as a flexible framework for graph contrastive learning.
- A joint learning algorithm called jNCDC is proposed. This algorithm integrates feature learning, evolution graph construction, and contrastive learning into an optimization problem. The experimental results show that it outperforms state-of-the-art baselines in terms of accuracy.

This paper is structured as follows. Section 2 provides an overview of the related literature, while Section 3 outlines the preliminary concepts. Sections 4 and 5 present the algorithmic procedure and its results, respectively. Lastly, Section 6 concludes the study.

II. Related work

Temporal networks consist of multiple snapshots, where dynamic community detection simultaneously addresses clustering accuracy at the current time (denoted as CS) and clustering drift at the historical time (denoted as CT). A large number of algorithms address these two issues with various principles that are loosely grouped into two main categories, namely, independence strategies [49], [50], [40], [42], [40], [51], [52] and temporal smoothing-based methods [26], [30], [32], [33], [34], [43], [53], [54], [55].

A. Progression of algorithms with independent strategies

These independence-based algorithms simply extend static community detection approaches, which are divided into two classes, namely, one-stage- and two-stage-based methods. The former first transform temporal networks into a single coupling graph by adding time labels on edges and then performing graph clustering on the constructed network with single-layer clustering approaches [56], [57], [58], [59], [60]. HOP-NMF[59] adopts an iterative network enhancement scheme to encode higher-order proximity into the network, and then utilizes symmetric non-negative matrix decomposition of the network to obtain the final community structure. JGSED[60]joint graph construction, spectral embedding and spectral rotation to learn the binary clustering indicator matrix to get community structure. Actually, these algorithms discard temporality to fit the conventional graph clustering, thereby reducing complexity by sacrificing the performance of methods because coupling graph fails to preserve the temporality of communities. As such, these methods are criticized for their poor preformance in terms of detecting dynamic communities.

To avoid destroying network structures, two-stage-based methods first performs static community detection independently for each time and then address the evolution of communities at the subsequent snapshots. These algorithms differ greatly in the principles of addressing evolution. For example, DYNAMO [42] employs adaptive and incremental learning to detect dynamic communities, whereas tNodeEmbed [51] utilizes long short-term memory (LSTM) to learn temporality from the static embedding of vertices. ePMCL[34] detects dynamic communities based on genetic algorithm adaptive search for optimal parameter combinations. Even though these algorithms overcome the limitation of coupling graph-based methods, they fail to enhance the performance of detecting dynamic communities because of the independence of clustering accuracy and drift, where clusters are extracted for each time by solely optimizing connectivity without considering temporality, causing evolution of clusters at the continuous time can to not be observed.

B. Progression of evolutionary clustering algorithms

To avoid the separation of CS and CT, temporal smoothing-based methods balance these two issues (i.e., temporality is incorporated into clustering). To balance CS and CT, evolutionary clustering [26] employs TSF to detect dynamic communities at each time via a weighted linear function as

$$Cost = \theta CS + (1-\theta) CT \tag{1}$$

where CS and CT are clustering accuracy and drift respectively, and parameter $\theta \in [0,1]$. Notice that Eq.(1)is the traditional graph clustering for static networks if θ=1.

According to the principles of balancing strategies, current algorithms are further grouped into two categories, namely, global smoothing- [43], [53], [61] and local smoothing-based approaches [26], [30], [31], [32], [33], [34], [35], [36], [37], [38]. The difference between these two classes lies in window size for historical snapshots for temporality. Specifically, the global smoothing-based methods employ all snapshots to capture clustering drift, whereas the local ones only uses a small-size window of historical snapshots. The typical global smoothing method is PisCES [43], where all snapshots are jointly factorized to measure temporality. It remarkably improves the performance of detecting dynamic communities because the global strategy provides a better way to model and characterize clustering drift. However, these algorithms are time-consuming because all snapshots are involved, thereby making its application for large-scale networks.

To address time issue, the local smoothing-based methods attack cluster drift with a few of subsequent historical snapshots rather than all ones. Evolutionary clustering [26] proposes TSF to balance CS and CT through a weighted linear function. The difference among them lies in how to characterize and model the dynamics of snapshots within a window. For example, DYNMOGA [30] poses these two issues as two competitive objectives and formulate the dynamic community detection problem as a multi-objective optimization one. sE-NMF [36] proves the equivalence of evolutionary clustering algorithms and proposes semi-supervised evolutionary non-ngeative matrix factorization for dynamic community detection. C-Blondel [38] derives knowledge of the historical snapshots for clustering drift, while jLDEC [33] learns graph representation and detects dynamic community. ERCOT[62] describes clustering drift by quantifying the importance of historical data to the current clustering structure, while GCIA[63] aggregates information from historical timestamp based on a gaming strategy. DMOPs[64] designs a higher-order knowledge transfer strategy to capture dynamic communities, while the constraint interval graph-based approach[65] maintains the structural and temporal information of the network by constructing a constraint interval graph. VGRGMM[66] constructs gated recurrent unit to capture dependencies between vertices in the embedding space and combines it with variational autoencoder (VAE) to simultaneously learn dynamic network embeddings and community affiliations. These algorithms not only reduce running time but also achieve excellent performance, demonstrating that local smoothness is also promising for dynamic community detection. However, local smoothing-based algorithms cannot make use the global structure of temporal networks, thereby failing to fully capture the dynamics of networks.

C. Limitations of evolutionary clustering

CS can be modeled in many different ways, such as normalized graph cut [17] and modularity [9]. The most critical technique is to characterize CT, and current algorithms under TSF address it by exploiting difference in terms of features or networks. However, several typical limitations for TSF in Eq.(1) are summarized as

- First, parameter θ is difficult to determine. Current algorithms fix the value of parameter θ in advance for all time, assuming that dynamics of networks at each time is equal, thereby deviating from reality of temporal networks. Thus, parameter θ must reflect the dynamics of temporal networks for each time.
- Second, CT is characterized at the global level, where dynamics at the vertex-level are neglected. Actually, the evolution of temporal networks usually occurs at the vertex-level, therefore, failing to fully model and capture the dynamics of networks.
- Finally, to the best of our knowledge, no previous research has focused on utilizing the partial structural information of temporal networks to detect communities. As in fact, evolution among subsequent snapshots implies the existence of preserved structure of networks, which can serve as semi-supervised information to enhance the performance of algorithms for clustering of temporal networks.

In this study, we design a novel method for detecting dynamic community in temporal networks, to provide an alternative for TSF. Furthermore, the self-supervised strategy is developed to enhance the performance of algorithms by exploiting partial information with contrastive learning.

III. Preliminaries

A. Notations

For sake of convince, scalars, vectors, and matrices are represented by lower-case, bold lower-case, and capital

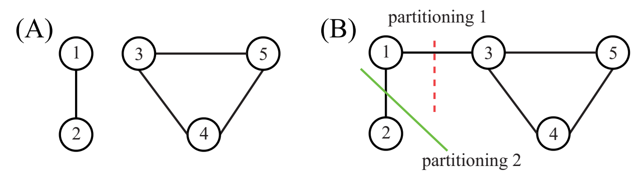


Fig. 1: Schematic example of dynamic communities: (A) visualization of $G^{[t-1]}$, and (B) two partitioning in $G^{[t]}$ with the dashed and solid lines, respectively.

letters, respectively. Graph with n vertices (i.e., $V = \{v_1, \ldots, v_n\}$), is denoted as $G = (V, E)$ with edge set $E = \{(v_i, v_j)|v_i, v_j \in V\}$. $W = (w_{ij})$ is the adjacent matrix of G and element w_{ij} is the weight on edge (v_i, v_j). The degree of vertex v_i is defined as $d_i = \sum_j w_{ij}$, and $D = diag(d_1, \ldots, d_n)$. The Laplacian matrix of G is defined as $L = D - W$. Let $\|W\| = \sqrt{\sum_{ij} w_{ij}^2}$ and W' be the Frobenius norm and transpose of matrix W. Let $W_{i.}(\mathrm{w}_{i.})$ and $W_{.j}(\mathrm{w}_{.j})$ be the i-th row and j-th column, respectively. $Tr(W) = \sum_i w_{ii}$ is trace of matrix W. Temporal networks consist of τ snapshots, denoted as $\mathcal{G} = \{G^{[1]}, \ldots, G^{[\tau]}\}$, where $G^{[t]} = (V, E^{[t]})$ is the t-th snapshot, and $G^{[t]}$ is derived from $G^{[t-1]}$. The adjacent matrix of $\mathcal{G}$ is $\mathcal{W} = \{W^{[1]}, \ldots, W^{[\tau]}\}$.

Clustering of G divides V into groups with strong connectivity inside and weak connectivity outside of groups. In other words, communities $\{C_i\}_{i=1}^k$ such that $V = \bigcup C_i$, and $C_i \bigcap C_j = \emptyset$ for $(i \neq j)$, where C_i is the i-th community and k is the number of communities. $\{C_i\}_{i=1}^k$ can be represented with an index matrix $H \in R^{n \times k}$, where h_{ij}=1 if $v_i \in C_j$, 0 otherwise. Dynamic communities at time t of $\mathcal{G}$ are denoted as $\{C_i^{[t]}\}_{i=1}^{k^{[t]}}$, where $C_i^{[t]}$ simultaneously reflects topological structure of $G^{[t]}$ and $G^{[t-1]}$. More specifically, connectivity of $C_i^{[t]}$ is strong in $G^{[t]}$ and $G^{[t-1]}$, where clustering accuracy reflects $G^{[t]}$, and clustering drift addresses temporality of $G^{[t-1]}$. A schematic dynamic example is shown in Fig. 1, where panel A is the visualization of $G^{[t]}$ with 5 vertices, and B contains two partitioning of $G^{[t]}$ with different lines. Specifically, partitioning 1 corresponds two communities $\{1, 2\}$ and $\{3, 4, 5\}$, and partitioning 2 also has two communities $\{1, 2, 3\}$ and $\{4, 5\}$. These two partitioning are equal in terms of clustering accuracy at $G^{[t]}$ because there is only one edge across these two communities. However, partitioning 1 is better than partitioning 2 because vertices in community $2, 3, 4, 5$ are disconnected in $G^{[t-1]}$, implying that it fails to preserve the topology structure of historical snapshot.

The main symbols are listed in Table I.

B. NMF and contrastive learning

Non-negative matrix factorization (NMF) [67] learns the partial representation of the original data with two low-rank matrices with non-negative constraint. Specifically, it approximates matrix W by the product matrix B and

TABLE I: Symbol description

Symbol	Description
$\mathcal{G}$	Temporal network $\{G^{[1]},\ldots,G^{[\tau]}\}$
$\mathcal{W}$	Adjacent matrix of $\mathcal{G}$
$\mathcal{L}$	Laplacian matrix of $\mathcal{G}$
$G^{[t]}=(V,E^{[t]})$	Snapshot at time t
$M^{[t]}$	PMI matrix of $G^{[t]}$
$\{C_i^{[t]}\}_{i=1}^{k^{[t]}}$	Dynamic communities at time t
$B^{[t]}$	Basis matrix of $G^{[t]}$
$F^{[t]}$	Coefficient matrix of $G^{[t]}$
$\mathbb{N}^{[t][s]}$	Set of static vertices of $G^{[t]}$
$\mathbb{N}^{[t][d]}$	Set of dynamic vertices of $G^{[t]}$
$\mathbb{N}_i^{[t]}$	Positive samples of v_i at time t
$\mathrm{w}_{i.}$	The i-th row of matrix W
$\mathrm{w}_{.j}$	The j-th column of matrix W
$k^{[t]}$	Number of communities at time t
τ	Number of time steps

F as

$$W \approx BF, \quad s.t. \quad B \geq 0, F \geq 0, \tag{2}$$

where B and F are the basis and feature matrix respectively. Eq.(2) is solved by minimizing the reconstruction error, i.e.,

$$\mathcal{O} = \|W - BF\|^2. \tag{3}$$

The ultimate goal of contrastive learning [45], [46], [47] is to enhance the quality of features with the partial information. Specifically, it deliberately selects both positive and negative samples, optimizing measurements to discriminate negative samples in the feature space [68], [69], [70]. In addition, graph contrastive learning [48] narrows distance between similar vertices (positive samples) and increases that dissimilar ones (negative samples), i.e.,

$$\mathcal{J} = \sum_{i=1}\sum_{j\in\mathbb{N}_i} -\log\frac{\exp(s_{ij})}{\sum_{p\neq i}\exp(s_{ip})}, \tag{4}$$

where $S \in \mathbb{R}^{n\times n}$ is the similarity matrix for vertices in G, and $\mathbb{N}_i$ represents neighbors of vertex v_i.

IV. Algorithm

We first formulate the objective function, then derive the optimization of the proposed algorithm. Finally, we perform algorithm analysis in this section.

A. Objective function

As shown in Fig. 2, jNCDC consists of four major components (i.e., feature learning, evolution graph construction, graph contrastive learning, and clustering). Therefore, the objective function of jNCDC is composed three costs, corresponding to the first three components.

1) Feature learning: On the feature learning issue, NMF [67] is widely adopted to factorize matrix $W^{[t]}$ of $G^{[t]}$ into two non-negative matrices $B^{[t]}$ and $F^{[t]}$ by minimizing approximation, i.e.,

$$\mathcal{O}(G^{[t]}) = \|W^{[t]} - B^{[t]}F^{[t]}\|^2. \tag{5}$$

However, Eq.(5) has two limitations. First, $W^{[t]}$ just depicts the 1-order topological structure of vertices without exploiting high-order interactions of vertices. Second, it ignores the temporality of snapshot at time t. Recently, evidence proves that point mutual information (PMI) matrix outlines the high-order structure, which offers a superior method for characterizing the structures of networks [71]. Given snapshot $G^{[t]}$, the element $m_{ij}^{[t]}$ of PMI matrix $M^{[t]}$ is defined as

$$m_{ij}^{[t]} = \max\{\log w_{ij}^{[t]}\sum_i d_i^{[t]} - \log(d_i^{[t]}d_j^{[t]} - \kappa), 0\}, \tag{6}$$

where κ is a hyper-parameter controlling sizes of negative sampling (usually, κ=2 [72]). By replacing $W^{[t]}$ with $M^{[t]}$, Eq.(5) is reformulated as

$$\mathcal{O}(G^{[t]}) = \|M^{[t]} - B^{[t]}F^{[t]}\|^2. \tag{7}$$

However, Eq.(7) ignores the temporality of snapshots within the window around time t, thereby feature $F^{[t]}$ fails to capture the dynamics of networks. This problem can be effectively solved by factorizing matrices of subsequent snapshots. In this case, Eq.(7) is re-written as

$$\mathcal{O}(\{G^{[l]}\}_{l=t-1}^{t+1}) = \sum_{l=t-1}^{t+1}\|M^{[l]} - B^{[l]}F^{[l]}\|^2. \tag{8}$$

Our previous study [36] demonstrates that joint factorization is more precise to model the dynamics of networks by projecting snapshots $G^{[l]}(l=t-1,t,t+1)$ into a subspace. Here, we adopt the same strategy, and Eq.(8) is modified as

$$\mathcal{O}(\{G^{[l]}\}_{l=t-1}^{t+1}) = \sum_{l=t-1}^{t+1}\|M^{[l]} - B^{[t]}F^{[l]}\|^2. \tag{9}$$

Furthermore, we anticipate that the vertex feature in $F^{[t]}$ will maintain the local topological structure of $G^{[t]}$. In other words, vertex v_i and v_j are well connected in $G^{[t]}$, they are also close to each in the feature space (i.e., Euclidean distance of $\mathrm{f}_{i.}^{[t]}$ and $\mathrm{f}_{j.}^{[t]}$ is small), vice versa. And, it can be formulated as trace optimization as [73]

$$\mathcal{O}(F^{[t]}) = \sum_{i,j} w_{ij}^{[t]}\|\mathrm{f}_{i.}^{[t]} - \mathrm{f}_{j.}^{[t]}\|^2 = Tr((F^{[t]})L^{[t]}(F^{[t]})'). \tag{10}$$

2) Evolution graph construction: To model and quantify the dynamics of temporal networks, current evolutionary clustering algorithms [36], [26] measure CT by comparing difference features at various time as

$$\|(F^{[t]})'F^{[t]} - (F^{[t-1]})'F^{[t-1]}\|^2. \tag{11}$$

But, Eq.(11) fails to fully measure the dynamics of temporal networks for two reasons. First, it quantifies the dynamics at the global level, rather than at the vertex-level. In other words, Eq.(11) is unable to differentiate

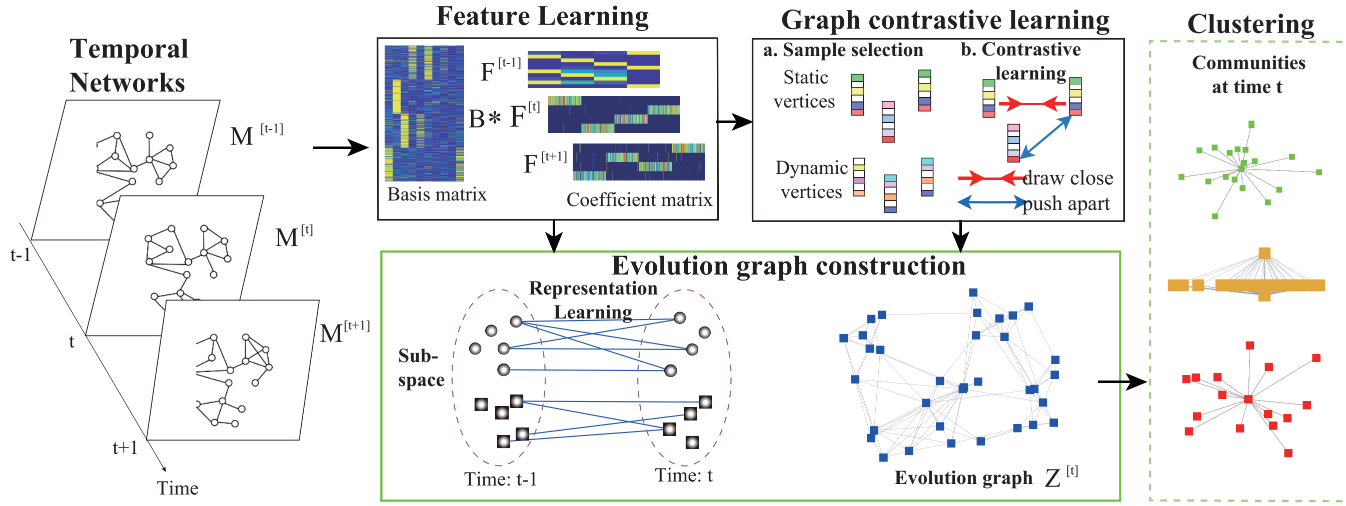


Fig. 2: Overview of the proposed algorithm. It consists of feature learning, evolution graph construction, graph contrastive learning and clustering, where the first procedure learns features of vertices at time t with joint NMF, evolution graph construction procedure represents vertices at time t by using features at time $t-1$ with representation learning, graph contrastive learning is performed on the static and dynamic vertices to improve the quality of features and evolution graph. Finally, clustering analysis is performed on the evolution graph to obtain communities at time t.

between slight perturbation occurring at a large subnetwork versus intense evolution at a small group of vertices, thus hindering downstream analysis. Second, the interpretability of dynamics captured by Eq.(11) is relatively weak.

To address the above problems, jNCDC measures the dynamics of networks at the vertex-level by exploiting relations among successive features. More specifically, because $G^{[t]}$ is smoothly evolved from $G^{[t-1]}$, there is a close relation between $F^{[t]}$ and $F^{[t-1]}$. Thus, jNCDC utilizes self-representation learning to construct an affinity graph with $F^{[t-1]}$ and $F^{[t]}$ as with as

$$\mathcal{O}(Z^{[t]}) = \|F^{[t]} - F^{[t-1]}Z^{[t]}\|^2. \tag{12}$$

$Z^{[t]}$ is the affinity graph representing the relations of vertices from $F^{[t-1]}$ and $F^{[t]}$, where $z_{ij}^{[t]}$ denotes the weight of v_j in $G^{[t-1]}$ to represent v_i in $G^{[t]}$. Furthermore, $z_{ij}^{[t]}$ is considered the similarity between $\mathrm{f}_{.j}^{[t-1]}$ and $\mathrm{f}_{.i}^{[t]}$. In this case, the evolution of vertices can be reflected from $Z^{[t]}$. Small $z_{ii}^{[t]}$ implies that v_i in $G^{[t]}$ cannot be directly obtain from v_i in $G^{[t-1]}$ (i.e., v_i at time t is dynamic), otherwise static.

What we want to point out is that $Z^{[t]}$ brings in two advantages. First, the relations of vertices from successive features are explicitly quantified, which ensures the characterization of network dynamics at the micro-level, thereby providing a better way to model and depict the temporality of networks. Furthermore, for every vertex in $G^{[t]}$, $Z^{[t]}$ identifies the closely related vertices in $G^{[t-1]}$, thereby enhancing the explanation of network dynamics.

3) Graph contrastive learning: Contrastive learning improves the quality of features by exploiting positive and negative samples [45], [46], [47], and we also want utilizes it to capture dynamics of networks (i.e., $Z^{[t]}$). There are two critical techniques involved: selecting positive and negative vertices, and improving the features with partial information.

On the vertex selection concerning, our previous research [55] illustrates that the role of vertices facilitates feature learning in temporal networks. Analogously, vertices for each time t are divided into two classes (i.e., dynamic and static one). In details, the dynamics of vertex v_i at time t is defined as the sum of difference of weights on edges connecting to it, $\Delta_i^{[t]} = \sum_j |w_{ij}^{[t]} - w_{ij}^{[t-1]}|$. Top (bottom) μ% of vertices are selected as dynamic (static) ones (according to Ref.[55], μ=5 is a good choice).

On the feature improvement concerning, jNCDC expects that static vertices preserve the features at successive time (i.e, $\mathrm{f}_{.i}^{[t-1]}$ and $\mathrm{f}_{.i}^{[t]}$ are similar), which can be fulfilled by maximizing $z_{ii}^{[t]}$ with the loss function of contrastive learning for static vertex v_i as [46]

$$\mathcal{O}(z_{ii}^{[t]}) = -\log \frac{\exp(z_{ii}^{[t]})}{\sum_p \exp(z_{ip}^{[t]})}. \tag{13}$$

Furthermore, $Z^{[t]}$ also characterizes the relations among vertices. For each static vertex v_i, we can select δ closest ones as positive samples (denoted as $\mathbb{N}_i^{[t]}$) in terms of values in $\mathrm{z}_{i.}^{[t]}$. Similarly, the loss function is formulated as

$$\mathcal{O}(v_i) = \sum_{j \in \mathbb{N}_i^{[t]}} -\log \frac{\exp(z_{ij}^{[t]})}{\sum_{p \neq i} \exp(z_{ip}^{[t]})}. \tag{14}$$

By combining Eq.(13) and Eq.(14), the loss function for static vertices at time t (denoted as $\mathbb{N}^{[t][s]}$) is formulated as

$$\mathcal{O}(\mathbb{N}^{[t][s]}) = \sum_{i \in \mathbb{N}^{[t][s]}} (\mathcal{O}(z_{ii}^{[t]}) + \mathcal{O}(v_i)). \tag{15}$$

Different from static vertices, for each vertex v_i in dynamic set $\mathbb{N}^{[t][d]}$, it fails to be represented by $\mathrm{f}_{.i}^{[t-1]}$ (i.e., $z_{ii}^{[t]}$ is close to 0). Therefore, we select δ closest ones in $G^{[t]}$ as positive samples, denoted as $\mathbb{N}_i^{[t]}$, and others as negative samples. The loss function for $\mathbb{N}^{[t][d]}$ is formulated as

$$\mathcal{O}(\mathbb{N}^{[t][d]}) = \sum_{i\in\mathbb{N}^{[t][d]}} \sum_{j\in\mathbb{N}_i^{[t]}} -\log \frac{\exp(z_{ij}^{[t]})}{\sum_p \exp(z_{ip}^{[t]})}. \tag{16}$$

By combining Eq.(15) and (16), the loss of graph contrastive learning is written as

$$\mathcal{O}_{gc}(G^{[t]}) = \mathcal{O}(\mathbb{N}^{[t][s]}) + \mathcal{O}(\mathbb{N}^{[t][d]}). \tag{17}$$

Eq.(17) consequently improves the quality of features by discriminating positive and negative vertices, which narrows down the distance of similar vertices and increases the distance of dissimilar vertices.

By combining Eqs.(9, 10, 12, 17), the objective function of jNCDC is formulated as

$$\begin{aligned}
\mathcal{O} &= \mathcal{O}(\{G^{[l]}\}_{l=t-1}^{t+1}) + \mathcal{O}(F^{[t]}) + \alpha\mathcal{O}(Z^{[t]}) + \beta\mathcal{O}_{gc}(G^{[t]}) \\
&= \sum_{l=t-1}^{t+1} \|M^{[l]} - B^{[t]}F^{[l]}\|^2 + Tr(F^{[t]}L^{[t]}(F^{[t]})') \\
&+ \alpha\|F^{[t]} - F^{[t-1]}Z^{[t]}\|^2 \\
&+ \beta\Big(\sum_{i\in\mathbb{N}^{[t][s]}} \Big(-\log \frac{\exp(z_{ii}^{[t]})}{\sum_p \exp(z_{ip}^{[t]})} \\
&+ \sum_{j\in\mathbb{N}_i^{[t]}} -\log \frac{\exp(z_{ij}^{[t]})}{\sum_{p\neq i} \exp(z_{ip}^{[t]})}\Big) \\
&+ \sum_{i\in\mathbb{N}^{[t][d]}} \sum_{j\in\mathbb{N}_i^{[t]}} -\log \frac{\exp(z_{ij}^{[t]})}{\sum_p \exp(z_{ip}^{[t]})}\Big) \\
&\quad \text{s.t. } B^{[t]} \geq 0, \quad F^{[t]} \geq 0, \quad Z^{[t]} \geq 0, \quad B^{[t]}(B^{[t]})' = I.
\end{aligned} \tag{18}$$

where α and β determines the importance of evolution graph construction and contrastive learning.

There are at least three difference between the proposed algorithm and current algorithms, which are summarized as

- Current algorithms measure the dynamics of temporal networks by comparing either features or topological structure of vertices at successive time, whereas jNCDC automatically learns evolution graph.
- Graph contrastive learning for clustering of temporal networks is first proposed, where the partial information of dynamic and static vertices is fully exploited to characterize and measure the dynamics of networks.
- Available methods utilizes TSF to balance CS and CT, while jNCDC avoids it with joint learning framework.

B. Optimization

The objective function in Eq.(18) is non-convex because of the graph contrastive learning item, which cannot be directly optimized. Thus, an iterative approach is utilized where one variable is optimized while keeping the others fixed. This process continues until convergence or the maximum number of iterations is reached.

1) Updating $B^{[t]}$: By fixing other variables, Eq.(18) in terms of $B_{[t]}$ is transformed as

$$\mathcal{O} = \sum_{l=t-1}^{t+1} \|M^{[l]} - B^{[t]}F^{[l]}\|^2. \tag{19}$$

Since Eq.(19) is convex, it has analytical solution. The partial derivative of $\mathcal{O}$ in terms of $B^{[t]}$ is formulated as

$$\frac{\partial\mathcal{O}}{\partial B^{[t]}} = 2\sum_{l=t-1}^{t+1} \|M^{[l]} - B^{[t]}F^{[l]}\|. \tag{20}$$

By setting $\frac{\partial\mathcal{O}}{\partial B^{[t]}}$=0, Its update rule is formulated as

$$B^{[t]} = \frac{\sum_{l=t-1}^{t+1} M^{[l]}}{\sum_{l=t-1}^{t+1} F^{[l]}}. \tag{21}$$

2) Updating $F^{[t]}$: By fixing other variables and removing irrelevant terms to $F^{[t]}$, Eq.(18) is rewritten as

$$\begin{aligned}
\mathcal{O} = \sum_{l=t-1}^{t+1} \|M^{[l]} - B^{[t]}F^{[l]}\|^2 + Tr(F^{[t]}L^{[t]}(F^{[t]})') \\
+ \alpha\|F^{[t]} - F^{[t-1]}Z^{[t]}\|^2.
\end{aligned} \tag{22}$$

The Lagrangian function of Eq.(22) is formulated as

$$\begin{aligned}
\mathcal{O} =& \|M^{[t]} - B^{[t]}F^{[t]}\|^2 + Tr(F^{[t]}L^{[t]}(F^{[t]})') \\
&+ \alpha\|F^{[t]} - F^{[t-1]}Z^{[t]}\|^2 + Tr(\Psi F^{[t]}),
\end{aligned} \tag{23}$$

where Ψ is the Lagrange multiplier for non-negativity of $F^{[t]}$.

The partial derivative of Eq.(23) in terms of $F^{[t]}$ is deduced as

$$\begin{aligned}
\frac{\partial\mathcal{O}}{\partial F^{[t]}} =& (B^{[t]})'B^{[t]}F^{[t]} - (B^{[t]})'M^{[t]} \\
&+ F^{[t]}L^{[t]} + \alpha(F^{[t]} - F^{[t-1]}Z^{[t]}).
\end{aligned} \tag{24}$$

In accordance with the KKT conditions(Karush-Kuhn-Tucher), by setting the partial derivative of Eq.(24) to 0, the rules of $F^{[t]}$ is obtained as

$$F^{[t]} = F^{[t]} \odot \frac{(B^{[t]})'M^{[t]} + \alpha F^{[t-1]}Z^{[t]} + F^{[t]}W^{[t]}}{(B^{[t]})'B^{[t]}F^{[t]} + F^{[t]}D^{[t]} + \alpha F^{[t]}}. \tag{25}$$

3) Updating $Z^{[t]}$: Matrix $Z^{[t]}$ is involved in representation learning and contrastive learning, and Eq.(18) is deduced as

$$\begin{aligned}
\mathcal{O} =& \alpha\|F^{[t]} - F^{[t-1]}Z^{[t]}\|^2 + \beta\Big(\sum_{i\in\mathbb{N}^{[t][s]}} \Big(-\log \frac{\exp(z_{ii}^{[t]})}{\sum_p \exp(z_{ip}^{[t]})} \\
&+ \sum_{j\in\mathbb{N}_i^{[t]}} -\log \frac{\exp(z_{ij}^{[t]})}{\sum_{p\neq i} \exp(z_{ip}^{[t]})}\Big) \\
&+ \sum_{i\in\mathbb{N}^{[t][d]}} \sum_{j\in\mathbb{N}_i^{[t]}} -\log \frac{\exp(z_{ij}^{[t]})}{\sum_p \exp(z_{ip}^{[t]})}\Big).
\end{aligned}$$

$$(26)$$

According to the linear additive property of the derivative, $\partial_{Z^{[t]}}(\mathcal{O})$ consists of two terms, i.e.,

$$\nabla_{Z^{[t]}} = \alpha \nabla_{rs} + \beta \nabla_{gc} \tag{27}$$

where ∇_{rs} and ∇_{gc} are the partial derivatives for representation learning and contrastive learning, respectively. ∇_{rs} is formulated as

$$\nabla_{rs} = -(F^{[t-1]})' F^{[t]} + (F^{[t-1]})' F^{[t-1]} Z^{[t]}. \tag{28}$$

However, ∇_{gc} involves dynamic and static vertices, which can be handled separately. For each static vertex $v_i \in \mathbb{N}^{[t][s]}$, if vertex $v_j \in \mathbb{N}_i^{[t]}$, the second item in Eq.(26) is re-written as

$$\begin{aligned}\mathcal{L} &= -\log \frac{\exp(z_{ii}^{[t]})}{\sum_p \exp(z_{ip}^{[t]})} + \sum_{j \in \mathbb{N}_i^{[t]}} -\log \frac{\exp(z_{ij}^{[t]})}{\sum_{p \neq i} \exp(z_{ip}^{[t]})} \\ &= -z_{ii}^{[t]} + log(\sum_p \exp(z_{ip}^{[t]}) + \sum_{j \in \mathbb{N}_i^{[t]}} (-z_{ij}^{[t]} + \log(\sum_{p \neq i} \exp(z_{ip}^{[t]}))).\end{aligned} \tag{29}$$

The partial derivative of $z_{ij}^{[t]}$ is deduced as

$$\frac{\partial \mathcal{O}}{\partial z_{ij}^{[t]}} = -1 + \frac{m \exp(z_{ij}^{[t]})}{\sum_{p \neq i} \exp(z_{ip}^{[t]})} + \frac{m \exp(z_{ij}^{[t]})}{\sum_p \exp(z_{ip}^{[t]})}, \tag{30}$$

where m is the number of positive vertices.

If vertex $v_j \notin \mathbb{N}_i^{[t]}$, the second item of Eq.(26) is re-written as

$$\begin{aligned}\mathcal{O} &= -z_{ii}^{[t]} + log(\sum_p \exp(z_{ip}^{[t]})) + \sum_{j \in \mathbb{N}_i^{[t]}} (\log(\sum_{p \neq i} \exp(z_{ip}^{[t]}))) \\ &= -z_{ii}^{[t]} + log(\sum_{j \in \mathbb{N}_i^{[t]}} \exp(z_{ij}^{[t]}) + \sum_{p \notin \mathbb{N}_i^{[t]}} \exp(z_{ip}^{[t]})) \\ &+ \sum_{j \in \mathbb{N}_i^{[t]}} (\log(\sum_{j \neq i, j \in \mathbb{N}_i^{[t]}} \exp(z_{ij}^{[t]}) + \sum_{p \neq i, p \notin \mathbb{N}_i^{[t]}} \exp(z_{ip}^{[t]}))).\end{aligned} \tag{31}$$

The partial derivative of $z_{ij}^{[t]}$ is formulated as

$$\frac{\partial \mathcal{L}}{\partial z_{ij}^{[t]}} = \frac{m \exp(z_{ij}^{[t]})}{\sum_{p \neq i} \exp(z_{ip}^{[t]})} + \frac{m \exp(z_{ij}^{[t]})}{\sum_p \exp(z_{ip}^{[t]})}. \tag{32}$$

And, the update rule for $z_{ij}^{[t]}$ is deduced as

$$\begin{cases} -1 + \frac{m \exp(z_{ij}^{[t]})}{\sum_{p \neq i} \exp(z_{ip}^{[t]})} + \frac{m \exp(z_{ij}^{[t]})}{\sum_p \exp(z_{ip}^{[t]})}, & \text{if } j \in \mathbb{N}_i^{[t]} \\ \frac{m \exp(z_{ij}^{[t]})}{\sum_{p \neq i} \exp(z_{ip}^{[t]})} + \frac{m \exp(z_{ij}^{[t]})}{\sum_p \exp(z_{ip}^{[t]})}, & \text{otherwise.} \end{cases} \tag{33}$$

Analogously, the update rule for dynamic vertex v_i is obtained as

$$\begin{cases} -1 + \frac{m \exp(z_{ij}^{[t]})}{\sum_p \exp(z_{ip}^{[t]})}, & \text{if } j \in \mathbb{N}_i^{[t]}, \\ \frac{m \exp(z_{ij}^{[t]})}{\sum_p \exp(z_{ip}^{[t]})}, & \text{otherwise.} \end{cases} \tag{34}$$

TABLE II: Statistic of temporal networks.

	Data	$\lvert V \rvert$	$\lvert E \rvert$	τ
Artificial Temporal Networks	SYN-FIX	128	20,480	10
	SYN-VAR	256	59,256	10
	Birthdeath	10,000	18,970,847	10
	Expansion	10,000	24,574,662	10
	Hide	10,000	24,190,738	10
	Mergesplit	10,000	23,854,220	10
Real-world Temporal Networks	Cellphone	400	10,400	10
	Email	1,005	332,334	10
	Wikipedia	8,400	162,000	10
	Dublin	24,818	415,900	4

The procedure of jNCDC is illustrated in Algorithm 1.

Algorithm 1 The jNCDC algorithm

Require:
- $\mathcal{G}$: Temporal networks
- α, β: Parameter for regularization items;

Ensure:
- $\{C_i^{[t]}\}_{i=1}^{k^{[t]}}$: Dynamic communities;

1. Construct PMI matrix $M^{[t]}$ for each time t;
2. Update $B^{[t]}$ according to Eq.(21);
3. Update $F^{[t]}$ according to Eq.(25;
4. Update $Z^{[t]}$ according to Eq.(33) and Eq.(34);
5. Go to step 2 until convergence;
6. Performing spectral clustering on $\frac{Z^{[t]} + (Z^{[t]})'}{2}$.
7.return $\{C_i^{[t]}\}_{i=1}^{k^{[t]}}$.

C. Algorithm analysis

On the space complexity, the space for adjacency matrix of $\mathcal{G}$ is $O(n^2\tau)$. The space for basis matrix $B^{[t]}$ and feature matrix $F^{[t]}$ is $O(nd\tau)$, where d is the number of dimension. The space for affinity matrix $Z^{[t]}$ is $O(n^2\tau)$. PMI matrix and Laplacian matrix take space $O(n^2\tau)$. Thus, the total space complexity for jNCDC is $O(n^2\tau)$ because of $d \ll n$, demonstrating that the proposed method is efficient in terms of space complexity. On the time complexity, matrix factorization requires time $O(n^2dl)$, where l is the number of iterations. The time for updating matrix $Z^{[t]}$ is $O(n^2l)$. The time for spectral clustering is $O(n^3)$. Thus, the time complexity of jNCDC is $O(n^3\tau)$.

V. Experiments

Parameter effect, accuracy, and ablation analysis of the proposed algorithm are investigated to fully validate the performance of jNCDC on the clustering of temporal networks.

A. Settings

A total of 10 benchmark temporal networks, comprising 6 artificial and 4 real world datasets, are selected for experiments. SYN-FIX/SYN-VAR originated from the GN network [7] by incorporating dynamics, where the number communities in SYN-FIX networks for each time is fixed but varies for SYN-VAR. Greene dataset [74] contains 4 evolution events (i.e., Birthdeath, Hide, Expansion, and Mergesplit), where the dynamics of networks is complicated.

In addition, 4 real-world temporal networks are included. Cellphone[1] consists of records from the 400 members of the fictitious Paraiso movement covering a period of 10 days in June 2006, where each member is treated as a node, the call records between members as an edge, and every day corresponds to a snapshot. Email[2] comprises emails among 1005 persons from an institution, which contains 1,005 vertices, 332,334 edges, and 10 time. Wikipedia[3] contains 8,400 nodes, 162,000 edges, and 10 time, Dublin[4] is a social network for communication among individuals, which has 24,818 vertices, more than 400,000 edges, and 4 time. The statistics of networks are summarized in Table II.

Normalized mutual information (NMI) [75] and accuracy (ACC) are selected as measurements to quantify the performance of algorithms. Eight MetaFac [27], PisCES [43], sE-NMF [36], DYNMOGA [30], DPGM [76], ECD [77], LSNMF [78], and jLMDC [79] are selected as baselines, which cover typical evolutionary clustering. MetaFac is selected because it is the first evolutionary clustering-based method for dynamic community detection. PiCES and DPGM are chosen because they are global smoothness-based algorithms with an excellent performance. DYNMOGA and ECD are deliberately employed because they are popular TSF-based algorithms. sE-NMF is also included because it also makes use of matrix factorization to learn features of vertices. LSNMF is selected is because it decomposes features of vertices into common and specific parts, where the specific features of vertices are promising for the dynamics of temporal networks. jLMDC is selected because it characterizes the dynamics of temporal networks by exploiting the roles of vertices, which enhances the performance of tracking dynamic communities in temporal networks.

B. Parameter analysis

The jNCDC algorithm has parameters (i.e., α, β, d, and δ), where α and β are the parameters of the evolutionary graph construction and graph contrastive learning, respectively; d is the number of features; and δ is used to select neighbors to construct the positive sample. d is selected using the instability of matrix factorization [80]. For parameter δ, we set as n/k.

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

[2]https://snap.stanford.edu/data/email-Eu-core-temporal.html

[3]http://networkrepository.com/edit-enwikibooks.php.

[4]http://networkrepository.com

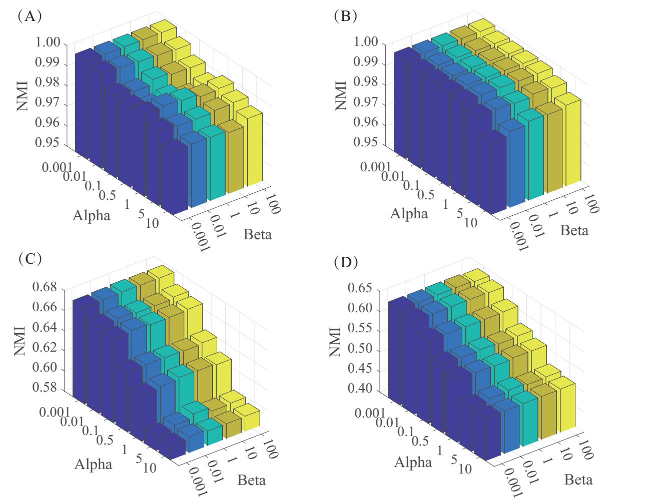


Fig. 3: Parameter effect of jNCDC on various networks: (A) Birthdeath, (B) Expansion, (C) Cellphone, and (D) Email.

jNCDC involves parameters α and β, which determine the importance of evolution graph construction and contrastive learning. Two artificial (Birthdeath and Expansion) and two real-world temporal networks (Cellphone and Email) are selected to investigate parameter effect. In detail, we investigate the effects of parameter variations on the performance of jNCDC, where $\alpha \in [10^{-3}, 10^{-2}, 10^{-1}, 0.5, 1, 10]$ and $\beta \in [10^{-3}, 10^{-2}, 1, 10, 10^2]$.

Performance of jNCDC for various networks with different parameter values is depicted in Fig. 3, where panel A is for Birthdeath, B for Expansion, C for Cellphone, and D for Email networks. From Figs. 3 A and B, it is easy to assert that the NMI of the proposed algorithm is quite stable as parameter α increases from 0.001 to 10 for the artificial networks. Moreover, jNCDC is insensitive to parameter β because NMI changes smoothly as β increases from 0.001 to 10. jNCDC is stable because of two possible reasons. First, jNCDC take advantage of contrastive learning to enhance quality of features, thereby offerring an improved approach to characterize and quantify the temporality of networks. Second, the proposed algorithm constructs an evolution graph to exploit the dynamics of networks at the vertex-level, where the relations of vertices at successive time are explicitly explored, shedding light on the intrinsic structure of evolving communities.

Then, we further investigate parameter effect by replacing artificial networks with real-world ones, which is shown in Figs. 3 C and D. From these panels, it is easy to find that jNCDC is sensitive to parameter α but insensitive to parameter β for real-world networks. Specifically, jNCDC is quit stable when parameter $\alpha \leq 0.1$. Moreover, the NMI of jNCDC decreases as parameter α keeps increasing from 0.1 to 10. The reason is that the objective function is primarily influenced by the evolution graph when α is large. In this case, jNCDC fails to reach a good balance between feature learning and temporality, thereby decreasing the quality of features. There is a valid rationale to explain why jNCDC is stable for parameter β.

Performing a thorough comparison between Fig. 3 A/B and C/D demonstrates that jNCDC is more stable in artificial networks than real-world ones. The reason is that structure and patterns in artificial networks are much easier to characterize than those in real ones because evolution events in artificial networks are regular. In this study, we set α=0.1 and β=1 for all experiments.

C. Performance on artificial temporal networks

1) SYN-FIX/SYN-VAR networks: For each type of artificial networks, we generate 100 networks to remove the randomness of evolution. All these algorithms are performed on these networks and the average NMI to measure performance of these algorithms (mean ± standard deviation). The performance of different algorithms on the SYN-FIX and SYN-VAR is shown in Table III.

According to the result presented in Table III, jNCDC demonstrates superior performance on SYN-FIX and SYN-VAR networks. Specifically, jNCDC, PisCES, MetaFac, LSNMF, and jLMDC exhibited the best performance, followed by DYNMOGA and sE-NMF. In detail, the NMI of jNCDC, PisCES, MetaFac, LSNMF, and jLMDC is 1.000, whereas it is 0.999 for DYNMOGA, 0.975 for sE-NMF, and is 0.967 for ECD. DPGM algorithm has the worst performance on SYN-FIX networks with an average NMI of 0.921. Furthermore, on the SYN-VAR networks, jNCDC achieves the best performance with NMI 0.998. jLMDC, ECD, LSNMF, PisCES, DYNMOGA, and sE-NMF are inferior to the proposed algorithm, where NMI of jLMDC is 0.994, of ECD is 0.993, of LSNMF is 0.969, of PisCES is 0.961, of DYNMOGA is 0.961, and NMI of sE-NMF is 0.945. Notice that DPGM and MetaFace achieve the worst performance on SYN-VAR networks.

DPGM achieves the worst performance because it is a probabilistic model based on topological structure, which is practical if and only if structure of dynamic communities is easy to detect (i.e., there are much more edges within communities than outside). In other words, it is very sensitive to network perturbation. Notice that all these algorithms achieve a good performance on the SYN-FIX/SYN-VAR networks because the evolution events are simple and regular, which are easy to characterize and capture. However, jNCDC is superior to these baselines, implying that it is more accurate to describe and model the dynamics of networks than state-of-the-art methods.

The superiority of the proposed algorithm can be attributed to several factors. First, jNCDC utilizes the high-order topological structure of networks, rather than the adjacent matrix, which provides a better way to capture the dynamics of networks. Second, jNCDC measures the dynamics of networks at the vertex-level, rather than on the global level, by exploiting the representation relations among vertices with features at the successive time, which effectively avoids the accumulated effect of subtle perturbation at the large-scale regions of networks. Third, jNCDC introduces graph contrastive learning for clustering of temporal networks, where partial information of positive and negative vertices leads to improved feature quality, thus enhancing algorithm performance.

2) Greene networks: SYN-FIX and SYN-VAR alone cannot fully assess the performance of diverse algorithms due to two primary factors. First, the evolution events are regular and fixed for all time, indicating that these dynamic communities are relatively easy to detect. Second, the sizes of networks are limited (i.e., less than 300 vertices), thereby failing to testify the performance of algorithms. To address this issue, Greene [74] is a typical benchmark for tracking dynamic community, which include four evolution events with 10,000 vertices.

Table III illustrates the performance of various algorithms on the Greene networks. The results indicate that jNCDC achieves the best performance across all four types of networks, suggesting it as a superior way to model and depict network temporality. Specifically, jNCDC, DYNOMGA, jLMDC, and LSNMF have similar performance in the four evolutionary events. NMI of jNCDC is 0.999 for Birthdeath, 0.999 for Expansion, 0.997 for Hide, and 0.998 for Mergesplit, respectively. The NMI of jLMDC, DYNMOGA, and are close to that of jNCDC. However, other algorithms, such as sE-NMF and PisCES, are inferior to jNCDC. In detail, the NMI of sE-NMF is 0.961 for Birthdeath, 0.975 for Expansion, 0.968 for Hide, and 0.977 for Mergesplit. However, the NMI of PisCES is inferior to sE-NMF, where NMI is 0.941 for Birthdeath, 0.963 for Expansion, 0.955 for Hide and 0.968 for Mergesplit, respectively. DPGM is also the worst for all four types of networks, indicating that probabilistic model fails to characterize the distribution of dynamics in networks because topology is incomprehensive.

PisCES is inferior to jNCDC because the global smoothness strategy is effective if and only if the temporality of networks is gentle. It also fails to precisely characterize the dynamics of networks at a particular time, leading to decreased algorithm performance for detecting dynamic communities in networks with large τ. LSNMF achieves excellent performance because it decomposes features into common features and specific features and explicitly measures the specificity of vertex features to characterize the dynamics of vertices. The reason for the excellent performance of jLMDC is that it uses the roles of vertices to measure the dynamics of the network at the micro-level to describe the evolution of the networks more accurately. Furthermore, the global smoothing strategy is time-consuming, implying that it is inapplicable to large-scale temporal networks. DYNMOGA takes the multi-objective optimization to balance CS and CT, where Pareto solutions are difficult to solve.

Table III shows that jNCDC is promising in identifying dynamic communities because it outperforms all these baselines in terms of accuracy, indicating that graph contrastive learning is effective for characterizing the dynamics of communities. There are several reasons accounting for the superiority of the proposed algorithms. First, jNCDC abandons the well-known temporal smoothness framework [26], which avoids the balance of CS and CT. Second, the learning of features and characterization of temporality are smoothly merged into an overarching objective function. Here, the features of vertices are learned with the guidance of network temporality. As such, features simultaneously reflect the connectivity of communities at the current time and temporality from the historical snapshot. Third, evolution graph is constructed with representation learning, where the relations among vertices at successive time are explicitly quantified, facilitating the identification of dynamic communities. Finally, the positive and negative vertices serve as partial information to promote the performance of jNCDC.

By replacing NMI with ACC, the performance of these algorithms is consistent, as shown in Table IV, showing these algorithms identify the truth dynamic communities. These results further validate the potential of contrastive learning in capturing and identifying dynamic communities within temporal networks.

D. Performance on real-world temporal networks

The previous experiments prove the superiority of jNCDC with artificial networks, and then we select these widely used real-world networks to verify the applicability of the proposed algorithm. In detail, four real-world temporal networks, namely, Cellphone, Email, Wikipedia, and Dublin are selected, where the number of vertices ranging from 400 to 25,000 as shown in Table II. For real-world networks, we follow the strategy in Ref. [81] to obtain the truth-ground communities.

NMI of various algorithms on real-world networks is shown in Table III, where jNCDC is the best algorithm for all these real networks. Specifically, the NMI of jNCDC is 0.699, 0.645, 0.314, and 0.450 for Cellphone, Email, Wikipedia and Dublin, respectively, whereas that of jLMDC is 0.607, 0.581, 0.282, 0.412 and of MetaFac is 0.597, 0.560, 0.263, 0.405. To our surprise, PisCES performs poorly in real networks. The possible reason is that evolution events in real networks are irregular, which are difficult to be characterized and captured with the global smoothing strategy. What we want to point out is that NMI of PisCES dramatically decreases as the number of vertices increases, showing the global smoothing strategy is inapplicable for large-scale networks. Notably, LSNMF performs well on small datasets (Cellphone and Email), but its performance on large datasets (Dublin and Wikipedia) is poor. The possible reason is that large-scale temporal network changes are irregular, LSNMF fails to fully utilize the temporality of the network and cannot accurately depict the evolution of the community. Furthermore, DYNMOGA and DPGM are inferior to others because they are criticized for their failure to balance clustering accuracy and drift.

Similar performance is shown in Table IV, showing that jNCDC identifies truth communities. jNCDC achieves an excellent performance because of several reasons. First, current algorithms quantify network dynamics based on the assumption that they are constant over time, which does not reflect the reality of temporal networks. jNCDC overcomes this limitation with self-representation learning to construct the evolution graph, where dynamics of snapshot at each time are precisely learned. Furthermore, the learned evolution graph specifies the relations of vertices between the previous and current time and improves the interpretability of dynamics of networks. Second, graph contrastive learning not only improves the quality of features with partial supervision information but also exploits the indirected relations among vertices in the evolution graph to characterize the dynamics of networks at different resolutions. Finally, jNCDC jointly learns feature learning, graph contrastive learning, and evolution graph construction, where features reflect the topological structure and temporality of networks.

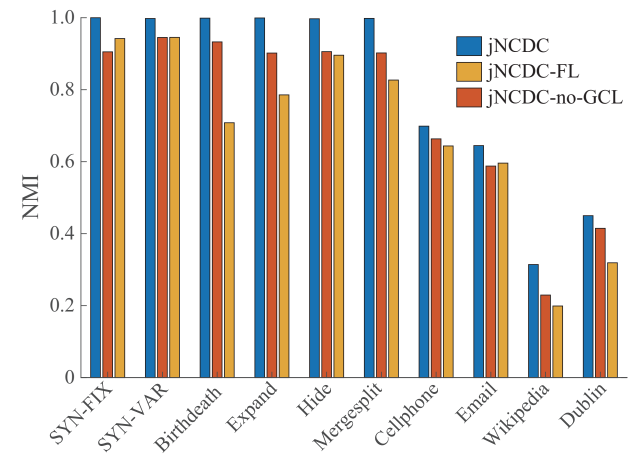


Fig. 4: Ablation study of among jNCDC, jNCDC-no-GCL and jNCDC-FL in terms of NMI.

E. Ablation Study

Given that jNCDC joins feature learning, evolution graph construction, and graph contrastive learning, an ablation study must be conducted with an immediate intention to verify the importance of these items. Therefore, two variants of jNCDC are proposed. These variants are jNCDC-FL, and jNCDC-no-GCL, where jNCDC-FL removes evolution graph construction and graph contrastive learning, and jNCDC-no-GCL only deletes graph contrastive learning.

The performance of the variants of jNCDC on various networks is shown in Fig. 4, where jNCDC significantly outperforms its variants for all 10 temporal networks. In detail, the NMI of jNCDC is 1.000 for SYN-FIX, 0.998 for SYN-VAR, 0.999 for Birthdeath, 0.999 for Expansion, 0.997 for Hide, 0.998 for Mergesplit, 0.699 for Cellphone, 0.645 for Email, 0.314 for Wikipedia, and 0.450 for Dublin. However, that of jNCDC-no-GCL is 0.905 for SYN-FIX, 0.945 for SYN-VAR, 0.933 for Birthdeath, 0.902 for Expansion, 0.906 for Hide, 0.902 for Mergesplit, 0.663

TABLE III: NMI of various algorithms on temporal networks, where bold values represent the best performance of these algorithms, and - denotes no output (means±sd).

Datasets	PisCES	DPGM	DYNMOGA	MetaFac	sE-NMF	ECD	LSNMF	jLMDC	jNCDC
SYN-FIX	1.000±0.000	0.921±0.007	0.999±0.001	1.000±0.000	0.975±0.010	0.967±0.028	1.000±0.000	1.000±0.000	1.000±0.000
SYN-VAR	0.961±0.002	0.920±0.005	0.961±0.006	0.895±0.014	0.945±0.020	0.993±0.006	0.969±0.000	0.994±0.005	0.998±0.000
Birthdeath	0.941±0.006	0.905±0.003	0.999±0.000	0.895±0.016	0.961±0.005	-	0.989±0.000	0.996±0.002	0.999±0.000
Expansion	0.963±0.003	0.929±0.004	0.981±0.000	0.976±0.008	0.975±0.004	-	0.998±0.000	0.997±0.002	0.999±0.000
Hide	0.955±0.001	0.928±0.007	0.991±0.000	0.975±0.003	0.968±0.003	-	0.991±0.000	0.996±0.003	0.997±0.000
Mergesplit	0.968±0.000	0.938±0.003	0.997±0.000	0.968±0.005	0.977±0.004	-	0.997±0.000	0.994±0.004	0.998±0.000
Cellphone	0.456±0.001	0.495±0.007	0.507±0.005	0.597±0.002	0.421±0.004	0.633±0.009	0.588± 0.000	0.607±0.005	0.699±0.002
Email	0.196±0.008	0.498±0.011	0.486±0.004	0.560±0.004	0.491±0.005	0.385±0.009	0.568±0.000	0.581±0.002	0.645±0.001
Wikipedia	0.259±0.002	0.206±0.003	0.241±0.001	0.263±0.008	0.258±0.001	-	0.212±0.000	0.282±0.008	0.314±0.000
Dublin	0.260±0.001	0.310±0.006	0.380±0.000	0.405±0.041	0.260±0.002	-	0.263±0.000	0.412±0.005	0.450±0.000

TABLE IV: ACC of various algorithms on temporal networks, where bold values represent the best performance of these algorithms, and - denotes no output (means±sd).

Datasets	PisCES	DPGM	DYNMOGA	MetaFac	sENMF	ECD	LSNMF	jLMDC	jNCDC
SYN-FIX	1.000±0.000	0.974±0.002	0.999±0.000	0.910±0.048	0.985±0.012	0.933±0.028	1.000±0.000	1.000±0.000	1.000±0.000
SYN-VAR	0.920±0.003	0.945±0.006	0.956±0.005	0.862±0.006	0.917±0.011	0.988±0.007	0.975±0.000	0.986±0.002	0.999±0.000
Birthdeath	0.803±0.005	0.786±0.010	0.888±0.009	0.895±0.016	0.872±0.008	-	0.977±0.000	0.974±0.001	0.999±0.000
Expansion	0.866±0.005	0.828±0.005	0.976±0.000	0.890±0.005	0.887±0.007	-	0.997±0.000	0.990±0.002	0.998±0.000
Hide	0.833±0.005	0.811±0.010	0.987±0.000	0.882±0.007	0.865±0.003	-	0.981±0.000	0.981±0.005	1.000±0.000
Mergesplit	0.882±0.005	0.824±0.007	0.999±0.000	0.850±0.008	0.891±0.005	-	0.982±0.000	0.973±0.004	0.994±0.001
Cellphone	0.195±0.008	0.254±0.006	0.287±0.005	0.290±0.001	0.217±0.002	0.366±0.006	0.342±0.000	0.385±0.012	0.449±0.003
Email	0.361±0.002	0.485±0.006	0.449±0.004	0.472±0.003	0.537±0.005	0.156±0.004	0.483±0.000	0.491±0.006	0.587±0.002
Wikipedia	0.147±0.017	0.063±0.006	0.157±0.008	0.156±0.024	0.146±0.007	-	0.149±0.000	0.128±0.006	0.174±0.004
Dublin	0.156±0.003	0.090±0.003	0.268±0.001	0.264±0.004	0.240±0.001	-	0.251±0.000	0.261±0.008	0.296±0.007

for Cellphone, 0.588 for Email, 0.229 for Wikipedia, and 0.415 for Dublin, which are 5%∼10% less than jNCDC. Moreover, the NMI of jNCDC-FL is 0.942 for SYN-FIX, 0.945 for SYN-VAR, 0.708 for Birthdeath, 0.786 for Expansion, 0.896 for Hide, 0.827 for Mergesplit, 0.644 for Cellphone, 0.596 for Email, 0.199 for Wikipedia, and 0.319 for Dublin respectively.

These results demonstrate that evolution graph and graph contrastive learning are important for the performance of jNCDC because removing either of them dramatically decreases performance. There several possible reasons explain why these items are critical. First, evolution graph provides supplemental information for features of vertices to characterize the dynamics of networks, which cannot be fulfilled only with the low-dimensional features of vertices and topological structure of temporality networks. Second, graph contrastive learning utilizes the functions of vertices in temporal networks, where the semantic information also provides additional information to characterize the dynamics of networks, which is ignored by current baselines.

VI. Conclusion

Temporal networks provide a more precise method for modeling the evolution of complex systems than static ones. However, it present a remarkable challenge in identifying dynamic communities. Compared with static communities, detecting dynamic community is highly intricate in temporal networks because it requires to balance clustering accuracy and clustering drift. Current algorithms also make use of temporal smoothness framework, which is criticized for the difficulty on the quantification of temporality, as well as determination of its importance. In this study, we propose the first graph contrastive learning to address these two issues, where the dynamics of networks are characterized at the vertex-level by learning an evolution graph, and partial information is also incorporated with graph contrastive learning. The results from our experiments reveal that our algorithm is highly effective and outperforms existing state-of-the-art methods.

In future research, we aim to address the following issues, which are summarized as

- jNCDC only exploits the roles of vertices in temporal networks by classifying them into static and dynamic ones, failing to comprehensively explore semantic of temporal networks. How to further investigate the roles of graph patterns in temporal networks for graph contrastive learning is promising for the characterization of dynamics of networks.
- Even though jNCDC is the first attempt to apply graph contrastive learning to clustering of temporal networks, it is executed on the constructed evolution graph, which is consistent with the traditional strategy at some content. Exploring methods is needed for directly applying contrastive learning on original temporal networks, specifically in the selection of positive and negative samples.

- jNCDC takes matrix factorization to extract low-dimensional representation of vertices in temporal networks, which runs the risk of ignoring the intricate structure. How to exploit more sophisticated representation of vertices to characterize dynamic communities is also interesting.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (62272361), and Shaanxi Natural Science Funds for Distinguished Young Scholar (Program No. 2022JC-38).

References

[1] Duncan J Watts, Peter Sheridan Dodds, and Mark EJ Newman. Identity and search in social networks. science, 296(5571):1302–1305, 2002.
[2] Song Bian, Qintian Guo, Sibo Wang, and Jeffrey Xu Yu. Efficient algorithms for budgeted influence maximization on massive social networks. Proceedings of the VLDB Endowment, 13(9):1498–1510, 2020.
[3] Qiyan Li, Yuanyuan Zhu, and Jeffrey Xu Yu. Skyline cohesive group queries in large road-social networks. In 2020 IEEE 36th International Conference on Data Engineering (ICDE), pages 397–408. IEEE, 2020.
[4] Torbjörn Säterberg, Stefan Sellman, and Bo Ebenman. High frequency of functional extinctions in ecological networks. Nature, 499(7459):468–470, 2013.
[5] Xiaoke Ma, Long Gao, and Kai Tan. Modeling disease progression using dynamics of pathway connectivity. Bioinformatics, 30(16):2343–2350, 2014.
[6] Xiaoke Ma, Penggang Sun, and Zhong-Yuan Zhang. An integrative framework for protein interaction network and methylation data to discover epigenetic modules. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 16(6):1855–1866, 2018.
[7] Mark EJ Newman. Modularity and community structure in networks. Proceedings of the national academy of sciences, 103(23):8577–8582, 2006.
[8] Santo Fortunato. Community detection in graphs. Physics reports, 486(3-5):75–174, 2010.
[9] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. Physical review E, 69(2):026113, 2004.
[10] Supriya Krishnamurthy, Sameh El-Ansary, Erik Aurell, and Seif Haridi. A statistical theory of chord under churn. In International Workshop on Peer-to-Peer Systems, pages 93–103. Springer, 2005.
[11] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 551–556, 2004.
[12] Xiaoke Ma, Lin Gao, Xuerong Yong, and Lidong Fu. Semi-supervised clustering algorithm for community structure detection in complex networks. Physica A: Statistical Mechanics and its Applications, 389(1):187–197, 2010.
[13] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In Computer and Information Sciences-ISCIS 2005: 20th International Symposium, Istanbul, Turkey, October 26-28, 2005. Proceedings 20, pages 284–293. Springer, 2005.
[14] Punam Bedi and Chhavi Sharma. Community detection in social networks. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 6(3):115–135, 2016.
[15] Muhammad Aqib Javed, Muhammad Shahzad Younis, Siddique Latif, Junaid Qadir, and Adeel Baig. Community detection in networks: A multidisciplinary review. Journal of Network and Computer Applications, 108:87–111, 2018.
[16] Sandro Cavallari, Vincent W Zheng, Hongyun Cai, Kevin Chen-Chuan Chang, and Erik Cambria. Learning community embedding with community detection and node embedding on graphs. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pages 377–386, 2017.
[17] Shi Jianbo and Malik J. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):888–905, 2000.
[18] Tiantian He, Yang Liu, Tobey H Ko, Keith CC Chan, and Yew-Soon Ong. Contextual correlation preserving multiview featured graph clustering. IEEE transactions on cybernetics, 50(10):4318–4331, 2019.
[19] Liang Yang, Xiaochun Cao, Di Jin, Xiao Wang, and Dan Meng. A unified semi-supervised community detection framework using latent space graph regularization. IEEE transactions on cybernetics, 45(11):2585–2598, 2014.
[20] Dongxiao He, Dayou Liu, Di Jin, and Weixiong Zhang. A stochastic model for detecting heterogeneous link communities in complex networks. In Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.
[21] Di Jin, Kunzeng Wang, Ge Zhang, Pengfei Jiao, Dongxiao He, Francoise Fogelman-Soulie, and Xin Huang. Detecting communities with multiplex semantics by distinguishing background, general, and specialized topics. IEEE Transactions on Knowledge and Data Engineering, 32(11):2144–2158, 2019.
[22] Holme Petter and Saramäki Jari. Temporal networks. Physics Reports, 519:97–125, 2012.
[23] Liao Hao, Mariani Manuel S., Medo Matúš, Zhang Yi-Cheng, and Zhou Ming-Yang. Ranking in evolving complex networks. Physics Reports, 689:1–54, 2017.
[24] Boccaletti S., Lellis P.De., Genio C.I., Alfaro-Bittner K., Criado R., Jalan S., and Romance M. The structure and dynamics of networks with higher order interactions. Physics Reports, 519:97–125, 2012.
[25] Gergely Palla, Albert-László Barabási, and Tamás Vicsek. Quantifying social group evolution. Nature, 446(7136):664–667, 2007.
[26] Deepayan Chakrabarti, Ravi Kumar, and Andrew Tomkins. Evolutionary clustering. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 554–560, 2006.
[27] Yu-Ru Lin, Jimeng Sun, Paul Castro, Ravi Konuru, Hari Sundaram, and Aisling Kelliher. Metafac: community discovery via relational hypergraph factorization. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 527–536, 2009.
[28] Wenjing Wang and Xiang Li. Temporal stable community in time-varying networks. IEEE Transactions on Network Science and Engineering, 7(3):1508–1520, 2019.
[29] Haozhe Wu, Zhiyuan Hu, Jia Jia, Yaohua Bu, Xiangnan He, and Tat-Seng Chua. Mining unfollow behavior in large-scale online social networks via spatial-temporal interaction. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 254–261, 2020.
[30] Francesco Folino and Clara Pizzuti. An evolutionary multi-objective approach for community discovery in dynamic networks. IEEE Transactions on Knowledge and Data Engineering, 26(8):1838–1852, 2013.
[31] Yanning Shen, Brian Baingana, and Georgios B Giannakis. Tensor decompositions for identifying directed graph topologies and tracking dynamic networks. IEEE Transactions on Signal Processing, 65(14):3675–3687, 2017.
[32] Xiangxiang Zeng, Wen Wang, Cong Chen, and Gary G Yen. A consensus community-based particle swarm optimization for dynamic community detection. IEEE transactions on cybernetics, 50(6):2502–2513, 2019.
[33] Dongyuan Li, Qiang Lin, and Xiaoke Ma. Identification of dynamic community in temporal network via joint learning graph representation and nonnegative matrix factorization. Neurocomputing, 435:77–90, 2021.
[34] Zhen Wang, Chunyu Wang, Xianghua Li, Chao Gao, Xuelong Li, and Junyou Zhu. Evolutionary markov dynamics for network community detection. IEEE Transactions on Knowledge & Data Engineering, 34(3):1206–1220, 2022.
[35] Myeongjin Park, Seung-Hoon Lee, Oh-Min Kwon, and Alexandre Seuret. Closeness-centrality-based synchronization criteria for complex dynamical networks with interval time-varying coupling delays. IEEE transactions on cybernetics, 48(7):2192–2202, 2017.
[36] Xiaoke Ma and Di Dong. Evolutionary nonnegative matrix factorization algorithms for community detection in dynamic

networks. IEEE transactions on knowledge and data engineering, 29(5):1045–1058, 2017.
[37] Yun Chi, Xiaodan Song, Dengyong Zhou, Koji Hino, and Belle L Tseng. On evolutionary spectral clustering. ACM Transactions on Knowledge Discovery from Data (TKDD), 3(4):1–30, 2009.
[38] Mahsa Seifikar, Saeed Farzi, and Masoud Barati. C-blondel: an efficient louvain-based dynamic community detection algorithm. IEEE Transactions on Computational Social Systems, 7(2):308–318, 2020.
[39] Manoj K Agarwal, Krithi Ramamritham, and Manish Bhide. Real time discovery of dense clusters in highly dynamic graphs: identifying real world events in highly dynamic environments. arXiv preprint arXiv:1207.0138, 2012.
[40] Jimeng Sun, Christos Faloutsos, Spiros Papadimitriou, and Philip S Yu. Graphscope: parameter-free mining of large time-evolving graphs. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 687–696, 2007.
[41] Ravi Kumar, Jasmine Novak, and Andrew Tomkins. Structure and evolution of online social networks. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 611–617, 2006.
[42] Di Zhuang, J Morris Chang, and Mingchen Li. Dynamo: Dynamic community detection by incrementally maximizing modularity. IEEE Transactions on Knowledge and Data Engineering, 33(5):1934–1945, 2019.
[43] Fuchen Liu, David Choi, Lu Xie, and Kathryn Roeder. Global spectral clustering in dynamic networks. Proceedings of the National Academy of Sciences, 115(5):927–932, 2018.
[44] Tianbao Yang, Yun Chi, Shenghuo Zhu, Yihong Gong, and Rong Jin. Detecting communities and their evolutions in dynamic social networks—a bayesian approach. Machine learning, 82(2):157–189, 2011.
[45] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 2, pages 1735–1742. IEEE, 2006.
[46] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 8547–8555, 2021.
[47] Vivek Sharma, Makarand Tapaswi, M Saquib Sarfraz, and Rainer Stiefelhagen. Clustering based contrastive learning for improving face representations. In 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020), pages 109–116. IEEE, 2020.
[48] Erlin Pan and Zhao Kang. Multi-view contrastive graph clustering. Advances in neural information processing systems, 34:2148–2159, 2021.
[49] Leto Peel and Aaron Clauset. Detecting change points in the large-scale structure of evolving networks. In Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.
[50] Tue Herlau, Morten Mørup, and Mikkel Schmidt. Modeling temporal evolution and multiscale structure in networks. In International Conference on Machine Learning, pages 960–968. PMLR, 2013.
[51] Uriel Singer, Ido Guy, and Kira Radinsky. Node embedding over temporal graphs. arXiv preprint arXiv:1903.08889, 2019.
[52] Hogun Park and Jennifer Neville. Exploiting interaction links for node classification with deep graph neural networks. In IJCAI, volume 2019, pages 3223–3230, 2019.
[53] Thomas Aynaud and Jean-Loup Guillaume. Multi-step community detection and hierarchical time segmentation in evolving networks. In Proceedings of the 5th SNA-KDD workshop, volume 11, 2011.
[54] Jianlei Zhang, Yuying Zhu, and Zengqiang Chen. Evolutionary game dynamics of multiagent systems on multiple community networks. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 50(11):4513–4529, 2018.
[55] Dongyuan Li, Xiaoke Ma, and Maoguo Gong. Joint learning of feature extraction and clustering for large-scale temporal networks. IEEE Transactions on Cybernetics, 2021.
[56] Di Jin, Binbin Zhang, Yue Song, Dongxiao He, Zhiyong Feng, Shizhan Chen, Weihao Li, and Katarzyna Musial. Modmrf: A modularity-based markov random field method for community detection. Neurocomputing, 405:218–228, 2020.
[57] Dongxiao He, Yue Song, Di Jin, Zhiyong Feng, Binbin Zhang, Zhizhi Yu, and Weixiong Zhang. Community-centric graph convolutional network for unsupervised community detection. In Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence, pages 3515–3521, 2021.
[58] Li Guo, Long Chen, Xiliang Lu, and CL Philip Chen. Membership affinity lasso for fuzzy clustering. IEEE Transactions on Fuzzy Systems, 28(2):294–307, 2019.
[59] Zhigang Liu, Yugen Yi, and Xin Luo. A high-order proximity-incorporated nonnegative matrix factorization-based community detector. IEEE Transactions on Emerging Topics in Computational Intelligence, 7(3):700–714, 2023.
[60] Yong Peng, Wenna Huang, Wanzeng Kong, Feiping Nie, and Bao-Liang Lu. Jgsed: An end-to-end spectral clustering model for joint graph construction, spectral embedding and discretization. IEEE Transactions on Emerging Topics in Computational Intelligence, 7(6):1687–1701, 2023.
[61] Peter J Mucha, Thomas Richardson, Kevin Macon, Mason A Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. science, 328(5980):876–878, 2010.
[62] Qi Zhao, Bai Yan, Jian Yang, and Yuhui Shi. Evolutionary robust clustering over time for temporal data. IEEE Transactions on Cybernetics, 53(7):4334–4346, 2023.
[63] Jianrui Chen, Tingting Zhu, Maoguo Gong, and Zhihui Wang. A game-based evolutionary clustering with historical information aggregation for personal recommendation. IEEE Transactions on Emerging Topics in Computational Intelligence, 7(2):552–564, 2022.
[64] Huixin Ma, Kai Wu, Handing Wang, and Jing Liu. Higher-order knowledge transfer for dynamic community detection with great changes. IEEE Transactions on Evolutionary Computation, pages 1–1, 2023.
[65] Ali Jazayeri and Christopher C. Yang. Frequent pattern mining in continuous-time temporal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46(1):305–321, 2024.
[66] Tianpeng Li, Wenjun Wang, Pengfei Jiao, Yinghui Wang, Ruomeng Ding, Huaming Wu, Lin Pan, and Di Jin. Exploring temporal community structure via network embedding. IEEE Transactions on Cybernetics, 53(11):7021–7033, 2023.
[67] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755):788–791, 1999.
[68] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.
[69] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), volume 1, pages 539–546. IEEE, 2005.
[70] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In International conference on machine learning, pages 1597–1607. PMLR, 2020.
[71] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. Advances in neural information processing systems, 27, 2014.
[72] Jingyi You, Chenlong Hu, Hidetaka Kamigaito, Kotaro Funakoshi, and Manabu Okumura. Robust dynamic clustering for temporal networks. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pages 2424–2433, 2021.
[73] Cai Deng, He Xiaofei, Han Jiawei, and Huang Thomas S. Graph regularized nonnegative matrix factorization for data representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(8):1548–1560, 2011.
[74] Derek Greene, Donal Doyle, and Padraig Cunningham. Tracking the evolution of communities in dynamic social networks. In 2010 international conference on advances in social networks analysis and mining, pages 176–183. IEEE, 2010.
[75] Leon Danon, Albert Diaz-Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identification. Journal of

statistical mechanics: Theory and experiment, 2005(9), 2005. Art.no. P09008.

[76] Sikun Yang and Heinz Koeppl. A poisson gamma probabilistic model for latent node-group memberships in dynamic networks. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018.

[77] Fanzhen Liu, Jia Wu, Chuan Zhou, and Jian Yang. Evolutionary community detection in dynamic social networks. In 2019 International Joint Conference on Neural Networks (IJCNN), pages 1–7. IEEE, 2019.

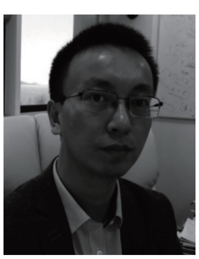
[78] Xiaoke Ma, Wei Zhao, and Wenming Wu. Layer-specific modules detection in cancer multi-layer networks. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 20(2):1170–1179, 2022.

[79] Dongyuan Li, Xiaoke Ma, and Maoguo Gong. Joint learning of feature extraction and clustering for large-scale temporal networks. IEEE Transactions on Cybernetics, 53(3):1653–1666, 2023.

[80] Wu Siqi, Joseph Antony, and et al. S, Ann. Stability-driven nonnegative matrix factorization to interpret spatial gene expression and build local gene networks. Proc. Nat. Academy Sci. USA, 113(16):4290–4295, 2016.

[81] Yu-Ru Lin, Yun Chi, Shenghuo Zhu, Hari Sundaram, and Belle L Tseng. Analyzing communities and their evolutions in dynamic social networks. ACM Transactions on Knowledge Discovery from Data (TKDD), 3(2):1–31, 2009.

Yun Ai received the BS degree in software engineering from Nanchang University in 2021. He is currently working toward the master degree with the School of Computer Science and Technology, Xidian University. His research interests include machine learning and data mining.



Xianghua Xie received the M.Sc. and Ph.D. degrees in computer science from the University of Bristol, Bristol, U.K., in 2002 and 2006, respectively. He is currently a Full Professor with the Department of Computer Science, Swansea University, Swansea, U.K., and is leading the Computer Vision and Machine Leaning Laboratory, Swansea University. He has published more than 160 refereed conference and journal publications and (co-)edited several conference proceedings. His research interests include various aspects of pattern recognition and machine intelligence and their applications to real-world problems. He is a member of BMVA. He is an Associate Editor of a number of journals, including Pattern Recognition and IET Computer Vision.

Xiaoke Ma received his Ph.D. degree in computer science from Xidian University in 2012. He was a post-doctor at the University of Iowa (USA) during 2012-2015. He is a full professor of School of computer science and technology, Xidian University (P.R.China). His research interests include machine learning, data mining and bioinformatics.He is an ad hoc reviewer for many international journals and publishes about 100 papers in the peer-reviewed international journals, such as IEEE Transactions Knowledge and Data Engineering, IEEE Transactions on Cybernetics, ACM Transactions on Knowledge Discovery from Data, IEEE Transactions on Neural Network and Learning Systems, Pattern Recognition, Information Sciences, Bioinformatics, Nuclear Acids Research, IEEE/ACM Transactions on Computational Biology and Bioinformatics, IEEE Transactions NanoBioScience.