Community Detection via Autoencoder-Like Nonnegative Tensor Decomposition

Jiewen Guan, Bilian Chen, and Xin Huang

***Abstract*— Community detection aims at partitioning a network into several densely connected subgraphs. Recently, nonnegative matrix factorization (NMF) has been widely adopted in many successful community detection applications. However, most existing NMF-based community detection algorithms neglect the multihop network topology and the extreme sparsity of adjacency matrices. To resolve them, we propose a novel conception of adjacency tensor, which extends adjacency matrix to multihop cases. Then, we develop a novel tensor Tucker decomposition-based community detection method—autoencoder-like nonnegative tensor decomposition (ANTD), leveraging the constructed adjacency tensor. Distinct from simply applying tensor decomposition on the constructed adjacency tensor, which only works as a decoder, ANTD also introduces an encoder component to constitute an autoencoder-like architecture, which can further enhance the quality of the detected communities. We also develop an efficient alternative updating algorithm with convergence guarantee to optimize ANTD, and theoretically analyze the algorithm complexity. Moreover, we also study a graph regularized variant of ANTD. Extensive experiments on real-world benchmark networks by comparing 27 state-of-the-art methods, validate the effectiveness, efficiency, and robustness of our proposed methods.**

***Index Terms*— Community detection, graph clustering, nonnegative tensor decomposition, optimization.**

I. Introduction

NETWORKS are prevalent on modeling entities and their mutual relationships in many scientific areas [1], [2]. Communities, in which nodes are densely connected but between which nodes are sparsely connected, naturally exist as functional modules in many real-world networks, such as collaboration networks, social networks, biological networks, just to name a few [3]. Revealing the community structure of a network, namely *community detection*, which serves as a fundamental analysis tool for analyzing and understanding complex networks [4], is an important but difficult problem. Recently, community detection has been widely applied into many real-world applications, such as promoting trust-aware recommender systems [5], analyzing COVID-19 data [6], and so on.

Detecting communities in complex networks has been extensively investigated in the last few decades, and numerous methods have been proposed. Since there is no consensus in academia about the strict definition of community detection, traditional community detection algorithms mainly focus on partitioning a graph under different heuristic criteria, such as optimizing modularity [7], minimizing the description length of random walks [8], and iteratively removing edges with maximum betweenness [3]. However, these methods use heuristic strategies to optimize structural objectives only. Moreover, these methods are unable to detect overlapping communities, which are common and natural in reality. Recently, nonnegative matrix factorization (NMF) [9] has been widely adopted for detecting communities due to its good interpretability derived from its nonnegative nature, wide applicability to both disjoint and overlapping community detection tasks, and great versatility to detect any specific number of communities. NMF aims to factorize the adjacency matrix into two nonnegative factor matrices, where one represents the centers of communities and the other represents the soft community assignments of nodes. In view of this, the underlying community structure can be determined in both disjoint and overlapping manner according to the community assignment matrix.

Nevertheless, there are two main issues inherently lie in existing NMF-based community detection methods. On the one hand, the adjacency matrix is extremely sparse. Generally, for real-world networks, over 99% elements of the adjacency matrix are zeros. However, although the sparsity of the adjacency matrix can alleviate data storage burden, such zero elements provide almost no information about the underlying community structure. When this fact has been neglected, as what existing NMF-based methods do, zero entries will dominate the iterative optimization process and eventually cause the detected community structure less reliable. On the other hand, the adjacency matrix only contains one-hop relationships between nodes, lacking explicit higher-order ones. As pointed out in [10] and [11], the higher-order structural information is critical for uncovering the community structure in complex networks. However, most existing NMF-based community detection methods only deal with the adjacency matrix, which restricts the quality of the detected communities.

Manuscript received 16 July 2021; revised 22 June 2022; accepted 22 August 2022. This work was supported in part by the Youth Innovation Fund of Xiamen under Grant 3502Z20206049, in part by the National Natural Science Foundation of China under Grant 61836005, and in part by The Research Grants Council of Hong Kong (HK RGC) under Grant 22200320. *(Corresponding author: Bilian Chen.)*

Jiewen Guan and Bilian Chen are with the Department of Automation, Xiamen University, Xiamen 361005, China, and also with the Xiamen Key Laboratory of Big Data Intelligent Analysis and Decision-Making, Xiamen 361005, China (e-mail: jwguan@stu.xmu.edu.cn; blchen@xmu.edu.cn).

Xin Huang is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong (e-mail: xinhuang@comp.hkbu.edu.hk).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2022.3201906.

Digital Object Identifier 10.1109/TNNLS.2022.3201906

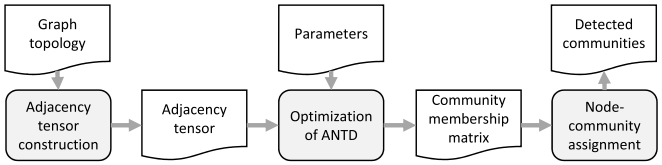


Fig. 1. Workflow of ANTD-based community detection.

To tackle the above limitations, in this article, we first propose a novel concept of *adjacency tensor*, which uniformly resolves the aforementioned two issues in adjacency matrix. Then, we propose a novel tensor Tucker decomposition [12] based community detection method named autoencoder-like nonnegative tensor decomposition (ANTD) based on the proposed adjacency tensor, which fully considers the higher-order connectivity patterns of a network. Specifically, the adjacency tensor is constructed by stacking normalized powers of the adjacency matrix in order. Since the kth power of the adjacency matrix gives the number of k-length paths connecting the vertex pairs in the network [13], the higher-order structural information can be integrated into the adjacency tensor, which resolves the information scarcity issue. Besides, in general, as the power of the adjacency matrix gets higher (i.e., the allowed length of walks becomes longer), the possibility of two nodes get connected becomes larger, and hence the stacked adjacency tensor becomes denser, which resolves the zero overload issue. For processing the constructed adjacency tensor, we propose the ANTD method. Instead of directly applying tensor decomposition on the constructed adjacency tensor, which only works as a decoder, our proposed ANTD method has an autoencoder-like architecture, which is proven to be critical for community detection [14], [15]. Specifically, the decoder component aims to reconstruct the adjacency tensor by node and community representations, in which node-community memberships and community–community interactions are embedded, respectively. The encoder component is dual with the decoder component and is responsible for mapping the original adjacency tensor into the community interaction space with the aid of node representations. As a result, the autoencoder-like architecture of ANTD endows the community membership matrix with implicit orthogonality, which increases its quality and thereby leads to a better community detection performance. Fig. 1 describes the schematic workflow of the whole ANTD-based community detection process.

In summary, we highlight our contributions as follows.

1) We propose a novel concept of adjacency tensor, which is an extension of adjacency matrix, to depict multihop topological structures in a network. The good side is that our proposed adjacency tensor is relatively denser than the original adjacency matrix.
2) We propose a novel community detection method ANTD, which is built upon the proposed adjacency tensor and thus has the capacity to incorporate multihop relational information. Besides, we also study a variant of ANTD, namely ANTDg, which imposes a graph regularizer on top of ANTD.
3) We derive an efficient iterative optimization algorithm with convergence guarantee to optimize our proposed ANTD method. The computational complexity of our proposed algorithm scales quadratically instead of cubically with the number of nodes in the network, which is the same as many existing NMF-based algorithms and thus guarantees its efficiency. Moreover, we also design and analyze an optimization algorithm for ANTDg.
4) We conduct extensive experiments to test the ANTD and ANTDg methods. Specifically, on a variety of real-world benchmark networks, we compare them with the state of the art. Besides, for ANTD, we also validate its convergence, test its running time, scalability and robustness.

The rest of this article is organized as follows. We review related community detection algorithms in Section II and introduce related preliminaries in Section III. Then, we present our novel conception of adjacency tensor and the ANTD and ANTDg methods in Section IV. Subsequently, we present the optimization algorithms with theoretical analysis for ANTD and ANTDg in Section V. Finally, we report the experimental results in Section VI and conclude this article in Section VII.

II. Related Work

In this section, we present a brief review of related studies on community detection.

A. Heuristic Criteria Based Community Detection

Classical community detection algorithms focus on partitioning a network based on optimizing different heuristic criteria [14], [15], [16]. Clauset *et al.* [7] proposed to optimize a well-known metric of modularity, which is an indicator of to what extent a network partition is distinct from randomness. Along this line, many other optimization methods are investigated, such as taking the leading eigenvector of the modularity matrix as indicator [17] and using greedy-based algorithm to maximize modularity [7]. Other community criteria include permanence [18] and conductance [19]. As these heuristics are not the focus of this article, we refer interested readers to a full survey [16] of them. Most heuristic algorithms can determine the number of communities automatically. However, such criteria-based heuristic algorithms may suffer from the issue of *resolution limit* [20] and thus affect the quality of the detected communities.

B. Learning Model Based Community Detection

Learning model-based community detection algorithms try to learn compact node representations to determine the underlying community structure. As a widely adopted learning model, NMF has good interpretability [9] and many applications. Psorakis *et al.* [21] proposed a Bayesian NMF model (BNMF) to learn node-community memberships in a Bayesian inference manner. Kuang *et al.* [22] proposed the symmetric NMF model (SymmNMF) to cluster graph nodes, which fully preserves the symmetry of the adjacency matrix.

Zhang and Yeung [23] proposed the bounded nonnegative matrix tri-factorization model to tackle the overlapping community detection problem. Yang and Leskovec [24] proposed a NMF-based method BigClam for overlapping community detection, which is very efficient that could handle very large networks. Sun *et al.* [14] proposed a nonnegative symmetric encoder–decoder approach (NNSED) for community detection, which was the first to show the importance of an autoencoder-like architecture. Ye *et al.* [15] proposed a novel deep autoencoder-like NMF model (DANMF) for community detection, which had extended Sun's NNSED [14] to a deep autoencoder-like architecture. Moreover, Ye *et al.* [25], [26], [27] proposed three more advanced NMF-based models for community detection. However, all these algorithms use the adjacency matrix only, neglecting higher-order network topology, which is proven to be conducive for community detection [10], [11]. Even worse, their optimization process may be overwhelmed by zero entries in the adjacency matrix, leading to a suboptimal community structure.

As another line of work, network embedding methods have also been adopted for community detection, and usually they can incorporate higher-order network topology. Wang *et al.* [28] proposed the modularized NMF (MNMF) to learn node representations by simultaneously factorizing the second-order affinity matrix and optimizing modularity. Neural network-based network embedding methods such as DeepWalk [29], Node2Vec [30], and GraRep [31] are also prevalent. However, all these network embedding methods usually perform unsatisfactorily on community detection, since they need to conduct post clustering on the learned embeddings to get communities instead of directly learning them.

Recently, due to the advances of deep learning technologies, many neural network-based community detection models have been proposed. E.g., as pioneers, Zhang *et al.* [32], [33], [34], [35] proposed the adaptive graph convolution method, the spectral embedding network, the GraphNet framework, as well as a capsual network-based community detection framework. As most of these models target at detecting communities in attributed networks, which is beyond the scope of this article, we only briefly review them and refer interested readers to a recent nice survey article [36] for a comprehensive treatment.

III. Preliminaries

In this section, we present basic notations and terminologies used throughout this article.

A. Notations

We use boldface calligraphic letters to denote tensors, boldface capital letters to denote matrices, boldface lowercase letters to denote vectors, and italic lowercase letters to denote scalar values. An element of a vector $\mathbf{x}$, a matrix $\mathbf{X}$ and a third-order tensor $\boldsymbol{\mathcal{X}}$ is denoted by x_i, x_{ij}, and x_{ijk}, respectively. We use $\mathbf{I}_d$ to denote an identity matrix in $\mathbb{R}^{d\times d}$, $\mathbf{1}_d$ to denote a d-dimensional all-one vector, $\mathrm{diag}(\mathbf{x})$ to denote a diagonal matrix whose diagonal entries are composed of $\mathbf{x}$ in order, and $\langle\cdot,\cdot\rangle$ to denote the inner product between two vectors, matrices, or tensors. For a matrix $\mathbf{X}$, $\mathbf{x}^i$, and $\mathbf{x}_j$ are used to represent the ith row and the jth column of $\mathbf{X}$. We adopt $\mathrm{Tr}(\mathbf{X})$ to denote the trace of $\mathbf{X}$ if it is square, $\mathbf{X}^T$ to denote the transpose of $\mathbf{X}$, and $\|\mathbf{X}\|_F$ to denote the Frobenius norm of $\mathbf{X}$. The Kronecker product is denoted as $\otimes$, and the Hadamard product is denoted by $\circledast$. For a third-order tensor $\boldsymbol{\mathcal{X}}$, each vector along its ith mode is called the mode-i fiber. $\mathbf{X}_{(i)}$ denotes the matricization of $\boldsymbol{\mathcal{X}}$ along the ith mode, which can be constructed by arranging the mode-i fibers to be the columns of the resulting matrix. The n-mode (matrix) product of $\boldsymbol{\mathcal{X}}$ with $\mathbf{U}$ is denoted by $\boldsymbol{\mathcal{X}}\times_n\mathbf{U}$, and the Frobenius norm of $\boldsymbol{\mathcal{X}}$ is denoted by $\|\boldsymbol{\mathcal{X}}\|_F$. Besides, we use $\boldsymbol{\mathcal{X}}(:,:,i)$ and $\mathbf{X}^{(i)}$ interchangeably to denote the ith frontal slice of $\boldsymbol{\mathcal{X}}$. More details of tensor manipulations can be referred to [12].

In this article, we consider an undirected and unweighted network $\mathcal{G}=(\mathcal{V},\mathcal{E})$, where $\mathcal{V}=\{v_1,v_2,\ldots,v_n\}$ represents the node set of $\mathcal{G}$ and $\mathcal{E}=\{e_1,e_2,\ldots,e_m\}\subset\mathcal{V}\times\mathcal{V}$ denotes the edge set of $\mathcal{G}$ respectively. We denote $n=|\mathcal{V}|$ and $m=|\mathcal{E}|$. Besides, we use $N_{\mathcal{G}}(v_i)=\{u:\langle u,v_i\rangle\in\mathcal{E},u\in\mathcal{V}\}$ to denote the neighborhood of v_i in $\mathcal{G}$ and $d_{\mathcal{G}}(v_i)=|N_{\mathcal{G}}(v_i)|$ to denote the degree of v_i in $\mathcal{G}$. The diameter of $\mathcal{G}$ is the maximum length of the shortest path between two nodes in $\mathcal{G}$, denoted as $\mathrm{diam}(\mathcal{G})$. The network $\mathcal{G}$ is also represented by a Boolean adjacency matrix $\mathbf{A}\in\mathbb{B}^{n\times n}$, whose (i,j)th entry $a_{ij}=1$ if there is an edge connecting nodes v_i and v_j, or 0 if not. Assume that we know *a priori* that there are k communities to be detected in $\mathcal{G}$, then the community detection algorithm can return k detected disjoint communities as $\mathcal{C}=\{c_i:c_i\neq\varnothing,\bigcup_{i=1}^k c_i=\mathcal{V},c_i\cap c_j=\varnothing,\forall i\neq j\}$, where c_i denotes the ith community for $1\le i\le k$. Besides, we use $\xi(v_i)$ to denote the index of the community containing v_i, i.e., $v_i\in c_{\xi(v_i)}$. The induced subgraph of $\mathcal{G}$ by c_i is denoted as $\mathcal{G}_{c_i}=(\mathcal{V}_{c_i},\mathcal{E}_{c_i})$.

B. Nonnegative Tucker Decomposition

Nonnegative Tucker decomposition (NTD) [37] is a special case of the Tucker decomposition [12], where in the former case all the factors are required to be nonnegative. It works as a building block of our proposed ANTD method. The nonnegative Tucker decomposition decomposes a nonnegative data tensor into a nonnegative core tensor multiplied by a nonnegative factor matrix along each mode. In the three-way case, $\boldsymbol{\mathcal{X}}\in\mathbb{R}_+^{n_1\times n_2\times n_3}$ is decomposed by NTD as

$$\boldsymbol{\mathcal{X}}\approx\boldsymbol{\mathcal{T}}\times_1\mathbf{F}\times_2\mathbf{H}\times_3\mathbf{K}=[\![\boldsymbol{\mathcal{T}};\mathbf{F},\mathbf{H},\mathbf{K}]\!]$$

where $\mathbf{F}\in\mathbb{R}_+^{n_1\times m_1}$, $\mathbf{H}\in\mathbb{R}_+^{n_2\times m_2}$, $\mathbf{K}\in\mathbb{R}_+^{n_3\times m_3}$, $\boldsymbol{\mathcal{T}}\in\mathbb{R}_+^{m_1\times m_2\times m_3}$, and $[\![\boldsymbol{\mathcal{T}};\mathbf{F},\mathbf{H},\mathbf{K}]\!]$ is a shorthand for $\boldsymbol{\mathcal{T}}\times_1\mathbf{F}\times_2\mathbf{H}\times_3\mathbf{K}$. In most cases, solving the nonnegative Tucker decomposition resorts to the following Frobenius norm approximation:

$$\min_{\mathbf{F},\mathbf{H},\mathbf{K},\boldsymbol{\mathcal{T}}}\left\|\boldsymbol{\mathcal{X}}-[\![\boldsymbol{\mathcal{T}};\mathbf{F},\mathbf{H},\mathbf{K}]\!]\right\|_F^2\quad\text{s.t. }\mathbf{F},\mathbf{H},\mathbf{K},\boldsymbol{\mathcal{T}}\ge 0.$$

IV. ANTD and Its Variant

In this section, we first propose a novel concept of adjacency tensor, which uniformly resolves the issues of information shortage and zero overload of adjacency matrices. Based on the constructed adjacency tensor, we then derive the ANTD method and a graph regularized variant of ANTD.

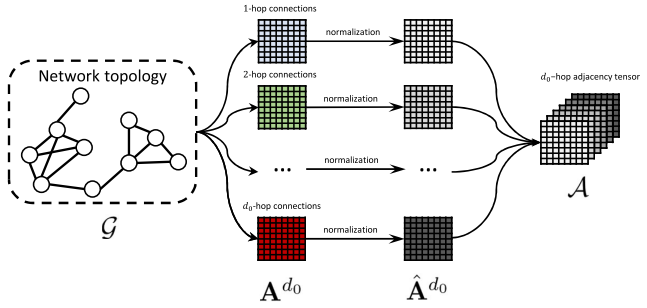


Fig. 2. Construction framework of adjacency tensor $\mathcal{A}$.

A. Adjacency Tensor Construction

As the social diffusion expresses a dynamic recursive pattern to influence nodes in the network, multihop node relations provide much more additional conducive information for modeling the community structure [10], [11]. Nevertheless, most existing NMF-based community detection algorithms only consider the first-order node connectivity information, which is inadequate as it is restricted to only one hop and extremely sparse. To this end, we propose the concept of adjacency tensor. Due to the fact that the kth power of the adjacency matrix gives the number of k-length paths connecting the vertex pairs in the network [13], we construct our multihop adjacency tensor $\mathcal{A}$ as follows.

1) *Exponentiation:* Given the adjacency matrix $\mathbf{A}$ of the network $\mathcal{G}$, we first get a series of matrices $\{\mathbf{A}, \mathbf{A}^2, \ldots, \mathbf{A}^{d_0}\}$, where $\mathbf{A}^i$ represents the ith exponentiation of $\mathbf{A}$ and contains a different view (i.e., ith hop) of node–node connections, and d_0 represents the number of total hops needed.
2) *Normalization:* For each matrix $\mathbf{A}^i$ in $\{\mathbf{A}, \mathbf{A}^2, \ldots, \mathbf{A}^{d_0}\}$, we normalize it as $\hat{\mathbf{A}}^i = \mathbf{D}_i^{-1/2}\mathbf{A}^i\mathbf{D}_i^{-1/2}$, where $\mathbf{D}_i = \text{diag}(\mathbf{A}^i\mathbf{1}_n)$ is the degree matrix.
3) *Combination:* By stacking $\hat{\mathbf{A}}^i$ in order, we obtain the adjacency tensor $\mathcal{A} \in \mathbb{R}_+^{n\times n\times d_0}$, where $\mathcal{A}(:,:,i) = \hat{\mathbf{A}}^i$.

To sum up, Fig. 2 illustrates the construction process of $\mathcal{A}$.

Our proposed adjacency tensor resolves the aforementioned two issues of adjacency matrix from two aspects. On the one hand, our proposed adjacency tensor explicitly incorporates multihop information into its different frontal slices, which provides more conducive information than the adjacency matrix. On the other hand, as the allowed path length gets longer, the possibility of a node getting touched with other nodes based on the walks of that length becomes higher, and therefore the higher power of the adjacency matrix becomes denser. This directly increases the density of our proposed adjacency tensor $\mathcal{A}$.

Nevertheless, a straightforward but tricky problem is that how do we select the value of d_0. We expect that the adjacency tensor $\mathcal{A}$ contains as much information as possible. Thus, the number of hops d_0 should be set as $\text{diam}(\mathcal{G})$ such that $\mathcal{A}$ contains connectivity patterns of every possible node pair. However, in reality, the diameter of a network can be large, which brings difficulties in both computing $\mathcal{A}$ and storing $\mathcal{A}$. Fortunately, thanks to the *six degrees of separation* [38], on average, all people pairs can reach each other in no more than six hops in a social network. Although some large-scale networks have large diameters, the structural information in the first six hops is already considerable, which suggests the parameter setting of $d_0 = \min(\text{diam}(\mathcal{G}), 6)$.

B. ANTD Method

In this section, we derive the optimization problem of our proposed ANTD method. A fundamental assumption in network topology modeling is that, if two nodes share more similar communities, they are more likely to form an edge [26]. Following this assumption, we adopt the symmetric nonnegative matrix tri-factorization [39], which is an extension of the classical NMF, to model the edge generation process of a network. Specifically, suppose we have a nonnegative matrix $\mathbf{U} \in \mathbb{R}_+^{n\times k}$ which represents the community memberships of nodes. That is, each element u_{ij} in the learned $\mathbf{U}$ can be interpreted as the propensity that the ith node belongs to the jth community [23]. Moreover, suppose we have a nonnegative tensor $\mathcal{W} \in \mathbb{R}_+^{k\times k\times d_0}$, whose frontal slices represent different interaction patterns (similarities) between communities in terms of different hops. E.g., the corresponding elements in the learned $\mathcal{W}$ of communities "r/JapanTravel" and "r/IWantOut" would be relatively larger since these two communities are semantically similar, while the corresponding elements of "r/Photography" and "r/ChineseLanguage" would be relatively smaller since they are irrelevant. Then, following the assumption introduced at the beginning of this section, $u_{fp}w_{pq}^{(i)}u_{gq}$ can be interpreted as the contribution to the expected normalized number of edges between nodes v_f and v_g in the ith hop from c_p and c_q.[1] Summing over all possible community pairs (p, q), the expected normalized number of edges between nodes v_f and v_g in the ith hop can be computed as $\hat{a}_{fg}^{(i)} = \sum_{p,q=1}^{k} u_{fp}w_{pq}^{(i)}u_{gq} = \mathbf{u}^f\mathbf{W}^{(i)}\mathbf{u}^{gT}$. Obviously, the generated $\hat{a}_{fg}^{(i)}$ should be as identical as possible with $a_{fg}^{(i)}$ [40], which gives rise to the following objective function and its related optimization problem

$$\min_{\mathbf{U},\mathcal{W}} \left\|\mathcal{A} - [\![\mathcal{W}; \mathbf{U}, \mathbf{U}, \mathbf{I}_{d_0}]\!]\right\|_F^2 \quad \text{s.t. } \mathbf{U}, \mathcal{W} \geq 0$$

which is a special case of the nonnegative Tucker decomposition introduced in Section III-B. Essentially, the above problem performs symmetric nonnegative matrix tri-factorization $\|\mathbf{A}^{(i)} - \mathbf{U}\mathbf{W}^{(i)}\mathbf{U}^T\|_F^2$ slicewisely. Note that the above optimization problem aims to reconstruct the original adjacency tensor $\mathcal{A}$ by $\mathcal{W}$ and $\mathbf{U}$, i.e., it is a *decoder*. However, as pointed out in [14] and [15], an *encoder* component which projects the original network into the community interaction space with the aid of node representations is also important. In our case, the encoder component should map the adjacency tensor $\mathcal{A}$ to the community interaction tensor $\mathcal{W}$ with the aid of the community membership indicator $\mathbf{U}$, therefore the

[1]We assume that the underlying community engagement of nodes is invariant in different hops, hence we devise only one community membership matrix $\mathbf{U}$. However, the connection patterns of nodes in different hops are different, therefore, we devise d_0 different community interaction matrices, stacked to be $\mathcal{W} \in \mathbb{R}_+^{k\times k\times d_0}$.

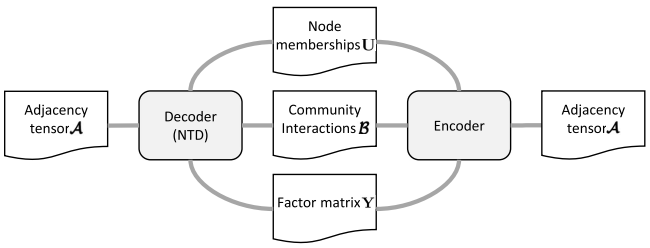


Fig. 3. Architecture of ANTD.

corresponding optimization problem can be described as

$$\min_{\mathbf{U},\mathcal{W}} \left\| [\![\mathcal{A}; \mathbf{U}^T, \mathbf{U}^T, \mathbf{I}_{d_0}]\!] - \mathcal{W} \right\|_F^2 \quad \text{s.t. } \mathbf{U}, \mathcal{W} \geq 0.$$

As we will discuss in Section V-C, the encoder component will greatly benefit the community detection process. Moreover, since the adjacency tensor $\mathcal{A}$ is constructed based on the sole information source $\mathcal{G}$, $\mathcal{A}$ might have strong dependence among its frontal slices. Correspondingly, frontal slices of the learned community interaction tensor $\mathcal{W}$ would also be highly correlated. To better capture the dependence that inherently exists in $\mathcal{W}$, we further factorize $\mathcal{W}$ into two low-rank factors $\mathcal{B} \in \mathbb{R}_+^{k\times k\times d}$ and $\mathbf{Y} \in \mathbb{R}_+^{d_0\times d}$ via $\mathcal{W} = \mathcal{B} \times_3 \mathbf{Y}$, where $d < d_0$ controls the degree of correlation among $\mathcal{W}$'s frontal slices: When $d = 1$, all frontal slices of $\mathcal{W}$ are proportional, but when $d = d_0$, the dependence in $\mathcal{W}$ may degenerate to nothing due to the trivial solution $\mathbf{Y} = \mathbf{I}_{d_0}$. Based on the above discussions, the optimization problem of our proposed ANTD method is given as

$$\begin{aligned} \min_{\mathbf{U},\mathbf{Y},\mathcal{B}} \quad & \left\| \mathcal{A} - [\![\mathcal{B}; \mathbf{U}, \mathbf{U}, \mathbf{Y}]\!] \right\|_F^2 \\ & + \lambda \left\| [\![\mathcal{A}; \mathbf{U}^T, \mathbf{U}^T, \mathbf{I}_{d_0}]\!] - [\![\mathcal{B}; \mathbf{I}_k, \mathbf{I}_k, \mathbf{Y}]\!] \right\|_F^2 \\ \text{s.t.} \quad & \mathbf{U}, \mathbf{Y}, \mathcal{B} \geq 0 \end{aligned} \tag{1}$$

where λ is used to balance the encoder and decoder. Fig. 3 illustrates the architecture of our proposed ANTD method.

C. Graph Regularized ANTD

Note that our proposed ANTD method is just a backbone, which can naturally impose regularizers to further improve community detection performance. We here introduce a variant of ANTD, namely ANTDg, which imposes a graph regularizer [41] on top of ANTD, leading to the following optimization problem:

$$\begin{aligned} \min_{\mathbf{U},\mathbf{Y},\mathcal{B}} \quad & \left\| \mathcal{A} - [\![\mathcal{B}; \mathbf{U}, \mathbf{U}, \mathbf{Y}]\!] \right\|_F^2 + \gamma \ \mathrm{Tr}(\mathbf{U}^T\mathbf{L}\mathbf{U}) \\ & + \lambda \left\| [\![\mathcal{A}; \mathbf{U}^T, \mathbf{U}^T, \mathbf{I}_{d_0}]\!] - [\![\mathcal{B}; \mathbf{I}_k, \mathbf{I}_k, \mathbf{Y}]\!] \right\|_F^2 \\ \text{s.t.} \quad & \mathbf{U}, \mathbf{Y}, \mathcal{B} \geq 0 \end{aligned}$$

where γ is a tunable parameter to adjust the importance of graph regularization and $\mathbf{L} = \mathbf{D}_1 - \mathbf{A}$ (recall $\mathbf{D}_1 = \mathrm{diag}(\mathbf{A}\mathbf{1}_n)$). We will report the performance of ANTDg in experiments as well.

V. Optimization and Theoretical Analysis

In this section, we first present an alternating optimization algorithm to solve (1). Then, we further theoretically analyze the utility of the encoder and the convergence as well as computational complexity of the optimization algorithm. Based on the above developments, we finally briefly introduce the optimization of ANTDg with its analysis on the convergence and computaional complexity.

A. Problem Reformulation

The first term in the objective function of (1) is a quartic polynomial function w.r.t. $\mathbf{U}$, which makes itself highly impossible to be convex w.r.t. $\mathbf{U}$ [26]. Besides, provided $\mathcal{B}$ is a unit super-diagonal tensor, this term also coincides with the objective function of INDSCAL [12], whose best solution is still an open problem. Therefore, (1) is difficult to solve. Alternatively, we introduce an auxiliary variable $\mathbf{Z} \in \mathbb{R}_+^{n\times k}$ and propose an equivalent problem as

$$\begin{aligned} \min_{\mathbf{U},\mathbf{Z},\mathbf{Y},\mathcal{B}} \quad & \left\| \mathcal{A} - [\![\mathcal{B}; \mathbf{U}, \mathbf{Z}, \mathbf{Y}]\!] \right\|_F^2 \\ & + \lambda \left\| [\![\mathcal{A}; \mathbf{U}^T, \mathbf{Z}^T, \mathbf{I}_{d_0}]\!] - [\![\mathcal{B}; \mathbf{I}_k, \mathbf{I}_k, \mathbf{Y}]\!] \right\|_F^2 \\ \text{s.t.} \quad & \mathbf{U}, \mathbf{Z}, \mathbf{Y}, \mathcal{B} \geq 0, \mathbf{U} = \mathbf{Z} \end{aligned}$$

whose objective function is now separately convex to every variable. By defining a penalty function $\|\mathbf{U} - \mathbf{Z}\|_F^2$ which describes the cost of violating the constraint $\mathbf{U} = \mathbf{Z}$ and transferring this constraint to the objective function, we arrive at our reformulated optimization problem

$$\begin{aligned} \min_{\mathbf{U},\mathbf{Z},\mathbf{Y},\mathcal{B}} \quad & \left\| \mathcal{A} - [\![\mathcal{B}; \mathbf{U}, \mathbf{Z}, \mathbf{Y}X]\!] \right\|_F^2 + \eta \|\mathbf{U} - \mathbf{Z}\|_F^2 \\ & + \lambda \left\| [\![\mathcal{A}; \mathbf{U}^T, \mathbf{Z}^T, \mathbf{I}_{d_0}]\!] - [\![\mathcal{B}; \mathbf{I}_k, \mathbf{I}_k, \mathbf{Y}]\!] \right\|_F^2 \\ \text{s.t.} \quad & \mathbf{U}, \mathbf{Z}, \mathbf{Y}, \mathcal{B} \geq 0 \end{aligned} \tag{2}$$

where η is another parameter to adjust the weight of the penalty term $\|\mathbf{U} - \mathbf{Z}\|_F^2$. In Section V-B, we will derive an efficient solution to the above problem.

B. Solution Method

Inspired by the optimization of NTD [37], we derive an alternating optimization algorithm for solving (2) as follows.

1) Updating $\mathbf{U}$*:* The corresponding subproblem of updating $\mathbf{U}$ is

$$\begin{aligned} \min_{\mathbf{U}} \quad & \|\mathbf{P}_\mathbf{U} - \mathbf{U}\mathbf{Q}_\mathbf{U}\|_F^2 + \lambda \left\| \mathbf{U}^T\mathbf{R}_\mathbf{U} - \mathbf{S}_\mathbf{U} \right\|_F^2 + \eta \|\mathbf{U} - \mathbf{Z}\|_F^2 \\ \text{s.t.} \quad & \mathbf{U} \geq 0 \end{aligned}$$

where $\mathbf{P}_\mathbf{U} = \mathcal{A}_{(1)}$, $\mathbf{Q}_\mathbf{U} = \mathcal{B}_{(1)}(\mathbf{Y}\otimes\mathbf{Z})^T$, $\mathbf{R}_\mathbf{U} = \mathcal{A}_{(1)}(\mathbf{I}_{d_0}\otimes\mathbf{Z}^T)^T$ and $\mathbf{S}_\mathbf{U} = \mathbf{I}_k\mathcal{B}_{(1)}(\mathbf{Y}\otimes\mathbf{I}_k)^T$. By introducing the Lagrangian multiplier $\mathbf{\Theta}_\mathbf{U}$, and setting the partial derivative of the resulting Lagrangian function w.r.t. $\mathbf{U}$ to 0, we get

$$\begin{aligned} \mathbf{\Theta}_\mathbf{U} = & -2\mathbf{P}_\mathbf{U}\mathbf{Q}_\mathbf{U}^T + 2\mathbf{U}\mathbf{Q}_\mathbf{U}\mathbf{Q}_\mathbf{U}^T + 2\lambda\mathbf{R}_\mathbf{U}\mathbf{R}_\mathbf{U}^T\mathbf{U} \\ & - 2\lambda\mathbf{R}_\mathbf{U}\mathbf{S}_\mathbf{U}^T + 2\eta\mathbf{U} - 2\eta\mathbf{Z}. \end{aligned}$$

According to the complementary slackness of KKT conditions [42], the optimal solution must satisfy $\mathbf{U} \circledast \mathbf{\Theta}_\mathbf{U} = 0$,

from which we can derive the updating rule for $\mathbf{U}$ as

$$\mathbf{U} \leftarrow \mathbf{U} \circledast \frac{\mathbf{P_U}\mathbf{Q_U}^T + \lambda \mathbf{R_U}\mathbf{S_U}^T + \eta \mathbf{Z}}{\mathbf{U}\mathbf{Q_U}\mathbf{Q_U}^T + \lambda \mathbf{R_U}\mathbf{R_U}^T\mathbf{U} + \eta \mathbf{U}}. \tag{3}$$

2) Updating $\mathbf{Z}$*:* The corresponding subproblem of updating $\mathbf{Z}$ is

$$\begin{aligned} \min_{\mathbf{Z}} \quad & \|\mathbf{P_Z} - \mathbf{Z}\mathbf{Q_Z}\|_F^2 + \lambda \left\|\mathbf{Z}^T\mathbf{R_Z} - \mathbf{S_Z}\right\|_F^2 + \eta \|\mathbf{Z} - \mathbf{U}\|_F^2 \\ \text{s.t.} \quad & \mathbf{Z} \geq 0 \end{aligned}$$

where $\mathbf{P_Z} = \boldsymbol{\mathcal{A}}_{(2)}$, $\mathbf{Q_Z} = \boldsymbol{\mathcal{B}}_{(2)}(\mathbf{Y} \otimes \mathbf{U})^T$, $\mathbf{R_Z} = \boldsymbol{\mathcal{A}}_{(2)}(\mathbf{I}_{d_0} \otimes \mathbf{U}^T)^T$ and $\mathbf{S_Z} = \mathbf{I}_k \boldsymbol{\mathcal{B}}_{(2)}(\mathbf{Y} \otimes \mathbf{I}_k)^T$. Similarly, by introducing the Lagrange multiplier $\boldsymbol{\Theta}_\mathbf{Z}$, and setting the partial derivative of the resulting Lagrangian function w.r.t. $\mathbf{Z}$ to 0, we get

$$\begin{aligned} \boldsymbol{\Theta}_\mathbf{Z} = & -2\mathbf{P_Z}\mathbf{Q_Z}^T + 2\mathbf{Z}\mathbf{Q_Z}\mathbf{Q_Z}^T + 2\lambda \mathbf{R_Z}\mathbf{R_Z}^T\mathbf{Z} \\ & - 2\lambda \mathbf{R_Z}\mathbf{S_Z}^T + 2\eta \mathbf{Z} - 2\eta \mathbf{U}. \end{aligned}$$

According again to the complementary slackness of the KKT conditions, the optimal solution must satisfy $\mathbf{Z} \circledast \boldsymbol{\Theta}_\mathbf{Z} = 0$, from which we can derive the updating rule for $\mathbf{Z}$ as

$$\mathbf{Z} \leftarrow \mathbf{Z} \circledast \frac{\mathbf{P_Z}\mathbf{Q_Z}^T + \lambda \mathbf{R_Z}\mathbf{S_Z}^T + \eta \mathbf{U}}{\mathbf{Z}\mathbf{Q_Z}\mathbf{Q_Z}^T + \lambda \mathbf{R_Z}\mathbf{R_Z}^T\mathbf{Z} + \eta \mathbf{Z}}. \tag{4}$$

3) Updating $\mathbf{Y}$*:* The corresponding subproblem of updating $\mathbf{Y}$ is

$$\min_{\mathbf{Y}} \ \|\mathbf{P_Y} - \mathbf{Y}\mathbf{Q_Y}\|_F^2 + \lambda \|\mathbf{R_Y} - \mathbf{Y}\mathbf{S_Y}\|_F^2 \quad \text{s.t. } \mathbf{Y} \geq 0$$

where $\mathbf{P_Y} = \boldsymbol{\mathcal{A}}_{(3)}$, $\mathbf{Q_Y} = \boldsymbol{\mathcal{B}}_{(3)}(\mathbf{Z} \otimes \mathbf{U})^T$, $\mathbf{R_Y} = \mathbf{I}_{d_0} \boldsymbol{\mathcal{A}}_{(3)}(\mathbf{Z}^T \otimes \mathbf{U}^T)^T$ and $\mathbf{S_Y} = \boldsymbol{\mathcal{B}}_{(3)}(\mathbf{I}_k \otimes \mathbf{I}_k)^T$. Following the same routine, by introducing the Lagrange multiplier $\boldsymbol{\Theta}_\mathbf{Y}$, and setting the partial derivative of the resulting Lagrangian function w.r.t. $\mathbf{Y}$ to 0, we get

$$\boldsymbol{\Theta}_\mathbf{Y} = -2\mathbf{P_Y}\mathbf{Q_Y}^T + 2\mathbf{Y}\mathbf{Q_Y}\mathbf{Q_Y}^T - 2\lambda \mathbf{R_Y}\mathbf{S_Y}^T + 2\lambda \mathbf{Y}\mathbf{S_Y}\mathbf{S_Y}^T.$$

According to the complementary slackness of the KKT conditions, we can derive the updating rule for $\mathbf{Y}$ as

$$\mathbf{Y} \leftarrow \mathbf{Y} \circledast \frac{\mathbf{P_Y}\mathbf{Q_Y}^T + \lambda \mathbf{R_Y}\mathbf{S_Y}^T}{\mathbf{Y}\mathbf{Q_Y}\mathbf{Q_Y}^T + \lambda \mathbf{Y}\mathbf{S_Y}\mathbf{S_Y}^T}. \tag{5}$$

4) Updating $\boldsymbol{\mathcal{B}}$*:* The corresponding subproblem of updating $\boldsymbol{\mathcal{B}}$ is

$$\begin{aligned} \min_{\boldsymbol{\mathcal{B}}} \quad & \left\|\boldsymbol{\mathcal{A}} - [\![\boldsymbol{\mathcal{B}}; \mathbf{U}, \mathbf{Z}, \mathbf{Y}]\!]\right\|_F^2 \\ & + \lambda \left\|[\![\boldsymbol{\mathcal{A}}; \mathbf{U}^T, \mathbf{Z}^T, \mathbf{I}_{d_0}]\!] - [\![\boldsymbol{\mathcal{B}}; \mathbf{I}_k, \mathbf{I}_k, \mathbf{Y}]\!]\right\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{\mathcal{B}} \geq 0. \end{aligned}$$

Applying the theorem introduced in our supplementary material, we can directly obtain the updating rule for $\boldsymbol{\mathcal{B}}$ as

$$\boldsymbol{\mathcal{B}} \leftarrow \boldsymbol{\mathcal{B}} \circledast \frac{(1+\lambda)[\![\boldsymbol{\mathcal{A}}; \mathbf{U}^T, \mathbf{Z}^T, \mathbf{Y}^T]\!]}{[\![\boldsymbol{\mathcal{B}}; \mathbf{U}^T\mathbf{U}, \mathbf{Z}^T\mathbf{Z}, \mathbf{Y}^T\mathbf{Y}]\!] + \lambda [\![\boldsymbol{\mathcal{B}}; \mathbf{I}_k, \mathbf{I}_k, \mathbf{Y}^T\mathbf{Y}]\!]}. \tag{6}$$

Based on the above analysis, we summarize the detailed optimization algorithm, which is outlined in Algorithm 1. The algorithm first constructs the adjacency tensor $\boldsymbol{\mathcal{A}}$ (line 1). It then initializes all factors $\mathbf{U}$, $\mathbf{Z}$, $\mathbf{Y}$, $\boldsymbol{\mathcal{B}}$ randomly and community sets $c_i = \varnothing$ for $1 \leq i \leq k$ (line 2). Afterward, it iteratively updates $\mathbf{U}$, $\mathbf{Z}$, $\mathbf{Y}$ and $\boldsymbol{\mathcal{B}}$ until the stopping criteria are met (lines 3–9). Finally, it partitions the network $\mathcal{G}$ by assigning each node $v_i \in \mathcal{V}$ into $c_{\arg\max_j u_{ij}}$ (lines 10–14).

Algorithm 1 Optimization Algorithm for ANTD

Input: Network $\mathcal{G}$, parameters λ, η, maximum number of iterations Φ, number of communities to be detected k.

Output: The k detected communities $\mathcal{C} = \{c_1, c_2, \ldots, c_k\}$ in the network $\mathcal{G}$.

1: Extract adjacency matrix $\mathbf{A}$ from $\mathcal{G}$ and construct adjacency tensor $\boldsymbol{\mathcal{A}}$ accordingly, as shown in Section IV-A;
2: Set the number of iterations $t = 0$ and $c_i = \varnothing$ for $1 \leq i \leq k$, and initialize $\mathbf{U}_t$, $\mathbf{Z}_t$, $\mathbf{Y}_t$ and $\boldsymbol{\mathcal{B}}_t$ randomly;
3: **while** $t < \Phi$ and not converged **do**
4: Update $\mathbf{U}_t$ according to (3);
5: Update $\mathbf{Z}_t$ according to (4);
6: Update $\mathbf{Y}_t$ according to (5);
7: Update $\boldsymbol{\mathcal{B}}_t$ according to (6);
8: $t \leftarrow t + 1$;
9: **end while**
10: **for all** $v_i \in \mathcal{V}$ **do**
11: Community membership: $\xi(v_i) \leftarrow \arg\max_j (u_t)_{ij}$;
12: Assign node v_i into its corresponding community as $c_{\xi(v_i)} \leftarrow c_{\xi(v_i)} \cup \{v_i\}$;
13: **end for**
14: **return** The k detected communities $\mathcal{C} = \{c_1, c_2, \ldots, c_k\}$ in the network $\mathcal{G}$.

C. Utility of the Encoder Component

We here analyze the utility of the encoder component $\|[\![\boldsymbol{\mathcal{A}}; \mathbf{U}^T, \mathbf{U}^T, \mathbf{I}_{d_0}]\!] - [\![\boldsymbol{\mathcal{B}}; \mathbf{I}_k, \mathbf{I}_k, \mathbf{Y}]\!]\|_F^2$. After (1) has been well solved, the following two equations hold approximately:

$$\boldsymbol{\mathcal{A}} \approx \boldsymbol{\mathcal{B}} \times_1 \mathbf{U} \times_2 \mathbf{U} \times_3 \mathbf{Y} = (\boldsymbol{\mathcal{B}} \times_3 \mathbf{Y}) \times_1 \mathbf{U} \times_2 \mathbf{U} \tag{7}$$

$$\boldsymbol{\mathcal{A}} \times_1 \mathbf{U}^T \times_2 \mathbf{U}^T \approx (\boldsymbol{\mathcal{B}} \times_3 \mathbf{Y}). \tag{8}$$

By substituting (7) into (8), we obtain an important relation

$$(\boldsymbol{\mathcal{B}} \times_3 \mathbf{Y}) \times_1 \left(\mathbf{U}^T\mathbf{U}\right) \times_2 \left(\mathbf{U}^T\mathbf{U}\right) \approx (\boldsymbol{\mathcal{B}} \times_3 \mathbf{Y})$$

which implies that $\mathbf{U}$ is implicitly forced to be orthogonal (i.e., this relation provides a necessary condition for $\mathbf{U}$'s orthogonality). According to [43], if $\mathbf{U}$ is simultaneously nonnegative and orthogonal, each row of $\mathbf{U}$ will contain only one positive element, indicating the community index that the corresponding node belongs to. Therefore, the introduced encoder component greatly benefits the task of community detection, because we expect the node-community memberships to be very clear.

D. Convergence and Complexity Analysis

First, we analyze the convergence of Algorithm 1. As our proposed multiplicative updating rules in Algorithm 1 are directly derived from their NMF counterparts, the monotonic convergence analysis in [44] can be applied to our case as well. Therefore, we omit the proof here and conclude that Algorithm 1 converges.

Next, we analyze the computational complexity of Algorithm 1 step by step. Recall that n and m denote the number of nodes and edges in the network $\mathcal{G}$ respectively, k is the number of communities to be detected, d_0 denotes

the number of hops needed, and d represents the number of frontal slices of $\boldsymbol{\mathcal{B}}$. We assume that $k \ll n$, $d_0 \ll n$ and $d < d_0$, which generally hold in reality. Constructing $\boldsymbol{\mathcal{A}}$ needs to compute $\{\mathbf{A}^1, \mathbf{A}^2, \ldots, \mathbf{A}^{d_0}\}$ and their corresponding matrix normalization first, which can be computed in $\mathcal{O}(d_0 mn)$ time, using the fast Boolean square matrix power algorithm [45], and $\mathcal{O}(d_0 n^2)$ time, respectively. In addition, updating $\mathbf{U}$ needs to calculate $\boldsymbol{\mathcal{B}}_{(1)}(\mathbf{Y} \otimes \mathbf{Z})^T$, $\boldsymbol{\mathcal{A}}_{(1)}(\mathbf{I}_{d_0} \otimes \mathbf{Z}^T)^T$ and $\mathbf{I}_k \boldsymbol{\mathcal{B}}_{(1)}(\mathbf{Y} \otimes \mathbf{I}_k)^T$ first. By applying the fast algorithm introduced in [46], these three terms can be efficiently computed in $\mathcal{O}(d_0 k^2 n)$, $\mathcal{O}(d_0 k n^2)$ and $\mathcal{O}(d d_0 k^2 + d k^3)$ time, respectively. Besides, computing the multiplicative updating rule in (3) costs $\mathcal{O}(d_0 k n^2)$ time, thus the total time complexity of updating $\mathbf{U}$ is $\mathcal{O}(d_0 k n^2)$. Following the same routine, updating $\mathbf{Z}$ and $\mathbf{Y}$ cost $\mathcal{O}(d_0 k n^2)$ and $\mathcal{O}(d_0 k n^2 + d_0 d n^2)$ time, respectively. Moreover, the computational complexity of updating $\boldsymbol{\mathcal{B}}$ is $\mathcal{O}(d_0 k n^2)$. As a result, the time complexity of Algorithm 1 is $\mathcal{O}(d_0 mn + \Phi_0(d_0 k n^2 + d_0 d n^2))$, where Φ_0 is the total number of iterations. It is noted that, the computational complexity of the iterative optimization process of Algorithm 1 still scales quadratically instead of cubically with the number of nodes n. This implies that, compared to many existing NMF-based methods, like NNSED [14] which takes $\mathcal{O}(n^2 k)$ time, although our proposed ANTD method adopts nonnegative tensor decomposition to process the higher-order adjacency tensor, the order of magnitude of its computational complexity has not been increased, which guarantees its efficiency.

E. Optimization and Theoretical Analysis of ANTDg

Following the derivation routine in Section V-B, all we need to modify Algorithm 1 to the optimization of ANTDg is adding $\gamma \mathbf{A}\mathbf{U}$ and $\gamma \mathbf{D}_1\mathbf{U}$ to the numerator and denominator of the updating rule of $\mathbf{U}$ (i.e., (3)), respectively. Moreover, the convergence and computational complexity of ANTD are easily carried over to ANTDg as the resulting algorithm is also Lee-Seung type and the complexity of the additional computations is of $\mathcal{O}(n^2 k)$. It is remarked that, other regularizers can also be easily incorporated into ANTD, with minor modifications to its updating rules.

VI. Experiments

In this section, we evaluate our proposed methods on a variety of real-world benchmark networks. The source code of our proposed methods is implemented in MATLAB 2020b and publicly available.[2] All experiments run on a Ubuntu server with 3.70-GHz i9-10900K CPU, 128-GB main memory.

A. Comparative Methods

We compare our methods with three categories of state-of-the-art community detection methods as follows.

• ***Structural Community Detection Without Ground-Truth Communities*:** Ten community detection methods are compared on networks without ground-truth communities, including Betweenness [3], Fast-Greedy [7], InfoMap [8], Label-Prop [47], Leading-EV [17], WalkTrap [48], Watset [49], EdMot [10], SBM [50], and Belief [51]. All these methods can automatically detect an optimal number of communities.

• ***One-Stage Community Detection With Ground-Truth Communities*:** Eleven one-stage community detection methods are compared on networks with ground-truth communities, including ten NMF-based methods of NMF [9], ONMF [52], BNMF [21], NNSED [14], DANMF [15], HPNMF [26], AANMF [25], SymmNMF [22], SCNMF [53], and PGS [54], and also one tensor-based method of GraphFuse [55]. All these methods can directly get network partition from the learned node representations.

• ***Two-Stage Community Detection With Ground-Truth Communities*:** Six two-stage community detection methods are compared on networks with ground-truth communities, including two tensor-based multiview clustering methods of WTNNM [56] and CGL [57], and four network embedding methods of MNMF [28], RandNE [58], NodeSketch [59], and BoostNE [60]. All these methods need to conduct clustering on the learned node representations to partition the network.

For all tensor-based methods, we take the adjacency tensor $\boldsymbol{\mathcal{A}}$ as their inputs. In addition, we implement pruned versions of ANTD and ANTDg, namely **ANTDp** and **ANTDgp**, which ignore their encoder components, for ablation study. ANTDg and ANTDgp are only involved in the task of community detection with ground-truth communities.

B. Parameter Settings

For ANTD and its three variants, we set $d = 1$ (resp. search d in $\{1, 2, 3\}$) on the task of community detection without (resp. with) ground-truth communities, and on all tasks, we tune λ and η (and γ if graph regularized) in the search grid $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$, which is kept the same as in [15]. Besides, we set the maximum number of iterations $\Phi = 2000$, and stop the optimization process when the change in the loss function is less than the floating-point relative accuracy (i.e., `eps` in MATLAB). Moreover, we set $d_0 = \min(\text{diam}(\mathcal{G}), 6)$ for these four methods as suggested in Section IV-A.

For all network embedding methods, we set the dimensionality of node representations as 64, which is the default setting in the literature [14], [15], [29]. Besides, since the learned node representations from all these network embedding methods cannot be directly used to extract community structure, we conduct post-k-means clustering on the learned embeddings to get a network partition.

For the second group of methods of DANMF, HPNMF, SCNMF, PGS, and Graphfuse, we tune each of their parameters in the search grid $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$. Besides, for PGS, we tune the thresholding parameter (i.e., c in their article) in $\{\texttt{eps}, 2, 4, 6, 8\}$, and for DANMF, we set the layer size as $n \to 256 \to 128 \to k$ and the maximum number of pretraining iterations as 1000.

[2]https://github.com/Kwan1997/ANTD

TABLE I
STATISTICS OF DOLPHINS, NET-SCIENCE, AND WIKI-VOTE

Statistic	Dolphins	Net-Science	Wiki-Vote
# of Nodes	62	379	889
# of Edges	159	914	2,914
Diameter	8	17	13

TABLE II
DEFINITIONS OF MODULARITY, PERMANENCE, AND COVERAGE

Metric	Definition
Modularity [7]	$\frac{1}{2m}\sum_{i,j=1}^{n}\left(a_{ij}-\frac{d_{\mathcal{G}}(i)d_{\mathcal{G}}(j)}{2m}\right)\delta\left(c_{\xi(v_i)},c_{\xi(v_j)}\right)$
Permanence [18]	$\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\lvert N_{\mathcal{G}}(v_i)\cap c_{\xi(v_i)}\rvert}{d_{\mathcal{G}}(v_i)\max_{l\neq\xi(v_i)}\lvert N_{\mathcal{G}}(v_i)\cap c_l\rvert}+\frac{\lvert\mathcal{E}_{c_{\xi(v_i)}}\rvert}{\binom{\lvert N_{\mathcal{G}}(v_i)\cap c_{\xi(v_i)}\rvert}{2}}-1\right)$
Coverage [62]	$\sum_{i=1}^{k}\frac{\lvert\mathcal{E}_{c_i}\rvert}{m}$

For the two tensor-based multiview clustering methods of WTNNM and CGL, we tune each of their parameters in the search grid $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$. Besides, similarly, we conduct post spectral clustering on the learned similarity graphs of these two methods for a network partition.

For all the other methods, they either do not have any parameter to be tuned, or the number of their tunable parameters is larger than four. For the latter ones, we set their parameters as their original authors or other researchers suggested.

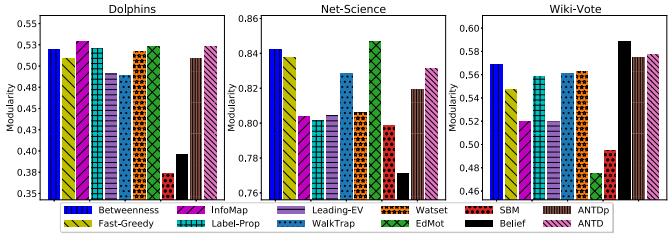


Fig. 4. Modularity evaluations on three datasets.

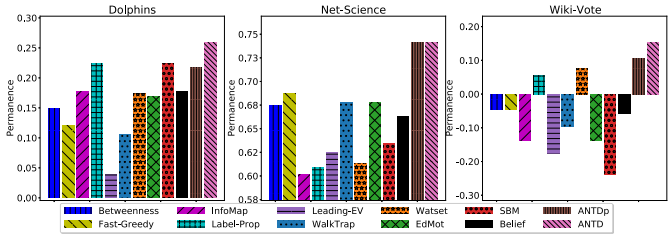


Fig. 5. Permanence evaluations on three datasets.

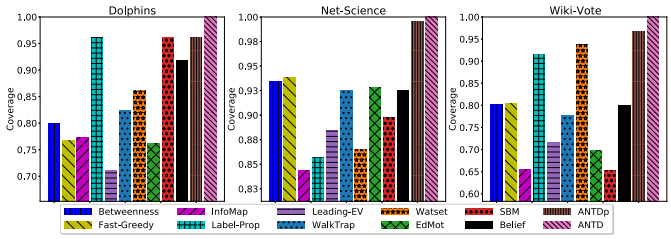


Fig. 6. Coverage evaluations on three datasets.

C. Community Detection without Ground-Truth Communities

Datasets: We use three real-world networks,[3] i.e., Dolphins, Net-Science, and Wiki-Vote, whose statistics are summarized in Table I. These networks do not have ground-truth communities, but are very classical benchmarks which are widely evaluated in the literature [26], [27], [61].

Evaluation Metrics: Given no ground-truth communities, we measure the community quality using structural information. We employ three widely used "unsupervised" community quality metrics, namely modularity [7], permanence [18], and coverage [62], as defined in Table II. In Table II, the indicator function $\delta(c_{\xi(v_i)}, c_{\xi(v_j)}) = 1$ if and only if $c_{\xi(v_i)} = c_{\xi(v_j)}$, or 0 otherwise. The larger these values, the better community results. Note that modularity and permanence can be negative, while coverage is always positive. As comparative methods can automatically determine the best number of communities, we vary k for ANTD and ANTDp from 2 to 30 with step size 1 for fairness. For all methods, we report the best results under all possible parameter combinations.

Exp-1: Community Quality Evaluations Without Ground-Truth Communities in Terms of Modularity, Permanence, and Coverage: We evaluate the quality of different community detection algorithms over three networks without ground-truth communities. Based on the results in Figs. 4–6, we have the following experimental observations.

• ***ANTD Is Effective*:** In terms of modularity, our proposed ANTD method achieves a relatively high performance, ranking in the top four among 12 methods. In terms of permanence and coverage, our proposed ANTD method consistently outperforms all the other methods. Moreover, in terms of coverage, our ANTD method consistently achieves full marks on all the networks. These evidences significantly demonstrate the effectiveness of ANTD.

• ***Encoder Makes Sense*:** For all the three metrics, ANTD consistently outperforms its pruned version ANTDp on all the networks. This phenomenon shows that the autoencoder-like architecture is indeed beneficial for improving community detection performance.

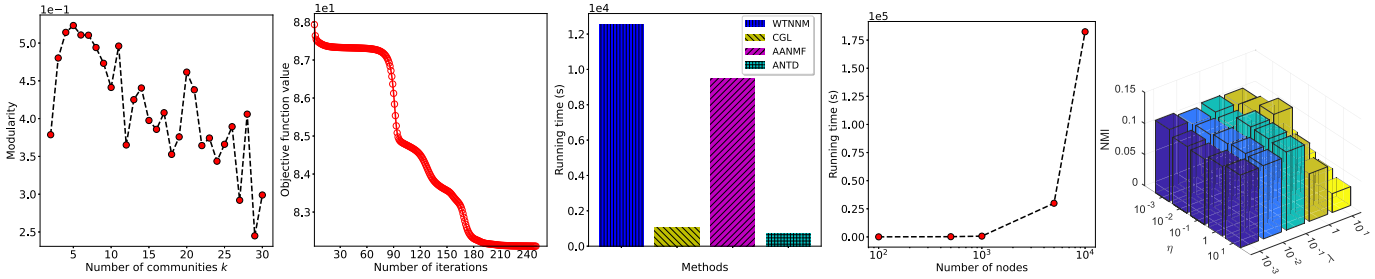
Exp-2: Sensitivity Analysis on k: We analyze the sensitivity of ANTD in terms of its modularity performance w.r.t. k. The experimental result on the Dolphins dataset is shown in Fig. 7(a). As observed, with k increases, the modularity shows an overall trend of first increasing and then decreasing, which is in accord with our expectation. It is thus suggested to set k to a medium value when k is not known *a priori*.

D. Community Detection with Ground-Truth Communities

Datasets: We use 12 widely used datasets of real-world networks with ground-truth communities.[4] Table III reports

[3] http://networkrepository.com

[4] https://snap.stanford.edu/data/index.html, https://linqs.soe.ucsc.edu/data and http://mlg.ucd.ie/aggregation/index.html



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

Fig. 7. Sensitivity, convergence, running time, and scalability analysis of ANTD. (a) Sensitivity of k on Dolphins. (b) Objective function value on Cornell. (c) Running time on LFR-1k. (d) Scalability on five networks. (e) Sensitivity of η, λ on Cornell.

TABLE III
STATISTICS OF 12 REAL-WORLD NETWORKS WITH GROUND-TRUTH COMMUNITIES

Statistic	Cora	Citeseer	Email	Texas	Cornell	Washington	Wisconsin	Football	PoliticsIE	PoliticsUK	Rugby	Olympics
# of Nodes	2,708	3,327	1,005	187	195	230	265	248	348	419	854	464
# of Edges	5,429	4,732	16,706	328	304	446	530	2,645	12,567	19,950	22,861	7,787
Density	0.148%	0.086%	3.311%	1.886%	1.607%	1.694%	1.515%	8.636%	20.814%	22.782%	6.277%	7.249%
# of Communities	7	6	42	5	5	5	5	20	7	5	15	28
Diameter	19	28	7	8	8	8	8	4	3	3	4	5

TABLE IV
STATISTICS OF FIVE SYNTHETIC NETWORKS

Statistic	LFR-100	LFR-500	LFR-1k	LFR-5k	LFR-10k
# of Nodes	100	500	1,000	5,000	10,000
# of Edges	240	1,090	2,247	11,277	23,187
# of Communities	6	34	60	310	614
Diameter	7	11	11	15	16

TABLE V
DEFINITIONS OF PURITY, F-SCORE, AND NMI

Metric	Definition
Purity [16]	$\frac{1}{n}\sum_{i=1}^{k}\max_{1\le j\le l}\lvert c_i \cap s_j\rvert$
F-score [16]	$\frac{2tp}{2tp+fp+fn}$
NMI [16]	$\frac{I(\mathcal{C},\mathcal{S})}{(H(\mathcal{C})+H(\mathcal{S}))/2}$

graph statistics of these real-world datasets, where the density is computed by $m/\binom{n}{2}$. Besides, we also generate five synthetic networks with ground-truth communities for efficiency and scalability evaluations. Specifically, we use the LFR benchmark toolkit [63] to generate five synthetic networks with different number of nodes varied in 100, 500, 1000, 5000, and 10 000, and we set mixing parameter as 0.2, average degree as 5, maximum degree as 25, degree distribution exponent as 2, community size distribution exponent as 1, and community size bounds as #nodes/50 and #nodes/20. Table IV shows the statistics of these generated synthetic networks.

Evaluation Metrics: We employ three evaluation metrics of purity, F-score, and NMI [16], as defined in Table V. In Table V, $\mathcal{S} = \{s_1, s_2, \ldots, s_l\}$ denotes the set of l ground-truth communities, $I(\mathcal{C},\mathcal{S}) = \sum_{i=1}^{k}\sum_{j=1}^{l}(|c_i \cap s_j|/n)\log(n|c_i \cap s_j|/|c_i||s_j|)$ is the mutual information between $\mathcal{C}$ and $\mathcal{S}$, $H(\mathcal{C}) = -\sum_{i=1}^{k}(|c_k|/n)\log(|c_k|/n)$ denotes the entropy of $\mathcal{C}$ (and similar applies to $H(\mathcal{S})$), and tp, fp, fn denote the number of true positive, false positive and false negative hits obtained by the pair confusion matrix of $\{\kappa(v_1), \kappa(v_2), \ldots, \kappa(v_n)\}$ and $\{\xi(v_1), \xi(v_2), \ldots, \xi(v_n)\}$, respectively, where $\kappa(v) \in \{1, 2, \ldots, l\}$ denotes the index of the ground-truth community containing v. For both metrics, the larger values, the better community results. For all the networks in Table III, we set the number of communities to be detected k as the same as the number of ground-truths l. We report the average results of different algorithms on ten runs under the best parameter combination.

Exp-3: Community Quality Evaluations With Ground-Truth Communities in Terms of Purity, F-Score and NMI: We evaluate the community quality of different algorithms over 12 networks with ground-truth communities. Based on the results in Tables VI–VIII, where the shaded (resp. boldface) number represents the top-three (resp. best) performance value respectively, we have the following observations.

• ***Our Proposed Methods Are Effective*:** In terms of these three metrics, we can observe that in most cases, at least one of the four versions of ANTD can achieve top-three performance, under comparisons with 17 state-of-the-art competitors. Besides, ANTDg strikes the best performance the most times. These evidences fully show the effectiveness of our proposed methods. Moreover, we can find that except on the Cora, Citeseer, Washington, and Wisconsin networks, ANTDg consistently performs better than other tensor-based methods with our adjacency tensor as inputs, indicating that in

TABLE VI
PURITY EVALUATIONS ON 12 REAL-WORLD DATASETS

Method	Cora	Citeseer	Texas	Cornell	Washington	Wisconsin	Email	Football	PoliticsIE	PolitcsUK	Rugby	Olympics
NMF [9]	0.4884	0.3461	0.6567	0.4400	0.5726	0.5204	0.6821	0.8206	0.8963	0.9605	0.8338	0.8248
ONMF [52]	0.4919	0.3427	0.6364	0.4436	0.5730	0.5268	0.6948	0.8097	0.8750	0.9483	0.8308	0.8104
BNMF [21]	0.4865	0.3351	0.5995	0.4574	0.5352	0.5094	0.4105	0.4486	0.8954	0.9665	0.8094	0.5259
NNSED [14]	0.4743	0.3569	0.6481	0.4436	0.5770	0.5200	0.6949	0.7976	0.9075	0.9593	0.8335	0.8045
DANMF [15]	0.5482	0.3881	0.6652	0.4621	0.6074	0.5362	0.6696	0.8028	0.9078	0.9620	0.8481	0.8421
HPNMF [26]	**0.5879**	0.4415	0.5882	0.4503	0.5357	0.5158	0.7068	**0.8761**	0.9402	0.9751	0.8599	0.9173
AANMF [25]	0.4219	0.2978	0.6380	0.4621	0.6213	0.5336	0.4920	0.5838	0.7244	0.8287	0.7018	0.6544
SymmNMF [22]	0.5293	0.3625	0.5930	0.4333	0.5896	0.5411	0.6920	0.8364	0.9239	0.9593	0.8262	0.8575
SCNMF [53]	0.5662	0.4134	**0.6754**	0.4487	0.6074	0.5521	0.7061	0.8615	0.9368	0.9713	0.8429	0.9032
PGS [54]	0.5511	**0.4808**	0.5802	0.4477	0.5487	0.5004	0.6777	0.6235	0.8632	0.9560	**0.8638**	0.8894
GraphFuse [55]	0.3618	0.3581	0.6064	0.4538	0.5452	0.4860	0.6277	0.8215	0.9049	0.9632	0.8329	0.8464
WTNNM [56]	0.5529	0.3801	0.5882	0.4462	0.6426	0.5264	0.6756	0.7988	0.9037	0.9545	0.8091	0.8397
CGL [57]	0.4918	0.4046	0.6310	0.4872	**0.6478**	**0.5623**	0.6379	0.8259	0.8649	0.9498	0.8430	0.8387
MNMF [28]	0.5151	0.3446	0.6102	0.4692	0.6226	0.5200	0.4884	0.6498	0.8052	0.9419	0.8269	0.6944
RandNE [58]	0.3667	0.2620	0.5535	0.4328	0.5157	0.4977	0.4378	0.7502	0.8851	0.9636	0.8265	0.7479
NodeSketch [59]	0.3343	0.2603	0.5636	0.4282	0.5978	0.4842	0.3469	0.4065	0.8239	0.9258	0.6711	0.5158
BoostNE [60]	0.3928	0.2450	0.5551	0.4323	0.4878	0.4691	0.4338	0.6753	0.8658	0.9596	0.8222	0.7477
ANTD	0.3195	0.2971	0.6588	0.4862	0.6078	0.5075	0.7127	0.8664	0.9382	0.9737	0.8459	0.9119
ANTDp	0.3146	0.2309	0.6449	0.4908	0.6013	0.5008	0.7085	0.8611	0.9394	0.9727	0.8487	0.9259
ANTDg	0.4034	0.3387	0.6636	0.4913	0.6209	0.5068	**0.7184**	0.8753	**0.9474**	**0.9792**	0.8547	0.9192
ANTDgp	0.3132	0.2446	0.6535	**0.5056**	0.6252	0.5079	0.7105	0.8660	0.9408	0.9725	0.8546	**0.9289**

TABLE VII
F-SCORE EVALUATIONS ON 12 REAL-WORLD DATASETS

Method	Cora	Citeseer	Texas	Cornell	Washington	Wisconsin	Email	Football	PolitcsIE	PolitcsUK	Rugby	Olympics
NMF [9]	0.3181	0.2868	0.5533	0.3302	0.4415	0.3563	0.4904	0.7457	0.7096	0.7224	0.5123	0.7224
ONMF [52]	0.3184	0.2825	0.5389	0.3373	0.4271	0.3533	0.5101	0.7353	0.6717	0.7626	0.5153	0.6850
BNMF [21]	0.3252	0.2251	0.4786	0.3117	0.3602	0.3134	0.3407	0.4169	0.9169	0.9134	0.6202	0.5228
NNSED [14]	0.3193	0.2796	0.5540	0.3457	0.4409	0.3647	0.5397	0.7196	0.7117	0.7443	0.5318	0.6665
DANMF [15]	0.4126	0.2956	0.5702	0.4209	0.5072	0.4332	0.5016	0.7051	0.7285	0.8516	0.6496	0.7813
HPNMF [26]	**0.4345**	0.3143	0.5373	0.4334	0.4796	0.4771	0.5208	0.8013	0.9067	**0.9835**	0.6756	0.8210
AANMF [25]	0.3075	0.2946	0.5453	0.3659	0.5048	0.3829	0.2269	0.3294	0.6559	0.7765	0.5028	0.4419
SymmNMF [22]	0.3518	0.2860	0.5160	0.3399	0.4669	0.3690	0.4573	0.7580	0.6954	0.7205	0.5094	0.7444
SCNMF [53]	0.4144	0.3164	**0.5998**	0.4187	0.5053	0.4552	0.5070	0.7835	0.8722	0.9698	0.7185	0.8079
PGS [54]	0.4020	**0.3478**	0.5400	0.4318	0.4842	0.4818	0.5059	0.5033	0.8816	0.9585	**0.7605**	0.7726
GraphFuse [55]	0.3101	0.3030	0.5400	0.4278	0.4846	0.4825	0.3645	0.7599	0.8630	0.9385	0.5765	0.7639
WTNNM [56]	0.3661	0.3077	0.3630	0.2903	0.3967	0.4826	0.4656	0.6842	0.7446	0.9061	0.5053	0.6687
CGL [57]	0.3628	0.3143	0.3394	0.3049	0.3609	0.3182	0.3414	0.7013	0.6492	0.7960	0.5333	0.6202
MNMF [28]	0.3661	0.2937	0.4348	0.3409	0.4899	0.3564	0.2594	0.4749	0.7564	0.8810	0.5248	0.5095
RandNE [58]	0.2130	0.1949	0.3639	0.3016	0.3440	0.2992	0.2509	0.5727	0.8862	0.9529	0.5943	0.5391
NodeSketch [59]	0.1958	0.1878	0.3211	0.2633	0.3356	0.2724	0.1334	0.2053	0.5836	0.6945	0.3605	0.3063
BoostNE [60]	0.3030	0.2991	0.3917	0.3342	0.4095	0.3771	0.2196	0.4298	0.8523	0.9440	0.5943	0.4614
ANTD	0.2901	0.2915	0.5603	0.4311	0.5262	0.4826	0.5655	0.7958	0.8888	0.8892	0.5670	0.8310
ANTDp	0.3030	0.2953	0.5835	0.4335	0.5162	0.4720	0.5503	0.7903	0.8171	0.8645	0.5496	0.8403
ANTDg	0.3252	0.3021	0.5763	0.4333	0.5368	**0.4830**	**0.5768**	**0.8140**	**0.9427**	0.9822	0.6906	**0.8577**
ANTDgp	0.3026	0.2949	0.5865	**0.4403**	**0.5393**	0.4713	0.5662	0.7978	0.8749	0.9070	0.5904	0.8555

our scheme the adjacency tensor is utilized in a more effective way.

• ***Encoder Is Not Omnipotent*:** Occasionally, the pruned versions of ANTD and ANTDg perform better than their originals. This suggests that the autoencoder-like architecture is not always conducive, although it gives better performance in most cases.

• ***Graph Regularizer Makes Sense*:** We can find that the performance of ANTDg is better than ANTD in nearly all cases, reflecting that the performance of ANTD can be further improved by incorporating a graph regularizer.

• ***Network Embedding Methods Are Not Effective*:** As observed, in terms of NMI, three network embedding methods RandNE, NodeSketch, and BoostNE perform not as good as expected on the first seven networks. The reason is that they are not community detection oriented, although they can encode higher-order node relations into embeddings. Besides, these methods have to conduct a post k-means clustering to partition the network. This two-stage methodology may also affect the performance of community detection.

• ***There Is No Free Lunch* [64]:** Although our proposed methods perform fairly well in most cases, their performance deteriorates on the Cora and Citeseer networks, which may be partly due to the extreme sparsity of these networks as shown in Table III. This shows that different methods may be suitable for different scenarios, and we suggest to use our proposed methods in relatively dense networks.

Exp-4: Convergence Evaluations: We conduct convergence evaluations for ANTD. Fig. 7(b) reports the objective function value curve of ANTD on the Cornell network with $\eta = \lambda = 0.1$ and $d = 2$. As observed, the objective function value monotonically decreases, which validates our theoretical analysis. Besides, the convergence speed of ANTD is not very fast: The objective function value first decreases slowly, but

TABLE VIII
NMI EVALUATIONS ON 12 REAL-WORLD DATASETS

Method	Cora	Citeseer	Texas	Cornell	Washington	Wisconsin	Email	Football	PoliticsIE	PoliticsUK	Rugby	Olympics
NMF [9]	0.2557	0.1214	0.2138	0.0993	0.1221	0.0782	0.6893	0.8401	0.7429	0.7214	0.6343	0.8304
ONMF [52]	0.2635	0.1213	0.2030	0.1097	0.1177	0.0846	0.7018	0.8377	0.7188	0.7187	0.6304	0.8208
BNMF [21]	0.2586	0.0646	0.1265	0.0869	0.0946	0.0686	0.5609	0.6548	0.8377	0.8728	0.6827	0.7276
NNSED [14]	0.2329	0.1166	0.2137	0.1064	0.1208	0.0919	0.7036	0.8292	0.7514	0.7294	0.6367	0.8163
DANMF [15]	0.3683	0.1463	0.2142	0.1254	0.1538	0.0874	0.6813	0.8179	0.7630	0.8057	0.6764	0.8537
HPNMF [26]	0.3524	0.1439	0.1094	0.0829	0.0754	0.0807	0.7014	0.8815	0.8480	0.9356	0.6975	0.9107
AANMF [25]	0.2021	0.1090	0.1971	0.1269	0.1776	0.0990	0.5276	0.6408	0.5586	0.6206	0.5591	0.7031
SymmNMF [22]	0.3262	0.1338	0.1476	0.0950	0.1333	0.1098	0.6917	0.8505	0.7698	0.7151	0.6254	0.8584
SCNMF [53]	0.3714	0.1786	**0.2384**	0.1108	0.1528	0.1056	0.7035	0.8685	0.8456	0.9153	0.6954	0.8992
PGS [54]	**0.3884**	**0.2364**	0.0854	0.0855	0.0835	0.0610	0.6690	0.7449	0.7948	0.8722	**0.7495**	0.8799
GraphFuse [55]	0.1337	0.0936	0.1539	0.0846	0.1153	0.0498	0.6427	0.8594	0.8359	0.8766	0.6645	0.8763
WTNNM [56]	0.3367	0.1273	0.1466	0.1008	0.1972	0.0971	0.6689	0.8221	0.7530	0.8101	0.6052	0.8406
CGL [57]	0.2682	0.1423	0.1512	0.1082	0.1912	**0.1103**	0.6324	0.8288	0.7152	0.7375	0.6578	0.8379
MNMF [28]	0.3176	0.1501	0.1638	0.1213	0.1855	0.0814	0.5395	0.6942	0.6431	0.7822	0.6366	0.7207
RandNE [58]	0.0932	0.0180	0.0857	0.0589	0.0799	0.0641	0.4816	0.7970	0.8042	0.8739	0.6534	0.7844
NodeSketch [59]	0.0739	0.0201	0.1063	0.0558	0.1324	0.0557	0.3449	0.4704	0.5844	0.6279	0.4547	0.5471
BoostNE [60]	0.1802	0.0535	0.0757	0.0685	0.0754	0.0682	0.5229	0.7473	0.7851	0.8794	0.6615	0.7830
ANTD	0.0719	0.0684	0.2197	0.1351	0.1853	0.0601	0.7185	0.8813	0.8412	0.8570	0.6606	0.9133
ANTDp	0.0430	0.0228	0.1952	0.1349	0.1759	0.0570	0.7120	0.8792	0.8429	0.8180	0.6604	0.9219
ANTDg	0.2073	0.0954	0.2223	**0.1369**	0.2035	0.0645	**0.7186**	**0.8969**	**0.9008**	**0.9488**	0.7028	0.9195
ANTDgp	0.0394	0.0335	0.1969	0.1363	**0.2047**	0.0616	0.7148	0.8876	0.8561	0.8415	0.6742	**0.9231**

then drops rapidly, and finally steadily converges to a local optimum. It is also noted that the objective function value of ANTD is relatively small (i.e., dozens), which is due to the fact that the frontal slices of $\mathcal{A}$ are all normalized adjacency matrices. This suggests that we should not set large values for the parameters of ANTD.

Exp-5: Efficiency and Scalability Evaluations: We conduct efficiency and scalability evaluations for ANTD. For all experiments here, we set the number of iterations as 2000 and all tunable parameters as 0.1 (except $d = 2$ for ANTD) for all adopted methods.

First, we evaluate the running time (i.e., the time of the whole community detection process) of ANTD. Fig. 7(c) reports the running time of WTNNM, CGL, AANMF, and ANTD on the LFR-1k network. As observed, ANTD runs much faster than WTNNM and AANMF, demonstrating the efficiency of ANTD to some extent. Besides, ANTD consumes nearly the same time with CGL.

Next, we test the scalability of ANTD over five synthetic networks as shown in Fig. 7(d). As observed, with the grown size of networks, ANTD needs exponentially more time to detect all communities, reflecting a challenging task for tensor-based community detection over large-scale networks. Fortunately, our ANTD method takes around 8 h to detect all communities on the LFR-5k network, which is not slow. Note that for networks with more than 10 000 nodes, building the corresponding adjacency tensor will cause memory overflow. This is due to the limited memory of our local machine, therefore we here only consider networks with fewer than 10 000 nodes. Distributed optimization methods can be adopted to solve this problem, and we leave it as our future work.

Exp-6: Sensitivity Evaluations: We conduct sensitivity evaluations for ANTD. Specifically, we turn λ and η in the search grid $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$, while fixing $d = 2$, and report the community detection performance of ANTD under all possible parameter combinations on the Cornell network in terms of NMI in Fig. 7(e). As observed, in general, smaller λ tends to give better NMI. This is because the objective function of ANTD is of small order of magnitude, as we analyzed in Exp-4, and larger λ would therefore cause the Tucker decomposition under-fitting to $\mathcal{A}$ and eventually sacrifice the quality performance. In addition, ANTD is relatively nonsensitive to these parameters in the search grid $(\lambda, \eta) \in \{10^{-3}, 10^{-2}, 10^{-1}\} \times \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$, indicating a very stable performance in terms of NMI.

VII. CONCLUSION

In this article, we proposed a novel concept of adjacency tensor, which contains multihop topological information, and a novel tensor Tucker decomposition-based community detection method ANTD, which can effectively process the adjacency tensor, as well as a graph regularized version of ANTD namely ANTDg. To optimize ANTD, we derived an efficient iterative optimization algorithm. Our proposed optimization algorithm scales only quadratically with the number of nodes, which is the same as many existing NMF-based methods, although ANTD is a tensor decomposition-based method. Moreover, with minor modifications, the developed optimization algorithm for ANTD can be easily adapted to optimize ANTDg, with convergence and computational complexity carried over. To comprehensively test our proposed methods, we conducted extensive experiments on a variety of real-world benchmark networks. Comparative experimental results showed that our proposed methods manifested outstanding performance and outperformed the state-of-the-art community detection algorithms. For future work, we plan to investigate how to utilize multihop local manifold structure to regularize the detected communities, and how to adapt our method to large-scale networks by distributed optimization techniques.

REFERENCES

[1] X. Luo, Z. Liu, L. Jin, Y. Zhou, and M. Zhou, "Symmetric nonnegative matrix factorization-based community detection models and their convergence analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 3, pp. 1203–1215, Mar. 2021.

[2] D. Shi, L. Zhu, Y. Li, J. Li, and X. Nie, "Robust structured graph clustering," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4424–4436, Nov. 2019.

[3] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, Apr. 2002.

[4] Y. Li, C. Sha, X. Huang, and Y. Zhang, "Community detection in attributed graphs: An embedding approach," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 338–345.

[5] S. Deng *et al.*, "On deep learning for trust-aware recommendations in social networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 5, pp. 1164–1177, May 2017.

[6] M. Milan and M. Cannataro, "Statistical and network-based analysis of Italian COVID-19 data: Communities detection and temporal evolution," *Int. J. Environ. Res. Public Health*, vol. 17, no. 12, p. 4182, Jun. 2020.

[7] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 6, 2004, Art. no. 066111.

[8] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 4, pp. 1118–1123, Oct. 2008.

[9] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[10] P.-Z. Li, L. Huang, C.-D. Wang, and J.-H. Lai, "EdMot: An edge enhancement approach for motif-aware community detection," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 479–487.

[11] P.-Z. Li, L. Huang, C.-D. Wang, J.-H. Lai, and D. Huang, "Community detection by motif-aware label propagation," *ACM Trans. Knowl. Discovery Data*, vol. 14, no. 2, pp. 1–19, Apr. 2020.

[12] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[13] M. Newman, *Networks*. New York, NY, USA: Oxford Univ. Press, 2018.

[14] B.-J. Sun, H. Shen, J. Gao, W. Ouyang, and X. Cheng, "A non-negative symmetric encoder-decoder approach for community detection," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 597–606.

[15] F. Ye, C. Chen, and Z. Zheng, "Deep autoencoder-like nonnegative matrix factorization for community detection," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 1393–1402.

[16] T. Chakraborty, A. Dalmia, A. Mukherjee, and N. Ganguly, "Metrics for community analysis: A survey," *ACM Compt. Surv.*, vol. 50, no. 4, pp. 1–37, 2017.

[17] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 74, no. 3, 2006, Art. no. 036104.

[18] T. Chakraborty, S. Srinivasan, N. Ganguly, A. Mukherjee, and S. Bhowmick, "On the permanence of vertices in network communities," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 1396–1405.

[19] J. Leskovec, K. J. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 631–640.

[20] S. Fortunato and M. Barthélemy, "Resolution limit in community detection," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 1, pp. 36–41, 2007.

[21] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon, "Overlapping community detection using Bayesian non-negative matrix factorization," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 83, no. 6, 2011, Art. no. 066114.

[22] D. Kuang, C. Ding, and H. Park, "Symmetric nonnegative matrix factorization for graph clustering," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2012, pp. 106–117.

[23] Y. Zhang and D.-Y. Yeung, "Overlapping community detection via bounded nonnegative matrix tri-factorization," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 606–614.

[24] J. Yang and J. Leskovec, "Overlapping community detection at scale: A nonnegative matrix factorization approach," in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, 2013, pp. 587–596.

[25] F. Ye, S. Li, Z. Lin, C. Chen, and Z. Zheng, "Adaptive affinity learning for accurate community detection," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 1374–1379.

[26] F. Ye, C. Chen, Z. Wen, Z. Zheng, W. Chen, and Y. Zhou, "Homophily preserving community detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 2903–2915, Aug. 2020.

[27] F. Ye, C. Chen, Z. Zheng, R.-H. Li, and J. X. Yu, "Discrete overlapping community detection with pseudo supervision," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2019, pp. 708–717.

[28] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, "Community preserving network embedding," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 203–209.

[29] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 701–710.

[30] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.

[31] S. Cao, W. Lu, and Q. Xu, "GraRep: Learning graph representations with global structural information," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2015, pp. 891–900.

[32] X. Zhang, H. Liu, Q. Li, and X.-M. Wu, "Attributed graph clustering via adaptive graph convolution," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 4327–4333.

[33] X. Zhang, H. Liu, X.-M. Wu, X. Zhang, and X. Liu, "Spectral embedding network for attributed graph clustering," *Neural Netw.*, vol. 142, pp. 388–396, Oct. 2021.

[34] X. Zhang, J. Mu, H. Liu, and X. Zhang, "GraphNet: Graph clustering with deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 3800–3804.

[35] X. Zhang, J. Mu, H. Liu, X. Zhang, L. Zong, and G. Wang, "Graph clustering with graph capsule network," *Neural Comput.*, vol. 34, no. 5, pp. 1256–1287, Apr. 2022.

[36] X. Su *et al.*, "A comprehensive survey on community detection with deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 9, 2022, doi: 10.1109/TNNLS.2021.3137396.

[37] Y.-D. Kim and S. Choi, "Nonnegative tucker decomposition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[38] L. Zhang and W. Tu, "Six degrees of separation in online society," in *Proc. Web Sci. Conf.*, 2009, pp. 1–5.

[39] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 126–135.

[40] X. Wang, D. Jin, X. Cao, L. Yang, and W. Zhang, "Semantic community identification in large attribute networks," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 265–271.

[41] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2010.

[42] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.

[43] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2005, pp. 606–610.

[44] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. 14th Int. Conf. Neural Inf. Process. Syst.*, 2001, pp. 556–562.

[45] M. A. Razzaque, C. S. Hong, M. Abdullah-Al-Wadud, and O. Chae, "A fast algorithm to calculate powers of a Boolean matrix for diameter computation of random graphs," in *Proc. Int. Workshop Algorithms Comput.*, 2008, pp. 58–69.

[46] P. L. Fackler, "Algorithm 993: Efficient computation with Kronecker products," *ACM Trans. Math. Softw.*, vol. 45, no. 2, pp. 1–9, Jun. 2019.

[47] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, no. 3, 2007, Art. no. 036106.

[48] P. Pons and M. Latapy, "Computing communities in large networks using random walks," in *Proc. Int. Symp. Comput. Inf. Sci.*, 2005, pp. 284–293.

[49] D. Ustalov, A. Panchenko, C. Biemann, and S. P. Ponzetto, "Watset: Local-global graph clustering with applications in sense and frame induction," *Comput. Linguistics*, vol. 45, no. 3, pp. 423–479, Sep. 2019.

[50] T. P. Peixoto, "Efficient Monte Carlo and greedy heuristic for the inference of stochastic block models," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 89, no. 1, Jan. 2014, Art. no. 012804.

[51] P. Zhang and C. Moore, "Scalable detection of statistically significant communities and hierarchies, using message passing for modularity," *Proc. Nat. Acad. Sci. USA*, vol. 111, no. 51, pp. 18144–18149, Dec. 2014.

[52] F. Pompili, N. Gillis, P.-A. Absil, and F. Glineur, "Two algorithms for orthogonal nonnegative matrix factorization with application to clustering," *Neurocomputing*, vol. 141, pp. 15–25, Oct. 2014.

[53] Z. Liu, X. Luo, and M. Zhou, "Symmetry-constrained non-negative matrix factorization approach for highly-accurate community detection," in *Proc. IEEE 17th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2021, pp. 1521–1526.
[54] X. Luo, Z. Liu, M. Shang, J. Lou, and M. Zhou, "Highly-accurate community detection via pointwise mutual information-incorporated symmetric non-negative matrix factorization," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 463–476, Jan. 2021.
[55] E. E. Papalexakis, L. Akoglu, and D. Ience, "Do more views of a graph help? Community detection and clustering in multi-graphs," in *Proc. 16th Int. Conf. Inf. Fusion*, 2013, pp. 899–905.
[56] Q. Gao, W. Xia, Z. Wan, D. Xie, and P. Zhang, "Tensor-SVD based graph learning for multi-view subspace clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 3930–3937.
[57] Z. Li, C. Tang, X. Liu, X. Zheng, W. Zhang, and E. Zhu, "Consensus graph learning for multi-view clustering," *IEEE Trans. Multimedia*, vol. 24, pp. 2461–2472, 2022.
[58] Z. Zhang, P. Cui, H. Li, X. Wang, and W. Zhu, "Billion-scale network embedding with iterative random projection," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2018, pp. 787–796.
[59] D. Yang, P. Rosso, B. Li, and P. Cudre-Mauroux, "NodeSketch: Highly-efficient graph embeddings via recursive sketching," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1162–1172.
[60] J. Li, L. Wu, R. Guo, C. Liu, and H. Liu, "Multi-level network embedding with boosted low-rank matrix approximation," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2019, pp. 49–56.
[61] D. Jin *et al.*, "Incorporating network embedding into Markov random field for better community detection," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 160–167.
[62] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, Feb. 2010.
[63] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 80, no. 1, 2009, Art. no. 016118.
[64] D. H. Wolper and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

Jiewen Guan received the M.Eng. degree from Xiamen University, Xiamen, China, in 2022.

Bilian Chen received the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2012.

She is currently an Associate Professor with Xiamen University, Xiamen, China. Her research interests include machine learning, optimization theory, and recommendation system. Her publications appear in the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *SIAM Journal on Optimization*, *Journal of Global Optimization, Information Sciences*, and so on.

Xin Huang received the Ph.D. degree from the Chinese University of Hong Kong (CUHK), Hong Kong, in 2014.

He is currently an Assistant Professor with Hong Kong Baptist University, Hong Kong. His research interests include graph data management and mining.