The Annals of Statistics
2021, Vol. 49, No. 6, 3181–3205
https://doi.org/10.1214/21-AOS2079

COMMUNITY DETECTION ON MIXTURE MULTILAYER NETWORKS VIA REGULARIZED TENSOR DECOMPOSITION

BY BING-YI JING[1,*], TING LI[2], ZHONGYUAN LYU[1,†] AND DONG XIA[1,‡]

[1]*Department of Mathematics, The Hong Kong University of Science and Technology,* [*]*majing@ust.hk;* [†]*zlyuab@connect.ust.hk;* [‡]*madxia@ust.hk*

[2]*Department of Applied Mathematics, The Hong Kong Polytechnic University, tingeric.li@polyu.edu.hk*

We study the problem of community detection in multilayer networks, where pairs of nodes can be related in multiple modalities. We introduce a general framework, that is, mixture multilayer stochastic block model (MMSBM), which includes many earlier models as special cases. We propose a tensor-based algorithm (TWIST) to reveal both global/local memberships of nodes, and memberships of layers. We show that the TWIST procedure can accurately detect the communities with small misclassification error as the number of nodes and/or number of layers increases. Numerical studies confirm our theoretical findings. To our best knowledge, this is the first systematic study on the mixture multilayer networks using tensor decomposition. The method is applied to two real datasets: worldwide trading networks and malaria parasite genes networks, yielding new and interesting findings.

1. Introduction. Networks arise in many areas of research and applications, which come in all shapes and sizes. The most studied and best understood are static network models. Many other network models are also in existence, but have been less studied. One such example is the multilayer networks, which are a powerful representation of relational data, and commonly encountered in contemporary data analysis [26]. The nodes in a multilayer network represent the entities of interest and the edges in different layers indicate the multiple relations among those entities. Examples include brain connectivity networks, world trading networks, gene-gene interactive networks and so on. In this paper, we focus on the multilayer networks with the same nodes set of each layer and there are no edges between two different layers.

The study on multilayer networks has received an increasing interest. Considering the dependency among the different layers, [42] derives consistency results for the community assignments from the maximum likelihood estimators in two models. Consistency properties of various methods for community detection under the multilayer stochastic block model are investigated in [43]. Three different matrix factorization-based algorithms are employed in [41, 52] and [14] separately. Common community structures for multiple networks are identified via two spectral clustering algorithms with theoretical guarantee in [6]. In [2], authors introduce the common subspace independent-edge multiple random graph model to describe a heterogeneous collection of networks with a shared latent structure and propose a joint spectral embedding of adjacency matrices to simultaneously and consistently estimate underlying parameters for each graph. Consistency results for a least squares estimation of memberships under the multilayer stochastic block model framework are derived in [33]. Several literature focus on recovering the network from a collection of networks with edge contamination. The original network is estimated from multiple noisy realizations utilizing community structure in [31] and low-rank expectation in [36]. A weighted latent position

Received February 2020; revised April 2021.
MSC2020 subject classifications. 62H30, 91C20.
Key words and phrases. Network community detection, multilayer network, tensor, tucker decomposition.

graph model contaminated via an edge weight gross error model is proposed in [51] with an estimation methodology based on robust L_q estimation followed by low-rank adjacency spectral decomposition.

In applications, a random effects stochastic block model is proposed by [44] for the neuroimaging data, and a statistical framework with a significance and a robustness test for detecting common modules in the Drosophila melanogaster dynamic gene regulation network is proposed in [63].

Most of the literature about community detection in multilayer networks is limited to consistent membership setting, which means all the layers carry information about the same community assignment. However, in reality, different layers may have different community structures. For instance, in a social network, layers related with sports (people connected with the same sport hobbies) may have different community structure with layers about movie taste (people connected with similar movie taste). Understanding the large-scale structure of multilayer networks is made difficult by the fact that the patterns of one type of link may be similar to, uncorrelated with, or different from the patterns of another type of link. These differences from layer to layer may exist at the level of individual links, connectivity patterns among groups of nodes, or even the hidden groups themselves to which each node belongs. In [4], authors pointed out that, in order to do community detection on multilayer networks, it is crucial to know which layers have related structure and which layers are unrelated, since redundant information across layers may provide stronger evidence for clear communities than each layer would on its own. Such situation is not clearly discussed in the works mentioned above. A strata multilayer stochastic block model (sMLSBM) is proposed in [49], which assumes that the layers in a stratum follow an identical SBM model. They propose to analyze each layer separately, and conduct network comparisons pairwisely. It is unclear how the multiple strata of layers contribute to the network structures. Moreover, their estimating method does not integrate multiple layers, which causes potential information loss. Although, authors in [40, 45] introduced community structure variety as time varying, it is hard to be applied in general multilayer networks without time ordering. A joint embedding for multiple networks analysis is introduced in [3], which collects multiple adjacency matrices into a single large matrix with some off-diagonal tethering. Theoretical results are proved under the random dot product graph model, which requires stringent sparsity conditions.

In this paper, we introduce a general framework, that is, mixture multilayer stochastic block model (MMSBM), and propose a tensor-based algorithm (TWIST) to reveal both global/local memberships of nodes, and memberships of layers. To fix ideas, we start with a simple motivating example, illustrated in Figure 1. We have $L = 3$ layers of networks $\{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3\}$, each containing 3 local communities. The 3 networks are of $m = 2$ types: $\{\mathcal{G}_2\}$ and $\{\mathcal{G}_1, \mathcal{G}_3\}$. The community structure differs between $\{\mathcal{G}_2\}$ and the other two networks as some members in the third community g_{23} are in the second one g_{12} in $\{\mathcal{G}_1\}$ and $\{\mathcal{G}_3\}$. Viewing the 3 layers of networks together, we notice that there are 4 global communities, in which members stay in all layers throughout. Clearly, the global communities are related to, but different from the local ones in each network. Our interest lies in detecting both local as well as global community structures, which are of great value in theory and practice. There is an increasing literature on the global community structure as mention earlier. However, to the best of our knowledge, there is no systematic investigation into detecting local and global community structure together.

Our line of attack can be illustrated via the following diagram in Figure 2. First, we pool adjacency matrices from all layers of networks to form a tensor (multiway array), and then apply the TWIST (to be introduced later) to obtain the global community structure as well as labels of each layer. We then group the layers of networks with the same labels, which will be used to detect local community structures. Details will be unfolded next.

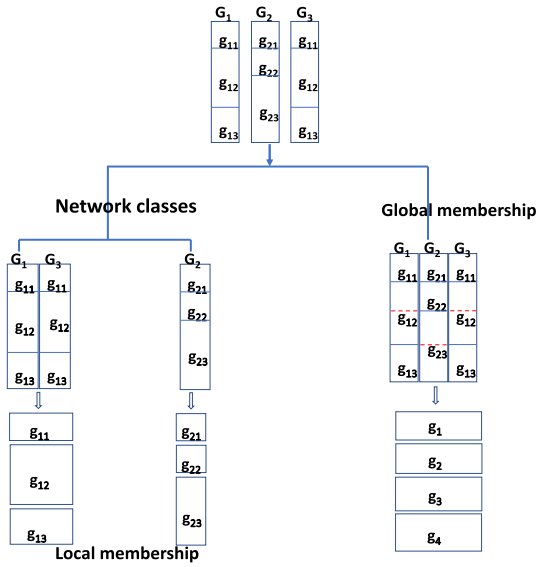


FIG. 1. *A toy example.*

The main contributions of this paper are summarized as follows.

First, we propose a very general model to handle the type of problems discussed above. To be more specific, we will introduce the so-called mixture multilayer stochastic block model (MMSBM), which can characterize the different community structures among different layers of the multilayer network. In some way, the MMSBM resembles the relatively well studied multilayer stochastic block model (MLSBM) [17, 42, 43, 49]. However, the MMSBM is more general in that it allows the multilayer network to contain different block structures. Thus, the MMSBM not only allows each layer to have different community structures, but also can maintain the consistent structure in the network.

Second, we propose a tensor-based method to study the MMSBM. The approach is referred to as the Tucker decomposition with integrated SVD transformation (TWIST). Unlike earlier approaches for multilayer network analysis, TWIST can uncover the clusters of layers, the local and global membership of nodes simultaneously. On the theoretical front, we prove

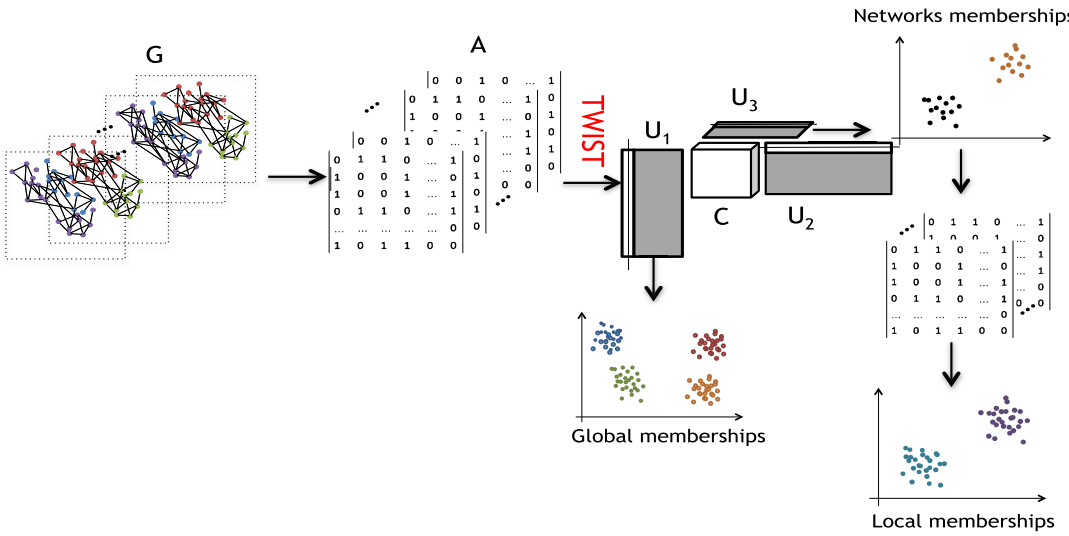


FIG. 2. *The general procedure of TWIST.*

for MMSBM that TWIST can consistently recover the layer labels and global memberships of nodes under near optimal network sparsity conditions. In addition, network labels can be exactly recovered under a slightly stronger network sparsity condition. To the best of our knowledge, this is the first systematic study on statistical guarantees about community detection in a mixture multilayer networks using tensor decomposition. Our primary technical tool is a sharp concentration inequality of sparse tensors, which might be of independent interest.

Finally, two real-world applications of the proposed methodology are demonstrated to be a powerful tool in analysing multilayer networks. The algorithm is easy to use and can help practitioners quickly uncover interesting findings, which would otherwise be difficult by using other tools.

The rest of the paper is organized as follows. Section 2 introduces the mixture multilayer stochastic block model (MMSBM) for describing the mixture structure. A new algorithm, the TWIST, is proposed in Section 3. We explore the theoretical properties of TWIST under the MMSBM in Section 4. Moreover, we make comparisons between our main results and the cutting-edge theoretical results in Section 5. The advantages of the proposed method are numerically evaluated with comprehensive simulations in Section 6 and two real data examples in Section 7. Section 8 gives concluding remarks and discussions. More numerical examples, and all the proofs are shown in the Supplementary Material [22].

2. Model framework.

2.1. *Mixture multilayer stochastic block model* (*MMSBM*). The observed data contains L-layers of networks on the same set of vertices: $\mathcal{V} = [n] := \{1, 2, \ldots, n\}$:

$$\mathcal{G} = \{\mathcal{G}_l : l = 1, \ldots, L\}.$$

Assume that there is a mixture of m latent network models, and each network $\mathcal{G}_l$ is sampled independently from this mixture of models with probability $\pi = (\pi_1, \ldots, \pi_m)$. Denoting $s_l \in \{1, \ldots, m\}$ as a random latent label of $\mathcal{G}_l$ with $1 \le l \le L$, then

$$\mathbb{P}(s_l = j) = \pi_j \quad \text{with } \sum_{j=1}^{m} \pi_j = 1.$$

Assume that each of the m classes of networks satisfies the *stochastic block model* (SBM). More specifically, for $j \in [m]$, the jth class SBM is described by *membership matrix* $Z_j \in \{0, 1\}^{n \times K_j}$ and the probability matrix $B_j := \bar{p} B_j^0 \in [0, 1]^{K_j \times K_j}$ (both are deterministic), where K_j is the number of communities and $\bar{p} \in (0, 1]$ characterizes the overall network sparsity. We assume $\max_j \|B_j^0\|_{\max} = 1$ for identifiability. Note that each row of Z_j has exactly one entry that is nonzero and $A_l \in \{0, 1\}^{n \times n}$ is the observed adjacency matrix of $\mathcal{G}_l$. For simplicity, we denote:

- $\mathrm{SBM}(Z_j, B_j)$ = the jth SBM with parameter Z_j and B_j, $j = 1, \ldots, m$.
- $\mathcal{V}_k^j$ = the kth community in the jth SBM. So $\mathcal{V}_k^j \subset \mathcal{V}$ and $\bigcup_{k=1}^{K_j} \mathcal{V}_k^j = \mathcal{V}$.
- $L_j = \#\{l : s_l = j, 1 \le l \le L\}$ = the number of layers generated by $\mathrm{SBM}(Z_j, B_j)$. Clearly, $L = \sum_{j=1}^{m} L_j$.
- $\mathring{K} = K_1 + \cdots + K_m$ and $\mathbb{S} = \{s_l\}_{l=1}^{L}$ and $\mathbb{V}^j := \{\mathcal{V}_k^j\}_{k=1}^{K_j}$.

Conditioned on the class label s_l, the observed adjacency matrix $A_l \in \{0, 1\}^{n \times n}$ of $\mathcal{G}_l$ obeys Bernoulli distribution:

$$A_l(i_1, i_2) | s_l \overset{i.i.d.}{\sim} \mathrm{Bern}\big(Z_{s_l}(i_1, :) B_{s_l} Z_{s_l}(i_2, :)^\top\big) \tag{2.1}$$

for all $i_1 \le i_2 \in [n]$, where $Z(i, :)$ denotes the ith row of Z.

The resulting model is referred to as "mixture multilayer stochastic block model" (MMSBM). By observing $\{\mathcal{G}_l\}_{l=1}^L$ or their adjacency matrices $\{A_l\}_{l=1}^L$, our goal is to recover the latent classes $\{s_l\}_{l=1}^L$ and hidden community structures. Hereinafter, we view $\{s_l\}_{l=1}^L$ as hidden and deterministic labels, rather than random variables. We note that MMSBM can be generalized to cases that each layer has different probability matrix B_l's.

2.2. *Adjacency tensor and its decomposition.* Observing the L layers of networks, we define the adjacency tensor $\mathbf{A} \in \mathbb{R}^{n\times n\times L}$ so that $\mathbf{A}$'s lth slice

$$A(:,:,l) = A_l, \quad \forall 1 \le l \le L.$$

See [27] for an introduction to tensor algebra. It follows from (2.1) that

$$\mathbb{E}(A_l|s_l) = Z_{s_l} B_{s_l} Z_{s_l}^\top, \quad \forall 1 \le l \le L,$$

from which we can derive the following tensor representation, whose proof is in the Supplementary Material [22]. Note that the multilinear product in (2.2) is defined by

$$\mathbb{E}\big(A(i_1,i_2,i_3)|\mathbb{S}\big) = \sum_{j_1=1}^{\mathring{K}} \sum_{j_2=1}^{\mathring{K}} \sum_{j_3=1}^{m} B(j_1,j_2,j_3)\bar{Z}(i_1,j_1)\bar{Z}(i_2,j_2)\bar{W}(i_3,j_3).$$

LEMMA 2.1 (Tensor representation). *We have*

$$\mathbb{E}(\mathbf{A}|\mathbb{S}) = \mathbf{B} \times_1 \bar{Z} \times_2 \bar{Z} \times_3 W, \tag{2.2}$$

where $\mathbb{S} = \{s_l\}_{l=1}^L$ *and*:

- $\bar{Z} = (Z_1, Z_2, \ldots, Z_m) \in \{0,1\}^{n\times \mathring{K}}$ *is the* global *membership matrix*, *whereas each* Z_j *is the* local *membership matrix*,
- $W = (e_{s_1}, e_{s_2}, \ldots, e_{s_L})^\top \in \{0,1\}^{L\times m}$ *is the network label matrix with each row of* W *having exactly one nonzero entry, and* $e_j \in \mathbb{R}^m$ *being the* j*th canonical basis vector,*
- $\mathbf{B} \in \mathbb{R}^{\mathring{K}\times\mathring{K}\times m}$ *is a* 3*-way probability tensor whose* j*th frontal slice is*

$$B(:,:,j) = \operatorname{diag}(0_{K_1}, \ldots, 0_{K_{j-1}}, B_j, 0_{K_{j+1}}, \ldots, 0_{K_m}), \quad 1 \le j \le m$$

with 0_K *being a* $K \times K$ *zero matrix.*

2.3. *Local versus global memberships via Tucker decomposition.* The matrix $\bar{Z}$ defined in Lemma 2.1 suggests the existence of *global community structures*. We say that two nodes i_1 and i_2 belong to the same global community if and only if they belong to the same local community for all the m classes of SBM, that is,

$$\bar{Z}(i_1,:) = \bar{Z}(i_2,:).$$

Let $\bar{K}$ denote the number of global communities, that is, number of distinct rows of $\bar{Z}$. Clearly, $\max_j K_j \le \bar{K} \le \prod_j K_j$. Denote $\bar{\mathbb{V}} = \{\bar{\mathcal{V}}_k\}_{k=1}^{\bar{K}}$ the global community clusters such that $\bigcup_{k=1}^{\bar{K}} \bar{\mathcal{V}}_k = \mathcal{V}$. Therefore, for two nodes $i_1 \ne i_2$,

$$\{i_1,i_2\} \in \bar{\mathcal{V}}_k \iff \{i_1,i_2\} \in \mathcal{V}_{k_j}^j \quad \text{for some } k_j \in [K_j], \forall j \in [m]. \tag{2.3}$$

Let $r = \operatorname{rank}(\bar{Z})$ denote the rank of $\bar{Z}$. We hereby write the thin SVD of $\bar{Z}$ as

$$\bar{Z} = \bar{U}\bar{D}\bar{R}^\top, \tag{2.4}$$

where $\bar{U} \in \mathbb{R}^{n\times r}$, $\bar{R} \in \mathbb{R}^{\mathring{K}\times r}$ have orthonormal columns, and $\bar{D}$ is the singular value diagonal matrix

$$\bar{D} = \operatorname{diag}\big(\sigma_1(\bar{Z}), \ldots, \sigma_r(\bar{Z})\big) \in \mathbb{R}^{r\times r}, \quad \sigma_1(\bar{Z}) \geq \cdots \geq \sigma_r(\bar{Z}) > 0.$$

We note that $\bar{Z}$ cannot be full rank in general, which is different from the canonical SBM. Clearly, we have $\max_j K_j \leq r \leq \min\{\mathring{K} - (m-1), \bar{K}\}$. If $K_j \equiv K$, the maximum rank of $\bar{Z}$ is $\min\{mK - m + 1, \bar{K}\}$. If we define a $\bar{K} \times \mathring{K}$ matrix Z^* containing the $\bar{K}$ distinct rows of $\bar{Z}$, then r essentially equals the rank of Z^*. The real data examples in Section 7 show that it often suffices to take $r = \bar{K}$ in practice.

The global community structure can be checked by $\bar{U}$ as in Lemma 2.2.

LEMMA 2.2. *For $i_1 \in \bar{\mathcal{V}}_{k_1}$ and $i_2 \in \bar{\mathcal{V}}_{k_2}$ with $k_1 \neq k_2$, then $\|\bar{U}(i_1,:) - \bar{U}(i_2,:)\|_{\ell_2} \geq 1/\sigma_1(\bar{D})$.*

By (2.4), the population adjacency tensor $\mathbb{E}(\mathbf{A}|\mathbb{S})$ admits the *Tucker decomposition* as

$$\mathbb{E}(\mathbf{A}|\mathbb{S}) = \bar{\mathbf{C}} \times_1 \bar{U} \times_2 \bar{U} \times_3 \bar{W}, \tag{2.5}$$

where the core tensor $\bar{\mathbf{C}} \in \mathbb{R}^{r\times r\times m}$ is defined by

$$\bar{\mathbf{C}} = \mathbf{B} \times_1 \big(\bar{D}\bar{R}^\top\big) \times_2 \big(\bar{D}\bar{R}^\top\big) \times_3 D_L^{1/2} \tag{2.6}$$

and $\bar{W} = W D_L^{-1/2} \in \mathbb{R}^{L\times m}$ so that $\bar{W}^\top \bar{W} = I_m$, and the diagonal matrix $D_L = \operatorname{diag}(L_1, L_2, \ldots, L_m)$.

We assume that $\bar{\mathbf{C}}$ has Tucker ranks (r, r, m), which implies that $m \leq r^2$. The decomposition (2.5) shows that the singular vectors of $\mathbb{E}(\mathbf{A}|\mathbb{S})$ contain the latent network information. More exactly, the singular vectors in the first dimension of $\mathbb{E}(\mathbf{A}|\mathbb{S})$ could identify the global community structures and singular vectors in the third dimension could identify the latent network labels. After identifying the latent network labels, a post-processing procedure can identify the local community structures.

Compared with sMLSBM [49], our proposed MMSBM introduces the global memberships of vertices. It essentially characterizes how multiple strata of layers contribute to the network structures. The random-effect SBM [44] allows nodes to change memberships in different layers according to random effects. However, in MMSBM, the membership varies across different layers according to layers' classes.

3. Methodology: TWIST. By observing the multilayer networks $\{\mathcal{G}_l\}_{l=1}^L$ satisfying model (2.1), our goals are to:

(1) recover the global community structures of vertices $\{\bar{\mathcal{V}}_k\}_{k=1}^{\bar{K}}$;

(2) identify network classes $\{s_l\}_{l=1}^L$, and grouping networks with the same class;

(3) recover the local community structures of vertices $\mathbb{V}^j = \{\mathcal{V}_k^j : k \in [K_j]\}$ for all $j \in [m]$.

Note that in order to efficiently recover the local community structures, it is necessary to first identify the network classes. As a result, task (3) usually follows from task (2).

By the decomposition of oracle tensor (2.5), the singular vectors $\bar{U}$ contains information of global memberships since its column space comes from $\bar{Z}$. Additionally, the singular vectors $\bar{W}$ contains information of network classes. Therefore, task (1) and task (2) are both related with the Tucker decomposition of oracle tensor $\mathbb{E}(\mathbf{A}|\mathbb{S})$. Since the oracle is unavailable, we seek a low-rank approximation of $\mathbf{A}$.

Algorithm 1 Regularized power iterations for sparse tensor decomposition

Input: $\mathbf{A} \in \{0,1\}^{n\times n\times L}$, warm initialization $\widehat{U}^{(0)}$ and $\widehat{W}^{(0)}$
maximum iterations $\text{iter}_{\max}$ and regularization parameters $\delta_1, \delta_2 > 0$.
Output: $\widehat{U}$ and $\widehat{W}$
Set counter $\text{iter} = 0$.
while $\text{iter} < \text{iter}_{\max}$ **do**
Regularization: $\widetilde{U}^{(\text{iter})} \leftarrow \mathcal{P}_{\delta_1}(\widehat{U}^{(\text{iter})})$ and $\widetilde{W}^{(\text{iter})} \leftarrow \mathcal{P}_{\delta_2}(\widehat{W}^{(\text{iter})})$ by (3.1).
$\text{iter} \leftarrow \text{iter} + 1$
Set $\widehat{U}^{(\text{iter})}$ to be the top r left singular vectors of $\mathcal{M}_1(\mathbf{A} \times_2 \widetilde{U}^{(\text{iter}-1)\top} \times_3 \widetilde{W}^{(\text{iter}-1)\top})$.
set $\widehat{W}^{(\text{iter})}$ to be the top m left singular vectors of $\mathcal{M}_3(\mathbf{A} \times_1 \widetilde{U}^{(\text{iter}-1)\top} \times_2 \widetilde{U}^{(\text{iter}-1)\top})$.
end while
Return $\widehat{U} \leftarrow \widehat{U}^{(\text{iter})}$ and $\widehat{W} \leftarrow \widehat{W}^{(\text{iter})}$.

3.1. *Tucker decomposition with integrated SVD transformation* (*TWIST*). In order to utilize the low rank structure of the tensor and the 0-or-1 property of the elements, we propose a new algorithm called Tucker decomposition with integrated SVD transformation (TWIST). The general procedure is summarized below and illustrated in Figure 2.

- *Step 1: Decomposition of adjacency tensor*
  Apply the regularized tensor power iterations to $\mathbf{A}$ to obtain its low-rank approximation. The outputs are $\widehat{U}$ and $\widehat{W}$. Details are given in Algorithm 1.
- *Step 2: Global memberships*
  Apply the standard K-means algorithm on the rows of $\widehat{U}$ to identify the global community memberships and output $\widehat{\mathbb{V}} = \{\widehat{\mathcal{V}}_k\}_{k=1}^{K}$.
- *Step 3: Network classes*
  Use the rows of $\widehat{W}$ to identify the network classes and output the network classes: $\widehat{\mathbb{S}} = \{\hat{s}_l \in [m]\}_{l=1}^{L}$. We can use either the standard K-means or the sup-norm related algorithm (Algorithm 2).
- *Step 4: Local memberships*
  We can find the local membership $\mathbb{V}^j = \{\mathcal{V}_k^j\}$ by focusing on networks with the same labels [34, 47]. More precisely, for each $j \in \{1,\dots,m\}$, we can apply K-means either:
  – to the sum of those networks with the same label $\sum_{l:\hat{s}_l=j} A_l$, or
  – to the subtensor $A(:,:,\{l : \hat{s}_l = j\}))$, those slices with the same labels.
  Outputs are $\widehat{\mathbb{V}}^j = \{\widehat{\mathcal{V}}_k^j\}_{k=1}^{K_j}$.

3.2. *Features about TWIST.* There are several key features concerning TWIST.

Warm starts for $\widehat{U}^{(0)}$ and $\widehat{W}^{(0)}$ in Algorithm 1. Computing the optimal low-rank approximation of a tensor $\mathbf{A}$ is NP-hard in general; see [18]. Algorithms with random initializations can be always trapped in noninformative local minimals that can be nearly orthogonal to the truth; see [5]. To avoid these issues, tensor decomposition algorithms usually run from a warm starting point [10, 19, 24, 46, 50, 54, 55, 57, 60–62].

In Section 5.5, we will introduce a warm initialization algorithm, obtained by applying a spectral method for initializing $\widehat{U}^{(0)}$ by summing up all the network layers. Initialization of $\widehat{W}^{(0)}$ is easy whenever $\widehat{U}^{(0)}$ is available. We show in Lemma 5.6 that these initializations are indeed warm under reasonable conditions.

Regularized power iterations for sparse tensor decomposition. Adjacency matrices from some layers are often very sparse and the individual layers are even disconnected graphs.

Algorithm 2 Network clustering by sup-norm K-means

Input: $\widehat{W}$, number of clusters m and threshold $\varepsilon \in (0, 1)$
Output: Network labels $\widehat{\mathbb{S}} = \{\hat{s}_l\}_{l=1}^L$

```
Initiate C ← {1}, ŝ_1 ← 1, k ← 1 and l ← 2.
while l ≤ L do
    Compute j ← arg min_{j∈C} ‖Ŵ(l,:) − Ŵ(j,:)‖
    if ‖Ŵ(l,:) − Ŵ(j,:)‖ > ε then
        k ← k + 1; ŝ_l ← k; C ← C ∪ {l}
    else
        ŝ_l ← ŝ_j
    end if
    l ← l + 1
end while
if k > m (or k < m) then
    Set ε ← 2ε (or set ε ← ε/2); Rerun the algorithm.
else
    Output Ŝ = {ŝ_l}_{l=1}^L
end if
```

For example, in the Malaria parasite genes networks given in Section 7, three out of nine networks are very sparse and disconnected. Under these circumstances, the popular tensor power iteration algorithm, that is, high-order orthogonal iterations (HOOI, see [48]) may not work. In fact, its statistical optimality was proved by [62] only for dense tensors, while its properties on sparse random tensors remain much more challenging.

To handle sparse random tensors, we employ a regularized tensor power iteration algorithm in Algorithm 1, which was used in [24] to deal with sparse hypergraph networks. Regularizations to singular vectors $\widehat{U}^{(t)}$, $\widehat{W}^{(t)}$ are applied before each power iteration. We take $\widehat{U}^{(t)}$, for example, as $\widehat{W}^{(t)}$ is treated similarly. The regularization is done by

$$\begin{aligned} &\mathcal{P}_\delta(U) = \mathrm{SVD}_r(U_*) \\ &\quad \text{where } U_*(i,:) := U(i,:) \cdot \min\{\delta, \|U(i,:)\|\}/\|U(i,:)\| i \in [n]. \end{aligned} \tag{3.1}$$

The effect of regularization is to dampen the influence of "large" rows of $\widehat{U}^{(t)}$, which is due to the communities of small sizes, besides stochastic errors. Following Lemma 4.2, the true singular vectors $\bar{U}$ is incoherent with $\max_{1\le j\le n} \|e_j^\top \bar{U}\| = O(\sqrt{r/n})$ if $\bar{D}$ is well conditioned. In practice, we suggest

$$\begin{aligned} \hat{\delta}_1 &= 2\sqrt{r} \cdot \max_{1\le i\le n} \deg_i \cdot \left(\sum_{i=1}^n \deg_i^2\right)^{-1/2} \quad \text{and} \\ \hat{\delta}_2 &= 2\sqrt{m} \cdot \max_{1\le l\le L} \mathrm{neg}_l \cdot \left(\sum_{l=1}^L \mathrm{neg}_l^2\right)^{-1/2}, \end{aligned} \tag{3.2}$$

where the node degree $\deg_i = \sum_{j,l} A(i,j,l)$ and layer degree $\mathrm{neg}_l = \sum_{i,j} A(i,j,l)$.

K-means with sup-norm distance. Clearly, the accuracy of local membership clustering (Step 4) hinges on the reliability of layer labelling. In Algorithm 2, a sup-norm version of K-means is applied to the singular vectors $\widehat{W}$ obtained from Algorithm 1, which then outputs the network labels $\widehat{\mathbb{S}} = \{\hat{s}_l\}_{l=1}^L$. The sup-norm K-means has recently been extensively investigated

and shown to perform well in network community detection. See, for example, [1, 12, 25, 35] and references therein.

The rationale of Algorithm 2 is that when the rows of $\bar{W}$ are well separated (similar to Lemma 2.2), a rowwise screening of $\widehat{W}$ can immediately recover the true network labels as long as the rowwise perturbation bound of $\widehat{W} - W$ is small enough. As shown in Section 5, Algorithm 2 guarantees the exact clustering of networks under weak conditions.

In Steps 2–4, one could use alternative methods other than K-means clustering, which might improve its performances. For example, we can use DBSCAN, Gaussian mixture model, the SCORE method [20, 21, 24].

Estimating r, m and $\bar{K}$. In practice, the numbers of node communities and network classes are unknown. Various methods are available for estimating the number of communities in a single network. The famous scree plot [11] method estimates rank by the number of statistically significant components of the adjacency matrix or its normalization [23, 24]. Borrowing such ideas, if r and m are unknown, we apply Tensor decomposition with relatively large ranks on $\mathbf{A}$, and then estimate r and m based on the significant entries of the core tensor. A numerical example in the Supplementary Material [22] illustrates how the idea works in tensors. While doing data exploration, we can take $\bar{K}$ from small to large. This produces few large groups initially, and gradually splits them into small groups with hierarchical structure. See, for instance, [35, 38, 39]. The procedure stops when a reasonable community structure is reached.

4. Preliminary results.

4.1. *Notations and definitions.* For ease of exposition, we introduce the following notation:

- Denote c, c_j, C, C_j, C'_j, $j \geq 1$ as generic constants, which may vary from line to line.
- Denote e_k as the kth canonical basis vector in Euclidean space (i.e., with only the kth entry equal to 1 and others 0), whose dimension depends on each context.
- For a matrix $M = \{m_{ij}\}$, let
  $\sigma_i(M) =$ the ith largest singular value of M,
  $\|M\|_{\max} = \max_{i,j} |m_{ij}|$, the maximal absolute value of all entries of M,
  $\|M\| = \max\{\sigma_1(M^T M)\}^{1/2}$, the spectral norm (Euclidean norm for vectors).
- For a $d_1 \times d_2 \times d_3$ tensor $\mathbf{T}$, let $\mathcal{M}_j(\mathbf{T})$ be a $d_j \times (d_1 d_2 d_3 / d_j)$ matrix by unfolding $\mathbf{T}$ in the jth dimension.

4.2. *Signal strengths.* Recall the low-rank decomposition of $\mathbb{E}(\mathbf{A}|\mathbb{S})$ in (2.5) with

$$\mathbb{E}(\mathbf{A}|\mathbb{S}) = \bar{\mathbf{C}} \times_1 \bar{U} \times_2 \bar{U} \times_3 \bar{W},$$

where $\mathbb{S} = \{s_l\}_{l=1}^L$. The core tensor $\bar{\mathbf{C}} = \mathbf{B} \times_1 (\bar{D}\bar{R}^\top) \times_2 (\bar{D}\bar{R}^\top) \times_3 D_L^{1/2} = \bar{\mathbf{B}} \times_1 \bar{D} \times_2 \bar{D} \times_3 D_L^{1/2}$ where

$$\bar{\mathbf{B}} = \mathbf{B} \times_1 \bar{R}^\top \times_2 \bar{R}^\top \in \mathbb{R}^{r \times r \times m} \tag{4.1}$$

Denote the signal strengths of $\bar{\mathbf{C}}$ and $\bar{\mathbf{B}}$ by $\sigma_{\min}(\bar{\mathbf{C}})$ and $\sigma_{\min}(\bar{\mathbf{B}})$, respectively. Generally,

$$\sigma_{\min}(\mathbf{T}) = \min\{\sigma_{r_j}(\mathcal{M}_j(\mathbf{T})) : j = 1, 2, 3\} \tag{4.2}$$

for any $\mathbf{T}$ with Tucker ranks (r_1, r_2, r_3).

Recall that $B_j = \bar{p} B_j^0$ for $j \in [m]$, then we can define a third-order tensor $\bar{\mathbf{B}}^0$ such that $\bar{\mathbf{B}} = \bar{p}\bar{\mathbf{B}}^0$. Also denote $x \asymp y$ iff $x = O(y)$ and $y = O(x)$. Then the following conditions greatly simplify our presentation.

CONDITION 1. *Assume that*:

- (*A*1): $\bar{\mathbf{B}}^0$ *has Tucker ranks* (r, r, m) *and* $\sigma_{\min}(\bar{\mathbf{B}}^0) \geq c_1$ *for some constant* $c_1 > 0$;
- (*A*2): $\bar{D}$ *is well conditioned, that is,* $\sigma_1(\bar{D}) \leq \kappa_0 \sigma_r(\bar{D})$ *for some* $\kappa_0 \geq 1$;
- (*A*3): *Minimal network balance condition*: $L_{\min} \asymp L/m$, *where* $L_{\min} = \min_{1\leq j\leq m} L_j$;
- (*A*4): *Maximal network balance condition*: $L_{\max} \asymp L/m$, *where* $L_{\max} = \max_{1\leq j\leq m} L_j$.

Condition (A1) requires the core tensor $\bar{\mathbf{B}}$ to have full rank, which is mild for low-rank tensor analysis. In the Supplementary Material [22], an example is included to explain condition (A1) more specifically. We also prove that κ_0 is related to the largest and smallest community sizes of global memberships in the Supplementary Material [22].

LEMMA 4.1 (Signal strength). *If conditions* (*A*1) *and* (*A*2) *hold, we have*

$$\sigma_{\min}(\bar{\mathbf{C}}) \geq c_1 \kappa_0^{-2} r^{-1} m \cdot n\bar{p}\sqrt{L_{\min}}.$$

Further if (*A*3) *holds and* m, r, κ_0 *are fixed, then* $\sigma_{\min}(\bar{\mathbf{C}}) \geq c_0\sqrt{L} n\bar{p}$.

By Lemma 4.1, the signal strength of $\bar{\mathbf{C}}$ is characterized by the overall network sparsity.

4.3. *Incoherence property.* Theoretically, the ideal regularization parameters in Algorithm 1 are $\delta_1 = \max_{1\leq j\leq n} \|e_j^\top \bar{U}\|$ and $\delta_2 = \max_{1\leq j\leq L} \|e_j^\top \bar{W}\|$. Incoherence property ensures that singular vectors $\bar{U}$ and $\bar{W}$ are not too correlated with or incoherent to the standard basis e_j's, as stated in the next lemma, which the sharp convergence rates of regularized tensor power iteration algorithm rely crucially on.

LEMMA 4.2 (Incoherence of $\bar{U}$ and $\bar{W}$). *If conditions* (*A*1) *and* (*A*2) *hold, we have*

$$\delta_1 \leq \kappa_0\sqrt{r/n} \quad \text{and} \quad \delta_2 \leq \kappa_0 m^{-1} r/\sqrt{L_{\min}}.$$

Then under conditions (A1)–(A3), it follows from Lemma 4.2: $\delta_2 \leq C_1\kappa_0 r/\sqrt{mL}$.

4.4. *Tensor incoherent norms and a concentration inequality.* In this section, we fix $\mathbb{S}$ and write $\mathbb{E}\mathbf{A}$ in short for $\mathbb{E}(\mathbf{A}|\mathbb{S})$. Given a random tensor $\mathbf{A}$, we write

$$\mathbf{A} = \mathbb{E}\mathbf{A} + (\mathbf{A} - \mathbb{E}\mathbf{A}) = \text{the signal} + \text{noise part}.$$

A tensor norm is needed to measure the size of the noise part. In order to deal with extremely sparse tensors, we will adopt the following definition, first introduced in [59].

DEFINITION 4.3 (Tensor incoherent norm [59]). For $\delta \in (0, 1]$ and $k = 1, 2, 3$, define

$$\|\mathbf{A} - \mathbb{E}\mathbf{A}\|_{k,\delta} := \sup_{\mathbf{U}\in\mathcal{U}_k(\delta)} \langle \mathbf{A} - \mathbb{E}\mathbf{A}, \mathbf{U}\rangle,$$

where $\mathcal{U}_k(\delta) := \{\mathbf{U} = u_1 \otimes u_2 \otimes u_3 : \|u_j\|_{\ell_2} \leq 1, \forall j; \|u_k\|_{\ell_\infty} \leq \delta\}$, $\|u\|_{\ell_p}$ is the l_p-norm of u, and $\langle\cdot,\cdot\rangle$ denotes the vectorized inner product.

We now present a concentration inequality for tensor incoherent norms for sparse random tensors, which is essential in proving Theorem 5.1. Let $\bar{p} = \max_j \|\mathbb{E}A(:,:,j)\|_{\max}$.

THEOREM 4.4 (A concentration inequality for tensor incoherent norm). *Suppose that* $L \leq n$ *and* $Ln\bar{p} \geq \log n$. *Denote* $n_1 = n_2 = n$ *and* $n_3 = L$. *Then for* $k = 1, 2, 3$, *we have*

$$\mathbb{P}\{\|\mathbf{A} - \mathbb{E}\mathbf{A}\|_{k,\delta} \geq 3t\} \leq \frac{2}{n^2} + 10(\log n)^2 \lceil \log_2(\delta^2 n_k)\rceil \left[\exp\left(-\frac{t^2}{C_3\bar{p}}\right) + \exp\left(-\frac{3t}{C_4\delta}\right)\right]$$

provided $t \geq \max\{C_1, C_2\delta\sqrt{n_k}\log(n)\}\sqrt{n\bar{p}}\log(\delta^2 n_k)\log(n)$ *for some constants* $C_1, C_2 > 0$.

We make several remarks concerning the inequality.

1. The bound in Theorem 4.4 is sharper than that in [59] by a more sophisticated cardinality calculation, in order to deal with extremely sparse networks. Analogous results were previously established for sparse hypergraph networks [24] where the random tensor is symmetric, however, the dimension sizes (n and L) in our model can be drastically different (e.g., $L \ll n$), which needs more careful treatments.

2. Clearly, if $\delta = 1$, $\|\mathbf{A} - \mathbb{E}\mathbf{A}\|_{k,\delta}$ reduces to the standard tensor operator norm $\|\mathbf{A} - \mathbb{E}\mathbf{A}\|$. It is easy to check, by the maximum number of nonzero entries on the fibers of $\mathbf{A} - \mathbb{E}\mathbf{A}$, that $\|\mathbf{A} - \mathbb{E}\mathbf{A}\| \gtrsim 1$ with high probability (see [33], Theorem 2). By comparison, if $\delta_1 = O(1/\sqrt{n})$ and $\delta_2 = O(1/\sqrt{L})$, Theorem 4.4 shows that $\|\mathbf{A} - \mathbb{E}\mathbf{A}\|_{1,\delta_1}, \|\mathbf{A} - \mathbb{E}\mathbf{A}\|_{3,\delta_2} = O_p(\sqrt{n\bar{p}})$ up to some logarithmic factor. It can be much smaller than $\|\mathbf{A} - \mathbb{E}\mathbf{A}\| \gtrsim 1$ (w.h.p.) when L is large.

3. To apply tensor incoherent norms to analyze the convergence property of power iterations, it is necessary to prove that $\{\widehat{U}^{(t)}\}$ and $\{\widehat{W}^{(t)}\}$ are incoherent. It is possible to generalize the methods in [9, 28, 56, 58] for this purpose whose actual proof can be very involved. For simplicity, we adopt an auxiliary regularization step (3.1) to truncate those singular vectors.

4. Without the condition $L \le n$, we can obtain a similar result by replacing all n's in the concentration inequality and constraint on t with $(n \vee L)$. Here, we focus on the regime $L \le n$ because it is the most common in multilayer network data.

5. Main results.

5.1. *Error bound of regularized power iteration.* Theorem 5.1 states that regularized power iteration method (Algorithm 1) works if we have a warm initialization and a strong enough signal-to-noise ratio. These conditions are typically required (see, e.g., [24, 55, 57, 62]) and generally unavoidable [62] in tensor data analysis.

For $\widehat{V}, V \in \mathbb{O}_{p,r} = \{V \in \mathbb{R}^{p\times r} : V^T V = I_r\}$, the distance between their column spaces is

$$\mathrm{d}(\hat{V}, V) := \inf_{O\in\mathbb{O}_{r,r}} \|\hat{V} - VO\|.$$

Define $\mathrm{Err}(t) = \max\{\mathrm{d}(\widehat{U}^{(t)}, \bar{U}), \mathrm{d}(\widehat{W}^{(t)}, \bar{W})\}$. We assume $L \le n$ throughout this section. We have the following result.

THEOREM 5.1 (General convergence results of regularized power iterations). *Assume that $L \le n$ and*:

- *the initializations $\widehat{U}^{(0)}$ and $\widehat{W}^{(0)}$ are warm, that is*, $\mathrm{Err}(0) \le 1/4$,
- *the signal strength of $\bar{\mathbf{C}}$ satisfies*

$$\sigma_{\min}(\bar{\mathbf{C}}) \ge \big(C_1 + C_2\big((\delta_1\sqrt{n}) \vee (\delta_2\sqrt{L})\big)\log n\big)\sqrt{r \wedge m} \cdot \sqrt{n\bar{p}}\log(n)\log\big((\delta_1^2 n) \vee (\delta_2^2 L)\big).$$

Then with probability at least $1 - 2n^{-2}$:

1. *for all $t \le t_{\max} := C_0 \log(\sigma_{\min}(\bar{\mathbf{C}})/(\sqrt{n\bar{p}} + \delta_1\delta_2))$, we have*

$$\mathrm{Err}(t) \le \frac{1}{2} \cdot \mathrm{Err}(t-1) + C_3 \frac{\sqrt{n\bar{p}\log n} + \delta_1\delta_2 \log n}{\sigma_{\min}(\bar{\mathbf{C}})}$$

2. *and we have* $\mathrm{Err}(t_{\max}) \le C_3(\sqrt{n\bar{p}\log n} + \delta_1\delta_2 \log n)/\sigma_{\min}(\bar{\mathbf{C}})$.

Theorem 5.1 holds true on general tensor structures. Specializing Theorem 5.1 to the mixture multilayer network model (Section 2) immediately yields the following corollary.

COROLLARY 1. *Assume that* (A1)–(A3) *hold. Further, assume*:

- *the initializations* $\widehat{U}^{(0)}$ *and* $\widehat{W}^{(0)}$ *are warm, that is*, $\mathrm{Err}(0) \le 1/4$,
- *the network sparsity satisfies*

$$\sqrt{Ln\bar{p}} \ge C_0\kappa_0^3\big(r^2/\sqrt{m}\big)\log(\kappa_0 r)\log^2 n. \tag{5.1}$$

Then with probability at least $1-2n^{-2}$, *after at most* $t_{\max} = O(\log n)$ *iterations*, $\mathrm{Err}(t_{\max}) \le C_3\kappa_0^2 r\cdot\sqrt{\log(n)/(mLn\bar{p})}$.

By Corollary 1, if κ_0, r, m are fixed and the network sparsity satisfies $Ln\bar{p} \ge C_0'\log^4 n$, then $\mathrm{Err}(t_{\max}) = O_p(\sqrt{\log(n)/(Ln\bar{p})})$.

5.2. *Consistency of recovering global memberships*. Recall from Section 2 that the global community structure is denoted as $\{\bar{\mathcal{V}}_j\}_{j=1}^{\bar{K}}$ where nodes i_1 and i_2 belong to the same global community if and only if $(e_{i_1}-e_{i_2})^\top \bar{Z} = 0$. In the TWIST algorithm, after applying the K-means to the rows of $\widehat{U}$, we get the vertices' global membership $\widehat{\bar{\mathbb{V}}} = \{\widehat{\bar{\mathcal{V}}}_k, k\in[\bar{K}]\}$.

We measure the performance by the Hamming error of clustering:

$$\mathcal{L}(\widehat{\bar{\mathbb{V}}}, \bar{\mathbb{V}}) = \min_{\tau:\text{ a permutation on }[\bar{K}]} \sum_{i=1}^{n}\sum_{k=1}^{\bar{K}} \mathbb{1}(i\in\bar{\mathcal{V}}_k, i\notin\widehat{\bar{\mathcal{V}}}_{\tau(k)}),$$

where we denote $\bar{\mathbb{V}} = \{\bar{\mathcal{V}}_k, k\in[\bar{K}]\}$ and $\widehat{\bar{\mathbb{V}}} = \{\widehat{\bar{\mathcal{V}}}_k, k\in[\bar{K}]\}$.

THEOREM 5.2 (Consistency of global clustering). *Assume that* (A1)–(A4) *hold, and that* $\min_k|\bar{\mathcal{V}}_k| \asymp n/\bar{K}$. *Then with probability at least* $1-n^{-2}$, *we have*

$$n^{-1}\cdot\mathcal{L}(\widehat{\bar{\mathbb{V}}}, \bar{\mathbb{V}}) \le C_3\kappa_0^6 r^2\log(n)/(Ln\bar{p})$$

provided that the network sparsity satisfies

$$\sqrt{Ln\bar{p}} \ge \big(C_1(\bar{K})^{1/2} + C_2 r\log n\big)\big(\kappa_0^3 r/\sqrt{m}\big)\log(\kappa_0 r)\log(n). \tag{5.2}$$

REMARK 1. Although bound (5.2) appears to imply that increasing m would weaken the sparsity condition, the parameters $m, r, \bar{K}$ are mutually related. For example, $m \le r^2$ and $r \le \bar{K}$ follows from Condition (A1) and the definition, respectively. Note that, for ease of exposition, we only consider the case of balanced community sizes.

From Theorem 5.2, it follows that the relative clustering error is $O_p(\log^{-3}(n))$ when $\bar{K}, m, \kappa_0$ are bounded. Therefore, vertices' global memberships can be consistently recovered. We now compare our method with some other available ones in the literature.

- In the special case $L=1$, MMSBM reduces to the standard SBM model. Then $r=\bar{K}$ and the sparsity condition (5.2) becomes $n\bar{p} \ge C_1\bar{K}^4\log^4(n)$, which is weaker than [47] but stronger than [34]. Our misclustering error is larger than [34, 47], due to additional factors emerged from tensor techniques.
- A special case when $m=1$ is considered by [33], who shows that their algorithm is able to consistently recover the communities if $n\bar{p}\sqrt{L} \gg \log^{3/2} n$. On the other hand, our result deals with more general mixture multilayer model, is computationally more efficient and requires weaker network sparsity: $n\bar{p}L \gg \log^4(n)$. As shown in Theorem 5.3, the dependence of L in (5.2) is optimal if we ignore the logarithmic term. This improvement is due to a sharper concentration inequality of $\mathbf{A}-\mathbb{E}\mathbf{A}$ in terms of tensor incoherent norm.

- A joint matrix factorization method (Co-reg) is proposed in [43] for a special case with $m = 1$ and different B_js, in which they prove that their method can consistently recover the vertices memberships if $Ln\bar{p} \gg \log n$ and the signal strengths of B_js are similar. Their network sparsity condition is similar to (5.2) up to logarithmic factor. On the other hand, our approach differs from [43] in several aspects. Our method can perform vertices clustering and network clustering simultaneously when $m > 1$. Computationally, [43] employs a BFGS algorithm to solve the nonconvex programming, which is computationally more intensive than TWIST.

We now prove that the sparsity condition (5.2) is nearly optimal up to logarithmic terms. Consider a special MMSBM with $m = 1$, $K_1 = 2$, $L_1 = L$ and define the parameter space $\Theta_{n,\bar{p}} := \{(\bar{Z}, B) : \bar{Z} \in \{0,1\}^{n\times 2}, \bar{Z}1_2 = 1_n, B = B_0 = \bar{p}[1, 0.5; 0.5, 1]\}$, where 1_n denotes the n-dimensional all one vector and $\bar{p} \in (0, 1/2)$ is a fixed constant. For any $\theta = (\bar{Z}, B) \in \Theta_{n,\bar{p}}$, we denote $\mathbb{P}_\theta$ the probability distribution of $\mathbf{A}$ generated under SBM$(\bar{Z}, B)$, and the definition of $\bar{\mathbb{V}}$ is the same as Theorem 5.2.

THEOREM 5.3 (Lower bound for global clustering). *There exist absolute constants $c_0, c_1, \beta > 0$ such that if the network sparsity satisfies $Ln\bar{p} \le c_0$, then*

$$\inf_{\hat{\bar{\mathbb{V}}}} \sup_{\theta\in\Theta_{n,\bar{p}}} \mathbb{P}_\theta\big(n^{-1}\cdot\mathcal{L}(\hat{\bar{\mathbb{V}}}, \bar{\mathbb{V}}) \ge c_1\big) \ge \beta,$$

where $\inf_{\hat{\bar{\mathbb{V}}}}$ denotes the infimum over all estimators of $\bar{\mathbb{V}}$ based on the data $\mathbf{A}$.

5.3. *Network classification.* We now show that the standard K-means algorithm on $\widehat{W}$ can consistently uncover the network classes of L layers under the network sparsity condition (5.2). Further under a slightly stronger network sparsity condition (5.3), we can apply Algorithm 2 to exactly recover the layer labels with high probability. This shows that more layers will provide more information about layer structure and be very helpful in exact clustering of networks. Similarly, denote

$$\mathcal{L}(\hat{\mathbb{S}}, \mathbb{S}) = \min_{\tau:\text{ permutation of }[m]} \sum_{l=1}^{L} \mathbb{1}\big(s_l \ne \tau(\hat{s}_l)\big).$$

THEOREM 5.4 (Consistency and exact recovery of network classes). *Let $\widetilde{\mathbb{S}} = \{\tilde{s}_l\}_{l=1}^L$ be the output of the standard K-means algorithm applied to $\widehat{W}$.*

1. *Under the same conditions in Theorem 5.2, we have, with probability at least $1 - n^{-2}$,*

$$L^{-1}\cdot\mathcal{L}(\widetilde{\mathbb{S}}, \mathbb{S}) \le C_3\kappa_0^4 r^2\log(n)/(mLn\bar{p}),$$

where $\mathbb{S} = \{s_l\}_{l=1}^L$.

2. *We further assume*

$$\sqrt{L}n\bar{p} \ge C_1 m^{-1}\kappa_0^5 r^{5/2}\log(r\kappa_0)\log^{5/2}(n). \tag{5.3}$$

There exist constants $c_1, c_2 \in (0, 1)$ such that with probability at least $1 - 3n^{-2}$,

$$\mathcal{L}(\widehat{\mathbb{S}}, \mathbb{S}) = 0, \tag{5.4}$$

where $\widehat{\mathbb{S}} = \{\hat{s}_l\}_{l=1}^L$ is the output of Algorithm 2 with parameters m and $\varepsilon \in [c_1, c_2]\sqrt{m/L}$.

By Theorem 5.4, in the case $r, m, \kappa_0 = O(1)$, Algorithm 2 is capable to exactly recover the network classes using appropriately chosen parameter ε if the network sparsity satisfies $\sqrt{L}n\bar{p} \gg \log^{5/2} n$. On the other hand, consistent network clustering requires, by (5.2), network sparsity $Ln\bar{p} \gg \log^4 n$. Therefore, condition (5.3) is stronger with respect to the number of layers L.

5.4. *Consistency of local clustering.* After obtaining the network classes, we can apply spectral clustering on $\sum_{l:\hat{s}_l=j} A_l$ to recover the local memberships $\mathbb{V}^j = \{\mathcal{V}_k^j\}_{k=1}^{K_j}$. Its consistency can be directly proved by existing results in the literature. See [43].

THEOREM 5.5. *Suppose that the conditions of Theorem 5.4 and equation (5.4) hold. For all $j \in [m]$, let $\widehat{\mathbb{V}}^j = \{\hat{\mathcal{V}}_k^j\}_{k=1}^{K_j}$ denote the output of K-means algorithm on $\sum_{l:\hat{s}_l=j} A_l$. If $\sigma_{K_j}(B_j^0) \geq c_1$ for some absolute constant $c_1 > 0$ and $|\mathcal{V}_k^j| \asymp n/K_j$ for all $k \in [K_j]$, then with probability at least $1 - n^{-2}$,*

$$n^{-1} \cdot \mathcal{L}(\widehat{\mathbb{V}}^j, \mathbb{V}^j) \leq C_1 m K_j^2 \log(n)/(Ln\bar{p}).$$

5.5. *Warm initialization for regularized power iteration.* An important condition for the success of Algorithm 1 is the existence of warm initialization,

$$\mathrm{Err}(0) = \max\{\mathrm{d}(\widehat{U}^{(0)}, \bar{U}), \mathrm{d}(\widehat{W}^{(0)}, \bar{W})\} \leq 1/4.$$

In this section, we introduce a spectral method for initializing $\widehat{U}^{(0)}$ by summing up all the layers of networks. After that, we initialize $\widehat{W}^{(0)}$ by taking the left singular vectors of $\mathcal{M}_3(\mathbf{A})(\widetilde{U}^{(0)} \otimes \widetilde{U}^{(0)})$ where $\widetilde{U}^{(0)} = \mathcal{P}_{\delta_1}(\widehat{U}^{(0)})$. Here, we abuse the notation and denote $\otimes$ the Kronecker product. The following lemma shows that these initializations are indeed close to the truth under reasonable conditions.

LEMMA 5.6 (Initialization). *Let $\widehat{U}^{(0)}$ denote the top-r left singular vectors of $\sum_{l=1}^{L} A_l$ and let $\widehat{W}^{(0)}$ be the top-r left singular vectors of $\mathcal{M}_3(\mathbf{A})(\widetilde{U}^{(0)} \otimes \widetilde{U}^{(0)})$ where $\widetilde{U}^{(0)} = \mathcal{P}_{\delta_1}(\widehat{U}^{(0)})$ with $\delta_1 = \max_{1\leq j\leq n} \|e_j^\top \bar{U}\|$. Then with probability at least $1 - 3n^{-2}$,*

$$\mathrm{d}(\widehat{U}^{(0)}, \bar{U}) \leq \min\{C_3\sqrt{n\bar{p}}\log^2(n)/\sigma_r(\bar{\mathbf{C}} \times_3 (d_L^\top/L)^{1/2}), 2\}, \tag{5.5}$$

where $d_L = (L_1, \ldots, L_m)^\top$. If $\delta_1 = O(\kappa_0\sqrt{r/n})$ and

$$\sigma_r(\bar{\mathbf{C}} \times_3 (d_L^\top/L)^{1/2}) \geq C_3'\sqrt{n\bar{p}}\log^2 n, \tag{5.6}$$

then with same probability,

$$\mathrm{d}(\widehat{W}^{(0)}, \bar{W}) \leq \min\{C_4 r\kappa_0\sqrt{n\bar{p}}\log^2(n)\log(\kappa_0 r)/\sigma_{\min}(\bar{\mathbf{C}}), 2\}.$$

Comparing the rate of initialization (5.5) and the rate after regularized power iterations in Theorem 5.1, Algorithm 1 improves the estimation error by a ratio of $\sigma_{\min}(\bar{\mathbf{C}})$ and $\sigma_r(\bar{\mathbf{C}} \times_3 (d_L^\top/L)^{1/2})$. In special cases, such an improvement can be significant. For instance, consider $m = r = 2$ and $L_1 = L_2$ and $\bar{\mathbf{C}} \in \mathbb{R}^{2\times 2\times 2}$ with

$$\bar{C}(:,:,1) = \begin{pmatrix} 1+\varepsilon & 0 \\ 0 & 1+\varepsilon \end{pmatrix} \quad \text{and} \quad \bar{C}(:,:,2) = \begin{pmatrix} 0 & 1-\varepsilon \\ 1-\varepsilon & 0 \end{pmatrix}$$

for some small number $\varepsilon \in (0, 1)$. It is easy to check that $\sigma_{\min}(\bar{\mathbf{C}}) = \sigma_2(\mathcal{M}_3(\bar{\mathbf{C}})) = \sqrt{2}(1-\varepsilon)$. On the other hand,

$$\sigma_2(\bar{\mathbf{C}} \times_3 (d_L^\top/L)^{1/2}) = \sqrt{2}\varepsilon.$$

Moreover, if $\varepsilon = 0$, then $\bar{\mathbf{C}} \times_3 (d_L^\top/L)^{1/2}$ is rank deficient implying that simply projecting the multilayer networks into a graph can potentially cause serious information loss. See more details in [24] and a similar discussion in [33].

It is worthwhile pointing out that one could use other methods to initialize $\widehat{U}^{(0)}$ (the initialization of $\widehat{W}^{(0)}$ is easy once it is done for $\widehat{U}^{(0)}$). Examples include the HOSVD by extracting the top-r left singular vectors of $\mathcal{M}_1(\mathbf{A})$, the joint matrix factorization method in [43], and random projection [24].

6. Simulation studies. We conduct several simulations to test the performance of TWIST on the MMSBM with different choices of network sparsity, "out-in" ratio, number of layers and the size of each layer. We use K-means as the clustering algorithm. The evaluation criterion is the misclustering rate. All the experiments are replicated 100 times and the average performances are reported.

We generate the data according to MMSBM in the following fashion. The underlying class s_l for the lth layer is generated from the multinomial distribution with $\mathbb{P}(s_l = j) = 1/m$, $j = 1, \ldots, m$. The membership z_i^j for node i in layer type j is generated from the multinomial distribution with $\mathbb{P}(z_i^j = s) = 1/K, s = 1, \ldots, K$. We choose the probability matrix as $B = pI_K + q(1_K 1_K^\top - I_K)$, where 1_K is a K-dimensional all-one vector and I_K is the $K \times K$ identity matrix. Let $\alpha = q/p$ be the out-in ratio.

6.1. *Global memberships*. First, we consider the task of detecting the global memberships defined in Section 2. We compare the performance of TWIST with Tucker decomposition initialized by HOSVD (HOSVD-Tucker), and we also adopt a baseline method by performing spectral clustering on the sum of adjacency matrices from all layers (Sum–Adj). Sum–Adj has been considered in literature [14, 43, 52] as a simple but effective procedure [29]. The function "*tucker*" from the R package "*rTensor*" [37] is used to apply Tucker decomposition for HOSVD-Tucker.

In simulation 1, the networks are generated with the number of nodes $n = 600$, the number of layers $L = 20$, number of types of networks $m = 3$, number of communities of each network $K = 2$ and out-in ratio of each layer $\alpha = 0.4$. The average degree d of each layer varies from 2 to 20.

In simulation 2, the setting is the same as in Simulation 1 except the average degree of each layer is fixed at $d = 10$ and the out-in ration α of each layer varies from 0.1 to 0.8.

In simulation 3, the setting is the same as in Simulation 1, except that the out-in ratio is fixed at $\alpha = 0.6$ and the number of layers L varies from 10 to 60.

In simulation 4, the setting is the same as that in Simulation 3, except that the number of layers is $L = 20$, $d = 0.02n$ and the number of nodes n varies from 100 to 1200.

In simulation 5, $n = 600$, $m = 3$, $K = 3$, the out-in ratio for each layer is drawn from a uniform distribution $\alpha_l \sim U(0.5, 0.7), l \in \{1, 2, \ldots, L\}$ the average degree for each layer is drawn from a uniform distribution $d_l \sim U(8, 12), l \in \{1, 2, \ldots, L\}$ and the number of layers L varies from 10 to 60.

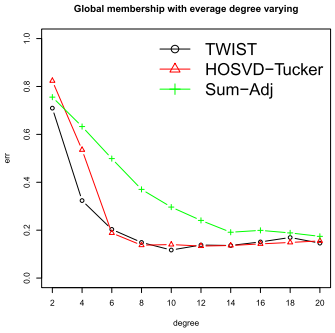
In simulation 6, $L = 30$, $m = 3$, $K = 3$, the out-in ratio for each layer is drawn from a uniform distribution $\alpha_l \sim U(0.5, 0.7), l \in \{1, 2, \ldots, L\}$, the average degree for each layer is drawn from a uniform distribution $d_l \sim U(0.015n, 0.025n), l \in \{1, 2, \ldots, L\}$ and the number of nodes n varies from 100 to 1200.

The results of simulations 1–6 are given in Figure 3.

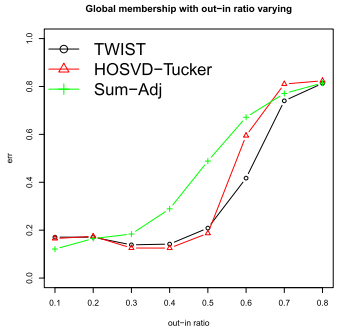
1. Clearly, the misclustering rate of all the methods decreases as the average degree of each layer increases, the out-in ratio of each layer decreases and the number of layers increases. This is consistent with our theoretical findings.

2. TWIST and HOSVD-Tucker, both utilizing tensor structure, perform much better than Sum–Adj, which only uses matrix structure. The misclustering rate of TWIST and HOSVD-Tucker decreases more rapidly.

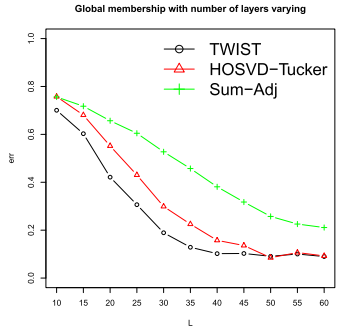
3. TWIST outperforms HOSVD-Tucker when the signal is not strong enough, for example, for $d < 6$ in Simulation 1; for $\alpha > 0.5$ in Simulation 2; for $L < 50$ in Simulations 3 and 5; and for $n < 800$ in Simulations 4 and 6.



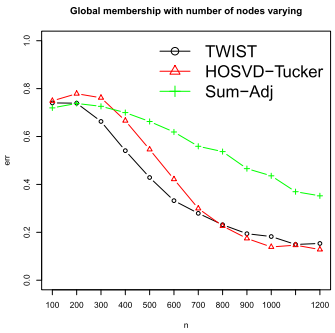
(a) The result of simulation 1: $n = 600$, $K = 2$, $m = 3$, $L = 20$, $\alpha = 0.4$, varying d.



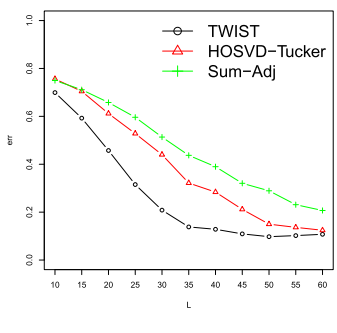
(b) The result of simulation 2: $n = 600$, $K = 2$, $m = 3$, $L = 20$, $d = 10$, varying α.



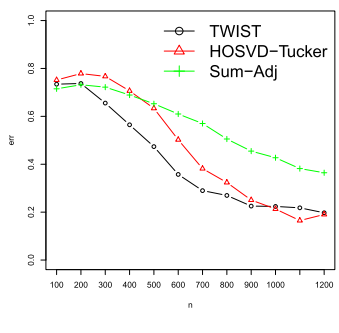
(c) The result of simulation 3: $n = 600$, $K = 2$, $m = 3$, $d = 10$, $\alpha = 0.6$, varying L.



(d) The result of simulation 4: $K = 2$, $m = 3$, $d = 0.02n$, $L = 20$, $\alpha = 0.6$, varying n.



(e) The result of simulation 5: $n = 600$, $K = 3$, $m = 3$, $d_l \sim U(8, 12)$, $\alpha_l \sim U(0.5, 0.7)$, varying L.



(f) The result of simulation 6: $K = 3$, $m = 3$, $d_l \sim U(0.015n, 0.025n)$, $\alpha_l \sim U(0.5, 0.7)$, $L = 30$, varying n.

FIG. 3. *Overall, TWIST and HOSVD-Tucker perform much better than Sum–Adj. TWIST outperforms HOSVD–Tucker when the signal is not strong enough, for instance* $d < 6$ *in* (*a*), $\alpha > 0.5$ *in* (*b*), $L < 50$ *in* (*c*) *and* (*e*) *and* $n < 800$ *in* (*d*) *and* (*f*).

6.2. *Layers' labels*. We now explore the task of clustering different types of layers. We compare TWIST with HOSVD-Tucker and spectral clustering applied to the mode-3 flatting of $\mathbf{A}$ (M3-SC).

In simulation 7, the networks are generated with the number of nodes $n = 600$, the number of layers $L = 20$, number of types of networks $m = 3$, number of communities of each network $K = 3$ and out-in ratio of each layer $\alpha = 0.6$. The average degree d of each layer varies from 3 to 30.

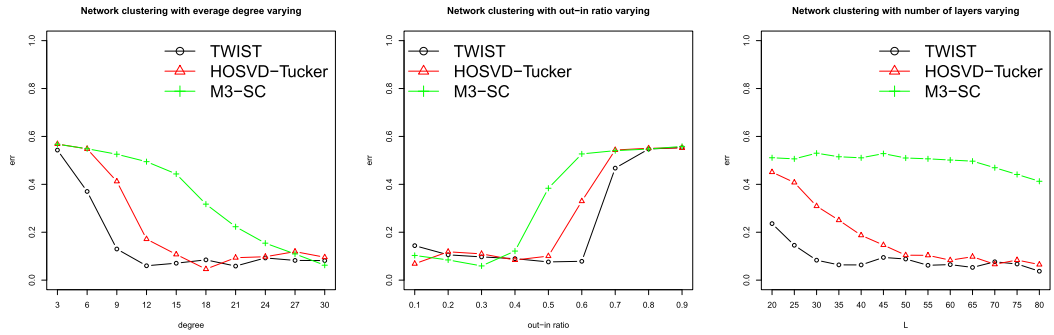
In simulation 8, the networks are generated as in simulation 7, except that the average degree of each layer $d = 10$, the number of layers $L = 30$ and the out-in ration α of each layer varies from 0.1 to 0.9.

In simulation 9, the networks are the same as in simulation 8, except that the out-in ratio $\alpha = 0.6$ and the number of layers L varies from 20 to 80.

In simulation 10, the networks are the same as in simulation 9, except that the average degree of each layer $d = 0.02n$ and the the size of each layer n varies from 100 to 1200.

In simulation 11, $n = 600$, $m = 3$, $K = 3$, the out-in ratio for each layer is drawn from a uniform distribution $\alpha_l \sim U(0.5, 0.7), l \in \{1, 2, \ldots, L\}$ the average degree for each layer is drawn from a uniform distribution $d_l \sim U(8, 12), l \in \{1, 2, \ldots, L\}$ and the number of layers L varies from 20 to 80.

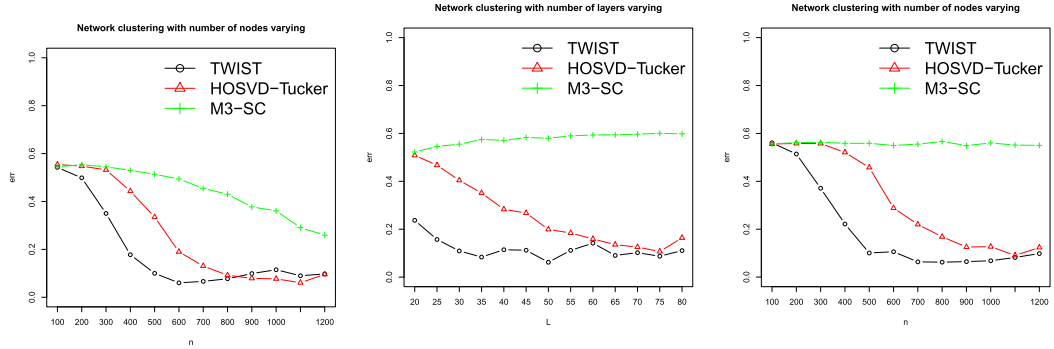
In simulation 12, $L = 30$, $m = 3$, $K = 3$, the out-in ratio for each layer is drawn from a uniform distribution $\alpha_l \sim U(0.5, 0.7), l \in \{1, 2, \ldots, L\}$, the average degree for each layer is



(a) The result of simulation 7: $n = 600$, $K = 3$, $m = 3$, $L = 20$, $\alpha = 0.6$, varying d.

(b) The result of simulation 8: $n = 600$, $K = 3$, $m = 3$, $d = 10$, $L = 30$, varying α.

(c) The result of simulation 9: $n = 600$, $K = 3$, $m = 3$, $d = 10$, $\alpha = 0.6$, varying L.



(d) The result of simulation 10: $K = 3$, $m = 3$, $d = 0.02n$, $\alpha = 0.6$, $L = 30$, varying n.

(e) The result of simulation 11: $n = 600$, $K = 3$, $m = 3$, $d_l \sim U(8, 12)$, $\alpha_l \sim U(0.5, 0.7)$, varying L.

(f) The result of simulation 12: $K = 3$, $m = 3$, $d_l \sim U(0.015n, 0.025n)$, $\alpha_l \sim U(0.5, 0.7)$, $L = 30$, varying n.

FIG. 4. *TWIST is the best overall, particularly when the signal is not strong enough, for instance,* $d < 15$ *in* (*a*), $\alpha > 0.4$ *in* (*b*), $L < 50$ *in* (*c*) *and* (*e*) *and* $n < 800$ *in* (*d*) *and* (*f*). *From Simulations* 9 *and* 11, *the naive method M3-SC hardly changes as the number of layers increases.*

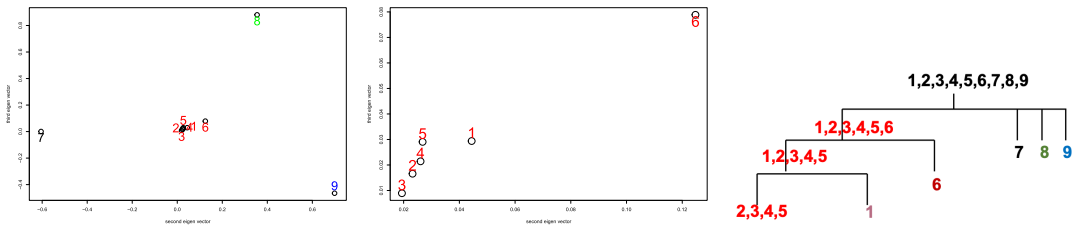
drawn from a uniform distribution $d_l \sim U(0.015n, 0.025n)$, $l \in \{1, 2, \ldots, L\}$ and the number of nodes n varies from 100 to 1200.

The results are presented in Figure 4. We make the following observations:

1. The misclustering rates of all three methods decrease as the average degree of each layer increases, the out-in ratio of each layer decreases, the number of layers increases and the size of each layer increases. This agrees with our theoretical results.
2. From Simulations 9 and 11, the naive method M3-SC shows no response to the increase of the number of layers, as might be expected.
3. Overall, TWIST performs the best among the three methods. This can be clearly seen when the signal is not strong enough, for instance, $d < 15$ in Simulation 7, $\alpha > 0.4$ in Simulation 8, $L < 50$ in Simulations 9 and 11 and $n < 800$ in Simulations 10 and 12.

7. Real data analysis. In this section, we apply TWIST to two real data sets: worldwide food trading networks and Malaria parasite genes networks. The two datasets have been studied in the literature before. However, with TWIST, we are able to make some new, interesting and sometimes surprising findings, which the earlier methods have failed to do so.

7.1. *Malaria parasite genes networks.* The var genes of the human malaria parasite Plasmodium falciparum present a challenge to population geneticists due to their extreme diversity, which is generated by high rates of recombination. Var gene sequences are characterized



(a) Embedding of each network. (b) Zoom center area in (a). (c) Hierarchical structure of network classes.

FIG. 5. *Embedding of each layer in malaria parasite genes networks and hierarchical structure of network classes.*

by pronounced mosaicism, precluding the use of traditional phylogenetic tools. Larremore et al. [30] identify 9 highly variable regions (HVRs), and then maps each HVR to a complex network; see the figure in the Supplementary Material [22]. They show that the recombinational constraints of some HVRs are correlated, while others are independent, suggesting that this micromodular structuring facilitates independent evolutionary trajectories of neighboring mosaic regions, allowing the parasite to retain protein function while generating enormous sequence diversity.

Despite the innovative network approach, there are still some drawbacks in [30].

1. Even though 9 HVRs have been identified, only 6 HVRs have been used in the analysis, while the other three HRVs are discarded due to their sparse structures, as seen in the figure in the Supplementary Material [22]. However, these sparse networks still contain valuable information, which would be of great interest to researchers and practitioners.
2. Community structures are identified individually for each network, and then compared with each other to identify similar structures. This is not only very demanding and tedious computationally, but also involves much human intervention. This becomes increasingly undesirable as the number of networks grows bigger.

Here, we propose to employ TWIST to the problem, in order to overcome the above difficulties. The data under investigation are the 9 HVRs used in [30]. Each network is derived from the same set of 307 genetic sequences from var genes of malaria parasites. A node represents a specific gene and an edge is generated by comparing sequences pairwisely within each HVR. More information about the data and data pre-processing could be found in [30]. In our study, we consider 212 nodes, which appear on all 9 layers. This results in a $212 \times 212 \times 9$ tensor, as shown in the figure in the Supplementary Material [22].

We apply TWIST to this $212 \times 212 \times 9$ tensor with a core tensor of size $15 \times 15 \times 3$. The embedding of each layer is plotted in Figure 5. We make the following comments:

1. The 9 HVRs fall into 4 groups (Figure 5(a)): $\{1, 2, 3, 4, 5, 6\}$, $\{7\}$, $\{8\}$, $\{9\}$.

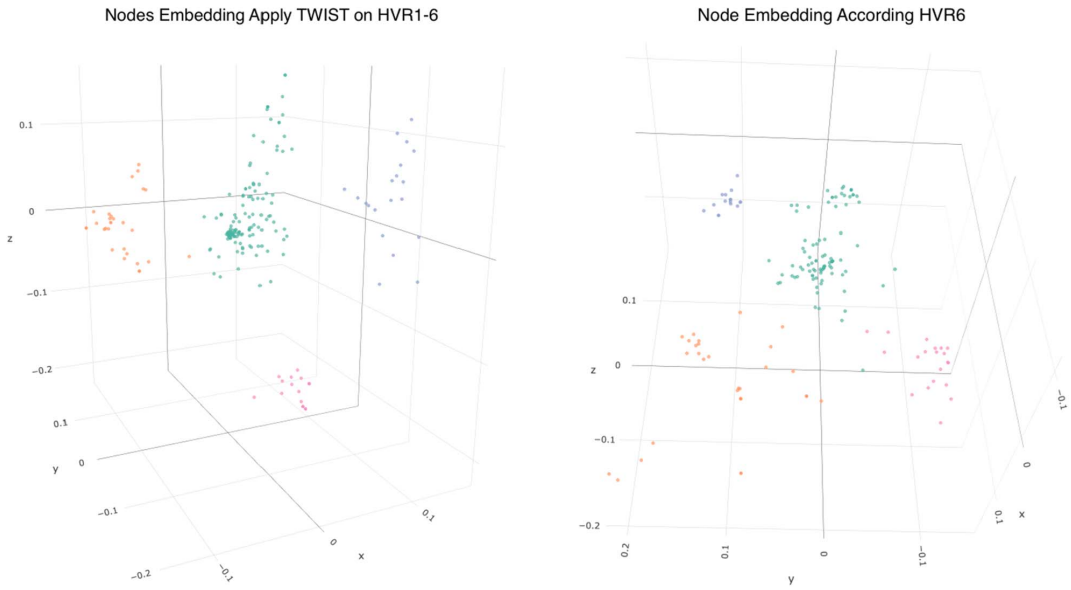
By comparison, [30] found that the 6 HVRs fall into 4 groups (without layers 2–4): $\{1, 5, 6\}$, $\{7\}$, $\{8\}$, $\{9\}$. The two findings are consistent.

2. TWIST places sparse networks of layers 2–4 to the same group as layers 1, 5 and 6.

By comparison, the sparse layers 2–4 had to be discarded in [30]. The new result implies that the sequences remain mostly unchanged in the beginning (HVRs 1–6), and start to diversify from HVR 7 onward.

3. Hierarchical structure of the 9 HVRs.

If we zoom in the mini group $\{1, 2, 3, 4, 5, 6\}$ (Figure 5(b)), we notice that the first 5 layers are more tied together, so we have a finer partition: $\{1, 2, 3, 4, 5\}$, $\{6\}$. This operation can be repeated. Therefore, TWIST can be easily used to form a hierarchical structure of the 9 HVRs (Figure 5(c)).



(a) TWIST applied to HVRs 1-6. (b) Spectral clustering applied to HVR 6.

FIG. 6. *Nodes embedding using TWIST and spectral decompositions with* $K = 4$.

4. Computational ease of TWIST

TWIST can easily cluster layers and nodes using K-means. This is much easier than the procedure in [30], which first finds the community structure for each layer, and then computes their similarities.

5. Better community structure is obtained by combining information from similar layers.

TWIST is applied to the first 6 similar layers $\{1, 2, 3, 4, 5, 6\}$ to identify their common local structure, while spectral clustering is applied to HVR 6 to find its community structure as was done in [30]; see Figure 6(a)–(b). Clearly, the 4 local communities are much more separated in Figure 6(a) than in (b).

7.2. *Worldwide food trading networks*. We consider the data set on the worldwide food trading networks, which is collected by [13], and is available at http://www.fao.org. The data contains an economic network in which layers represent different products, nodes are countries and edges at each layer represent trading relationships of a specific food product among countries.

We focus on the trading data in 2010 only. We convert the original directed networks to undirected ones by ignoring the directions. We delete the links with weight less than 8 (the first quartile) and abandon the layers whose largest component consists of less than 150 nodes. These are done to filter out the less important information. Finally, we extract the intersections of the largest components of the remaining layers.

After data preprocessing, we obtained a 30-layers network with 99 nodes at each layer. Each layer represents trading relationships between 99 countries/regions worldwide with respect to one of the 30 different food products. Together they form a third-order tensor of dimension $99 \times 99 \times 30$.

We first apply Algorithm 1 in the TWIST procedure to the data tensor, which results in a tensor decomposition with a core tensor of dimension $20 \times 20 \times 2$. The resulting two clusters of layers are listed in Table 1. We then apply Algorithm 2 in the TWIST procedure to each cluster of networks separately (here we have two clusters) to find the community structures for each cluster, in order to obtain the clustering result of countries. This time, we take the

TABLE 1
The resulting two clusters of layers

Food cluster 1:	Beverages nonalcoholic, Food prep nes, Chocolate products nes, Crude materials, Fruit prepared nes, Beverages distilled alcoholic, Coffee green, Pastry, Sugar confectionery, Wine, Tobacco unmanufactured
Food cluster 2:	Cheese whole cow milk, Cigarettes, Flour wheat, Beer of barley, Cereals breakfast, Milk skimmed dried, Juice fruit nes, Maize, Macaroni, Oil palm, Milk whole dried, Oil essential nes, Rice milled, Sugar refined, Tea, Spices nes, Vegetables preserved nes, Waters ice, etc, Vegetables fresh nes

core tensor of dimension $4 \times 4 \times 1$. The embedding of 99 countries with clustering results from K-means are shown in Figure 7. For the two types of networks, we plot in Figure 8 the sum of adjacency matrices with nodes arranged according to the community labels to have a glance of different community structures of two network types.

We make the following remarks from Table 1, Figures 7 and 8.

1. Trading patterns of food are different for unprocessed and processed foods.

Specifically, cluster 1 consists mainly of raw or unprocessed food (e.g., crude materials, coffee green, unmanufactured tobacco), while cluster 2 is mainly made of processed food (e.g., such as cigarettes, flour wheat, essential oil, milled rice, refined sugar).

2. For unprocessed food, global trading is the more dominant trading pattern than regional one. Some countries have closer trading ties with countries across the globe.

From cluster 1, a small number of countries, such as China, Canada, United Kingdom, United States, France, Germany, are very active in trading with others as well as among themselves. This small group of countries is called a hub community. This reflects the fact that these large countries import unprocessed food from, and/or export unprocessed food to a great number of other countries worldwide.

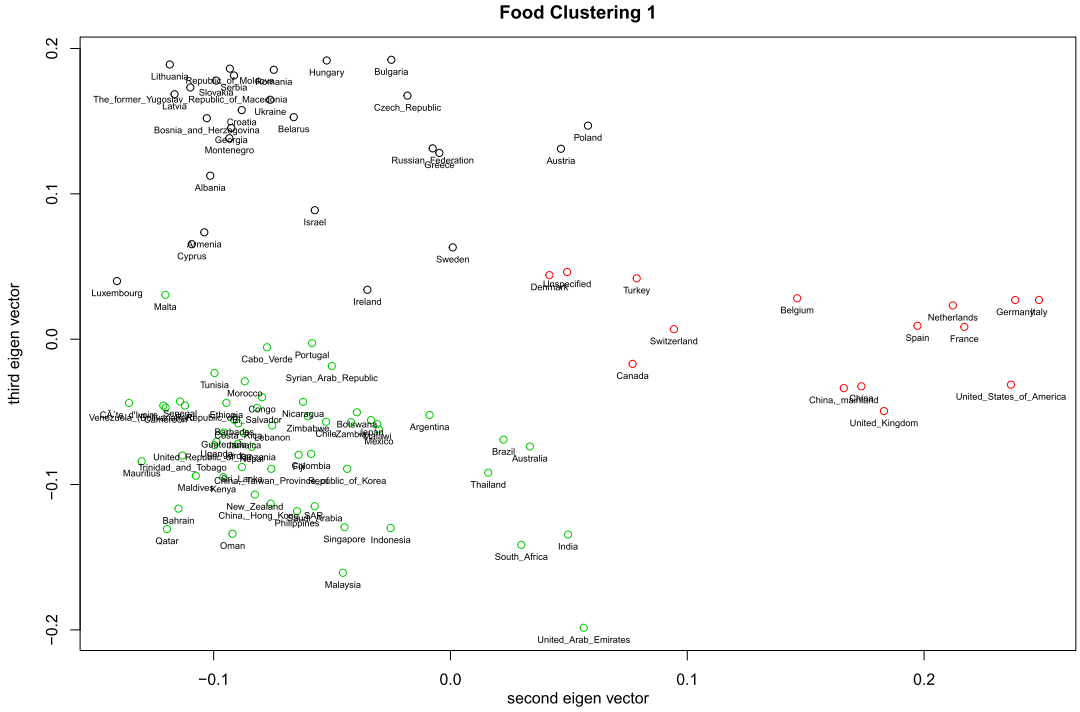
3. For processed foods, regional trading is very dominant. In fact, the world trading map is strikingly similar to the world geography map in Figure 7(b).

In cluster 2, countries are mainly clustered by the geographical location, that is, countries in the same continent have closer trading ties. Examples of these clusters include countries in America (United State, Canada, Mexico, Brazil, Chile), in Asia and Africa (China, Japan, Singapore, Thailand, Indonesia, Philippines, India) and in Europe (Germany, Italy, Poland, Spain, Denmark, Switzerland). Regional trading of processed food can have many advantages, for example, keeping the food cost low due to lower transportation cost, and keeping food fresh due to faster delivery.

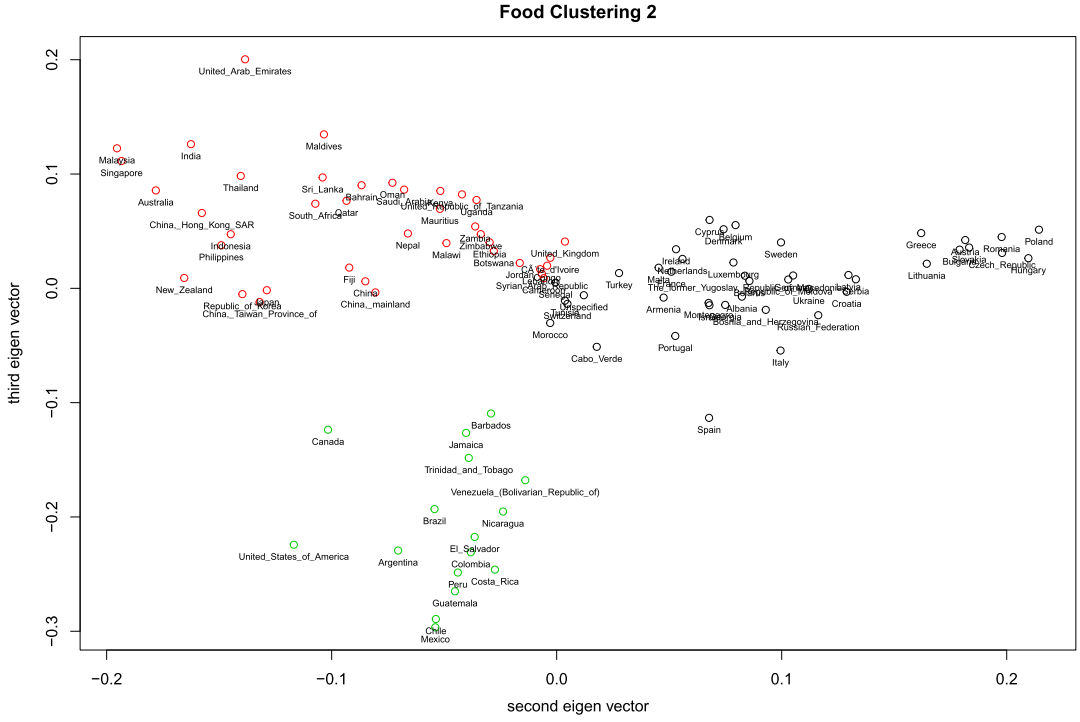
There are some interesting "outliers" as well. For instance, United Kingdom has closer trading ties with African and Middle Eastern countries than its European neighbor, which might be interesting to delve into further.

8. Conclusion and discussion. In this paper, we have proposed a novel mixture multilayer stochastic block model (MMSBM) to capture the intrinsic local as well as global community structures. A tensor-based algorithm, TWIST, was proposed to conduct community detection on multilayer networks and shown to be consistent under generally weak conditions in the MMSBM framework. In particular, the method allows for very sparse networks in many layers. The proposed method outperforms other state of the art methods both in nodes community detection and layers clustering by extensive simulation studies. We also applied the algorithm to two real data sets and found some interesting results.

A number of future directions are worth exploring. As a natural extension, one can generalize the tensor-based representation to account for adjacency matrices capturing the degree

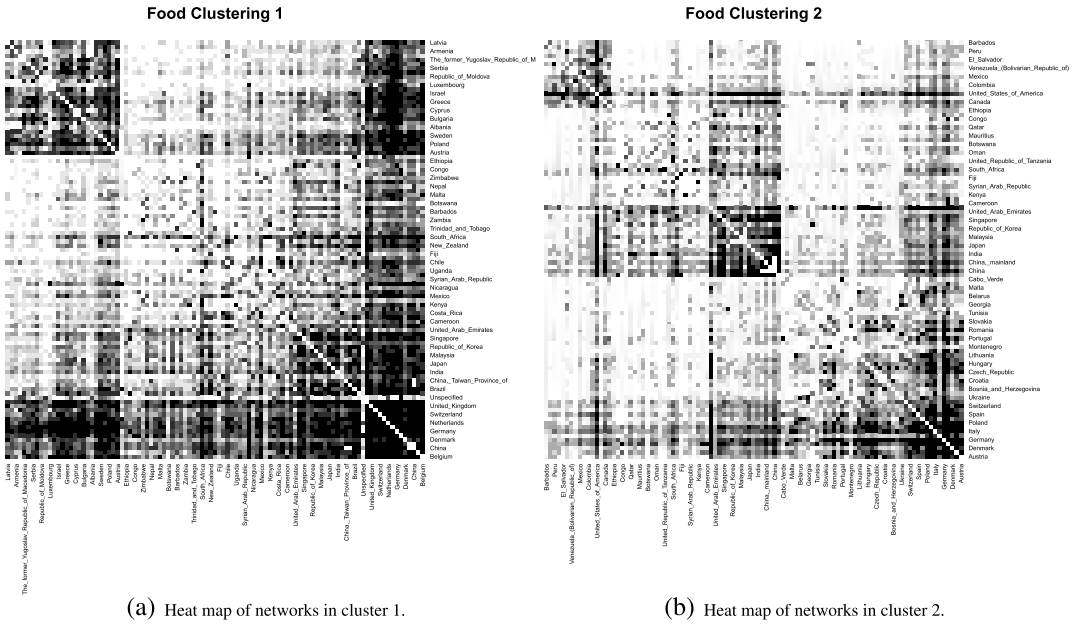


(a) Embedding of countries for networks in cluster 1.



(b) Embedding of countries for networks in cluster 2.

FIG. 7. *Embedding of countries on two different types of food trading networks.*



(a) Heat map of networks in cluster 1. (b) Heat map of networks in cluster 2.

FIG. 8. *Heat maps of two types of networks.*

heterogeneity of nodes. The layers of networks could have the spatial and temporal structures of networks in many real-world applications, so one could incorporate these into the model. On a more theoretical level, it is of interest to explore theoretical properties in other random graph models. It is also important to develop scalable algorithms that can handle millions of nodes with thousands of layers in this big data era.

Estimation of the *block* probability matrices $\{B_j\}_{j=1}^m$ can be useful in applications; see, for example, [15, 45] and [53]. The low-rank expected adjacency tensor $\mathbb{E}(\mathbf{A}|\mathbb{S})$, also called probability tensor, is built from B_j's. The step 1 of TWIST, seeking a low-rank approximation of $\mathbf{A}$, immediately yields an estimate of the probability tensor. However, this procedure cannot directly deliver the estimates for B_j's. Toward that end, after the step 4 of TWIST, one can utilize the learned local membership $\widehat{\mathbb{V}}_j$ to construct the respective membership matrix $\widehat{Z}_j \in \{0,1\}^{n\times K_j}$. Finally, one can estimate the block probability matrix by

$$\widehat{B}_j = (\widehat{Z}_j^\top \widehat{Z}_j)^{-1} \widehat{Z}_j^\top \Big(\frac{1}{|\{l : \hat{s}_l = j\}|} \sum_{l:\hat{s}_l=j} A_l \Big) \widehat{Z}_j (\widehat{Z}_j^\top \widehat{Z}_j)^{-1} \quad \forall j \in [m].$$

The error of $\widehat{B}_j$ relies crucially on the clustering performances of layers, that is, how many layers in $\{l : \hat{s}_l = j\}$ really belong to the same SBM. Interested readers are suggested to refer to a recent work [15] for the methodology and theory of estimating block probability matrices under MMSBM.

Statistical inferences, for example, for the number of local communities, under MMSBM are of great importance. After the step 3 of TWIST, based on the collection of $\{\hat{A}_l : \hat{s}_l = j\}$, we can extend many existing approaches, for example, based on the distributions of singular values [8, 32] or network moments [7, 16, 64], to test the number of local communities. Substantial efforts are required to investigate the theoretical performances of these methods under MMSBM, which we leave for future works.

Acknowledgments. The authors thank the Editor, Associate Editor and four anonymous referees for their constructive comments on an earlier version of the manuscript.

Funding. Jing and Li's research is partially supported by the HK RGC Grants GRF 16304419 and GRF 16305616. Lyu and Xia's research is partially supported by the HK RGC Grant ECS 26302019, GRF Grant 16303320 and WeBank-HKUST project WEB19EG01-g.

SUPPLEMENTARY MATERIAL

Supplement to "Community detection on mixture multilayer networks via regularized tensor decomposition" (DOI: 10.1214/21-AOS2079SUPP; .pdf). The supplementary file contains all the technical proofs, some more examples and remarks.

REFERENCES

[1] ABBE, E., FAN, J., WANG, K. and ZHONG, Y. (2020). Entrywise eigenvector analysis of random matrices with low expected rank. *Ann. Statist.* **48** 1452–1474. MR4124330 https://doi.org/10.1214/19-AOS1854

[2] ARROYO, J., ATHREYA, A., CAPE, J., CHEN, G., PRIEBE, C. E. and VOGELSTEIN, J. T. (2019). Inference for multiple heterogeneous networks with a common invariant subspace. Preprint. Available at arXiv:1906.10026.

[3] ATHREYA, A., PRIEBE, C. E., TANG, M., LYZINSKI, V., MARCHETTE, D. J. and SUSSMAN, D. L. (2016). A limit theorem for scaled eigenvectors of random dot product graphs. *Sankhya A* **78** 1–18. MR3494576 https://doi.org/10.1007/s13171-015-0071-x

[4] BACCO, C. D., POWER, E. A., LARREMORE, D. B. and MOORE, C. (2017). Community detection, link prediction, and layer interdependence in multilayer networks. *Phys. Rev. Lett.* **95** 042317. https://doi.org/10.1103/PhysRevE.95.042317

[5] BEN AROUS, G., MEI, S., MONTANARI, A. and NICA, M. (2019). The landscape of the spiked tensor model. *Comm. Pure Appl. Math.* **72** 2282–2330. MR4011861 https://doi.org/10.1002/cpa.21861

[6] BHATTACHARYYA, S. and CHATTERJEE, S. (2018). Spectral clustering for multiple sparse networks: I. Preprint. Available at arXiv:1805.10594.

[7] BICKEL, P. J., CHEN, A. and LEVINA, E. (2011). The method of moments and degree distributions for network models. *Ann. Statist.* **39** 2280–2301. MR2906868 https://doi.org/10.1214/11-AOS904

[8] BICKEL, P. J. and SARKAR, P. (2016). Hypothesis testing for automated community detection in networks. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **78** 253–273. MR3453655 https://doi.org/10.1111/rssb.12117

[9] CAI, C., LI, G., CHI, Y., POOR, H. V. and CHEN, Y. (2021). Subspace estimation from unbalanced and incomplete data matrices: $\ell_{2,\infty}$ statistical guarantees. *Ann. Statist.* **49** 944–967. MR4255114 https://doi.org/10.1214/20-aos1986

[10] CAI, C., LI, G., POOR, H. V. and CHEN, Y. (2019). Nonconvex low-rank tensor completion from noisy data. In *Advances in Neural Information Processing Systems* 1861–1872.

[11] CATTELL, R. B. (1966). The scree test for the number of factors. *Multivar. Behav. Res.* **1** 245–276.

[12] CHEN, Y., FAN, J., MA, C. and WANG, K. (2019). Spectral method and regularized MLE are both optimal for top-K ranking. *Ann. Statist.* **47** 2204–2235. MR3953449 https://doi.org/10.1214/18-AOS1745

[13] DOMENICO, M. D., NICOSIA, V., ARENAS, A. and LATORA, V. (2015). Structural reducibility of multilayer networks. *Nat. Commun.* **6** 6864. https://doi.org/10.1038/ncomms7864

[14] DONG, X., FROSSARD, P., VANDERGHEYNST, P. and NEFEDOV, N. (2012). Clustering with multilayer graphs: A spectral perspective. *IEEE Trans. Signal Process.* **60** 5820–5831. MR2990287 https://doi.org/10.1109/TSP.2012.2212886

[15] FAN, X., PENSKY, M., YU, F. and ZHANG, T. (2021). ALMA: Alternating minimization algorithm for clustering mixture multilayer network. Preprint. Available at arXiv:2102.10226.

[16] GAO, C. and LAFFERTY, J. (2017). Testing network structure using relations between small subgraph probabilities. Preprint. Available at arXiv:1704.06742.

[17] HAN, Q., XU, K. and AIROLDI, E. (2015). Consistent estimation of dynamic and multi-layer block models. In *International Conference on Machine Learning* 1511–1520.

[18] HILLAR, C. J. and LIM, L.-H. (2013). Most tensor problems are NP-hard. *J. ACM* **60** Art. 45, 39. MR3144915 https://doi.org/10.1145/2512329

[19] JAIN, P. and OH, S. (2014). Provable tensor factorization with missing data. In *Advances in Neural Information Processing Systems* 1431–1439.

[20] JIN, J. (2015). Fast community detection by SCORE. *Ann. Statist.* **43** 57–89. MR3285600 https://doi.org/10.1214/14-AOS1265

[21] JIN, J., KE, Z. T. and LUO, S. (2017). Estimating network memberships by simplex vertex hunting. Preprint. Available at arXiv:1708.07852.

[22] JING, B.-Y., LI, T., LYU, Z. and XIA, D. (2021). Supplement to "Community detection on mixture multilayer networks via regularized tensor decomposition." https://doi.org/10.1214/21-AOS2079SUPP
[23] JOSSE, J. and HUSSON, F. (2012). Selecting the number of components in principal component analysis using cross-validation approximations. *Comput. Statist. Data Anal.* **56** 1869–1879. MR2892383 https://doi.org/10.1016/j.csda.2011.11.012
[24] KE, Z. T., SHI, F. and XIA, D. (2019). Community detection for hypergraph networks via regularized tensor power iteration. Preprint. Available at arXiv:1909.06503.
[25] KIM, C., BANDEIRA, A. S. and GOEMANS, M. X. (2018). Stochastic block model for hypergraphs: Statistical limits and a semidefinite programming approach. Preprint. Available at arXiv:1807.02884.
[26] KIVELÄ, M., ARENAS, A., BARTHELEMY, M., GLEESON, J. P., MORENO, Y. and PORTER, M. A. (2014). Multilayer networks. *J. Complex Netw.* **2** 203–271.
[27] KOLDA, T. G. and BADER, B. W. (2009). Tensor decompositions and applications. *SIAM Rev.* **51** 455–500. MR2535056 https://doi.org/10.1137/07070111X
[28] KOLTCHINSKII, V. and XIA, D. (2016). Perturbation of linear forms of singular vectors under Gaussian noise. In *High Dimensional Probability VII. Progress in Probability* **71** 397–423. Springer, Cham. MR3565274 https://doi.org/10.1007/978-3-319-40519-3_18
[29] KUMAR, A., RAI, P. and DAUME, H. (2011). Co-regularized multi-view spectral clustering. In *Advances in Neural Information Processing Systems* 1413–1421.
[30] LARREMORE, D. B., CLAUSET, A. and BUCKEE, C. O. (2013). A network approach to analyzing highly recombinant malaria parasite genes. *PLoS Comput. Biol.* **9** e1003268. https://doi.org/10.1371/journal.pcbi.1003268
[31] LE, C. M., LEVIN, K. and LEVINA, E. (2018). Estimating a network from multiple noisy realizations. *Electron. J. Stat.* **12** 4697–4740. MR3894068 https://doi.org/10.1214/18-ejs1521
[32] LEI, J. (2016). A goodness-of-fit test for stochastic block models. *Ann. Statist.* **44** 401–424. MR3449773 https://doi.org/10.1214/15-AOS1370
[33] LEI, J., CHEN, K. and LYNCH, B. (2020). Consistent community detection in multi-layer network data. *Biometrika* **107** 61–73. MR4064140 https://doi.org/10.1093/biomet/asz068
[34] LEI, J. and RINALDO, A. (2015). Consistency of spectral clustering in stochastic block models. *Ann. Statist.* **43** 215–237. MR3285605 https://doi.org/10.1214/14-AOS1274
[35] LEI, L., LI, X. and LOU, X. (2020). Consistency of spectral clustering on hierarchical stochastic block models. Preprint. Available at arXiv:2004.14531.
[36] LEVIN, K., LODHIA, A. and LEVINA, E. (2019). Recovering low-rank structure from multiple networks with unknown edge distributions. Preprint. Available at arXiv:1906.07265.
[37] LI, J., BIEN, J. and WELLS, M. T. (2018). rTensor: An R package for multidimensional array (tensor) unfolding, multiplication, and decomposition. *J. Stat. Softw.* **87** 1–31.
[38] LI, T., LEI, L., BHATTACHARYYA, S., SARKAR, P., BICKEL, P. J. and LEVINA, E. (2018). Hierarchical community detection by recursive partitioning. Preprint. Available at arXiv:1810.01509.
[39] LYZINSKI, V., TANG, M., ATHREYA, A., PARK, Y. and PRIEBE, C. E. (2017). Community detection and classification in hierarchical stochastic blockmodels. *IEEE Trans. Netw. Sci. Eng.* **4** 13–26. MR3625952 https://doi.org/10.1109/TNSE.2016.2634322
[40] MATIAS, C. and MIELE, V. (2017). Statistical clustering of temporal networks through a dynamic stochastic block model. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **79** 1119–1141. MR3689311 https://doi.org/10.1111/rssb.12200
[41] NICKEL, M., TRESP, V. and KRIEGEL, H.-P. (2011). A three-way model for collective learning on multi-relational data. In *ICML* **11** 809–816.
[42] PAUL, S. and CHEN, Y. (2016). Consistent community detection in multi-relational data through restricted multi-layer stochastic blockmodel. *Electron. J. Stat.* **10** 3807–3870. MR3579677 https://doi.org/10.1214/16-EJS1211
[43] PAUL, S. and CHEN, Y. (2020). Spectral and matrix factorization methods for consistent community detection in multi-layer networks. *Ann. Statist.* **48** 230–250. MR4065160 https://doi.org/10.1214/18-AOS1800
[44] PAUL, S. and CHEN, Y. (2020). A random effects stochastic block model for joint community detection in multiple networks with applications to neuroimaging. *Ann. Appl. Stat.* **14** 993–1029. MR4117838 https://doi.org/10.1214/20-AOAS1339
[45] PENSKY, M. and ZHANG, T. (2019). Spectral clustering in the dynamic stochastic block model. *Electron. J. Stat.* **13** 678–709. MR3914178 https://doi.org/10.1214/19-ejs1533
[46] RICHARD, E. and MONTANARI, A. (2014). A statistical model for tensor PCA. In *Advances in Neural Information Processing Systems* 2897–2905.
[47] ROHE, K., CHATTERJEE, S. and YU, B. (2011). Spectral clustering and the high-dimensional stochastic blockmodel. *Ann. Statist.* **39** 1878–1915. MR2893856 https://doi.org/10.1214/11-AOS887

[48] SHEEHAN, B. N. and SAAD, Y. (2007). Higher order orthogonal iteration of tensors (HOOI) and its relation to PCA and GLRAM. In *Proceedings of the* 2007 *SIAM International Conference on Data Mining* 355–365. SIAM, Philadelphia.

[49] STANLEY, N., SHAI, S., TAYLOR, D. and MUCHA, P. J. (2016). Clustering network layers with the strata multilayer stochastic block model. *IEEE Trans. Netw. Sci. Eng.* **3** 95–105. MR3515211 https://doi.org/10.1109/TNSE.2016.2537545

[50] SUN, W. W., LU, J., LIU, H. and CHENG, G. (2017). Provable sparse tensor decomposition. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **79** 899–916. MR3641413 https://doi.org/10.1111/rssb.12190

[51] TANG, R., TANG, M., VOGELSTEIN, J. T. and PRIEBE, C. E. (2017). Robust estimation from multiple graphs under gross error contamination. Preprint. Available at arXiv:1707.03487.

[52] TANG, W., LU, Z. and DHILLON, I. S. (2009). Clustering with multiple graphs. In 2009 *Ninth IEEE International Conference on Data Mining* 1016–1021. IEEE, New York.

[53] WANG, D., YU, Y. and RINALDO, A. (2018). Optimal change point detection and localization in sparse dynamic networks. Preprint. Available at arXiv:1809.09602.

[54] WANG, M. and LI, L. (2020). Learning from binary multiway data: Probabilistic tensor decomposition and its statistical optimality. *J. Mach. Learn. Res.* **21** Paper No. 154. MR4209440

[55] XIA, D. and YUAN, M. (2019). On polynomial time methods for exact low-rank tensor completion. *Found. Comput. Math.* **19** 1265–1313. MR4029842 https://doi.org/10.1007/s10208-018-09408-6

[56] XIA, D. and YUAN, M. (2021). Statistical inferences of linear forms for noisy matrix completion. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **83** 58–77. MR4220984 https://doi.org/10.1111/rssb.12400

[57] XIA, D., YUAN, M. and ZHANG, C.-H. (2021). Statistically optimal and computationally efficient low rank tensor completion from noisy entries. *Ann. Statist.* **49** 76–99. MR4206670 https://doi.org/10.1214/20-AOS1942

[58] XIA, D. and ZHOU, F. (2019). The sup-norm perturbation of HOSVD and low rank tensor denoising. *J. Mach. Learn. Res.* **20** Paper No. 61. MR3960915

[59] YUAN, M. and ZHANG, C.-H. (2017). Incoherent tensor norms and their applications in higher order tensor completion. *IEEE Trans. Inf. Theory* **63** 6753–6766. MR3707566 https://doi.org/10.1109/TIT.2017.2724549

[60] ZHANG, A. (2019). Cross: Efficient low-rank tensor completion. *Ann. Statist.* **47** 936–964. MR3909956 https://doi.org/10.1214/18-AOS1694

[61] ZHANG, A. and HAN, R. (2019). Optimal sparse singular value decomposition for high-dimensional high-order data. *J. Amer. Statist. Assoc.* **114** 1708–1725. MR4047294 https://doi.org/10.1080/01621459.2018.1527227

[62] ZHANG, A. and XIA, D. (2018). Tensor SVD: Statistical and computational limits. *IEEE Trans. Inf. Theory* **64** 7311–7338. MR3876445 https://doi.org/10.1109/TIT.2018.2841377

[63] ZHANG, J. and CAO, J. (2017). Finding common modules in a time-varying network with application to the *Drosophila melanogaster* gene regulation network. *J. Amer. Statist. Assoc.* **112** 994–1008. MR3735355 https://doi.org/10.1080/01621459.2016.1260465

[64] ZHANG, Y. and XIA, D. (2020). Edgeworth expansions for network moments. Preprint. Available at arXiv:2004.06615.