



The Bethe Hessian and Information Theoretic Approaches for Online Change-Point Detection in Network Data

Neil Hwang

City University of New York - Bronx Community College, New York, USA

Jiarui Xu and Sharmodeep Bhattacharyya 

Oregon State University, Corvallis, USA

Shirshendu Chatterjee

City University of New York - City College and Graduate Center, New York, USA

Abstract

Sequences of networks are currently a common form of network data sets. Identification of structural change-points in a network data sequence is a natural problem. The problem of change-point detection can be classified into two main types - *offline* change-point detection and *online* or *sequential* change-point detection. In this paper, we propose three different algorithms for *online* change-point detection based on certain cusum statistics for network data with community structures. For two of the proposed algorithms, we use information theoretic measures to construct the statistic for the estimation of a change-point. In the third algorithm, we use eigenvalues of the Bethe Hessian matrix to construct the statistic for the estimation of a change-point. We show the consistency property of the estimated change-point theoretically under networks generated from the multi-layer stochastic block model and the multi-layer degree-corrected block model. We also conduct an extensive simulation study to demonstrate the key properties of the algorithms as well as their efficacy.

AMS (2000) subject classification. Primary 62H30, 62F12; Secondary 91D30.
Keywords and phrases. Change-point detection, Bethe hessian operator, Spectral clustering, Community detection, Sparse networks, Variation of information.

N. Hwang was partially supported by the Rich Internship awarded by the Department of Mathematics, City College of New York, CUNY, in Summer 2019, and S. Chatterjee was partially supported by the PSC-CUNY Enhanced Research Award #62781-00 50

Supplementary Information The online version of this article (<https://doi.org/10.1007/s13171-021-00248-1>) contains supplementary material, which is available to authorized users.

1 Introduction

Network data sets have grown vastly in size and complexity in recent decades with the rapid advances in data generation and collection technologies. Although statistical analysis of network data initially focused on single networks, the study of multiple network data sets has gained a larger audience with the advent of multilayer and temporal networks. A special form of multiple networks involves network sequences, both dependent and independent. Network sequence datasets have emerged in several fields of study, including time-series of social networks (Panisson et al., 2013; Stopczynski et al., 2014; Rocha et al., 2010; Van de Bunt et al., 1999; Mislove, 2009; Hogg and Lerman, 2012), epidemiological networks (Salathé et al., 2010; Rocha et al., 2011; Masuda and Holme, 2017), animal networks (Gates and Woolhouse, 2015; Lahiri and Berger-Wolf, 2007; Chen et al., 2015), mobile and online communication networks (Krings et al., 2012; Ferraz Costa et al., 2015; Jacobs et al., 2015; Viswanath et al., 2009; Omodei et al., 2015), economic networks (Popović et al., 2014; Zhang et al., 2014; Zhao et al., 2018), brain networks (Park and Friston, 2013; Sporns, 2013; Thompson et al., 2017), genetic networks (Rigbolt et al., 2011) and ecological networks (Blonder et al., 2012), to name a few. Analysis of network sequences in terms of modeling, summary statistical analysis, analysis of dynamics, community detection, and change-point detection has been investigated in several recent works (see Holme and Saramäki (2012), Holme (2015), Peixoto (2015), Sikdar et al. (2016), & Peixoto and Gauvin (2018) for some review of recent works). In this paper, we concentrate on the problem of change-point detection for network sequences.

The study of change-point detection has a long history in the statistics literature, starting from the early days of quality control (Page, 1954; 1957; Girshick and Rubin, 1952) to recent genomic studies (Siegmund, 2013). Applications of statistical methods for change-point detection are widespread. The disciplines where statistical analysis has been used for change-point detection include medical diagnostics (Yang et al., 2006; Staudacher et al., 2005; Bosc et al., 2003; Cribben et al., 2012), gene expression (Picard et al., 2005; Hocking et al., 2013; Bleakley and Vert, 2011), online activity (Lévy-Leduc et al., 2009), speech and image analysis (Harchaoui et al., 2009; Radke et al., 2005; Kasetkasem and Varshney, 2002; Celik, 2009; 2010), climate science (Reeves et al., 2007), economics (Bai and Perron, 1998) and finance (Lavielle and Teyssiere, 2007; Matteson and James, 2014). The study of the change-point detection problem started with Gaussian models with changes in the mean parameter (Page, 1954), but since then the models studied for

structural change-point detection has varied widely, ranging from parametric multivariate data models (Chen and Gupta, 2011) and non-parametric models (Brodsky and Darkhovsky, 2013) to models for dependent and time-series data (Cho and Fryzlewicz, 2015; Aminikhanghahi and Cook, 2017).

The change-point detection problem can be broadly classified into two types.

1. *Offline change-point detection*: The whole data sequence is available and the change-points are detected within the data sequence. This problem was studied in the beginning by Page (1954) and Girshick and Rubin (1952).
2. *Online or sequential change-point detection*: The data is available sequentially and change-points are detected based on the available data. This problem in classical setting was initially studied by Kolmogorov (1950), Shiryaev (1963), Lorden et al. (1971) and others.

There is deep literature on both types of change-point detection problems and possible methods and theories. An excellent account can be found in the book (Brodsky and Darkhovsky, 2013). *In this paper, we concentrate on the problem of online change-point detection for network sequence data.*

The problem of change-point detection in sequences of network data has recently received some interest with the increase in the availability of multiple network data sets. However, most efforts have concentrated on the detection of offline change-points. For instance, Lévy-Leduc et al. (2009) was an early work on offline change-point detection in networks using hypothesis testing, Peel and Clauset (2015) used a hierarchical random graph model and a Bayesian procedure to detect change-points, Park et al. (2013) used local graph statistics for change-point and anomaly detection in dynamic networks, and Roy et al. (2017) used a Markov random field model for generating networks and estimated the change-point using a penalized pseudo-likelihood. Another point to note is that one approach to determining change-points is by comparing networks, so hypothesis tests for network comparisons (e.g. Bickel and Sarkar (2016), Wang et al. (2017), Cape et al. (2017), Gao and Lafferty (2017), & Jin et al. (2018)) can also be used for change-point detection in network data with some modification. For a survey of techniques used in the related problem of anomaly detection in graphs, see Ranshous et al. (2015). Some recent works (Wang et al., 2017; Bhamidi et al., 2018; Bao and Michailidis, 2018; Wang et al., 2018; Bhattacharjee et al., 2018; Wills and Meyer, 2019; Padilla et al., 2019; Zhao et al., 2019) propose methods for offline change-point detection in networks generated

from block models and graphon models with some theoretical results on the consistency of the detection methods. Development and analysis of online change-point detection methods for network data are relatively rare with some possible exceptions such as (Chen et al., 2019).

In this paper, we focus on the problem of online change-point detection in network data with community structures. We consider the special case where the change-point occurs due to the change in the community structure of the networks. At the same time, the connection probabilities and the degree parameters remain the same. The main contributions of our work are as follows.

- (a) We propose two types of algorithms for online change-point detection based on network data. For the first type, we consider a statistic that captures the variation of information between estimated community structures (denoted $\hat{\mathbf{Z}}$) to construct two algorithms for online change-point detection. The two algorithms are based on two different cumulative sum measures (cusum) of the network adjacency matrices to estimate community structures. For the second type, we develop a statistic based on the eigenvalues of window-sums of the Bethe Hessian matrices obtained from the input networks. We construct an algorithm that uses this statistic to detect a change in estimated numbers of communities (denoted \hat{K}).
- (b) We provide the consistency results for the change-point estimators for all three algorithms under multilayer versions of the stochastic block model (MSBM). For the $\hat{\mathbf{Z}}$ -based algorithms, we also provide the consistency results for the multilayer degree-corrected block models. For the \hat{K} -based algorithm, we prove the theoretical results only for the MSBM.
- (c) We provide extensive simulation results to demonstrate the three algorithms' efficacy for detecting change-points in an online setup.

The paper is structured as follows. In Section 2, we introduce the data generative model and support algorithms for recovering community structure that will be used in our $\hat{\mathbf{Z}}$ -based algorithms for online change-point detection. In Section 3, we propose two $\hat{\mathbf{Z}}$ -based and one \hat{K} -based change-point detection algorithms for multilayer networks. In Section 4, we provide theoretical results for the estimators of change points. In Section 5, we present simulation studies to demonstrate the performance of the proposed algorithms and discuss their results. We provide full proofs for all of the

theoretical results discussed herein and relevant preliminary material in the Appendices A and B.

2 Background and Preliminaries

2.1. Network Sequence Data We consider that the data is given in the form of a sequence of $N \times N$ adjacency matrices $(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(t)}, \dots)$ corresponding to the sequence of networks $(G_N^{(1)}, G_N^{(2)}, \dots, G_N^{(t)}, \dots)$ having the same set of N vertices $V_N = \{v_1, \dots, v_N\}$ but with varying edge sets. $G_N^{(t)}$ is referred as the t -th network layer. Since we are considering the online change-point detection problem, we assume that the adjacency matrices are available to us sequentially. At any given point, T , of the sequence, the available set of adjacency matrices are $(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(T)})$.

For the purpose of this paper, we only consider undirected and unweighted graphs, that is, $A_{i,j}^{(t)} \in \{0, 1\}$ for all $i, j \in \{1, \dots, N\}$ and $t \in \mathbb{N}$ and $\mathbf{A}^{(t)} = (\mathbf{A}^{(t)})^T$. However, the conclusions of the paper can be extended to positively weighted graphs with non-random weights in a quite straightforward way by considering weighted adjacency matrices.

We consider that each network $G_N^{(t)}$ has an assortative community structure with $K^{(t)}$ ($K^{(t)} \in \mathbb{N}$) communities. Let us denote the $N \times K^{(t)}$ matrix $\mathbf{Z}^{(t)}$ to be the actual common community membership matrix of the nodes in each of the graphs $G_N^{(t)}$, where $\mathbf{Z}_{i,k}^{(t)} = 1$ if the i -th node belongs to the k -th community for all $G_N^{(t)}$ and zero otherwise.

2.2. Notations Let $[n] := \{1, 2, \dots, n\}$ for $n \in \mathbb{N}$ and $\mathcal{M}_{m,n}$ be the set of all $m \times n$ matrices which have exactly one 1 and $(n - 1)$ 0s in each row. $\mathbf{1}_m$ (resp. $\mathbf{0}_m$) denotes the vector in $\mathbb{R}^{m \times 1}$ consisting of all 1s (resp. 0s). \mathbf{I}_N is the $N \times N$ identity matrix. $\lambda_+(\mathbf{A})$ denotes the minimum positive eigenvalue, $\lambda_\ell^\downarrow(\mathbf{A})$ the ℓ -th largest eigenvalue, and $\lambda_\ell^\uparrow(\mathbf{A})$ the ℓ -th smallest eigenvalue of the matrix \mathbf{A} . $\text{Tr}(\mathbf{A})$ denotes the trace of the matrix \mathbf{A} . $\|\mathbf{A}\|_{p,q}$ denotes the $L_{p,q}$ norm of the matrix \mathbf{A} . $|[m]|$ denotes the cardinality of the set $[m]$. For a given matrix \mathbf{M} , let \mathbf{M}_{i*} denote the i -th row of \mathbf{M} and \mathbf{M}_{*j} its j -th column. $\mathbf{Z}_{[m]}$ denotes the sub-matrix of $(\mathbf{Z})_{n \times K}$ such that $(\mathbf{Z}_{[m]})_{i*} = (\mathbf{Z})_{[m]i*}$, where $i \in [m]$, $[m] \subset [n]$, $[m]_i$ is the index of the node i in $[n]$. $\mathbf{Z}_{[\bar{m}]}$ denotes the sub-matrix of $(\mathbf{Z})_{n \times K}$ such that $(\mathbf{Z}_{[\bar{m}]})_{j*} = (\mathbf{Z})_{[\bar{m}]j*}$, where $j \in [\bar{m}]$, $[\bar{m}] = [n]/[m]$, $[\bar{m}]_j$ is the index of the node j in $[n]$. \mathbf{P}_ω denotes the matrix form of a label permutation $\omega : \{1, \dots, K\} \rightarrow \{1, \dots, K\}$ on a clustering. For $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\langle \mathbf{A} \rangle$ denotes the matrix \mathbf{A} with its diagonal zeroed out: $\langle \mathbf{A} \rangle_{i,j} = A_{i,j}$ if $i \neq j$, $i, j \in [n]$ and $\langle \mathbf{A} \rangle_{i,i} = 0$ for $i \in [n]$. For any

subset S of layers, let $\mathbf{A}^S := \sum_{s \in S} \mathbf{A}^{(s)}$ denote the sum of the corresponding adjacency matrices. For some constant C , if for all $n \geq C$, denote

$$f(n) \in \begin{cases} O(g(n)) & \text{if } \limsup_{n \rightarrow \infty} \frac{f(n)}{g(n)} < \infty \\ o(g(n)) & \text{if } \limsup_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0 \\ \Omega(g(n)) & \text{if } \liminf_{n \rightarrow \infty} \frac{f(n)}{g(n)} > 0 \\ \omega(g(n)) & \text{if } \liminf_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \infty \\ \Theta(g(n)) & \text{if } \limsup_{n \rightarrow \infty} \frac{f(n)}{g(n)} = c \in (0, \infty). \end{cases}$$

2.3. Data Generative Models To set the context to investigate the theoretical properties of the change-point estimators, we consider a flexible data generative model that we call the *multilayer degree-corrected block model with one change-point* (MDBM). MDBM has six sets of parameters: (i) the change-point $\tau \in (1, T]$; (ii) the number of communities $K^{(t)}$, such that $K^{(t)} = K$ for $t \in [1, T]$ (i.e., the number of communities stays the same before and after τ); (iii) the $N \times 1$ membership vectors $\mathbf{z} = (z_1, \dots, z_N)$ for layers $t \in [1, \tau)$ and $\tilde{\mathbf{z}} = (\tilde{z}_1, \dots, \tilde{z}_N)$ for layers $t \in [\tau, T]$, where each $z_i, \tilde{z}_i \in \{1, \dots, K\}$; (iv) the $K \times K$ connectivity probability matrices $\mathbf{B} := (\mathbf{B}^{(t)} : t \in [1, \tau))$ and $\tilde{\mathbf{B}} := (\tilde{\mathbf{B}}^{(t)} : t \in [\tau, T])$; (v) the $N \times 1$ *degree parameters* vector $\boldsymbol{\psi} = (\psi_1, \dots, \psi_N)$; and (vi) the $K \times 1$ vector of probabilities of allocation for each community, $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)$ for layers $t \in [1, \tau)$ and $\tilde{\boldsymbol{\pi}} = (\tilde{\pi}_1, \dots, \tilde{\pi}_{\tilde{K}})$ for layers $t \in [\tau, T]$. For $i > j$, $i, j \in [N]$, and $t \in [1, \tau)$ MDBM with the six sets of parameters is given by

$$z_1, \dots, z_N \stackrel{iid}{\sim} \text{Mult}(1; (\pi_1, \dots, \pi_K)), \quad (2.1)$$

$$\mathbb{P}\left(A_{i,j}^{(t)} = 1 | z_i, z_j\right) = \psi_i \psi_j B_{z_i, z_j}^{(t)} \quad (2.2)$$

and for $i > j$, $i, j \in [N]$, and $t \in [\tau, T]$

$$\tilde{z}_1, \dots, \tilde{z}_N \stackrel{iid}{\sim} \text{Mult}(1; (\tilde{\pi}_1, \dots, \tilde{\pi}_K)), \quad (2.3)$$

$$\mathbb{P}\left(A_{i,j}^{(t)} = 1 | \tilde{z}_i, \tilde{z}_j\right) = \psi_i \psi_j \tilde{B}_{\tilde{z}_i, \tilde{z}_j}^{(t)}. \quad (2.4)$$

The inclusion of $\boldsymbol{\psi}$ entails the obvious issue of identifiability. In order to avoid this issue we assume as in Lei et al. (2015) that

$$\max_{i: z_i=k} \psi_i = 1 \text{ for all } k \in [K]. \quad (2.5)$$

Suppose $\mathbf{Z} \in \mathcal{M}_{N,K}$ and $\tilde{\mathbf{Z}} \in \mathcal{M}_{N,K}$ denote the actual membership matrices before and after the change-point, respectively. \mathbf{Z} and $\tilde{\mathbf{Z}}$ are unknown and we wish to estimate them along with τ . If for $i \in [N]$ the corresponding community index is $\mathbf{z}_i \in [K]$ (or, $\tilde{\mathbf{z}}_i \in [K]$), then clearly

$$\mathbf{Z}_{ij} = \mathbf{1}_{\{\mathbf{z}_i=j\}} \quad (\text{or, } \tilde{\mathbf{Z}}_{ij} = \mathbf{1}_{\{\tilde{\mathbf{z}}_i=j\}}).$$

In MDBM, each edge is drawn independently given the edge probability matrices $\mathbf{P}^{(t)} := \left(P_{ij}^{(t)} \right)_{i,j \in [N]}$ (or, $\tilde{\mathbf{P}}^{(t)} := \left(\tilde{P}_{ij}^{(t)} \right)_{i,j \in [N]}$). So, for $i > j$, $i, j \in [N]$, and $t \in [1, \tau)$

$$A_{i,j}^{(t)} \sim \text{Bernoulli} \left(P_{i,j}^{(t)} \right), \quad \text{where } \mathbf{P}^{(t)} := \mathfrak{D}(\boldsymbol{\psi}) \mathbf{Z} \mathbf{B}^{(t)} \mathbf{Z}^T \mathfrak{D}(\boldsymbol{\psi}), \quad (2.6)$$

and for $t \in [\tau, T]$

$$A_{i,j}^{(t)} \sim \text{Bernoulli} \left(\tilde{P}_{i,j}^{(t)} \right), \quad \text{where } \tilde{\mathbf{P}}^{(t)} := \mathfrak{D}(\boldsymbol{\psi}) \tilde{\mathbf{Z}} \tilde{\mathbf{B}}^{(t)} \tilde{\mathbf{Z}}^T \mathfrak{D}(\boldsymbol{\psi}), \quad (2.7)$$

where, $\mathfrak{D}(\boldsymbol{\psi}) = \text{Diag}(\boldsymbol{\psi})$.

We also consider a special case of MDBM, which we refer to as the *multilayer stochastic block model with one change-point* (MSBM). MSBM makes the simplifying assumption that $\psi_i = 1$ for all $i \in [N]$; however, the number of communities is allowed to change after the change-point. In other words, $K^{(t)} = K$ for $t \in [1, \tau)$ and $K^{(t)} = \tilde{K}$ for $t \in [\tau, T]$. While $K = \tilde{K}$ in MDBM, it is not necessarily the case in MSBM. Hence, $i > j$, $i, j \in [N]$, and $t \in [1, \tau)$ the parameter conditions of MSBM are as follows:

$$z_1, \dots, z_N \stackrel{iid}{\sim} \text{Mult}(1; (\pi_1, \dots, \pi_K)), \quad (2.8)$$

$$\mathbb{P} \left(A_{i,j}^{(t)} = 1 | z_i, z_j \right) = B_{z_i, z_j}^{(t)} \quad (2.9)$$

and for $i > j$, $i, j \in [N]$, and $t \in [\tau, T]$

$$\tilde{z}_1, \dots, \tilde{z}_N \stackrel{iid}{\sim} \text{Mult}(1; (\tilde{\pi}_1, \dots, \tilde{\pi}_{\tilde{K}})), \quad (2.10)$$

$$\mathbb{P} \left(A_{i,j}^{(t)} = 1 | \tilde{z}_i, \tilde{z}_j \right) = \tilde{B}_{\tilde{z}_i, \tilde{z}_j}^{(t)}. \quad (2.11)$$

As with MDBM, each edge is drawn independently as follows: For $i > j$, $i, j \in [N]$ and $t \in [1, \tau)$,

$$A_{i,j}^{(t)} \sim \text{Bernoulli} \left(P_{i,j}^{(t)} \right), \quad \text{where } \mathbf{P}^{(t)} := \mathbf{Z} \mathbf{B}^{(t)} \mathbf{Z}^T, \quad (2.12)$$

and for $t \in [\tau, T]$

$$A_{i,j}^{(t)} \sim \text{Bernoulli} \left(\tilde{P}_{i,j}^{(t)} \right), \text{ where } \tilde{\mathbf{P}}^{(t)} := \tilde{\mathbf{Z}} \tilde{\mathbf{B}}^{(t)} \tilde{\mathbf{Z}}^T. \quad (2.13)$$

2.4. Clustering Algorithms from Relevant Works First, we make precise the notion of clustering and define a simple measure of distance between two clusterings as follows.

Definition 1. ξ is a clustering on the set of nodes V_n if $\xi = \{\xi_1, \dots, \xi_K\}$ such that $\xi_i \cap \xi_j = \emptyset$ for $i \neq j$ and $\cup_{k=1}^K \xi_k = [N]$. ξ can be represented by the matrix $\mathbf{Z} \in \mathcal{M}_{N,K}$, where $\mathbf{Z}_{i*} = e_k$ (e_k is k -th unit vector) if node i is assigned to cluster k in ξ .

One simple way to measure the distance between clusterings is to count the number of mismatched elements between two clusterings.

Definition 2. Let ξ, ξ' be two clusterings on $[n]$, \mathbf{Z} and \mathbf{n}_K^ξ the matrix and the cluster size vector based on ξ (\mathbf{Z}' and $\mathbf{n}_K^{\xi'}$ resp. on ξ'). Let $[m]^{\xi, \xi'} \subset [n]$ be the set of node indices such that $\mathbf{Z}_{i*} = \mathbf{Z}_{j*} \Leftrightarrow \mathbf{Z}'_{i*} = \mathbf{Z}'_{j*}$ for all $i, j \in [m]^{\xi, \xi'}$. Denote the set of mismatched elements between two clusterings by $[\tilde{m}]^{\xi, \xi'} = [n] \setminus [m]^{\xi, \xi'}$.

Algorithms for recovery of community structures will be adapted from Bhattacharyya and Chatterjee (2020b) & Bhattacharyya and Chatterjee (2020a) and are reproduced below for reference. The first is based on the sum of adjacency matrices and the second on the sum of squares of adjacency matrices. We denote these two algorithms as ‘‘Clustering Algorithms 1 and 2’’ or ‘‘Algorithms 1 and 2’’ for brevity.

Before reciting the consistency results for Algorithms 1 and 2, we recall below definitions of pertinent parameters involving ψ , ξ and \mathbf{B} :

1. Measures of heterogeneity of ψ , for all $a \in [K]$:

- (a) $\tilde{N}_a := \sum_{i \in \xi_a} \psi_i^2$
- (b) $\tilde{N}_{\max} := \max_a \tilde{N}_a$ and $\tilde{N}_{\min} := \min_a \tilde{N}_a$
- (c) $\tilde{N}'_a := \sum_{i \in \xi_a \cap \{k_1, \dots, k_{N'}\}} \psi_i^2$,
- (d) $\tilde{N}'_{\max} := \max_a \tilde{N}'_a$ and $\tilde{N}'_{\min} := \min_a \tilde{N}'_a$
- (e) $\tau_a := \sum_{i \in \xi_a} \psi_i^2 \sum_{i \in \xi_a} \psi_i^{-2}$ is a parameter controlling the variation in heterogeneity within each community.

2. $\psi_{\min} := \min_{i \in [N]} \psi_i$
3. $\lambda_A := (T)^{-1} \sum_{t=1}^T \lambda_K \left(\frac{N}{d} \mathbf{B}^{(t)} \right)$, the minimum eigenvalue parameter based on sum of normalized connection probability matrices.
4. $\lambda_B := (T)^{-1} \sum_{t=1}^T \lambda_K \left(\left(\frac{N}{d} \mathbf{B}^{(t)} \right)^2 \right)$, the minimum eigenvalue parameter based on sum of squares of normalized connection probability matrices
5. $N_{\max} := \max(\mathbf{Z}^T \mathbf{1}_N)$ and $N_{\min} = \min(\mathbf{Z}^T \mathbf{1}_N)$ are the sizes of largest and smallest communities respectively.

Algorithm 1 Bhattacharyya and Chatterjee (2020a) Spherical spectral clustering of the sum of adjacency matrices.

Input: Adjacency matrices $\mathbf{A}^{(T_1)}, \dots, \mathbf{A}^{(T_2)}$; the number of communities K ; the approximation parameter ϵ ; the truncation parameter δ .

Output: Community membership matrix $\hat{\mathbf{Z}}_0$.

- 1: Obtain the sum of the adjacency matrices, $\mathbf{A}_0 := \sum_{t=1}^T \mathbf{A}^{(t)}$.
- 2: Get $\bar{d} := \frac{1}{NT} \mathbf{1}_N^T \mathbf{A}_0 \mathbf{1}_N$. Let N' be the number of rows (having indices $1 \leq k_1 < k_2 < \dots < k_{N'} \leq N$) of \mathbf{A}_0 having row sum at most $e(T\bar{d})^{1+\delta}$.
- 3: Let $\mathbf{A} \in \mathbb{R}^{N' \times N'}$ be the submatrix of \mathbf{A}_0 : $A_{i,j} = (A_0)_{k_i, k_j}$, $i, j \in [N']$.
- 4: Obtain $\hat{\mathbf{U}} \in \mathbb{R}^{N' \times K}$ consisting of the leading K eigenvectors of \mathbf{A} corresponding to its largest absolute eigenvalues.
- 5: Let N_+ be the number of nonzero rows of $\hat{\mathbf{U}}$ (having indices $1 \leq k'_1 < k'_2 < \dots < k'_{N_+} \leq N'$). Obtain $\hat{\mathbf{U}}^+ \in \mathbb{R}^{N_+ \times K}$ consisting of normalized nonzero rows of $\hat{\mathbf{U}}$, i.e. $\hat{\mathbf{U}}^+_{i,*} = \hat{\mathbf{U}}_{k'_i,*} / \left\| \hat{\mathbf{U}}_{k'_i,*} \right\|_2$ for $i \in [N_+]$.
- 6: Use $(1 + \epsilon)$ -approximate K -median clustering algorithm on the row vectors of $\hat{\mathbf{U}}^+$ to obtain $\hat{\mathbf{Z}}^+ \in \mathcal{M}_{N_+, K}$ and $\hat{\mathbf{X}} \in \mathbb{R}^{K \times K}$ satisfying

$$\left\| \hat{\mathbf{Z}}^+ \hat{\mathbf{X}} - \hat{\mathbf{U}}^+ \right\|_F \leq (1 + \epsilon) \min_{\Gamma \in \mathcal{M}_{N_+, K}, \mathbf{X} \in \mathbb{R}^{K \times K}} \left\| \Gamma \mathbf{X} - \hat{\mathbf{U}}^+ \right\|_F \quad (2.14)$$

- 7: Extend $\hat{\mathbf{Z}}^+$ to obtain $\hat{\mathbf{Z}} \in \mathcal{M}_{N', K}$ by (arbitrarily) adding $N' - N_+$ many canonical unit row vectors as follows. $(\hat{\mathbf{Z}})_{j,*} = \hat{\mathbf{Z}}^+_{i,*}$ (resp. $(1, 0, \dots, 0)$) for $j = k'_i$ (resp. $j \notin \{k'_1, k'_2, \dots, k'_{N_+}\}$)
 - 8: Extend $\hat{\mathbf{Z}}$ to obtain $\hat{\mathbf{Z}}_0 \in \mathcal{M}_{N, K}$ by (arbitrarily) adding $N - N'$ many canonical unit row vectors, like in Step 7.
 - 9: $\hat{\mathbf{Z}}_0$ is the estimate of \mathbf{Z} .
-

Algorithm 2 Bhattacharyya and Chatterjee (2020b) Spherical spectral clustering of the sum of squared adjacency matrices.

Input: Adjacency matrices $\mathbf{A}^{(T_1)}, \dots, \mathbf{A}^{(T_2)}$; the number of communities K ; the approximation parameter ϵ ; the truncation parameter δ .

Output: Community membership matrix $\hat{\mathbf{Z}}_0$.

- 1: Obtain the sum of the squared adjacency matrices, $\mathbf{A}_0^{[2]} := \sum_{t=1}^T (\mathbf{A}^{(t)})^2$, and zero out its diagonal to get $\langle \mathbf{A}_0^{[2]} \rangle$.
 - 2: Get $D_i^{[1]} := \max_{t=1}^T \sum_{j=1}^N A_{i,j}^{(t)}$ and $D_i^{[2]} := \max_{t=1}^T \sum_{j=1}^N \langle \mathbf{A}_0^{[2]} \rangle_{i,j}$ for $i \in [N]$.
 - 3: Get the order statistics $D_{(1)}^{[1]} \leq \dots \leq D_{(N)}^{[1]}$ and $D_{(1)}^{[2]} \leq \dots \leq D_{(N)}^{[2]}$.
 - 4: Get $\bar{d}^2 := \frac{1}{NT} \sum_{i=1}^N D_{(i)}^{[2]}$. Get $\Gamma_1 := \lceil N \exp(-\frac{1}{2}T^{1/2}[\bar{d}^2]^{3/4}) \rceil$ and $\Gamma_2 := \lceil N \exp(-\frac{1}{3}T[\bar{d}^2]^{1/2}) \rceil$.
 - 5: Get $\{i \in [N] : D_i^{[l]} \leq D_{(N+1-\Gamma_l)}^{[l]}\}$ for both $l = 1, 2$ and sort its entries in ascending order to have $1 \leq k_1 < \dots < k_{N'} \leq N$.
 - 6: Get submatrix $\mathbf{A}^{[2]} \in \mathbb{R}^{N' \times N'}$ of $\langle \mathbf{A}_0^{[2]} \rangle$, where $\mathbf{A}_{i,j}^{[2]} = \langle \mathbf{A}_0^{[2]} \rangle_{k_i, k_j}$, $i, j \in [N']$.
 - 7: Obtain $\hat{\mathbf{U}} \in \mathbb{R}^{N' \times K}$ consisting of the leading K eigenvectors of $\mathbf{A}^{[2]}$ corresponding to its largest eigenvalues.
 - 8: Let N_+ be the number of nonzero rows of $\hat{\mathbf{U}}$ (having indices $1 \leq k'_1 < k'_2 < \dots < k'_{N_+} \leq N'$). Obtain $\hat{\mathbf{U}}^+ \in \mathbb{R}^{N_+ \times K}$ consisting of normalized nonzero rows of $\hat{\mathbf{U}}$, i.e. $\hat{\mathbf{U}}_{i,*}^+ = \hat{\mathbf{U}}_{k'_i,*} / \left\| \hat{\mathbf{U}}_{k'_i,*} \right\|_2$ for $i \in [N_+]$.
 - 9: Use $(1 + \epsilon)$ -approximate K -median clustering algorithm on the row vectors of $\hat{\mathbf{U}}^+$ to obtain $\hat{\mathbf{Z}}^+ \in \mathcal{M}_{N_+, K}$ and $\hat{\mathbf{X}} \in \mathbb{R}^{K \times K}$ satisfying Eq. 2.14.
 - 10: Extend $\hat{\mathbf{Z}}^+$ to obtain $\hat{\mathbf{Z}} \in \mathcal{M}_{N', K}$ by (arbitrarily) adding $N' - N_+$ many canonical unit row vectors as follows. $(\hat{\mathbf{Z}})_{j,*} = \hat{\mathbf{Z}}_{i,*}^+$ (resp. $(1, 0, \dots, 0)$) for $j = k'_i$ (resp. $j \notin \{k'_1, k'_2, \dots, k'_{N_+}\}$)
 - 11: Extend $\hat{\mathbf{Z}}$ to obtain $\hat{\mathbf{Z}}_0 \in \mathcal{M}_{N, K}$ by (arbitrarily) adding $N - N'$ many canonical unit row vectors, like in Step 10.
 - 12: $\hat{\mathbf{Z}}_0$ is the estimate of \mathbf{Z} .
-

In addition, we let $\hat{\xi}$ and $\tilde{\xi}$ denote the estimated community labels using Algorithms 1 and 2, respectively. The theoretical results for Algorithms 1 and 2 reproduced in Theorems 1 and 2 and are based on the following conditions:

(A) The number of communities, K , is constant.

(B) Community sizes are balanced, i.e., $N_{\max}/N_{\min} = O(1)$.

(C) $\psi_i = \alpha_i / \max\{\alpha_j : z_i = z_j\}$, where $(\alpha_i)_{i=1}^N$ are i.i.d. positive weights such that $\mathbb{E} [\max\{\alpha_1^2, \alpha_1^{-2}\}] < \infty$.

(D) $N \geq 3K$.

(E) $\lambda_B \left(\frac{\tilde{N}_{\min}}{N}\right)^2 > \frac{7}{N}$.

THEOREM 1. (Bhattacharyya and Chatterjee, 2020a) *Under Conditions (A) through (D) above, for any $\epsilon, \eta, \delta > 0$ and $c \in (0, 1)$, there are constants $C_1 > 0$ as function of ϵ, c, δ and $C_2 > 0$ as function of c, δ , such that if $Td \geq C_2(K/\lambda_A)^{1+\frac{1}{\delta}}$, then*

$$|[\bar{m}]^{\xi, \hat{\xi}}| \leq \bar{N}_{pr} + C_1 \left[\frac{K \tilde{N}'_{\max}}{(\psi_{\min} \lambda_A \tilde{N}'_{\min})^2} + \frac{N(Td)^{-0.5+\delta+\eta} (K \sum_{k \in [K]} \tau_k)^{1/2}}{\lambda_A \tilde{N}'_{\min}} \right] \quad (2.15)$$

with probability at least $1 - o(1)$ as $Td\lambda_A \rightarrow \infty$, where $\bar{N}_{pr} := \frac{N}{e^{(1-c)Td}}$ is the upper bound on the number of pruned nodes in Step 2 of Algorithm 1.

THEOREM 2. (Bhattacharyya and Chatterjee, 2020b) *Under Conditions (A), (B), (C), and (E), for any $\epsilon > 0$ and $\Delta > 8$, there are constants $C > 0$ as a function of ϵ and $C' > 0$ such that if*

$$N_{\min} > \frac{C(K\tilde{N}_{\max})^3 \lambda_B^{-2}}{\psi_{\min}^2 \tilde{N}_{\min}^4} + \frac{C\Delta(K \sum_{a \in K} \tau_a)^{1/2}}{(Td)^{1/4} \lambda_B (\tilde{N}_{\min}/N)^2}$$

then

$$|[\bar{m}]^{\xi, \tilde{\xi}}| \leq \bar{N}_{pr} + C \left[\frac{(K\tilde{N}_{\max})^3}{(\psi_{\min} \lambda_B)^2 (\tilde{N}_{\min})^4} + \frac{\Delta(K \sum_{k \in [K]} \tau_k)^{1/2}}{(Td)^{1/4} \lambda_B (\tilde{N}_{\min}/N)^2} \right] \quad (2.16)$$

with probability at least $1 - o(1)$ as $(Td)^{1/4} \lambda_B \rightarrow \infty$, where the upper bound on the number of pruned nodes in Step 5 of Algorithm 2 is $\bar{N}_{pr} := \frac{N}{(Td)^{1/4} \lambda_B (\tilde{N}_{\min}/N)^2}$.

2.5. Background on Information Theory We define preliminary terms from information theory and conclude with a measure of the distance between clusterings.

Definition 3. Let $\mathbf{n}_K^\xi = \mathbf{Z}^T \mathbf{1} = (n_1^\xi, \dots, n_K^\xi)^T$ be the vector of cluster sizes of clustering ξ . Then the entropy of ξ is defined as:

$$H(\xi) := - \sum_{k=1}^K \frac{n_k^\xi}{N} \log \left(\frac{n_k^\xi}{N} \right). \quad (2.17)$$

Definition 4. Let ξ, ξ' be clusterings on $[N]$, \mathbf{Z} and \mathbf{n}_K^ξ be defined as in Definitions 1 and 3 for ξ , and \mathbf{Z}' and $\mathbf{n}_K^{\xi'}$ for ξ' . Let $\mathbf{n}^{\xi, \xi'} = \mathbf{Z}^T \mathbf{Z}' = \left[n_{ij}^{\xi, \xi'} \right]_{K \times K}$ be the contingency table of ξ and ξ' , where $n_{ij}^{\xi, \xi'}$ is the number of nodes in both ξ_i and ξ'_j for $i, j \in [K]$. The mutual information between ξ and ξ' is defined as:

$$I(\xi, \xi') := \sum_{i=1}^K \sum_{j=1}^K \frac{n_{ij}^{\xi, \xi'}}{N} \log \left(N \frac{n_{ij}^{\xi, \xi'}}{n_i^\xi n_j^{\xi'}} \right). \quad (2.18)$$

The mutual information between two clusterings measures the information one clustering has about the other. The normalized mutual information has been widely used in the literature as an information-theoretic measure of similarity between clustering. Here, we use a measure of dissimilarity between clusterings called *variation of information* (\mathcal{VI}), which is based on entropy and mutual information.

Definition 5. Let $\xi, \xi', \mathbf{Z}, \mathbf{n}_K^\xi, \mathbf{Z}'$ and $\mathbf{n}_K^{\xi'}$ be defined as in Definition 4. The *variation of information* is defined as:

$$\mathcal{VI}(\xi, \xi') := H(\xi) + H(\xi') - 2I(\xi, \xi'). \quad (2.19)$$

Properties of \mathcal{VI} are discussed in Meilă (2007) and its exhaustive list is given in Appendix A.1. The most relevant property for our purposes is as follows:

PROPERTY 1. (Meilă, 2007) \mathcal{VI} is a metric.

Note that as a metric, \mathcal{VI} satisfies the following three axioms: positive definiteness, symmetry, and the triangle inequality.

2.6. The Bethe Hessian Matrix Recently, a certain class of graph operators has received an increasing level of attention for their spectral properties that allow for a simple and efficient recovery of community structure. While some algorithms, such as those based on belief propagation, require a generative model with correct parameters as inputs, and those based on adjacency matrices and graph Laplacians are ineffective in the presence of degree fluctuations, non-parametric spectral clustering methods based on the non-backtracking and the Bethe Hessian matrices are robust to such challenges. As a result, they have become popular, reliable tools of choice when faced with network data that are sparse and heterogeneous (Krzakala et al., 2013; Saade et al., 2014a; Bordenave et al., 2015; Coste and Zhu, 2019; Gulikers et al., 2016; Bruna and Li, 2017; Saade et al., 2014b; Watanabe and

Fukumizu, 2009; Dall’Amico and Couillet, 2019; Dall’Amico et al., 2020; Le and Levina, 2015). We start by defining these matrices on which some of the robust non-parametric methods are based.

2.6.1. Definitions.

Definition 6. *The non-backtracking operator, \mathbf{B} , is a $2|E| \times 2|E|$ matrix indexed by directed edges $i \rightarrow j$ and defined $\mathbf{B}_{i \rightarrow j, k \rightarrow l} = \delta_{jk}(1 - \delta_{il})$, where $|E|$ is the number of edges in the adjacency matrix \mathbf{A} , i, j, k, l are vertices of \mathbf{A} and δ is the Kronecker delta.*

Definition 7. *The Bethe Hessian matrix is defined as $\mathbf{H}_r^S := (r^2 - 1)|S|\mathbf{I}_N + \mathbf{D}^S - r\mathbf{A}^S$ based on \mathbf{A}^S with parameter r , where $\mathbf{D}_{N \times N}^S = \text{Diag}(d_1^S, \dots, d_N^S)$ and $d_i^S := \sum_{j \in [N]} \mathbf{A}_{ij}^S$ for $i \in [N]$. When $|S| = 1$ and it is clear from context which \mathbf{A} is used, we write $\mathbf{H}_r := (r^2 - 1)\mathbf{I}_N + \mathbf{D} - r\mathbf{A}$ with $\mathbf{D} := \text{Diag}(\mathbf{A}\mathbf{1}_N)$.*

Definition 8. *The moving window sum is defined as $\mathbf{A}^{[t_0, t]} := \sum_{s \in [t_0, t]} \mathbf{A}^{(s)}$ where $t_0 = \max\{1, t - \theta + 1\}$ and the window $\theta \geq 1, \theta \in \mathbb{N}$.*

2.6.2. *Background.* It was noted in Krzakala et al. (2013) that for networks containing K communities, while high degree variations suppress signal from informative eigenvalues of \mathbf{A} and the Laplacian, the K largest eigenvalues of \mathbf{B} are real-valued and well-separated from the circle of radius $\|\mathbf{B}\|^{1/2}$ where the bulk of the eigenvalues of \mathbf{B} is contained. Further, Bordnave et al. (2015) showed that the largest K eigenvalues of \mathbf{B} concentrate around the informative eigenvalues of $\mathbb{E}[\mathbf{A}]$.

However, \mathbf{B} are non-symmetric, preventing the use of linear algebraic tools for symmetric matrices, and can be much larger than the adjacency matrices, presenting computational challenges. In response, Saade et al. (2014a) gave an intuitive argument rooted in statistical physics and numerical simulations to demonstrate that \mathbf{H}_r is simpler given its symmetric nature and is at least as effective as \mathbf{B} at detecting communities, while offering significant gains in computational efficiency. In statistical physics, \mathbf{H}_r is an approximation of the Hessian matrix of the Bethe free energy at the system equilibria corresponding to the trivial fixed-points of the belief propagation equations, where every vertex belongs to each community with equal probability. Here, the parameter r denotes the temperature of the system (Yedidia et al., 2003). Then, the informative eigenvalues of \mathbf{H}_r are the phase transitions in the Ising Hamiltonian model where new communities become visible (Bruna and Li, 2017).

An important property of \mathbf{H}_r that bears connection to \mathbf{B} is that when r is set to an eigenvalue of the latter, the determinant of \mathbf{H}_r vanishes (Hashimoto, 1989; Angel et al., 2015).

On the topic of what specific values should be used for the parameter r , several proposals have been put forth. Krzakala et al. (2013) proposed $\|\mathbf{B}\|^{1/2}$, i.e., the radius of the circle containing the bulk, which is approximated by r_d below, while Saade et al. (2014a) proposed a square root of the mean degree r_a for SBM networks:

$$r_d^2 = \left(\sum_{i=1}^N d_i \right)^{-1} \left(\sum_{i=1}^N d_i^2 \right) - 1 \quad r_a = \sqrt{\frac{\sum_{i=1}^N d_i}{N}}$$

where d_i denotes the degree of node i . Dall’Amico et al. (2019) presented an intuitive argument that for 2-community degree-corrected block model networks, $r = \frac{a+b}{a-b}$ is insensitive to degree heterogeneity and showed through simulations that it outperforms certain other choices for r in such networks. On the topic of estimating the number of communities K , Le and Levina (2015) proposed counting the number of negative eigenvalues in \mathbf{H}_r , with the parameter choices r_d based on Krzakala et al. (2013) and r_a based on Saade et al. (2014a). Noting that simulations using r_d and r_a both underestimated K when network is unbalanced, Le and Levina (2015) suggested an additional method of estimating K whereby the number of eigenvalues that are separated from the bulk by a predetermined integer multiple of the bulk radius is used instead.

Despite the rich literature, however, a formal proof of the intuition that the number of communities in a network is given by the number of negative eigenvalues of \mathbf{H}_r , and precisely what value of r enables such a detection and under what conditions, is still lacking.

3. Online Change-point Detection Methods

We propose the following two types of algorithms for online change-point detection. In both cases, we compute an estimate of the structural property of interest using aggregated networks and infer that a change-point has occurred when we detect a change in the estimates.

1. $\hat{\mathbf{Z}}$ -based algorithms detect a change using the estimated community structure.
2. \hat{K} -based algorithm detects a change using the estimated number of communities.

These algorithms are discussed in detail below in Sections 3.1 and 3.2, followed by theoretical results in Section 4.

3.1. $\hat{\mathbf{Z}}$ -based Change-Point Detection Algorithms The community structure of a “recent” (to be made precise later) sequence of networks is estimated, and is compared to the estimated structure of an “old” sequence using \mathcal{VI} . We denote an estimated community structure by $\hat{\mathbf{Z}}$. In constructing the two types of aggregated networks to compare, we employ two approaches. In one, we let the two sequences have a common set of layers; in the other, we keep the two sequences disjoint. These two approaches are encoded in Algorithms 3 and 4 discussed later in Section 3.1.2.

In the absence of a change-point, the latent community structures of a recent and an old sequence of networks are expected to be similar. In other words, the \mathcal{VI} between the two sequences should be small by Property 1. While latent community structures are unobserved, Theorem 4 below guarantees that as long as the number of mismatched nodes is bounded, the \mathcal{VI} between the estimated and the latent structures is also bounded. This forms the basis for using estimated community structures in Algorithms 1 and 2.

3.1.1. Upper Bound on $\mathcal{VI}(\hat{\mathbf{Z}}, \mathbf{Z})$. Recall that Theorems 1 and 2 provide upper bounds on the number of misclassified nodes of estimated community structures computed using Algorithms 1 and 2, respectively. Theorems 3 and 4 below state that this bound directly implies an upper bound on $\mathcal{VI}(\hat{\mathbf{Z}}, \mathbf{Z})$. The proofs of Theorems 3 and 4 are given in Appendices A.2 and A.3.

THEOREM 3. *Given two clusterings ξ and ξ' (alternatively, \mathbf{Z} and \mathbf{Z}' in a matrix form), let $[m]^{\xi, \xi'} \subset [N]$ be the set of node indices such that $\mathbf{Z}_{i*} = \mathbf{Z}_{j*} \Leftrightarrow \mathbf{Z}'_{i*} = \mathbf{Z}'_{j*}$ for all $i, j \in [m]^{\xi, \xi'}$. Then there exists a $K \times K$ permutation matrix \mathbf{P}_ω such that*

$$|[m]^{\xi, \xi'}| = \text{Tr}(\mathbf{Z}^T \mathbf{Z}'_\omega) = \text{Tr}(\mathbf{Z}^T \mathbf{Z}' \mathbf{P}_\omega), \text{ where } \mathbf{Z}'_\omega = \mathbf{Z}' \mathbf{P}_\omega. \quad (3.1)$$

THEOREM 4. *Let ξ, ξ' be two clusterings. Let \mathbf{P}_ω be the permutation matrix satisfying Eq. 3.1, and denote the contingency table by $\mathbf{Z}^T \mathbf{Z}'_\omega = [n_{ij}]$. Let n_1, \dots, n_K be the cluster sizes with ξ and n'_1, \dots, n'_K of ξ' , respectively. Without loss of generality, assume $n_1 < \dots < n_K$ and $n'_1 < \dots < n'_K$. Then $\mathcal{VI}(\xi, \xi')$ given $|[m]^{\xi, \xi'}|$ is upper bounded by $M(\xi, \xi', |[m]^{\xi, \xi'}|)$, which is defined as:*

$$\begin{aligned} M\left(\xi, \xi', |[m]^{\xi, \xi'}|\right) = & M_{\xi, |[m]^{\xi, \xi'}|}^{(a)} + M_{\xi, |[m]^{\xi, \xi'}|}^{(b)} + M_{\xi, |[m]^{\xi, \xi'}|}^{(c)} + M_{\xi, |[m]^{\xi, \xi'}|}^{(d)} \\ & + M_{\xi', |[m]^{\xi, \xi'}|}^{(a)} + M_{\xi', |[m]^{\xi, \xi'}|}^{(b)} + M_{\xi', |[m]^{\xi, \xi'}|}^{(c)} + M_{\xi', |[m]^{\xi, \xi'}|}^{(d)} \end{aligned} \quad (3.2)$$

where,

$$M_{\xi, |[m]^{\xi, \xi'}|}^{(a)} = \sum_{i=0}^{K^{(a)}} \frac{n_i}{N} \log \frac{N}{n_i} + \frac{n^{(a)}}{N} \left[\log(K - K^{(a)}) + \log \frac{N}{n^{(a)}} \right], \text{ while } n_0 = 0 \quad (3.3)$$

$$K^{(a)} = \min\{j | j \in 0, \dots, K-1\}, \text{ such that } |[m]^{\xi, \xi'}| - \sum_{i=0}^j n_i \leq (K-j)n_{j+1}$$

$$n^{(a)} = |[m]^{\xi, \xi'}| - \sum_{i=0}^{K^{(a)}} n_i$$

$$M_{\xi, |[m]^{\xi, \xi'}|}^{(b)} = - \left[\frac{K^{(b)} - 1}{N} \log N + \frac{n^{(b)}}{N} \log \frac{N}{n^{(b)}} + \sum_{i=K^{(b)+1}^{K+1} \frac{n_i}{N} \log \frac{N}{n_i} \right],$$

while $n_{K+1} = 0$ (3.4)

$$K^{(b)} = \max\{j | j \in 1, \dots, K\}, \text{ such that } |[m]^{\xi, \xi'}| - (j-1) - \sum_{i=j+1}^{K+1} n_i \leq n_j$$

$$n^{(b)} = |[m]^{\xi, \xi'}| - (K^{(b)} - 1) - \sum_{i=K^{(b)+1}^{K+1} n_i$$

$$M_{\xi, |[m]^{\xi, \xi'}|}^{(c)} = \sum_{i=0}^{K^{(c)}} \frac{n_i - 1}{N} \left[\log(K-1) + \log \frac{N}{n_i - 1} \right] + \frac{n^{(c)}}{N} \left[\log(K - K^{(c)})(K-1) + \log \frac{N}{n^{(c)}} \right], \quad (3.5)$$

while $n_0 = 1$

$$K^{(c)} = \min\{j | j \in 0, \dots, K-1\}, \text{ such that } N - |[m]^{\xi, \xi'}| - \sum_{i=0}^j (n_i - 1)$$

$$\leq (K-j)(n_{j+1} - 1)$$

$$n^{(c)} = N - |[m]^{\xi, \xi'}| - \sum_{i=0}^{K^{(c)}} (n_i - 1)$$

$$M_{\xi, |[m]^{\xi, \xi'}|}^{(d)} = - \left[\frac{n^{(d)}}{N} \log \frac{N}{n^{(d)}} + \sum_{i=K^{(d)+1}^{K+1} \frac{n_i - 1}{N} \log \frac{N}{n_i - 1} \right],$$

while $n_0 = n_{K+1} = 1$ (3.6)

$$K^{(d)} = \max\{j | j \in 1, \dots, K\}, \text{ s.t. } N - |[m]^{\xi, \xi'}| - \sum_{i=j+1}^{K+1} (n_i - 1) \leq n_j - 1$$

$$n^{(d)} = N - |[m]^{\xi, \xi'}| - \sum_{i=K^{(d)}+1}^{K+1} (n_i - 1)$$

$M_{\xi', |[m]^{\xi, \xi'}|}^{(a)}, \dots, M_{\xi', |[m]^{\xi, \xi'}|}^{(d)}$ have bounds of the same form as $M_{\xi, |[m]^{\xi, \xi'}|}^{(a)}, \dots, M_{\xi, |[m]^{\xi, \xi'}|}^{(d)}$, with n'_i replacing n_i .

3.1.2. Two Algorithms. In what follows, we formally introduce the two algorithms based on $\hat{\mathbf{Z}}$ below as Algorithms 3 and 4. Given a sequence of adjacency matrices $\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(T)}$, the goal is to detect a change-point τ based on \mathcal{VI} between recovered community structures for layers $\{1, \dots, \tau - 1\}$ and $\{\tau, \dots, T\}$. This is done in two steps. First, for each layer t , the \mathcal{VI} between two sequences of layers differing by a segment containing t is checked whether it is larger than the upper bound in Theorem 4. If so, it is inferred that $t \geq \tau$. In addition, if for some $v_J, v_C \in \mathbb{N}$, the \mathcal{VI} at $t + v_J$ through $t + v_J + v_C - 1$ progressively becomes larger, then it is concluded that t is the estimated change-point.

In addition to the adjacency matrices, inputs include the following: cushion κ , which denotes some sufficient number of layers needed for Algorithms 1 and 2 to recover community structure; the number of communities K ; and window θ , which denotes the most recent number of layers representing the difference in lengths of any two pairs of sequences of networks being compared.

Algorithms 3 and 4 differ only on the sequences of networks used for comparison. In Algorithm 3, the recent sequence consists of layers $\max\{1, t - m\theta + 1\}$ through t , while the old sequence consists of $\max\{1, t - m\theta + 1\}$ through $t - \theta$, where m denotes the number of windows and controls the proportion of old layers in the recent sequence. Hence the two sequences share at most $(m - 1)\theta$ layers. In Algorithm 4, the recent sequence consists of layers $t - \theta + 1$ through t , while the old sequence is comprised of layers 1 through $t - \theta$ and is thus disjoint from the recent sequence.

In lines 3 to 7 in Algorithm 3 and 3 to 6 in Algorithm 4, the community structure at each t is estimated and \mathcal{VI} between recent and old sequences are computed at layers $t - 1$ through $t + v_J + v_C - 1$. In lines 9 to 11 in Algorithm 3 and lines 8 to 10 in Algorithm 4, \mathcal{VI} are checked against the upper bound from Theorem 4 and whether they progressively increase to determine if t is a change-point.

Algorithm 3 $\hat{\mathbf{Z}}$ -based CHANGE-POINT DETECTION.

Input: Adjacency matrices $\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(T)}$; the number of communities K ; cushion κ ; window $\theta \in \mathbb{N}$; the number of windows $m \in \mathbb{N}$; jump length $v_J \in \mathbb{N}$; checking length $v_C \in \mathbb{N}$; Algorithm 1 or 2.

Output: layer $\hat{\tau}$ where a change-point is detected; \emptyset if no change-point is detected.

```

1:  $t \leftarrow \kappa + \theta$ 
2: while  $t \leq (T - v_J - v_C + 1)$  do
3:   for  $r := -1$  to  $(v_J + v_C - 1)$  do
4:      $t_0 \leftarrow \max\{1, t + r - m * \theta + 1\}$ 
5:      $\hat{\xi}_{t+r} \leftarrow \text{Algorithm 1}(\mathbf{A}^{(t_0)}, \dots, \mathbf{A}^{(t+r)}, K)$   $\triangleright$  Algorithm 2 could
       also be used
6:      $\hat{\xi}_{t+r-\theta} \leftarrow \text{Algorithm 1}(\mathbf{A}^{(t_0)}, \dots, \mathbf{A}^{(t+r-\theta)}, K)$ 
7:      $\mathcal{VI}_{t+r} \leftarrow \mathcal{VI}(\hat{\xi}_{t+r}, \hat{\xi}_{t+r-\theta})$   $\triangleright$  By Eq. 2.19
8:      $l \leftarrow 0$ 
9:     for  $s := 0$  to  $v_C - 1$  do
10:       $l \leftarrow l + \mathbf{1}(\mathcal{VI}_{t+v_J+s} - \mathcal{VI}_{t-1} > 0)$   $\triangleright \mathbf{1}(\cdot)$  is the indicator
        function
11:      if  $\mathcal{VI}_t > M \left( \xi_{t-\theta}, \hat{\xi}_t, N - |[\bar{m}]_{t_0:t}| \right) + M \left( \xi_{t-\theta}, \hat{\xi}_{t-\theta}, N - |$ 
         $[\bar{m}]_{t_0:(t-\theta)}| \right)$  and  $l = v_C$  then
12:        return  $\{\hat{\tau} \leftarrow t\}$ 
13:      return  $\emptyset$ 

```

REMARK 1. Calculating \mathcal{VI} in line 9 of Algorithm 3 involves computing upper bounds on $|[\bar{m}]_{t_1:t_2}|$ using either (2.15) or (2.16) depending on whether Algorithm 1 or 2 is used. However, since ξ is unknown, letting N_{pr} be the number of pruned high-degree nodes, we can calculate an upper bound on $|[\bar{m}]_{t_1:t_2}|$ using Eqs. 3.7 and 3.8 below (for Algorithms 1 and 2, respectively):

$$|[\bar{m}]_{t_1:t_2}| = N_{pr} + C \left[\frac{(K \tilde{N}_{\max})^3}{(\psi_{\min} \lambda_B)^2 (\tilde{N}_{\min})^4} + \frac{\Delta (K \sum_{k \in [K]} \tau_k)^{1/2}}{((t_2 - t_1)d)^{1/4} \lambda_B (\tilde{N}_{\min}/N)^2} \right] \quad (3.7)$$

$$|[\bar{m}]_{t_1:t_2}| = N_{pr} + C_1 \left[\frac{K \tilde{N}'_{\max}}{(\psi_{\min} \lambda_A \tilde{N}'_{\min})^2} + \frac{N((t_2 - t_1)d)^{-0.5 + \delta + \eta} (K \sum_{k \in [K]} \tau_k)^{1/2}}{\lambda_A \tilde{N}'_{\min}} \right] \quad (3.8)$$

where C, Δ, C_1, δ and η are constants.

Algorithm 4 $\hat{\mathbf{Z}}$ -based CHANGE-POINT DETECTION.

Input: Adjacency matrices $\mathbf{A}^{(T_1)}, \dots, \mathbf{A}^{(T_2)}$; the number of communities K ; cushion κ ; window $\theta \in \mathbb{N} \setminus \{1\}$; jump length $v_J \in \{0, \dots, \theta - 1\}$; checking length $v_C \in \mathbb{N}$; Algorithm 1 or 2.

Output: layer $\hat{\tau}$ where a change-point is detected; \emptyset if no change-point is detected.

```

1:  $t \leftarrow \kappa + \theta$ 
2: while  $t \leq (T - v_J - v_C + 1)$  do
3:   for  $r := -1$  to  $(v_J + v_C - 1)$  do
4:      $\tilde{\xi}_{t+r} \leftarrow$  Algorithm 1( $\mathbf{A}^{(t+r-\theta+1)}, \dots, \mathbf{A}^{(t+r)}, K$ )  $\triangleright$  Algorithm 2
       could also be used
5:      $\hat{\xi}_{t+r-\theta} \leftarrow$  Algorithm 1( $\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(t+r-\theta)}, K$ )
6:      $\mathcal{VI}_{t+r} \leftarrow \mathcal{VI}(\tilde{\xi}_{t+r}, \hat{\xi}_{t+r-\theta})$   $\triangleright$  By Eq. 2.19
7:      $l \leftarrow 0$ 
8:     for  $s := 0$  to  $v_C - 1$  do
9:        $l \leftarrow l + \mathbf{1}(\mathcal{VI}_{t+v_J+s} - \mathcal{VI}_{t-1} > 0) \triangleright \mathbf{1}(\cdot)$  is the indicator function
10:    if  $\mathcal{VI}_t > M \left( \xi_{t-\theta}, \tilde{\xi}_t, N - |[\bar{m}]_{(t-\theta+1):t}| \right) + M \left( \xi_{t-\theta}, \hat{\xi}_{1:(t-\theta)}, \right.$ 
        $\left. N - |[\bar{m}]_{1:(t-\theta)}| \right)$  and  $l = v_C$  then
11:      return  $\{\hat{\tau} \leftarrow t\}$ 
12:    return  $\emptyset$ 

```

If no change-point is detected by layer t , the layers from $\max\{1, t - m * \theta + 1\}$ to $t - 1$ are assumed to have the same latent community structure and $\hat{\xi}_{t-\theta}$ given by $\hat{\mathbf{Z}}$ is considered an estimate of $\xi_{t-\theta}$. For $\boldsymbol{\pi}$, $\boldsymbol{\psi}$ and \mathbf{B} needed in the calculation of $|[\bar{m}]_{t_1:t_2}|$, we use adjusted profile likelihood estimates from layers $t_1 : t_2 - \theta$. We derive the MDBM parameter estimates as follows:

$$\hat{\pi}_a = \frac{n_a}{N}, \quad \hat{\psi}_i = \frac{\sum_{t=t_1}^{t_2-\theta} d_i^{(t)}}{T}, \quad \hat{B}_{a,b} = \frac{n_{ab}}{n_a n_b},$$

where $d_i^{(t)}$ denotes the degree of node $i \in [N]$ in layer $t \in [t_1, t_2 - \theta]$, $T := t_2 - t_1 - \theta$, n_a denotes the number of nodes with community label a , and $n_{ab}^{(t)}$ denotes number of edges between communities with labels a and b in layer t , for $a, b \in [K]$. To satisfy the assumptions that (1) the connectivity probability matrix does not change with time, and that (2) the local maximal degree equals 1, we normalize and adjust the estimates as follows:

$$\hat{\psi}'_i = \frac{\hat{\psi}_i}{\max_{j: \mathbf{Z}_{j^*} = \mathbf{Z}_{i^*}} \hat{\psi}_j}, \quad \hat{B}'_{a,b} = \frac{\sum_t n_{ab}^{(t)}}{(t - \theta) n_a n_b} \max_{i: \mathbf{Z}_{i^*} = \mathbf{e}_a} \hat{\psi}_i \max_{j: \mathbf{Z}_{j^*} = \mathbf{e}_b} \hat{\psi}_j.$$

Algorithm 5 \hat{K} -based CHANGE-POINT DETECTION ALGORITHM.

Input: Adjacency matrices $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(T)}$, a window $\theta \in [2, 4]$, a cushion κ , and a sample size $M \ll N$.

Output: $\hat{\tau} \in [T]$ if it is a change-point; \emptyset otherwise

Procedure to initialize estimates for r and K

- 1: **procedure** ESTIMATEPARAMS($\mathbf{A}^S, |S|$)
 - 2: Compute the eigenvalues of $\mathbf{A}^S, \{\lambda_i\}_{i=1}^N$
 - 3: Sort in non-increasing order $\lambda_1(\mathbf{A}^S) \geq \dots \geq \lambda_N(\mathbf{A}^S)$
 - 4: $\Delta\lambda_i(\mathbf{A}^S) \leftarrow \lambda_i(\mathbf{A}^S) - \lambda_{(i+1)}(\mathbf{A}^S)$ for $1 \leq i \leq M$ ▷ See Remark 5
 - 5: $\bar{x}_\Delta \leftarrow \frac{1}{M} \sum_{i \in [M]} \Delta\lambda_i(\mathbf{A}^S)$
 - 6: $sd_\Delta \leftarrow \left(\sum_{i \in [M]} \frac{(\Delta\lambda_i(\mathbf{A}^S) - \bar{x}_\Delta)^2}{M-1} \right)^{0.5}$
 - 7: $\hat{k}_0 \leftarrow |\{i | \Delta\lambda_i(\mathbf{A}^S) > \bar{x}_\Delta + sd_\Delta\}|$
 - 8: $\bar{d}^{|S|} \leftarrow \frac{1}{N^{|S|}} \mathbf{1}_N^T \mathbf{A}^S \mathbf{1}_N$
 - 9: $\gamma_{\bar{d}^{|S|}} \leftarrow \min\{1, (0.7 + 0.0125\bar{d}^{|S|})\}$ ▷ See Remark 7
 - 10: $\hat{r}_0 \leftarrow \lambda_{\hat{k}_0}(\mathbf{A}^S) \cdot \frac{\gamma_{\bar{d}^{|S|}}}{|S|}$
 - 11: **return** (\hat{k}_0, \hat{r}_0)
-

Main procedure to detect change-point

- 1: **procedure** CHANGEPOINTDETECTION($\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(T)}, \theta, \kappa$)
 - 2: $c \leftarrow \min\{\kappa, \theta\}$
 - 3: $\mathbf{A}^{[c]} \leftarrow \sum_{s \in [c]} \mathbf{A}^{(s)}$
 - 4: $\hat{k}_0, \hat{r}_0 \leftarrow \text{EstimateParams}(\mathbf{A}^{[c]}, c)$
 - 5: $t \leftarrow c + 1$
 - 6: **while** $t \leq T$ **do**
 - 7: $t_0 \leftarrow \max\{1, t - \theta + 1\}$
 - 8: $\mathbf{A}^{[t_0, t]} \leftarrow \sum_{s \in [t_0, t]} \mathbf{A}^{(s)}$
 - 9: $\mathbf{H}^{[t_0, t]}(\hat{r}_0) \leftarrow (\hat{r}_0^2 - 1)\mathbf{I} - \hat{r}_0 \mathbf{A}^{[t_0, t]} + \mathbf{D}^{[t_0, t]}$
 - 10: Compute the eigenvalues of $\mathbf{H}^{[t_0, t]}(\hat{r})$, $\{\lambda_i\}_{i=1}^N$
 - 11: Sort in non-decreasing order, $\lambda_1 \leq \dots \leq \lambda_N$
 - 12: $\hat{k} \leftarrow \max\{k | \lambda_k < 0\}$
 - 13: **if** $\hat{k} < \hat{k}_0$ **then**
 - 14: **return** $\hat{\tau} = t$
 - 15: **else**
 - 16: **if** ($c < \theta$) **then**
 - 17: $\mathbf{A}^{[c+1]} \leftarrow \mathbf{A}^{[c]} + \mathbf{A}^{(c+1)}$
 - 18: $c \leftarrow c + 1$
 - 19: $\hat{k}_0, \hat{r}_0 \leftarrow \text{EstimateParams}(\mathbf{A}^{[c]}, c)$
 - 20: $t \leftarrow t + 1$
 - 21: **return** \emptyset
-

REMARK 2. The calculation for $|\lceil \bar{m} \rceil_{t_1:t_2}|$ in Algorithm 4 is similar to that in Algorithm 3. The only difference is that for values of π , ψ and B that are needed for calculation of $|\lceil \bar{m} \rceil_{t_1:t_2}|$, we use adjusted profile likelihood estimates with data from layers 1 to $t - \theta$.

REMARK 3. The time complexity of both Algorithms 3 and 4 is $O(TKN^2)$, driven by the partial eigen-decomposition in Steps 5-6 and Steps 4-5 in Algorithms 3 and 4, respectively.

REMARK 4. In both algorithms, tuning parameters consist of the cushion κ , the window $\theta \in \mathbb{N} \setminus \{1\}$, the number of windows $m \in \mathbb{N}$, jump length $v_J \in \{0, \dots, \theta - 1\}$, and check length $v_C \in \mathbb{N}$. In our simulations, we let $\kappa \in \{2, 5\}$; $\theta \in \{2, 6\}$; $v_C, v_J \in \{1, 2\}$.

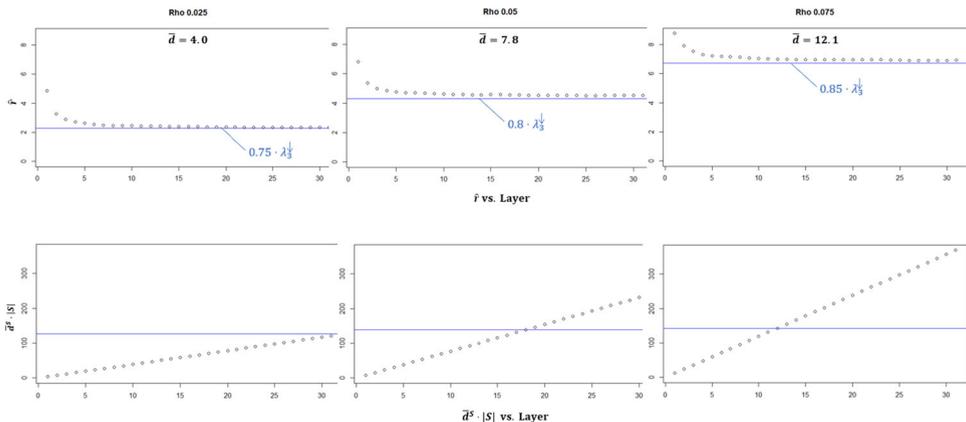
3.2. *\hat{K} -based Change-Point Detection Algorithm* Given a sequence of adjacency matrices for networks, the aggregated adjacency matrices are represented by sum of the adjacency matrices. Once aggregated adjacency matrix is obtained from the sequence, it is transformed into the corresponding Bethe Hessian matrix \mathbf{H}_r^S , and the number of communities \hat{K} is estimated for each \mathbf{H}_r^S . Then, change-points are detected when a change in \hat{K} is observed. The details of the algorithm are given in Algorithm 5.

REMARK 5. The sample size parameter M would need to be set based on initial guesses for K informed by one's domain knowledge such that $M \gg K$. In our simulations with $K \in \{3, 5\}$, the choice $M = 50$ worked well.

REMARK 6. Tuning parameters of the algorithm consist of cushion κ and window θ . κ controls how close the change-point is to the beginning of the sequence such that no change-point occurring before κ can be detected. κ was assigned a value of 2 and 5. θ controls the length of the interval for estimating change-point and was assigned values of 2, 4, and 6.

REMARK 7. Estimating r that yielded accurate results in the main procedure required that \mathbf{A}^S be of sufficient density. In our simulations, that density ranged from 120 to 130, which equated to approximately 15 to 30 layers depending on the setting of the density parameter ρ . The top row in Fig. 1 shows that the first-order linear approximation of r in Steps 9 and 10 yielded an estimate quite close to the limiting value of r given the sparsity setting in the simulations. The bottom row in Fig. 1 shows that the mean density of \mathbf{A}^S at which the estimated r is close to the limiting value is similar at different levels of the density parameter ρ . For other network data, the parameters in the linear approximation should first be estimated prior to applying the algorithm.

REMARK 8. The time complexity of the sub-procedure *EstimateParams* is $O(N^3)$ driven by the eigenvalue computation in Step 2, and $O(TN^3)$ for the main procedure where the eigenvalue computation for $\mathbf{H}(\hat{r})$ in Step 10 is repeated T times.

Figure 1: $\hat{\rho}_0$ and Mean Density by Layer

4 Theoretical Results

In this section, we provide theoretical results of the change-point estimators proposed in Section 3. Recall that the data is given in the form of a sequence of $N \times N$ adjacency matrices $(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(t)}, \dots)$. Since we are considering an online change-point detection problem, we assume that the adjacency matrices are available to us sequentially and at any given point, T , of the sequence, the available set of adjacency matrices are $(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(T)})$. We consider that the data are generated from MDBM as defined in Section 2.3 and MSBM as its special case with $\psi_i = 1$ for all $i \in [N]$ but with potentially different number of communities post-change-point. Our goal now becomes obtaining theoretical properties of the change-point estimators proposed in Section 3. Below, we state the main results for both classes of algorithms. Detailed proofs for all of the results are given in Appendices A and B.

4.1. Assumptions

4.1.1. Assumptions for Theoretical Results for the $\hat{\mathbf{Z}}$ -based Algorithms.

Although the change-point estimator works for any consistent community recovery algorithm, for the sake of convenience, without loss of generality, we consider that the community labels have been estimated by Algorithm 1 to prove theoretical results on consistency. Lemma 1 gives a bound on the misclassification error based on the \mathcal{VI} between the estimated community structures using Algorithm 1. Note that in order to use the bound in Eqs. 2.15, we need Conditions (A) through (D) in Section 2.4 on the parameters of MDBM.

With error bounds on the estimated community labels given in Eqs. 2.15 and 2.16, in Appendix A.3, we provide the relationship between the error bounds and the \mathcal{VI} . In what follows, we prove the consistency of the change-point estimator, $\hat{\tau}$, output by Algorithms 3 and 4.

4.1.2. *Assumptions for Theoretical Results for the \hat{K} -based Algorithm.*

We consider network data generated from MSBM as described in Section 2.1. In addition, we assume the following:

- (F) $\mathbf{B}^{(t)} = \mathbf{B}$ for all $t \geq 1$, and a special assortative structure $\mathbf{B}_{K \times K} := \frac{a-b}{N} \mathbf{I}_K + \frac{b}{N} \mathbf{1}_K \mathbf{1}_K^T$. We note that this special structure is commonly used in the literature to show theoretical properties of SBMs.
- (G) Identical community sizes, i.e., $n_1 = \dots = n_K = N/K$ for all $t \in [1, T]$. Hence, after the change-point, we assume that each community is allocated the same number of nodes.

4.2. *Theoretical Results for the $\hat{\mathbf{Z}}$ -based Algorithms*

4.2.1. *Theoretical Properties of $\hat{\tau}$ in Algorithms 3 and 4.* Both Algorithms 3 and 4 output an estimate of the change-point, $\hat{\tau}$, if the true change-point τ exists within the sequence; otherwise, both return an empty set.

We start with providing a result on the accuracy of estimated community structure. Lemma 1 provides a high probability bound on the errors in estimated community structure.

Lemma 1. *Let $\hat{\xi}_{1:T}$ be an estimate of $\xi_{1:T}$ based on layers $1 : T$ in the network data $(\mathbf{A}_{N \times N}^{(s)})_{s=1}^T$. Let $|\bar{m}_{1:T}|$ be an upper bound on the total number of misclassified nodes in $\hat{\xi}_{1:T}$ with respect to $\xi_{1:T}$ as stated in Theorem 4.9 from Bhattacharyya and Chatterjee (2020b) under Conditions (A) through (D) in Section 2.4. If $|\bar{m}_{1:T}| \geq \frac{K-1}{K}N$, then, $\mathcal{VI}(\hat{\xi}_{1:T}, \xi_{1:T}) \leq 2 \log(K)$. If $|\bar{m}_{1:T}| < \frac{K-1}{K}N$, then as $(Td)^{1/4} \lambda_B \rightarrow \infty$,*

$$\mathbb{P} \left(\mathcal{VI}(\hat{\xi}_{1:T}, \xi_{1:T}) \leq M \left(\xi_{1:T}, \hat{\xi}_{1:T}, N - |\bar{m}_{1:T}| \right) \right) = 1 - o(1), \quad (4.1)$$

where $M \left(\xi_{1:T}, \hat{\xi}_{1:T}, N - |\bar{m}_{1:T}| \right)$ is as defined in Eq. 3.2 of Theorem 4.

The main theoretical results for $\hat{\tau}$ estimated from Algorithm 3 are given in Theorems 5 and 6. Theorem 5 shows that in absence of a change-point, with high probability Algorithm 3 would return a null set. Theorem 6 shows that estimated change-point $\hat{\tau}$ is close to the true change-point τ with high probability. Denote $\hat{\xi}_{1:t}$ as $\hat{\xi}_t$, $\xi_{1:t}$ as ξ_t , $|\bar{m}_{1:t}|$ as $|\bar{m}_t|$. Let ξ_1 be the clustering of layers $1 : \tau - 1$ and ξ_2 be the clustering of layers $t \geq \tau$. Also,

recall the quantities defined in terms of the parameters of MSBM and MDBM of Section 2.3 in Section 3.

THEOREM 5. *There exists $\delta_0 \in \mathbb{N}$ such that for any $t < \tau$ and $t \geq \tau + \delta_0 + \theta$, if $|\bar{m}_t| < \frac{K-1}{K}N$, as $((t - \theta - 1)d)^{1/4}\lambda_B \rightarrow \infty$,*

$$\begin{aligned} \mathbb{P}\left(\mathcal{VI}\left(\hat{\xi}_t, \hat{\xi}_{t-\theta}\right) \leq M\left(\xi_{t-\theta}, \hat{\xi}_t, N - |\bar{m}_t|\right) \right. \\ \left. + M\left(\xi_{t-\theta}, \hat{\xi}_{t-\theta}, N - |\bar{m}_{t-\theta}|\right)\right) = 1 - o(1). \end{aligned} \quad (4.2)$$

THEOREM 6. *Let $\hat{\tau}$ be the estimated change-point from Algorithm 3. Then there exists $\delta_0 \in \mathbb{N}$ such that if $|\bar{m}_t| < \frac{K-1}{K}N$, as $((t - \theta - 1)d)^{1/4}\lambda_B \rightarrow \infty$, $\mathbb{P}(\hat{\tau} \in [\tau, \tau + \theta + \delta_0]) = 1 - o(1)$.*

The proof of Theorem 5 follows from Lemma 1 and 2, where as, proof of Theorem 6 follows from Lemmas 1, 2 and Theorem 7. The details of the proofs are given in the Appendices A.4 - A.8.

Lemma 2. *For any t :*

$$\begin{aligned} \mathcal{VI}(\hat{\xi}_t, \hat{\xi}_{t-\theta}) &\leq \mathcal{VI}(\xi_t, \xi_{t-\theta}) + \mathcal{VI}(\hat{\xi}_t, \xi_t) + \mathcal{VI}(\hat{\xi}_{t-\theta}, \xi_{t-\theta}) \\ \mathcal{VI}(\hat{\xi}_t, \hat{\xi}_{t-\theta}) &\geq \mathcal{VI}(\xi_t, \xi_{t-\theta}) - \mathcal{VI}(\hat{\xi}_t, \xi_t) - \mathcal{VI}(\hat{\xi}_{t-\theta}, \xi_{t-\theta}). \end{aligned} \quad (4.3)$$

THEOREM 7. *There exist $\theta \in \mathbb{Z}^+$, $\delta_1 \in \mathbb{N}$ and $\delta_2 \in \mathbb{N}$ such that for any $\tau + \delta_1 \leq t \leq \tau + \delta_2 < \tau - 1 + 2\theta$, as $((t - \theta - 1)d)^{1/4}\lambda_B \rightarrow \infty$,*

$$\mathbb{P}\left(\mathcal{VI}(\hat{\xi}_t, \hat{\xi}_{t-\theta}) > \mathcal{VI}(\hat{\xi}_{\tau-1}, \hat{\xi}_{\tau-\theta})\right) = 1 - o(1). \quad (4.4)$$

Thus, according to Theorems 5 and 7, $\mathcal{VI}(\hat{\xi}_t, \hat{\xi}_{t-\theta})$ crossing its upper bound indicates a change-point and can be confirmed by a subsequent increase in $\mathcal{VI}(\hat{\xi}_t, \hat{\xi}_{t-\theta})$ in the interval $(\hat{\tau} + \delta_1, \hat{\tau} + \delta_2)$. Theorem 6 indicates that the estimated change-point is not far from the true change-point.

The following theoretical results provide further support to Algorithm 4. Theorem 8 shows that in absence of a change-point, with high probability Algorithm 3 would return a null set. Theorem 9 shows that estimated change-point $\hat{\tau}$ is close to the true change-point τ with high probability.

THEOREM 8. *There exists $\delta_0 \in \mathbb{N}$ such that for any $t < \tau$ and $t \geq \tau + \delta_0 + \theta$, if $|\bar{m}_{(t-\theta+1):t}| < \frac{K-1}{K}N$ and $|\bar{m}_{t-\theta}| < \frac{K-1}{K}N$, as $(d*\theta)^{1/4}\lambda_B \rightarrow \infty$,*

$$\begin{aligned} \mathbb{P}\left(\mathcal{VI}(\tilde{\xi}_t, \hat{\xi}_{t-\theta}) \leq M\left(\xi_{t-\theta}, \hat{\xi}_{(t-\theta+1):t}, N - |\bar{m}_{(t-\theta+1):t}|\right) \right. \\ \left. + M\left(\xi_{t-\theta}, \hat{\xi}_{t-\theta}, N - |\bar{m}_{t-\theta}|\right)\right) = 1 - o(1). \end{aligned} \quad (4.5)$$

THEOREM 9. *Let $\hat{\tau}$ be the estimated change-point from Algorithm 4. There exists $\delta_0 \in \mathbb{N}$ such that if $|\bar{m}_{(t-\theta+1):t}| < \frac{K-1}{K}N$ and $|\bar{m}_{t-\theta}| < \frac{K-1}{K}N$, as $(d * \theta)^{1/4} \lambda_B \rightarrow \infty$, $\mathbb{P}(\hat{\tau} \in [\tau, \tau + \theta + \delta_0]) = 1 - o(1)$.*

The proof of Theorem 8 follows from Lemmas 1 and 3, where as, proof of Theorem 9 follows from Lemmas 1, 3 and Theorem 10. The details of the proofs are given in the Appendices A.9- A.12.

Lemma 3. *For any t :*

$$\begin{aligned} \mathcal{VI}(\tilde{\xi}_t, \hat{\xi}_{t-\theta}) &\leq \mathcal{VI}(\xi_{(t-\theta+1):t}, \xi_{t-\theta}) + \mathcal{VI}(\hat{\xi}_{(t-\theta+1):t}, \xi_{(t-\theta+1):t}) + \mathcal{VI}(\hat{\xi}_{t-\theta}, \xi_{t-\theta}) \\ \mathcal{VI}(\tilde{\xi}_t, \hat{\xi}_{t-\theta}) &\geq \mathcal{VI}(\xi_{(t-\theta+1):t}, \xi_{t-\theta}) - \mathcal{VI}(\hat{\xi}_{(t-\theta+1):t}, \xi_{(t-\theta+1):t}) - \mathcal{VI}(\hat{\xi}_{t-\theta}, \xi_{t-\theta}). \end{aligned} \quad (4.6)$$

THEOREM 10. *There exists $\theta \in \mathbb{Z}^+ \setminus \{1\}$, $\delta_1 \in \mathbb{N}$ and $\delta_2 \in \mathbb{N}$ such that for any $\tau + \delta_1 \leq t \leq \tau + \delta_2$, as $(d * \theta)^{1/4} \lambda_B \rightarrow \infty$*

$$\mathbb{P}\left(\mathcal{VI}(\tilde{\xi}_t, \hat{\xi}_{t-\theta}) > \mathcal{VI}(\tilde{\xi}_{\tau-1}, \hat{\xi}_{\tau-1-\theta})\right) = 1 - o(1). \quad (4.7)$$

Theorems 8 and 10 state that a change-point is inferred when (1) $\mathcal{VI}(\tilde{\xi}_t, \hat{\xi}_{t-\theta})$ crosses its upper bound in Theorem 8 and (2) there is a subsequent increase in $\mathcal{VI}(\tilde{\xi}_t, \hat{\xi}_{t-\theta})$ in $(\hat{\tau} + \delta_1, \hat{\tau} + \delta_2)$. Theorem 9 indicates that the estimated change-point is not far from the true change-point.

4.3. Theoretical Results for the \hat{K} -based Algorithm We prove theoretical results for the change-point estimate $\hat{\tau}$ obtained from Algorithm 5 for the MSBM defined in Section 2.3.

4.3.1. Notation. Define $d := (a + (K - 1)b)/K$ as the expected degree of each node in each layer, $\bar{d}^S := \frac{1}{N|S|} \mathbf{1}_N^T \mathbf{A}^S \mathbf{1}_N$ as the average observed degree for a specific sequence, S , of layers, and $\Delta := \sqrt{N/K} \mathbf{I}_K$. Define $\mathbf{W} := \Delta^{-1} \mathbf{Z}^T \tilde{\mathbf{Z}} \Delta^{-1} = \frac{1}{K} \mathbf{1}_K \mathbf{1}_K^T$. Recall that $\mathbf{Z} \in \mathcal{M}_{N,K}$ (resp. $\tilde{\mathbf{Z}}$) denotes the membership matrix before (resp. after) the change-point $\tau \in [T]$.

4.3.2. Preliminary Results. We start with a few results on the eigenstructure of aggregated adjacency matrices.

Lemma 4. *Under the conditions (F) and (G) on the MSBM, if no change-point is present within the layers in $[T]$, then $\lambda_K^\downarrow(\mathbb{E}\mathbf{A}^{[T]}) \geq \frac{T(a-b)}{K} - \frac{T_a}{N}$ and $\lambda_{K+1}^\downarrow(\mathbb{E}\mathbf{A}^{[T]}) \leq \frac{T_a}{N}$.*

Lemma 5. *For network sequences generated from MSBM with conditions (F) and (G), if \mathbf{Z} (resp. $\tilde{\mathbf{Z}}$) denotes the membership matrix before (resp.*

after) the change-point $\tau \in [T]$, then the spectral decomposition of $t\mathbf{Z}\mathbf{B}\mathbf{Z}^T + (T-t)\tilde{\mathbf{Z}}\tilde{\mathbf{B}}\tilde{\mathbf{Z}}^T$ is given by

$$\mathbf{X} \begin{bmatrix} Td & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & t\frac{a-b}{K}\mathbf{I}_{K-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & (T-t)\frac{a-b}{K}\mathbf{I}_{K-1} \end{bmatrix} \mathbf{X}^T, \text{ where} \\ \mathbf{X} := \left[\frac{1}{\sqrt{N}}\mathbf{1}_N \quad \mathbf{Z}\mathbf{\Delta}^{-1}\mathbf{U} \quad (\tilde{\mathbf{Z}}\mathbf{\Delta}^{-1} - \mathbf{Z}\mathbf{\Delta}^{-1}\mathbf{W})\mathbf{U} \right],$$

and $\mathbf{U} \in \mathbb{R}^{K \times (K-1)}$ is such that $\mathbf{U}^T\mathbf{U} = \mathbf{I}_{K-1}$ and

$$\left[\mathbf{U} \quad \frac{1}{\sqrt{k}}\mathbf{1}_K \right] \text{ is an orthogonal matrix, so } \mathbf{U}\mathbf{U}^T = \mathbf{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^T.$$

4.3.3. Theoretical Properties of the Change-Point Estimator $\hat{\tau}$ in Algorithm 5. Now using the results on the eigen-structure of the aggregated adjacency matrix $\mathbf{A}^{[T]}$, we show that Algorithm 5 estimates the change-point τ with high probability.

THEOREM 11. *Suppose $\psi_{\pm}(r) := r + (1 \pm 1/4)Td/r$. There are constants $c, C > 0$ such that if (a) $Td > C \log N$, (b) either $\psi_+(r) \leq \min\{t, T-t\}\frac{a-b}{K} - T\frac{a}{N}$ or $\max\{t, T-t\}\frac{a-b}{K} - T\frac{a}{N} < \psi_+(r) \leq T\frac{a-b}{K} - T\frac{a}{N}$, and (c) $\psi_-(r) > Ta/N$, then the values of $\hat{K}_r := |\{\ell \in [N] : \lambda_{\ell}^{\uparrow}(H_r^{[T]}) < 0\}|$ in the two cases (i) no change-point within layers $[T]$ and (ii) one change-point at layer $t \in [T]$ are different with probability $\geq 1 - cNe^{-2Td/C}$.*

Theorem 11 indicates that if there is a change at a layer t of the number of negative eigenvalues, \hat{K}_r , of a Bethe Hessian matrix with r in a specific range, it implies the presence of a change-point at the instance t . So, with high probability t becomes an estimate of the change-point τ .

5. Simulation

5.1. Simulation Setup We simulate network data under the frameworks of MSBM and MDBM with $N = 1200$ nodes, $T = 40$ layers and the numbers of communities $K \in \{3, 5\}$. For membership vectors, we set $\mathbf{Z}_{N \times K^{(t)}}^{(t)} = \mathbf{Z}$ for $t < \tau$ and $\mathbf{Z}_{N \times K^{(t)}}^{(t)} = \tilde{\mathbf{Z}}$ for $t \geq \tau$, where τ is the change-point. We generate \mathbf{z}_{i*} from $\text{Mult}\left(1; \left(\frac{1}{K}, \dots, \frac{1}{K}\right)\right)$. We fix the community change ratio $\Xi \in [0, 1]$ and form $\tilde{\mathbf{Z}}$ by (1) randomly sampling nodes from each community with probability Ξ to obtain a node subset of size N' from $[N]$ and (2) for each $k \in [1, K]$ and each index i of the sampled nodes, change \mathbf{z}_{i*} from \mathbf{e}_k to \mathbf{e}_m , $m \neq k$, where m is chosen from $\{[1, K], m \neq k\}$ with probability

$1/(K-1)$. We set the connectivity probability matrix as $\mathbf{B} := \rho[C(\mathbf{I}_K) + b(\mathbf{1}_K\mathbf{1}_K^T)]$, where ρ controls the expected degree of each node at each layer, C controls the in/out ratio and thus determines the degree of assortativity, and b is the baseline value set to 0.1. For MDBM networks, we consider degree parameters $\psi_i \sim \text{Uniform}(0.6, 1)$. To simulate networks with different levels of sparsity and the prominence of the community structure, we consider $\rho \in \{0.0059, 0.025, 0.05\}$ and $C \in \{0.4, 0.5, 0.6\}$. For change-points, we let $\tau \in \{1, 5, 10, 20\}$ and for the community change ratio, we consider $\Xi \in (0.25, 0.5, 0.75)$. Table 1 summarizes the model parameter settings used in the simulations. To examine the performance of Algorithms 3, 4, and 5 under different settings, we use three different windows $\theta \in \{2, 4, 6\}$ when running each algorithm. For the clustering steps of Algorithms 3 and 4, we apply Algorithm 1. To calculate the quantities in Eqs. 3.7 and 3.8, we followed the suggestions in (Bhattacharyya and Chatterjee, 2020a) and (Bhattacharyya and Chatterjee, 2020b) and chose $\Delta = 9$, $\delta = 0.01$ and $\eta = 0.01$ for Algorithms 3 and 4. For Algorithm 3, we let $C = 0.00005$, $C_1 = 0.03$. For Algorithm 4, we chose $C = 0.00025$, $C_1 = 1/6$. Under these settings, \mathcal{VI} performed as expected according to Theorem 4.

5.2. Evaluation As indicators of the quality of input signals for change-point detection, we consider the *in/out ratio* defined as $\Pi := (C + 0.1)/0.1$. We assess the performance of each algorithm based on the following three criteria:

1. *False positive (FP)*: $\hat{\tau}$ is treated as a FP if \emptyset is not the *output* when there is no change-point, and if $\hat{\tau} \notin [\tau, \tau + 2\theta - 2]$ when there is a change-point.
2. *False negative (FN)*: $\hat{\tau}$ is treated as a FN if \emptyset is the *output* or if $\hat{\tau} \notin [\tau, \tau + 2\theta - 2]$ when there is a change-point.
3. *Delay* := $\hat{\tau} - \tau$ when $\hat{\tau} \in [\tau, \tau + 2\theta - 2]$.

5.3. Results We evaluated Algorithms 3, 4, and 5 on synthetic networks generated under all four settings in Table 1. Below is a presentation based on networks generated under Setting D. Performance results for the other settings are presented in the Appendix C. We use the evaluation criteria in Section 5.2 and assess the performance by varying the following model parameters: (i) the community change ratio Ξ ; (ii) the in/out ratio Π ; and (iii) the network sparsity parameter ρ . FP, FN, and delay of Algorithms 3, 4, and 5 with varying levels of Ξ are shown in Fig. 2, the impact of Π on

Table 1: Model Parameter Settings for Simulations

Setting	K	ρ	C	τ	Ξ
A	$\{3, 5\}$	0.025	0.5	$\{1, 5, 10, 20\}$	$\{0.25, 0.5, 0.75\}$
B	$\{3, 5\}$	$\{0.0059, 0.025, 0.05\}$	0.5	$\{1, 5, 10, 20\}$	0.5
C	$\{3, 5\}$	0.025	$\{0.4, 0.5, 0.6\}$	$\{1, 5, 10, 20\}$	0.5
D	3	$\{0.0059, 0.025, 0.05\}$	$\{0.4, 0.5, 0.6\}$	10	$\{0.25, 0.5, 0.75\}$

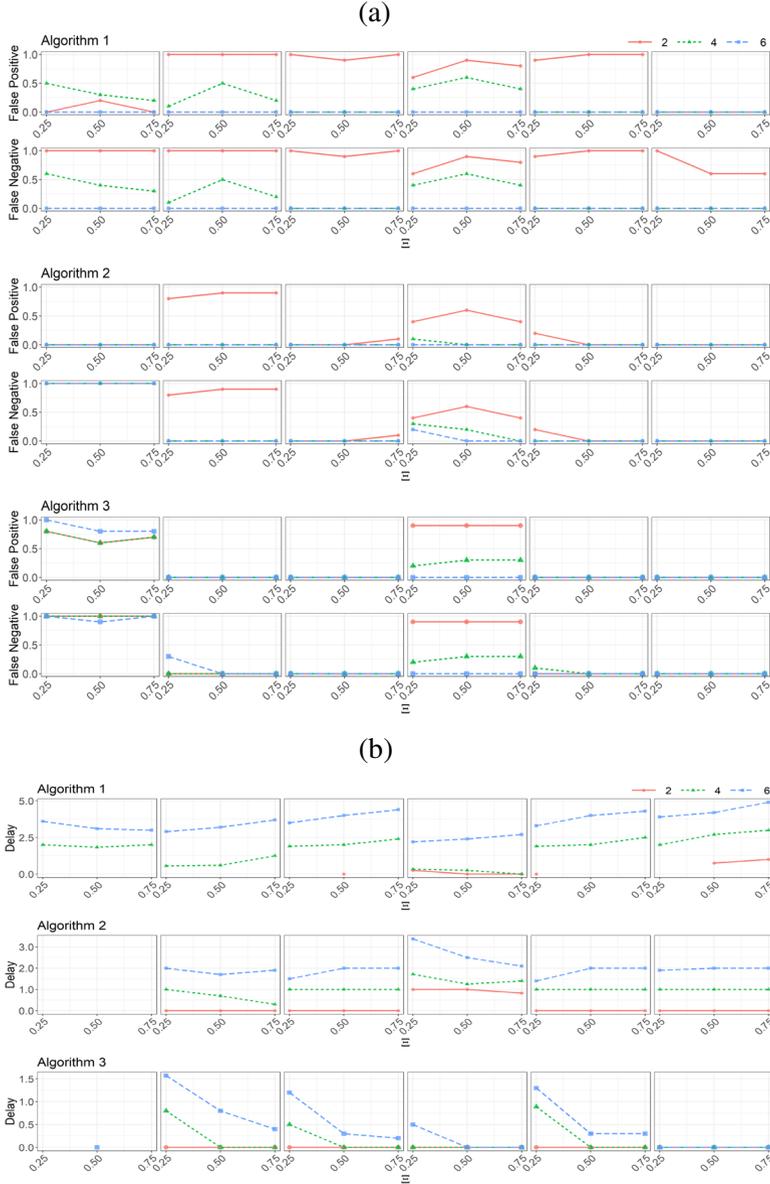


Figure 2: The performance in terms of FP and FN (Subfigure (a)) and delay (Subfigure (b)) of Algorithms 3, 4, and 5 in detecting change-points vs. Ξ with $\Pi = 6$ and $\theta \in \{2, 4, 6\}$. Algorithm 1 was used for the $\hat{\mathbf{Z}}$ -based algorithms. From left to right, subplots in each row are based on the data generated from (MDBM, $\rho = 0.006$), (MDBM, $\rho = 0.025$), (MDBM, $\rho = 0.05$), (MSBM, $\rho = 0.006$), (MSBM, $\rho = 0.025$) and (MSBM, $\rho = 0.05$)

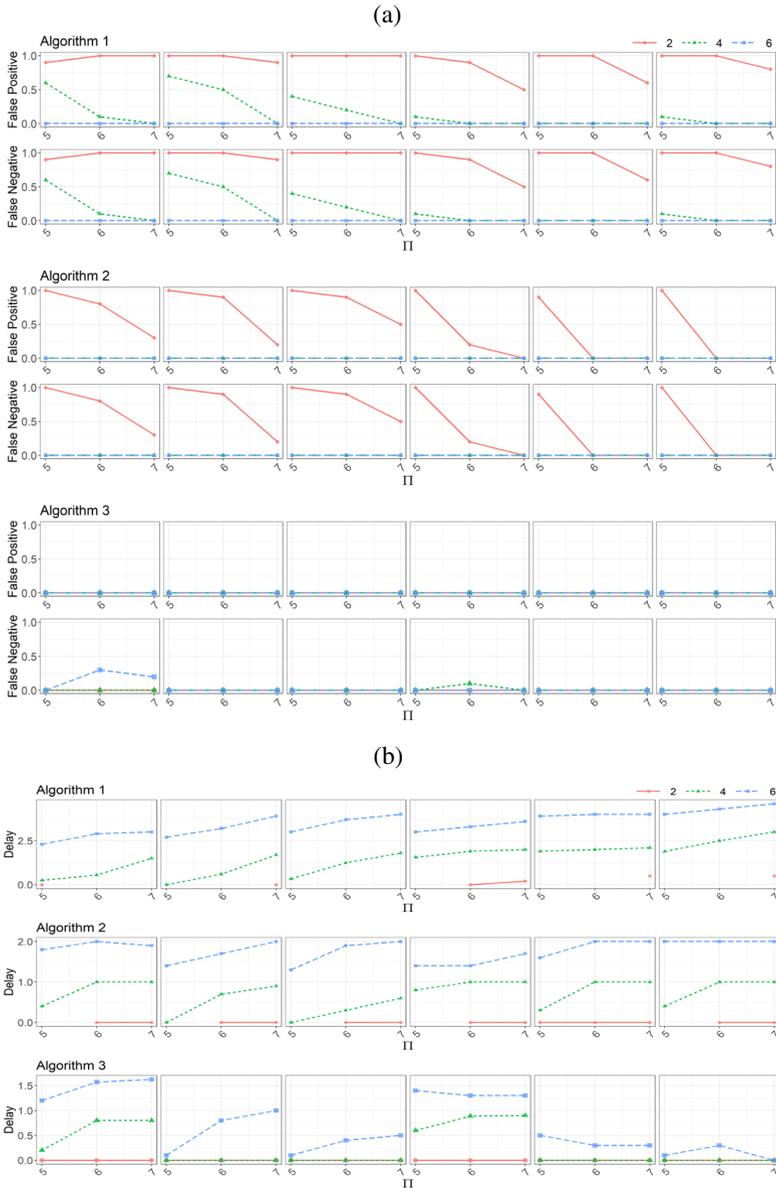


Figure 3: The performance in terms of FP and FN (Subfigure (a)) and delay (Subfigure (b)) of Algorithms 3, 4, and 5 in detecting change-points vs. II with $\rho = 0.025$ and $\theta \in \{2, 4, 6\}$. Algorithm 1 was used for the $\hat{\mathbf{Z}}$ -based algorithms. From left to right, the subplots in each row are based on the data generated from (MDBM, $\Xi = 0.25$), (MDBM, $\Xi = 0.5$), (MDBM, $\Xi = 0.75$), (MSBM, $\Xi = 0.25$), (MSBM, $\Xi = 0.5$) and (MSBM, $\Xi = 0.75$)

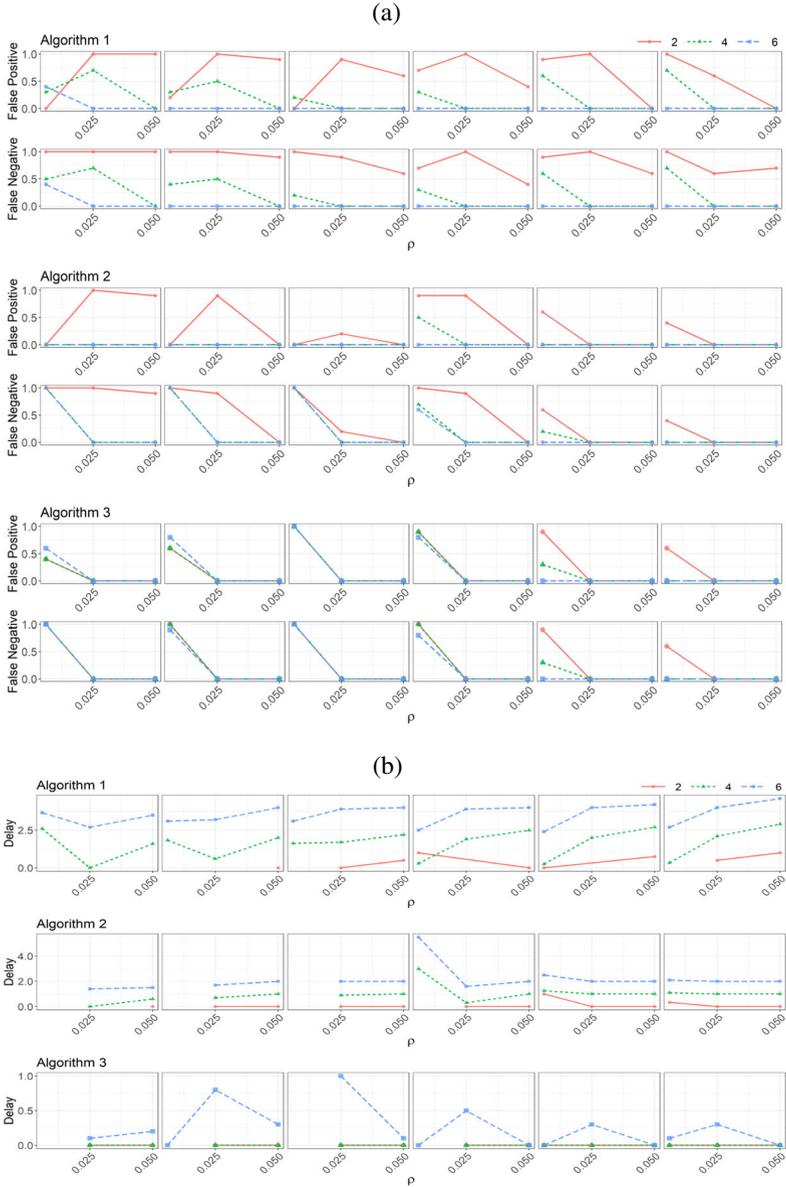


Figure 4: The performance in terms of FP and FN (Subfigure (a)) and delay (Subfigure (b)) of Algorithms 3, 4, and 5 in detecting change-points vs. ρ with $\Xi = 0.5$ and $\theta \in \{2, 4, 6\}$. Algorithm 1 was used for the $\hat{\mathbf{Z}}$ -based algorithms. From left to right, subplots in each row are based on the data generated from (MDBM, $II = 5$), (MDBM, $II = 6$), (MDBM, $II = 7$), (MSBM, $II = 5$), (MSBM, $II = 6$) and (MSBM, $II = 7$)

the performance is shown in Fig. 3, and ρ in Fig. 4. In Figs. 2 through 4, FP and FN are shown in Subfigure (A) while $Delay$ are shown in Subfigure (B).

In Figs. 2 through 4, FP , FN , and delay of Algorithms 3, 4, and 5 versus the window θ are shown. Higher θ results in higher accuracy as measured by FP and FN while leading to an increase in delay, since the weight of each new layer gets relatively smaller with larger θ . This can be confirmed in Subfigure (A) in Figs. 2, 3, and 4 where all algorithms show better performance with $\theta = 6$ while Subfigure (B) show less delay with $\theta = 2$ for all algorithms. Hence there is an accuracy-delay trade-off in the choice of θ . Noticing that the maximal value of $Delay$ is θ , a general strategy is to choose θ that equals to the maximal acceptable $Delay$.

It can be seen from Subfigure (A) in Figs. 2, 3, and 4 that all algorithms tend to have better accuracy (as reflected by FP and FN) on networks with larger values of either Ξ , Π or ρ , especially when θ is set to a large value. However, as can be confirmed in Subfigure (B) in Figs. 2, 3, and 4, there is no evidence of a decrease in $Delay$ for networks with larger values of Ξ , Π or ρ . There are multiple factors that can affect the $Delay$ of the algorithms. Take Algorithm 3 as an example. The idea of the algorithm is to compare clustering estimation based on historical and updated sub-sequences of layers of networks. This means that when there is an increase in Ξ , Π or ρ , clustering estimations become more precise on the one hand, while more layers of networks after the change-point are needed in the updated sub-sequence.

It is evident in Figs. 2 and 4 that while all three algorithms perform similarly in sparse networks ($\rho = 0.006$), the \hat{K} -based Algorithm (5) outperforms the \hat{Z} -based Algorithms (3 and 4) for higher values of ρ . That \hat{K} -based algorithm performs better than \hat{Z} -based algorithms makes sense since community structure estimation is generally harder than community number estimation.

6 Conclusion

In this paper, we proposed three algorithms based on two broad approaches for online change-point detection for network sequences with community structure. Two of the three algorithms are based on the change in estimated community structures and the third based on the change in the estimated numbers of communities in networks represented as the Bethe Hessian matrices. We proved theoretical results showing the efficacy of the change-point estimates asymptotically as aggregated degree of the networks

grows to infinity. We proved the theoretical results for general cases of both the MSBM and MDBM for Algorithms 3 and 4. For Algorithm 5, we proved the theoretical results for the special case of the MSBM.

One of the obvious next steps would be to generalize the theoretical framework for Algorithm 5 and show that the change-point estimate can recover the change-point under a much general model set up. Tackling dependent sequence of networks will also be another promising set of future problems.

References

- AMINIKHANGHAHI, S. and COOK, D.J. (2017). A survey of methods for time series change point detection. *Knowl. Inf. Syst.* **51**, 339–367.
- ANGEL, O., FRIEDMAN, J. and HOORY, S. (2015). The non-backtracking spectrum of the universal cover of a graph. *Trans. Am. Math. Soc.* **367**, 4287–4318.
- BAI, J. and PERRON, P. (1998). Estimating and testing linear models with multiple structural changes. *Econometrica* 47–78.
- BAO, W. and MICHAELIDIS, G. (2018). Core community structure recovery and phase transition detection in temporally evolving networks. *Sci. Rep.* **8**, 1–16.
- BHAMIDI, S., JIN, J., NOBEL, A. et al. (2018). Change point detection in network models: Preferential attachment and long range dependence. *Ann. Appl. Probab.* **28**, 35–78.
- BHATTACHARJEE, M., BANERJEE, M. and MICHAELIDIS, G. (2018). Change point estimation in a dynamic stochastic block model. arXiv:[1812.03090](https://arxiv.org/abs/1812.03090).
- BHATTACHARYYA, S. and CHATTERJEE, S. (2020). Consistent recovery of communities from sparse multi-relational networks: a scalable algorithm with optimal recovery conditions. *Complex networks XI*, pp. 92–103. Springer.
- BHATTACHARYYA, S. and CHATTERJEE, S. (2020). General community detection with optimal recovery conditions for multi-relational sparse networks with dependent layers.
- 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.
- BLEAKLEY, K. and VERT, J.P. (2011).
- BLONDER, B., WEY, T.W., DORNHAUS, A., JAMES, R. and SIH, A. (2012). Temporal dynamics and network analysis. *Methods Ecol. Evol.* **3**, 958–972.
- BORDENAVE, C., LELARGE, M. and MASSOULIÉ, L. (2015). Non-backtracking spectrum of random graphs: community detection and non-regular ramanujan graphs. In *2015 IEEE 56th annual symposium on foundations of computer science*, pp. 1347–1357. IEEE.
- BOSC, M., HEITZ, F., ARMSPACH, J.P., NAMER, I., GOUNOT, D. and RUMBACH, L. (2003). Automatic change detection in multimodal serial mri: application to multiple sclerosis lesion evolution. *Neuroimage* **20**, 643–656.
- BRODSKY, E. and DARKHOVSKY, B.S. (2013). Nonparametric methods in change point problems, vol. 243 Springer Science & Business Media.
- BRUNA, J. and LI, X. (2017). Community detection with graph neural networks. *Stat.* **1050**, 27.
- VAN DE BUNT, G.G., VAN DULJN, M.A. and SNIJDERS, T.A. (1999). Friendship networks through time: an actor-oriented dynamic statistical network model. *Comput. Math. Organ. Theory* **5**, 167–192.

- CAPE, J., TANG, M. and PRIEBE, C.E. (2017). The kato–temple inequality and eigenvalue concentration with applications to graph inference. *Electron. J. Stat.* **11**, 3954–3978.
- CELIK, T. (2009). Unsupervised change detection in satellite images using principal component analysis and k -means clustering. *IEEE Geosci. Remote Sens. Lett.* **6**, 772–776.
- CELIK, T. (2010). Image change detection using gaussian mixture model and genetic algorithm. *J. Vis. Commun. Image Represen.* **21**, 965–974.
- CHEN, H. et al. (2019). Sequential change-point detection based on nearest neighbors. *Ann. Stat.* **47**, 1381–1407.
- CHEN, J. and GUPTA, A.K. (2011). Parametric statistical change point analysis: with applications to genetics, medicine, and finance. Springer Science & Business Media.
- CHEN, S., ILANY, A., WHITE, B.J., SANDERSON, M.W. and LANZAS, C. (2015). Spatial-temporal dynamics of high-resolution animal networks: what can we learn from domestic animals? *PloS one* 10(6).
- CHO, H. and FRYZLEWICZ, P. (2015). Multiple-change-point detection for high dimensional time series via sparsified binary segmentation. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **77**, 475–507.
- COSTE, S. and ZHU, Y. (2019). Eigenvalues of the non-backtracking operator detached from the bulk. arXiv:1907.05603.
- CRIBBEN, I., HARALDSDOTTIR, R., ATLAS, L.Y., WAGER, T.D. and LINDQUIST, M.A. (2012). Dynamic connectivity regression: determining state-related changes in brain connectivity. *Neuroimage* **61**, 907–920.
- DALL’AMICO, L. and COUILLET, R. (2019). *Community detection in sparse realistic graphs: Improving the bethe hessian*. In *ICASSP 2019-2019 IEEE International conference on acoustics, speech and signal processing (ICASSP)*, pp. 2942–2946. IEEE.
- DALL’AMICO, L., COUILLET, R. and TREMBLAY, N. (2019). Revisiting the bethe-hessian: improved community detection in sparse heterogeneous graphs. In *Advances in neural information processing systems*, pp. 4039–4049.
- DALL’AMICO, L., COUILLET, R. and TREMBLAY, N. (2020). Optimal laplacian regularization for sparse spectral community detection. In *ICASSP 2020-2020 IEEE International conference on acoustics, speech and signal processing (ICASSP)*, pp. 3237–3241. IEEE.
- FERRAZ COSTA, A., YAMAGUCHI, Y., JUCI MACHADO TRAINA, A., TRAINA, JR C. and FALOUTSOS, C. (2015). Rsc: Mining and modeling temporal activity in social media. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 269–278. ACM.
- GAO, C. and LAFFERTY, J. (2017). Testing network structure using relations between small subgraph probabilities. arXiv:1704.06742.
- GATES, M.C. and WOOLHOUSE, M.E. (2015). Controlling infectious disease through the targeted manipulation of contact network structure. *Epidemics* **12**, 11–19.
- GIRSHICK, M.A. and RUBIN, H. (1952). A bayes approach to a quality control model. *Ann. Math. Stat.*, 114–125.
- GULIKERS, L., LELARGE, M. and MASSOULIÉ, L. (2016). Non-backtracking spectrum of degree-corrected stochastic block models. arXiv:1609.02487.
- HARCHAOUI, Z., VALLET, F., LUNG-YUT-FONG, A. and CAPPÉ, O. (2009). A regularized kernel-based approach to unsupervised audio segmentation. In *2009 IEEE International conference on acoustics, speech and signal processing*, pp. 1665–1668. IEEE.
- HASHIMOTO, K.I. (1989). Zeta functions of finite graphs and representations of p -adic groups. In *Automorphic forms and geometry of arithmetic varieties*, pp. 211–280. Elsevier.

- HOCKING, T.D., SCHLEIERMACHER, G., JANOUÉIX-LEROSEY, I., BOEVA, V., CAPPO, J., DELATTRE, O., BACH, F. and VERT, J.P. (2013). Learning smoothing models of copy number profiles using breakpoint annotations. *BMC Bioinform.* **14**, 164.
- HOGG, T. and LERMAN, K. (2012). Social dynamics of digg. *EPJ Data Sci.* **1**, 5.
- HOLME, P. (2015). Modern temporal network theory: a colloquium. *Eur. Phys. J. B* **88**, 234.
- HOLME, P. and SARAMÄKI, J. (2012). Temporal networks. *Phys. Rep.* **519**, 97–125.
- JACOBS, A.Z., WAY, S.F., UGANDER, J. and CLAUSET, A. (2015). Assembling thefacebook: Using heterogeneity to understand online social network assembly. In *Proceedings of the ACM Web Science Conference*, pp. 1–10.
- JIN, J., KE, Z. and LUO, S. (2018). Network global testing by counting graphlets. In *International conference on machine learning*, pp. 2333–2341.
- KASETKASEM, T. and VARSHNEY, P.K. (2002). An image change detection algorithm based on markov random field models. *IEEE Trans. Geosci. Remote Sens.* **40**, 1815–1823.
- KOLMOGOROV, A.N. (1950). Unbiased estimates. *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya* **14**, 303–326.
- KRINGS, G., KARSAI, M., BERNHARDSSON, S., BLONDEL, V.D. and SARAMÄKI, J. (2012). Effects of time window size and placement on the structure of an aggregated communication network. *EPJ Data Sci.* **1**, 4.
- KRZAKALA, F., MOORE, C., MOSSEL, E., NEEMAN, J., SLY, A., ZDEBOROVÁ, L. and ZHANG, P. (2013). Spectral redemption in clustering sparse networks. *Proc. Natl. Acad. Sci.* **110**, 20935–20940.
- LAHIRI, M. and BERGER-WOLF, T.Y. (2007). Structure prediction in temporal networks using frequent subgraphs. In *2007 IEEE Symposium on computational intelligence and data mining*, p. 35–42. IEEE.
- LAVIELLE, M. and TEYSSIERE, G. (2007). Adaptive detection of multiple change-points in asset price volatility. In *Long memory in economics*, pp. 129–156. Springer.
- LE, C.M. and LEVINA, E. (2015). Estimating the number of communities in networks by spectral methods. arXiv:1507.00827.
- LEI, J., RINALDO, A. et al. (2015). Consistency of spectral clustering in stochastic block models. *Ann. Stat.* **43**, 215–237.
- LÉVY-LEDUC, C., ROUEFF, F. et al. (2009). Detection and localization of change-points in high-dimensional network traffic data. *Ann. Appl. Stat.* **3**, 637–662.
- LORDEN, G. et al. (1971). Procedures for reacting to a change in distribution. *Ann. Math. Stat.* **42**, 1897–1908.
- MASUDA, N. and HOLME, P. (2017). *Temporal network epidemiology*. Springer.
- MATTESON, D.S. and JAMES, N.A. (2014). A nonparametric approach for multiple change point analysis of multivariate data. *J. Am. Stat. Assoc.* **109**, 334–345.
- MEILÄ, M. (2007). Comparing clusterings—an information based distance. *J. Multi. Anal.* **98**, 873–895.
- MISLOVE, A.E. (2009). *Online social networks: measurement, analysis, and applications to distributed information systems*. Ph.D thesis.
- OMODEI, E., DE DOMENICO, M.D. and ARENAS, A. (2015). Characterizing interactions in online social networks during exceptional events. *Front. Phys.* **3**, 59.
- PADILLA, O.H.M., YU, Y. and PRIEBE, C.E. (2019). Change point localization in dependent dynamic nonparametric random dot product graphs. arXiv:1911.07494.
- PAGE, E.S. (1954). Continuous inspection schemes. *Biometrika* **41**, 100–115.
- PAGE, E.S. (1957). On problems in which a change in a parameter occurs at an unknown point. *Biometrika* **44**, 248–252.

- PANISSON, A., GAUVIN, L., BARRAT, A. and CATTUTO, C. (2013). Fingerprinting temporal networks of close-range human proximity. In *2013 IEEE International conference on pervasive computing and communications workshops (PERCOM workshops)*, pp. 261–266. IEEE.
- PARK, H.J. and FRISTON, K. (2013). Structural and functional brain networks: from connections to cognition. *Science* **342**, 1238411.
- PARK, Y., PRIEBE, C.E. and YOUSSEF, A. (2013). Anomaly detection in time series of graphs using fusion of graph invariants. *IEEE J. Select. Top. Signal Process.* **7**, 67–75.
- PEEL, L. and CLAUSET, A. (2015). *Detecting change points in the large-scale structure of evolving networks*. In *AAAI*, pp. 2914–2920.
- PEIXOTO, T.P. (2015). Inferring the mesoscale structure of layered, edge-valued, and time-varying networks, vol. 92.
- PEIXOTO, T.P. and GAUVIN, L. (2018). Change points, memory and epidemic spreading in temporal networks. *Scient. Rep.* **8**, 15511.
- PICARD, F., ROBIN, S., LAVIELLE, M., VAISSE, C. and DAUDIN, J.J. (2005). A statistical approach for array cgh data analysis. *BMC Bioinform.* **6**, 27.
- POPOVIĆ, M., ŠTEFANČIĆ, H., SLUBAN, B., NOVAK, P.K., GRČAR, M., MOZETIČ, I., PULIGA, M. and ZLATICIĆ, V. (2014). Extraction of temporal networks from term co-occurrences in online textual sources. *PloS one* **9**, e99515.
- RADKE, R.J., ANDRA, S., AL-KOFAHI, O. and ROYSAM, B. (2005). Image change detection algorithms: a systematic survey. *IEEE Trans. Image Process.* **14**, 294–307.
- RANSHOUS, S., SHEN, S., KOUTRA, D., HARENBERG, S., FALOUTSOS, C. and SAMATOVA, N.F. (2015). Anomaly detection in dynamic networks: a survey. *Wiley Interdiscip. Rev. Comput. Stat.* **7**, 223–247.
- REEVES, J., CHEN, J., WANG, X.L., LUND, R. and LU, Q.Q. (2007). A review and comparison of changepoint detection techniques for climate data. *J. Appl. Meteorol. Climatol.* **46**, 900–915.
- RIGBOLT, K.T., PROKHOROVA, T.A., AKIMOV, V., HENNINGSEN, J., JOHANSEN, P.T., KRATCHMAROVA, I., KASSEM, M., MANN, M., OLSEN, J.V. and BLAGOEV, B. (2011). System-wide temporal characterization of the proteome and phosphoproteome of human embryonic stem cell differentiation. *Sci. Signal.* **4**, rs3–rs3.
- ROCHA, L.E., LILJEROS, F. and HOLME, P. (2010). Information dynamics shape the sexual networks of internet-mediated prostitution. *Proc. Natl. Acad. Sci.* **107**, 5706–5711.
- ROCHA, L.E., LILJEROS, F. and HOLME, P. (2011). Simulated epidemics in an empirical spatiotemporal network of 50,185 sexual contacts. *PLoS computational biology* **7**(3).
- ROY, S., ATCHADÉ, Y. and MICHAILIDIS, G. (2017). Change point estimation in high dimensional markov random-field models. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **79**, 1187–1206.
- SAADE, A., KRZAKALA, F. and ZDEBOROVÁ, L. (2014). Spectral clustering of graphs with the bethe hessian. *Advances in neural information processing systems*, pp. 406–414.
- SAADE, A., KRZAKALA, F. and ZDEBOROVÁ, L. (2014). Spectral density of the non-backtracking operator on random graphs. *EPL Europhys. Lett.* **107**, 50005.
- SALATHÉ, M., KAZANDJIEVA, M., LEE, J.W., LEVIS, P., FELDMAN, M.W. and JONES, J.H. (2010). A high-resolution human contact network for infectious disease transmission. *Proc. Natl. Acad. Sci.* **107**, 22020–22025.
- SHIRYAEV, A.N. (1963). On optimum methods in quickest detection problems. *Theory Probab. App.* **8**, 22–46.
- SIEGMUND, D. (2013). Change-points: from sequential detection to biology and back. *Seq. Anal.* **32**, 2–14.

- SIKDAR, S., GANGULY, N. and MUKHERJEE, A. (2016). Time series analysis of temporal networks. *Eur. Phys. J. B* **89**, 11.
- SPORNS, O. (2013). Structure and function of complex brain networks. *Dialogues Clin. Neurosci.* **15**, 247.
- STAUDACHER, M., TELSER, S., AMANN, A., HINTERHUBER, H. and RITSCH-MARTE, M. (2005). A new method for change-point detection developed for on-line analysis of the heart beat variability during sleep. *Phys. A Stat. Mech. Appl.* **349**, 582–596.
- STOPCZYNSKI, A., SEKARA, V., SAPIEZYNSKI, P., CUTTONE, A., MADSEN, M.M., LARSEN, J.E. and LEHMANN, S. (2014). Measuring large-scale social networks with high resolution. *PloS one* **9**, e95978.
- THOMPSON, W.H., BRANTEFORS, P. and FRANSSON, P. (2017). From static to temporal network theory: Applications to functional brain connectivity. *Netw. Neurosci.* **1**, 69–99.
- VISWANATH, B., MISLOVE, A., CHA, M. and GUMMADI, K.P. (2009). *On the evolution of user interaction in facebook. In Proceedings of the 2nd ACM workshop on Online social networks, pp. 37–42.*
- WANG, D., YU, Y. and RINALDO, A. (2018). Optimal change point detection and localization in sparse dynamic networks. arXiv:[1809.09602](https://arxiv.org/abs/1809.09602).
- WANG, Y., CHAKRABARTI, A., SIVAKOFF, D. and PARTHASARATHY, S. (2017). *Fast change point detection on dynamic social networks. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp. 2992–2998. AAAI Press.*
- WANG, Y.R., BICKEL, P.J. et al. (2017). Likelihood-based model selection for stochastic block models. *Ann. Stat.* **45**, 500–528.
- WATANABE, Y. and FUKUMIZU, K. (2009). Graph zeta function in the bethe free energy and loopy belief propagation. *Advances in neural information processing systems*, pp. 2017–2025.
- WILLS, P. and MEYER, F.G. (2019). Change point detection in a dynamic stochastic block-model. *International conference on complex networks and their applications*, pp. 211–222. Springer.
- YANG, P., DUMONT, G. and ANSERMINO, J.M. (2006). Adaptive change detection in heart rate trend monitoring in anesthetized children. *IEEE Trans. Biomed. Eng.* **53**, 2211–2219.
- YEDIDIA, J.S., FREEMAN, W.T. and WEISS, Y. (2003). Understanding belief propagation and its generalizations. *Explor. Artif. Intell. New Millennium* **8**, 236–239.
- ZHANG, X., SHAO, S., STANLEY, H.E. and HAVLIN, S. (2014). Dynamic motifs in socio-economic networks. *EPL Europhys. Lett.* **108**, 58001.
- ZHAO, L., WANG, G.J., WANG, M., BAO, W., LI, W. and STANLEY, H.E. (2018). Stock market as temporal network. *Physic. A Stat. Mech. Appl.* **506**, 1104–1112.
- ZHAO, Z., CHEN, L. and LIN, L. (2019). Change-point detection in dynamic networks via graphon estimation. arXiv:[1908.01823](https://arxiv.org/abs/1908.01823).

NEIL HWANG

CITY UNIVERSITY OF NEW YORK - BRONX
COMMUNITY COLLEGE, NEW YORK, NY
USA

E-mail: neil.hwang@bcc.cuny.edu

JIARUI XU

SHARMODEEP BHATTACHARYYA
OREGON STATE UNIVERSITY, CORVALLIS,
OR USA

E-mail: xujiar@oregonstate.edu

E-mail: bhattash@science.oregonstate.edu

SHIRSHENDU CHATTERJEE

CITY UNIVERSITY OF NEW YORK - CITY
COLLEGE AND GRADUATE CENTER, NEW
YORK, NY USA

E-mail: shirshendu@ccny.cuny.edu

Paper received: 16 May 2020; accepted 16 March 2021.