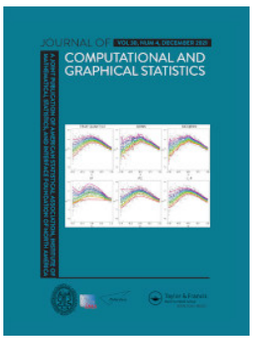

Taylor & Francis
Taylor & Francis Group

Journal of Computational and Graphical Statistics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ucgs20

Fast and Scalable Algorithm for Detection of Structural Breaks in Big VAR Models

Abolfazl Safikhani, Yue Bai & George Michailidis

To cite this article: Abolfazl Safikhani, Yue Bai & George Michailidis (2021): Fast and Scalable Algorithm for Detection of Structural Breaks in Big VAR Models, Journal of Computational and Graphical Statistics, DOI: 10.1080/10618600.2021.1950005

To link to this article: https://doi.org/10.1080/10618600.2021.1950005

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2021, VOL. 00, NO. 0, 1–14
https://doi.org/10.1080/10618600.2021.1950005

Fast and Scalable Algorithm for Detection of Structural Breaks in Big VAR Models

Abolfazl Safikhani, Yue Bai, and George Michailidis

Department of Statistics and Informatics Institute, University of Florida, Gainesville, FL

ABSTRACT

Many real time series datasets exhibit structural changes over time. A popular model for capturing their temporal dependence is that of vector autoregressions (VAR), which can accommodate structural changes through time evolving transition matrices. The problem then becomes to both estimate the (unknown) number of structural break points, together with the VAR model parameters. An additional challenge emerges in the presence of very large datasets, namely on how to accomplish these two objectives in a computational efficient manner. In this article, we propose a novel procedure which leverages a *block segmentation scheme* (BSS) that reduces the number of model parameters to be estimated through a regularized least-square criterion. Specifically, BSS examines appropriately defined blocks of the available data, which when combined with a fused lasso-based estimation criterion, leads to significant computational gains *without* compromising on the statistical accuracy in identifying the number and location of the structural breaks. This procedure is further coupled with new local and exhaustive search steps to consistently estimate the number and relative location of the break points. The procedure is scalable to big high-dimensional time series datasets with a computational complexity that can achieve $O\left(\sqrt{n}\right)$, where n is the length of the time series (sample size), compared to an exhaustive procedure that requires $O\left(n\right)$ steps. Extensive numerical work on synthetic data supports the theoretical findings and illustrates the attractive properties of the procedure. Finally, an application to a neuroscience dataset exhibits its usefulness in applications. Supplementary files for this article are available online.

ARTICLE HISTORY
Received May 2020
Revised June 2021

KEYWORDS
Block segmentation; Fused lasso; High-dimensional time series; Piecewise stationarity; Structural breaks

1. Introduction

Multivariate stationary time series models represent an effective toolkit for modeling interactions of entities observed over time with numerous applications across different scientific fields including engineering, social sciences, biology, and economics. However, many real datasets do not satisfy the stationarity assumption (see, e.g., discussion in Cho and Fryzlewicz (2015) for financial data and a specific example of log-returns of stocks exhibiting structural breaks due to economic shocks in Lin and Michailidis (2017), as well as Ombao, Von Sachs, and Guo (2005) for brain signal data). Hence, there has been interest in models that can accommodate the presence of nonstationarity. Piecewise stationary models comprise a popular class due to their simple form, but also wide applicability. Their main assumption is that the model parameters remain constant over stretches of time and only change at certain time points called "break points." In the most general setup, both the number of break (change) points and their locations are unknown and need to be estimated from the data together with the model parameter values before and after each break point. Detecting the break points and estimating all other parameters requires a search over all time points, which for large datasets becomes computationally expensive.

Due to the wide applicability of time series models exhibiting structural breaks, there exists a large body of literature addressing the problem of change point detection (see, e.g., the earlier books by Basseville and Nikiforov 1993; Csörgö and Horváth 1997, and more recently the review article by Aue and Horváth 2013). This literature can be categorized into three groups with respect to the number of time series components considered by the model—that is, univariate, multivariate, and high-dimensional.

Most of the earlier work focused on the univariate "signal plus noise" model, where mean shifts occur between change points. More recently, the focus has shifted to more complex models. For example, Davis, Lee, and Rodriguez-Yam (2006) used a minimum description length procedure to locate break points in univariate auto-regressive models, Fryzlewicz and Subba Rao (2014) developed a detection method for piecewise constant parameter ARCH model, while Aue, Rice, and Sönmez (2018) developed a method for detecting shifts in the mean of functional data models. Another line of research has examined the case of multiple change points; for example, Harchaoui and Lévy-Leduc (2010) used a total variation penalty to find sudden changes in the mean structure and the computational complexity of the proposed procedure is of order $O\left(n \log n\right)$, where n is the sample size, while Fryzlewicz (2017) applied a tail-greedy Haar transformation to consistently estimate the break points with computational complexity of order $O\left(n \log^2 n\right)$. Moreover, Killick, Fearnhead, and Eckley (2012) introduced a pruning step within a dynamic

CONTACT Abolfazl Safikhani a.safikhani@ufl.edu Department of Statistics and Informatics Institute, University of Florida, Gainesville, FL 32611-7011.
Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JCGS.

programming (DP)-based procedure to detect break points with computational complexity of order $O(n \log n)$. In the multivariate case with the number of time series p fixed, Preuss, Puchstein, and Dette (2015) proposed a nonparametric method to detect anomalies in the auto-covariance function of a multivariate (second order) piecewise stationary process, Ombao, Von Sachs, and Guo (2005) developed a spectral representation to locate the break points with computational complexity of order $O(n \log^2 n + n^3 \log n)$ (see also Rinaldo et al. (2020) for similar ideas for linear regression models), while in Matteson and James (2014), a nonparametric method is developed for detecting abrupt changes in a distribution over time, with computational complexity of order $O(n^2)$. Further, Kaul et al. (2021) established inference procedures for the location of break points in high-dimensional mean shift models. In Wang et al. (2019), an l_0-optimization procedure is utilized for break point detection in VAR models with computational complexity of order $O(n^2\text{Lasso}(n))$, while in Leonardi and Bühlmann (2016), an exact DP algorithm is proposed to detect changes in high-dimensional linear regression with computational complexity of order $O(n^2\text{Lasso}(n))$, where $\text{Lasso}(n)$ is the cost to compute the Lasso estimator for a sample of size n. Combined with a (wild) binary segmentation (BS) algorithm, the CUSUM statistics can consistently detect multiple change-points, in a univariate time series (Fryzlewicz 2014) and also high-dimensional ones (Cho and Fryzlewicz 2015; Cho 2016). Roy, Atchadé, and Michailidis (2017) developed a likelihood-based method for locating a single break point for high-dimensional Markov random fields and provide the rate of estimating the change point, as well as the model parameters. Finally, Safikhani and Shojaie (2020) used fused lasso and a screening step to estimate multiple break points in a VAR model with computational complexity of order $O(n)$ for a fixed number of time series components, and also establish consistency results for both the break points and the model parameters, while a similar procedure is developed in Bai, Safikhani, and Michailidis (2020) to deal with transition matrices exhibiting low rank and sparse structure.

However, in the presence of relatively few change points and very long time series, it would be beneficial to devise much faster algorithms than those available in the current literature. To that end, this article introduces an algorithm for structural break detection in high-dimensional piecewise stationary VAR models that achieves *sublinear computational cost* in the number of the observations, by putting mild conditions on the spacing of consecutive break points. Specifically, it segments the original time series into k_n blocks of size b_n ($n = k_n b_n$) that reduces the dimensionality of the parameter space to $O(k_n)$. The proposed *block segmentation scheme* (BSS) speeds up computations in locating the change points, which together with a screening procedure (see details in Section 3) *identifies consistently* all true change points. This combination reduces—through appropriately choosing the block size b_n (and hence the number of blocks k_n)—computational complexity from $O(n)$ to approximately $O\left(\frac{n}{b_n} + b_n\right)$ for a fixed number of time series components, which for relatively sparsely spaced break points becomes $O(\sqrt{n})$, and thus attractive for a number of applications.

The fastest current method for break point detection in high-dimensional time series with theoretical guarantees takes at least $O(n)$ time. Such a computational time becomes prohibitive in the era of big data where tens of thousands or more temporally observed data points are easy to collect, as the neuroscience application discussed in this article shows. On the other hand, the proposed BSS algorithm reduces this time to $O(\sqrt{n})$ (in the best case scenario), while exhibiting one of the best detection accuracy rates in the literature both theoretically and empirically. Further, the minimum spacing required between consecutive true break points in BSS (which is the bottleneck of all detection methods) is less than most of the currently available detection methods for multivariate time series, including Cho and Fryzlewicz (2015), Cho (2016), Wang, Yu, and Rinaldo (2017), Wang and Samworth (2018), and Barigozzi, Cho, and Fryzlewicz (2018) (see more details in Remark 6, as well as numerical comparisons in Section 5.2).

Extensive numerical comparisons with competing methods (Wang et al. 2019; Safikhani and Shojaie 2020; Cho and Fryzlewicz 2015; Cho 2016) show that the BSS algorithm outperforms them both in terms of detection accuracy and computation time; see details in Section 5.2 and Appendix E in the supplementary material.

The proposed BSS poses a number of technical challenges for establishing consistency of the number, locations and VAR model parameters that are satisfactory resolved in this study. The first involves selection of the block size b_n that needs to be adequately large to reduce computational time (through its impact on the number of blocks k_n), but also not exceedingly large that would lead to missing any of the true break points. Further, very large block sizes would also make it impossible to verifying the restricted eigenvalue (RE) condition (Basu and Michailidis 2015) needed to establish theoretical guarantees provided by BSS. This issue is carefully addressed through Assumption A3 and also in Remark 3 and Section 4.1. In addition, the BSS method introduces several additional theoretical/technical challenges including the introduction of a local screening step based on a corresponding local information criterion to "thin out" candidate change points, the subsequent verification of the RE and deviation bound (DB) conditions for the local screening step, see Theorems 3 and 4.

The remainder of the article is organized as follows. Section 2 introduces the modeling framework, while Section 3 provides a detailed description of the proposed methodology based on BSS. Asymptotic properties of the BSS method including the consistency of the number of break points and their locations are established in Section 4, while the computational complexity of BSS is discussed in Section 4.1. The numerical performance of the proposed BSS in various simulation settings together with a real data application are presented in Sections 5 and 6, respectively.

2. Model Formulation

We start by introducing a piecewise stationary VAR(q) model exhibiting several break points. This model comprises of independent stationary VAR(q) processes concatenated at certain time points, henceforth called *break points*. This modeling framework is similar to the one developed in Safikhani and Shojaie (2020); see also model 1 in Wang et al. (2019).

Specifically, suppose there exist m_0 break points $0 < t_1 < \cdots t_{m_0} < T$ (with $t_0 = 0$ and $t_{m_0+1} = T$) in such a way that for $t_{j-1} \le t < t_j$, we have:

$$y_t = \Phi^{(1,j)} y_{t-1} + \cdots + \Phi^{(q,j)} y_{t-q} + \Sigma_j^{1/2} \epsilon_t, \qquad (1)$$

for $j = 1, 2, \ldots, m_0 + 1$, where y_t is a p-dimensional vector of observations at time t, $\Phi^{(l,j)} \in \mathbb{R}^{p\times p}$ is a sparse coefficient matrix corresponding to the lth lag of a VAR process of order q during the jth stationary segment, and ϵ_t is a white-noise process with zero mean and variance matrix Σ_j (see additional discussion on distributional assumptions in Section 4). In each segment $[t_{j-1}, t_j)$, all model parameters are assumed to be fixed. However, the auto-regressive (AR) parameters $\Phi^{(l,j)}$ will change values between segments. The error covariance is assumed to be $\Sigma_j = \sigma^2 I$ across all segments (similar to Wang et al. 2019; see definition 1 and model 1), since segment-specific covariance structure for the error terms may introduce nontrivial identifiability issues. Specifically, the latter choice may lead to identical second order structure of the stochastic processes involved before and after a break point through simultaneous changes in both the transition matrices and the covariance of the error term.

In this setup, the number of break points m_0, their locations $t_j, j = 1, \ldots, m_0$, the VAR parameters $\Phi^{(q,j)}$ together with the covariance matrix are unknown in each segment. The objective is then to detect the break points t_j, in a computationally efficient manner that is also scalable for very large values of T. Of interest is also to estimate accurately the VAR parameters $\Phi^{(l,j)}$, under a high-dimensional regime ($p \gg T$).

Notation: Denoting $\Phi^{(\cdot,j)} = \left(\Phi^{(1,j)} \ldots \Phi^{(q,j)}\right) \in \mathbb{R}^{p\times pq}$, define the number of nonzero elements in the kth row of $\Phi^{(\cdot,j)}$ as d_{kj}, $k = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, m_0 + 1$. Further, for each $j = 1, 2, \ldots, m_0 + 1$ and $k = 1, \ldots, p$, denote by $\mathcal{I}_{kj}$ the set of all column indexes of $\Phi_k^{(\cdot,j)}$ at which there is a nonzero term, where $\Phi_k^{(\cdot,j)}$ denotes the kth row of $\Phi^{(\cdot,j)}$. Let $\mathcal{I}_j = \cup_k \mathcal{I}_{kj}$ and $d_j = \sum_{k=1}^p d_{kj}$. Let $d_n^\star = \max_{1\le j\le m_0+1} d_j$ be the maximum sparsity of the model among $m_0 + 1$ segments.

Note that our theoretical analysis deals with the high-dimensional case, wherein p, m_0 and the sparsity levels d_{kj} increase with the sample size, T. Specifically, we define $p \equiv p(n)$ and $m_0 \equiv m_0(n)$ and $d_{kj} \equiv d_{kj}(n)$, where $n = T - q + 1$, and we use the suppressed n-index throughout the article. In addition, we denote the transpose of a matrix A as A', denote $|\mathcal{S}|$ as the cardinal of a set $\mathcal{S}$. For a vector $v \in \mathbb{R}^p$, we use $||v||_1, ||v||_2, ||v||_\infty$ to represent ℓ_1, ℓ_2, and ℓ_∞ norm, respectively. We use $||A||_1$, $||A||_F$ and $||A||_\infty$ to represent $\sum_{ij} |A_{ij}|$, Frobenius norm of matrix A and $\max_{ij} |A_{ij}|$, respectively. We also denote the minimum distance between two consecutive break points by $\Delta_n = \min_{1\le j\le m_0+1} |t_j - t_{j-1}|$.

3. A BSS-Based Algorithm

The main idea of BSS is to partition the time points into blocks of size b_n and fix the VAR parameters within each block. To this end, define a sequence of time points $q = r_0 < r_1 < \ldots < r_{k_n} = T + 1$ which play the role of end points for the blocks; that is, $r_{i+1} - r_i = b_n$ for $i = 0, \ldots, k_n - 2$, and $k_n = \lceil \frac{n}{b_n} \rceil$ is the total number of blocks. Next, we form the following linear regression:

$$\underbrace{\begin{pmatrix} y_q' \\ \vdots \\ y_{r_1-1}' \\ y_{r_1}' \\ \vdots \\ y_{r_2-1}' \\ \vdots \\ y_{r_{k_n-1}}' \\ \vdots \\ y_T' \end{pmatrix}}_{\mathcal{Y}} = \underbrace{\begin{pmatrix} Y_{q-1}' & & & \\ \vdots & 0 & \ldots & 0 \\ Y_{r_1-2}' & & & \\ Y_{r_1-1}' & Y_{r_1-1}' & & \\ \vdots & \vdots & \ldots & 0 \\ Y_{r_2-2}' & Y_{r_2-2}' & & \\ \vdots & \vdots & \ddots & \vdots \\ Y_{r_{k_n-1}-1}' & Y_{r_{k_n-1}-1}' & & Y_{r_{k_n-1}-1}' \\ \vdots & \vdots & \ldots & \vdots \\ Y_{T-1}' & Y_{T-1}' & & Y_{T-1}' \end{pmatrix}}_{\mathcal{X}} \underbrace{\begin{pmatrix} \theta_1' \\ \theta_2' \\ \vdots \\ \theta_{k_n}' \end{pmatrix}}_{\Theta} + \underbrace{\begin{pmatrix} \varepsilon_q' \\ \vdots \\ \varepsilon_{r_1-1}' \\ \varepsilon_{r_1}' \\ \vdots \\ \varepsilon_{r_2-1}' \\ \vdots \\ \varepsilon_{r_{k_n-1}}' \\ \vdots \\ \varepsilon_T' \end{pmatrix}}_{E}, \qquad (2)$$

where $Y_l' = \left(y_l' \ldots y_{l-q+1}'\right)_{1\times pq}$, $\mathcal{Y} \in \mathbb{R}^{n\times p}$, $\mathcal{X} \in \mathbb{R}^{n\times k_n pq}$, $\Theta \in \mathbb{R}^{k_n pq\times p}$ and $E \in \mathbb{R}^{n\times p}$. Note that in this parameterization, $\theta_i \neq 0$ for $i \ge 2$ implies a change in the VAR coefficients. Therefore, for $j = 1, \ldots, m_0$, the structural break points t_j can be estimated as block-end time point r_{i-1}, where $i \ge 2$ and $\theta_i \neq 0$. We can rewrite the linear regression model (2) in vector form as

$$\mathbf{Y} = \mathbf{Z}\boldsymbol{\Theta} + \mathbf{E}, \qquad (3)$$

where $\mathbf{Y} = \text{vec}(\mathcal{Y}) \in \mathbb{R}^{np\times 1}$, $\mathbf{Z} = I_p \otimes \mathcal{X} \in \mathbb{R}^{np\times \pi_b}$, $\boldsymbol{\Theta} = \text{vec}(\Theta) \in \mathbb{R}^{\pi_b\times 1}$ and $\mathbf{E} = \text{vec}(E) \in \mathbb{R}^{np\times 1}$, with $\otimes$ denoting the tensor product of two matrices and $\pi_b = k_n p^2 q$.

The model parameters $\boldsymbol{\Theta}$ can be estimated via regularized least squares. We introduce two ℓ_1 penalty terms to handle the growing number of nonzero parameters due to the number of break points m_0, as well as the number of time series p. The

initial estimate of parameter $\boldsymbol{\Theta}$ is given by

$$\widehat{\boldsymbol{\Theta}} = \text{argmin}_{\boldsymbol{\Theta}} \frac{1}{n}||\mathbf{Y} - \mathbf{Z}\boldsymbol{\Theta}||_2^2 + \lambda_{1,n}||\boldsymbol{\Theta}||_1 + \lambda_{2,n}\sum_{i=1}^{k_n}\left\|\sum_{j=1}^{i}\theta_j\right\|_1. \quad (4)$$

Problem (4) uses a fused lasso penalty (Tibshirani et al. 2005), with two ℓ_1 penalties controlling the number of break points and the sparsity of the VAR model. This problem is convex and can be solved efficiently with available algorithms. Asymptotic results of this estimator are established in Theorem 1 in Section 4.

Denote the sets of indices of blocks with nonzero jumps and corresponding estimated change points obtained from solving Equation (4) by

$$\widehat{I}_n = \left\{\widehat{i}_1, \widehat{i}_2, \ldots, \widehat{i}_{\widehat{m}}\right\} = \left\{i : \left\|\widehat{\boldsymbol{\Theta}}_i\right\|_F^2 \neq 0,\ i = 2, \ldots, k_n\right\},$$

and

$$\widehat{\mathcal{A}}_n = \left\{\widehat{t}_1, \widehat{t}_2, \ldots, \widehat{t}_{\widehat{m}}\right\} = \left\{r_{i-1} : i \in \widehat{I}_n\right\}.$$

The total number of estimated change points in this step corresponds to the cardinality of the set $\widehat{\mathcal{A}}_n$; let $\widehat{m} = |\widehat{\mathcal{A}}_n|$. Then, the relationship between $\widehat{\theta}_j$'s and $\widehat{\Phi}_j$'s is given by

$$\widehat{\Phi}_1 = \widehat{\theta}_1, \qquad \text{and} \qquad \widehat{\Phi}_j = \sum_{k=1}^{\widehat{i}_j}\widehat{\theta}_k, \qquad j = 2, \ldots, \widehat{m}, \quad (5)$$

where $\left\{\widehat{\theta}_k, k = 1, \ldots, k_n\right\}$ are matrix form parameters estimated from Equation (4).

Note that the block size b_n acts as a tuning parameter that *regulates* the number of model parameters to be estimated, given by $\pi_b = \lceil \frac{n}{b_n} \rceil p^2 q$. In the extreme case with $b_n = 1$, BSS reverts to an exhaustive search of all time points to locate the structural breaks. Nevertheless, b_n cannot also be too large. In Section 4 (Assumption A3), we provide conditions that b_n needs to satisfy.

Local screening step: The set $\widehat{\mathcal{A}}_n$ of candidate change points overestimates their number, as the result of Theorem 1 shows. To that end, a screening step to "thin out" redundant break points is needed. The main idea is to estimate the VAR parameters *locally* on the left and right side of each selected break point in the first step and compare them to one VAR parameter estimated from combining the left and right of the selected break point as one large stationary segment. Then, the sum of squared errors is calculated on each segment. Now, if the selected break point is close to a true break point, the sum of squared errors calculated assuming stationarity around the true break point will be much larger compared to the sum of squared errors calculated from two separate VAR parameter estimates on the left and right of the selected break point. Therefore, we can get consistent estimates of the number of break points by minimizing a localized information criterion (LIC) comprising of the sum of squared errors and a penalty term on the number of break points. Next, the localized screening step is formally defined.

Recall that $\widehat{\mathcal{A}}_n = \left\{\widehat{t}_1, \ldots, \widehat{t}_{\widehat{m}}\right\}$ is the set of candidate break points selected in the first step in Equation (4). Then, for each subset $A \subseteq \widehat{\mathcal{A}}_n$, we define the following local VAR parameter estimates: if $\widehat{t}_i \in A$, then

$$\widehat{\psi}_{\widehat{t}_i,1} = \text{argmin}_{\psi_{\widehat{t}_i,1}}\left\{\frac{1}{a_n}\sum_{t=\widehat{t}_i-a_n}^{\widehat{t}_i-1}\left\|y_t - \psi_{\widehat{t}_i,1}Y_{t-1}\right\|_2^2 + \eta_{\widehat{t}_i,1}||\psi_{\widehat{t}_i,1}||_1\right\}, \quad (6)$$

$$\widehat{\psi}_{\widehat{t}_i,2} = \text{argmin}_{\psi_{\widehat{t}_i,2}}\left\{\frac{1}{a_n}\sum_{t=\widehat{t}_i}^{\widehat{t}_i+a_n-1}\left\|y_t - \psi_{\widehat{t}_i,2}Y_{t-1}\right\|_2^2 + \eta_{\widehat{t}_i,2}||\psi_{\widehat{t}_i,2}||_1\right\}. \quad (7)$$

If $\widehat{t}_i \in \widehat{\mathcal{A}}_n \backslash A$, then

$$\widehat{\psi}_{\widehat{t}_i} = \text{argmin}_{\psi_{\widehat{t}_i}}\left\{\frac{1}{2a_n}\sum_{t=\widehat{t}_i-a_n}^{\widehat{t}_i+a_n-1}\left\|y_t - \psi_{\widehat{t}_i}Y_{t-1}\right\|_2^2 + \eta_{\widehat{t}_i}||\psi_{\widehat{t}_i}||_1\right\}, \quad (8)$$

where $\eta_{\widehat{t}_i,1}$ and $\eta_{\widehat{t}_i,2}$ are the tuning parameters for the left and right side of $\widehat{t}_i$, respectively, when $\widehat{t}_i \in A$. If $\widehat{t}_i \in \widehat{\mathcal{A}}_n \backslash A$, then there is only one tuning parameter which is denoted by $\eta_{\widehat{t}_i}$. Note that the dimension of the VAR parameter estimate $\widehat{\Psi}_A$ depends on the size of A, that is, $\widehat{\Psi}_A \in \mathbb{R}^{p\times(pq(2|A|+(\widehat{m}-|A|)))} = \mathbb{R}^{p\times(pq(|A|+\widehat{m}))}$. Also, a_n is the neighborhood size in which the VAR parameters are estimated. Now, the LIC can be defined as follows:

$$\begin{aligned}\text{LIC}(A;\eta_n) &= \left\{\sum_{\widehat{t}_i\in A}\left(\sum_{t=\widehat{t}_i-a_n}^{\widehat{t}_i-1}\left\|y_t - \widehat{\psi}_{\widehat{t}_i,1}Y_{t-1}\right\|_2^2 + \sum_{t=\widehat{t}_i}^{\widehat{t}_i+a_n-1}\left\|y_t - \widehat{\psi}_{\widehat{t}_i,2}Y_{t-1}\right\|_2^2\right)\right.\\ &\quad \left. + \sum_{\widehat{t}_i\in\widehat{\mathcal{A}}_n\backslash A}\sum_{t=\widehat{t}_i-a_n}^{\widehat{t}_i+a_n-1}\left\|y_t - \widehat{\psi}_{\widehat{t}_i}Y_{t-1}\right\|_2^2\right\} + |A|\,\omega_n\\ &\stackrel{\text{def}}{=} L_n(A;\eta_n) + |A|\,\omega_n, \end{aligned} \quad (9)$$

and

$$(\widetilde{m}, \widetilde{t}_j; j = 1, \ldots, \widetilde{m}) = \text{argmin}_{0\le m\le\widehat{m},\,\mathbf{s}=(s_1,\ldots,s_m)\subseteq\widehat{\mathcal{A}}_n}\text{LIC}(\mathbf{s};\eta_n). \quad (10)$$

Denote the set of selected break points from Equation (10) by

$$\widetilde{\mathcal{A}}_n = \{\widetilde{t}_1, \ldots, \widetilde{t}_{\widetilde{m}}\}.$$

Remark 1. The LIC needs $\widehat{m}$ time parameter estimation on segments of size $2a_n$ which is much smaller than the total sample size n. Further, these $\widehat{m}$ time parameter estimates are independent of each other, and therefore, can be calculated in parallel.

Exhaustive Search Step. The LIC manages to eliminate candidate break points that are located far from any true break points. In other words, all selected break points $\widetilde{t}_1, \ldots, \widetilde{t}_{\widetilde{m}}$ are close enough to true break points, with the distance being at most a_n. However, in a_n-neighborhoods of each true break point, there may be more than one estimated candidate break points remaining in the set $\widetilde{\mathcal{A}}_n = \{\widetilde{t}_1, \ldots, \widetilde{t}_{\widetilde{m}}\}$. Therefore, examining the number of clusters in $\widetilde{\mathcal{A}}_n$ with sizes at most $2a_n$, leads to detection of the true number of break points. The last step involves carefully analyzing each cluster and only keeping one element in each of them. The latter task can be accomplished

by employing an exhaustive search for each cluster, which is computationally inexpensive, since the cluster sizes are at most $2a_n$. To this end, we formally state the exhaustive search step in the BSS algorithm.

For a set $A \subset \{1,\ldots,T\}$, define cluster (A,x) to be the minimal partition of A, where the diameter for each subset is at most x (for a set B, the diameter of B is defined as $\text{diam}(B) = \max_{a,b\in B}|a-b|$). Now, denote the subsets in cluster $\left(\widetilde{\mathcal{A}}_n, 2a_n\right)$ by cluster $\left(\widetilde{\mathcal{A}}_n, 2a_n\right) = \left\{B_1,\ldots,B_{m_0}\right\}$, where each subset B_i has a diameter at most $2a_n$. Note that based on Theorem 2 in Section 4, with high probability converging to one, the number of subsets in cluster $\left(\widetilde{\mathcal{A}}_n, 2a_n\right)$ is exactly m_0.

For each subset B_i, we apply the exhaustive search method for each time point s in the interval $[l_i,u_i] = [\min(B_i)-a_n, \max(B_i)+a_n]$. Specifically, define the final estimated break point $\widetilde{t}_i^f$ as

$$\widetilde{t}_i^f = \arg\min_{s\in(l_i,u_i)}\left\{\sum_{t=\min(B_i)-a_n}^{s-1}\left\|y_t-\widetilde{\psi}_{i,1}Y_{t-1}\right\|_2^2 + \sum_{t=s}^{\max(B_i)+a_n-1}\left\|y_t-\widetilde{\psi}_{i,2}Y_{t-1}\right\|_2^2\right\}, \tag{11}$$

for $i = 1,\ldots,m_0$, where $\widetilde{\psi}_{i,1}$ and $\widetilde{\psi}_{i,2}$ are the local VAR parameter estimates within the $\widetilde{R}_n$-radius interval of time point $s_i = \text{median}(B_i)$, that is,

$$\widetilde{\psi}_{i,1} = \text{argmin}_{\psi_{i,1}}\left\{\frac{1}{\widetilde{R}_n}\sum_{t=s_i-\widetilde{R}_n}^{s_i-1}||y_t-\psi_{i,1}Y_{t-1}||_2^2+\widetilde{\eta}_{i,1}||\psi_{i,1}||_1\right\}, \tag{12}$$

$$\widetilde{\psi}_{i,2} = \text{argmin}_{\psi_{i,2}}\left\{\frac{1}{\widetilde{R}_n}\sum_{t=s_i}^{s_i+\widetilde{R}_n-1}||y_t-\psi_{i,2}Y_{t-1}||_2^2+\widetilde{\eta}_{i,2}||\psi_{i,2}||_1\right\}, \tag{13}$$

where $\widetilde{\eta}_{i,1}$ and $\widetilde{\eta}_{i,2}$ are the tuning parameters for time point $s_i = \text{median}(B_i)$, and $\widetilde{R}_n$ is a carefully chosen sequence (see Assumption A4 for rates of $\widetilde{R}_n$). Denote the set of final estimated change points from (11) by $\widetilde{\mathcal{A}}_n^f = \left\{\widetilde{t}_1^f,\ldots,\widetilde{t}_{m_0}^f\right\}$.

Remark 2. The theoretical rate for $\widetilde{R}_n$ is provided in Assumption A4 in Section 4. To obtain theoretical guarantees (estimation consistency), we re-estimate the parameters within $\widetilde{R}_n$-radius of median of clusters B_i, which is slightly more than a_n. In practice, however, we can just use the local AR parameter estimates in the second step, that is, from Equations (6) and (7) to avoid increasing computation time. These local AR estimated parameters perform very well as investigated in Section 5.

Model Parameter Estimation. The key to consistent estimation of the model parameters is the result of Theorem 1 in Section 4. This result implies that removing the selected break points using a large enough R_n-radius neighborhood will also remove true break points with high probability converging to one as sample size tends to infinity. We can thus obtain stationary segments at the cost of discarding some portions of the observed time series. In other words, removing a number of data (time points) around the identified break points by the previous steps ensures that the remaining segments are stationary with high probability. Theorem 1 suggests that the radius R_n can be as small as $n\gamma_n$ (examples of γ_n include $\gamma_n = K(\log n\log p)/n$ and $\gamma_n = K\frac{\log p}{\sqrt{n}}$ for some $K>0$, see more details in Remark 3). However, based on Theorem 3 , in order not to keep any redundant break points, R_n needs to be at least $Kd_n^\star\log p$ for a large value $K>0$.

Formally, assume without loss of generality, that we have selected m_0 break points, denoted by $\widetilde{t}_1^f,\ldots,\widetilde{t}_{m_0}^f$. Then, by Theorem 3,

$$\mathbb{P}\left(\max_{1\le j\le m_0}|\widetilde{t}_j^f - t_j| \le R_n\right)\to 1,$$

as $n\to\infty$. Further, denote $r_{j1} = \widetilde{t}_j^f - R_n - 1$, $r_{j2} = \widetilde{t}_j^f + R_n + 1$ for $j = 1,\ldots,m_0$, and set $r_{02} = q$ and $r_{(m_0+1)1} = T$. Next, define the intervals $I_j = [r_{(j-1)2}, r_{j1}]$ for $j = 1,\ldots,m_0+1$. The idea is to form a linear regression on $\cup_{j=1}^{m_0+1} I_j$ and estimate the AR parameters by minimizing an ℓ_1-regularized least squares criterion. Specifically, we form the following linear regression:

$$\underbrace{\begin{pmatrix} y_q' \\ \vdots \\ y_{r_{11}}' \\ y_{r_{12}}' \\ \vdots \\ y_{r_{21}}' \\ \vdots \\ y_{r_{m_02}}' \\ \vdots \\ y_T' \end{pmatrix}}_{\mathcal{Y}_\mathbf{r}} = \underbrace{\begin{pmatrix} Y_{q-1}' & & & \\ \vdots & 0 & \cdots & 0 \\ Y_{r_{11}-1}' & & & \\ & Y_{r_{12}-1}' & & \\ 0 & \vdots & \cdots & 0 \\ & Y_{r_{21}-1}' & & \\ \vdots & \vdots & \ddots & \vdots \\ & & & Y_{r_{m_02}-1}' \\ 0 & 0 & & \vdots \\ & & & Y_{T-1}' \end{pmatrix}}_{\mathcal{X}_\mathbf{r}} \underbrace{\begin{pmatrix} \beta_1' \\ \beta_2' \\ \vdots \\ \beta_{m_0+1}' \end{pmatrix}}_{B} + \underbrace{\begin{pmatrix} \zeta_q' \\ \vdots \\ \zeta_{r_{11}}' \\ \zeta_{r_{12}}' \\ \vdots \\ \zeta_{r_{21}}' \\ \vdots \\ \zeta_{r_{m_02}}' \\ \vdots \\ \zeta_T' \end{pmatrix}}_{E_\mathbf{r}}. \tag{14}$$

This regression can be written in vector form, as

$$\mathbf{Y_r} = \mathbf{Z_r B} + \mathbf{E_r} \tag{15}$$

where $\mathbf{Y_r} = \text{vec}(\mathcal{Y}_\mathbf{r}) \in \mathbb{R}^{Np\times 1}$, $\mathbf{Z_r} = I_p \otimes \mathcal{X}_\mathbf{r} \in \mathbb{R}^{Np\times\tilde{\pi}}$, $\mathbf{B} = \text{vec}(B) \in \mathbb{R}^{\tilde{\pi}\times 1}$, $\mathbf{E_r} = \text{vec}(E_\mathbf{r}) \in \mathbb{R}^{Np\times 1}$. Here, $\mathbf{r}$ is the collection of all r_{j1} and r_{j2} for $j = 0,\ldots,m_0+1$, $\otimes$ denotes the tensor product of two matrices, $\tilde{\pi} = (m_0+1)p^2q$, $N_j = \text{length}(I_j) = r_{j1} - r_{(j-1)2}$ for $j = 1,\ldots,m_0+1$ and $N = \sum_{j=1}^{m_0+1} N_j$. We estimate the VAR parameters by solving

$$\widehat{\mathbf{B}} = \text{argmin}_\mathbf{B} N^{-1} \|\mathbf{Y_r} - \mathbf{Z_r B}\|_2^2 + \rho_n \|\mathbf{B}\|_1 . \tag{16}$$

The estimator defined in Equation (16) provides simultaneously m_0+1 estimated transition matrices over the estimated m_0+1 stationary segments which is computationally attractive since it requires tuning of a single hyperparameter (ρ_n). However, the performance of estimator $\widehat{\mathbf{B}}$ may be poor in cases where the sparsity level is unbalanced over the m_0+1 stationary segments and/or the number of change point is diverging with the sample size. In such cases, one could separately estimate the VAR parameters for each segment. Specifically, for jth segment, where $j = 1\ldots,m_0+1$, the following linear regression equation holds:

$$\underbrace{\begin{pmatrix} y'_{r_{(j-1)2}} \\ \vdots \\ y'_{r_{j1}} \end{pmatrix}}_{\mathcal{Y}_j} = \underbrace{\begin{pmatrix} Y'_{r_{(j-1)2}-1} \\ \vdots \\ Y'_{r_{j1}-1} \end{pmatrix}}_{\mathcal{X}_j} \beta'_j + \underbrace{\begin{pmatrix} \zeta'_{r_{(j-1)2}} \\ \vdots \\ \zeta'_{r_{j1}} \end{pmatrix}}_{E_j}. \tag{17}$$

where $\mathcal{Y}_j \in \mathbb{R}^{N_j\times p}$, $\mathcal{X}_j \in \mathbb{R}^{N_j\times pq}$, $\beta_j \in \mathbb{R}^{p\times pq}$ and $E_j \in \mathbb{R}^{N_j\times p}$. Now, transition matrices in the jth segment can be estimated as

$$\widetilde{\beta}_j = \text{argmin}_{\beta_j} N_j^{-1} \left\|\text{vec}(\mathcal{Y}_j) - (I_p \otimes \mathcal{X}_j)\text{vec}(\beta'_j)\right\|_2^2 + \rho_{n,j} \left\|\text{vec}(\beta'_j)\right\|_1 , \tag{18}$$

for $j = 1\ldots,m_0+1$, where $\text{vec}(\mathcal{Y}_j) \in \mathbb{R}^{N_jp\times 1}$, $I_p \otimes \mathcal{X}_j \in \mathbb{R}^{N_jp\times p^2q}$, $\text{vec}(\beta'_j) \in \mathbb{R}^{p^2q\times 1}$, and $\text{vec}(E_j) \in \mathbb{R}^{N_jp\times 1}$.

4. Consistency of the BSS Estimator

We start by stating the assumptions needed to establish properties of the BSS-based estimator.

A1. For each $j = 1,2,\ldots,m_0+1$, the process $y_t^{(j)} = \Phi^{(1,j)}y_{t-1}^{(j)} + \cdots + \Phi^{(q,j)}y_{t-q}^{(j)} + \Sigma_j^{1/2}\varepsilon_t$ is a stationary Gaussian time series. Denote the covariance matrices $\Gamma_j(h) = \text{cov}\left(y_t^{(j)}, y_{t+h}^{(j)}\right)$ for $t,h \in \mathbb{Z}$. Also, assume that for $\kappa \in [-\pi,\pi]$, the spectral density matrices $f_j(\kappa) = (2\pi)^{-1}\sum_{l\in\mathbb{Z}}\Gamma_j(l)e^{-\sqrt{-1}\kappa l}$ exist; further

$$\max_{1\le j\le m_0+1} \mathcal{M}(f_j) = \max_{1\le j\le m_0+1} \left(\sup_{\kappa\in[-\pi,\pi]}\Lambda_{\max}(f_j(\kappa))\right) < +\infty,$$

and

$$\min_{1\le j\le m_0+1} \mathbf{m}(f_j) = \min_{1\le j\le m_0+1} \left(\sup_{\kappa\in[-\pi,\pi]}\Lambda_{\min}(f_j(\kappa))\right) > 0,$$

where $\Lambda_{\max}(A)$ and $\Lambda_{\min}(A)$ are the largest and smallest eigenvalues of the symmetric or Hermitian matrix A, respectively.

A2. The matrices $\Phi^{(.,j)}$ are sparse. Specifically, for all $k = 1,2,\ldots,p$ and $j = 1,2,\ldots,m_0$, $d_{kj} \ll p$, that is, $d_{kj}/p = o(1)$. Moreover, there exists a positive constant $M_\Phi > 0$ such that

$$\max_{1\le j\le m_0+1} \left\|\Phi^{(.,j)}\right\|_\infty \le M_\Phi.$$

A3. There exists a positive constant ν such that

$$\min_{1\le j\le m_0} \left\|\Phi^{(.,j+1)} - \Phi^{(.,j)}\right\|_F \ge \nu > 0.$$

Moreover, there exists a vanishing positive sequence γ_n such that, as $n\to\infty$,

$$\frac{\Delta_n}{n\gamma_n} \to +\infty,\ \text{limsup}\frac{b_n}{n\gamma_n} \le C < 1/12,\ \text{and}$$
$$d_n^\star\sqrt{\frac{\log p}{n\gamma_n}} \to 0.$$

Assumption A1 is standard for sparse VAR models (see Basu and Michailidis 2015) and allows us to obtain necessary concentration inequalities in high dimensions. This assumption does not restrict the applicability of the method, since it holds for large families of VAR models (Basu and Michailidis 2015). Note that the Gaussian assumption could be relaxed to sub-Gaussian or sub-Weibull distributional assumptions as long as the RE and DB conditions hold (Loh and Wainwright 2012). In Wong et al. (2020), it is verified that these two conditions hold for a large family of sparse VAR models under certain mixing conditions. The second part of A1 is also used in the proof of consistency of the VAR model parameters, once the break points are detected. Assumption A2 ensures all transition matrices are sparse which is a common assumption in high-dimensional VAR models (Basu and Michailidis 2015), while it controls the magnitudes of elements in all transition matrices as well. The sequence γ_n in Assumption A3 is directly related to the consistency rate for locating the break points t_j, where $j = 1,\ldots,m_0$. Assumption A3 connects this rate to the tuning parameter chosen in the estimation procedure and also to the block sizes. Also, this assumption puts a minimum distance-type requirement on the coefficients in different segments. Note that the jump sizes $\left\|\Phi^{(.,j+1)} - \Phi^{(.,j)}\right\|_F$ can potentially converge to zero at the price of worsening the consistency rate for locating the break points (see more details in Remark 5). Assumption A3 can be regarded as the extension of Assumption H2 in Chan, Yau, and Zhang (2014) for univariate time series to the high-dimensional case. Note that the last part of Assumption A3 puts an upper bound on the block length, b_n and shows its connection to the sequence γ_n. More details are provided in the sequel. Note that in the case in which the locations of the break points are known, the total sparsity should satisfy $d_n^\star\sqrt{\frac{\log p}{\Delta_n}} \to 0$, since $d_n^\star$ is the maximum sparsity over all stationary segments, see, for example, Basu and Michailidis (2015). However, since in our setting, these locations must be estimated from data, the detection/estimation error of the algorithm should be accounted for, which yields a slightly stronger condition as stated in Assumption A3, that is, $d_n^\star\sqrt{\frac{\log p}{n\gamma_n}} \to 0$.

The next result establishes that the number of selected change points, $\widehat{m}$, based on Equation (4) will be at least as large as the true number, m_0. Moreover, there exists at least one estimated change point in a $n\gamma_n$-radius neighborhood of *each* true change point. Before stating the theorem, we introduce some additional notation. Let $\mathcal{A}_n = \{t_1, t_2, \ldots, t_{m_0}\}$ be the set of true change points. Following Boysen et al. (2009) and Chan, Yau, and Zhang (2014), define the Hausdorff distance between two countable sets on the real line as

$$d_H(A, B) = \max_{b\in B} \min_{a\in A} |b - a|.$$

Note that the above definition is not symmetric and therefore not a real distance. Nevertheless, this is the version of function $d_H(A, B)$ used in the next theorem.

Theorem 1. Suppose A1–A3 hold. Choose $\lambda_{1,n} = 2C_1\sqrt{\frac{\log(n)+2\log(p)+\log(q)}{n}}$, and $\lambda_{2,n} = C_2\frac{b_n}{n}\sqrt{\log p}$ for some large constants $C_1, C_2 > 0$. Then, as $n \to +\infty$,

$$\mathbb{P}\left(\left|\widehat{\mathcal{A}}_n\right| \geq m_0\right) \to 1,$$

and

$$\mathbb{P}\left(d_H\left(\widehat{\mathcal{A}}_n, \mathcal{A}_n\right) \leq n\gamma_n\right) \to 1.$$

In this theorem, the first tuning parameter could be as large as $\lambda_{1,n} = O\left(\sqrt{\frac{\gamma_n \log p}{n}}\right)$. The consistency rate for break point detection in Theorem 1 is $n\gamma_n$, which can be chosen as small as possible assuming that Assumptions A2 and A3 hold. Note that γ_n also depends both on the minimum distance between consecutive true break points, as well as the number of time series p. When m_0 is finite, one can choose $\gamma_n = (\log n \log p)/n$. This implies that the convergence rate for estimating the relative locations of the break points, that is, t_j/T using $\widehat{t}_j/T$, could be as low as $(\log n \log p)/n$.

Remark 3. Based on Assumption A3, there is a connection between the consistency rate $n\gamma_n$ and the block size b_n. For the choice of $\gamma_n = K(\log n \log p)/n$ for some $K > 0$, b_n can be as large as $\log n \log p$. If we restrict the minimum distance between consecutive break points to be at least $\left(\sqrt{n}\log p\right)^{1+\epsilon}$, then one could choose $\gamma_n = K\frac{\log p}{\sqrt{n}}$ and $b_n = \sqrt{n}\log p$. Therefore, there is a tradeoff between computational gains by BSS and the distance between consecutive true break points.

To establish the consistency of the screening procedure (10), we require two additional assumptions. Recall that $d_n^\star = \max_{1\leq j\leq m_0+1} d_j$ denotes the maximum sparsity of the model among $m_0 + 1$ segments.

A4. Let $\Delta_n = \min_{1\leq j\leq m_0} |t_{j+1} - t_j|$. Then, $\omega_n = n\gamma_n d_n^{\star 3}$ and $\omega_n/a_n \to 0$. Also, $\frac{\Delta_n}{4} \geq \widetilde{R}_n = \frac{d_n^{\star 2} a_n^2}{\log p}$.

A5. There exists a large positive constant $c > 0$ such that (a) if there exists one true break point t_i in the interval $\left(\widehat{t}_j - a_n, \widehat{t}_j + a_n\right)$ such that $|\widehat{t}_j - t_i| \leq Kn\gamma_n$ for some positive constant K, then $\eta_{\widehat{t}_j} - cd_n^\star = \eta_{\widehat{t}_j,1} = \eta_{\widehat{t}_j,2} = c\sqrt{\frac{d_n^\star n\gamma_n}{a_n}}$; (b) if there exists no true break point in the interval $\left(\widehat{t}_j - a_n, \widehat{t}_j + a_n\right)$, then $\eta_{\widehat{t}_j} = \eta_{\widehat{t}_j,1} = \eta_{\widehat{t}_j,2} = c\sqrt{\frac{\log p}{a_n}}$.

Assumption A4 essentially puts a lower bound on the minimum spacing between consecutive break points (equivalently an upper bound on the number of true break points allowed, i.e., m_0) and connects it with the penalty term ω_n in the local screening step. Further, Assumption A5 states sufficient conditions on the rate of tuning parameters in the local screening step in order to reach optimal consistency rate for locating the true break points. This specific selection of tuning parameters are mainly due to the fact that in the presence of break points, one works with misspecified models and hence a more careful and complex selection of the various tuning parameters are required (Chan et al. 2017; Roy, Atchadé, and Michailidis 2017). Although the tuning parameters under this assumption are segment-specific, this assumption can be relaxed by putting universal rates of tuning parameters at the cost of worsening the consistency rates as discussed in Safikhani and Shojaie (2020).

Recall that the selected break points after the local screening step are denoted by $\widetilde{\mathcal{A}}_n = \{\widetilde{t}_1, \ldots, \widetilde{t}_{\widetilde{m}}\}$. Further, recall that for a set $A \subset \{1, \ldots, T\}$, we define cluster (A, x) to be the minimal partition of A, where the diameter for each subset is at most x (for a set B, the diameter of B is defined as $\text{diam}(B) = \max_{a,b\in B} |a - b|$). Next, we formally state the result for the set $\widetilde{\mathcal{A}}_n$. The next theorem establishes that the number of clusters obtained in the LIC screening step are consistent, despite the fact that the total number of estimated break points can be larger than the true number of break points.

Theorem 2. Suppose A1–A5 hold. Then, as $n \to +\infty$, the minimizer $(\widetilde{m}, \widetilde{t}_j, j = 1, \ldots, \widetilde{m})$ of (10) satisfies

$$\mathbb{P}\left(\widetilde{m} \geq m_0, \left|\text{cluster}\left(\widetilde{\mathcal{A}}_n, 2a_n\right)\right| = m_0\right) \to 1.$$

Moreover,

$$\mathbb{P}\left(d_H\left(\widetilde{\mathcal{A}}_n, \mathcal{A}_n\right) \leq n\gamma_n, \text{ and } d_H\left(\mathcal{A}_n, \widetilde{\mathcal{A}}_n\right) \leq a_n\right) \to 1.$$

Despite the fact that Theorem 2 does not guarantee consistency of the number of break points, it exhibits two advantages compared to Theorem 1: (i) one can estimate consistently the number of break points by looking at the cardinality of cluster $\left(\widetilde{\mathcal{A}}_n, 2a_n\right)$; (ii) all the remaining estimated break points in $\widetilde{\mathcal{A}}_n$ are within an a_n-neighborhood of at least one true break point. These advantages are used in the final step of our procedure (exhaustive search) in which we consistently estimate both the number of break points and their locations. As previously explained, the exhaustive search step reduces to employing the prediction error to each subset in cluster $\left(\widetilde{\mathcal{A}}_n, 2a_n\right)$ in order to remove any additional break points within each cluster and only select one.

The next theorem establishes that the estimated locations of the break points obtained through the exhaustive search step are consistent.

Theorem 3. Suppose A1–A5 hold and $\widetilde{\eta}_{j,1} = \widetilde{\eta}_{j,2} = c\sqrt{\frac{\log p}{\widetilde{R}_n}}$ in (12) and (13) with a large enough constant $c > 0$ for $j = 1, \ldots, m_0$. Then, as $n \to +\infty$, there exists a large enough constant $K > 0$ such that

$$\mathbb{P}\left(\max_{1\leq j\leq m_0} \left|\widetilde{t}_j^f - t_j\right| \leq Kd_n^\star \log p\right) \to 1.$$

Remark 4. Theorem 3 shows that the BSS method achieves a better consistency rate in terms of the localization error than the DP method developed in Wang et al. (2019)—as shown in Theorem 1 of Wang et al. (2019), which is $O_p\left(d_n^{\star 2}\log p\right)$—and the three-step procedure (TSP) method developed in Safikhani and Shojaie (2020)—as shown in Theorem 3 of Safikhani and Shojaie (2020), which is $O_p\left(m_0 d_n^{\star 2}\log p\right)$—while it matches the consistency rate of DP after post-processing group Lasso (PGL) procedure (see Wang et al. 2019, theor. 2).

Remark 5. In Assumption A3, it is possible to relax the assumption by allowing the jump sizes to vanish as a function of the sample size, at the cost of worsening the consistency rates. In fact, the consistency rate in Theorem 3 is of order $Kd_n^{\star}\log p/\min_{1\le j\le m_0}\left\|\Phi^{(.,j+1)}-\Phi^{(.,j)}\right\|_F^2$. In other words, the reciprocal of the squared of the minimum jump sizes for AR parameters appears in the consistency rate for locating the break points. The new consistency rate depends on how fast this quantity vanishes. A similar role for a vanishing jump size appeared in Wang et al. (2019); Kaul, Jandhyala, and Fotopoulos (2019).

Remark 6. The minimum spacing required between consecutive break points (Δ_n) is the bottleneck of all detection procedures. In the proposed BSS method, there is an important connection between the block sizes b_n and Δ_n as stated in Assumption A3. Specifically, the assumption is that $b_n/\Delta_n \to 0$ at a certain rate when the sample size tends to $+\infty$. For example, for the choice of $b_n = O\left(n^{1/3}\right)$, Δ_n must be of order $n^{\frac{2}{3}+\epsilon}$ for some small positive ϵ based on Assumption A4. Although this assumption may seem strong, it is nevertheless weaker than many existing detection methods in the literature including the SBS (Cho and Fryzlewicz 2015) and DCBS methods (Cho 2016), wherein Δ_n must be of order n^{ψ} for some $\psi \in (6/7, 1)$. Note that the minimum spacing assumption for the BSS method is stronger than the detection methods developed in Wang et al. (2019) and Safikhani and Shojaie (2020), and reflects a tradeoff between a sub-linear break point detection algorithm and the corresponding minimum spacing allowed.

Finally, after removing data points in an appropriately size neighborhood of the estimated break points, consistent estimation of the VAR model parameters is achieved, as stated in the following theorem.

Theorem 4. Suppose A1–A5 hold and m_0 is unknown and $R_n = a_n$. Assume also that $\Delta_n > \varepsilon n$ for some large positive $\varepsilon > 0$ and $\rho_n = C\sqrt{\frac{\log\tilde{\pi}}{N}}$ for large enough $C > 0$. (Note that $N/n = O(1)$.) Then, as $n \to +\infty$, the minimizer $\widehat{\mathbf{B}}$ of Equation (16) satisfies

$$\left\|\widehat{\mathbf{B}} - \Phi\right\|_\ell = O_p\left((d_n^{\star})^{1/\ell}\rho_n\right) \text{ for } \ell = 1, 2.$$

Theorem 4 verifies that the estimator $\widehat{\mathbf{B}}$ achieves the same consistency rate in the case of stationary sparse VAR models (Basu and Michailidis 2015) as long as $\Delta_n > \varepsilon n$ for some large positive $\varepsilon > 0$, which is equivalent of assuming a finite number of break points. In the case of diverging m_0, Corollary 1 states that separate estimation of model parameters in each segment, that is, $\widetilde{\boldsymbol{\beta}}_j$, for $j = 1, 2, \ldots, m_0+1$, achieves a similar consistency rate.

Corollary 1. Suppose A1–A5 hold and $R_n = a_n$. Assume that $\rho_{n,j} = C\sqrt{\frac{\log p^2 q}{N_j}}$ for large enough $C > 0$. Then, as $n \to +\infty$, the minimizer $\left\{\widetilde{\beta}_j\right\}_{j=1}^{m_0+1}$ of Equation (18) satisfies

$$\left\|\text{vec}(\widetilde{\boldsymbol{\beta}}_j) - \text{vec}(\Phi^{(.,j)})\right\|_\ell = O_p\left((d_j)^{1/\ell}\rho_{n,j}\right) \text{ for } \ell = 1, 2; j = 1, \ldots, m_0 + 1,$$

where $d_j = \sum_{k=1}^{p} d_{kj}$.

Both Theorem 4 and Corollary 1 are stated under Assumptions A1–A5 to ensure that the number and locations of detected break points are consistent, while the tuning parameter rates assumed in their respective statements are in accordance with results for stationary high-dimensional models (Basu and Michailidis 2015).

To illustrate the individual performance of each step in BSS, we report its performance based on a single replicate from the following simulation setting: $T = 50,000$, $p = 50$, $q = 1$ and $m_0 = 2$, with the break points located at $t_1 = \lfloor T/3\rfloor = 16,666$ and $t_2 = \lfloor 2T/3\rfloor = 33,333$. The autoregressive coefficients are chosen to have different sparsity patterns. In this scenario, we set $b_n = 500$. Figure 1 depicts all selected break points in the three steps of the algorithm. As seen in the upper left panel, in the first step (fused lasso) of BSS, the method over-estimates the number of break points, which confirms the suboptimality of fused lasso if used alone for break point detection. However, the true break points are not isolated, as expected from Theorem 1. For this example, around 20 points are selected as candidate break points. Some of them are not close to any true break points, which is why we need the second step in our method based on screening using the LIC. After the local screening step, only three break points remain (see plot in the upper right panel). The method still over-estimates the number of break points. However, as anticipated by Theorem 2, the number of clusters is a consistent estimate of the number of break points, which is 2 in this example. Moreover, note that after the second step, there are no selected break points far from the true break points, which confirms the second part of Theorem 2. Finally, as depicted in the bottom panel of Figure 1, after applying the last step of BSS, only two selected break points remain, and both are close to their corresponding true values.

4.1. Computational Complexity Considerations

In the following, the number of break points m_0 and the number of time series components p are assumed to be fixed and finite. An exact calculation of computation time for the BSS method is hard, due to the presence of several optimization steps within the BSS algorithm, for which closed form solutions are not available; hence, numerical approximations are needed. Further, the number of iterations to reach a small tolerance for such numerical approximations may not be known (Bleakley and Vert 2011), which makes it hard to compute exact number of

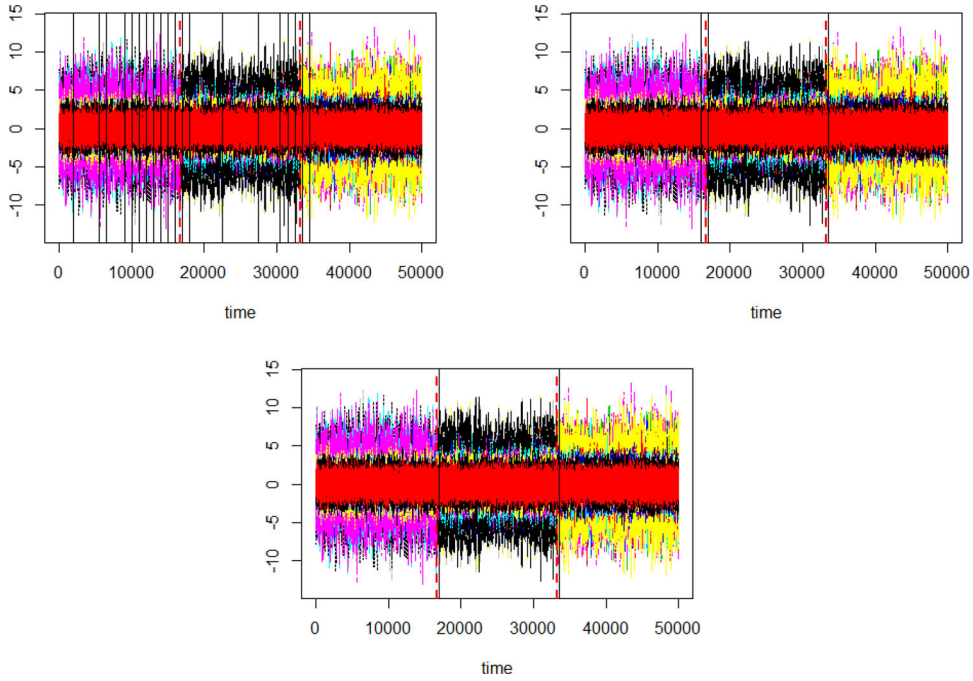

Figure 1. Upper left panel: Estimated break points from the first stage of our proposed BSS procedure (Equation (4)) for a single run under Simulation Scenario 2; ∼20 points are selected in the first stage. Upper right panel: Selected break points after the LIC screening step. Bottom panel: Final selected break points.

operations for the BSS algorithm. As a result, computational complexity calculations presented next, serve as approximations of the algorithm's overall computational complexity. Note that the computational time required in the first step of BSS (penalized regression) is of order $O(k_n)$ (Bleakley and Vert 2011). Further, the computational complexity of each candidate break point in the local screening step is of order $O(2a_n)$ (Beck and Teboulle 2009), since the calculations for each time series component can be done separately. This rate is linear with respect to the sample size a_n. Finally, the computational complexity of each candidate break point in the exhaustive search step is of order $O(4a_n)$, since the calculations for each segment can be done separately. Specifically, in the exhaustive search step, it takes $O(2a_n)$ for computing the sum of squared errors (SSE) in Equation (11) and $O(2a_n)$ for searching the break point that minimizes the SSE. Note that as mentioned in Remark 2, the local AR parameter estimates in the exhaustive search step are the ones estimated in the local screening step (Step 2), hence their computational times are not considered again in Step 3. Therefore, the total computational complexity of BSS is of order $O(k_n + a_n)$. Recall that $k_n \sim n/b_n$ and based on Assumption A4, a_n can essentially be selected as $b_n^{1+\epsilon}$ for a small positive ϵ. Thus, the computational complexity of BSS can be written in terms of b_n as $O\left(n/b_n + b_n^{1+\epsilon}\right)$. Selecting $b_n = 1$ yields to linear computational complexity, while $b_n \sim \sqrt{n}$ reaches the optimal computational complexity of $\sim O(\sqrt{n})$. Specifically, selecting $b_n = n^{\frac{1-\epsilon}{2(1+\epsilon)}}$, the computational complexity of BSS is of order $O\left(n^{\frac{1}{2}+\frac{\epsilon}{1+\epsilon}}\right)$ which for a small ϵ reaches $O(\sqrt{n})$.

It is worth noting that there is a trade-off between the minimum distance Δ_n allowed between two consecutive break points, and the computational gains of BSS. The optimal computational gains (as discussed above) occur when $b_n = n^{\frac{1-\epsilon}{2(1+\epsilon)}}$ which implies that Δ_n is at least of order $\sim n^{\frac{1-\epsilon}{(1+\epsilon)}}$ based on Assumption A5. However, this assumption is somewhat strong, and may be violated in selected real datasets. Note that this rate can be reduced at the cost of increasing computational time. Specifically, one can set $b_n = n^{\xi}$ for a small positive ξ, and obtain computational complexity $O\left(n^{\max((1+\epsilon)\xi, 1-\xi)}\right)$. Note that as long as $0 < \xi < 1/2$, the BSS method still detects the break points in *sublinear computation time* with respect to the sample size.

5. Performance Evaluation of BSS

We evaluate the performance of BSS with respect to both estimating the number of break points and also their locations. In all scenarios considered, we set the convergence tolerance to 10^{-2} for the first fused lasso step of BSS to choose candidate break points, the covariance matrix of the noise process is set to $\Sigma_\varepsilon = I_T$ and the results are averaged over 100 random replicates. All simulations are run in R version 4.0.3 on Intel E5-2698v3 processors with 4 GB of RAM per core.[1]

5.1. Simulation Scenarios

We consider different simulation settings. Different values for the sample size T, number of time series components p, AR order q, block size b_n, number of true break points m_0 and structure of AR parameters Φ are considered as summarized in Table 1. For all settings, we report the error of locations of the estimated break points and the selection rate, that is, the percentage of replicates where each break point is correctly identified. The error of the locations of the estimated break points is

[1]The R/Rcpp codes to perform the BSS algorithm are available at the author's GitHub page: *https://github.com/abolfazlsafikhani/BSS-ChangePoint-VAR*.

defined as $\text{error}_j = |\tilde{t}_j^f - t_j|, j = 1, \ldots, m_0$. The selection rate is calculated as the proportion of replicates, wherein the estimated break points by BSS are close to each of the true break points. Specifically, to compute the selection rate, a selected break point is counted as a "success" for the jth true break point, t_j, if it falls in the interval $[t_j - \frac{t_j - t_{j-1}}{5}, t_j + \frac{t_{j+1} - t_j}{5}], j = 1, \ldots, m_0$. Details of each simulation setting are provided in Table 1.

Setting A (effect of block size b_n under small T, small p case). In scenario A, $T = 500$, $p = 2$, $q = 1$, $m_0 = 2$, $t_1 = \lfloor \frac{T}{3} \rfloor$, $t_2 = \lfloor \frac{2T}{3} \rfloor$, while the AR coefficients are chosen to have a similar pattern as in Preuss, Puchstein, and Dette (2015), also depicted in the top left panel of Figure 2. The diagonal elements for three segments have magnitudes -0.8, 0.8, and -0.8, respectively. The upper right element is fixed to be 0.1. The block sizes vary across scenarios. Specifically, in scenarios A.1 to A.3, the block sizes are selected to be $b_n = 5, 10$, and 15, respectively.

Setting B (t-distributed error case). In scenario B, $T = 5000$, $p = 15, q = 1, m_0 = 2, t_1 = \lfloor \frac{T}{3} \rfloor, t_2 = \lfloor \frac{2T}{3} \rfloor$, block size $b_n = 70$ and the AR coefficients are chosen to have the same simple 1-off diagonal structure, but different magnitude -0.8, 0.8, and -0.8 as depicted in the top right panel of Figure 2. In scenario B, the error term is set to follow Student's t-distribution. The degree of freedom vary across scenarios. Specifically, in Scenarios B.1 to B.4, the degrees of freedom are set to $df = 5, 10, 15$, and ∞, respectively.

Table 1. Details of model parameters for simulation settings A-D.

Sim	T	p	AR order q	block size b_n	m_0	AR structure
A.1	500	2	1	5	2	Simple
A.2	500	2	1	10	2	Simple
A.3	500	2	1	15	2	Simple
B.1	5000	15	1	70	2	Simple
B.2	5000	15	1	70	2	Simple
B.3	5000	15	1	70	2	Simple
B.4	5000	15	1	70	2	Simple
C.1	1000	20	1	31	2	Random
C.2	1000	40	1	31	2	Random
C.3	1000	60	1	31	2	Random
D.1	5000	15	2	70	2	Random
D.2	5000	15	2	70	2	Random
D.3	5000	15	2	70	2	Random

Setting C (High-dimensional case). In scenario C, $T = 1000$, $q = 1$, $m_0 = 2$, $t_1 = \lfloor \frac{T}{3} \rfloor = 333$, $t_2 = \lfloor \frac{2T}{3} \rfloor = 666$, $b_n = \lfloor n^{\frac{1}{2}} \rfloor = 31$ and the location of nonzero AR coefficients are randomly chosen with repeated entries -0.8, 0.8, and -0.8 as illustrated in the middle panel of Figure 2. The number of time series components p varies across scenarios. Specifically, in Scenarios C.1 to C.3, it is set to $p = 20, 40$, and 60, respectively. Note that in this setting, the number of parameters are $(m_0 + 1)p^2 = 1200, 4800, 10800$, and all of them are larger than the sample size T which is why this stetting is called the high-dimensional case.

Setting D (AR lag effect). In Scenario D, $T = 5000$, $p = 15$, $q = 2$, $m_0 = 2$, $t_1 = \lfloor \frac{T}{3} \rfloor = 1666$, $t_2 = \lfloor \frac{2T}{3} \rfloor = 3333$, and $b_n = 70$, while the structure of AR coefficients are chosen to be random in both location and magnitude. All VAR parameters are depicted in the bottom panel of Figure 2. Specifically, the $\Phi^{(1)}$, $\Phi^{(3)}$, and $\Phi^{(5)}$ stand for the lag 1 AR coefficients for the three segments, respectively, while the $\Phi^{(2)}$, $\Phi^{(4)}$ and $\Phi^{(6)}$ stand for the lag 2 AR coefficients. In Scenario D.1, the values of lag 1 effect and lag 2 effect in the first segment equal to $-(0.3 + \text{unif}(0, 0.05))$ and $(0.6 + \text{unif}(0, 0.05))$, the values of lag 1 effect and lag 2 effect in the second segment equal to $(0.3 + \text{unif}(0, 0.05))$ and $-(0.6 + \text{unif}(0, 0.05))$, and the values of lag 1 effect and lag 2 effect in the third segment equal to $-(0.3 + \text{unif}(0, 0.05))$ and $(0.6 + \text{unif}(0, 0.05))$, where $\text{unif}(a, b)$ denotes the uniform distribution in the finite interval (a, b). The magnitudes of lag effects in Scenarios D.2 and D.3 are similar to D.1, only with different signs. In Scenario D.1, both lag 1 and 2 have jumps, the block size $b_n = 70$. In Scenario D.2, only lag 1 has jumps and the rest are fixed over segments, the block size $b_n = 70$. In Scenario D.3, only lag 2 has jumps and the rest are fixed over segments, the block size $b_n = 70$.

For simulation settings A–D, the mean and standard deviation for the estimates' distance from the true break point, as well as the selection rate (proportion of correctly identifying the specific break point) are reported in Table 2. The table clearly indicates that in all settings, BSS accurately detects both the number of break points, as well as their locations. The performance of the proposed BSS algorithm is robust to the changes in the AR parameters' zero/nonzero pattern, block size, presence of heavier tailed errors (consistent with results for stationary

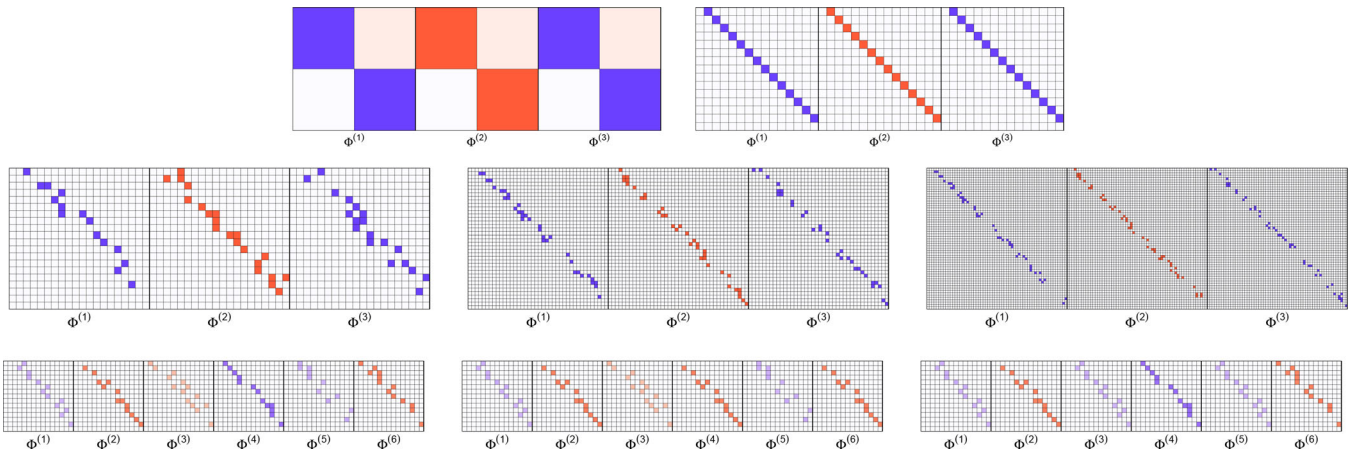

Figure 2. (Top) True AR coefficients in Scenario A (left), B (right); (Middle) True AR coefficients in Scenario C.1 (left), C.2 (middle), and C.3 (right); (Bottom) True AR coefficients in Scenario D.1 (left), D.2 (middle), and D.3 (right).

Table 2. Results of BSS performance for simulation settings A–D.

	Break point	Mean (error)	std (error)	Selection rate
Simulation A.1				
	1	2.02	3.1654	1
	2	1.69	3.2836	1
Simulation A.2				
	1	1.6667	3.5399	1
	2	1.1111	2.1471	1
Simulation A.3				
	1	0.697	1.8208	1
	2	1.5051	4.5387	0.99
Simulation B.1				
	1	0.02	0.2	1
	2	0	0	1
Simulation B.2				
	1	0	0	1
	2	0	0	1
Simulation B.3				
	1	0.01	0.1	1
	2	0.01	0.1	1
Simulation B.4				
	1	0	0	1
	2	0.01	0.1	1
Simulation C.1				
	1	0	0	1
	2	0.18	1.7019	1
Simulation C.2				
	1	1.0316	8.5645	0.94
	2	0	0	0.97
Simulation C.3				
	1	4.1837	17.8991	0.96
	2	1.0612	8.3836	0.97
Simulation D.1				
	1	0.03	0.1714	1
	2	0.02	0.1407	1
Simulation D.2				
	1	4.99	20.0809	1
	2	6.27	21.2745	1
Simulation D.3				
	1	0.05	0.219	1
	2	0.08	0.3075	1

VAR models (Lin and Michailidis 2017), increasing number of time series components, and larger number of lags. This solid performance justifies the data-driven tuning parameter methods discussed in Appendix D. As expected, the selection rate in simulation scenario A slightly decreases as the block size increases (from 100% to 99%), whereas the average computation time drops significantly ($\sim$ %74 computation time reduction using large block size compared to small block size). Note that with a decreasing block size b_n, BSS can always accommodate more break points in the model at the cost of increasing the computation time. The selection rate in simulation Scenario B is robust to changes in the degrees-of-freedom in the Student's t-distributed error. In Simulation C.3, the mean and standard deviation of distances is larger due to the small T large p setting. It is worth noting that in Simulation D.2, the mean and standard deviation of errors is slightly larger than other simulation settings. This is mainly due to the smaller jump size. As discussed in Remark 5, the method allows the jump size to vanish as a function of the sample size at the cost of worsening the consistency rates.

5.2. Comparison With Selected Competing Methods

Next, we compare the BSS method with the three-stage procedure (TSP) method in Safikhani and Shojaie (2020) and the DP method in Wang et al. (2019). This comparison is performed in two steps. First, detection accuracy of these three methods are compared based on simulation setting E. Second, computational time of these methods are compared under simulation setting F. Model settings for Scenarios E and F are summarized in Table 3. Similar to the four simulation settings A–D, we compute the selection rate for each true break point which is calculated as the proportion of replicates, where the estimated break points by each detection method are close to each of the true break points. Specifically, a selected break point is counted as a "success" for the jth true break point, t_j, if it falls in the interval $[t_j - \frac{t_{j+1}-t_j}{5}, t_j + \frac{t_{j+1}-t_j}{5}]$, $j = 1, \ldots, m_0$. Finally, in all simulations, the results are averaged over 100 replicates.

Table 3. Details of model parameters for simulation settings E and F.

Sim	T	p	AR order q	block size b_n	m_0	AR structure
E.1	1000	10	1	10, 15, 20	1	simple
E.2	1000	10	1	10, 15, 20	2	simple
E.3	1000	10	1	10, 15, 20	3	simple
E.4	1000	10	1	10, 15, 20	4	simple
E.5	1000	10	1	10, 15, 20	5	simple
E.6	1000	10	1	10, 15, 20	6	simple
F.1	200	8	1	$\lfloor n^{\frac{1}{2}} \rfloor, \lfloor n^{\frac{2}{5}} \rfloor, \lfloor n^{\frac{1}{3}} \rfloor$	1	simple
F.2	400	8	1	$\lfloor n^{\frac{1}{2}} \rfloor, \lfloor n^{\frac{2}{5}} \rfloor, \lfloor n^{\frac{1}{3}} \rfloor$	1	simple
F.3	600	8	1	$\lfloor n^{\frac{1}{2}} \rfloor, \lfloor n^{\frac{2}{5}} \rfloor, \lfloor n^{\frac{1}{3}} \rfloor$	1	simple
F.4	800	8	1	$\lfloor n^{\frac{1}{2}} \rfloor, \lfloor n^{\frac{2}{5}} \rfloor, \lfloor n^{\frac{1}{3}} \rfloor$	1	simple
F.5	1000	8	1	$\lfloor n^{\frac{1}{2}} \rfloor, \lfloor n^{\frac{2}{5}} \rfloor, \lfloor n^{\frac{1}{3}} \rfloor$	1	simple

Setting E (detection comparison). In Scenario E, there are several true break points in the data-generating process with $T = 1000$, $p = 10$, $q = 1$ with break points being equally spaced: $\lfloor \frac{T}{m_0+1} \rfloor, \lfloor \frac{2T}{m_0+1} \rfloor, \ldots, \lfloor \frac{m_0 T}{m_0+1} \rfloor$. In Scenarios E.1 though E.6, the true number of break points are $m_0 = 1, 2, 3, 4, 5,$ and 6, respectively. The true coefficient matrices are similar to simulation B, as depicted in Figure 2 (top right panel) with repeated entries -0.6, 0.6, and -0.6 off the main diagonal. We consider the BSS method with three different block size settings: large $b_n = 20$, medium $b_n = 15$ and small $b_n = 10$.

All methods achieve selection rates over 90% (left panel of Figure 3). In fact, all methods reach 100% when $m_0 < 5$, while the selection for TSP and BSS with large block size are within the interval [90%, 100%] for $m_0 = 5, 6$. This is consistent with the discussion in Remark 6 on minimum spacing between consecutive break points. Specifically, larger block sizes for BSS imply fewer break points allowed, while medium and small block sizes yield similar results compared to TSP and DP. Further, the Hausdorff distance between the set of estimated and true break points—$d_H\left(\widehat{\mathcal{A}}_n^f, \mathcal{A}_n\right)$—is a reasonable measure for estimation accuracy of the location of break points. The middle panel in Figure 3 illustrates the performance of all three methods in terms of this metric (averaged over 100 replicates). It can be seen that BSS outperforms the TSP and DP methods across all settings, while the advantage of BSS becomes more significant for larger m_0 values. On the other hand, the average computation time ($\sim$5 sec for $b_n = 10$; $\sim$2.5 sec for $b_n = 15$ and $b_n = 20$) of the BSS method is significantly lower compared to DP ($\sim$4500 sec) and TSP methods ($\sim$ 500 sec).

Next, we compare the computation time for the following three methods: BSS, DP, and TSP. Five additional simulation

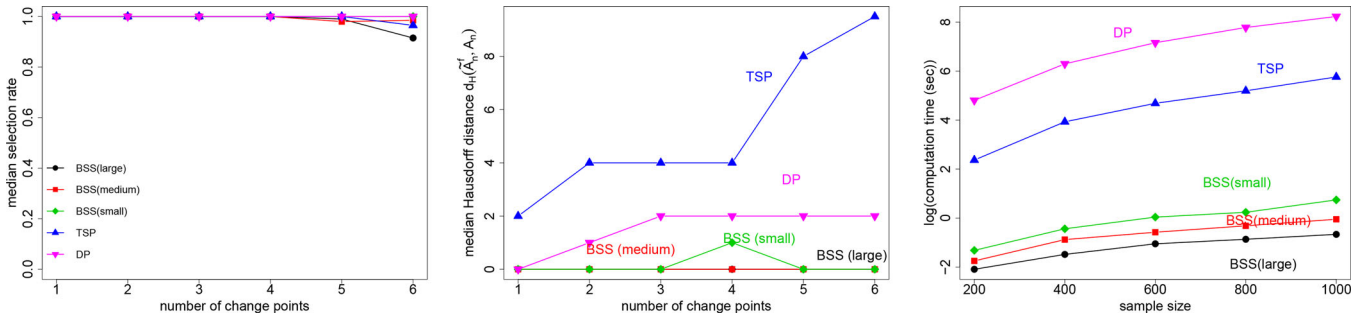

Figure 3. (Left) Median selection rate for the BSS (large, medium, and small), DP and TSP methods in simulation E; (Middle) Median Hausdorff distance for the BSS (large, medium, small), DP and TSP methods in simulation E; (Right) Logarithm of average computational time for the BSS (large, medium, and small), DP and TSP methods in simulation F.

scenarios (F.1–F.5) are considered with model parameter values summarized in Table 3. Details of the simulation settings are as follows:

Setting F (computation time comparison). In Scenario F, $p = 8$, $q = 1$, $m_0 = 2$, $t_1 = \lfloor \frac{T}{3} \rfloor$, $t_2 = \lfloor \frac{2T}{3} \rfloor$. The AR coefficients are chosen to have the same simple 1-off diagonal structure as in Scenario B as shown in the top right panel of Figure 2 with repeated entries -0.8, 0.8, and -0.8 in the 1-off main diagonal. The sample size for Scenarios F.1 through F.5 are $T = 200, 400, 600, 800,$ and 1000, respectively.

The average computation time over 100 replicates (in logarithmic scale) for simulation setting F is plotted in the right panel of Figure 3. BSS with large block sizes is the fastest method overall, while DP is the slowest one. It is worth noting that BSS with small block sizes remains faster than both TSP and DP, while its estimation accuracy and selection rate are the best over all these methods. In this numerical experiment, the reduction in computation time in BSS (small block size) compared to TSP and DP are over 95% and 98%, respectively, while BSS with medium and large block sizes achieve even a higher reduction in computation time.

Experiments E and F reveal the fact that BSS-based methods are among the fastest detection methods for VAR models, while their selection rate and estimation accuracy also outperform some of the current competing methods. The upshot of this extensive numerical work is that carefully selecting blocks where the model parameters are kept fixed offers large computational gains in change point detection, without sacrificing estimation accuracy.

We also compared the BSS method to the SBS (Cho and Fryzlewicz 2015) and DCBS methods (Cho 2016) in terms of detection accuracy and computation time. Details of this comparison are given in Appendix C.

6. An Application to Electroencephalogram (EEG) Data

We apply BSS, TSP and DP to an EEG dataset analyzed in Trujillo (2019). In this database, EEG signals from active electrodes for $p = 21$ channels are recorded at a sampling frequency of 256Hz, for a total of 187 sec ($T \sim 48{,}000$). The stimulus procedure tested on the selected subject comprised of three 1-min duration interleaved sessions with eyes open and closed. The time series for all 21 EEG channels (after de-trending and scaling the data) are shown in Figure 4. The changes of status (eyes open (EO) to eyes closed (EC) or eyes closed (EC) to eyes open (EO)) were estimated to take place at $t_1 = 15{,}896$ and $t_2 = 32{,}120$. We consider these two time points as the "true" break points, since it is likely for the brain connectivity to change at these time points due to the stimulus procedure.

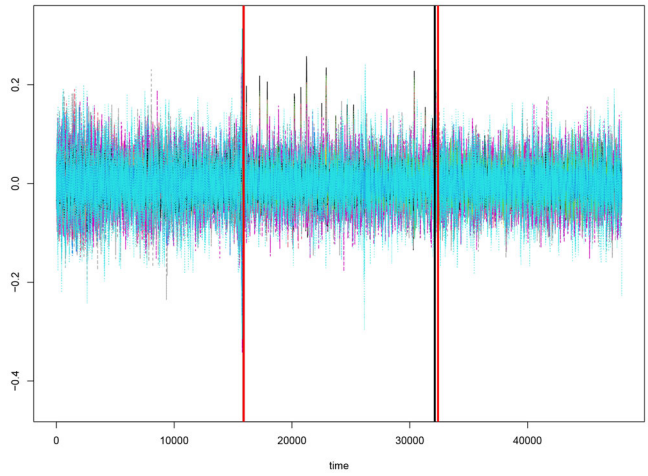

Figure 4. EEG data with 21 channels over 187 sec. Red solid lines locate the two selected break points using the BSS method with $b_n = 300$ while the black solid lines represent the true change point locations. The estimated three segments - from left to right- represent eyes closed (EC), eyes open (EO), and eyes closed (EC), respectively.

Table 4. Location of break points detected in the EEG data using three estimation methods with different settings (true change points are $t_1 = 15{,}896$ and $t_2 = 32{,}120$).

Method	Estimated change points	Computation time (sec)	sample size
DP ($\gamma = 0.5$)	15,552, 16,384, 31,744, 33,600	5222	1500
DP (default setting)	–	47,310	1500
TSP	1472 , 26,56 ,10,848 ,15,616, 15,968, 26,112, 32,160	1331	1500
BSS ($b_n = 200$)	15,804, 32,001	1023	48,000
BSS ($b_n = 250$)	15,826, 32,221	1072	48,000
BSS ($b_n = 300$)	15,601, 32,231	1247	48,000

NOTE: The estimated change points based on the sub-sampled dataset are rescaled back to the original time scale.

BSS with three different block sizes $b_n = 200, 250,$ and 300 was applied to this data. As seen in Table 4, all BSS methods detected two break points around the true ones, that is, $t_1 = 15{,}896$ and $t_2 = 32{,}120$. Further, the BSS method is robust to

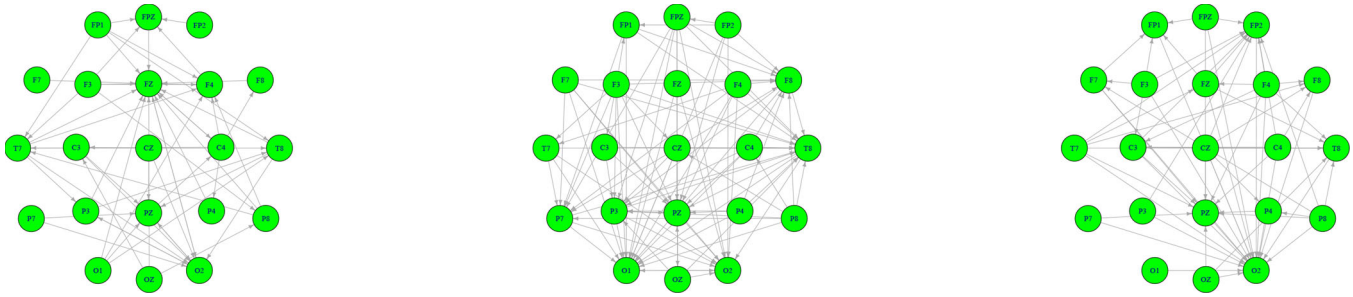

Figure 5. Network of Granger causal interactions among EEG channels based on data from Figure 4. Three networks—from left to right—represent eyes closed (EC), eyes open (EO), and eyes closed (EC), respectively.

the choice of block size b_n. The selected break points by the BSS method with $b_n = 300$ are also depicted in Figure 4 (red solid lines). While the computation time for BSS was $\sim$ 20 min, a simple calculation based on the results in Section 5.2 reveals that it would take 1–2 ($\sim$ 20) days for TSP (DP) to detect break points on this dataset due to the exceedingly large sample size. Thus, we did not apply these methods to the original data. Instead, we applied them on a sub-sampled version of the data in which 1 in every 32 observations is retained. This was mainly to reduce the sample size (to $\sim$ 1500) in order for TSP and DP to be able to perform detection of break points within hours. A summary of the results of TSP and DP are reported in Table 4 as well. The TSP method selected 7 estimated break points, with some of them being far away from the true ones. Under a manual selection of the tuning parameter $\gamma_n = 0.5$, the DP method detected 4 break points around the two true ones, while with the default setting of γ_n calculated by cross-validation (selected as $\gamma_n = 1$), the DP can not detect any change points. The computation time for TSP (DP) for the sub-sampled data set is $\sim$ 22 min ($\sim$ 87 min for the fixed tuning parameter case and $\sim$ 13 hr for the data-driven tuning parameter case), still larger than BSS. Note that the sub-sampled data most likely exhibit different temporal dynamics than the original high-frequency time series and in addition, size of the jumps are also altered, both factors contributing to the poor performance of TSP and DP. However, as previously mentioned, the latter two methods are computationally expensive (especially DP) for routine use with such large datasets.

After detecting two change points using BSS with $b_n = 300$, and in order to provide insights into changes in the neuronal interactions between the two states—eyes closed (EC) and eyes open (EO)—we estimated the AR parameters in each segment (obtained from Equation (18)). The Granger causal network associated with these estimated transition matrices are depicted in Figure 5. These networks are constructed as follows. We discarded observations in the R_n radius neighborhood around $\widetilde{t}_1^f = 15601$ and $\widetilde{t}_2^f = 32,231$ in order to ensure stationarity of the remaining observations ($R_n = 350$). We then used the ℓ_1-penalized least square estimator in Equation (18) to obtain estimates of the VAR parameters for the three segments. Network edges in Figure 5 correspond to nonzero estimated coefficients. It is worth noting that we only plot coefficients that are at least larger than 10^{-5} in magnitude. This thresholding step is motivated by the known over-selection property of lasso (Shojaie, Basu, and Michailidis 2012) and is used to improve the interpretability of the estimated networks. Different brain connectivity structures among the three networks are depicted in Figure 5 and provide further evidence for the presence of break points in the data set. Moreover, comparing the second network (eyes open (EO) status) with the first and third networks (eyes closed (EC) status), although they have many common edges, they also exhibit several differences. Of interest are the brain activity changes related to channels within the visual cortex including P3, Pz, O1, and O2 (Nezamfar et al. 2011). Moreover, it can be seen that during the second segment (EO), the overall network connectivity increases compared to the ones in the EC segments.

7. Concluding Remarks

In this article, we developed a novel scheme that can consistently identify structural breaks in large scale high-dimensional non-stationary VAR models while reducing significantly computing time. The proposed BSS is applicable in settings where there are relatively few structural breaks compared to the number of time points available. Key technical developments include the calibration of the block size and the introduction of a novel local information criterion for screening out redundant candidate change points. Note that as a byproduct of this study, similar computational gains can be achieved in other models that employ a similar parameterization; for example, the settings in Harchaoui and Lévy-Leduc (2010), Chan, Yau, and Zhang (2014).

Supplementary Material

Appendix: Appendix A contains technical lemmas needed to prove the main results. Proofs of the main results are given in Appendix B. Details of the algorithm for solving the optimization problem (4) are given in Appendix C, while tuning parameter selections are summarized in Appendix D. Finally, additional comparison results are provided in Appendix E. (.pdf file)

R code: R code for the developed BSS detection algorithm described in the article with a PDF file for instruction. (.zip file)

References

Aue, A., and Horváth, L. (2013), "Structural Breaks in Time Series," *Journal of Time Series Analysis*, 34, 1–16. [1]

Aue, A., Rice, G., and Sönmez, O. (2018), "Detecting and Dating Structural Breaks in Functional Data Without Dimension Reduction," *Journal of the Royal Statistical Society*, Series B 80, 509–529. [1]

Bai, P., Safikhani, A., and Michailidis, G. (2020), "Multiple Change Points Detection in Low Rank and Sparse High Dimensional Vector Autoregressive Models," *IEEE Transactions on Signal Processing*, 68, 3074–3089. [2]

Barigozzi, M., Cho, H., and Fryzlewicz, P. (2018), "Simultaneous Multiple Change-Point and Factor Analysis for High-Dimensional Time Series," *Journal of Econometrics*, 206, 187–225. [2]

Basseville, M., and Nikiforov, I. V. (1993), *Detection of Abrupt Changes: Theory and Application*, Vol. 104. Englewood Cliffs, NJ: Prentice-Hall. [1]

Basu, S., and Michailidis, G. (2015), "Regularized Estimation in Sparse High-Dimensional Time Series Models," *The Annals of Statistics*, 43, 1535–1567. [2,6,8]

Beck, A., and Teboulle, M. (2009), "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM Journal on Imaging Sciences*, 2, 183–202. [9]

Bleakley, K., and Vert, J.-P. (2011), "The Group Fused Lasso for Multiple Change-Point Detection," arXiv:1106.4199. [8,9]

Boysen, L., Kempe, A., Liebscher, V., Munk, A., and Wittich, O. (2009), "Consistencies and Rates of Convergence of Jump-Penalized Least Squares Estimators," *The Annals of Statistics*, 157–183. [7]

Chan, N. H., Ing, C.-K., Li, Y., and Yau, C. Y. (2017), "Threshold Estimation Via Group Orthogonal Greedy Algorithm," *Journal of Business & Economic Statistics*, 35, 334–345. [7]

Chan, N. H., Yau, C. Y., and Zhang, R.-M. (2014), "Group Lasso for Structural Break Time Series," *Journal of the American Statistical Association*, 109, 590–599. [6,7,13]

Cho, H. (2016), "Change-Point Detection in panel Data Via Double Cusum Statistic," *Electronic Journal of Statistics*, 10, 2000–2038. [2,8,12]

Cho, H., and Fryzlewicz, P. (2015), "Multiple-Change-Point Detection for High Dimensional Time Series Via Sparsified Binary Segmentation," *Journal of the Royal Statistical Society*, Series B, 77, 475–507. [1,2,8,12]

Csörgö, M., and Horváth, L. (1997), *Limit Theorems in Change-Point Analysis*, Vol. 18, New Jersey: Wiley. [1]

Davis, R. A., Lee, T. C. M., and Rodriguez-Yam, G. A. (2006), "Structural Break Estimation for Nonstationary Time Series Models," *Journal of the American Statistical Association*, 101, 223–239. [1]

Fryzlewicz, P. (2014), "Wild Binary Segmentation for Multiple Change-Point Detection," *The Annals of Statistics*, 42, 2243–2281. [2]

——— (2017), "Tail-Greedy Bottom-Up Data Decompositions and Fast Multiple Change-Point Detection," *Annals of Statistics*, 46, 3390–3421. [1]

Fryzlewicz, P., and Subba Rao, S. (2014), "Multiple-Change-Point Detection for Auto-Regressive Conditional Heteroscedastic Processes," *Journal of the Royal Statistical Society*, Series B, 76, 903–924. [1]

Harchaoui, Z., and Lévy-Leduc, C. (2010), "Multiple Change-Point Estimation With a Total Variation Penalty," *Journal of the American Statistical Association*, 105, 1480–1493. [1,13]

Kaul, A., Fotopoulos, S. B., Jandhyala, V. K., and Safikhani, A. (2021), "Inference on the Change Point Under a High Dimensional Sparse Mean Shift," *Electronic Journal of Statistics*, 15, 71–134. [2]

Kaul, A., Jandhyala, V. K., and Fotopoulos, S. B. (2019), "An Efficient Two Step Algorithm for High Dimensional Change Point Regression Models Without Grid Search," *Journal of Machine Learning Research*, 20, 1–40. [8]

Killick, R., Fearnhead, P., and Eckley, I. A. (2012), "Optimal Detection of Changepoints With a Linear Computational Cost," *Journal of the American Statistical Association*, 107, 1590–1598. [1]

Leonardi, F., and Bühlmann, P. (2016), "Computationally Efficient Change Point Detection for High-Dimensional Regression," arXiv: 1601.03704. [2]

Lin, J., and Michailidis, G. (2017), "Regularized Estimation and Testing for High-Dimensional Multi-Block Vector-Autoregressive Models," *The Journal of Machine Learning Research*, 18, 4188–4236. [1,11]

Loh, P.-L., and Wainwright, M. J. (2012), "High-Dimensional Regression With Noisy and Missing Data: Provable Guarantees With Nonconvexity," *Annals of Statistics*, 40, 1637–1664. [6]

Matteson, D. S., and James, N. A. (2014), "A Nonparametric Approach for Multiple Change Point Analysis of Multivariate Data," *Journal of the American Statistical Association*, 109, 334–345. [2]

Nezamfar, H., Orhan, U., Purwar, S., Hild, K., Oken, B., and Erdogmus, D. (2011), "Decoding of Multichannel Eeg Activity From the Visual Cortex in Response to Pseudorandom Binary Sequences of Visual Stimuli," *International Journal of Imaging Systems and Technology*, 21, 139–147. [13]

Ombao, H., Von Sachs, R., and Guo, W. (2005), "Slex Analysis of Multivariate Nonstationary Time Series," *Journal of the American Statistical Association*, 100, 519–531. [1,2]

Preuss, P., Puchstein, R., and Dette, H. (2015), "Detection of Multiple Structural Breaks in Multivariate Time Series," *Journal of the American Statistical Association*, 110, 654–668. [2,10]

Rinaldo, A., Wang, D., Wen, Q., Willett, R., and Yu, Y. (2020), "Localizing Changes in High-dimensional Regression Models," arXiv: 2010.10410. [2]

Roy, S., Atchadé, Y., and Michailidis, G. (2017), "Change Point Estimation in High Dimensional Markov Random-Field Models," *Journal of the Royal Statistical Society*, Series B, 79, 1187–1206. [2,7]

Safikhani, A., and Shojaie, A. (2020), "Joint Structural Break Detection and Parameter Estimation in High-Dimensional Non-Stationary VAR Models," *Journal of American Statistical Association,* 1–14. [2,7,8,11]

Shojaie, A., Basu, S., and Michailidis, G. (2012), "Adaptive Thresholding for Reconstructing Regulatory Networks From Time-Course Gene Expression Data," *Statistics in Biosciences*, 4, 66–83. [13]

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005), "Sparsity and Smoothness Via the Fused Lasso," *Journal of the Royal Statistical Society*, Series B, 67, 91–108. [4]

Trujillo, L. (2019), *Raw Empirical EEG Data*, available at *https://dataverse.tdl.org/dataset.xhtml?persistentId=doi:10.18738/T8/ANS9Q1*. [12]

Wang, D., Yu, Y., and Rinaldo, A. (2017), "Optimal Covariance Change Point Detection in High Dimension," arXiv: 1712.09912. [2]

Wang, D., Yu, Y., Rinaldo, A., and Willett, R. (2019), "Localizing Changes in High-Dimensional Vector Autoregressive Processes," arXiv: 1909.06359. [2,3,8,11]

Wang, T., and Samworth, R. J. (2018), "High Dimensional Change Point Estimation Via Sparse Projection," *Journal of the Royal Statistical Society*, Series B, 80, 57–83. [2]

Wong, K. C., Li, Z., Tewari, A. (2020), "Lasso Guarantees for β-Mixing Heavy-Tailed Time Series," *Annals of Statistics*, 48, 1124–1142. [6]