Sequential Change Point Detection in High-dimensional Vector Auto-regressive Models

University of Florida and George Mason University

Supplementary Material

This supplementary material provides proofs of lemmas and theorems, as well as additional simulations and real data experiments. Definitions and lemmas used in the proofs are introduced in Sections S1 and S2, respectively. The proofs of the theorems appear in Section S3. Additional simulations are presented in Section S4, and further real data analysis is provided in Section S5. The use of a sequential updating technique for transition matrix estimation is discussed in Section S6, and post-change analysis is provided in Section S7.

S1 DEFINITIONS

In this appendix, we provide definitions for symbols and terms utilized throughout the appendices.

Definition 1. *For any $\gamma > 0$, a random variable X satisfies any of the following equivalent properties is called a sub-Weibull (γ) random variable:*

1. $\mathbb{P}(|X| > t) \leq 2\exp\{-(t/K_1)^\gamma\}$ *for all* $t \geq 0$,
2. $(\mathbb{E}|X|^p)^{1/p} \leq K_2 p^{1/\gamma}$ *for all* $p \geq 1 \wedge \gamma$,
3. $\mathbb{E}\left[\exp\left(|X|/K_3\right)^\gamma\right] \leq 2$.

The constants K_1, K_2 and K_3 differ from each other at most by a constant depending only on γ. The sub-Gaussian random variable is a special case of a sub-Weibull random variable with $\gamma = 2$. The sub-Weibull norm of X is defined as $\|X\|_{\psi_\gamma} := \sup_{p\geq 1}\left(\mathbb{E}|X|^p\right)^{1/p} p^{-1/\gamma}$.

Definition 1 is a straightforward combination of Lemma 5 and Definition 3 in Wong et al. (2020).

Definition 2. *For any $\gamma > 0$, a random vector $X \in \mathbb{R}^p$ is said to be a sub-Weibull (γ) random vector if all of its one-dimensional projections are sub-Weibull (γ) random variables. We define the sub-Weibull (γ) norm of a random vector as*

$$\|X\|_{\psi_\gamma} := \sup_{v\in S^{p-1}} \|v'X\|_{\psi_\gamma},$$

where S^{p-1} is the unit sphere in $\mathbb{R}^p$.

Definition 2 is from Definition 4 in Wong et al. (2020).

Definition 3. *Every VAR(h) process can be rewritten into a VAR(1) form that is $\tilde{X}_t = \tilde{A}\tilde{X}_{t-1} + \tilde{\varepsilon}_t$, and $\tilde{X}_t$ is stable if and only if X_t is stable.*

Definition 3 is from Lütkepohl (2005). We omit the details here to save space; for more information, please refer to page 15 in Lütkepohl (2005).

S2 LEMMAS AND PROOFS

In this appendix, we introduce several lemmas that will be employed in the proof of the Theorems. We introduce the notation $D^l \coloneqq \hat{A}_l - A_l$ and define d_{ij}^l as the (i, j)-th element of D^l. We utilize the symbol N to represent a generic sample size, as opposed to exclusively using n or ω in the subsequent lemmas. These lemmas will be applied with $N = n$ or ω during the proof of the Theorems.

Lemma 1. Consider a random realization $\{X_{-h+1}, \ldots, X_N\}$ generated from a VAR(h) process with Assumption 1-4 satisfied. Then, there exist constants $c_i > 0$ such that for all $N \gtrsim \max\{\nu_{LB}^2, 1\}\, s(2\log p + \log h)$, with probability at least $1 - c_1 \exp\left(-c_2 N \min\left\{\nu_{LB}^{-2}, 1\right\}\right)$,

$$\theta' \hat{\Gamma}_N \theta \geq \alpha_{LB} \|\theta\|_2^2 - \tau_{LB}^N \|\theta\|_1^2, \quad \text{for all } \theta \in \mathbb{R}^{hp^2}$$

where

$$\hat{\Gamma}_N \coloneqq I_p \otimes \left(\mathcal{X}_N'\mathcal{X}_N/N\right), \nu_{LB} = c_3\frac{\mu_{\max}(\mathcal{A})}{\mu_{\min}(\tilde{\mathcal{A}})}, \alpha_{LB} = \frac{\sigma^2}{2\mu_{\max}(\mathcal{A})},$$

$$\tau_{LB}^N = \alpha_{LB}\max\left\{\nu_{LB}^2, 1\right\}\frac{\log h + 2\log p}{N},$$

$$\mu_{\min}(\mathcal{A}) := \min_{|z|=1}\Lambda_{\min}\left(\mathcal{A}^*(z)\mathcal{A}(z)\right), \mu_{\max}(\mathcal{A}) := \max_{|z|=1}\Lambda_{\max}\left(\mathcal{A}^*(z)\mathcal{A}(z)\right),$$

$$\mathcal{A}(z) := I_p - \sum_{l=1}^{h} A_l z^l \text{ and } \tilde{\mathcal{A}}(z) := I_{hp} - \tilde{A}z.$$

Proof. This lemma results from a straightforward application of Proposition 4.2 in Basu and Michailidis (2015) to a VAR(h) process. □

Lemma 2. Consider a random realization $\{X_{-h+1}, \ldots, X_N\}$ generated from a VAR(h) process with Assumption 1-4 satisfied. Then, there exist constants $c_i > 0$ (different from Lemma 1) such that for all $N \gtrsim \max\left\{\nu_{UB}^2, 1\right\} s\left(2\log p + \log h\right)$, with probability at least $1 - c_1\exp\left(-c_2 N\min\{\nu_{UB}^{-2}, 1\}\right)$,

$$\theta'\hat{\Gamma}_N\theta \leq 3\alpha_{UB}\|\theta\|_2^2 + \tau_{UB}^N\|\theta\|_1^2, \quad \text{for all } \theta \in \mathbb{R}^{hp^2}$$

where

$$\alpha_{UB} = \frac{\sigma^2}{2\mu_{\min}(\mathcal{A})},\ \nu_{UB} = 54\frac{\mu_{\min}(\mathcal{A})}{\mu_{\min}(\tilde{\mathcal{A}})} \text{ and } \tau_{UB}^N = c_3\alpha_{UB}\max\left\{\nu_{UB}^2, 1\right\}\frac{\log h + 2\log p}{N}.$$

Proof. This proof closely resembles the proof of Proposition 4.2 in Appendix B of Basu and Michailidis (2015), so we will only highlight the necessary modifications. The unmentioned portions should adhere to the proof provided in Basu and Michailidis (2015).

At the beginning of the proof in Basu and Michailidis (2015), besides $\Lambda_{\min}\left(\Gamma_{\tilde{X}}(0)\right) \geq \frac{\Lambda_{\min}(\Sigma_\epsilon)}{\mu_{\max}(\mathcal{A})}$, we also have $\Lambda_{\max}\left(\Gamma_{\tilde{X}}(0)\right) \leq \frac{\Lambda_{\max}(\Sigma_\epsilon)}{\mu_{\min}(\mathcal{A})}$ from Proposition 2.3 and the bounds in (4.1) in Basu and Michailidis (2015). Before applying Lemma 12, we set $\eta = \nu_{UB}^{-1}$ instead of ω^{-1}. Then, applying the Lemma 12 in Loh and Wainwright (2012) with $\delta = \Lambda_{\max}\left(\Sigma_\epsilon\right)/54\mu_{\min}(\mathcal{A})$ and $\Gamma = S - \Gamma_{\tilde{X}}(0)$ where S $= (\mathcal{X}_N'\mathcal{X}_N/N)$, we have $\theta' S\theta - \theta'\Gamma_{\tilde{X}}(0)\theta \leq \alpha_{UB}(\|\theta\|_2^2 + \frac{1}{k}\|\theta\|_1^2)$, so $\theta' S\theta \leq 3\alpha_{UB}\|\theta\|_2^2 + \frac{\alpha_{UB}}{k}\|\theta\|_1^2$ for all $\theta \in \mathbb{R}^{hp}$ with probability at least $1 - 2\exp\left[-cN\min\{\nu_{UB}^{-2}, 1\} + 2k\log(dp)\right]$. Finally, we set k $= \left\lceil cN\min\{\nu_{UB}^{-2}, 1\}/4\log(hp)\right\rceil$ and follow the rest of proof in Basu and Michailidis (2015) to get this Lemma. $\lceil x \rceil$ represents the smallest integer that is greater than or equal to x. □

To maintain symbol consistency, we use "k" to denote the constant "s" in Basu and Michailidis (2015), and "s" represents the sparsity parameter "k" in Basu and Michailidis (2015). In this proof, $\Lambda_{\max}\left(\Sigma_\epsilon\right)$ and $\Lambda_{\min}\left(\Sigma_\epsilon\right)$ degenerate to σ^2 because of the variance structure of errors in our model.

Lemma 3. Under the same setup of Lemma 1, there exist constants $c_i > 0$ such that for $N \gtrsim \log h + 2\log p$, with probability at least $1 - c_1\exp\left[-c_2(\log h + 2\log p)\right]$, we have

$$\left\|\hat{\gamma}_N - \hat{\Gamma}_N\beta^*\right\|_\infty \leq \mathbb{Q}\left(\beta^*, \sigma^2\right)\sqrt{\frac{\log h + 2\log p}{N}}$$

where $\hat{\gamma}_N = (I \otimes \mathcal{X}_N')Y_N/N$ and $\mathbb{Q}(\beta^*, \sigma^2) = c_0[\sigma^2 + \frac{\sigma^2}{\mu_{\min}(\mathcal{A})} + \frac{\sigma^2\mu_{\max}(\mathcal{A})}{\mu_{\min}(\mathcal{A})}]$.

Proof. This lemma results from a straightforward application of Proposition 4.3 in Basu and Michailidis (2015) to a VAR(h) process. □

Lemma 4. Consider the ℓ_1 estimation problem (3.2) discussed in Section 3.1 in the main paper, under the same setup of Lemma 1, with $N \geq 32 \max\{\nu_{LB}^2, 1\}\, s(\log h + 2\log p)$. Then, there exist constants $c_i > 0$ such that, for any $\lambda_N \geq 4\mathbb{Q}\left(\beta^*, \sigma^2\right)\sqrt{(\log h + 2\log p)/N}$, any solution $\hat{\beta}_N$ of (3) satisfies

$$\left\|\hat{\beta}_N - \beta^*\right\|_1 \leq 64 s\lambda_N/\alpha_{LB}, \left\|\hat{\beta}_N - \beta^*\right\|_2 \leq 16\sqrt{s}\lambda_N/\alpha_{LB}$$
$$\text{and } \left(\hat{\beta}_N - \beta^*\right)' \hat{\Gamma}_N \left(\hat{\beta}_N - \beta^*\right) \leq 128 s\lambda_N^2/\alpha_{LB}$$

with probability at least $1 - c_1 \exp\left[-c_2(\log h + 2\log p)\right] - c_3 \exp\left(-c_4 N \min\left\{\nu_{LB}^{-2}, 1\right\}\right)$.

Proof. This lemma results from a straightforward application of Proposition 4.1, 4.2, and 4.3 in Basu and Michailidis (2015) to a VAR(h) process, aided by the union bound. □

Here, we have summarized several equivalent expressions for the terms found in the aforementioned lemmas. We have

$$\left\|\hat{\gamma}_N - \hat{\Gamma}_N \beta^*\right\|_\infty = \max_{\substack{j,j' \in (1,\ldots,p) \\ l \in (1,\ldots,h)}} \frac{1}{N}\left|\sum_{i=1}^{N} x_{i-l,j}\varepsilon_{i,j'}\right|,$$
$$\left\|\hat{\beta}_N - \beta^*\right\|_1 = \sum_{l=1}^{h}\sum_{j=1}^{p}\sum_{j'=1}^{p}\left|d_{j,j'}^{l}\right| \text{ and}$$
$$\left(\hat{\beta}_N - \beta^*\right)' \hat{\Gamma}_N \left(\hat{\beta}_N - \beta^*\right) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{p}\left(\sum_{l=1}^{h}\sum_{j'=1}^{p} d_{jj'}^{l} x_{i-l,j'}\right)^2.$$

S3 PROOF OF THEOREMS

In this appendix, we provide the proofs for Theorems 1 and 2 presented in the main paper.

Proof of Theorem 1:

The proof of Theorem 1 consists of two parts. The first part is the proof of $\sqrt{p\omega}(\frac{\hat{R}_t^{(n,\omega)}}{p} - \hat{\sigma}_n^2) \xrightarrow{D} \mathcal{N}(0, \mathrm{Var}(\varepsilon_{1,1}^2))$, and the second part is the proof of $\hat{V}_n \xrightarrow{p} \mathrm{Var}(\varepsilon_{1,1}^2)$. Finally, applying Slutsky's Theorem leads to Theorem 1. By some straightforward algebra, we have

$$\begin{aligned}
\sqrt{p\omega}\left(\frac{\hat{R}_t^{(n,\omega)}}{p} - \hat{\sigma}_n^2\right) = & \underbrace{\sqrt{p\omega}\left(\frac{1}{p\omega}\sum_{i=t+1}^{t+\omega}\|\varepsilon_i\|_2^2 - \sigma^2\right)}_{\text{term 1}} \\
& \underbrace{- \sqrt{\frac{\omega}{n}}\sqrt{pn}\left(\frac{1}{pn}\sum_{i=1}^{n}\|\varepsilon_i\|_2^2 - \sigma^2\right)}_{\text{term 2}} \underbrace{- \frac{1}{\sqrt{p\omega}}\sum_{i=t+1}^{t+\omega}\left\|\sum_{l=1}^{h}\left(\hat{A}_l - A_l\right)X_{i-l}\right\|_2^2}_{\text{term 3}} \\
& \underbrace{- \frac{2}{\sqrt{p\omega}}\sum_{i=t+1}^{t+\omega}\sum_{l=1}^{h}X_{i-l}'\left(\hat{A}_l - A_l\right)'\varepsilon_i}_{\text{term 4}} \underbrace{+ \frac{\sqrt{\omega}}{n\sqrt{p}}\sum_{i=1}^{n}\left\|\sum_{l=1}^{h}\left(\hat{A}_l - A_l\right)X_{i-l}\right\|_2^2}_{\text{term 5}} \\
& \underbrace{- \frac{2\sqrt{\omega}}{n\sqrt{p}}\sum_{i=1}^{n}\sum_{l=1}^{h}X_{i-l}'\left(\hat{A}_l - A_l\right)'\varepsilon_i}_{\text{term 6}}.
\end{aligned}$$

- For term 1 and 2, because errors are iid with variance matrix $\sigma^2 I_p$, we have

$$\sqrt{p\omega}\left(\frac{1}{p\omega}\sum_{i=t+1}^{t+\omega}\|\varepsilon_i\|_2^2 - \sigma^2\right) \xrightarrow{D} \mathcal{N}\left(0, \mathrm{Var}\left(\varepsilon_{1,1}^2\right)\right) \text{ as } \omega \to \infty, \text{ and}$$

$$\sqrt{\frac{\omega}{n}}\sqrt{pn}\left(\frac{1}{pn}\sum_{i=1}^{n}\|\varepsilon_i\|_2^2-\sigma^2\right)\xrightarrow{p}0 \text{ as } n\to\infty$$

by Central Limit Theorem Montgomery and Runger (2010) and Slutsky's Theorem Van der Vaart (2000) under the condition $\omega = o(n)$.

- Under the condition for Theorem 1, by the union bound in probability, with probability at least $1-c_1\exp(-c_2\omega\min\left\{\nu_{UB}^{-2},1\right\})-c_3\exp\left(-c_4 n\min\left\{\nu_{LB}^{-2},1\right\}\right)-c_5\exp[-c_6(2\log p+\log h)]$, We have

$$\begin{aligned}
&\left(\hat{\beta}_n-\beta^*\right)'\hat{\Gamma}_\omega\left(\hat{\beta}_n-\beta^*\right)\leq 3\alpha_{UB}\left\|\hat{\beta}_n-\beta^*\right\|_2^2+\tau_{UB}^{\omega}\left\|\hat{\beta}_n-\beta^*\right\|_1^2\\
&\leq\left(3\alpha_{UB}+c\alpha_{UB}\max\left\{\nu_{UB}^2,1\right\}s\frac{\log h+2\log p}{\omega}\right)\left\|\hat{\beta}_n-\beta^*\right\|_2^2\\
&\leq\tilde{c}\frac{\alpha_{UB}}{\alpha_{LB}^2}\max\left\{\nu_{UB}^2,1\right\}\mathbb{Q}^2\left(\beta^*,\sigma^2\right)s\frac{\log h+2\log p}{n}\max\left(1,s\frac{(\log h+2\log p)}{\omega}\right).
\end{aligned}$$

The first inequality is by Lemma 2; the second inequality comes from the fact that $\left\|\hat{\beta}_n-\beta^*\right\|_1\leq 4\sqrt{s}\left\|\hat{\beta}_n-\beta^*\right\|_2$ which is proved in Appendix B: Proof of Proposition 4.1 in Basu and Michailidis (2015); for the last inequality, we apply Lemma 4 with $\lambda_n = 4\mathbb{Q}\left(\beta^*,\sigma^2\right)\sqrt{(\log h+2\log p)/n}$ and the fact that $(a+b)\leq 2\max(a,b)$, where $\tilde{c}$ is a finite positive constant. Hence, with $\omega = o(n)$ and $\sqrt{n}\gtrsim\frac{s(\log h+2\log p)}{\sqrt{p}}$, term 3 converges to zero in probability as n goes to infinity.

- For term 4, we have

$$\begin{aligned}|\text{term 4}| &= \left| \frac{2}{\sqrt{p\omega}} \sum_{l=1}^{h} \sum_{j=1}^{p} \sum_{j'=1}^{p} \left(d_{jj'}^{l} \sum_{i=t+1}^{t+\omega} x_{i-l,j} \varepsilon_{i,j'} \right) \right| \\ &\leq 2\sqrt{\frac{\omega}{p}} \left(\sum_{l=1}^{h} \sum_{j=1}^{p} \sum_{j'=1}^{p} \left| d_{ij'}^{l} \right| \right) \left(\max_{\substack{j,j' \in (1,\ldots,p) \\ l \in (1,\ldots,h)}} \frac{1}{\omega} \left| \sum_{i=t+1}^{t+\omega} x_{i-l,j} \varepsilon_{i,j'} \right| \right) \\ &\leq \frac{c}{\alpha_{LB}} \mathbb{Q}^2 \left(\beta^*, \sigma^2 \right) \frac{s \left(\log h + 2 \log p \right)}{\sqrt{np}}\end{aligned}$$

with probability at least $1 - c_1 \exp[-c_2(\log h + 2\log p)] - c_3 \exp(-c_4 n \min\left\{\nu_{LB}^{-2}, 1\right\})$. Conditions that $\omega \gtrsim \log h + 2\log p$ and $n \geq 32 \max\left\{\nu_{LB}^2, 1\right\} s(\log h + 2\log p)$ are needed. The last inequality comes from the application of Lemma 3 on $\max_{j,j' \in (1,\ldots,p)} \frac{1}{\omega} \left| \sum_{i=t+1}^{t+\omega} x_{i-1,j} \varepsilon_{i,j'} \right|$ and the application of Lemma 4 on $\sum_{j=1}^{p} \sum_{j'=1}^{p} |d_{ij'}|$ by choosing λ_n to be the smallest possible value. Then, we applied the union bound to get the result. Hence, under condition $\frac{s(\log h + 2\log p)}{\sqrt{p}} = o(\sqrt{n})$, we have the absolute value of term 4 converges to zero in probability as n goes to infinity.

- For term 5, we have

$$\text{term 5} \leq c \frac{1}{\alpha_{LB}} \mathbb{Q}^2 \left(\beta^*, \sigma^2 \right) \sqrt{\frac{\omega}{n}} \frac{s \left(\log h + 2 \log p \right)}{\sqrt{np}},$$

with probability at least $1 - c_1 \exp[-c_2(\log h + 2\log p)] - c_3 \exp\left(-c_4 n \min\left\{\nu_{LB}^{-2}, 1\right\}\right)$, under condition $n \geq \max 32 \left\{\nu_{LB}^2, 1\right\} s(\log h + 2\log p)$, by directly applying Lemma 4. Hence, under condition $\omega = o(n)$ and $\sqrt{n} \gtrsim \frac{s(\log h + 2\log p)}{\sqrt{p}}$, we have term 5 converges to zero in probability as n goes to infinity.

- For term 6, we have

$$|\text{term 6}| \leq 2\sqrt{\frac{\omega}{p}} \left(\sum_{l=1}^{h} \sum_{j=1}^{p} \sum_{j'=1}^{p} \left| d_{ij'}^{l} \right| \right) * \left(\max_{\substack{j,j' \in (1,\ldots,p) \\ l \in (1,\ldots,h)}} \frac{1}{n} \left| \sum_{i=1}^{n} x_{i-l,j} \varepsilon_{i,j'} \right| \right)$$
$$\leq c \frac{1}{\alpha_{LB}} \mathbb{Q}^2 \left(\beta^*, \sigma^2 \right) \sqrt{\frac{\omega}{n}} \frac{s \left(\log h + 2 \log p \right)}{\sqrt{np}}$$

  with probability at least $1 - c_1 \exp[-c_2(\log h + 2\log p)] - c_3 \exp\left(-c_4 n \min\left\{\nu_{LB}^{-2}, 1\right\}\right)$, under condition $n \geq 32 \max\left\{\nu_{LB}^2, 1\right\} s(\log h + 2\log p)$. Hence, under additional condition $\omega = o(n)$ and $\sqrt{n} \gtrsim \frac{s(\log h + 2\log p)}{\sqrt{p}}$, the absolute value of term 6 converges to zero in probability as n goes to infinity.

Finally, by applying Slutsky's Theorem, we conclude the proof of the first part. In order to prove the main theorem, we still need to prove the second part: $\hat{V}_n \xrightarrow{p} \mathrm{Var}(\varepsilon_{1,1}^2)$. Firstly, note that

$$\hat{\sigma}_n^2 = \underbrace{\frac{1}{pn} \sum_{i=1}^{n} \left\| \sum_{l=1}^{h} \left(\hat{A}_l - A_l \right) X_{i-l} \right\|_2^2}_{\text{term 1}} - \underbrace{\frac{2}{pn} \sum_{i=1}^{n} \sum_{l=1}^{h} X_{i-l}' \left(\hat{A}_l - A_l \right)' \varepsilon_i}_{\text{term 2}} + \underbrace{\frac{1}{pn} \sum_{i=1}^{n} \left\| \varepsilon_i \right\|_2^2}_{\text{term 3}}.$$

- For term 1, under the same condition in the first part of the proof, similar to the term 5 in the first part, we have term 1 converges to zero in probability as n goes to infinity by directly applying Lemma 4.

- For term 2, under the same condition in the first part of the proof, similar to the term 6 in the first part, we have term 2 converges to zero in probability as n goes to infinity by directly applying Lemmas 3 and 4.

- For term 3, we have $\frac{1}{pn} \sum_{i=1}^{n} \left\| \varepsilon_i \right\|_2^2 \xrightarrow{p} \mathrm{E}(\varepsilon_{1,1}^2)$ as $n \to \infty$ by applying the weak

law of large number (Ross (2014)), where $\mathrm{E}(x)$ stands for the expectation of x. Thus, we have $\hat{\sigma}_n^4 \xrightarrow{p} \mathrm{E}(\varepsilon_{1,1}^2)^2$ by applying Slutsky's Theorem (Van der Vaart (2000)) and Continuous Mapping Theorem (Billingsley (2013)).

On the other hand, we have

$$\frac{1}{pn}\sum_{i=1}^{n}\left\|\left(\sum_{l=1}^{h}\hat{A}_l X_{i-l}\right)-X_i\right\|_4^4 = \underbrace{\frac{1}{pn}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(\sum_{j'=1}^{p}\sum_{l=1}^{h} d_{jj'}^{l} x_{i-l,j'}\right)^4}_{\text{term 1}}$$
$$+\underbrace{\frac{1}{pn}\sum_{i=1}^{n}\sum_{j=1}^{p}\varepsilon_{i,j}^4}_{\text{term 2}}+\underbrace{\frac{6}{pn}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(\left(\sum_{j'=1}^{p}\sum_{l=1}^{h} d_{jj'}^{l} x_{i-l,j'}\right)^2\varepsilon_{i,j}^2\right)}_{\text{term 3}}$$
$$-\underbrace{\frac{4}{pn}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(\left(\sum_{j'=1}^{p}\sum_{l=1}^{h} d_{jj'}^{l} x_{i-l,j'}\right)^3\varepsilon_{i,j}\right)}_{\text{term 4}}-\underbrace{\frac{4}{pn}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(\left(\sum_{j'=1}^{p}\sum_{l=1}^{h} d_{jj'}^{l} x_{i-l,j'}\right)\varepsilon_{i,j}^3\right)}_{\text{term 5}}.$$

- For term 1, under the same condition for the first part of the proof, we have

$$\text{term 1} \leq \frac{1}{pn}\left(\sum_{i=1}^{n}\sum_{j=1}^{p}\left(\sum_{j'=1}^{p}\sum_{l=1}^{h} d_{jj'}^{l} x_{i-l,j'}\right)^2\right)^2 \leq \frac{1}{pn}c^2\frac{\mathbb{Q}^4\left(\beta^*,\sigma^2\right)}{\alpha_{LB}^2}s^2\left(\log h+2\log p\right)^2$$

with probability at least $1-c_1\exp[-c_2(\log h+2\log p)]-c_3\exp(-c_4 n\min\left\{\nu_{LB}^{-2},1\right\})$ by directly applying Lemma 4. Thus, we have term 1 converges to zero in probability as n goes to inifinity.

- We have term 2 converges to $\mathrm{E}(\varepsilon_{1,1}^4)$ in probability as n goes to infinity by applying the weak law of large number Ross (2014).

- For term 3, under the same condition for the first part of the proof, we have

$$\begin{aligned}\text{term } 3 &\leq \max_{\substack{j\in(1,\ldots,p)\\ i\in(1,\ldots,n)}} \left(\frac{\varepsilon_{i,j}^2}{\sqrt{pn}}\right) \frac{6}{\sqrt{pn}} \sum_{i=1}^{n}\sum_{j=1}^{p}\left(\sum_{j'=1}^{p}\sum_{l=1}^{h} d_{jj'}^{l} x_{i-l,j'}\right)^2 \\ &\leq c\frac{\mathbb{Q}^2\left(\beta^*,\sigma^2\right)}{\alpha_{LB}} \frac{s\left(\log h + 2\log p\right)}{\sqrt{pn}}\end{aligned}$$

with probability at least

$1-2\exp(-c_5\sqrt{pn}+\log(np))-c_1\exp[-c_2(\log h+2\log p)]-c_3\exp(-c_4 n\min\left\{\nu_{LB}^{-2},1\right\})$.

The last inequality is from the fact that errors are independent and identically distributed sub-Gaussian random variables, so we have

$$\mathrm{P}(\max_{\substack{j\in(1,\ldots,p)\\ i\in(1,\ldots,n)}}(\frac{\varepsilon_{i,j}^2}{\sqrt{pn}}) > C) \leq np\,\mathrm{P}(\varepsilon_{1,1}^2 > C\sqrt{pn}) \leq 2\exp(-c_5\sqrt{pn}+\log(np))$$

by Definition 1 in Section S1, where C and c_5 are some finite positive constants. Then, by directly applying Lemma 4 on the rest of the term 3 together with union bound, we can get the last inequality for term 3. Thus, we have term 3 converges to zero in probability as n goes to infinity.

- Under the same condition for the first part of the proof, we have that the absolute

value of term 4 is less than or equal to

$$\begin{aligned}
&\frac{4}{pn}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(\left(\sum_{j'=1}^{p}\sum_{l=1}^{h}d_{jj'}^{l}x_{i-l,j'}\right)^{2}\left|\sum_{j'=1}^{p}\sum_{l=1}^{h}d_{jj'}^{l}x_{i-l,j'}\varepsilon_{i,j}\right|\right)\\
&\leq\left(\frac{4}{n^{1-a}p^{1-a}}\sum_{i=1}^{n}\sum_{j=1}^{p}\left(\sum_{j'=1}^{p}\sum_{l=1}^{h}d_{jj'}^{l}x_{i-l,j'}\right)^{2}\right)\\
&*\left(\max_{\substack{j,j'\in(1,\ldots,p)\\ i\in(1,\ldots,n),l\in(1,\ldots,h)}}\left|\frac{x_{i-l,j'}\varepsilon_{i,j}}{n^{a}p^{a}}\right|\right)\left(\sum_{l=1}^{h}\sum_{j=1}^{p}\sum_{j'=1}^{p}\left|d_{j,j'}^{l}\right|\right)\\
&\leq c\frac{s\left(\log h+2\log p\right)}{\sqrt{np}}\sqrt{\frac{s\left(\log h+2\log p\right)}{n}}\frac{\sqrt{s}}{n^{1/2-a}p^{1/2-a}}
\end{aligned}$$

with probability at least $1-2\exp(-\tilde{c}p^{a}n^{a}+\log(np^{2}h))-c_{1}\exp[-c_{2}(\log h+2\log p)]-c_{3}\exp(-c_{4}n\min\{\nu_{LB}^{-2},1\})$, where $\tilde{c}$ and c are some finite positive constants and a is an arbitrary small positive constant less than $1/2$. The last inequality is by applying the Lemma 4 on the first and last terms. For the middle term, we have for a positive finite constant c^{*}, $\mathrm{P}\left(\max_{\substack{j,j'\in(1,\ldots,p)\\ i\in(1,\ldots,n),l\in(1,\ldots,h)}}\left|\frac{x_{i-l,j'}\varepsilon_{i,j}}{n^{a}p^{a}}\right|>c^{*}\right)\leq\sum_{i,j,j',l}\mathrm{P}\left(|x_{i-l,j'}\varepsilon_{i,j}|>p^{a}n^{a}c^{*}\right)$. According to E.1 VAR section in Wong et al. (2020) with the assumption of stability and stationarity (Assumption 3), we know that $x_{i-l,j'}$ is sub-Gaussian for all i, l and j'. Then, according to Proposition 2.3 in Vladimirova et al. (2020), we have $x_{i-l,j'}\varepsilon_{i,j}$ is sub-weibull ($\gamma=1$) for all i, j, j' and l. Thus, according to Definition 1, we have for some finite positive constant $\tilde{c}$, $\sum_{i,j,j',l}\mathrm{P}\left(|x_{i-l,j'}\varepsilon_{i,j}|>p^{a}n^{a}c^{*}\right)\leq 2\exp\left(-\tilde{c}p^{a}n^{a}+\log(np^{2}h)\right)$. Finally, by the union bound we can get the last inequality. Thus, we have under the same condition for the proof of first part with additional condition, $n^{1/2-a}\gtrsim\frac{\sqrt{s}}{p^{1/2-a}}$ for

some $a \in (0, 1/2)$, the absolute value of term 4 converges to zero in probability as n goes to infinity.

- Under the same condition for the first part of the proof, the absolute value of term 5 is less than or equal to

$$4\left(\max_{\substack{j,j'\in(1,\ldots,p)\\ i\in(1,\ldots,n),l\in(1,\ldots,h)}}\left|\frac{x_{i-l,j'}\varepsilon_{i,j}^3}{n^b p^b}\right|\right)\frac{n^b}{p^{1-b}}\left(\sum_{l=1}^{h}\sum_{j=1}^{p}\sum_{j'=1}^{p}\left|d_{j,j'}^l\right|\right)$$
$$\leq c\frac{\mathbb{Q}\left(\beta^*,\sigma^2\right)}{\alpha_{LB}}\frac{\sqrt{s\left(\log h+2\log p\right)}}{n^{1/4}p^{1/4}}\frac{\sqrt{s}}{p^{3/4-b}n^{1/4-b}}$$

with probability at least $1-2\exp(-\tilde{c}p^{b/2}n^{b/2}+\log(np^2h))-c_1\exp[-c_2(\log h+2\log p)]-c_3\exp(-c_4 n\min\{\nu LB^{-2},1\})$, where $\tilde{c}$ and c are some finite positive constants and b is an arbitrary small positive constant less than $1/4$. Derivations of term 4 and 5 are similar with the only change that $x_{i-l,j'}\varepsilon_{i,j}^3$ is sub-weibull($\gamma = 1/2$) for all i, j, j' and l. Thus, we have under the same condition for the proof of first part with additional condition that $n^{1/4-b} \gtrsim \frac{\sqrt{s}}{p^{3/4-b}}$ for some $b \in (0, 1/4)$, the absolute value of term 5 converges to zero in probability as n goes to infinity.

Then, by applying Slutsky's Theorem and Continuous Mapping Theorem, we conclude the proof of the second part. Finally, applying Slutsky's Theorem again yields Theorem 1.

Proof of Theorem 2:

By some straightforward algebra, we have $\hat{R}^{(n,\omega)}_{t^*+h}$ is equal to

$$\underbrace{\frac{1}{\omega}\sum_{i=t^*+h+1}^{t^*+h+\omega}\left\|\left(\sum_{l=1}^{h}\left(\hat{A}_l-A_l\right)X_{i-l}\right)-\varepsilon_i\right\|_2^2}_{T_1}\underbrace{+\frac{1}{\omega}\sum_{i=t^*+h+1}^{t^*+h+\omega}\left\|\sum_{l=1}^{h}\left(A_l-A_l^*\right)X_{i-l}\right\|_2^2}_{T_2}$$
$$\underbrace{-\frac{2}{\omega}\sum_{i=t^*+h+1}^{t^*+h+\omega}\left(\varepsilon_i^T\left(\sum_{l=1}^{h}\left(A_l-A_l^*\right)X_{i-l}\right)\right)}_{T_3}$$
$$\underbrace{+\frac{2}{\omega}\sum_{i=t^*+h+1}^{t^*+h+\omega}\left(\left(\sum_{l=1}^{h}\left(\hat{A}_l-A_l\right)X_{i-l}\right)^T\left(\sum_{l=1}^{h}\left(A_l-A_l^*\right)X_{i-l}\right)\right)}_{T_4}.$$

Thus, we have that

$$\hat{T}^{(n,\omega)}_{t^*+h}=\underbrace{\sqrt{\frac{p\omega}{\hat{V}_n}}\left(\frac{T_1}{p}-\hat{\sigma}_n^2\right)}_{\text{term 1}}+\underbrace{\sqrt{\frac{p\omega}{\hat{V}_n}}\frac{T_2}{p}}_{\text{term 2}}+\underbrace{\sqrt{\frac{p\omega}{\hat{V}_n}}\frac{T_3}{p}}_{\text{term 3}}+\underbrace{\sqrt{\frac{p\omega}{\hat{V}_n}}\frac{T_4}{p}}_{term4}.$$

- We have term 1 converges to $\mathcal{N}(0,1)$ in distribution as n goes to infinity by the proof of Theorem 1.

- For term 2, we have

$$\begin{aligned}\sqrt{\frac{p\omega}{\hat{V}_n}}\frac{T_2}{p}&=\frac{1}{\sqrt{\hat{V}_n}}\sqrt{\frac{\omega}{p}}\left(\beta^*-\beta_{new}\right)'\hat{\Gamma}_\omega\left(\beta^*-\beta_{new}\right)\geq\frac{1}{\sqrt{\hat{V}_n}}\sqrt{\frac{\omega}{p}}(\alpha'_{LB}-s\left(\tau^\omega_{LB}\right)')\left\|\beta^*-\beta_{new}\right\|_2^2\\&=\frac{\alpha'_{LB}}{\sqrt{\hat{V}_n}}\sqrt{\frac{\omega}{p}}\left\|\beta^*-\beta_{new}\right\|_2^2-c\frac{s(\log h+2\log p)}{\sqrt{\omega p}}\frac{1}{\sqrt{\hat{V}_n}}\left\|\beta^*-\beta_{new}\right\|_2^2\end{aligned}$$

  with probability at least $1-c_1\exp(-c_2\omega\min((\nu'_{LB})^{-2},1))$ by using the sparsity assumption and Lemma 1.

On the other hand, by using the sparsity assumption and Lemma 2, we have

$$\sqrt{\frac{p\omega}{\hat{V}_n}}\frac{T_2}{p} \leq \frac{3\alpha'_{UB}}{\sqrt{\hat{V}_n}}\sqrt{\frac{\omega}{p}}\left\|\beta^* - \beta_{new}\right\|_2^2 + c'\frac{s(\log h + 2\log p)}{\sqrt{\omega p}}\frac{1}{\sqrt{\hat{V}_n}}\left\|\beta^* - \beta_{new}\right\|_2^2,$$

with probability at least $1 - c_3\exp(-c_4\omega\min((\nu'_{UB})^{-2}, 1))$. Further, c and c' are some positive constants; α'_{LB}, $(\tau^{\omega}_{LB})'$, ν'_{LB}, α'_{UB}, $(\tau^{\omega}_{UB})'$ and ν'_{UB} refer to the corresponding components for the new VAR process after the change point. Finally, by the condition $s(\log h + 2\log p) = o(\omega)$, we have $\sqrt{\frac{p\omega}{\hat{V}_n}}\frac{T_2}{p} \asymp \sqrt{\frac{\omega}{p}}\left\|\beta^* - \beta_{new}\right\|_2^2$.

- For term 3, we have

$$\begin{aligned}\left|\sqrt{\frac{p\omega}{\hat{V}_n}}\frac{T_3}{p}\right| &\leq \frac{2}{\sqrt{\hat{V}_n}}\sqrt{\frac{\omega}{p}}\left\|\beta^* - \beta_{new}\right\|_1 * \left\|\hat{\gamma}_\omega - \hat{\Gamma}_\omega\beta_{new}\right\|_\infty \\ &\leq 2\frac{\mathbb{Q}\left(\beta_{new}, \sigma^2\right)}{\sqrt{\hat{V}_n}}\sqrt{\frac{s(\log h + 2\log p)}{p}}\left\|\beta^* - \beta_{new}\right\|_2\end{aligned}$$

with probability at least $1 - c_5\exp\left[-c_6(\log h + 2\log p)\right]$ by sparsity assumption and Lemma 3, while c_5 and c_6 are some positive constant. With condition $\sqrt{\frac{s(\log h+2\log p)}{\omega}} = o(\|\beta^* - \beta_{new}\|_2)$, we have $|\sqrt{\frac{p\omega}{\hat{V}_n}}\frac{T_3}{p}| = o_p(\sqrt{\frac{\omega}{p}}\|\beta^* - \beta_{new}\|_2^2)$.

- We have that the absolute value of term 4 is less than or equal to

$$\begin{aligned}&\frac{2}{\sqrt{\hat{V}_n}}\sqrt{\frac{\omega}{p}}\omega^\eta p^\eta \max_{i,j',l}\left(\left|\frac{x^2_{i-l,j'}}{\omega^\eta p^\eta}\right|\right)\left\|\hat{\beta}_n - \beta^*\right\|_1\left\|\beta^* - \beta_{new}\right\|_1 \\ &\leq \frac{c}{\sqrt{\hat{V}_n}}\omega^\eta p^\eta s\sqrt{\frac{s(\log h + 2\log p)}{n}}\sqrt{\frac{\omega}{p}}\left\|\beta^* - \beta_{new}\right\|_2\end{aligned}$$

with probability at least $1 - c_7\exp[-c_8(\log h + 2\log p)] - c_9\exp(-c_{10}n\min\{\nu^{-2}_{LB}, 1\}) - 2\exp(-c_{11}p^\eta\omega^\eta + \log(\omega p h))$ for some positive constant c and some $\eta \in (0, \frac{1}{4})$. To get the inequality above, $\max_{i,j',l}\left(\left|\frac{x^2_{i-l,j'}}{\omega^\eta p^\eta}\right|\right)$ is bounded by a positive constant

with high probability by using the properties of Sub-Weibull distribution and the nature of stationary time series. This part is very similar to the second part of the proof of the Theorem 1. In addition, lemma 4 is applied to bound $\left\|\hat{\beta}_n - \beta^*\right\|_1$. With condition $\omega^\eta p^\eta \sqrt{\frac{s^3(\log h + 2\log p)}{n}} = o(\|\beta^* - \beta_{new}\|_2)$, we have $|\sqrt{\frac{p\omega}{\hat{V}_n}}\frac{T_4}{p}| = o_p(\sqrt{\frac{\omega}{p}}\|\beta^* - \beta_{new}\|_2^2)$.

Combining these four terms, we will have the inequality in Theorem 2 with probability at least 1 - $\epsilon_{n,p,\omega}$ by the union bound. We have $L_{t^*+h}^{(n,\omega)} = o_p(\sqrt{\frac{\omega}{p}}\|\beta^* - \beta_{new}\|_2^2)$ and $(L_{t^*+h}^{(n,\omega)})' = o_p(\sqrt{\frac{\omega}{p}}\|\beta^* - \beta_{new}\|_2^2)$, because we have (1) $\hat{V}_n \xrightarrow{p} \mathrm{Var}(\varepsilon_{1,1}^2)$ as $n \to \infty$ by the proof of Theorem 1, and (2) additional conditions for Theorem 2. This concludes the proof of Theorem 2.

S4 NUMERICAL STUDIES

In this section, we evaluate the performance of our algorithm using synthetic data generated by a VAR process. The primary metrics used for assessing the algorithm's effectiveness are the run length and detection delay, which are standard measures for online change point algorithms. These metrics have also been used in previous studies, such as Chen et al. (2022); Mei (2010); Xie and Siegmund (2013); Chan (2017). To compute the run length, we apply our algorithm to a data set without any change points and record the number of observations monitored before the first

alarm is raised. On the other hand, to determine the detection delay, we apply our algorithm to a data set containing a change point and record the distance between the location of the last observation read by the algorithm and the true location of the change point after the alarm is correctly triggered.

S4.1 Simulation A: Run Length

The majority of online change point detection algorithms offer parameters that allow practitioners to control the target average run length (ARL). The target average run length represents the expected number of observations or time steps required by the algorithm to raise an alarm when applied to a data sequence without any actual change points. The ARL is an essential measure to balance the algorithm's performance between being sensitive enough to detect changes promptly and avoiding excessive false alarms. In our algorithm, the target average run length is primarily influenced by the choice of parameter α. Specifically, setting α to be 1/1000 will result in a lower bound of 1000 for the ARL of our algorithm. For this simulation scenario, our focus is on exploring how to regulate the average run length by selecting an appropriate α, as well as investigating how the dimension of data and the size of training data affect the run length of our algorithm. In this simulation, we consider three different choices for the parameter α, namely 1/1000, 1/5000, and 1/10000. We estimate transition matrices and variances using training data with sizes n equal to

500, 1000, 1500, and 2000. Additionally, we vary the dimension of the data, setting it to be 10, 40, 70, and 100. For each combination of α, n, and p, we generate $10/\alpha + n$ data points from a lag-1 VAR process without any change points. After estimating the transition matrices and variances using n observations, we proceed to apply our algorithm on the remaining data. The algorithm is run with a pre-specified detection delay of ω set to 50 and h set to 1. We repeat this process 200 times, recording all the run lengths for each combination of parameters. The box plots of these run lengths are presented in Figure 1.

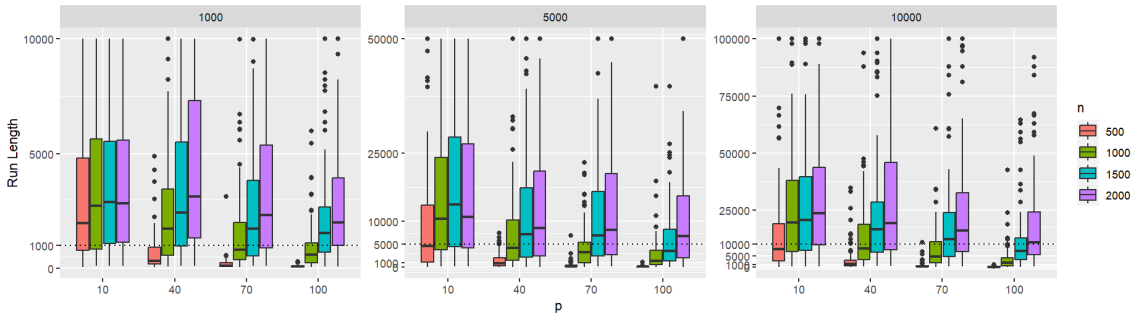


Figure 1: Simulation A: This plot displays the box plots representing the run lengths of our algorithm for different combinations of training data size n, data dimension p, and α. The values 1000, 5000, and 10000 correspond to the target ARL, which is controlled by setting α to 1/1000, 1/5000, and 1/10000, respectively.

As depicted in the plot, when the training sample size is adequately large, the run lengths of our algorithm are consistently lower bounded by $1/\alpha$ with high probability. However, when training data is limited, the run lengths may not reach the target

run length. Therefore, we recommend that practitioners set $1/\alpha$ to be equal to the length of the data that needs to be monitored when there is sufficient training data available. In cases where training data is limited, further decreasing the value of α might be a viable solution to reduce the probability of false alarms. Another noteworthy observation from this figure is that as the dimension of the data increases, the size of the training data set needed to maintain a satisfactory run length also increases.

S4.2 Simulation B: Detection Delay

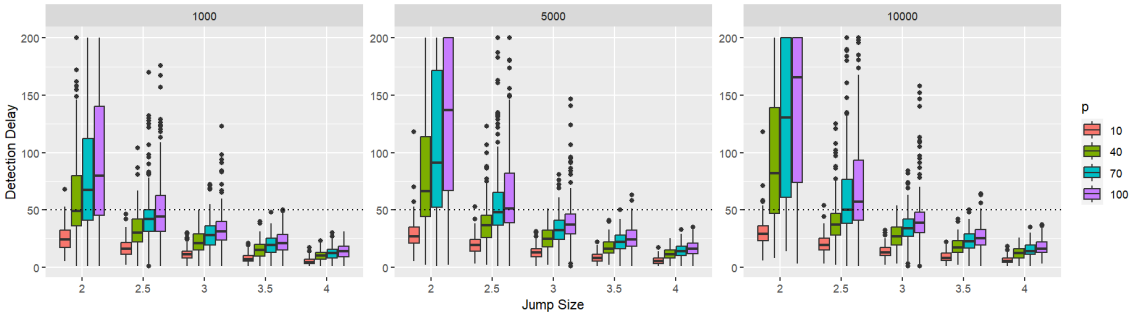


Figure 2: Simulation B: This plot displays the box plots representing the detection delays of our algorithm for different combinations of jump size, data dimension p, and α. The values 1000, 5000, and 10000 correspond to the target ARL, which is controlled by setting α to 1/1000, 1/5000, and 1/10000, respectively. The horizontal dashed line represents the pre-specified detection delay, denoted as ω.

Detection delay measures the time lag between the occurrence of a change point and the moment the algorithm successfully detects it. A shorter detection delay implies

that the algorithm can quickly identify and adapt to changes, which is critical in real-time systems where timely reactions are necessary to mitigate potential risks or capitalize on emerging opportunities. In this simulation, we explore how the detection delay of our algorithm is influenced by different choices of α, data dimension p, and the jump size of the change point. Specifically, we set α to three different values: 1/1000, 1/5000, and 1/10000, and vary the data dimension to 10, 40, 70, and 100, as well as the jump size to 2, 2.5, 3, 3.5, and 4. To focus solely on the detection delay and eliminate the impact of false alarms, we generate data points from a lag-1 VAR process with a total length of 2200. The change point is located at position 2000, which corresponds to the end of the training period. By doing so, we can consider the number of observations our algorithm reads before raising an alarm as the detection delay. We run our algorithm with a pre-specified detection delay set to 50 and recorded the corresponding detection delay. This process was repeated 200 times for each combination of parameters. The resulting detection delays were then summarized using box plots, as shown in Figure 2.

As depicted in the figure, when the jump size is large, the detection delay of our algorithm is consistently upper bounded by the pre-specified detection delay with high probability. This finding aligns with Corollary 2.1, confirming that the detection delay will be upper bounded by $\omega + h$ with high probability when the jump size is sufficiently large. On the other hand, when we choose a smaller value for α, the

detection delay of our algorithm increases. Although this effect is only pronounced when the jump size is small, it is still essential to select an appropriate α to strike a balance between the detection delay and the probability of false alarms in practical applications. Another observation from the figure is that as the dimension of the data increases, a larger jump size is required to achieve a small detection delay. This observation aligns with our assumption on the jump size, as introduced in Theorem 2.

S4.3 Simulation C: Choice of ω

The pre-specified detection delay can be regarded as a moving window that contains data points used to compute the test statistic for our algorithm. The selection of its size, denoted as ω, significantly impacts the performance of our algorithm in terms of both the probability of false alarms and the detection delay. Thus, in this simulation, our main objective is to investigate how various choices of ω influence the detection delay and early stop rate of our algorithm under different combinations of change point jump size and data dimension. For each combination of p, ω, and jump size, we generate a data set of 2600 data points using a lag-1 VAR process, with the change point occurring at time 2300. We then estimate the transition matrices and variances using the first 2000 data points and begin monitoring from that point onward. During the monitoring process, if our algorithm raises a false alarm before reaching the true change point, we consider it an early stop and record this occurrence. On the other

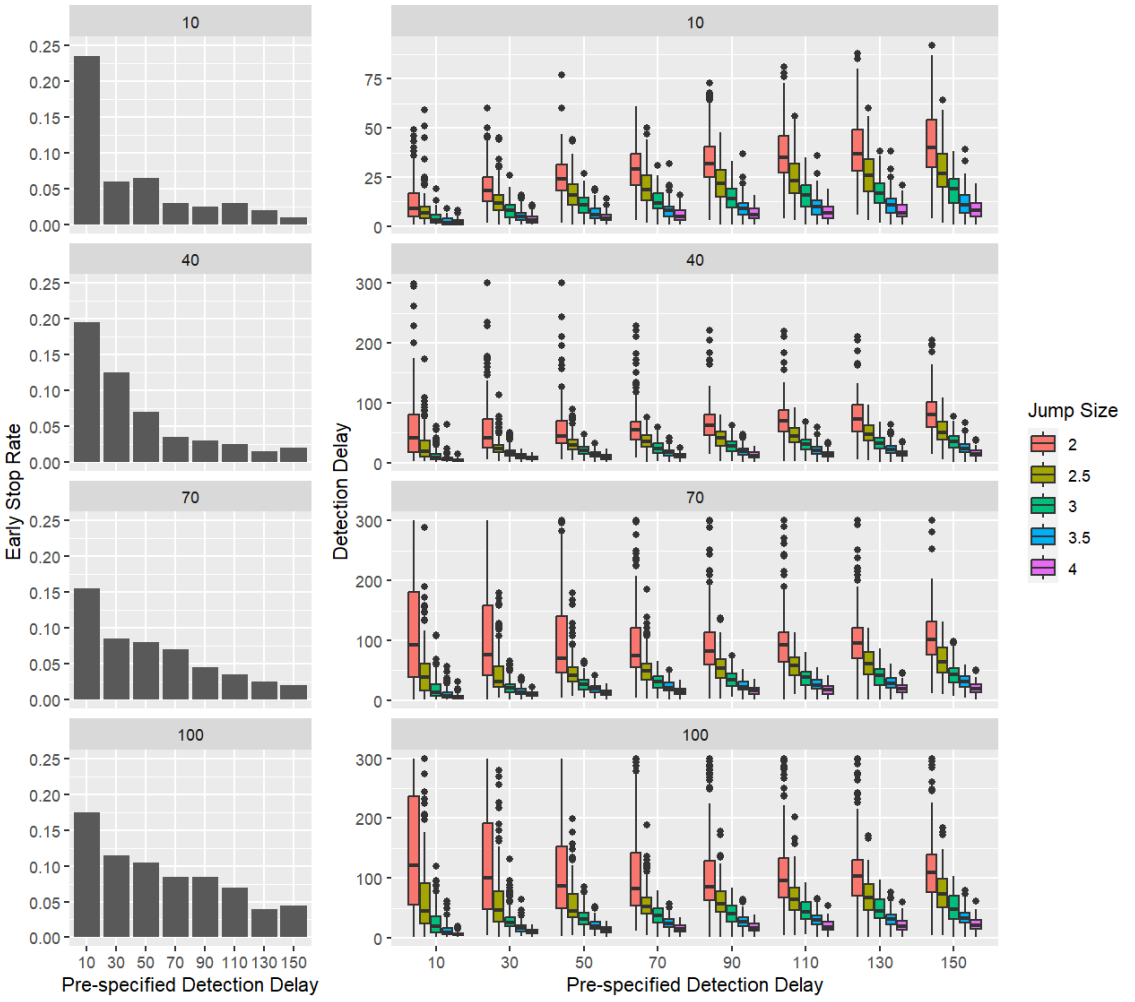


Figure 3: Simulation C: The plot on the left summarizes the early stop rates, while the plot on the right presents the detection delays. For each grid in the plots, the dimension of data is set to 10, 40, 70, and 100.

hand, if the algorithm raises an alarm after the true change point, we record the detection delay. The α is set to 1/1000 in all combinations. This entire process is repeated 200 times. The early stop rate is calculated by dividing the number of early

stops by 200 and all detection delays are recorded for each combination. The results are presented and summarized in Figure 3.

Figure 3 demonstrates that larger values of ω are preferable for effectively controlling the false alarm rate. This is reflected in the early stop rate shown in the left panel. However, selecting ω becomes more intricate when aiming to minimize detection delay, as it is highly sensitive to the jump size and the data dimensionality. In practice, this dependence makes it challenging to derive an optimal data-driven approach for selecting ω when the true changes and jump sizes are unknown. According to the conditions specified in the theoretical results, ω should scale as $c\log(hp^2)$ for some constant $c > 0$. Carefully reviewing the results in Figure 3, for practical implementation, $\omega = 10\log(hp^2)$ is recommended. This choice results in ω values of 46, 74, 85, and 92 for dimensions $p = 10, 40, 70$, and 100, respectively. These values effectively maintain a low early stop rate while minimizing detection delay for small jump sizes (e.g., jump size = 2). As shown in the figure, the impact of ω on detection delay is more pronounced for smaller jump sizes. Although this choice may not yield the optimal delay for larger jump sizes, it incurs only a minor increase in detection delay relative to the optimal ω. However, when the training sample size is small, adjusting ω has limited effect on enhancing detection quality. Moreover, when the estimation of transition matrices is imprecise, a larger window size can introduce more error into the test statistic, which aligns with the condition in Theorem 1, where

$\omega = o(n)$. In practical settings, a training sample size approximately 10–15 times the window size ω is advisable, which can serve as a reference for selecting ω when training data is limited.

S4.4 Simulation D: Effectiveness of Refinement

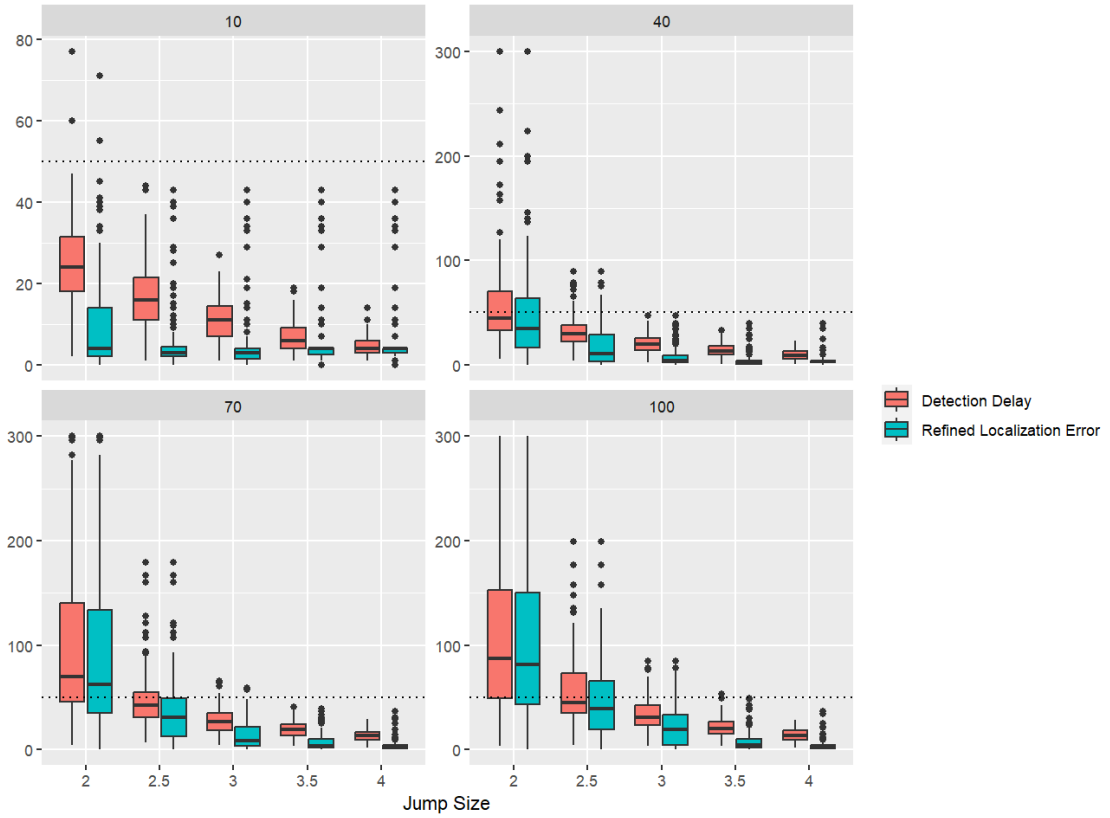


Figure 4: Simulation D: This plot corresponds to box plots of detection delays and refined localization errors. For each grid in the plots, the dimension of data is set to 10, 40, 70, and 100. The horizontal dashed line represents the pre-specified detection delay, denoted as ω.

In this simulation, our primary focus is to assess the effectiveness of the proposed

change point localization refinement process, which was introduced in Section 5. Before delving into the simulation setup, we first introduce a few terms related to the refinement process. The first term is the "refine size," which is defined as the ratio between the new pre-specified detection delay and the old pre-specified detection delay. For instance, if the refine size is set to 0.1, and the original ω is 50, then the value of ω' used in the refinement process will be 5. The second term is the "refined localization error," representing the distance between the refined location of the estimated change point and the true location of the change point. Formally, if an alarm is raised at time $\hat{t}$ (i.e., $\left|\hat{T}_{\hat{t}}^{(n,\omega)}\right| > \Phi(1-\alpha/2)$), and the alarm is not a false alarm, then the last observation read by our algorithm will be at $\tilde{t} = \hat{t} + \omega$. In this case, if the true change point is located at t^* and the refined location of the estimated change point is at $\hat{\hat{t}}$, then the detection delay and refined localization error will be $\tilde{t} - t^*$ and $|\hat{\hat{t}} - t^*|$, respectively. Similar to the previous simulations, we consider various values for the dimension of data and the jump size of the change point. Additionally, we introduce the refine size, which takes values of 1/2, 1/5, 1/10, and 1/50. The data points are generated with a total length of 2600, and the change point is located at position 2300. We estimate the transition matrices and variances using the first 2000 observations. Subsequently, we apply our algorithm to the remaining data points with α set to 1/1000 and ω set to 50, both with and without the confirmation step introduced at the end of Section 5. This entire process is repeated 200 times, during

which we calculate the early stop rate and record the detection delays and refined localization errors for all combinations.

Figure 4 presents the summarized box plots for detection delays and refined localization errors when the refine size is set to 0.1 for all combinations of data dimensions and jump sizes without the confirmation step. As depicted in the figure, the refinement process effectively reduces the localization error, specially when the jump size is relatively large. As illustrated in Figure 5, the confirmation step notably decreases the possibility of false alarms. Thus, the confirmation step can be considered as an option to minimize false alarm probabilities. To provide practical guidance on the choice of refine size based on the window size recommendation $\omega = 10\log(hp^2)$ in Section S4.3, we conducted a sensitivity analysis. Specifically, in each experimental iteration, we simulated scenarios in which alarms were triggered using $\omega = 10\log(hp^2)$ observations, with the true change point positioned at the center of this larger window. The refinement was then applied using refine sizes $(0.1, 0.2, 0.3, 0.4, 0.5)$. This analysis was conducted across various data dimensions ($p = 10, 40, 70, 100$) and jump sizes (2, 3, 4), with the goal of identifying the refine size that minimized refined localization error. After 100 repetitions, the average optimal refine sizes consistently clustered around 0.1 to 0.2, as shown in Table 1. Based on these results, we recommend using 0.15 for practical applications.

Jump Size	$p = 10$	$p = 40$	$p = 70$	$p = 100$
2	0.131	0.138	0.149	0.157
3	0.116	0.126	0.137	0.157
4	0.113	0.107	0.119	0.116

Table 1: Average of the optimal refine sizes for different dimensions and jump sizes.

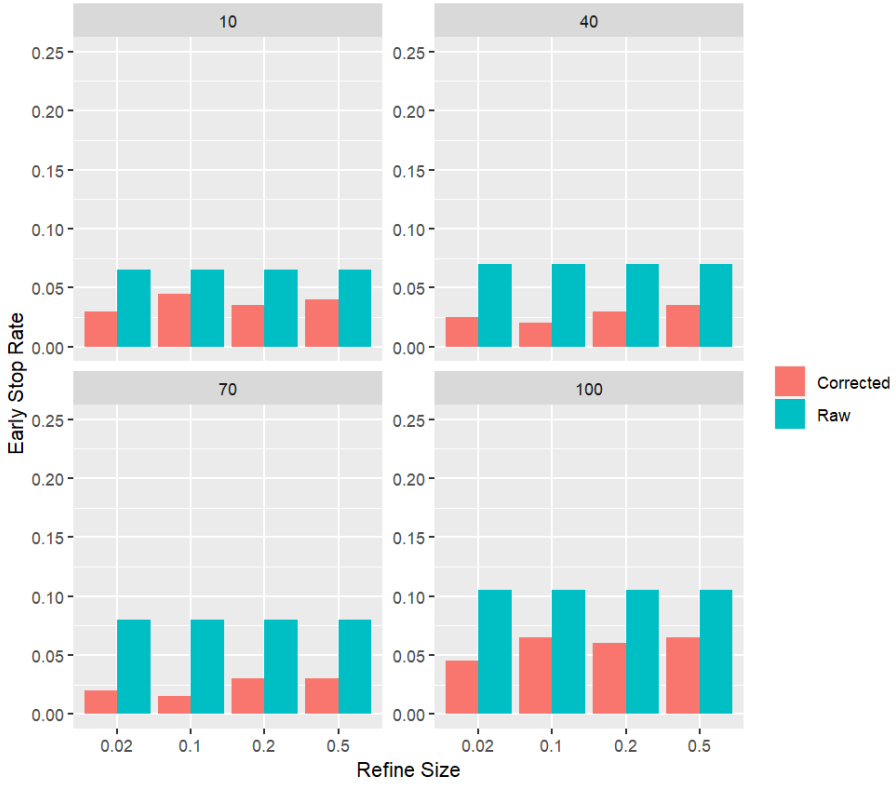


Figure 5: Simulation D: This plot provides a summary of the early stop rates for all combinations of refine sizes, data dimensions and whether confirmation is used or not. For each grid in the plots, the dimension of data is set to 10, 40, 70, and 100.

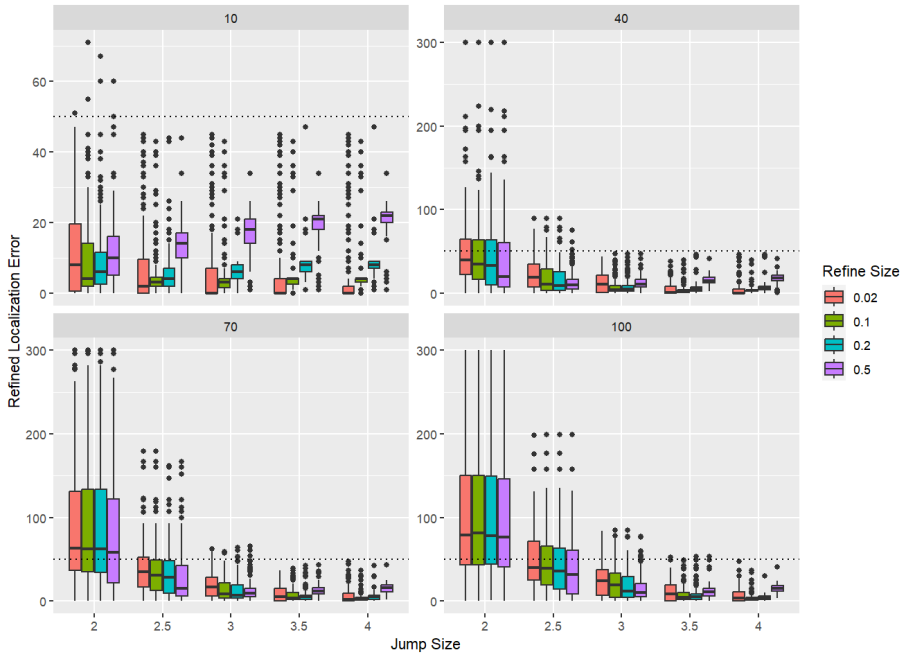


Figure 6: Simulation D: This plot provides a summary of the refined localization errors for all combinations of refine sizes and data dimensions. For each grid in the plots, the dimension of data is set to 10, 40, 70, and 100. The horizontal dashed line represents the pre-specified detection delay, denoted as ω.

S4.5 Simulation E: Multiple Change Point Detection

In this simulation, we evaluate the performance of our method in terms of F1 score when dealing with multiple change points in low-dimensional ($p = 10$) and high-dimensional ($p = 100$) setups. For a range of jump sizes, we generate VAR time series of size 6900 with change points located at positions 2300 and 4600. Specifically, for the first 2300 data points, we use the transition matrix $0.8 * I_p$. The subsequent

2300 data points are generated using a new transition matrix with a certain jump size compared to the previous one. Finally, the last 2300 data points are generated again using the transition matrix $0.8 * I_p$. We implement Algorithm 1 sequentially, as mentioned in Section 6, and consider the refined estimated change points within 2300 ± 10 and 4600 ± 10 as true positives. In each repetition, we calculate the following metrics: $F1\,Score = \frac{2\times TP}{2\times TP+FP+FN}$ where TP, FP, and FN represent true positives, false positives, and false negatives, respectively. We then calculate the averages among the 100 repetitions for different jump sizes. These metrics are commonly used in assessing detection algorithms in scenarios with multiple change points such as in Bai and Safikhani (2023). The results are summarized in Table 2. Under both low-dimensional and high-dimensional setups, we set $n = 2000$, $\omega = 50$, $\alpha = 0.0001$, and $h = 1$ for our algorithm. To reduce the number of false alarms, we perform the confirmation step as introduced in Section 5. As shown in Table 2, our algorithm exhibits strong capabilities in handling data with multiple change points, especially when the jump size is large, under both low-dimensional and high-dimensional setups.

S4.6 Simulation with Variance Heterogeneity

This section provides simulation results for the average run length and detection delay of our algorithm under the same setup as in Simulation A and B, with $\alpha = 1/1000$. However, in this simulation, the diagonal entries of the covariance matrix for the

Table 2: Simulation E: The F1 score for our algorithm is assessed in a multiple change point scenario. We consider jump sizes (JS) ranging from 2 to 4.5 under both low-dimensional ($p = 10$) and high-dimensional ($p = 100$) setups.

	JS = 2.0	JS = 2.5	JS = 3.0	JS = 3.5	JS = 4.0	JS = 4.5
$p = 10$	0.73	0.88	0.97	0.98	0.99	0.99
$p = 100$	0.06	0.26	0.45	0.66	0.88	1.00

noise is randomly generated from a uniform distribution ranging from 0.5 to 1.5 to assess our algorithm's performance with variance heterogeneity. The test statistic is calculated as described in Remark 1. Satisfactory performance is achieved for both average run length and detection delay in this scenario, as shown in Figure7.

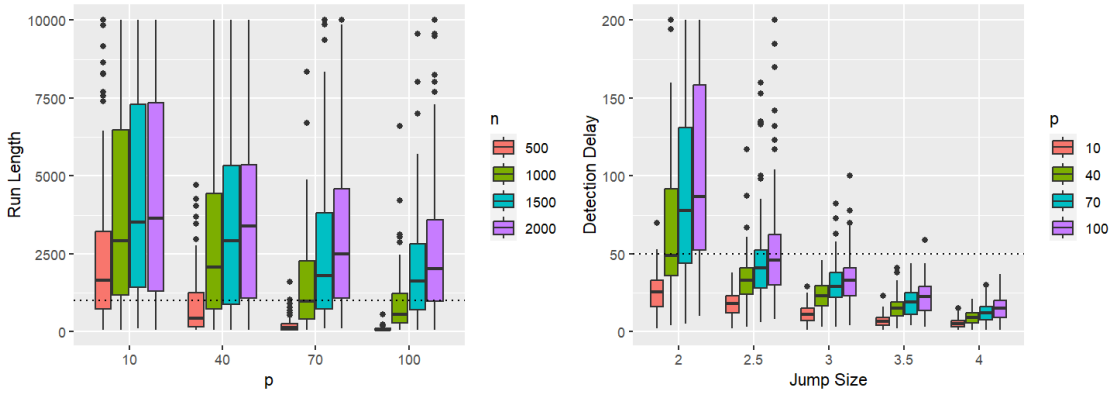


Figure 7: Simulation with variance heterogeneity: The covariance matrix's diagonal elements for errors are randomly selected from a uniform distribution ranging from 0.5 to 1.5. The rest of the settings align with those of Simulation A and B, with the value of α set to 1/1000.

S4.7 Numerical Comparison in High-Dimensional Settings

This section supplements Section 7 by extending the numerical comparison to high-dimensional settings with $p = 100$. In addition to this modification, we increased the training sample size from 500 to 2000 and adjusted the jump sizes from 2 and 3 to 3 and 4 to accommodate the higher dimensionality. The results, shown in Figure 8, demonstrate that our proposed algorithm performs comparably to the case when $p = 10$. Notably, the algorithm remains competitive with alternative methods when the data is generated without a VAR structure and continues to outperform all competing methods when the data is generated with a VAR structure. Additionally, we observed that the TSL method (Qiu and Xie, 2022) required an excessive amount of memory (over 8,388,608 GB) to allocate the necessary vectors in the larger dimensional setting. As a result, we were unable to obtain results for the TSL method in this scenario, and it is therefore not included in the comparison.

S4.8 Robustness to Time-Varying Transition Matrices

To illustrate the robustness of the proposed algorithm to small time-varying effects, we conducted a set of simulations with a transition matrix that varies slightly over time. These simulations, summarized in Figure 9, involved introducing time-varying behavior in three specific entries of the transition matrix. In the left panel of the figure, the entry in row 2, column 2 oscillates between 0.5 ± 0.3 with a period of 500.

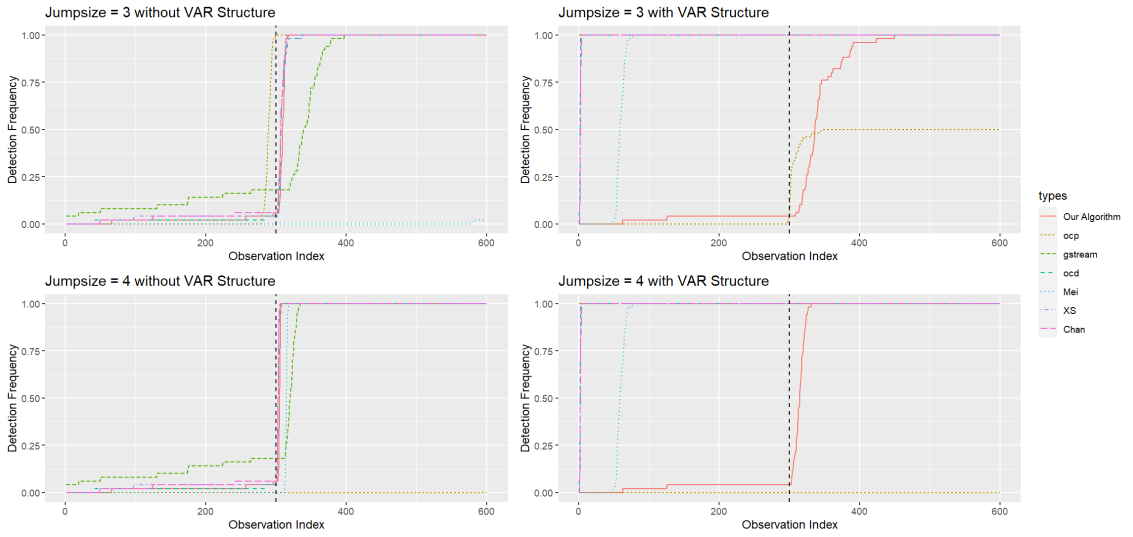


Figure 8: Summary of detection frequencies for all algorithms. The black dashed vertical line indicates the location of the true change point. An ideal algorithm would demonstrate a detection frequency of zero before the line and achieve one immediately after the line.

The other two time-varying entries oscillate similarly but start from different initial values. These oscillations persist throughout the simulation, even after change points. We varied the amplitude of oscillation across different runs, testing values of 0, 0.1, 0.2, and 0.3, where 0 represents no time-varying effect. Two sets of simulations were conducted to examine the algorithm's performance under these conditions. The first set of simulations, shown in the middle panel, evaluated how changes in amplitude affect the run length. With settings similar to those in S4.1—using $\alpha = 1/1000$, $n = 500$, $p = 10$, and $\omega = 50$—the results show that, for small oscillation amplitudes, the algorithm maintains control over the target ARL, keeping it above $1/\alpha$. However,

as the amplitude increases, the run length decreases, indicating that larger time-varying effects are more likely to be misidentified as true changes, leading to a higher false alarm rate. The second set of simulations, shown in the right panel, analyzed the effect of oscillation amplitude on detection delay. Under settings similar to those in S4.2 (with $\alpha = 1/1000$, $n = 500$, $p = 10$, and a jump size of 2), the results indicate that the detection delay remains relatively stable, even as oscillation amplitude increases. This demonstrates that the detection delay is less sensitive to moderate time-varying effects. In summary, while the full extension of this method to handle time-varying transition matrices lies beyond the scope of this study, these simulations show that the proposed algorithm is robust to small time-varying effects. Future research will further explore this aspect. For now, the focus of this work remains on the piecewise constant setting.

S4.9 Robustness to Complex Transition Matrix Structures

The proposed algorithm is capable of handling more complex transition matrices, provided there is a sufficiently large training sample size to enable accurate estimation. To illustrate its robustness, we conducted additional simulations using a low-rank plus sparse structure for the transition matrix, following the setup described in Bai et al. (2020). In this simulation, the transition matrix includes a low-rank component with a rank of 2, resulting in a structure that is no longer sparse. The simulation

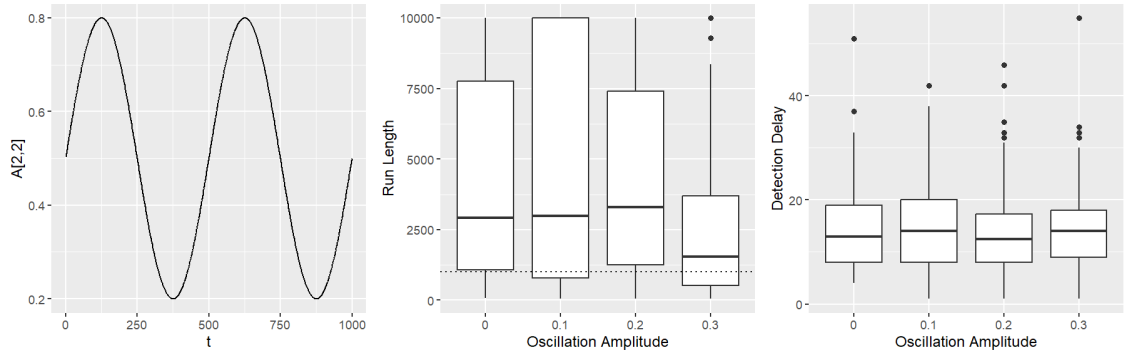


Figure 9: Simulation with Time-Varying Transition Matrices. (Left) Illustration of how specific entries in the transition matrix vary over time. The entry at row 2, column 2 oscillates between 0.5 $\pm$ 0.3 with a period of 500. (Middle) Effect of oscillation amplitude on the run length. The dashed line indicates the target ARL of $1/\alpha$, with run lengths expected to exceed this threshold. (Right) Effect of oscillation amplitude on the detection delay.

parameters were set to $p = 25$, $\alpha = 1/1000$, and $\omega = 50$, and Figure 10 summarizes the results. As shown in Figure 10, the false alarm rate remains well-controlled when the sample size is sufficiently large. However, more observations are required to ensure that the average run length (ARL) meets the target threshold of $1/\alpha = 1000$ when handling complex transition matrices. Notably, the detection delay appears to be more sensitive to the magnitude of the jump than to the structure of the transition matrix itself. The settings for these simulations align with those used in Sections S4.1 and S4.2, where we analyze the run length and detection delay under different scenarios. While these results demonstrate the algorithm's capability to handle more intricate transition matrix structures, we do not pursue a rigorous theoretical

analysis of this aspect here. Instead, we aim to provide an empirical illustration of the algorithm's robustness, leaving a deeper theoretical investigation for future work.

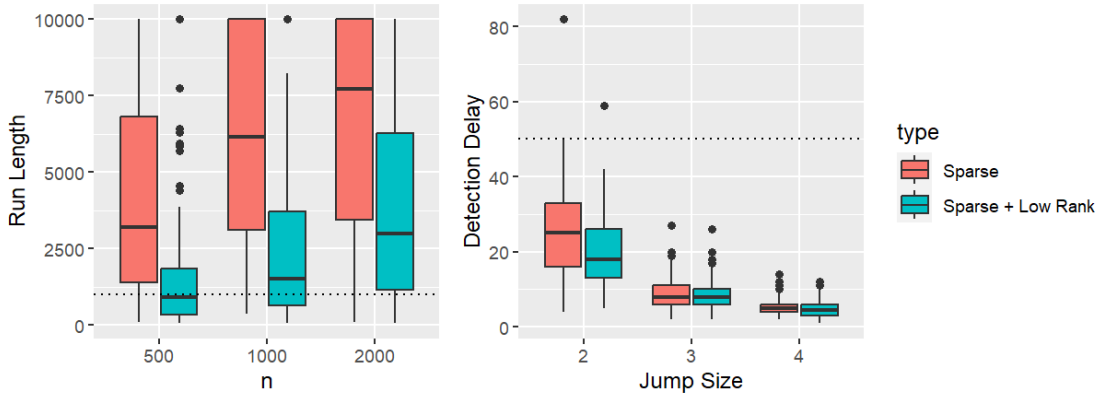


Figure 10: Performance of the Proposed Algorithm in Terms of Run Length and Detection Delay with Sparse vs. Sparse + Low-Rank Transition Matrices.

S5 ADDITIONAL RESULTS FOR REAL DATA ANALYSIS

This section provides additional details for the S&P 500 real data experiment and presents results from the real data experiment conducted on EEG data.

S5.1 Additional Details for S&P 500 Data

To establish a reference for the anomaly period, the return volatility is used, a standard measure of return dispersion (also used as a reference in Keshavarz et al. (2020)). Let $x_{t,j}$ represent the daily log return for stock j at time t, and let std(x)

denote the standard deviation of x. The return volatility of stock j at time t is estimated using the formula $z_{t,j} = \text{std}(x_{t,j}, \ldots, x_{t+\omega-1,j})$. The average $z_{t,j}$ across all 186 stocks is then computed, and this average return volatility is rescaled for visualization. The rescaled value is shown as the black line in Figure 11. A high average return volatility generally indicates an increased likelihood of a change point. Figure 11 also shows the locations of alarms (red vertical lines) and the estimated onsets (black vertical lines) of alarm clusters.

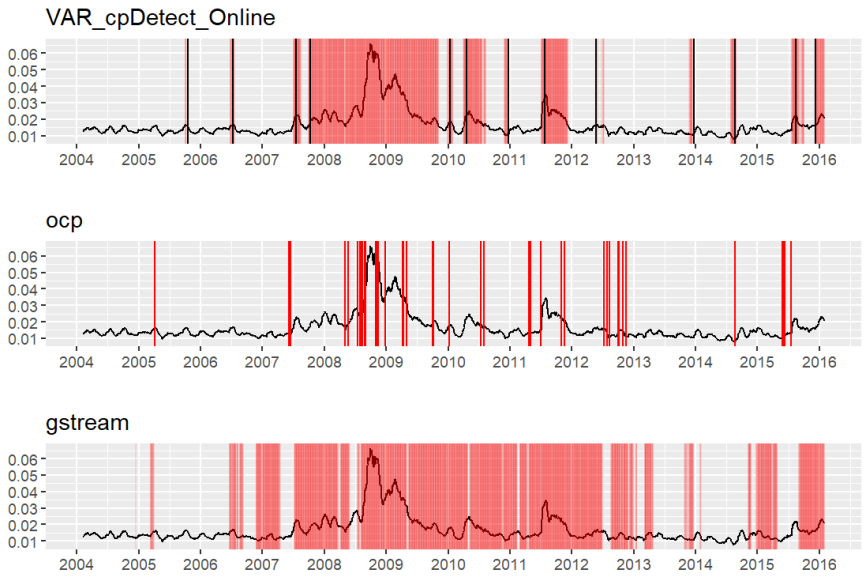


Figure 11: Experiment results on S&P 500 data: The black line represents the rescaled average return volatility, while the red lines correspond to alarm locations for (top) VAR_cpDetect_Online, (middle) ocp, and (bottom) gstream.

S5.2 Real Data Experiment on EEG Data

For the EEG data, this experiment aims to detect and raise an alarm indicating an impending seizure, occurring around $t = 85$, as confirmed by neurologists and validated by offline change point detection methods in Section 8 of Safikhani and Shojaie (2022). The data was collected from 18 EEG channels over a 227.68-second duration. To focus on seizure onset, data after $t = 150$ seconds were removed. The final dataset comprises 1500 data points over 150 seconds, with a dimension of $p = 18$. The first 300 data points were designated as historical data, with parameters $\omega = 30$ and $\alpha = 1/2000$ used in our method. The hyperparameters for the baseline algorithms were selected as described in Section 7. All methods were applied to the entire dataset without halting upon alarm, and the alarm locations are documented in Figure 12. As shown in the top panel of Figure 12, the alarms raised by the proposed algorithm form two clusters, indicating periods where the patient's brain activity deviates from baseline, potentially signaling seizure activity. The estimated start times (solid vertical black lines) of these clusters are at $t = 56.3$ (lasting 1.4 seconds) and $t = 81.5$ (lasting until the end of the data). The proposed algorithm requires 19 and 25 additional observations (detection delay) before issuing these alarms. Both estimates occur before the confirmed seizure onset at $t = 85$, suggesting that early shifts in brain electrical activity may be detectable in advance, consistent with findings in Ombao et al. (2005). Similarly, in another study (Safikhani and

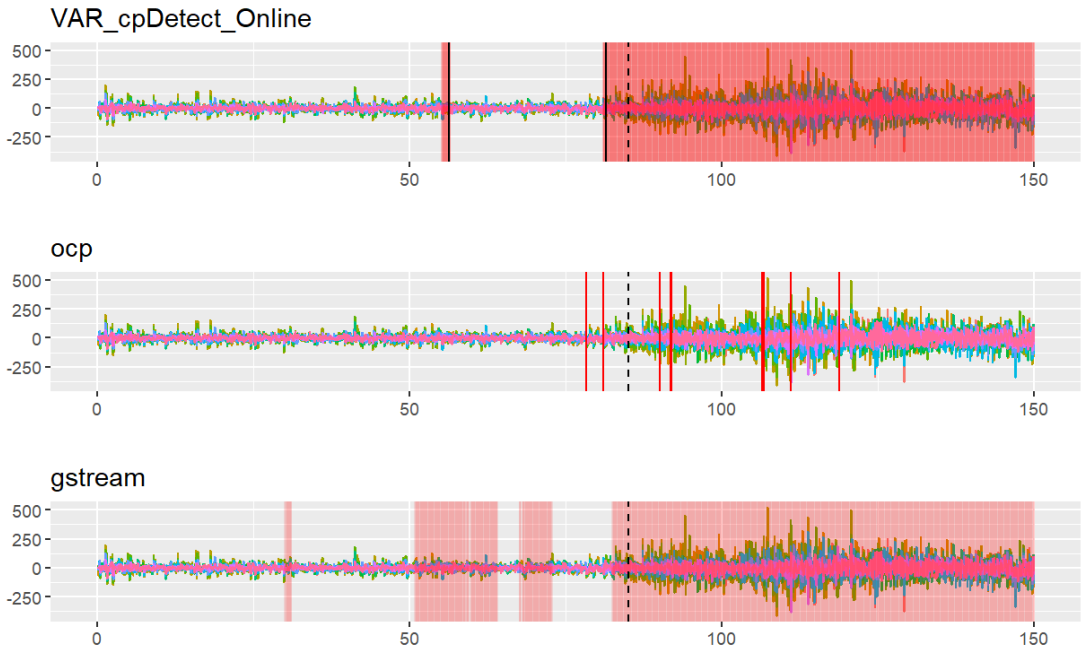


Figure 12: Experiment results on EEG data: The red lines indicate alarm locations for (top) VAR_cpDetect_Online, (middle) ocp, and (bottom) gstream.

Shojaie, 2022), an offline CPD algorithm based on a VAR model estimated a change point at $t = 83$, also slightly before the seizure began, further supporting the idea that changes in brain activity may be detectable prior to the seizure's onset. The middle panel shows that the ocp method detected two change points at $t = 78.3$ and $t = 81.0$, both preceding the seizure onset, with the latter closely aligning with our algorithm's estimate. The bottom panel indicates that the gstream method raised alarms forming four clusters, with start points at $t = 30.1$ (lasting 1.1 seconds), $t = 50.9$ (lasting 13.3 seconds), $t = 67.6$ (lasting 5.3 seconds), and $t = 82.5$ (lasting until the end).

The final cluster occurs slightly before $t = 85$, agreeing with our algorithm's results; however, the gstream method triggers numerous alarms before the seizure, limiting its practical utility for early warning. The average execution times were 1.77 seconds for our method, 9.50 seconds for ocp, and 27.14 seconds for gstream.

S6 SEQUENTIAL UPDATING FOR TRANSITION MATRICES

In this section, we examine the performance of a sequential updating approach (Messner and Pinson, 2019) for estimating transition matrices in high-dimensional VAR models. Sequential updating allows for efficient integration of new data, improving the estimation of transition matrices for the proposed algorithm when no alarm has been raised during monitoring. We discuss the benefits and limitations of sequential updating in various scenarios and present simulation results to illustrate its impact on estimation accuracy.

S6.1 Advantages and Limitations

Sequential updating provides a practical method to update the transition matrix estimates as new observations arrive. Instead of re-estimating the transition matrices from scratch using both old and new data, which incurs high time and space complexity, this approach applies a cyclic coordinate descent algorithm at each time step. By using

the previous step's coefficient estimates as starting values, it avoids the computational burden associated with full re-estimation. When the forgetting factor is set to $\nu = 1$, this approach efficiently updates the transition matrices without requiring all historical data. The detailed procedure can be found in Messner and Pinson (2019), particularly in Equations (10)–(14).

Incorporating sequential updating into the proposed algorithm is particularly advantageous when the training data is very limited. In such cases, it allows the transition matrix estimates to improve as additional observations are gathered, provided no alarm is raised. This can help the algorithm reduce false alarm rates and increase power, as the accuracy of the transition matrix estimates improves with more data. However, as the size of the initial training data grows, the relative benefits of sequential updating decrease.

One limitation of sequential updating is that it can introduce estimation error at the beginning of the process, which may increase the likelihood of false alarms. This issue is particularly critical in real-time applications where accuracy is essential. As a result, while sequential updating is valuable in cases with limited training data, its direct application may not be suitable for all scenarios, especially when minimizing false alarm rates is crucial.

S6.2 Simulation Study

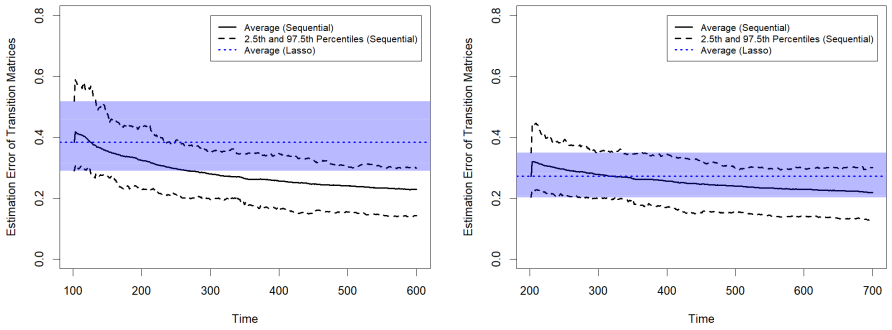
To illustrate the effects of sequential updating, we conducted simulations using data generated from a VAR process with transition matrix A and dimension $p = 10$. For each repetition, the initial estimate of the transition matrix was obtained using the regularization method (Basu and Michailidis, 2015) on training data of varying lengths, followed by sequential updates as new observations were collected.

Figure 13 shows the estimation error $\|A - \hat{A}\|_2$ with and without sequential updating. In this figure, the solid black line represents the average estimation error using sequential updating, with black dotted lines indicating the 2.5th and 97.5th percentiles. The blue dotted line represents the average estimation error when only the initial training data is used, with the shaded area showing the 2.5th and 97.5th percentile range.

The results show that sequential updating initially increases the estimation error, which may temporarily raise the false alarm probability. As the training sample size grows, however, the advantage of sequential updating diminishes, and the overall estimation error converges with that of the non-updated estimates.

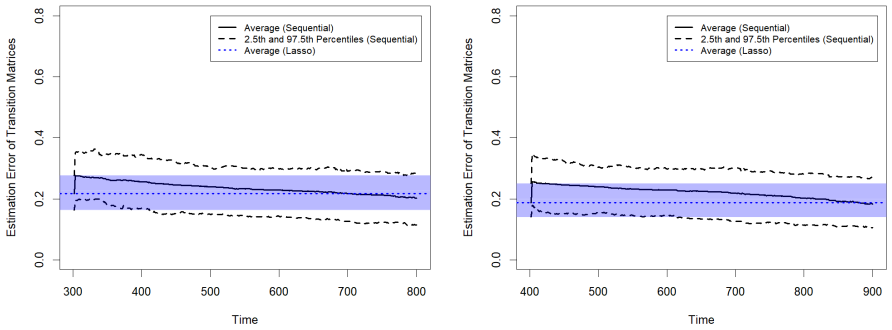
The simulation results suggest that sequential updating is advantageous in situations with limited training data, providing an efficient way to incorporate new data and improve estimation accuracy. However, as the amount of data increases, the benefits

of sequential updating wane, and its initial estimation error may contribute to a higher false alarm risk. Therefore, while sequential updating is effective in specific scenarios, we advise caution in applying it directly in cases where minimizing false alarms is a priority. Integrating sequential updating with the proposed algorithm presents an interesting but challenging direction for future research.



(a) Training data length = 100

(b) Training data length = 200



(c) Training data length = 300

(d) Training data length = 400

Figure 13: Comparison of estimation errors between the sequential update method and estimates based only on training data

S7 POST-CHANGE ANALYSIS

Identifying which variables undergo shifts after a change point, especially in high-dimensional contexts, is important yet challenging due to the limited number of post-change samples. When the post-change sample size is small, estimating the new model parameters reliably becomes difficult, complicating efforts to pinpoint which variables have shifted. Even with an accurately detected change point, a limited number of post-change observations can greatly reduce the reliability of diagnostic analysis. A straightforward approach might be to estimate the transition matrices before and after the change using Lasso, then compare these estimates. However, the bias inherent in Lasso makes it infeasible to directly infer which components of the transition matrices have changed. To address this, we recommend applying an online debiasing technique (Deshpande et al., 2023) both before and after the change point. This method debiases the Lasso estimates and allows for constructing confidence intervals (CIs) for the entries of the VAR transition matrices. By constructing CIs for the differences between the debiased estimates of the transition matrices before and after the change, we can identify which entries are likely to have changed. If the CI for a given entry excludes zero, we can infer a significant shift in that entry. To validate this approach, we conducted simulations with two groups of observations with $p = 10$ variables—one representing data before the change (with transition matrix A) and the other representing data after the change (with transition matrix A^*).

The pre-change sample size was fixed at $n_0 = 500$, allowing for accurate estimation, while the post-change sample size n_1 varied among 100, 200, and 300. The two transition matrices differed at six specific entries: (1,1), (2,2), (10,10), (3,7), (6,4), and (8,4). For each entry in the difference matrix $D = A - A^*$, we constructed CIs using debiased Lasso estimates and calculated their coverage rates of zero over 100 repetitions. As shown in Figure 14, entries with no changes maintain a zero coverage rate around 0.95, while entries with changes rarely include zero as the post-change sample size increases, accurately identifying the shifts. Although identifying these changes benefits from a moderate number of post-change samples, the CPD algorithm can continue collecting observations to improve diagnostic accuracy.

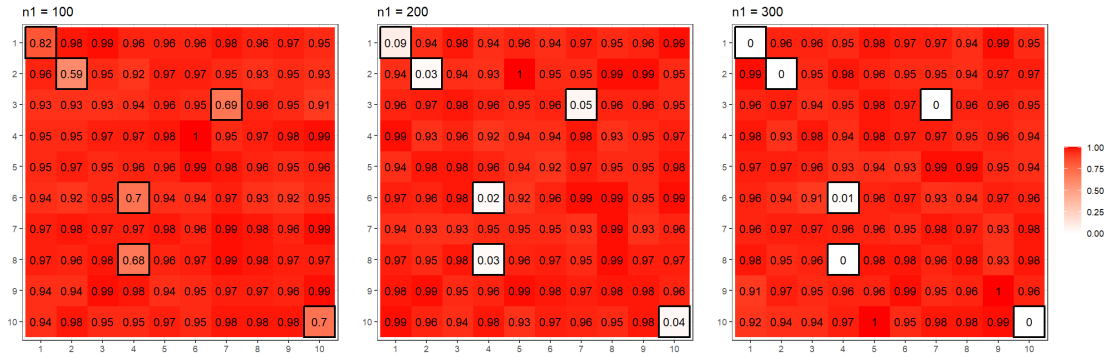


Figure 14: Coverage rates of zero for confidence intervals of the entries in $A - A^*$

References

Bai, P., A. Safikhani, and G. Michailidis (2020). Multiple change points detection in low rank and sparse high dimensional vector autoregressive models. *IEEE Transactions on Signal Processing 68*, 3074–3089.

Bai, Y. and A. Safikhani (2023). A unified framework for change point detection in high-dimensional linear models. *Statistica Sinica 33*, 1–28.

Basu, S. and G. Michailidis (2015). Regularized estimation in sparse high-dimensional time series models. *The Annals of Statistics 43*(4), 1535–1567.

Billingsley, P. (2013). *Convergence of probability measures*. John Wiley & Sons.

Chan, H. P. (2017). Optimal sequential detection in multi-stream data. *Annals of Statistics*.

Chen, Y., T. Wang, and R. J. Samworth (2022). High-dimensional, multiscale online changepoint detection. *Journal of the Royal Statistical Society Series B: Statistical Methodology 84*(1), 234–266.

Deshpande, Y., A. Javanmard, and M. Mehrabi (2023). Online debiasing for adaptively collected high-dimensional data with applications to time series analysis. *Journal of the American Statistical Association 118*(542), 1126–1139.

Keshavarz, H., G. Michailidis, and Y. Atchadé (2020). Sequential change-point detection in high-dimensional gaussian graphical models. *The Journal of Machine Learning Research 21*(1), 3125–3181.

Loh, P.-L. and M. J. Wainwright (2012). High-dimensional regression with noisy and missing data: Provable guarantees with nonconvexity. *The Annals of Statistics 40*(3), 1637–1664.

Lütkepohl, H. (2005). *New introduction to multiple time series analysis*. Springer Science & Business Media.

Mei, Y. (2010). Efficient scalable schemes for monitoring a large number of data streams. *Biometrika 97*(2), 419–433.

Messner, J. W. and P. Pinson (2019). Online adaptive lasso estimation in vector autoregressive models for high dimensional wind power forecasting. *International Journal of Forecasting 35*(4), 1485–1498.

Montgomery, D. C. and G. C. Runger (2010). *Applied statistics and probability for engineers*. John Wiley & Sons.

Ombao, H., R. Von Sachs, and W. Guo (2005). Slex analysis of multivariate nonstationary time series. *Journal of the American Statistical Association 100*(470), 519–531.

Qiu, P. and X. Xie (2022). Transparent sequential learning for statistical process control of serially correlated data. *Technometrics 64*(4), 487–501.

Ross, S. M. (2014). *A first course in probability*. Pearson.

Safikhani, A. and A. Shojaie (2022). Joint structural break detection and parameter estimation in high-dimensional nonstationary var models. *Journal of the American Statistical Association 117*(537), 251–264.

Van der Vaart, A. W. (2000). *Asymptotic statistics*, Volume 3. Cambridge university press.

Vladimirova, M., S. Girard, H. Nguyen, and J. Arbel (2020). Sub-weibull distributions: Generalizing sub-gaussian and sub-exponential properties to heavier tailed distributions. *Stat 9*(1), e318.

Wong, K. C., Z. Li, and A. Tewari (2020). Lasso guarantees for β-mixing heavy-tailed time series. *The Annals of Statistics 48*(2), 1124–1142.

Xie, Y. and D. Siegmund (2013). Sequential multi-sensor change-point detection. In *2013 Information Theory and Applications Workshop (ITA)*, pp. 1–20. IEEE.