Student's t VAR Modeling With Missing Data Via Stochastic EM and Gibbs Sampling

Rui Zhou, Junyan Liu, Sandeep Kumar, and Daniel P. Palomar, *Fellow, IEEE*

***Abstract*—The vector autoregressive (VAR) models provide a significant tool for multivariate time series analysis. Owing to the mathematical simplicity, existing works on VAR modeling are rigidly inclined towards the multivariate Gaussian distribution. However, heavy-tailed distributions are suggested more reasonable for capturing the real-world phenomena, like the presence of outliers and a stronger possibility of extreme values. Furthermore, missing values in observed data is a real problem, which typically happens during the data observation or recording process. Although there exist numerous works on VAR modeling with heavy-tailed distributions, they assume the availability of complete data and are not applicable in the presence of missing data. In this paper, we propose an algorithmic framework to estimate the parameters of a VAR model with heavy-tailed Student' s t distributed innovations from incomplete data based on the stochastic approximation expectation maximization (SAEM) algorithm coupled with a Markov Chain Monte Carlo (MCMC) procedure. We propose two fast and computationally cheap Gibbs sampling schemes, both based on MCMC procedure. The algorithms developed are effective in capturing the heavy-tailed phenomenon and being robust against outliers and missing data. In addition, owing to their low computational complexity, the algorithms are amenable for high-dimensional and big data applications. Extensive experiments with both synthetic data and real financial data corroborate our claims.**

***Index Terms*—Chain monte carlo (MCMC), heavy-tailed innovations, missing values, SAEM, Markov, VAR model.**

I. Introduction

THE autoregressive process is a simple mathematical structure widely used in the study of time series data [1]. A univariate autoregression is a single-equation, single-variable linear fit in which the present value of a variable is explained by its own lagged values. On the other hand, a vector autoregression (VAR) is a system of N-equation, N-variable linear model in which each variable is in turn explained by its own lagged values, plus current and past values of the remaining $N-1$ variables. More precisely, a VAR model of order or lag p, namely VAR(p), explains the current observation of N variables as the affine transformation of the previous p observations plus some innovation noises, i.e.,

$$\mathbf{y}_t = \boldsymbol{\phi}_0 + \sum_{i=1}^{p} \boldsymbol{\Phi}_i \mathbf{y}_{t-i} + \boldsymbol{\varepsilon}_t, \tag{1}$$

where $\mathbf{y}_t \in \mathbb{R}^N$ is the t-th observation of N time series variables, $\boldsymbol{\phi}_0 \in \mathbb{R}^N$ is a constant vector, $\boldsymbol{\Phi}_i \in \mathbb{R}^{N\times N}, i = 1, \ldots, p$ are the autoregressive coefficient matrices, and $\boldsymbol{\varepsilon}_t \in \mathbb{R}^N$ is the innovation noise [2]. This simple setup provides a systematic way to capture and analyze the rich dynamics in multivariate time series. The vector autoregressive (VAR) models constitute an important tool for multivariate time series analysis, and are widely used for data description, forecasting, structural inference, and policy analysis [1], [3]–[5].

Classical VAR modeling is rigidly inclined towards the multivariate Gaussian distribution, possibly owing to the simplicity in the mathematical analysis. The maximum likelihood estimation (MLE) method for estimating the parameters of a Gaussian VAR has been well studied [2]. However, it has been recognized and widely accepted that the empirical observations in various applications do not fit the Gaussian assumption, e.g., financial time series data [6], internet data [7], etc. The traditional method based on Gaussian distribution assumption on innovations is not appropriate for these applications. Furthermore, the Gaussian based methods are also not reliable to work under the presence of outliers [8]. The outliers are now ubiquitous in the majority of applications [9]. For example, sensors might return unreliable data because of the impulsive noise [10], the stock's return might be incorrectly recorded as 200% after a reverse three-for-one stock split as its price is tripled accordingly in market [11].

In this regard, using some heavy-tailed distributions, e.g., Student's t distribution, are shown to offer a more viable alternative for modeling real-world phenomena and tackling the spurious effect of outliers, particularly because their tails are non-thin and hence more realistic. The application of Student's t distributions has shown promising results in a wide variety of areas such as cluster analysis, discriminant analysis, multiple regression, robust projection indices, and missing data imputation [12]–[16]. Also for the VAR analysis modeling, the innovation with Student's t distribution has shown promising results [16].

Furthermore, missing values typically happen during the data observation or recording process, wherein values may not be measured, values may be measured but get lost, or values may be measured but are considered unusable. For example, the sensors

Manuscript received August 4, 2020; revised October 5, 2020; accepted October 19, 2020. Date of publication October 23, 2020; date of current version November 6, 2020. This work was supported by the Hong Kong GRF 16207019 research grant. The work of Junyan Liu was supported by the Hong Kong PhD Fellowship Scheme (HKPFS). *(Rui Zhou and Junyan Liu contributed equally to this work.) (Corresponding author: Rui Zhou.)*

Rui Zhou, Junyan Liu, and Daniel P. Palomar are with the Hong Kong University of Science and Technology (HKUST), Clear Water Bay, Kowloon, Hong Kong (e-mail: rui.zhou@connect.ust.hk; jliubl@connect.ust.hk; palomar@ust.hk).

Sandeep Kumar is with the Department of Electrical Engineering, Indian Institute of Technology Delhi, Delhi 110016, India (e-mail: ksandeep@iitd.ac.in).

Digital Object Identifier 10.1109/TSP.2020.3033378

might fail to upload the data from time to time because of a local power cut or communication interruption [17]. The issue of missing data also arises in the joint analysis of multiple univariate time series sampled at different frequencies. In financial data analysis, we usually have to jointly consider several variables that may have different sampling frequencies, e.g.,the stocks' return can be recorded daily while some aggregate macro indicators are available only at monthly, or even annual frequencies [18].

In theory, data are typically assumed complete, and algorithms are designed for complete data which may be not suitable for data with missing values [19]. There are some approaches to deal with missing values, but they are either statistically blind (e.g., discarding missing values, spline, interpolation that could destroy the statistics of the data), or are based on the Gaussian assumption [20], and not appropriate for heavy-tailed data. Thus, apart from heavy-tailed phenomena and the presence of outliers, another major obstacle in the analysis of VAR modeling is the issue of missing data. Existing methods, however, have not been able to tackle these challenges jointly. Very recently, the authors in [13], [21] have tackled these issues, but they are limited to the univariate model. The generalization of the univariate case to the multivariate case is non-trivial, due to the presence of cross relations between multiple variables.

To this end, the major goals of this paper are to develop MLE-based methods for parameter estimation of the Student's t VAR model with missing values. The likelihood of the Student's t VAR model with missing values does not lead itself to a closed-form expression, thus we utilize the expectation-maximization (EM) method [22]. To tackle the unavailability of a closed-form expression for the expectation, we resort to the stochastic approximation EM (SAEM) method, which performs the expectation step based on the stochastic approximation using samples of latent data generated from the distribution conditional on the observed data and current parameter estimates. Designing sampling schemes is critical to the effective implementation of the proposed framework, we propose two fast and computationally cheap Gibbs sampling schemes, both based on the Markov chain Monte Carlo (MCMC) procedure. The algorithms developed are effective in capturing the heavy-tailed phenomenon and robust against outliers and missing data. In addition, owing to their low computational complexity, the algorithms are amenable for high-dimensional and big data applications. Extensive experiments with both synthetic data and real financial data corroborate our claims.

The paper is organized as follows. We first give the preliminary knowledge on the multivariate Student's t distribution in Section II and then pose the problem formulation in Section III. In Section IV, we derived our algorithm based on the SAEM algorithm to solve the proposed MLE problem. In Section V, we propose two Gibbs sampling schemes for generating realizations of latent data. The complexity analysis of the two Gibbs sampling schemes is discussed in Section VI. The numerical experiments on synthetic data and real financial data are given in Section VII. Finally, the conclusion of this paper is given in Section VIII.

II. Preliminary Knowledge On The Multivariate Student's t Distribution

The multivariate Student's t distribution is a widely used heavy-tailed distribution. The N-dimensional multivariate Student's t distribution, denoted as $\mathrm{MVT}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu)$, has the probability density function (pdf)

$$f_{\mathrm{MVT}}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) = \frac{\Gamma\left(\frac{\nu+N}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right) \nu^{\frac{N}{2}} \pi^{\frac{N}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \left[1 + \frac{1}{\nu}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]^{-\frac{\nu+N}{2}}, \tag{2}$$

where ν is the degrees of freedom, $\boldsymbol{\Sigma}$ is the $N \times N$ positive definite scatter matrix, $\boldsymbol{\mu}$ is the N-dimensional mean vector and $\Gamma(a) = \int_0^\infty t^{(a-1)} \exp(-t) dt$ is the gamma function [12]. The smaller ν is, the heavier the tail is. Note that the multivariate Gaussian distribution is a special case of multivariate Student's t-distribution with $\nu \to +\infty$. Interestingly, the above multivariate Student's t distribution can be represented in a hierarchical structure as

$$\begin{aligned} \mathbf{x}|\tau &\overset{i.i.d}{\sim} \mathcal{N}\left(\boldsymbol{\mu}, \frac{1}{\tau}\boldsymbol{\Sigma}\right), \\ \tau &\overset{i.i.d}{\sim} \mathrm{Gamma}\left(\frac{\nu}{2}, \frac{\nu}{2}\right), \end{aligned} \tag{3}$$

where $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Gamma(a, b) denotes gamma distribution of shape a and rate b with pdf

$$f_{\mathrm{GM}}(\tau) = b^a \tau^{(a-1)} \frac{\exp(-b\tau)}{\Gamma(a)}. \tag{4}$$

The hierarchical structure will play an important role in our analysis and algorithm design, discussed in detail in Section IV.

III. Problem Formulation

Suppose we have observations from an N-dimensional time series $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T$ following a VAR model of order p as in (1). Collecting all the coefficients in $\boldsymbol{\Psi} \in \mathbb{R}^{N \times (Np+1)}$ as $\boldsymbol{\Psi} = [\boldsymbol{\phi}_0 \ \boldsymbol{\Phi}_1 \cdots \boldsymbol{\Phi}_p]$ and denoting by $\mathbf{x}_t = [1, \mathbf{y}_t^T, \cdots, \mathbf{y}_{t-p+1}^T]^T$, the VAR model of order p can be compactly written as:

$$\mathbf{y}_t = \boldsymbol{\Psi}\mathbf{x}_{t-1} + \boldsymbol{\varepsilon}_t. \tag{5}$$

where $\boldsymbol{\varepsilon}_t \in \mathbb{R}^N$ is the innovation noise following an i.i.d. process. To model the real world phenomena more closely, we assume $\boldsymbol{\varepsilon}_t$ in (5) to follow a zero-mean N-dimensional multivariate Student's t distribution, i.e., $\mathrm{MVT}(\mathbf{0}, \boldsymbol{\Sigma}, \nu)$.

We are interested in estimating the unknown parameters using the MLE method. Given all the parameters $\boldsymbol{\Psi}$, $\boldsymbol{\Sigma}$, and ν, the distribution of $\mathbf{y}_t$ $(t > p)$ conditional on all the preceding data $\mathcal{F}_{t-1}$, which consists of $\mathbf{y}_1$, $\mathbf{y}_2$, …, $\mathbf{y}_{t-1}$, only depends on the previous p samples $\mathbf{y}_{t-p}, \ldots, \mathbf{y}_{t-1}$, i.e.,

$$\begin{aligned} &p(\mathbf{y}_t|\boldsymbol{\Psi}, \boldsymbol{\Sigma}, \nu, \mathcal{F}_{t-1}) \\ &= p(\mathbf{y}_t|\boldsymbol{\Psi}, \boldsymbol{\Sigma}, \nu, \mathbf{y}_{t-p}, \ldots, \mathbf{y}_{t-1}) \\ &= f_{\mathrm{MVT}}(\mathbf{y}_t; \boldsymbol{\Psi}\mathbf{x}_{t-1}, \boldsymbol{\Sigma}, \nu) \end{aligned}$$

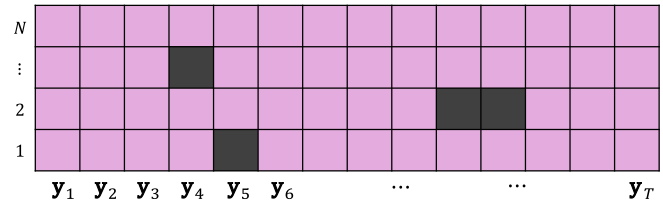


Fig. 1. The data matrix $\mathbf{Y} = (\mathbf{Y}_\mathsf{o}, \mathbf{Y}_\mathsf{m})$, where $\mathbf{Y}_\mathsf{o}$ consists of the observed data marked with purple blocks, while $\mathbf{Y}_\mathsf{m}$ consists of the missing data marked with black blocks.

$$= \frac{\Gamma\left(\frac{\nu+N}{2}\right)}{\sqrt{(\nu\pi)^N \det(\boldsymbol{\Sigma})}\Gamma\left(\frac{\nu}{2}\right)} \times \left(1 + \frac{1}{\nu}(\mathbf{y}_t - \boldsymbol{\Psi}\mathbf{x}_{t-1})^T \boldsymbol{\Sigma}^{-1}(\mathbf{y}_t - \boldsymbol{\Psi}\mathbf{x}_{t-1})\right)^{-\frac{\nu+N}{2}}, \tag{6}$$

Denote by $\boldsymbol{\theta} = (\boldsymbol{\Psi}, \boldsymbol{\Sigma}, \nu) \in \boldsymbol{\Theta}$ the unknown parameter set with $\boldsymbol{\Theta} := \{\boldsymbol{\theta} | \boldsymbol{\Sigma} \succ 0, \nu > 0\}$, where $\boldsymbol{\Sigma} \succ 0$ means $\boldsymbol{\Sigma}$ must be a positive definite matrix, and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_T] \in \mathbb{R}^{N\times T}$ the complete time series data matrix. Note that we are not enforcing stationarity of the VAR process. But it can be considered by introducing extra constraints on $\boldsymbol{\Psi}$ [23]. Ignoring the marginal distribution of $\mathbf{y}_1, \ldots, \mathbf{y}_p$, the (conditional) log-likelihood of the observed time series is

$$l(\boldsymbol{\theta}; \mathbf{Y}) = \log\left(\prod_{t=p+1}^{T} p(\mathbf{y}_t | \boldsymbol{\Psi}, \boldsymbol{\Sigma}, \nu, \mathbf{y}_{t-p}, \ldots, \mathbf{y}_{t-1})\right). \tag{7}$$

In many real world applications, however, the observations are not fully recorded or available resulting in partially observable data. For an illustration see Figure 1.

Denote by $\mathbf{Y}_\mathsf{o}$ the observed data and $\mathbf{Y}_\mathsf{m}$ the missing data. We can then write the log-likelihood of the observed data as

$$\begin{aligned} & l(\boldsymbol{\theta}; \mathbf{Y}_\mathsf{o}) \\ &= \log\left(\int \prod_{t=p+1}^{T} p(\mathbf{y}_t | \boldsymbol{\Psi}, \boldsymbol{\Sigma}, \nu, \mathbf{y}_{t-p}, \ldots, \mathbf{y}_{t-1}) \, \mathsf{d}\mathbf{Y}_\mathsf{m}\right) \\ &= \log\left(\int \prod_{t=p+1}^{T} f_{\mathrm{MVT}}(\mathbf{y}_t; \boldsymbol{\Psi}\mathbf{x}_{t-1}, \boldsymbol{\Sigma}, \nu) \, \mathsf{d}\mathbf{Y}_\mathsf{m}\right). \end{aligned} \tag{8}$$

The MLE problem for the Student's t VAR model with missing data is formulated as

$$\underset{\boldsymbol{\theta}}{\text{maximize}} \quad l(\boldsymbol{\theta}; \mathbf{Y}_\mathsf{o}). \tag{9}$$

The problem (9) is extremely challenging, with the difficulty stemming from the integral in (8) that does not have a closed-form expression. The MLE problem of the multivariate Student's t parameters with missing data has been tackled traditionally via the expectation-maximization (EM) approach, but these are limited to the i.i.d data model [12]. Developing EM-based approaches for modeling the Student's t VAR model under missing data is extremely challenging and still missing in the literature. The aim of this paper is to bridge the gap in the literature and develop an EM-based algorithm to estimate the parameters of the Student's t VAR process under missing data. But before that, some background on the EM algorithm and its variants are given in Appendix A.

IV. Parameter Estimation of Student's t VAR Model

Here we derive our algorithm to estimate the parameters $\boldsymbol{\theta} = (\boldsymbol{\Psi}, \boldsymbol{\Sigma}, \nu)$ of the Student's t VAR model based on the EM type algorithm. When applying the EM type algorithm, the selection of latent variables plays an important role. For the MLE problem (9), if we treat only the $\mathbf{Y}_\mathsf{m}$ as latent variables, it is still difficult to obtain the distribution $p(\mathbf{Z}|\mathbf{Y}, \boldsymbol{\theta}^{(k)})$, and then the expression of the subsequent $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})$ will become too difficult to obtain. Therefore, we incorporate the hierarchical structure of multivariate Student's t-distribution to avoid such dilemma. Since $\boldsymbol{\varepsilon}_t \sim \mathrm{MVT}(\mathbf{0}, \boldsymbol{\Sigma}, \nu)$, we can represent $\boldsymbol{\varepsilon}_t$ as

$$\begin{aligned} \boldsymbol{\varepsilon}_t | \tau_t &\overset{i.i.d}{\sim} \mathcal{N}\left(\mathbf{0}, \frac{1}{\tau_t}\boldsymbol{\Sigma}\right), \\ \tau_t &\overset{i.i.d}{\sim} \mathrm{Gamma}\left(\frac{\nu}{2}, \frac{\nu}{2}\right). \end{aligned} \tag{10}$$

Then we can develop an EM type algorithm to solve the MLE problem (9) by regarding both the missing data $\mathbf{Y}_\mathsf{m}$ and the mixture weights $\boldsymbol{\tau} = [\tau_{p+1}, \ldots, \tau_T]^T$ as latent variables.

A. E Step

The complete data log-likelihood can be expressed as in equation (11) shown at bottom of the next page, where the items independent of $\boldsymbol{\theta}$ are considered as a constant,

$$\begin{aligned} \mathbf{M}_0 &= \left[\mathbf{s}_0^{\tau\mathbf{y}} \; \mathbf{S}_{0,1}^{\tau\mathbf{y}\mathbf{y}} \; \cdots \; \mathbf{S}_{0,p}^{\tau\mathbf{y}\mathbf{y}}\right], \\ \mathbf{M}_1 &= \begin{bmatrix} s^\tau & (\mathbf{s}_1^{\tau\mathbf{y}})^T & \cdots & (\mathbf{s}_p^{\tau\mathbf{y}})^T \\ \mathbf{s}_1^{\tau\mathbf{y}} & \mathbf{S}_{1,1}^{\tau\mathbf{y}\mathbf{y}} & \cdots & \mathbf{S}_{1,p}^{\tau\mathbf{y}\mathbf{y}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{s}_p^{\tau\mathbf{y}} & \mathbf{S}_{p,1}^{\tau\mathbf{y}\mathbf{y}} & \cdots & \mathbf{S}_{p,p}^{\tau\mathbf{y}\mathbf{y}} \end{bmatrix}, \end{aligned} \tag{12}$$

$h(s(\mathbf{Y}_\mathsf{o}, \mathbf{Y}_\mathsf{m}, \boldsymbol{\tau}), \boldsymbol{\theta})$ contains the items depending on $\boldsymbol{\theta}$ and is linear in $s(\mathbf{Y}_\mathsf{o}, \mathbf{Y}_\mathsf{m}, \boldsymbol{\tau})$ for a given $\boldsymbol{\theta}$, and $s(\mathbf{Y}_\mathsf{o}, \mathbf{Y}_\mathsf{m}, \boldsymbol{\tau}) = (s^{\log\tau}, s^\tau, \{\mathbf{s}_i^{\tau\mathbf{y}}\}, \{\mathbf{S}_{i,j}^{\tau\mathbf{y}\mathbf{y}}\})$ is the collection of minimal sufficient statistics with

$$\begin{aligned} s^{\log\tau} &= \sum_{t=p+1}^{T} \log\tau_t, \\ s^\tau &= \sum_{t=p+1}^{T} \tau_t, \\ \mathbf{s}_i^{\tau\mathbf{y}} &= \sum_{t=p+1}^{T} \tau_t \mathbf{y}_{t-i}, \; i = 0, \ldots, p, \\ \mathbf{S}_{i,j}^{\tau\mathbf{y}\mathbf{y}} &= \sum_{t=p+1}^{T} \tau_t \mathbf{y}_{t-i}\mathbf{y}_{t-j}^T, \; i, j = 0, \ldots, p. \end{aligned} \tag{13}$$

Then, the expectation of the complete data log-likelihood $l(\boldsymbol{\theta}; \mathbf{Y}_\mathsf{o}, \mathbf{Y}_\mathsf{m}, \boldsymbol{\tau})$ over the latent data X_m and $\boldsymbol{\tau}$ can be expressed

as

$$\begin{aligned}
&Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)}\right)\\
&=\mathsf{E}_{p\left(\mathbf{Y}_{\mathsf{m}},\boldsymbol{\tau}|\mathbf{Y}_{\mathsf{o}},\boldsymbol{\theta}^{(k)}\right)}\left[l\left(\boldsymbol{\theta};\mathbf{Y}_{\mathsf{o}},\mathbf{Y}_{\mathsf{m}},\boldsymbol{\tau}\right)\right]\\
&=\iint l\left(\boldsymbol{\theta};\mathbf{Y}_{\mathsf{o}},\mathbf{Y}_{\mathsf{m}},\boldsymbol{\tau}\right)p\left(\mathbf{Y}_{\mathsf{m}},\boldsymbol{\tau}\mid\mathbf{Y}_{\mathsf{o}},\boldsymbol{\theta}^{(k)}\right)\mathsf{d}\mathbf{Y}_{\mathsf{m}}\mathsf{d}\boldsymbol{\tau}\\
&=\iint h\left(s\left(\mathbf{Y}_{\mathsf{o}},\mathbf{Y}_{\mathsf{m}},\boldsymbol{\tau}\right),\boldsymbol{\theta}\right)p\left(\mathbf{Y}_{\mathsf{m}},\boldsymbol{\tau}\mid\mathbf{Y}_{\mathsf{o}},\boldsymbol{\theta}^{(k)}\right)\mathsf{d}\mathbf{Y}_{\mathsf{m}}\mathsf{d}\boldsymbol{\tau}\\
&\quad+const.\\
&=h\left(\iint s\left(\mathbf{Y}_{\mathsf{o}},\mathbf{Y}_{\mathsf{m}},\boldsymbol{\tau}\right)p\left(\mathbf{Y}_{\mathsf{m}},\boldsymbol{\tau}\mid\mathbf{Y}_{\mathsf{o}},\boldsymbol{\theta}^{(k)}\right)\mathsf{d}\mathbf{Y}_{\mathsf{m}}\mathsf{d}\boldsymbol{\tau},\boldsymbol{\theta}\right)\\
&\quad+const.\\
&=h\left(\bar{s}^{(k+1)},\boldsymbol{\theta}\right)+const. \qquad (14)
\end{aligned}$$

where

$$\bar{s}^{(k+1)}=\iint s\left(\mathbf{Y}_{\mathsf{o}},\mathbf{Y}_{\mathsf{m}},\boldsymbol{\tau}\right)p\left(\mathbf{Y}_{\mathsf{m}},\boldsymbol{\tau}\mid\mathbf{Y}_{\mathsf{o}},\boldsymbol{\theta}^{(k)}\right)\mathsf{d}\mathbf{Y}_{\mathsf{m}}\mathsf{d}\boldsymbol{\tau}. \qquad (15)$$

Therefore, the computation of $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})$ actually reduces to the calculation of $\bar{s}^{(k+1)}$. To obtain the closed-form expression for the expectation $\bar{s}^{(k+1)}$, we need to first obtain the distribution $p(\mathbf{Y}_{\mathsf{m}},\boldsymbol{\tau}\mid\mathbf{Y}_{\mathsf{o}},\boldsymbol{\theta}^{(k)})$ and then compute the double integral. The $p(\mathbf{Y}_{\mathsf{m}},\boldsymbol{\tau}\mid\mathbf{Y}_{\mathsf{o}},\boldsymbol{\theta}^{(k)})$ can be expressed as

$$\begin{aligned}
&p\left(\mathbf{Y}_{\mathsf{m}},\boldsymbol{\tau}\mid\mathbf{Y}_{\mathsf{o}},\boldsymbol{\theta}^{(k)}\right)\\
&=\frac{p\left(\mathbf{Y}_{\mathsf{m}},\mathbf{Y}_{\mathsf{o}},\boldsymbol{\tau}\mid\boldsymbol{\theta}^{(k)}\right)}{p\left(\mathbf{Y}_{\mathsf{o}}\mid\boldsymbol{\theta}^{(k)}\right)}\\
&\propto p\left(\mathbf{Y}_{\mathsf{m}},\mathbf{Y}_{\mathsf{o}},\boldsymbol{\tau}\mid\boldsymbol{\theta}^{(k)}\right)\\
&=\prod_{t=p+1}^{T}\frac{\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}\left(\tau_t\right)^{\frac{\nu}{2}-1}\exp\left(-\frac{\nu}{2}\tau_t\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\det\left(2\pi\boldsymbol{\Sigma}/\tau_t\right)}}\\
&\quad\times\exp\left(-\frac{\tau_t}{2}\left(\mathbf{y}_t-\boldsymbol{\Psi}\mathbf{x}_{t-1}\right)^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{y}_t-\boldsymbol{\Psi}\mathbf{x}_{t-1}\right)\right). \qquad (16)
\end{aligned}$$

There is no closed-form expression for $p(\mathbf{Y}_{\mathsf{m}},\boldsymbol{\tau}\mid\mathbf{Y}_{\mathsf{o}},\boldsymbol{\theta}^{(k)})$, and thus, we cannot obtain an expression $\bar{s}^{(k+1)}$ in closed-form. It makes the subsequent double integral rather complicated and difficult. Therefore, instead of pursuing the exact expression of $\bar{s}^{(k+1)}$, we turn to the SAEM-MCMC algorithm, which generates L samples of latent variables $\{(\mathbf{Y}_{\mathsf{m}}^{(k+1,l)},\boldsymbol{\tau}^{(k+1,l)})\}_{l=1,\ldots,L}$ from the Markov chain, and approximate the $\bar{s}^{(k+1)}$ by a stochastic approximation. More specifically, the expected minimal sufficient statistics $\bar{s}^{(k+1)}$ can be approximated by

$$\begin{aligned}
&\hat{s}^{(k+1)}=\\
&\hat{s}^{(k)}+\gamma^{(k)}\left(\frac{1}{L}\sum_{l=1}^{L}s\left(\mathbf{Y}_{\mathsf{o}},\mathbf{Y}_{\mathsf{m}}^{(k+1,l)},\boldsymbol{\tau}^{(k+1,l)}\right)-\hat{s}^{(k)}\right). \qquad (17)
\end{aligned}$$

Then we can have the approximation of the expected log-likelihood

$$\hat{Q}\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)}\right)=h\left(\hat{s}^{(k+1)},\boldsymbol{\theta}\right)+\text{const.} \qquad (18)$$

B. M Step

After obtaining the approximation of the expected complete data log-likelihood function $\hat{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})$, we can update the parameter estimation via maximizing $\hat{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})$ over $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}^{(k+1)}=\arg\max_{\boldsymbol{\theta}}\hat{Q}\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)}\right), \qquad (19)$$

with

$$\begin{aligned}
&\hat{Q}\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)}\right)\\
&=(T-p)\left\{\frac{\nu}{2}\log\frac{\nu}{2}-\log\Gamma(\frac{\nu}{2})\right\}-\frac{T-p}{2}\log\left(\det\left(\boldsymbol{\Sigma}\right)\right)\\
&+\frac{\nu}{2}\left(\left(s^{\log\tau}\right)^{(k+1)}-\left(s^{\tau}\right)^{(k+1)}\right)\\
&-\frac{1}{2}\mathsf{Tr}\Big\{\boldsymbol{\Sigma}^{-1}\left(\left(\mathbf{S}_{0,0}^{\tau\mathbf{xx}}\right)^{(k+1)}\right.\\
&\left.-2\mathbf{M}_0^{(k+1)}\boldsymbol{\Psi}^T+\boldsymbol{\Psi}\mathbf{M}_1^{(k+1)}\boldsymbol{\Psi}^T\right)\Big\}+\text{const.} \qquad (20)
\end{aligned}$$

$$\begin{aligned}
l\left(\boldsymbol{\theta};\mathbf{Y}_{\mathsf{o}},\mathbf{Y}_{\mathsf{m}},\boldsymbol{\tau}\right)&=\log p\left(\mathbf{Y},\boldsymbol{\tau}|\boldsymbol{\theta}\right)=\log\prod_{t=p+1}^{T}\left(f_{\text{GM}}\left(\tau_t;\frac{\nu}{2},\frac{\nu}{2}\right)f_{\mathcal{N}}\left(\mathbf{y}_t;\boldsymbol{\Psi}\mathbf{x}_{t-1},\frac{\boldsymbol{\Sigma}}{\tau_t}\right)\right)\\
&=\sum_{t=p+1}^{T}\log\left\{\frac{\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}\left(\tau_t\right)^{\frac{\nu}{2}-1}\exp\left(-\frac{\nu}{2}\tau_t\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\det\left(2\pi\boldsymbol{\Sigma}/\tau_t\right)}}\exp\left(-\frac{\tau_t}{2}\left(\mathbf{y}_t-\boldsymbol{\Psi}\mathbf{x}_{t-1}\right)^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{y}_t-\boldsymbol{\Psi}\mathbf{x}_{t-1}\right)\right)\right\}\\
&=\sum_{t=p+1}^{T}\log\left\{\frac{\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}\left(\tau_t\right)^{\frac{\nu}{2}-1}\exp\left(-\frac{\nu}{2}\tau_t\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\det\left(2\pi\boldsymbol{\Sigma}/\tau_t\right)}}\exp\left(-\frac{\tau_t}{2}\mathsf{Tr}\left(\boldsymbol{\Sigma}^{-1}\left(\mathbf{y}_t\mathbf{y}_t^T-2\mathbf{y}_t\mathbf{x}_{t-1}^T\boldsymbol{\Psi}^T+\boldsymbol{\Psi}\mathbf{x}_{t-1}\mathbf{x}_{t-1}^T\boldsymbol{\Psi}^T\right)\right)\right)\right\}\\
&=\sum_{t=p+1}^{T}\left(\frac{N}{2}\log\frac{\tau_t}{2\pi}-\log\tau_t\right)+(T-p)\left(\frac{\nu}{2}\log\left(\frac{\nu}{2}\right)-\log\left(\Gamma\left(\frac{\nu}{2}\right)\right)\right)+\frac{\nu}{2}\left(s^{\log\tau}-s^{\tau}\right)\\
&\quad-\frac{T-p}{2}\log\left(\det\left(\boldsymbol{\Sigma}\right)\right)-\frac{1}{2}\mathsf{Tr}\left(\boldsymbol{\Sigma}^{-1}\left(\mathbf{S}_{0,0}^{\tau\mathbf{yy}}-2\mathbf{M}_0\boldsymbol{\Psi}^T+\boldsymbol{\Psi}\mathbf{M}_1\boldsymbol{\Psi}^T\right)\right)\\
&=h\left(s\left(\mathbf{Y}_{\mathsf{o}},\mathbf{Y}_{\mathsf{m}},\boldsymbol{\tau}\right),\boldsymbol{\theta}\right)+\text{const.} \qquad (11)
\end{aligned}$$

The optimization of parameter ν is decoupled with the optimization of $(\boldsymbol{\Psi}, \boldsymbol{\Sigma})$. Thus, ν can be updated as

$$\nu^{(k+1)} = \arg\max_{\nu} \left\{ \frac{\nu}{2}\log\left(\frac{\nu}{2}\right) - \log\left(\Gamma\left(\frac{\nu}{2}\right)\right) + \frac{\nu}{2}\left(\left(s^{\log\tau}\right)^{(k+1)} - \left(s^{\tau}\right)^{(k+1)}\right)\right\}, \tag{21}$$

which admits a unique solution [12, Proposition 1], and can be easily solved by the one-dimension search, e.g., the bisection method. The update for $\boldsymbol{\Psi}$ and $\boldsymbol{\Sigma}$ can be easily found by setting the derivatives of $\hat{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})$ with respect to $\boldsymbol{\Psi}$ and $\boldsymbol{\Sigma}$ to zero:

$$\begin{aligned}
\frac{\partial \hat{Q}\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)}\right)}{\partial \boldsymbol{\Psi}} &= -\boldsymbol{\Sigma}^{-1}\mathbf{M}_0^{(k+1)} + \boldsymbol{\Sigma}^{-1}\boldsymbol{\Psi}\mathbf{M}_1^{(k+1)} = \mathbf{0},\\
\frac{\partial \hat{Q}\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)}\right)}{\partial \boldsymbol{\Sigma}^{-1}} &= \frac{T-p}{2}\boldsymbol{\Sigma} - \frac{1}{2}\left\{\left(\mathbf{S}_{0,0}^{\tau\mathbf{xx}}\right)^{(k+1)} - 2\mathbf{M}_0^{(k+1)}\boldsymbol{\Psi}^T + \boldsymbol{\Psi}\mathbf{M}_1^{(k+1)}\boldsymbol{\Psi}^T\right\} = \mathbf{0},
\end{aligned} \tag{22}$$

which gives us

$$\boldsymbol{\Psi}^{(k+1)} = \mathbf{M}_0^{(k+1)}\left(\mathbf{M}_1^{(k+1)}\right)^{-1}, \tag{23}$$

$$\boldsymbol{\Sigma}^{(k+1)} = \frac{1}{T-p}\left\{\left(\mathbf{S}_{0,0}^{\tau\mathbf{xx}}\right)^{(k+1)} - \mathbf{M}_0^{(k+1)}\left(\boldsymbol{\Psi}^{(k+1)}\right)^T - \boldsymbol{\Psi}^{(k+1)}\left(\mathbf{M}_0^{(k+1)}\right)^T + \boldsymbol{\Psi}^{(k+1)}\mathbf{M}_1^{(k+1)}\left(\boldsymbol{\Psi}^{(k+1)}\right)^T\right\}. \tag{24}$$

The complete SAEM-MCMC algorithm is to perform the E step and M step iteratively until the convergence, that is:

1) **Stochastic E step**: generate L realizations $(\mathbf{Y}_{\mathsf{m}}^{(k+1,l)}, \boldsymbol{\tau}^{(k+1,l)})$ with $l = 1, \ldots, L$, and then evaluate $\hat{s}^{(k)}$ as in (17).
2) **M step**: obtain $\boldsymbol{\theta}^{(k+1)}$ as in (21), (23), and (24).

C. Maximization Step With Partly Known Information

The parameters might be partly known under some circumstances. For example, ν can be set as $+\infty$ if one assumes that the innovations are Gaussian distributed, and $\boldsymbol{\Phi}_1$ may be set as $\mathbf{I}$ in the random walk [24]. If ν or $\boldsymbol{\Sigma}$ is known a priori, we shall skip their update in the maximization step. However, when the parameter $\boldsymbol{\Psi} = [\,\boldsymbol{\phi}_0\ \boldsymbol{\Phi}_1\ \cdots\ \boldsymbol{\Phi}_p\,]$ is partly known, i.e., one or more items in collection $\{\boldsymbol{\phi}_0, \boldsymbol{\Phi}_1, \ldots, \boldsymbol{\Phi}_p\}$ are given as prior knowledge, the corresponding parameters should be fixed during the iterations [25]. Then the update scheme in (23) will not be applicable any more. We can easily handle this case by simple matrix manipulation: given the prior knowledge on $\boldsymbol{\Psi}$, we can always find a permutation matrix $\mathbf{P} \in \mathbb{R}^{(Np+1)\times(Np+1)}$ to make the columns interchanged on $\boldsymbol{\Psi}$, satisfying

$$\left[\,\tilde{\boldsymbol{\Psi}}_1\ \tilde{\boldsymbol{\Psi}}_2\,\right] = \boldsymbol{\Psi}\mathbf{P}, \tag{25}$$

where $\tilde{\boldsymbol{\Psi}}_1$ is the unknown part in $\boldsymbol{\Psi}$ and $\tilde{\boldsymbol{\Psi}}_2$ is known and fixed all along. Then we can write the $\hat{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})$ with respect to $\tilde{\boldsymbol{\Psi}}_1$ as in equation (27) shown at bottom of this page, where

$$\begin{aligned}
\tilde{\mathbf{M}}_0^{(k+1)} &= \mathbf{M}_0^{(k+1)}\mathbf{P} = \left[\,\left(\tilde{\mathbf{M}}_0^{(k+1)}\right)_1\ \left(\tilde{\mathbf{M}}_0^{(k+1)}\right)_2\,\right],\\
\tilde{\mathbf{M}}_1^{(k+1)} &= \mathbf{P}^T\mathbf{M}_1^{(k+1)}\mathbf{P} = \begin{bmatrix} \left(\tilde{\mathbf{M}}_1^{(k+1)}\right)_{11} & \left(\tilde{\mathbf{M}}_1^{(k+1)}\right)_{12} \\ \left(\tilde{\mathbf{M}}_1^{(k+1)}\right)_{21} & \left(\tilde{\mathbf{M}}_1^{(k+1)}\right)_{22} \end{bmatrix}.
\end{aligned} \tag{26}$$

Then we can set the derivative of $\hat{Q}(\tilde{\boldsymbol{\Psi}}_1|\boldsymbol{\theta}^{(k)})$ with respect to $\tilde{\boldsymbol{\Psi}}_1$ to zero:

$$\frac{\partial \hat{Q}\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)}\right)}{\partial \tilde{\boldsymbol{\Psi}}_1} = -\boldsymbol{\Sigma}^{-1}\tilde{\mathbf{M}}_0^{(k+1)} + \boldsymbol{\Sigma}^{-1}\tilde{\boldsymbol{\Psi}}_1\left(\tilde{\mathbf{M}}_1^{(k+1)}\right)_{11} + \boldsymbol{\Sigma}^{-1}\tilde{\boldsymbol{\Psi}}_2\left(\tilde{\mathbf{M}}_1^{(k+1)}\right)_{21} = \mathbf{0}. \tag{28}$$

Then the update for $\tilde{\boldsymbol{\Psi}}_1$ is

$$\tilde{\boldsymbol{\Psi}}_1^{(k+1)} = \left[\tilde{\mathbf{M}}_0^{(k+1)} - \tilde{\boldsymbol{\Psi}}_2\left(\tilde{\mathbf{M}}_1^{(k+1)}\right)_{21}\right]\left(\tilde{\mathbf{M}}_1^{(k+1)}\right)_{11}^{-1}. \tag{29}$$

The $\boldsymbol{\Psi}$ can be recovered by

$$\boldsymbol{\Psi}^{(k+1)} = \left[\,\tilde{\boldsymbol{\Psi}}_1^{(k+1)}\ \tilde{\boldsymbol{\Psi}}_2\,\right]\mathbf{P}^T. \tag{30}$$

V. Generation of Realizations

In this section, we discuss the way to generate samples of latent variables $(\mathbf{Y}_{\mathsf{m}}, \boldsymbol{\tau})$. As the joint pdf of $(\mathbf{Y}_{\mathsf{m}}, \boldsymbol{\tau})$ shown in

$$\begin{aligned}
\hat{Q}\left(\tilde{\boldsymbol{\Psi}}_1|\boldsymbol{\theta}^{(k)}\right) &= \frac{1}{2}\mathsf{Tr}\left\{\boldsymbol{\Sigma}^{-1}\left(-2\mathbf{M}_0^{(k+1)}\boldsymbol{\Psi}^T + \boldsymbol{\Psi}\mathbf{M}_1^{(k+1)}\boldsymbol{\Psi}^T\right)\right\} + \text{const.}\\
&= \frac{1}{2}\mathsf{Tr}\left\{\boldsymbol{\Sigma}^{-1}\left(-2\mathbf{M}_0^{(k+1)}\mathbf{P}\mathbf{P}^T\boldsymbol{\Psi}^T + \boldsymbol{\Psi}\mathbf{P}\mathbf{P}^T\mathbf{M}_1^{(k+1)}\mathbf{P}\mathbf{P}^T\boldsymbol{\Psi}^T\right)\right\} + \text{const.}\\
&= \frac{1}{2}\mathsf{Tr}\left\{\boldsymbol{\Sigma}^{-1}\left(-2\mathbf{M}_0^{(k+1)}\mathbf{P}\begin{bmatrix}\tilde{\boldsymbol{\Psi}}_1^T\\ \tilde{\boldsymbol{\Psi}}_2^T\end{bmatrix} + \left[\,\tilde{\boldsymbol{\Psi}}_1\ \tilde{\boldsymbol{\Psi}}_2\,\right]\mathbf{P}^T\mathbf{M}_1^{(k+1)}\mathbf{P}\begin{bmatrix}\tilde{\boldsymbol{\Psi}}_1^T\\ \tilde{\boldsymbol{\Psi}}_2^T\end{bmatrix}\right)\right\} + \text{const.}\\
&= \frac{1}{2}\mathsf{Tr}\left\{\boldsymbol{\Sigma}^{-1}\left(-2\tilde{\mathbf{M}}_0^{(k+1)}\begin{bmatrix}\tilde{\boldsymbol{\Psi}}_1^T\\ \tilde{\boldsymbol{\Psi}}_2^T\end{bmatrix} + \left[\,\tilde{\boldsymbol{\Psi}}_1\ \tilde{\boldsymbol{\Psi}}_2\,\right]\tilde{\mathbf{M}}_1^{(k+1)}\begin{bmatrix}\tilde{\boldsymbol{\Psi}}_1^T\\ \tilde{\boldsymbol{\Psi}}_2^T\end{bmatrix}\right)\right\} + \text{const.}\\
&= \frac{1}{2}\mathsf{Tr}\left\{\boldsymbol{\Sigma}^{-1}\left(-2\left(\tilde{\mathbf{M}}_0^{(k+1)}\right)_1\tilde{\boldsymbol{\Psi}}_1^T + \tilde{\boldsymbol{\Psi}}_1\left(\tilde{\mathbf{M}}_1^{(k+1)}\right)_{11}\tilde{\boldsymbol{\Psi}}_1^T + 2\tilde{\boldsymbol{\Psi}}_2\left(\tilde{\mathbf{M}}_1^{(k+1)}\right)_{21}\tilde{\boldsymbol{\Psi}}_1^T\right)\right\} + \text{const.}
\end{aligned} \tag{27}$$

(16) is complicated, and we cannot sample from it directly, we can generate samples of $(\mathbf{Y}_{\mathsf{m}}, \boldsymbol{\tau})$ in a MCMC procedure by the Gibbs sampling method. The Gibbs sampling is a MCMC algorithm that draws samples from a joint distribution by generating a sequence of realizations from the conditional distributions alternately. The sequence of realizations is a Markov chain, and they approximate the original joint distribution when reaching the stationary distribution.

A. Gibbs Sampling Between $\boldsymbol{\tau}$ and Entire $\mathbf{Y}_m$

One way to preform the Gibbs sampling is dividing the latent data into two blocks, i.e., $\boldsymbol{\tau}$ and $\mathbf{Y}_{\mathsf{m}}$. Then, we can use the Gibbs sampling by drawing $\boldsymbol{\tau}$ and $\mathbf{Y}_{\mathsf{m}}$ conditional on each other alternately. More specifically, given the current estimation of parameters $\boldsymbol{\theta}^{(k)}$ and the current sample $(\boldsymbol{\tau}^{(k,l)}, \mathbf{Y}_{\mathsf{m}}^{(k,l)})$ in l-th Markov chain, we can generate the next sample $(\boldsymbol{\tau}^{(k+1,l)}, \mathbf{Y}_{\mathsf{m}}^{(k+1,l)})$ via the following two steps:

1) sample $\boldsymbol{\tau}^{(k+1,l)}$ from $p(\boldsymbol{\tau}|\mathbf{Y}_{\mathsf{m}}^{(k,l)}, \mathbf{Y}_{\mathsf{o}}, \boldsymbol{\theta}^{(k)})$
2) sample $\mathbf{Y}_{\mathsf{m}}^{(k+1,l)}$ from $p(\mathbf{Y}_{\mathsf{m}}|\boldsymbol{\tau}^{(k+1,l)}, \mathbf{Y}_{\mathsf{o}}, \boldsymbol{\theta}^{(k)})$

As given in Lemma 1, sampling $\boldsymbol{\tau}^{(k+1,l)}$ is simply drawing random samples from the gamma distribution. However, the distribution of $p(\mathbf{Y}_{\mathsf{m}}|\boldsymbol{\tau}^{(k+1,l)}, \mathbf{Y}_{\mathsf{o}}, \boldsymbol{\theta}^{(k)})$ is still difficult to be directly identified. Therefore, we can derive such distribution in a two-stage way:

1) first figure out the distribution of $\{\mathbf{y}_t\}_{t=p+1}^{T}$ conditional on $\boldsymbol{\tau}$, $\boldsymbol{\theta}$, and $\{\mathbf{y}_t\}_{t=1}^{p}$, which turns out to be a multivariate Gaussian distribution as in Lemma 2;
2) then the missing data $\mathbf{Y}_{\mathsf{m}}$ in $\{\mathbf{y}_t\}_{t=p+1}^{T}$ can be easily inferred as a multivariate Gaussian distribution from Lemma 3.

Lemma 1: Given $\mathbf{Y}$ and $\boldsymbol{\theta}$, the mixture weights are independent from each other, i.e.,

$$p\left(\boldsymbol{\tau}|\mathbf{Y}, \boldsymbol{\theta}\right) = \prod_{t=p+1}^{T} p\left(\tau_t|\mathbf{Y}, \boldsymbol{\theta}\right). \tag{31}$$

In addition, τ_t follows a gamma distribution:

$$\tau_t|\mathbf{Y}, \boldsymbol{\theta} \sim \text{Gamma}\left(\tfrac{\nu+N}{2}, \tfrac{\nu+(\mathbf{y}_t-\boldsymbol{\Psi}\mathbf{x}_{t-1})^T\boldsymbol{\Sigma}^{-1}(\mathbf{y}_t-\boldsymbol{\Psi}\mathbf{x}_{t-1})}{2}\right). \tag{32}$$

See Appendix B. ■

Lemma 2: Given $\boldsymbol{\tau}$, $\tilde{\mathbf{y}}_p$, and $\boldsymbol{\theta}$, we have

$$\tilde{\mathbf{y}}_{-p}|\boldsymbol{\tau}, \tilde{\mathbf{y}}_p, \boldsymbol{\theta} \sim \mathcal{N}\left(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}}\right), \tag{33}$$

where $\tilde{\mathbf{y}}_p = [\mathbf{y}_1^T, \ldots, \mathbf{y}_p^T]^T$, $\tilde{\mathbf{y}}_{-p} = [\mathbf{y}_{p+1}^T, \mathbf{y}_{p+2}^T, \ldots, \mathbf{y}_T^T]^T$, i-th fragment of length N in $\tilde{\boldsymbol{\mu}}$ is

$$(\tilde{\boldsymbol{\mu}})_{(i)} = \left(\sum_{j=0}^{i-1}\mathbf{B}^j\right)_{[N]}\boldsymbol{\phi}_0 + \left(\mathbf{B}^i\tilde{\mathbf{y}}_p\right)_{[N]}, \tag{34}$$

and (i,j)-th block matrix of dimension $N \times N$ in $\tilde{\boldsymbol{\Sigma}}$ is

$$\left(\tilde{\boldsymbol{\Sigma}}\right)_{(i,j)} = \sum_{q=1}^{\min(i,j)} \frac{1}{\tau_{q+p}}\left(\mathbf{B}^{i-q}\right)_{[N]}\boldsymbol{\Sigma}\left(\left(\mathbf{B}^{j-q}\right)_{[N]}\right)^T, \tag{35}$$

Algorithm 1: SAEM-MCMC for Student's t VAR Parameter Estimation by Gibbs Sampling Between $\boldsymbol{\tau}$ and Entire $\mathbf{Y}_{\mathsf{m}}$.

```
Initialize θ^(0) ∈ Θ and Y_m^(0,l) for l = 1, 2 ..., L.
for k = 0, 1, 2, ... do
  Simulation: generate L realizations by first sampling
    τ^(k+1,l) from p(τ|Y_m^(k,l), Y_o, θ^(k)), and then
    sampling Y_m^(k+1,l) from p(Y_m|τ^(k+1,l), Y_o, θ^(k)).
  Stochastic approximation: evaluate ŝ^(k) as in (17).
  Maximization: update θ^(k+1) as in (21), (23), and
    (24).
  if stopping criteria is met then
    terminate loop
  end if
end for
```

with $(\mathbf{x})_{[N]}$ being the first N elements of the vector $\mathbf{x}$, $(\mathbf{X})_{[N]}$ being the upper left $N \times N$ block of the matrix $\mathbf{X}$, and

$$\mathbf{B} = \begin{bmatrix} \boldsymbol{\Phi}_1 & \boldsymbol{\Phi}_2 & \cdots & \cdots & \boldsymbol{\Phi}_p \\ \mathbf{I}_N & \mathbf{0}_N & \cdots & \cdots & \mathbf{0}_N \\ \mathbf{0}_N & \mathbf{I}_N & \mathbf{0}_N & \cdots & \mathbf{0}_N \\ \vdots & & \ddots & & \vdots \\ \mathbf{0}_N & \cdots & \cdots & \mathbf{I}_N & \mathbf{0}_N \end{bmatrix}. \tag{36}$$

Proof: See Appendix C. ■

Lemma 3: [26] Assume $(\boldsymbol{a}, \boldsymbol{b}) \overset{i.i.d}{\sim} \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then we have

$$\begin{aligned} \boldsymbol{a} &\overset{i.i.d}{\sim} \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{a}}, \boldsymbol{\Sigma}_{\boldsymbol{a}}), \\ \boldsymbol{b}|\boldsymbol{a} &\overset{i.i.d}{\sim} \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{b}|\boldsymbol{a}}, \boldsymbol{\Sigma}_{\boldsymbol{b}|\boldsymbol{a}}\right), \end{aligned} \tag{37}$$

where $\boldsymbol{\mu}_{\boldsymbol{b}|\boldsymbol{a}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{b}|\boldsymbol{a}}$ can be written as

$$\begin{aligned} \boldsymbol{\mu}_{\boldsymbol{b}|\boldsymbol{a}} &= \boldsymbol{\mu}_{\boldsymbol{b}} + \boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{a}}\boldsymbol{\Sigma}_{\boldsymbol{a}}^{-1}\left(\boldsymbol{a} - \boldsymbol{\mu}_{\boldsymbol{a}}\right), \\ \boldsymbol{\Sigma}_{\boldsymbol{b}|\boldsymbol{a}} &= \boldsymbol{\Sigma}_{\boldsymbol{b}} - \boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{a}}\boldsymbol{\Sigma}_{\boldsymbol{a}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}}. \end{aligned} \tag{38}$$

Finally, we find that the Gibbs sampling between $\boldsymbol{\tau}$ and entire $\mathbf{Y}_{\mathsf{m}}$ is simply drawing the random samples from gamma distribution and Gaussian distribution separately. Then the complete SAEM-MCMC algorithm for estimating the parameters of Student's t VAR model is given in Algorithm 1.

Parallel Implementation: As given in Lemma 2, sampling the entire $\mathbf{Y}_{\mathsf{m}}$ involves the calculation of mean and covariance matrix of a $N \times (T-p)$-dimensional Gaussian distribution and the subsequent drawing the realizations from the conditional Gaussian distribution. The computational cost of such operation is expected to be very heavy in practical applications. For example, for the financial data in daily or even higher frequencies, T could easily reach more than 10^3, and then the computational cost of the previous sampling scheme becomes unacceptable. Instead, we can accelerate the sampling procedure by first partitioning the missing data into groups wrapped by at least p consecutive fully observed samples before and after (if available) themselves, and then sampling the $\mathbf{Y}_{\mathsf{m}}$ inside each missing group in parallel. For example, as shown in Figure 2, we

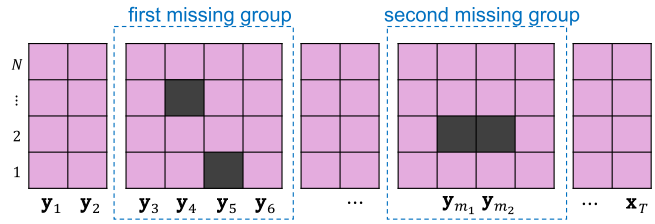


Fig. 2. An example of partitioning the missing data into groups for VAR(1) model parameter estimation.

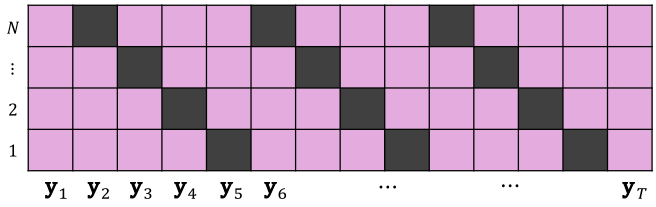


Fig. 3. An example of observed $\mathbf{Y}$ with consecutive missing values.

can partition out two missing groups from $\mathbf{Y}_{\mathsf{m}}$ in Figure 1 when we estimate the parameters of a VAR model with $p = 1$. Lemma 4 guarantees that the distribution of missing data are inter-group independent. More specifically, to sample the missing data in second missing group of Figure 2, we can apply the following two steps:

1) identify the distribution of $\{\mathbf{y}_t\}_{t=m_1}^{m_2+p}$ conditional on $\boldsymbol{\tau}$, $\boldsymbol{\theta}$, and $\{\mathbf{y}_t\}_{t=m_1-p}^{m_1-1}$ as a Gaussian distribution with the employment of Lemma 2 in each partitioned missing group,
2) sample missing data in $\{\mathbf{y}_t\}_{t=m_1}^{m_2}$ from conditional Gaussian distribution using Lemma 3.

The missing data in the first missing group of Figure 2 can be sampled by easily repeating the above two steps but setting $m_1 = 4$ and $m_2 = 5$. The total computational cost can be expected to reduce as the length of each group has been significantly smaller than the original dataset.

Lemma 4: Given $m_1 \leq m_2$, $\boldsymbol{\tau}$, $\mathbf{Y}_{\mathsf{o}}$, and $\boldsymbol{\theta}$, if $\{\mathbf{y}_t\}_{t=m_1-p}^{m_1-1}$ and $\{\mathbf{y}_t\}_{t=m_2+1}^{m_2+p}$ are fully observed, we have

$$p\left(\{\mathbf{y}_{t,\mathsf{m}}\}_{t=m_1}^{m_2} | \boldsymbol{\tau}, \boldsymbol{\theta}, \mathbf{Y}_{\mathsf{o}}\right) = p\left(\{\mathbf{y}_{t,\mathsf{m}}\}_{t=m_1}^{m_2} | \boldsymbol{\tau}, \boldsymbol{\theta}, \{\mathbf{y}_{t,\mathsf{o}}\}_{t=m_1-p}^{m_2+p}\right), \quad (39)$$

where $\mathbf{y}_{t,\mathsf{m}}$ and $\mathbf{y}_{t,\mathsf{o}}$ are the missing data and observed data in $\mathbf{y}_t$, respectively.

Proof: See Appendix D. ■

B. Gibbs Sampling Among $\{\boldsymbol{\tau}, \mathbf{y}_{p+1,m}, \ldots, \mathbf{y}_{T,m}\}$: An Atom Operation

We have given a Gibbs sampling scheme in the previous section, which works by employing Gibbs sampling between $\boldsymbol{\tau}$ and entire $\mathbf{Y}_{\mathsf{m}}$. The computational cost of sampling the entire $\mathbf{Y}_{\mathsf{m}}$ can be reduced by partitioning the data into several missing groups according to the rule and then sampling missing data in each missing group in parallel. However, in some extreme cases, we may meet the consecutive missing pattern, making it impossible to partition the $\mathbf{Y}_{\mathsf{m}}$ into more than one missing group or some partitioned missing groups are still significantly large. For example, as in Figure 3, the Gibbs sampling in missing groups can not help reduce any computational cost in this case. We can divide the latent data by a finer grained division as $\boldsymbol{\tau}$ and $\{\mathbf{y}_{t,\mathsf{m}}\}_{t=1}^{T}$, and then employ the Gibbs sampling among these blocks. We call such sampling scheme by the atom sampling operation. More specifically, given the current estimation of parameters $\boldsymbol{\theta}^{(k)}$ and the current sample $(\boldsymbol{\tau}^{(k,l)}, \mathbf{Y}_{\mathsf{m}}^{(k,l)})$ in l-th Markov chain, we can generate the next sampler via the following steps:

1) sample $\boldsymbol{\tau}^{(k+1,l)}$ from $p(\boldsymbol{\tau}|\mathbf{Y}_{\mathsf{m}}^{(k,l)}, \mathbf{Y}_{\mathsf{o}}, \boldsymbol{\theta}^{(k)})$
2) for $j = p+1, \ldots, T$, sample $\mathbf{y}_{j,\mathsf{m}}^{(k+1,l)}$ conditional on $\{\mathbf{y}_{t,\mathsf{m}}^{(k+1,l)}\}_{t=1}^{j-1}$, $\{\mathbf{y}_{t,\mathsf{m}}^{(k,l)}\}_{t=j+1}^{T}$, $\mathbf{Y}_{\mathsf{o}}$, $\boldsymbol{\tau}^{(k+1,l)}$, and $\boldsymbol{\theta}^{(k)}$.

The sampling procedure of $\mathbf{y}_{j,\mathsf{m}}^{(k+1,l)}$ can be performed by similar procedure as in previous section: first identify the distribution of $\{\mathbf{y}_t\}_{t=j}^{j+p}$ conditional on $\boldsymbol{\tau}$, $\boldsymbol{\theta}$, and $\{\mathbf{y}_t\}_{t=j-p}^{j-1}$, which can be easily found as a Gaussian distribution using Lemma 2 by regarding $\{\mathbf{y}_t\}_{t=j-p}^{j-1}$ as the first p full observations in a time series of length $2p+1$, then sample $\mathbf{y}_{t,\mathsf{m}}$ from a conditional Gaussian distribution. The complete algorithm for SAEM-MCMC method with atom sampling is given in Algorithm 2.

Algorithm 2: SAEM-MCMC for Student's t VAR Parameter Estimation by Gibbs Sampling Between $\boldsymbol{\tau}$ and $\mathbf{Y}_{\mathsf{m}}$ Atoms.

1: Initialize $\boldsymbol{\theta}^{(0)} \in \Theta$ and $\mathbf{Y}_{\mathsf{m}}^{(0,l)}$ for $l = 1, 2 \ldots, L$.
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: Simulation: generate L realizations by first sampling $\boldsymbol{\tau}^{(k+1,l)}$ from $p(\boldsymbol{\tau}|\mathbf{Y}_{\mathsf{m}}^{(k,l)}, \mathbf{Y}_{\mathsf{o}}, \boldsymbol{\theta}^{(k)})$, and then sampling $\mathbf{y}_{j,\mathsf{m}}^{(k+1,l)}$ conditional on $\{\mathbf{y}_{t,\mathsf{m}}^{(k+1,l)}\}_{t=1}^{j-1}$, $\{\mathbf{y}_{t,\mathsf{m}}^{(k,l)}\}_{t=j+1}^{T}$, $\mathbf{Y}_{\mathsf{o}}$, $\boldsymbol{\tau}^{(k+1,l)}$, and $\boldsymbol{\theta}^{(k)}$ for $j = p+1, \ldots, T$.
4: Stochastic approximation: evaluate $\hat{s}^{(k)}$ as in (17).
5: Maximization: update $\boldsymbol{\theta}^{(k+1)}$ as in (21), (23), and (24).
6: **if** stopping criteria is met **then**
7: terminate loop
8: **end if**
9: **end for**

Remark 5: It should be noted that we can also accelerate the procedure by first partitioning data into missing groups, and then performing the atom sampling on missing data inside each group in parallel. But the totally computational cost should be exactly the same as performing the atom sampling on the original $\mathbf{Y}_{\mathsf{m}}$.

VI. Complexity Analysis

In this section, we give a detailed discussion on the computational complexity of two Gibbs sampling schemes in our

proposed algorithm. We analyze the per-iteration complexity of the two Gibbs sampling schemes on $\mathbf{Y}_{\mathsf{m}}$, i.e., the entire sampling and atom sampling. In practice, we shall always choose to perform the Gibbs sampling by partitioning missing groups, which enable us to do the sampling in parallel without incurring additional computational cost. Therefore, it is reasonable to assume that $\mathbf{Y}_{\mathsf{m}}$ has already been a missing group partitioned out from the original data. In addition, we assume that $\mathbf{Y}$ is an observed data matrix of T_{m} ($T_{\mathsf{m}} < T$) observations contain missing values.

1) Entire Sampling: In each Markov chain of one entire sampling, we shall first compute the Gaussian moments $\tilde{\boldsymbol{\mu}}$ and $\tilde{\boldsymbol{\Sigma}}$. To simplify the computation, we can first compute and store $\{\mathbf{B}^j\}_{j=0}^{T-p-1}$, whose computational cost is $\mathcal{O}(N^3p^3(T-p-2))$. Then, for computing $\tilde{\boldsymbol{\mu}}$, the cost is $\mathcal{O}(N^2p^2(T-p-1)+N^2(p+1)(T-p))$; for computing $\tilde{\boldsymbol{\Sigma}}$, the cost is $\mathcal{O}((2N^3+N^2)(T-p)^2)$. The computational cost for drawing $\mathbf{Y}_{\mathsf{m}}$ from the conditional Gaussian distribution is $\mathcal{O}(N^3(T-p)^3)$. So, the total per iteration cost for the entire sampling scheme is about $\mathcal{O}(N^3p^3(T-p)+LN^3(T-p)^3)$.

2) Atom Sampling: In each Markov chain of one atom sampling, we can regard the sampling procedure as employing entire sampling in T_{m} short time series of length $2p+1$. Therefore in each iteration the computational cost for the atom sampling scheme is $\mathcal{O}(N^3p^4+LN^3p^3T_{\mathsf{m}})$. The above complexity analysis is consistent with our numerical analysis in Figure 5.

The two Gibbs sampling schemes should be properly chosen in order to reduce the computational cost as much as possible. From the above analysis, we recommend to use the atom sampling when the T is too large compared with the order number p. Besides, it can be expected that using the entire sampling can converge faster in terms of required iterations than using the atom sampling.

VII. Numerical Experiments

In this section, we illustrate the performance of our proposed methods on estimating the Student's t VAR parameters using both synthetic data and real data. We first compare the two proposed estimation methods in term of time usage and estimation error. The robustness of the two algorithms will be further assessed by feeding the algorithm with corrupted data containing missing values and outliers. Then in real data experiments, we make the prediction on stock returns by fitting our proposed Student's t VAR model with the historical returns. The dataset is always partitioned into missing groups when possible.

A. Synthetic Data

We first illustrate the performance of our proposed methods in synthetic datasets. We consider a VAR(2) model with number of variables $N = 20$, the diagonal elements of $\boldsymbol{\Phi}_1^{\mathsf{true}}$ and $\boldsymbol{\Phi}_2^{\mathsf{true}}$ are respectively set as 0.3 and 0.1, and the elements of $\boldsymbol{\phi}_0^{\mathsf{true}}$ and off-diagonal elements of $\boldsymbol{\Phi}_1^{\mathsf{true}}$ and $\boldsymbol{\Phi}_2^{\mathsf{true}}$ are independently drawn from a uniform distribution on $[-0.1, 0.1]$. The data is generated with innovations following the Student's t distribution with $\nu^{\mathsf{true}} = 5$ and $\boldsymbol{\Sigma}^{\mathsf{true}}$ a Toeplitz covariance matrix of the form $(\boldsymbol{\Sigma}^{\mathsf{true}})_{ij} = 0.5^{|i-j|}$. We also manually set some elements in $\mathbf{Y}$ missing in such way: some columns of $\mathbf{Y}$ are selected and 10 elements of each selected column are randomly removed. The missing percentage is defined as the ratio of the number of picked columns over T. The SAEM-MCMC algorithms are implemented with $L = 10$ Markov chains and the parameters are initialized as $\nu^{(0)} = 6$, $\boldsymbol{\Sigma}^{(0)} = \mathbf{I}$, and $\boldsymbol{\Psi}^{(0)} = [\mathbf{0}\ \mathbf{I}\ \mathbf{I}]$. For the step size, we set $\gamma^{(k)} = 1$ for $1 \le k \le 50$ and $\gamma^{(k)} = \frac{1}{k-50}$ for $k > 50$.

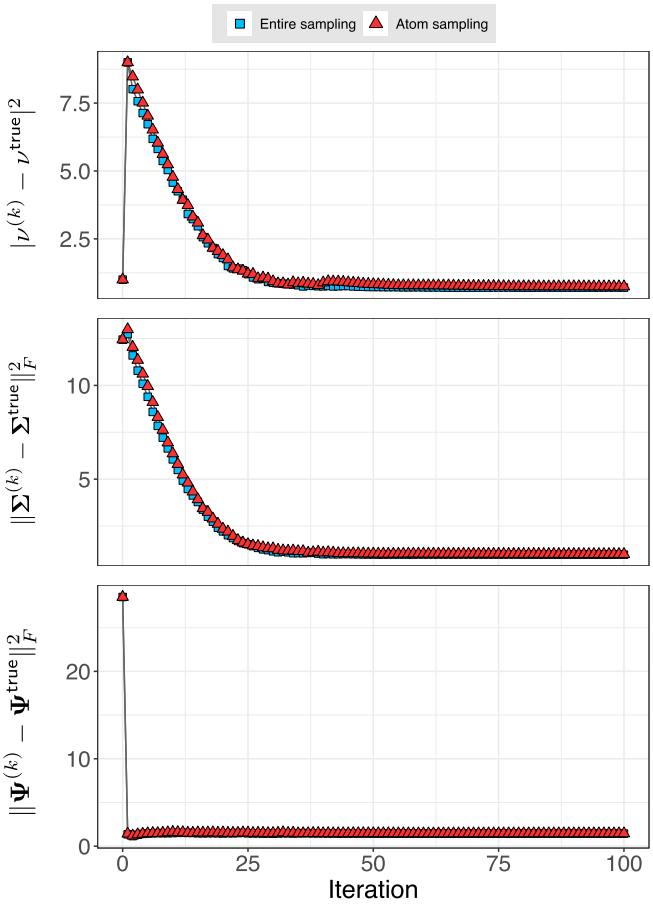


Fig. 4. Estimation error versus iterations.

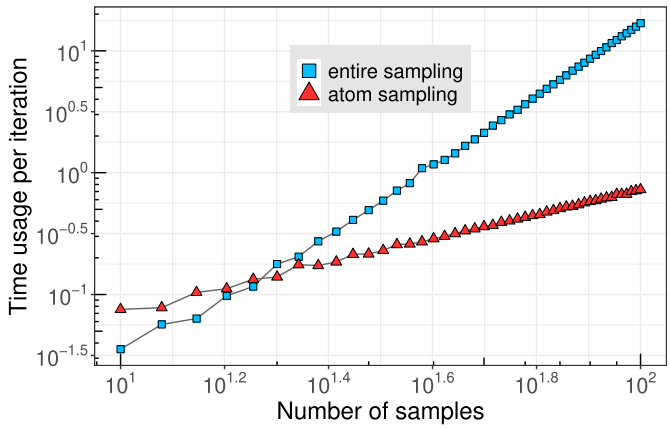


Fig. 5. Comparison on time usage per iteration versus the number of samples with consecutive missing values.

1) Algorithm Performance: Here we compare our two proposed algorithms in terms of final estimation results and time usage in each iteration. First, we generate the data with number of samples $T = 800$ and set missing percentage to 20% by manually deleting some randomly selected data. Figure 4 shows the square error of the estimated parameters versus iterations. The two proposed algorithms based on different sampling schemes can finally converge to almost the same results. The entire sampling method seems to converge faster than the atom sampling method. However, as we have discussed before, the computational load of the entire sampling become unacceptable when T grows large. In Figure 5, we set $N = 4$ and compare the average per-iteration time usage for sampling of two proposed schemes. It shows that the entire sampling has advantage over the atom sampling in terms of computational time when the T is relatively small. Besides, we also observe that the two proposed algorithms are not sensitive to the initialization in terms of final estimation accuracy.

2) Robustness To Missing Data: Now we illustrate the robustness of our proposed algorithms in comparison with other benchmarks. As we have mentioned in the introduction, the parameter estimation method assuming the Gaussian VAR model has been developed [20]. Besides, we consider two simple but general methods to dealing with missing data: the omit-variable method and imputation method. The omit-variable method excludes the variables with missing data from the analysis, i.e., removes the term $p(\mathbf{y}_t|\mathbf{\Psi}, \mathbf{\Sigma}, \nu, \mathbf{y}_{t-p}, \ldots, \mathbf{y}_{t-1})$ in $l(\boldsymbol{\theta}; \mathbf{Y})$ if any missing data exist in $\mathbf{y}_{t-p}, \ldots, \mathbf{y}_t$. The imputation method replaces the missing data with substituted data and then uses the imputed dataset to estimate the parameters. We use the popular R package Amelia, which can be used to generate several imputed datasets [27]. As suggested in [27], 5 imputed datasets are probably adequate unless the missing percentage is very high. The final estimates from the imputation method can be obtained by taking the average of estimates over all imputed datasets. The data are generated exactly following the Student's t VAR model. We first set the missing percentage to 20% and compare the final estimation results versus the number of samples T. The criteria of estimation performance is chosen as the mean square error (MSE) between the final estimate and the true parameters, i.e.,

$$\begin{aligned} \text{MSE}\left(\frac{\hat{\nu}}{\hat{\nu}-2}\hat{\mathbf{\Sigma}}\right) &= \frac{1}{R}\sum_{r=1}^{R}\left\|\frac{\hat{\nu}_r}{\hat{\nu}_r-2}\hat{\mathbf{\Sigma}}_r - \frac{\nu^{\text{true}}}{\nu^{\text{true}}-2}\mathbf{\Sigma}^{\text{true}}\right\|_F^2, \\ \text{MSE}(\hat{\mathbf{\Psi}}) &= \frac{1}{R}\sum_{r=1}^{R}\|\hat{\mathbf{\Psi}}_r - \mathbf{\Psi}^{\text{true}}\|_F^2, \end{aligned} \tag{40}$$

where R is number of data realizations and is set to 200 in our simulation part. In Figure 6, we can see that the MSE of all parameters decreases when T grows large, and our proposed algorithms show the best performance. Besides, we can see that assuming Student's t innovations is always better than assuming the Gaussian innovations. In Figure 7, we fix $T = 800$ and compare the final estimation results versus the missing percentage. With designed ability to directly handle the missing data, our

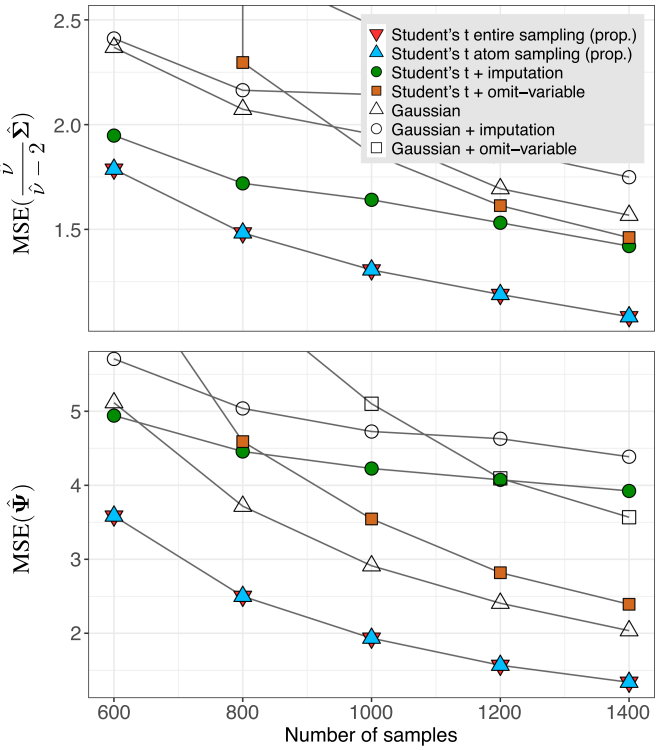


Fig. 6. Mean square error of estimated parameters versus the number of samples (with Student's t VAR synthetic data and missing percentage 20%).

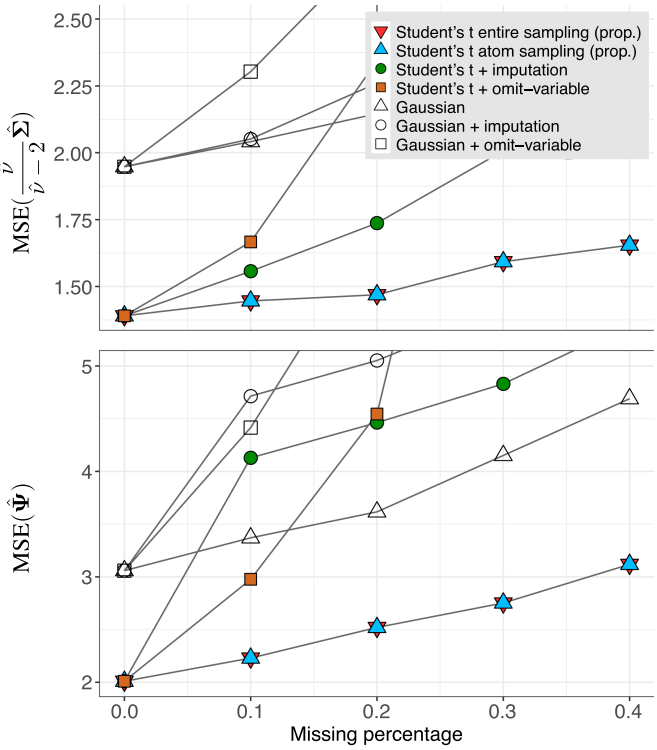


Fig. 7. Mean square error of estimated parameters versus the missing percentage (with Student's t VAR synthetic data and number of samples $T = 800$).

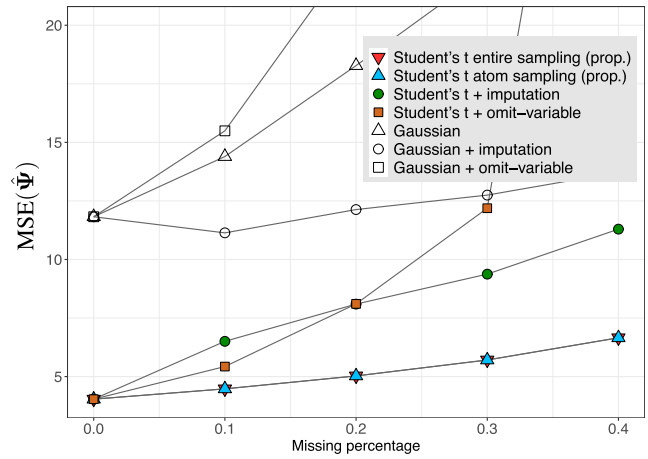


Fig. 8. Mean square error of the coefficient matrix versus the missing percentage (with Gaussian VAR synthetic data, number of samples $T = 800$, and outlier percentage 1.5%).

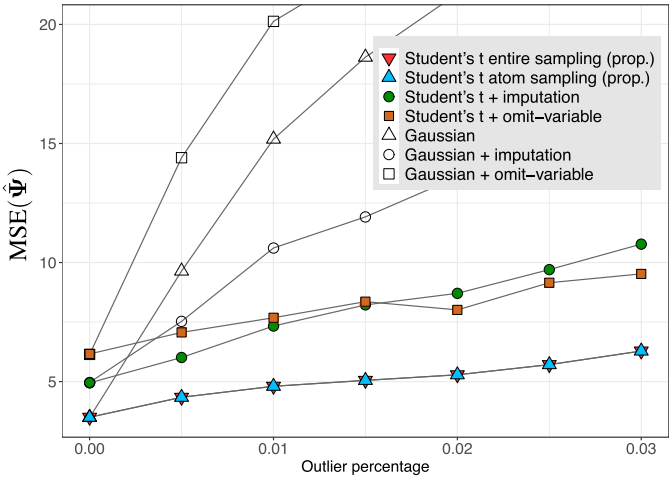


Fig. 9. Mean square error of the coefficient matrix versus the outlier percentage (with Gaussian VAR synthetic data, number of samples $T = 800$, and missing percentage 20%).

proposed algorithms achieve the best performance and are most unresponsive to missing value.

3) Robustness to Outliers: Now we illustrate the robustness of our proposed algorithm to outliers. We set $T = 800$ and generate the data with innovations following the Gaussian distribution, i.e., with $\nu \to +\infty$ in Student's t VAR model. Then the outliers are added in this way: some columns of $\mathbf{Y}$ are picked and 10 element of each picked column is randomly chosen and set to be 20. The outlier percentage is defined as the ratio of the number of picked columns over T. In Figure 8, we set the outlier percentage to 1.5% and compare the estimation performance versus missing percentage. It is clear the difference in estimation between assuming a Gaussian distribution or a Student's t. In Figure 9, we show the estimation results versus the outlier percentage. It should be emphasized that when outlier percentage is set as 0, i.e., the synthetic data is generated following exactly Gaussian VAR model, our proposed methods can achieve almost the same

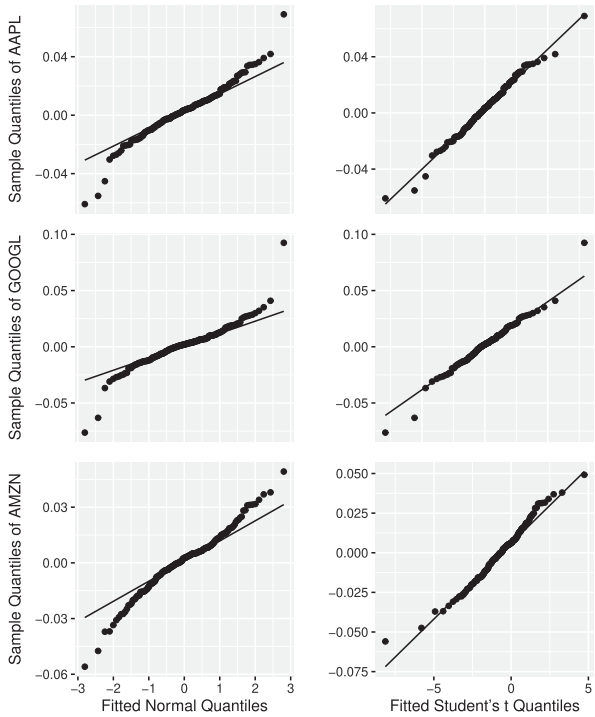


Fig. 10. Quantile-quantile plot of the innovations of real stocks returns.

estimation results as method assuming Gaussian innovation. Similar to previous simulation, our proposed algorithms can always achieve the best performance.

B. Real Data

Here we show a practical application of Student's t VAR model in predicting the future stocks return. Considering the daily returns of the three stocks: AAPL, GOOGL, and AMZN from Jan. 2019 to Oct. 2019 of overall 200 observations (excluding the weekends and public holidays). The order of VAR model is selected as $p = 1$ via Akaike information criterion [28] by fitting the data with the Student's t VAR model of different orders. In Figure 10, we first fit the stock returns using the Gaussian VAR model and the Student's t VAR model, and then plot the quantile-quantile (QQ) plot of the corresponding innovations versus the fitted Gaussian distribution and Student's t distribution. It is obvious that the innovations fits better with Student's t distribution.

We use the first 160 observations as the training data and the last 40 observations as the testing data. First, we fit the Student's t VAR model and Gaussian VAR model with the training data and obtain the estimated coefficient matrix $\hat{\mathbf{\Psi}}$. Then, we make the one-step-ahead predictions for the test data as $\hat{\mathbf{y}}_t = \hat{\mathbf{\Psi}}\mathbf{x}_{t-1}$ with $t = 161, \ldots, 200$, and measure the mean square prediction error (MSPE) as $\frac{1}{40}\sum_{t=161}^{200} \|\mathbf{y}_t - \hat{\mathbf{y}}_t\|_2^2$. Further more, we corrupt the training dataset by randomly picking 10% observations and removing 1 data points in each observation. The fitting procedure and the prediction using two models are performed again but fed with the corrupted training data. The one-step-ahead prediction performances of the models are reported in Table I where the relative MSPE of each model is calculated by comparing with the

TABLE I
COMPARISON OF PREDICTION ON REAL STOCK RETURNS

	Methods	MSPE	Relative MSPE	p-values		
				AAPL	GOOGL	AMZN
Complete data	Gaussian VAR	5.574×10^{-4}	93.3%	> 0.1	0.096	0.031
	Student's t VAR	5.518×10^{-4}	92.4%	> 0.1	0.098	0.035
Incomplete data	Gaussian VAR + omit-variable	6.150×10^{-4}	103.0%	> 0.1	> 0.1	0.025
	Gaussian VAR + imputation	6.099×10^{-4}	102.1%	> 0.1	> 0.1	0.043
	Gaussian VAR	5.972×10^{-4}	100%	–	–	–
	Student's t VAR + omit-variable	6.478×10^{-4}	108.5%	0.038	0.037	0.058
	Student's t VAR + imputation	6.237×10^{-4}	104.4%	0.057	> 0.1	0.042
	Student's t VAR (prop.)	5.674×10^{-4}	94.9%	0.077	> 0.1	0.025

Gaussian VAR model method directly fed with incomplete data. The Diebold-Mariano (DM) test [29] for the null hypothesis of global equal performances between two predictors is performed with the following H_0 : the candidate method has equal prediction performance as the benchmark Gaussian VAR model method directly fed with incomplete data. Finally, our proposed algorithms report smaller MPSEs than those of the methods not capable of directly handing the missing data or based on the assumption of Gaussian distributed innovations. Even fed with corrupted data set, the prediction from our proposed algorithms are still acceptable in comparsion with methods fed with complete data. Compared with the benchmark method, our proposed methods achieves significantly different results with almost 5% smaller MSPE. Besides, with the ability of directly handling the missing values, Gaussian VAR and our proposed Student's t VAR model can provide better MSPE results than those of methods based on omit-variable and imputation. This is because omit-variable method ignores part of the measured data, resulting in the information loss. While imputation method draws the missing data, which might not follow the statistical properties of the available time series data.

VIII. CONCLUSION

In this paper, we have considered the parameter estimation for the VAR model with heavy-tailed innovations and missing data. We have formulated the MLE problem for estimating the parameters assuming the innovations follow the multivariate Student's t distribution. An algorithmic framework based on SAEM-MCMC algorithm has been proposed to solve the MLE problem. In the framework, two sampling schemes have been proposed to draw the random realizations of missing data. The two optional sampling schemes are particularly designed for reducing computational cost under different problem dimensions and missing patterns. We have shown in the numerical experiments that our proposed framework has great advantage in resisting the missing values and outliers, and the two sampling schemes can achieve almost the same estimation results.

APPENDIX

A. EM and Its Stochastic Variants

The EM algorithm is a very powerful iterative algorithm to solve the MLE problem with missing values or latent variables [30]. Suppose we have observed the data $\mathbf{Y}_\mathsf{o}$ from a statistical model described by the parameter set $\boldsymbol{\theta}$, the MLE problem formulation is

$$\underset{\boldsymbol{\theta}}{\text{maximize}} \quad l\left(\boldsymbol{\theta}; \mathbf{Y}_\mathsf{o}\right), \tag{41}$$

where $l(\boldsymbol{\theta}; \mathbf{Y}_\mathsf{o}) = \log p(\mathbf{Y}_\mathsf{o}|\boldsymbol{\theta})$ is the log-likelihood of $\mathbf{Y}_\mathsf{o}$ given $\boldsymbol{\theta}$. Sometimes, the objective $l(\boldsymbol{\theta}; \mathbf{Y}_\mathsf{o})$ might not have a manageable expression since part of the data is missing or some latent variables cannot be observed. Then it could be very difficult to directly solve such MLE problem. The EM algorithm was proposed to deal with this by converting the maximization for $l(\boldsymbol{\theta}; \mathbf{Y}_\mathsf{o})$ into the maximization of a sequence of simpler and solvable problems. More specifically, denote by $\mathbf{Z}$ the missing data and latent variables, the EM algorithm solves the MLE problem by iteratively applying the expectation (E) step and maximization (M) step until convergence:

1) **E step**: calculate the expected log-likelihood of the complete data $(\mathbf{Y}_\mathsf{o}, \mathbf{Z})$ over the current conditional distribution of $\mathbf{Z}$ given $\mathbf{Y}_\mathsf{o}$ and current estimate of the parameters $\boldsymbol{\theta}^{(k)}$:

$$Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)}\right) = \int \log p\left(\mathbf{Y}_\mathsf{o}, \mathbf{Z}|\boldsymbol{\theta}\right) p\left(\mathbf{Z}|\mathbf{Y}_\mathsf{o}, \boldsymbol{\theta}^{(k)}\right) \mathrm{d}\mathbf{Z}, \tag{42}$$

where k is the iteration index.

2) **M step**: solve the optimization problem to update $\boldsymbol{\theta}$ as

$$\boldsymbol{\theta}^{(k+1)} = \arg\max_{\boldsymbol{\theta}} Q\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)}\right). \tag{43}$$

Actually, the EM algorithm is a particular instance of the more general majorization-minimization (MM) framework [31].

The EM algorithm is very useful for MLE problems with latent variables but may have some technical difficulties. One of the difficulties is obtaining the closed-form expression for the expectation $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})$. The Monte Carlo EM (MCEM) was proposed to tackle such difficulty by approximating the exact expectation with the sample average of many random simulations of latent variables [32]. But it is often criticized for being computationally intensive as it requires a large number of simulations to approximate well the expectation.

The stochastic approximation EM (SAEM) was proposed to reduce the number of simulations in the MCEM algorithm [33]. It combines the current simulations with the information in the previous expectation step. More specifically, the SAEM method can be summarized by iteratively performing the stochastic E step and M step:

1) **Stochastic E step**: first draw L realizations $\mathbf{Z}^{(k+1,l)}$ from the conditional distribution $p(\mathbf{Z}|\mathbf{Y}_\mathsf{o}, \boldsymbol{\theta}^{(k)})$ with $l =$

$1, \ldots, L$. Then calculate the $\hat{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)})$ by

$$\hat{Q}\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k-1)}\right) + \gamma^{(k)}\left(\frac{1}{L}\sum_{l=1}^{L}\log p\left(\mathbf{Y}_{\mathrm{o}}, \mathbf{Z}^{(k+1,l)}|\boldsymbol{\theta}\right) - \hat{Q}\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k-1)}\right)\right), \tag{44}$$

where $\{\gamma_k\}$ is a decreasing sequence of positive step sizes.

2) **M step**: solve the optimization problem to update $\boldsymbol{\theta}$ as

$$\boldsymbol{\theta}^{(k+1)} = \arg\max_{\boldsymbol{\theta}} \hat{Q}\left(\boldsymbol{\theta}|\boldsymbol{\theta}^{(k)}\right). \tag{45}$$

When the conditional distribution $p(\mathbf{Z}|\mathbf{Y}_{\mathrm{o}}, \boldsymbol{\theta})$ is very complicated, and the sampling in the stochastic E step of the SAEM cannot be directly performed, Kuhn and Lavielle proposed to combine the SAEM algorithm with a Markov Chain Monte Carlo (MCMC) procedure, which yields the SAEM-MCMC algorithm [22]. The MCMC is a popular method for sampling from a probability distribution by constructing a Markov chain. Suppose the conditional distribution $p(\mathbf{Z}|\mathbf{Y}_{\mathrm{o}}, \boldsymbol{\theta})$ is the unique stationary distribution of the transition probability density function $\Pi_{\boldsymbol{\theta}}$, the sampling part of the SAEM is replaced with drawing realizations $\mathbf{Z}^{(k+1,l)}(l = 1, 2 \ldots, L)$ based on the transition probability density function $\Pi_{\boldsymbol{\theta}^{(k)}}(\mathbf{Z}^{(k,l)}, \cdot)$. For each l, the sequence $\{\mathbf{Z}^{(k,l)}\}_{k\geq 0}$ is a Markov chain with the transition probability density function $\{\Pi_{\boldsymbol{\theta}^{(k)}}\}$. The Markov Chain generation mechanism needs to be well designed so that the sampling is efficient and the computational cost is not too high.

B. Proof for Lemma 1

Given $\mathbf{Y}$ and $\boldsymbol{\theta}$, the conditional pdf of the $\boldsymbol{\tau}$ is

$$\begin{aligned} p(\boldsymbol{\tau}|\mathbf{Y}, \boldsymbol{\theta}) &= \frac{p(\mathbf{Y}, \boldsymbol{\tau}|\boldsymbol{\theta})}{p(\mathbf{Y}|\boldsymbol{\theta})} \propto p(\mathbf{Y}, \boldsymbol{\tau}|\boldsymbol{\theta}) \\ &= \prod_{t=p+1}^{T} \frac{\left(\frac{\nu}{2}\right)^{\frac{\nu}{2}}(\tau_t)^{\frac{\nu}{2}-1}\exp\left(-\frac{\nu}{2}\tau_t\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\det\left(2\pi\boldsymbol{\Sigma}/\tau_t\right)}}\exp\left(-\frac{\nu+\delta_t}{2}\tau_t\right) \\ &\propto \prod_{t=p+1}^{T} \tau_t^{\frac{\nu+N}{2}-1}\exp\left(-\frac{\nu+\delta_t}{2}\tau_t\right), \end{aligned} \tag{46}$$

where $\delta_t = (\mathbf{y}_t - \boldsymbol{\Psi}\mathbf{x}_{t-1})^T\boldsymbol{\Sigma}^{-1}(\mathbf{y}_t - \boldsymbol{\Psi}\mathbf{x}_{t-1})$. It implies that every τ_t is independent from each other with

$$p(\tau_t|\mathbf{Y}, \boldsymbol{\theta}) \propto \tau_t^{\frac{\nu+N}{2}-1}\exp\left(-\frac{\nu+\delta_t}{2}\tau_t\right). \tag{47}$$

Comparing this expression with the pdf of the gamma distribution, we get that $\tau_t|\mathbf{Y}, \boldsymbol{\theta}$ follows a gamma distribution:

$$\tau_t|\mathbf{Y}, \boldsymbol{\theta} \sim Gamma\left(\frac{\nu+N}{2}, \frac{\nu+\delta_t}{2}\right). \tag{48}$$

C. Proof for Lemma 2

Denoting by $\mathbf{w}_t = [\mathbf{y}_t^T \cdots \mathbf{y}_{t-p+1}^T]^T$, $\boldsymbol{\alpha} = [\boldsymbol{\phi}_0^T, \mathbf{0}] \in \mathbb{R}^{N(T-p)}$ and $\mathbf{v}_t = [\boldsymbol{\varepsilon}_t, \mathbf{0}] \in \mathbb{R}^{N(T-p)}$, we can write the VAR($p$) model into VAR(1) form:

$$\mathbf{w}_t = \boldsymbol{\alpha} + \mathbf{B}\mathbf{w}_{t-1} + \mathbf{v}_t. \tag{49}$$

Through a recursive process based on VAR(1) form, we have

$$\begin{aligned} \mathbf{w}_{p+1} &= \boldsymbol{\alpha} + \mathbf{B}\mathbf{w}_p + \mathbf{v}_{p+1} \\ \mathbf{w}_{p+2} &= \boldsymbol{\alpha} + \mathbf{B}\mathbf{w}_{p+1} + \mathbf{v}_{p+2} \\ &= \boldsymbol{\alpha} + \mathbf{B}\left(\boldsymbol{\alpha} + \mathbf{B}\mathbf{w}_p + \mathbf{v}_{p+1}\right) + \mathbf{v}_{p+2} \\ &= (\mathbf{I} + \mathbf{B})\,\boldsymbol{\alpha} + \mathbf{B}^2\mathbf{w}_p + \mathbf{B}\mathbf{v}_{p+1} + \mathbf{v}_{p+2} \\ &\vdots \\ \mathbf{w}_t &= \boldsymbol{\alpha} + \mathbf{B}\mathbf{w}_{t-1} + \mathbf{v}_t \\ &= \boldsymbol{\alpha} + \mathbf{B}\left(\boldsymbol{\alpha} + \mathbf{B}\mathbf{w}_{t-2} + \mathbf{v}_t\right) + \mathbf{v}_t \\ &\vdots \\ &= \sum_{j=0}^{t-p-1}\mathbf{B}^j\boldsymbol{\alpha} + \mathbf{B}^{t-p}\mathbf{w}_p + \sum_{j=p+1}^{t}\mathbf{B}^{t-j}\mathbf{v}_j \end{aligned} \tag{50}$$

With the usage of above recursive expression, we can extract $\mathbf{y}_t$ as

$$\mathbf{y}_t = \left(\sum_{j=0}^{t-p-1}\mathbf{B}^j\right)_{[N]}\boldsymbol{\phi}_0 + \left(\mathbf{B}^{t-p}\mathbf{w}_p\right)_{[N]} + \sum_{j=p+1}^{t}\left(\mathbf{B}^{t-j}\right)_{[N]}\boldsymbol{\varepsilon}_j, \tag{51}$$

which shows that the $\tilde{\mathbf{y}}_{-p}$ is sum of a constant vector plus a affine transformation of a multivariate Gaussian random variables. Therefore, $\tilde{\mathbf{y}}_{-p}$ still follows the multivariate Gaussian distribution $\mathcal{N}(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}})$ with

$$\begin{aligned} \left(\tilde{\boldsymbol{\mu}}\right)_{(i)} &= \mathsf{E}\left[\mathbf{x}_{i+p}\right] \\ &= \mathsf{E}\left[\left(\sum_{j=0}^{i-1}\mathbf{B}^j\right)_{[N]}\boldsymbol{\phi}_0 + \left(\mathbf{B}^i\mathbf{w}_p\right)_{[N]} + \sum_{j=p+1}^{i+p}\left(\mathbf{B}^{i+p-j}\right)_{[N]}\boldsymbol{\varepsilon}_j\right] \\ &= \left(\sum_{j=0}^{i-1}\mathbf{B}^j\right)_{[N]}\boldsymbol{\phi}_0 + \left(\mathbf{B}^i\mathbf{w}_p\right)_{[N]}, \end{aligned} \tag{52}$$

$$\begin{aligned} \left(\tilde{\boldsymbol{\Sigma}}\right)_{(ij)} &= \mathsf{E}\left[\left(\mathbf{x}_{i+p} - \mathsf{E}\left[\mathbf{x}_{i+p}\right]\right)\left(\mathbf{x}_{j+p} - \mathsf{E}\left[\mathbf{x}_{j+p}\right]\right)^T\right] \\ &= \mathsf{E}\left[\left(\sum_{q=p+1}^{i+p}\left(\mathbf{B}^{i+p-q}\right)_{[N]}\boldsymbol{\varepsilon}_q\right)\left(\sum_{r=p+1}^{j+p}\left(\mathbf{B}^{j+p-r}\right)_{[N]}\boldsymbol{\varepsilon}_r\right)^T\right] \\ &= \mathsf{E}\left[\left(\sum_{q=p+1}^{i+p}\sum_{r=p+1}^{j+p}\left(\mathbf{B}^{i+p-q}\right)_{[N]}\boldsymbol{\varepsilon}_q\boldsymbol{\varepsilon}_r^T\right)^T\right] \\ &= \sum_{q=p+1}^{\min(i,j)+p}\frac{1}{\tau_q}\left(\mathbf{B}^{i+p-q}\right)_{[N]}\boldsymbol{\Sigma}\left(\left(\mathbf{B}^{j+p-q}\right)_{[N]}\right)^T \\ &= \sum_{q=1}^{\min(i,j)}\frac{1}{\tau_{q+p}}\left(\mathbf{B}^{i-q}\right)_{[N]}\boldsymbol{\Sigma}\left(\left(\mathbf{B}^{j-q}\right)_{[N]}\right)^T. \end{aligned} \tag{53}$$

D. Proof for Lemma 4

Given $\boldsymbol{\theta}$, $\boldsymbol{\tau}$, and $\mathbf{Y}_{\mathsf{o}}$, the conditional distribution of $\{\mathbf{y}_{t,\mathsf{m}}\}_{t=m_1}^{m_2}$ is

$$\begin{aligned} & p\left(\{\mathbf{y}_{t,\mathsf{m}}\}_{t=m_1}^{m_2}|\boldsymbol{\tau},\boldsymbol{\theta},\mathbf{Y}_{\mathsf{o}}\right) \\ &= \frac{p\left(\mathbf{Y}_{\mathsf{m}}|\boldsymbol{\tau},\boldsymbol{\theta},\mathbf{Y}_{\mathsf{o}}\right)}{p\left(\{\mathbf{y}_{t,\mathsf{m}}\}_{t=p+1}^{m_1-1},\{\mathbf{y}_{t,\mathsf{m}}\}_{t=m_2+1}^{T}|\boldsymbol{\tau},\boldsymbol{\theta},\mathbf{Y}_{\mathsf{o}}\right)} \\ &\propto p\left(\mathbf{Y}_{\mathsf{m}}|\boldsymbol{\tau},\boldsymbol{\theta},\mathbf{Y}_{\mathsf{o}}\right) \\ &= \frac{p\left(\mathbf{Y}_{\mathsf{m}},\mathbf{Y}_{\mathsf{o}}|\boldsymbol{\tau},\boldsymbol{\theta}\right)}{p\left(\mathbf{Y}_{\mathsf{o}}|\boldsymbol{\tau},\boldsymbol{\theta}\right)} \\ &\propto p\left(\mathbf{Y}_{\mathsf{m}},\mathbf{Y}_{\mathsf{o}}|\boldsymbol{\tau},\boldsymbol{\theta}\right) \\ &= \prod_{t=p+1}^{T} f_{\mathrm{MVT}}\left(\mathbf{y}_t;\boldsymbol{\Psi}\mathbf{x}_{t-1},\boldsymbol{\Sigma},\nu\right). \end{aligned} \tag{54}$$

Note here $\mathbf{x}_{t-1}=[1,\mathbf{y}_{t-1}^T,\cdots,\mathbf{y}_{t-p}^T]^T$, which implies that the distribution of $\{\mathbf{y}_{t,\mathsf{m}}\}_{t=m_1}^{m_2}$ depend only on $\boldsymbol{\tau}$, $\boldsymbol{\theta}$, and $\{\mathbf{y}_{t,\mathsf{o}}\}_{t=m_1-p}^{m_2+p}$.

REFERENCES

[1] R. S. Tsay, *Analysis of Financial Time Series*. Hoboken, NJ, USA, Wiley, 2005.

[2] J. D. Hamilton, *Time Series Analysis*. Princeton, NJ, USA, Princeton Univ. Press, 1994.

[3] J. H. Stock and M. W. Watson, "Vector autoregressions," *J. Econ. Perspectives*, vol. 15, no. 4, pp. 101–115, 2001.

[4] A. C. Emerencia, L. van der Krieke, E. H. Bos, P. de Jonge, N. Petkov, and M. Aiello, "Automating vector autoregression on electronic patient diary data," *IEEE J. Biomed. Health Inform.*, vol. 20, no. 2, pp. 631–643, Mar. 2016.

[5] P. Taveeapiradeecharoen, C. Jongsureyapart, and N. Aunsri, "Forecasting Daily Forex Using Large Dimensional Vector Autoregression with Time-Varying Parameters," in *Proc. Glob. Wireless Summit*, Nov. 2018, pp. 65–70.

[6] B. O. Bradley and M. S. Taqqu, "Financial risk and heavy tails," in *Handbook Heavy Tailed Distributions Finance*. Amsterdam, The Netherlands; NY, USA, Elsevier, 2003, pp. 35–103.

[7] J. Nair, A. Wierman, and B. Zwart, "The fundamentals of heavy-tails: Properties, emergence, and identification," in *Proc. ACM SIGMETRICS/Int. Conf. Meas. Model. Comput. Syst.*, 2013, pp. 387–388.

[8] Y. Feng and D. P. Palomar, *A Signal Processing Perspective On Financial Engineering*. Now Publishers, 2016.

[9] D. M. Hawkins, *Identification Of Outliers*. New York; Berlin, Germany, Springer, 1980, vol. 11.

[10] Y. Zhang, N. Meratnia, and P. J. Havinga, "Outlier detection techniques for wireless sensor networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 2, pp. 159–170, Apr.–Jun. 2010.

[11] H. Desai and P. C. Jain, "Long-run common stock returns following stock splits and reverse splits," *J. Bus.*, vol. 70, no. 3, pp. 409–433, Jul. 1997.

[12] C. Liu and D. B. Rubin, "ML estimation of the t distribution using EM and its extensions, ECM and ECME," *Statistica Sinica*, vol. 5, no. 1, pp. 19–39, Jan. 1995.

[13] J. Liu, S. Kumar, and D. P. Palomar, "Parameter estimation of heavy-tailed AR model with missing data via stochastic EM," *IEEE Trans. Signal Process.*, vol. 67, no. 8, pp. 2159–2172, Apr. 2019.

[14] S. Nadarajah and S. Kotz, "Estimation methods for the multivariate t distribution," *Acta Applicandae Mathematicae*, vol. 102, no. 1, pp. 99–118, May 2008.

[15] D. Fresoli, E. Ruiz, and L. Pascual, "Bootstrap multi-step forecasts of non-Gaussian VAR models," *Int. J. Forecasting*, vol. 31, no. 3, pp. 834–848, Jul. 2015.

[16] A. Mirniam and A. Nematollahi, "Maximum likelihood estimation in vector autoregressive models with multivariate scaled t-distributed innovations using EM-based algorithms," *Commun. Statist.-Simul. Comput.*, vol. 47, no. 3, pp. 890–904, Mar. 2018.

[17] S. Shakkottai, R. Srikant, and N. B. Shroff, "Unreliable sensor grids: Coverage, connectivity and diameter," *Ad Hoc Netw.*, vol. 3, no. 6, pp. 702–716, Nov. 2005.

[18] C. W. J. Chiu, B. Eraker, A. T. Foerster, T. B. Kim, and H. D. Seoane, "Estimating VAR's sampled at mixed or irregular spaced frequencies: A Bayesian approach," *Federal Reserve Bank of Kansas City, RWP*, 2011.

[19] R. J. Little and D. B. Rubin, *Statistical Analysis With Missing Data*. Hoboken, NJ, USA, Wiley, 2019.

[20] J. Antony and T. Klarl, "An EM-Algorithm for maximum likelihood estimation of vector autoregressions with mixed frequency data," 2017, manuscript submitted for publication.

[21] J. Liu, S. Kumar, and D. P. Palomar, "Parameter estimation of heavy-tailed AR(p) model from incomplete data," in *Proc. Eur. Signal Process. Conf.*, A Coruña, Spain, Sep. 2019, pp. 2–6.

[22] E. Kuhn and M. Lavielle, "Coupling a stochastic approximation version of EM with an MCMC procedure," *ESAIM: Probability Statist.*, vol. 8, pp. 115–131, Aug. 2004.

[23] H. Lütkepohl, *New Introduction To Multiple Time Series Analysis*. New York; Berlin, Germany; Springer Science & Business Media, 2005.

[24] G. DiCesare, "Imputation, estimation and missing data in finance," Ph.D thesis, Dept. Statis., Univ. Waterloo, Canada, 2006.

[25] R. A. Davis, P. Zang, and T. Zheng, "Sparse vector autoregressive modeling," *J. Comput. Graphical Statist.*, vol. 25, no. 4, pp. 1077–1096, Oct. 2016.

[26] M. L. Eaton, Multivariate statistics: A vector space approach, Hoboken, NJ, USA, Wiley.

[27] J. Honaker, G. King, and M. Blackwell *et al.*, "Amelia II: A program for missing data," *J. Stat. Softw.*, vol. 45, no. 7, pp. 1–47, Dec. 2011.

[28] K. Aho, D. Derryberry, and T. Peterson, "Model selection for ecologists: The worldviews of AIC and BIC," *Ecology*, vol. 95, no. 3, pp. 631–636, Mar. 2014.

[29] F. X. Diebold and R. S. Mariano, "Comparing predictive accuracy," *J. Bus. Econ. Statist.*, vol. 13, no. 3, pp. 253–265, Jan. 1995.

[30] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc., Ser. B*, vol. 39, no. 1, pp. 1–38, Sep. 1977.

[31] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 794–816, Feb. 2017.

[32] G. C. Wei and M. A. Tanner, "A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms," *J. Amer. Stat. Assoc.*, vol. 85, no. 411, pp. 699–704, Sep. 1990.

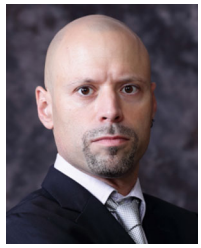
[33] B. Delyon, M. Lavielle, and E. Moulines, "Convergence of a stochastic approximation version of the EM algorithm," *Ann. Statist.*, vol. 27, no. 1, pp. 94–128, Feb. 1999.

Rui Zhou received the B.Eng. degree in information engineering from Southeast University, Nanjing, China in 2017. He is currently working toward the Ph.D. degree with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong.

His research interests include optimization algorithms, statistical signal processing, machine learning, and financial engineering.

Junyan Liu received the B.Eng. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China in 2015. She received the Ph.D. degree in electronic and computer engineeringfrom the Hong Kong University of Science and Technology, Hong Kong in 2020. She was the recipient of the Hong Kong Ph.D. Fellowship Scheme. Her research interests include data analytics and optimization algorithm design.

Sandeep Kumar received the B.Tech. degree from the College of Engineering Roorkee, India in 2011 and the M.Tech. and Ph.D. degrees from the Department of Electrical Engineering, the Indian Institute of Technology Kanpur, India, in 2013 and 2017, respectively. He was the recipient of the TCS Doctoral Fellowship. He worked as a Postdoctoral Researcher with the Department of Electronic and Computer Engineering and Department of Industrial Engineering and Decision Analytics, The Hong Kong University of Science and Technology, Hong Kong. He is currently working as an Assistant Professor with the Department of Electrical Engineering, The Indian Institute of Technology Delhi. His current research focuses on large scale non-convex optimization, graph-based algorithms, and signal processing techniques for applications in data analytics, communications, and networks.



Daniel P. Palomar (Fellow, IEEE) received the electrical engineering and Ph.D. degrees from the Technical University of Catalonia (UPC), Barcelona, Spain, in 1998 and 2003, respectively, and was a Fulbright Scholar with Princeton University during 2004–2006.

He is a Professor with the Department of Electronic & Computer Engineering and the Department of Industrial Engineering & Decision Analytics, the Hong Kong University of Science and Technology (HKUST), Hong Kong, which he joined in 2006. He had previously held several research appointments, namely, at King's College London (KCL), London, U.K., Stanford University, Stanford, CA, Telecommunications Technological Center of Catalonia (CTTC), Barcelona, Spain, Royal Institute of Technology (KTH), Stockholm, Sweden, University of Rome "La Sapienza", Rome, Italy, and Princeton University, Princeton, NJ.

His current research interests include applications of optimization theory, graph methods, and signal processing in financial systems and big data analytics.

Dr. Palomar was the recipient of the 2004/06 Fulbright Research Fellowship, the 2004 and 2015 (co-author) Young Author Best Paper Awards by the IEEE Signal Processing Society, the 2015–16 HKUST Excellence Research Award, the 2002/03 best Ph.D. prize in Information Technologies and Communications by the Technical University of Catalonia (UPC), the 2002/03 Rosina Ribalta first prize for the Best Doctoral Thesis in Information Technologies and Communications by the Epson Foundation, and the 2004 prize for the best Doctoral Thesis in Advanced Mobile Communications by the Vodafone Foundation and COIT.

He has been a Guest Editor of the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING 2016 special issue on "Financial Signal Processing and Machine Learning for Electronic Trading", an Associate Editor of IEEE TRANSACTIONS ON INFORMATION THEORY and of IEEE TRANSACTIONS ON SIGNAL PROCESSING, a Guest Editor of the *IEEE Signal Processing Magazine* 2010 special issue on "Convex Optimization for Signal Processing," the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS 2008 special issue on "Game Theory in Communication Systems," and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS 2007 special issue on "OPTIMIZATION OF MIMO TRANSCEIVERS FOR REALISTIC COMMUNICATION NETWORKS."