Bayesian Tensor Tucker Completion With A Flexible Core

Xueke Tong, Lei Cheng, and Yik-Chung Wu, *Senior Member, IEEE*

***Abstract*—Tensor completion is a vital task in multi-dimensional signal processing and machine learning. To recover the missing data in a tensor, various low-rank structures of a tensor can be assumed, and Tucker format is a popular choice. However, the promising capability of Tucker completion is realized only when we can determine a suitable multilinear rank, which controls the model complexity and thus is essential to avoid overfitting/underfitting. Rather than exhaustively searching the best multilinear rank, which is computationally inefficient, recent advances have proposed a Bayesian way to learn the multilinear rank from training data automatically. However, in prior arts, only a single parameter is dedicated to learn the variance of the core tensor elements. This rigid assumption restricts the modeling capabilities of existing methods in real-world data, where the core tensor elements may have a wide range of variances. To have a flexible core tensor while still retaining succinct Bayesian modeling, we first bridge the tensor Tucker decomposition to the canonical polyadic decomposition (CPD) with low-rank factor matrices, and then propose a novel Bayesian modeling based on the Gaussian-inverse Wishart prior. Inference algorithm is further derived under the variational inference framework. Extensive numerical studies on synthetic data and real-world datasets demonstrate the significantly improved performance of the proposed algorithm in terms of multilinear rank learning and missing data recovery.**

***Index Terms*—tensor decomposition, Bayesian Tucker model, multilinear rank estimation, Gaussian-Wishart priors**

I. Introduction

Tensor completion is widely used in high-dimensional data analytics as it could predict the missing values by using the hidden multilinear latent structures of the data. It found extensive applications in image processing [1]–[4], data mining [5], [6], machine learning [7], [8], and computer vision [9]–[11]. In particular, the commonly used underlying models are Canonical Polyadic decomposition (CPD) [12], [13], Tucker [14], and tensor train/ring [15], [16]. Among these models, Tucker decomposition is considered to be more general than CPD since it represents a tensor as a core tensor multiplied with factor matrices along different modes. CPD is a special case of Tucker decomposition when the core tensor is constrained to have a super-diagonal structure [17]. On the other hand, due to their more complicated structures, tensor train/ring [18]–[20] and nonlinear tensor models [21], [22] provide even more flexibility in data modeling. However, the learned results are oftentimes more difficult to interpret compared to Tucker or CPD.

Like any tensor decomposition, the most challenging problem in fitting data to the Tucker format is determining the multilinear rank [23], [24]. If the multilinear rank is known, various optimization techniques have been proposed [17], [25], [26] to solve the Tucker decomposition or completion problem. However, for real-world data, the multilinear rank of the Tucker decomposition is usually unknown. To fill this gap, Bayesian modelings of Tucker decomposition and completion have also been proposed in [27] [28] so that the multilinear rank is learned automatically together with the core tensor and the factor matrices. Unfortunately, in these two prior works [27] [28], only a single parameter is dedicated to the estimation of variance for the tensor core elements. This makes them restrictive in modeling real-world data, where the core tensor elements may possess diverse variances.

To introduce a more flexible core in the Bayesian Tucker model, this paper reveals an equivalent form of Tucker decomposition as a CPD with factor matrices having a large number of columns but being in a low-rank structure. With this newly found relationship, the Bayesian modeling deviates from the traditional sparsity inducing prior [27] [28] and becomes low-rank matrix modeling, in which Gaussian-inverse Wishart prior model is found suitable. For model learning, variational inference method is employed to derive an iterative update algorithm with each step having a closed-form expression. After inference, the multilinear rank of Tucker decomposition is revealed by performing singular value decomposition (SVD) on the learned factor matrices. The core tensor and factor matrices of Tucker decomposition can then be recovered accordingly. Simulation results on synthetic and real-world data demonstrate that the proposed equivlent Tucker model provides more accurate multilinear rank estimation and smaller tensor data recovery error compared to existing Bayesian and non-Bayesian Tucker completion methods.

The remainder of the paper is organized as follows. In Section II, a brief review on Tucker decomposition and existing Bayesian modeling are given. In Section III, an equivalent Tucker format and its Bayesian modeling are presented. In Section IV, an inference algorithm for the proposed model is derived under the variational inference framework. Simulation results and discussions are presented in Section V, and finally conclusions are drawn in Section VI.

Notation: Boldface lowercase and uppercase letters will be used for vectors and matrices, respectively. Tensors are written as calligraphic letters. $\mathbb{E}[\cdot]$ denotes the expectation of its argument. Superscript T denotes transpose. The operator $\mathrm{Tr}(\boldsymbol{A})$

Xueke Tong and Yik-Chung Wu are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: xktong@eee.hku.hk, ycwu@eee.hku.hk).

Lei Cheng is with College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, 310027, China (e-mail: lei_cheng@zju.edu.cn).

Lei Cheng and Yik-Chung Wu are the corresponding authors.

denotes the trace of a matrix $\boldsymbol{A}$. The symbol $\propto$ represents a linear scalar relationship between two real-valued functions. The operator $\otimes$ is the Kronecker product, $\odot$ is the Khatri–Rao product, $\circ$ is the outer product, and $*$ is the Hadamard product. The $N \times N$ diagonal matrix with diagonal components a_1 through a_N is represented as $\mathrm{Diag}\{a_1, a_2, ..., a_N\}$, while $\mathrm{diag}(\boldsymbol{A})$ takes the diagonal elements of $\boldsymbol{A}$ and put it as a vector. $\boldsymbol{I}_M$ represents the $M \times M$ identity matrix. The $(i,j)^{th}$ element, the i^{th} row and the j^{th} column of a matrix $\boldsymbol{A}$ are represented by $\boldsymbol{A}_{i,j}$, $\boldsymbol{A}_{i,:}$ and $\boldsymbol{A}_{:,j}$, respectively. The $(i,j,n)^{th}$ element, the n^{th} matrix atom of a third-order tensor $\mathcal{Y}$ are represented by $\mathcal{Y}_{i,j,n}$ and $\mathcal{Y}_{:,:,n}$, respectively.

II. Tucker Deomposition and Previous Bayesian Modeling

For a third-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ with its $(i_1, i_2, i_3)^{th}$ entry denoted by $\mathcal{X}_{i_1,i_2,i_3}$, the Tucker decomposition is defined as [17]

$$\begin{aligned}\mathcal{X} &= \mathcal{G} \times_1 \boldsymbol{A}^{(1)} \times_2 \boldsymbol{A}^{(2)} \times_3 \boldsymbol{A}^{(3)} \\ &\triangleq [\![\mathcal{G}; \boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)}]\!], \end{aligned} \tag{1}$$

where $\boldsymbol{A}^{(1)} \in \mathbb{R}^{I_1 \times R_1}$, $\boldsymbol{A}^{(2)} \in \mathbb{R}^{I_2 \times R_2}$ and $\boldsymbol{A}^{(3)} \in \mathbb{R}^{I_3 \times R_3}$ are the mode-1, mode-2 and mode-3 factor matrices, respectively; $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ is the core tensor; $\times_k$ denotes the tensor–matrix product along the k-mode [17]. The second line of (1) is a commonly used shorthand notation for Tucker decomposition.

The size of the Tucker core (R_1, R_2, R_3) denotes the dimension of the associated latent spaces. The (R_1, R_2, R_3) that leads to the minimum of $\sum_{k=1}^{3} R_k$ while making (1) hold are called the multilinear rank of tensor $\mathcal{X}$. On the other hand, the unfolding operation connects tensor decompositions to matrix operations. For example, if tensor $\mathcal{X}$ in (1) is unfolded along the first mode, the resulting matrix is

$$\boldsymbol{X}_{(1)} = \boldsymbol{A}^{(1)} \boldsymbol{G}_{(1)} (\boldsymbol{A}^{(3)} \otimes \boldsymbol{A}^{(2)})^T \in \mathbb{R}^{I_1 \times I_2 I_3}, \tag{2}$$

where $\boldsymbol{G}_{(1)}$ is the mode-1 unfolded matrix of the core tensor $\mathcal{G}$. Similar expressions can be obtained if $\mathcal{X}$ is unfolded along the second or third mode [17].

To automatically learn the multilinear rank from tensor data in a single run, pioneering work [27] proposed Bayesian Tucker decomposition modeling (abbreviated as BTD) that imposes sparsity-enhancing priors on the factor matrices $\{\boldsymbol{A}^{(k)}\}_{k=1}^{3}$. More specifically, BTD specifies that $\boldsymbol{A}^{(k)} \sim \prod_{i_k=1}^{I_k} \mathcal{N}(\boldsymbol{A}^{(k)}_{i_k,:} | \boldsymbol{0},\ (\mathrm{Diag}(\boldsymbol{\lambda}^{(k)}))^{-1})$, $\forall k$, with each element of $\boldsymbol{\lambda}^{(k)}$ following $\boldsymbol{\lambda}^{(k)}_{r_k} \sim \mathrm{Gamma}(a_k, b_k)$, $\forall k$. This Gaussian-gamma hierarchical construction induces sparsity in the columns of $\{\boldsymbol{A}^{(k)}\}_{k=1}^{3}$ [29], thus achieving automatic multilinear rank determination.

Furthermore, to ensure the number of columns of $\boldsymbol{A}^{(1)}$, $\boldsymbol{A}^{(2)}$, and $\boldsymbol{A}^{(3)}$ match to the dimensions of the core tensor, the core tensor $\mathcal{G}$ is modeled in a vector form $\mathrm{vec}(\mathcal{G}) \sim \mathcal{N}\Big(\mathrm{vec}(\mathcal{G}) | \boldsymbol{0},\ \big(\beta \mathrm{Diag}(\boldsymbol{\lambda}^{(3)}) \otimes \mathrm{Diag}(\boldsymbol{\lambda}^{(2)}) \otimes \mathrm{Diag}(\boldsymbol{\lambda}^{(1)})\big)^{-1}\Big)$, in which $\beta \sim \mathrm{Gamma}(a_0, b_0)$. Equivalently, the prior on the $(r_1, r_2, r_3)^{th}$ element of the core tensor $\mathcal{G}$ is $\mathcal{N}(\mathcal{G}_{r_1 r_2 r_3} | 0, (\beta \boldsymbol{\lambda}^{(3)}_{r_3} \boldsymbol{\lambda}^{(2)}_{r_2} \boldsymbol{\lambda}^{(1)}_{r_1})^{-1})$. Although this model allows some flexibility for the variances in the core tensor, $\{\boldsymbol{\lambda}^{(k)}\}_{k=1}^{3}$ are mainly determined by the precisions of the rows of $\{\boldsymbol{A}^{(k)}\}_{k=1}^{3}$, and therefore they offer limited capability in modeling the variation of the core elements. Effectively, this model only uses a single scale parameter β to adjust the variance of the core tensor elements. This obviously is not flexible enough to model all possible core tensors, and weakens the representation ability in practical applications. Together with Gaussian likelihood and Gamma prior distribution on the unknown noise precision, the graphical model of the above BTD [27] is shown in Figure 1(a).

On the other hand, another Bayesian variant (termed as ARD-Tucker [28]) models the hyper-parameters $\boldsymbol{\lambda}^{(k)}$ with non-informative Jeffreys priors instead of the Gamma priors. While this prior could also induce sparsity and thus determine the multilinear rank automatically, the same problem of inflexible core tensor as in [27] still occurs.

III. Bayeisan Model in Flexible Core Tucker Completion

Let $\mathcal{Y}_{\boldsymbol{\Omega}}$ be an incomplete third-order tensor of size $I_1 \times I_2 \times I_3$, where $\boldsymbol{\Omega}$ denotes a set of 3-tuple indices indicating which element is observed. Equivalently, we can define a binary tensor $\mathcal{O}$ with the same size as $\mathcal{Y}$ as an indicator of observed entries (if $\mathcal{Y}_{i_1 i_2 i_3}$ is observed $\mathcal{O}_{i_1 i_2 i_3} = 1$ and $\mathcal{O}_{i_1 i_2 i_3} = 0$ otherwise). We assume the complete data, denoted as $\mathcal{Y}$, is a noisy observation of true tensor $\mathcal{X}$, that is, $\mathcal{Y} = \mathcal{X} + \mathcal{E}$, where each element of the noise term follows i.i.d. Gaussian distribution, i.e., $\mathcal{E} \sim \prod_{i_1 i_2 i_3} \mathcal{N}(0, \alpha^{-1})$, and the tensor $\mathcal{X}$ obeys the Tucker decomposition model given in (1). The Tucker generative model, together with Gaussian noise assumption, give rise to the observation model

$$\begin{aligned} p\left(\mathcal{Y}_{\boldsymbol{\Omega}} | \mathcal{G}, \{\boldsymbol{A}^{(k)}\}_{k=1}^{3}, \alpha\right) = \prod_{i_1=1}^{I_1} \prod_{i_2=1}^{I_2} \prod_{i_3=1}^{I_3} \\ \mathcal{N}\left(\mathcal{Y}_{i_1 i_2 i_3} | [\![\mathcal{G}; \boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)}]\!]_{i_1 i_2 i_3}, \alpha^{-1}\right)^{\mathcal{O}_{i_1 i_2 i_3}}, \end{aligned} \tag{3}$$

where α denotes the noise precision.

In order to introduce flexibility to the core tensor in Bayesian modeling, we represent the core tensor $\mathcal{G}$ by a high rank CPD: $\mathcal{G} \triangleq [\![\boldsymbol{\Xi}^{(1)}, \boldsymbol{\Xi}^{(2)}, \boldsymbol{\Xi}^{(3)}]\!] = \sum_{l=1}^{L} \boldsymbol{\Xi}^{(1)}_{:,l} \circ \boldsymbol{\Xi}^{(2)}_{:,l} \circ \boldsymbol{\Xi}^{(3)}_{:,l}$ where $\boldsymbol{\Xi}^{(1)} \in \mathbb{R}^{R_1 \times L}$, $\boldsymbol{\Xi}^{(2)} \in \mathbb{R}^{R_2 \times L}$, $\boldsymbol{\Xi}^{(3)} \in \mathbb{R}^{R_3 \times L}$ are the factor matrices of the CPD. As any arbitrary third-order tensor could be represented by a CPD of finite rank [30], choosing L large enough would enable the modeling of any tensor core $\mathcal{G}$. With the above flexible core structure, the Tucker model in (3) is then represented as

$$\begin{aligned} p\left(\mathcal{Y}_{\boldsymbol{\Omega}} | \{\boldsymbol{\Xi}^{(k)}, \boldsymbol{A}^{(k)}\}_{k=1}^{3}, \alpha\right) = \prod_{i_1=1}^{I_1} \prod_{i_2=1}^{I_2} \prod_{i_3=1}^{I_3} \mathcal{N}\big(\mathcal{Y}_{i_1 i_2 i_3} | \\ [\![[\![\boldsymbol{\Xi}^{(1)}, \boldsymbol{\Xi}^{(2)}, \boldsymbol{\Xi}^{(3)}]\!]; \boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)}]\!]_{i_1 i_2 i_3}, \alpha^{-1}\big)^{\mathcal{O}_{i_1 i_2 i_3}}. \end{aligned} \tag{4}$$

If we extend the Bayesian modeling in [27] to (4), we would impose a Gaussian prior on the columns of $\boldsymbol{\Xi}^{(k)}$ and tie its covariance to $\mathrm{diag}(\boldsymbol{\lambda}^{(k)})$, which results in the probabilistic model shown in Figure 1(b). However, this model is complicated as there are six matrices to be estimated with

coupling between $\boldsymbol{A}^{(k)}$ and $\boldsymbol{\Xi}^{(k)}$. This makes the inference very challenging. In the following, we reveal an equivalent form of (4) that allows a simplier Bayesian modeling and inference.

Noticing that unfolding $\mathcal{G} = [\![\boldsymbol{\Xi}^{(1)}, \boldsymbol{\Xi}^{(2)}, \boldsymbol{\Xi}^{(3)}]\!]$ along the first mode gives $\boldsymbol{G}_{(1)} = \boldsymbol{\Xi}^{(1)}(\boldsymbol{\Xi}^{(3)} \odot \boldsymbol{\Xi}^{(2)})^T$ [17] and making use of (2), we have

$$\begin{aligned}
&[\![[\![\boldsymbol{\Xi}^{(1)}, \boldsymbol{\Xi}^{(2)}, \boldsymbol{\Xi}^{(3)}]\!]; \boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)}]\!]_{(1)} \\
=&\boldsymbol{A}^{(1)}\boldsymbol{\Xi}^{(1)}(\boldsymbol{\Xi}^{(3)} \odot \boldsymbol{\Xi}^{(2)})^T(\boldsymbol{A}^{(3)} \otimes \boldsymbol{A}^{(2)})^T \\
=&\boldsymbol{A}^{(1)}\boldsymbol{\Xi}^{(1)}(\boldsymbol{A}^{(3)}\boldsymbol{\Xi}^{(3)} \odot \boldsymbol{A}^{(2)}\boldsymbol{\Xi}^{(2)})^T \\
=&[\![\boldsymbol{A}^{(1)}\boldsymbol{\Xi}^{(1)}, \boldsymbol{A}^{(2)}\boldsymbol{\Xi}^{(2)}, \boldsymbol{A}^{(3)}\boldsymbol{\Xi}^{(3)}]\!]_{(1)}, \qquad (5)
\end{aligned}$$

where the second to the third lines are due to a known property of Khatri–Rao product [17], [31], [32]. Based on (5), it is clear that tensor $[\![[\![\boldsymbol{\Xi}^{(1)}, \boldsymbol{\Xi}^{(2)}, \boldsymbol{\Xi}^{(3)}]\!]; \boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)}]\!]$ equals $[\![\boldsymbol{A}^{(1)}\boldsymbol{\Xi}^{(1)}, \boldsymbol{A}^{(2)}\boldsymbol{\Xi}^{(2)}, \boldsymbol{A}^{(3)}\boldsymbol{\Xi}^{(3)}]\!]$, as they have the same size and their mode-1 unfoldings are the same.

For notational simplicity, if we define $\boldsymbol{B}^{(k)} = \boldsymbol{A}^{(k)}\boldsymbol{\Xi}^{(k)}$ for $k = 1,2,3$, (4) can be rewritten as

$$\begin{aligned}
&p\left(\mathcal{Y}_{\boldsymbol{\Omega}}|\{\boldsymbol{B}^{(k)}\}_{k=1}^3, \alpha\right) = \prod_{i_1=1}^{I_1}\prod_{i_2=1}^{I_2}\prod_{i_3=1}^{I_3} \\
&\mathcal{N}\left(\mathcal{Y}_{i_1i_2i_3}|[\![\boldsymbol{B}^{(1)}, \boldsymbol{B}^{(2)}, \boldsymbol{B}^{(3)}]\!]_{i_1i_2i_3}, \alpha^{-1}\right)^{\mathcal{O}_{i_1i_2i_3}}. \qquad (6)
\end{aligned}$$

As $\boldsymbol{B}^{(1)}$, $\boldsymbol{B}^{(2)}$, and $\boldsymbol{B}^{(3)}$ are constructed from multiplication of two matrices, and R_1, R_2, and R_3 are smaller than I_1, I_2, I_3 and L, this result reveals that Tucker decomposition $[\![\mathcal{G}; \boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \boldsymbol{A}^{(3)}]\!]$ is actually equivalent to a CPD $[\![\boldsymbol{B}^{(1)}, \boldsymbol{B}^{(2)}, \boldsymbol{B}^{(3)}]\!]$ if the factor matrices $\boldsymbol{B}^{(1)}$, $\boldsymbol{B}^{(2)}$, $\boldsymbol{B}^{(3)}$ have a large number of columns while being low-rank.

Before we present the Bayesian modeling for learning $\boldsymbol{B}^{(1)}$, $\boldsymbol{B}^{(2)}$, $\boldsymbol{B}^{(3)}$ with low-rank structure, let us see how we can recover the Tucker structure if we have $\boldsymbol{B}^{(1)}$, $\boldsymbol{B}^{(2)}$, $\boldsymbol{B}^{(3)}$. Since $\boldsymbol{B}^{(1)}$, $\boldsymbol{B}^{(2)}$, $\boldsymbol{B}^{(3)}$ are of low-rank, they can be decomposed using SVD: $\boldsymbol{B}^{(k)} = \boldsymbol{U}^{(k)}\boldsymbol{D}^{(k)}\boldsymbol{V}^{(k)^T}$ for $k = 1,2,3$. Then

$$\begin{aligned}
&[\![\boldsymbol{B}^{(1)}, \boldsymbol{B}^{(2)}, \boldsymbol{B}^{(3)}]\!] \\
=&[\![\boldsymbol{U}^{(1)}\boldsymbol{D}^{(1)}\boldsymbol{V}^{(1)^T}, \boldsymbol{U}^{(2)}\boldsymbol{D}^{(2)}\boldsymbol{V}^{(2)^T}, \boldsymbol{U}^{(3)}\boldsymbol{D}^{(3)}\boldsymbol{V}^{(3)^T}]\!] \\
=&[\![\boldsymbol{D}^{(1)}\boldsymbol{V}^{(1)^T}, \boldsymbol{D}^{(2)}\boldsymbol{V}^{(2)^T}, \boldsymbol{D}^{(3)}\boldsymbol{V}^{(3)^T}]\!] \\
&\qquad \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times_3 \boldsymbol{U}^{(3)}. \qquad (7)
\end{aligned}$$

From the last line of (7), we obtain that $[\![\boldsymbol{D}^{(1)}\boldsymbol{V}^{(1)^T}, \boldsymbol{D}^{(2)}\boldsymbol{V}^{(2)^T}, \boldsymbol{D}^{(3)}\boldsymbol{V}^{(3)^T}]\!]$ is the recovered core tensor, and $\boldsymbol{U}^{(1)}$, $\boldsymbol{U}^{(2)}$ and $\boldsymbol{U}^{(3)}$ are the Tucker factor matrices. Furthermore, the number of non-zero elements in $\boldsymbol{D}^{(1)}$, $\boldsymbol{D}^{(2)}$ and $\boldsymbol{D}^{(3)}$ give the estimates of R_1, R_2 and R_3, respectively.

Eq. (7) in fact provides another view on why the proposed equivalent Tucker model is more flexible than BTD. More specifically, for the $(r_1, r_2, r_3)^{th}$ element of the recovered core tensor in (7), it can be expressed as $\boldsymbol{D}^{(1)}_{r_1}\boldsymbol{D}^{(2)}_{r_2}\boldsymbol{D}^{(3)}_{r_3}\sum_{l=1}^L \boldsymbol{V}^{(1)}_{l,r_1}\boldsymbol{V}^{(2)}_{l,r_2}\boldsymbol{V}^{(3)}_{l,r_3}$. This shows that besides the $\boldsymbol{D}^{(1)}_{r_1}\boldsymbol{D}^{(2)}_{r_2}\boldsymbol{D}^{(3)}_{r_3}$ (which has a similar effect to $\boldsymbol{\lambda}^{(1)}_{r_1}$ $\boldsymbol{\lambda}^{(2)}_{r_2}$ $\boldsymbol{\lambda}^{(3)}_{r_3}$ in BTD [27]), each element is additionally characterized by $\sum_{l=1}^L \boldsymbol{V}^{(1)}_{l,r_1}\boldsymbol{V}^{(2)}_{l,r_2}\boldsymbol{V}^{(3)}_{l,r_3}$, which provides more learnable parameters and thus flexibility for adapting to different tensor cores than using a single parameter β in the BTD.

With the established equivalent model in (6), the next question is how can we model the prior of $\boldsymbol{B}^{(k)}$ such that low-rankness information is imposed. Fortunately, Gaussian-inverse Wishart prior [33] can serve such a purpose. In particular, we model $\boldsymbol{B}^{(k)} \sim \prod_{l=1}^L \mathcal{N}(\boldsymbol{B}^{(k)}_{:,l}|\boldsymbol{0}, \boldsymbol{\Sigma}_k^{-1})$, $\boldsymbol{\Sigma}_k \sim$ Wishart$(\boldsymbol{\Sigma}_k|v_k, \boldsymbol{\Psi}_k)$, where $v_k \in \mathbb{R}$ and $\boldsymbol{\Psi}_k \in \mathbb{R}^{I_k \times I_k}$ (being symmetric and positive definite) are fixed parameters. The graphical model for the Bayesian Tucker using this low-rank inducing prior is shown in Figure 1(c).

Notice that all the columns of $\boldsymbol{B}^{(k)}$ are controlled by the same covariance matrix $\boldsymbol{\Sigma}_k^{-1}$ because we desire the low-rankness of the whole matrix $\boldsymbol{B}^{(k)}$. To see why the Gaussian-inverse Wishart prior described above is a low-rank promoting prior, we can compute the marginal distribution of $\boldsymbol{B}^{(k)}$, which is [33]

$$\begin{aligned}
p(\boldsymbol{B}^{(k)}) &= \int \prod_{l=1}^L \mathcal{N}(\boldsymbol{B}^{(k)}_{:,l}|\boldsymbol{0}, \boldsymbol{\Sigma}_k^{-1})\text{Wishart}(\boldsymbol{\Sigma}_k|v_k, \boldsymbol{\Psi}_k)d\boldsymbol{\Sigma}_k \\
&\propto |\boldsymbol{\Psi}_k^{-1} + \boldsymbol{B}^{(k)}\boldsymbol{B}^{(k)^T}|^{-\frac{v_k+L}{2}}. \qquad (8)
\end{aligned}$$

Then the log-marginal distribution of $\boldsymbol{B}^{(k)}$ becomes

$$\begin{aligned}
\log p(\boldsymbol{B}^{(k)}) &\propto -\log|\boldsymbol{B}^{(k)}\boldsymbol{B}^{(k)^T} + \boldsymbol{\Psi}_k^{-1}| \\
&\propto -\log|\boldsymbol{I} + \boldsymbol{B}^{(k)^T}\boldsymbol{\Psi}_k\boldsymbol{B}^{(k)}| \\
&= -\sum_{l=1}^L \log(\lambda_l + 1), \qquad (9)
\end{aligned}$$

where λ_l is the l^{th} eigenvalue of the matrix $\boldsymbol{B}^{(k)^T}\boldsymbol{\Psi}_k\boldsymbol{B}^{(k)}$. To comply with the prior, (9) would be made as large as possible, which is equivalent to $\sum_{l=1}^L \log(\lambda_l + 1)$ as small as possible. This leads to two consequences. 1) This induces sparsity in λ_l [34]–[36], making $\boldsymbol{B}^{(k)^T}\boldsymbol{\Psi}_k\boldsymbol{B}^{(k)}$ low-rank. Since $\boldsymbol{\Psi}_k$ is of full rank, the low-rankness of $\boldsymbol{B}^{(k)^T}\boldsymbol{\Psi}_k\boldsymbol{B}^{(k)}$ translates into the low-rankness of $\boldsymbol{B}^{(k)}$. 2) $\text{Tr}\big(\boldsymbol{B}^{(k)^T}\boldsymbol{\Psi}_k\boldsymbol{B}^{(k)}\big)$ is the first-order approximation of $\log|\boldsymbol{I} + \boldsymbol{B}^{(k)^T}\boldsymbol{\Psi}_k\boldsymbol{B}^{(k)}|$, and minimizing $\log|\boldsymbol{I} + \boldsymbol{B}^{(k)^T}\boldsymbol{\Psi}_k\boldsymbol{B}^{(k)}|$ also minimizes $\text{Tr}\big(\boldsymbol{B}^{(k)^T}\boldsymbol{\Psi}_k\boldsymbol{B}^{(k)}\big)$, which can be interpreted as introducing manifold smoothness (defined by $\boldsymbol{\Psi}_k$) to the rows of $\boldsymbol{B}^{(k)}$ [37]–[39].

A common choice of $\boldsymbol{\Psi}_k$ in image or MRI tensor data is $\boldsymbol{\Psi}_k = \boldsymbol{F}^T\boldsymbol{F}$ [40], where $\boldsymbol{F}$ is the second-order difference operator with its $(i,j)^{th}$ element given by

$$F_{i,j} = \begin{cases} -2, & i = j = 0 \\ 1, & |i-j| = 1, \\ 0, & \text{else} \end{cases} \qquad (10)$$

which basically states that correlations among neighboring pixels exist. Another chiose of $\boldsymbol{\Psi}_k$ to promote a smooth solution is the Laplacian matrix, where the details can be found in [41], [42]. Obviously, if $\boldsymbol{\Psi}_k \propto \boldsymbol{I}$, then we would have prior on $\boldsymbol{B}^{(k)}$ induces only low-rankness but not smoothness.

Finally, with the prior of $\boldsymbol{B}^{(k)}$ established, the probabilistic model is completed by specifying the likelihood function of

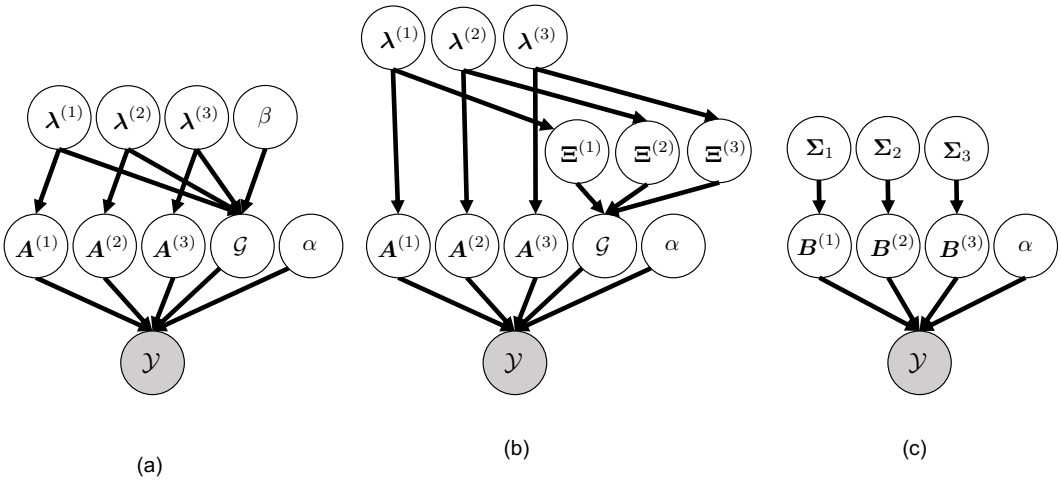

Figure 1: Bayesian model for Tucker decomposition. a) Bayesian tensor Tucker decompostion [27]; b) Straightforwardly extend the modeling of (a) to tensor Tucker decompositon with a flexible core; c) The proposed Bayesian model.

the observations, which is Gaussian with unknown precision $\alpha \sim \text{Gamma}(c, d)$. Let $\boldsymbol{\Theta}$ collects all the unknown variables, i.e., $\boldsymbol{\Theta} = \{\boldsymbol{B}^{(1)}, \boldsymbol{B}^{(2)}, \boldsymbol{B}^{(3)}, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \boldsymbol{\Sigma}_3, \alpha\}$, the joint distribution of $\mathcal{Y}_{\boldsymbol{\Omega}}$ and $\boldsymbol{\Theta}$ is given by

$$
\begin{aligned}
& p\left(\mathcal{Y}_{\boldsymbol{\Omega}}, \boldsymbol{\Theta}\right) \\
&= p\Big(\mathcal{Y}_{\boldsymbol{\Omega}} | \boldsymbol{B}^{(1)}, \boldsymbol{B}^{(2)}, \boldsymbol{B}^{(3)}, \alpha\Big) p\left(\boldsymbol{B}^{(1)} | \boldsymbol{\Sigma}_1\right) p\left(\boldsymbol{\Sigma}_1\right) \\
&\qquad p\left(\boldsymbol{B}^{(2)} | \boldsymbol{\Sigma}_2\right) p\left(\boldsymbol{\Sigma}_2\right) p\left(\boldsymbol{B}^{(3)} | \boldsymbol{\Sigma}_3\right) p\left(\boldsymbol{\Sigma}_3\right) p\left(\alpha\right) \\
&\propto \exp\Bigg\{ \frac{\sum_{i_1 i_2 i_3} \mathcal{O}_{i_1 i_2 i_3}}{2} \ln \alpha \\
&\qquad - \frac{\alpha}{2} \left\| \mathcal{O} * \left(\mathcal{Y} - \sum_{l=1}^{L} \left(\boldsymbol{B}^{(1)}_{:,l} \circ \boldsymbol{B}^{(2)}_{:,l} \circ \boldsymbol{B}^{(3)}_{:,l} \right) \right) \right\|_F^2 \\
&\qquad + \sum_{k=1,2,3} \Bigg(\frac{L}{2} \ln |\boldsymbol{\Sigma}_k| - \frac{1}{2} \sum_{l=1}^{L} \boldsymbol{B}^{(k)T}_{:,l} \boldsymbol{\Sigma}_k \boldsymbol{B}^{(k)}_{:,l} \\
&\qquad + \frac{v_k - I_k - 1}{2} \ln |\boldsymbol{\Sigma}_k| - \frac{1}{2} \text{Tr}\left(\boldsymbol{\Psi}_k^{-1} \boldsymbol{\Sigma}_k \right) \Bigg) \\
&\qquad + (c-1) \ln \alpha - d\alpha \Bigg\}. \qquad (11)
\end{aligned}
$$

We term the proposed model as Bayesian Flexible Core Tucker Completion via a hierarchical Wishart Prior (BFCTC-W).

Remark 1: Notice that the proposed format in this paper is a new representation of Tucker, rather than a new decomposition model. We need this new representation of Tucker because in the conventional Tucker Bayesian model, the number of parameters for estimating the variation of the Tucker core is small, thus offering limited flexibility to data modeling. With the new representation of Tucker, the core tensor can be represented more flexibly and accurately. One may suggest directly adding correlated priors in the core tensor of the conventional Bayesian Tucker model to enhance the core flexibility. However, as the multilinear rank is unknown, the dimension of the Tucker core is not fixed in prior. This brings a challenge on specifying the correlation priors of the Tucker core. In contrast, the proposed representation enables both flexible core and multilinear rank estimation.

Remark 2: Expressing the Tucker core tensor using CP format has also been considered in [45] for developing an approximate CPD based on the relationship between CPD and Tucker format. However, it requires at least two factor matrices of the CP representation of the core to be of full rank. In contrast, the model in this paper does not require this. In fact, we purposely makes the factor matrices of CP representation not to be full rank, so that rank estimation can be incorporated in the inference procedure.

Remark 3: Recently, a Bayesian model for a special case of block-term decomposition has been developed in [44]. In general, block-term decomposition can be interpreted as a sum of several Tucker decompositions [43]. Given that the proposed Tucker model in this paper could estimate the multilinear rank R_1, R_2, R_3, one may consider extending the model to general block-term decomposition by including an additional sparse latent variable multiplied to the covariance of $\boldsymbol{B}^{(k)}$ to indicate the presence or absence of a Tucker component. But this extension is beyond the scope of this work and thus is left for future study.

IV. Variational Inference algorithm

In Bayesian framework, the inference of $\boldsymbol{\Theta}$ is based on the posterior distribution $p(\boldsymbol{\Theta}|\mathcal{Y}_{\boldsymbol{\Omega}}) = p(\mathcal{Y}_{\boldsymbol{\Omega}}, \boldsymbol{\Theta}) / \int p(\boldsymbol{\Theta}, \mathcal{Y}_{\boldsymbol{\Omega}}) d\boldsymbol{\Theta}$. However, the multiple integrations involved are generally intractable. To tackle this, variational inference [46], [47] is a major tool for inferring parameters of a complicated probabilistic model. Although each of the prior of $\boldsymbol{B}^{(k)}$ follows Gaussian-inverse Wishart distribution, and is the same as in conventional Bayesian matrix completion problem [33], the joint distribution for the newly proposed equivalent Tucker model in (11) is more complicated than that in Bayesian matrix completion. In particular, there are three factor matrices to be estimated in Tucker model. Furthermore, the three factor matrices are coupled in a nonlinear way in (11). This makes

the inference algorithm very different from the simple matrix completion problem.

A. Derivation of the algorithm

The key idea in variational inference is to approximate the true posterior distribution $p(\boldsymbol{\Theta}|\mathcal{Y}_{\boldsymbol{\Omega}})$ by a variational distribution $Q(\boldsymbol{\Theta})$ that minimizes the Kullback-Leibler (KL) divergence:

$$\begin{aligned} &KL\left(Q\left(\boldsymbol{\Theta}\right)\right)||p\left(\boldsymbol{\Theta}|\mathcal{Y}_{\boldsymbol{\Omega}}\right) \\ &= -\int Q\left(\boldsymbol{\Theta}\right)\ln\left\{\frac{p\left(\boldsymbol{\Theta}|\mathcal{Y}_{\boldsymbol{\Omega}}\right)}{Q\left(\boldsymbol{\Theta}\right)}\right\}d\boldsymbol{\Theta}. \end{aligned} \tag{12}$$

To facilitate the KL divergence minimization, the variational distribution $Q(\boldsymbol{\Theta})$ is usually restricted to the mean-field family $Q(\boldsymbol{\Theta}) = \prod_m Q(\boldsymbol{\Theta}_m)$, where $\boldsymbol{\Theta}_m \in \boldsymbol{\Theta}$ with $\cup_{m=1}^{M}\boldsymbol{\Theta}_m = \boldsymbol{\Theta}$ and $\cap_{m=1}^{M}\boldsymbol{\Theta}_m = \emptyset$. Under the mean-field assumption, each optimal variational distribution $Q^*(\boldsymbol{\Theta}_m)$ that minimizes the KL divergence is obtained by computing [46]

$$Q^*\left(\boldsymbol{\Theta}_m\right) = \frac{\exp\left(\mathbb{E}_{\boldsymbol{\Theta}\backslash\boldsymbol{\Theta}_m}\left[\ln p\left(\boldsymbol{\Theta},\mathcal{Y}_{\boldsymbol{\Omega}}\right)\right]\right)}{\int\exp\left(\mathbb{E}_{\boldsymbol{\Theta}\backslash\boldsymbol{\Theta}_m}\left[\ln p\left(\boldsymbol{\Theta},\mathcal{Y}_{\boldsymbol{\Omega}}\right)\right]\right)d\boldsymbol{\Theta}_m}, \tag{13}$$

where $\boldsymbol{\Theta}\backslash\boldsymbol{\Theta}_m$ is the variables set in $\boldsymbol{\Theta}$ but excluding $\boldsymbol{\Theta}_m$.

Obviously, the variational distributions in (13) are coupled in the sense that the computation of the variational distribution of one parameter, e.g., $\boldsymbol{\Theta}_m$, requires the knowledge of variational distributions of other parameters $\boldsymbol{\Theta}\backslash\boldsymbol{\Theta}_m$. Therefore, these variational distributions are updated iteratively. In the following, an explicit expression for each $Q(\cdot)$ is derived under the mean-field $Q(\boldsymbol{\Theta}) = Q(\alpha)\prod_{k,l}Q(\boldsymbol{B}_{:,l}^{(k)})\prod_k Q(\boldsymbol{\Sigma}_k)$. To make the presentation concise, only the results are given below while the derivations are detailed in Appendix A.

Update of $Q(\boldsymbol{B}_{:,l}^{(k)})$:

$$Q\left(\boldsymbol{B}_{:,l}^{(k)}\right) \propto \mathcal{N}\left(\boldsymbol{m}_l^{(k)}, {\boldsymbol{\Upsilon}_l^{(k)}}^{-1}\right), \tag{14}$$

where

$$\begin{aligned} {\boldsymbol{\Upsilon}_l^{(k)}}^{-1} &= \Bigg(\mathbb{E}\left[\boldsymbol{\Sigma}_k\right] + \mathbb{E}\left[\alpha\right]\cdot\mathrm{Diag}\Big(\boldsymbol{O}_{(k)}\cdot \\ &\qquad \bigodot_{h\neq k}\left[\boldsymbol{m}_l^{(h)} * \boldsymbol{m}_l^{(h)} + \mathrm{diag}({\boldsymbol{\Upsilon}_l^{(h)}}^{-1})\right]\Big)\Bigg)^{-1} \\ \boldsymbol{m}_l^{(k)} &= -{\boldsymbol{\Upsilon}_l^{(k)}}^{-1}\mathbb{E}\left[\alpha\right]\Bigg\{ \\ &\quad \sum_{p\neq l}\mathrm{diag}\Big(\boldsymbol{O}_{(k)}\bigodot\nolimits_{h\neq k}(\boldsymbol{m}_l^{(h)} * \boldsymbol{m}_p^{(h)})\cdot{\boldsymbol{m}_p^{(k)}}^T\Big) \\ &\quad - \Big[\left(\boldsymbol{Y}_{(k)} * \boldsymbol{O}_{(k)}\right)\cdot\Big(\bigodot_{h\neq k}\boldsymbol{m}_l^{(h)}\Big)\Big]\Bigg\}, \end{aligned} \tag{15}$$

where diag(.) takes the diagonal of the matrix and put it as a vector, and $\boldsymbol{O}_{(k)}$ is the mode-k unfolding of tensor $\mathcal{O}$.

Update of $Q(\boldsymbol{\Sigma}_k)$:

$$Q(\boldsymbol{\Sigma}_k) \propto \mathrm{Wishart}(\hat{v}_k, \hat{\boldsymbol{\Psi}}_k), \tag{16}$$

where $\hat{v}_k$ and $\hat{\boldsymbol{\Psi}}_k$ are given by

$$\hat{v}_k = v_k + L,$$

$$\hat{\boldsymbol{\Psi}}_k = \left(\boldsymbol{\Psi}_k^{-1} + \boldsymbol{M}^{(k)}\left(\boldsymbol{M}^{(k)}\right)^T + \sum_{l=1}^{L}{\boldsymbol{\Upsilon}_l^{(k)}}^{-1}\right)^{-1}, \tag{17}$$

with $\boldsymbol{M}^{(k)} = \left[\boldsymbol{m}_1^{(k)}, \boldsymbol{m}_2^{(k)}, ..., \boldsymbol{m}_L^{(k)}\right]$.

Update of $Q(\alpha)$:

$$Q\left(\alpha\right) \propto \mathrm{Gamma}\left(\hat{c}, \hat{d}\right), \tag{18}$$

where

$$\begin{aligned} \hat{c} &= c + \frac{\sum_{i_1i_2i_3}\mathcal{O}_{i_1i_2i_3}}{2}, \\ \hat{d} &= d + \frac{1}{2}\mathrm{Tr}\Big\{(\boldsymbol{Y}_{(1)} * \boldsymbol{O}_{(1)})(\boldsymbol{O}_{(1)} * \boldsymbol{Y}_{(1)})^T\Big\} \\ &\quad - \mathrm{Tr}\Big\{(\boldsymbol{Y}_{(1)} * \boldsymbol{O}_{(1)})\Big((\boldsymbol{M}^{(3)}\odot\boldsymbol{M}^{(2)})(\boldsymbol{M}^{(1)})^T\Big)\Big\} \\ &\quad + \frac{1}{2}\Big\{\sum_{l=1}^{L}\sum_{p\neq l}^{L}\Big(\boldsymbol{m}_l^{(1)} * \boldsymbol{m}_p^{(1)}\Big)^T\boldsymbol{O}_{(1)}\bigodot_{h\neq 1}\Big(\boldsymbol{m}_l^{(h)} * \boldsymbol{m}_p^{(h)}\Big) \\ &\quad + \sum_{l=1}^{L}\Big(\boldsymbol{m}_l^{(1)} * \boldsymbol{m}_l^{(1)} + \mathrm{diag}({\boldsymbol{\Upsilon}_l^{(1)}}^{-1})\Big)^T \\ &\qquad \cdot\boldsymbol{O}_{(1)}\bigodot_{h\neq 1}\Big(\boldsymbol{m}_l^{(h)} * \boldsymbol{m}_l^{(h)} + \mathrm{diag}({\boldsymbol{\Upsilon}_l^{(h)}}^{-1})\Big)\Big\}. \end{aligned} \tag{19}$$

To compute various updates, we need a number of expectations on different variables. For example, in the covariance of $Q(\boldsymbol{B}_{:,l}^{(k)})$, we need $\mathbb{E}\left[\boldsymbol{\Sigma}_k\right]$ which can be computed from (17) as $\hat{v}_k\hat{\boldsymbol{\Psi}}_k$. For $\mathbb{E}\left[\alpha\right]$, it can be computed using (18) as $\mathbb{E}\left[\alpha\right] = \hat{c}/\hat{d}$.

B. Summary of the algorithm and properties

The procedure of BFCTC-W is summarized in Algorithm 1. As $Q(\boldsymbol{B}^{(k)})$ are Gaussian disttibuted with mean $\boldsymbol{M}^{(k)} = \left[\boldsymbol{m}_1^{(k)}, ..., \boldsymbol{m}_L^{(k)}\right]$, after the algorithm converges, we take $\boldsymbol{M}^{(k)}$ as an estimate of $\boldsymbol{B}^{(k)}$. To provide non-informative hyper-priors, we set the top level hyper-parameters $c = d = 10^{-6}$. Also, we choose $v_1 = v_2 = v_3 = 10$ [33]. For the parameters under inference, $\boldsymbol{M}^{(k)} = \mathbb{E}\left[\boldsymbol{B}^{(k)}\right]$ is initialized as $\boldsymbol{S}_0\boldsymbol{\Delta}_0^{\frac{1}{2}}$ [2], where $\boldsymbol{S}_0$ denotes the left singular vector matrix and $\boldsymbol{\Delta}_0$ denotes the diagonal singular values matrix from SVD of $\boldsymbol{Y}_{(k)}$. $\left\{\boldsymbol{\Upsilon}_l^{(k)}\right\}_{l=1}^{L}$ are initialized as $\left\{\boldsymbol{\Upsilon}_l^{(k)} = \boldsymbol{I}\right\}_{l=1}^{L}$. The initial covariance matrix $\boldsymbol{\Sigma}_k$ is simply set to $\boldsymbol{I}$ and α is initialized by $\alpha = c/d$. Based on the description in Section III, L should be a large value so we take $L = 150$. Finally, the algorithm is stopped when the normalized mean square error between the estimated tensors of two adjacent iterations is smaller than 10^{-6}.

We should notice that the parameter L is set to be a fixed value, and will not be updated in the process of inference. This parameter is related to how flexible the core tensor is, and has nothing to do with the multilinear rank of the Tucker model. On the other hand, with the low-rank promoting prior on $\boldsymbol{B}^{(k)}$, the multilinear rank is revealed by the number of non-zero singular values (or singular values above certain threshold if the data is noisy) from $\boldsymbol{D}^{(1)}$, $\boldsymbol{D}^{(2)}$, $\boldsymbol{D}^{(3)}$ in SVD

Algorithm 1: Bayesian Flexible Core Tucker Completion via a hierarchical Wishart Prior (BFCTC-W)

Input: a third-order noisy tensor $\mathcal{Y}$

Initialization: $\boldsymbol{M}^{(k)} = [\boldsymbol{m}_1^{(k)}, ..., \boldsymbol{m}_L^{(k)}]$, $\left\{\boldsymbol{\Phi}_i^{(k)}\right\}_{i=1}^{I_1}$, $\boldsymbol{\Psi}_k$, v_k, $\boldsymbol{\Sigma}_k = \boldsymbol{I}$, $\forall k \in [1,3]$, $\alpha = c/d$, L

repeat:

- Update $\{Q(\boldsymbol{B}_{:,l}^{(k)})\}_{l=1}^{L}$, $k = 1,2,3$ using (14) and (15);
- Update $Q(\boldsymbol{\Sigma}_k)$, $k = 1,2,3$ using (16) and (17);
- Update $Q(\alpha)$ using (18) and (19);

until convergence

Compute SVD of $\boldsymbol{M}^{(k)} = \boldsymbol{U}^{(k)}\boldsymbol{D}^{(k)}\boldsymbol{V}^{(k)^T}$. The core tensor of the Tucker decomposition is $[\![\boldsymbol{D}^{(1)}\boldsymbol{V}^{(1)^T}, \boldsymbol{D}^{(2)}\boldsymbol{V}^{(2)^T}, \boldsymbol{D}^{(3)}\boldsymbol{V}^{(3)^T}]\!]$ and the factor matrices are $\boldsymbol{U}^{(k)}$.

of $\boldsymbol{M}^{(1)}$, $\boldsymbol{M}^{(2)}$, $\boldsymbol{M}^{(3)}$. In this way, the multilinear rank could be automatically learned without updating L.

In the proposed algorithm, the computational complexity of the factor matrices $\{\boldsymbol{B}^{(k)}\}_{k=1}^{3}$, the hyperparameters $\{\boldsymbol{\Sigma}^{(k)}\}_{k=1}^{3}$, and the noise precision α are $O\big(L\cdot\sum_{k=1}^{3} {I_k}^3 + 3|\boldsymbol{\Omega}|(L^2+L)\big)$, $O\big(\sum_{k=1}^{3} {I_k}^3\big)$ and $O\big(|\boldsymbol{\Omega}|(L^2+L)\big)$, respectively, where $|\boldsymbol{\Omega}|$ denotes the number of observations. Thus the overall computational complexity is $O\big(L\sum_{k=1}^{3} {I_k}^3 + 4L^2|\boldsymbol{\Omega}|\big)$. On the other hand, the computational complexity of Bayesian Tucker Completion (abbreviated as BTC [27], [48], [49], partially observed version of BTD) is $O\big(\sum_{k=1}^{3} I_k R_k^3 + |\boldsymbol{\Omega}|\prod_{n=1}^{3} R_n \sum_{k=1}^{3} I_k\big)$ [27]. Based on the above complexity analysis, it can be seen that under high multilinear rank scenarios, which are likely in real-world data, the proposed BFCTC-W has a lower complexity order than BTC. The numerical experiments in the next section also confirm that the proposed algorithm run faster than BTC.

When the tensor $\mathcal{Y}$ is fully observed, we could directly set $\mathcal{O}$ as an all-one tensor in Algorithm 1. However, directly executing Algorithm 1 in fully observed data case is not the most efficient way, as many expressions in Algorithm 1 can be simplified when $\mathcal{O}$ is an all-one tensor. It turns out that the complexity of the proposed algorithm under fully observed data is $O\big(\sum_{k=1}^{3} {I_k}^3 + \prod_{k=1}^{3} I_k\big)$. The details of the simplification and the corresponding complexity analysis is given in Appendix B.

V. Experimental Results

We evaluate the proposed algorithm BFCTC-W by extensive experiments and compare it with state-of-the-art methods including HOOI [17], [26], ARD-Tucker [28], W-Tucker [50], CTNM [51], HaLRTC [52] and Bayesian Tucker completion (BTC) [27], which is an extension of BTD to incomplete data. Since this paper focuses on a new interpretation of the Tucker decomposition and its consequences in modeling and inference, the compared algorithms are all Tucker related. In all the experiments, the multilinear rank parameters of HOOI, W-Tucker and HaLRTC are set as the true rank (if ground true is available), or we follow the default settings suggested in these works (if ground truth is not available). For the competing algorithms, ARD-Tucker, CTNM and BTC can automatically learn the multilinear rank. For the proposed BFCTC-W, multilinear rank is determined by retaining singular values in $\boldsymbol{D}^{(k)}$ if its squared value is larger than 10^{-4} times of the squared value of the largest singular values. For all the experiments, the locations of the missing data are assumed to be uniformly distributed in the tensor data. The experiments are carried out in MATLAB R2020a on a macOS with 2.2 GHz Inter Core i7 CPU and 16 GB RAM.

A. Synthetic data

We generate two kinds of synthetic data. The first one is when the elements of the core tensor are i.i.d. and drawn from $\mathcal{G}_{r_1,r_2,r_3} \sim \mathcal{N}(0,1)$, $\forall r_k = 1,...,R_k$, $k = 1,2,3$. The second kind of synthetic data is with $\mathcal{G}_{r_1,r_2,r_3}$ obeying Gaussian distribution with zero mean and standard deviation $\prod_{k=1}^{3}\boldsymbol{\zeta}_{r_k}^{(k)}$ where $\boldsymbol{\zeta}_{r_k}^{(k)}$ is independent and uniformly drawn from $\{1,2,...,R_k\}$, $\forall r_k = 1,...,R_k$ and $k = 1,2,3$. This case represents the scenario where the elements of the Tucker core have diverse variances. For both synthetic data, all elements of factor matrices $\boldsymbol{A}^{(1)}$, $\boldsymbol{A}^{(2)}$ and $\boldsymbol{A}^{(3)}$ are drawn from $\mathcal{N}(0,1)$, and then orthogonalization among columns was performed such that $\boldsymbol{A}^{(k)^T}\boldsymbol{A}^{(k)} = \boldsymbol{I}_{R_k}$. As the synthetic data does not contain smoothness among neighboring values, we choose $\boldsymbol{\Psi}_1 = \boldsymbol{\Psi}_2 = \boldsymbol{\Psi}_3 = 10^{10}\mathbf{I}$. Gaussian noise is added to the generated tensor data at SNR = 5dB for denoising task and SNR = 10dB for completion task. To assess the performance, the relative root square error RRSE= $||\mathcal{X} - \widehat{\mathcal{X}}||_F/||\mathcal{X}||_F$ and estimated multilinear rank (if applicable) are shown, where $\widehat{\mathcal{X}}$ is the reconstructed tensor from various algorithms. All results in this subsection are averaged over 200 runs on independently generated tensors and noise realizations.

First, let us look at the case of fully observed data. We consider a tensor size of $I_1 = I_2 = I_3 = 30$ with two multilinear rank settings: $(15,15,15)$ or $(13,15,17)$. We also consider a tensor with unbalanced dimension $I_1 = 20, I_2 = I_3 = 30$ and with rank $(5,15,15)$. From Table I, it can be observed that when the core tensor elements are i.i.d., all Bayesian algorithms (ARD-Tucker, BTC, and the proposed BFCTC-W) give the correct multilinear rank estimate, and they achieve similar performance, although BTC performs slightly better. However, when the tensor core elements have diverse variances, ARD-Tucker and BTC show degradation in rank estimation, as their parameters $\{\boldsymbol{\lambda}^{(k)}\}_{k=1}^{3}$ have dual purposes of learning the multilinear rank and the variances of different core elements. When the core elements have diverse variances, this leads to degraded capability of multilinear rank estimation. On the other hand, the proposed algorithm can still give an average of the multi-linear rank estimates very close to the true values, and is more accurate than ARD-Tucker and BTC. The more accurate rank estimation also translates into smaller RRSE as shown in Table I.

Compared with the optimization algorithms, despite the proposed BFCTC-W does not have the knowledge of ground truth multilinear rank, it still in general outperforms HOOI which is equipped with perfect knowledge of multilinear rank. This is because HOOI does not model noise separately, thus noise is indistinguishable from the signal part and get fitted into the model. On the other hand, for HaLRTC, the model only fits the data to the unfolded matrix (see (2)), thus multilinear rank information is not exploited to achieve

Table I: Rank estimation, RRSE of tensor recovery and run times from the fully observed noisy tensors at SNR = 5dB.

Tensor Setting			HOOI	HaLRTC	ARD-Tucker	BTC	BFCTC-W
size(30,30,30) rank(15,15,15)	i.i.d. elements in the core	averaged estimated rank	-	-	(15,15,15)	(15,15,15)	(15,15,15)
		RRSE	0.2199	0.4032	0.2169	0.2169	0.2241
		run time (s)	0.0446	4.3364	10.5141	109.0463	25.0228
	elements in the core with different variances	averaged estimated rank	-	-	(12.9,12.9,12.9)	(12.8,12.9,12.9)	(14.9,14.8,14.8)
		RRSE	0.2277	0.3666	0.2378	0.2338	**0.2153**
		run time (s)	0.0215	4.1212	10.4463	51.3984	26.8785
size(30,30,30) rank(13,15,17)	i.i.d. elements in the core	averaged estimated rank	-	-	(13,15,17)	(13,15,17)	(13,15,17)
		RRSE	0.2181	0.4009	0.2158	0.2150	0.2223
		run time (s)	0.0228	4.0901	9.8527	97.3099	26.6467
	elements in the core with different variances	averaged estimated rank	-	-	(11.5,12.9,14.0)	(11.3,12.9,14.2)	(13,14.9,16.3)
		RRSE	0.2256	0.3661	0.2383	0.2359	**0.2139**
		run time (s)	0.0237	4.0637	8.1619	49.7182	27.1516
size(20,30,30) rank(5,15,15)	i.i.d. elements in the core	averaged estimated rank	-	-	(5,15,15)	(5,15,15)	(5,15,15)
		RRSE	0.1720	0.3758	0.1704	0.1711	0.1744
		run time (s)	0.0143	3.6567	6.7662	26.3378	25.9109
	elements in the core with different variances	averaged estimated rank	-	-	(5,12.7,11.9)	(5,12.3,12.6)	(5,14.4,14.3)
		RRSE	0.1786	0.3411	0.2045	0.1984	**0.1774**
		run time (s)	0.0147	3.6593	6.9771	25.7328	24.8938

Table II: Rank estimation, RRSE of tensor recovery and run times from the incomplete noisy tensors at SNR = 10dB.

Tensor Setting			CTNM	HaLRTC	W-Tucker	BTC	BFCTC-W
size(30,30,30) rank(10,10,10) SR = 30 %	i.i.d. elements in the core	averaged estimated rank	(12,12,12)	-	-	(10,10,10)	(10,10,10)
		RRSE	0.2115	0.6966	0.1688	0.1574	0.1633
		run time (s)	9.6557	4.3157	13.5903	1694.6	82.3652
	elements in the core with different variances	averaged estimated rank	(12,12,12)	-	-	(8.8,9.0,9.2)	(10,10,10)
		RRSE	0.2134	0.5948	0.3858	0.2061	**0.1691**
		run time (s)	6.7027	4.3352	31.4127	1248.2	80.7936
size(30,30,30) rank(8,10,12) SR = 30 %	i.i.d. elements in the core	averaged estimated rank	(9,9,9)	-	-	(8,10,12)	(8,10,12)
		RRSE	0.4938	0.6916	0.2323	0.1547	0.1603
		run time (s)	8.3659	2.7425	14.9459	1608.7	82.3668
	elements in the core with different variances	averaged estimated rank	(9,9,9)	-	-	(7.5,9.0,10.5)	(8,10,11.9)
		RRSE	0.2423	0.5903	0.3728	0.1989	**0.1662**
		run time (s)	8.2913	3.0755	32.3342	1358.8	78.4391
size(20,30,30) rank(3,10,10) SR = 30 %	i.i.d. elements in the core	averaged estimated rank	(3,12,12)	-	-	(3,10,10)	(3,10,10)
		RRSE	0.1579	0.6086	0.1694	0.1308	0.1325
		run time (s)	7.0098	3.7958	9.9095	436.0154	65.3670
	elements in the core with different variances	averaged estimated rank	(3,12,12)	-	-	(3,7.5,8.2)	(3,9.9,9.9)
		RRSE	0.1616	0.5133	0.2155	0.2157	**0.1394**
		run time (s)	7.0789	3.7571	25.1025	440.0624	66.7978

complexity control. This is reflected in the obviously large RRSE obtained from HaLRTC.

Next, we consider incomplete noisy tensor data and the results are shown in Table II. The tensor size is of $I_1 = I_2 = I_3 = 30$ with rank being $(10, 10, 10)$ or $(8, 10, 12)$. We also consider an unbalanced data dimension of $I_1 = 20, I_2 = I_3 = 30$ with rank $(3, 10, 10)$. The observed data are uniformly random sampled with sampling ratio (SR) = 30% from the synthetic noisy tensor. Since HOOI and ARD-Tucker cannot handle missing data, we compare the performance to CTNM and W-Tucker instead (CTNM could estimate the rank while the input rank to W-Tucker is the true rank). It can be observed from Table II that when the core tensor elements are i.i.d., BTC performs slightly better than the proposed BFCTC-W. This is not surprising, as BTC is a simpler model than the proposed algorithm, and in the i.i.d. core tensor case, the single β parameter in the BTC is sufficient to learn the variance of the core elements so that $\{\boldsymbol{\lambda}^{(k)}\}_{k=1}^{3}$ could dedicate their flexibility to learn an accurate multilinear rank. However, when the tensor core elements are having diverse variances, the proposed BFCTC-W shows unmistakably the best performance. In particular, the proposed BFCTC-W estimates the multilinear rank correctly for the rank $(10, 10, 10)$ while BTC underestimates the multilinear rank. For the cases with multilinear rank $(8, 10, 12)$ and $(3, 10, 10)$, the proposed BFCTC-W gives a closer rank estimate to the ground truth compared to BTC. This in turn reflects in the significantly smaller RRSE achieved by the proposed algorithm compared to BTC.

When comparing to optimization-based methods, the proposed algorithm achieves smaller RRSE than HaLRTC which does not exploit multilinear rank information and CTNM which has unstable multilinear rank estimation. For W-Tucker, even though it is given the accurate multilinear rank, its performance is still not as good as the proposed algorithm.

From the above results, we could see that the proposed BFCTC-W achieves comparable performance to BTC and ARD-Tucker if the elements of the core tensor are i.i.d. However, when the elements of the core tensor are having diverse variances, the proposed method not only determines the multilinear rank more accurately than BTC, ARD-Tucker and CTNM, but also recovers the tensor data more accurately. This shows the wider adaptability of the proposed algorithm under different core tensor scenarios. Finally, it is noted from Table I and II that the proposed BFCTC-W takes even shorter computation times than the BTC in most of the cases, making the proposed method being a better choice than BTC in terms of both performance and computation speed.

Table III: RRSE and PSNR of image de-noising at SNR = 5dB.

Color Image	HOOI		HaLRTC		ARD-Tucker		BCPF		BTC		BFCTC-W	
	RRSE	PSNR	RRSE	PSNR	RRSE	PSNR	RRSE	PSNR	RRSE	PSNR	RRSE	PSNR
peppers	0.1387	23.0752	0.1390	23.0564	0.2085	19.5346	0.1337	23.3949	0.1327	23.4593	**0.1051**	**25.4841**
lena	0.1135	24.0207	0.1142	23.9673	0.1572	21.1915	0.1026	24.8991	0.1117	24.1595	**0.0821**	**26.8368**
barbara	0.1199	24.8287	0.1324	23.9673	0.1836	21.1276	0.1124	25.3903	0.1209	24.7565	**0.0946**	**26.8918**
house	0.0907	25.4619	0.0943	25.1238	0.1451	21.3807	0.0808	26.4623	0.0890	25.6262	**0.0642**	**28.4688**
airplane	0.0666	26.2068	0.0842	24.1700	0.1268	20.6139	0.0610	26.9744	0.0824	24.3577	**0.0548**	**27.9076**
sailboat	0.1357	22.5032	0.1345	22.5804	0.1854	19.7926	0.1247	23.2399	0.1409	22.1766	**0.1102**	**24.3148**
facade	0.0826	27.4062	0.0936	26.3203	0.1153	24.5093	0.0696	28.8881	0.0815	27.5227	**0.0664**	**29.3082**
baboon	0.1482	21.9409	0.1444	22.1665	0.2045	19.1440	0.1330	22.8810	0.1568	21.4509	**0.1310**	**23.0100**

Table IV: RRSE and PSNR of image completion at SR = 30% and SNR = 20dB.

Color Image	CTNM		HaLRTC		W-Tucker		BCPF		BTC		BFCTC-W	
	RRSE	PSNR	RRSE	PSNR	RRSE	PSNR	RRSE	PSNR	RRSE	PSNR	RRSE	PSNR
peppers	0.1227	24.1398	0.1275	23.8065	0.1596	21.8560	0.1011	25.8240	0.1478	22.5232	**0.0837**	**27.4622**
lena	0.0991	25.1991	0.0994	25.1729	0.1230	23.3225	0.0777	27.3159	0.1102	24.2770	**0.0713**	**28.0588**
barbara	0.1171	25.0339	0.1187	24.9160	0.1305	24.0928	0.1058	25.9186	0.1235	24.5717	**0.0846**	**27.8576**
house	0.0733	27.3119	0.0786	26.7056	0.0982	24.7718	0.0809	26.4539	0.0932	25.2257	**0.0527**	**30.1778**
airplane	0.0791	24.7128	0.0766	24.9917	0.0907	23.5241	0.0724	25.4778	0.0971	22.9319	**0.0587**	**27.3035**
sailboat	0.1405	22.2013	0.1327	22.6974	0.1466	21.8321	0.1181	23.7101	0.1769	20.2003	**0.1015**	**25.0255**
facade	0.0702	28.8191	0.0680	29.0957	0.0789	27.8043	0.0770	28.0208	0.1058	25.2561	**0.0610**	**30.0392**
baboon	0.1673	20.8879	0.1500	21.8360	0.1850	20.0144	0.1522	21.7119	0.1982	19.4158	**0.1414**	**22.3489**

Table V: RRSE and PSNR of image completion at SR = 20% and SNR = 10dB.

Color Image	CTNM		HaLRTC		W-Tucker		BCPF		BTC		BFCTC-W	
	RRSE	PSNR	RRSE	PSNR	RRSE	PSNR	RRSE	PSNR	RRSE	PSNR	RRSE	PSNR
peppers	0.2159	19.2316	0.2101	19.4682	0.3604	14.7810	0.1393	23.0369	0.2183	19.1356	**0.1290**	**23.7056**
lena	0.1700	20.5116	0.1584	21.1255	0.2649	16.6590	0.1043	24.7548	0.2263	18.0269	**0.1009**	**25.0395**
barbara	0.1928	20.7029	0.1863	21.0008	0.2755	17.6026	0.1306	24.0854	0.1984	20.4542	**0.1202**	**24.8060**
house	0.1426	21.5316	0.1307	22.2885	0.1989	18.6413	0.0970	24.8778	0.1576	20.6629	**0.0812**	**26.4216**
airplane	0.1243	20.7869	0.1132	21.5994	0.1599	18.5993	0.0847	24.1183	0.1579	18.7086	**0.0809**	**24.5126**
sailboat	0.2305	17.9014	0.1966	19.2832	0.3244	14.9332	0.1460	21.8651	0.2199	18.3103	**0.1420**	**22.1065**
facade	0.1239	23.8844	0.1025	25.5314	0.1627	21.5181	0.1037	25.4319	0.1062	25.2234	**0.0871**	**26.9437**
baboon	0.2319	18.0518	0.1915	19.7145	0.3131	15.4442	0.1676	20.8737	0.2070	19.0384	**0.1627**	**21.1320**

B. Image data

Next we show the results of 8 RGB benchmark images each with size 256×256×3. For the denoising task, we set SNR = 5dB. For the completion task, we set SNR = 20dB, SR = 30% and SNR = 10dB, SR = 20%. As the image data contain smoothness among neighboring pixels, we choose $\boldsymbol{\Psi}_1 = \boldsymbol{\Psi}_2 = \boldsymbol{\Psi}_3 = \boldsymbol{F}^T\boldsymbol{F}$, where $\boldsymbol{F}$ is defined in (10). The results are shown in Table III for denoising and Tables IV, V for completion. For HOOI and W-Tucker, the multilinear rank is set at $(40, 40, 3)$ [50]. For CTNM, the upper bound of the multilinear rank is set as $(40, 40, 3)$ [51], while HaLRTC does not make use of the multilinear rank information. For this experiment, we also compare the result to BCPF [2], which is the Bayesian CPD and can be considered as a restricted form of Bayesian Tucker, and the CPD rank is learnt from the algorithm.

It can be seen from Tables III, IV and V that the proposed BFCTC-W method achieves the best RRSE and PSNR in all tested images, due to the more flexible Tucker core in the proposed model. To examine the visual differences, Figure 2 shows two examples (Lena and House) of the de-noised images and Figure 3 shows two examples (Sailboat and Peppers) of the completed images (SNR = 20dB, SR = 30%) from various algorithms (locally enlarged details are shown in the second and forth rows of the figures). It can be seen that the proposed BFCTC-W recovers the best images, which loses less image details than BTC, BCPF, ADR-Tucker and HaLRTC, and removes more noise than HOOI, CTNM and W-Tucker, achieving a better balance between noise removal and recovering image details.

C. MRI data

Finally, we evaluate BFCTC-W method on recovering MRI dataset "T1 ICBM normal 1mm pn0 rf0" [27] with size 181×217×181, and with i.i.d. Gaussian noise added to the full data at SNR = 5dB and to the incomplete data (SR = 30%) at SNR = 20dB. As the MRI data contain smoothness among neighboring pixels, we also choose $\boldsymbol{\Psi}_1 = \boldsymbol{\Psi}_2 = \boldsymbol{\Psi}_3 = \boldsymbol{F}^T\boldsymbol{F}$, where $\boldsymbol{F}$ is defined in (10). Since the dimensions and rank of MRI data are generally high, it is commonly believed that [27] the data as a whole does not has a globally low-rank structure. However, low-rank structure may exist in smaller block. Hence, we follow [27] and apply Tucker decompositions to non-overlapping blocks of data independently.

Table VI and Table VII show the denoising and completion performance of various algorithms for five blocks of the MRI data, each of size $181 \times 217 \times 5$. For the HOOI algorithm, we consider two ways to set its multilinear rank. The first one is by carefully tunning it to give the smallest RRSE, and it was found that the most suitable rank for HOOI is (30,30,2). The second one is by using the multilinear rank from the proposed BFCTC-W. And the upper bound of multilinear rank of CTNM and the input multilinear rank of W-Tucker are set in the same way as that in image data in the last section.

Figure 2: Examples of the de-noised images at SNR = 5dB.

It can be seen that the proposed BFCTC-W performs the best in terms of RRSE due to its more flexible core compared to other Bayesian algorithms. Furthermore, due to the advantages of explicit modeling of noise, the proposed algorithm also outperforms optimization-based methods. Figure 4 shows two examples (block1 and block5) of the MRI completion results from different algorithms (locally enlarged details are shown in the second and forth rows of the figures). It can be seen that the proposed BFCTC-W recovers the best images, which loses less image details and removes more noise compared to other algorithms.

VI. Conclusion

We have proposed a Bayesian Flexible Core Tucker tensor completion model by revealing that Tucker decomposition is equivalent to a CPD with factor matrices having large number of columns while being low-rank. With the low-rank structure in the factor matrices being modeled by the Gaussian-inverse Wishart prior, an inference algorithm with closed-form update has further been derived under the variational Bayesian framework. Extensive experiments on synthetic data have showed that the proposed model achieves more accurate multilinear rank estimation, and higher tensor recovery accuracies compared to previous Bayesian Tucker models when the elements of the core tensor are having diverse variances. Further experiments on color images and MRI data validated the superiority of the proposed method in practical applications.

References

[1] X. Zhang and M. K. Ng, "A corrected tensor nuclear norm minimization method for noisy low-rank tensor completion," *SIAM J. Imag. Sci.*,vol. 12, no. 2, pp. 1231–1273, 2019.

[2] Q. Zhao, L. Zhang and A. Cichocki, "Bayesian CP factorization of incomplete tensors with automatic rank determination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1751-1763, Sep. 2015.

[3] Z. Long, C. Zhu, J. Liu, and Y. Liu, "Bayesian low rank tensor ring for image recovery," *IEEE Trans. Image Process.*, vol. 30, pp. 3568–3580, 2021.

[4] H. Xu, J. Zheng, X. Yao, Y. Feng, and S. Chen, "Fast tensor nuclear norm for structured low-rank visual inpainting," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 2, pp. 538–552, Feb. 2022, doi: 10.1109/TCSVT.2021.3067022.

[5] P. A. Chew, B. W. Bader, T. G. Kolda, and A. Abdelali, "Cross-language information retrieval using PARAFAC2," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 143–152.

[6] Y Jin, M Liu, Y Li, R Xu, L Du, L Gao, Y Xiang, "Variational auto-encoder based Bayesian Poisson tensor factorization for sparse and imbalanced count data," in *Proc. Data Min. Knowl. Discov.*, 2020, pp. 1–28.

[7] M. Signoretto, Q. T. Dinh, L. De Lathauwer, and J. A. K. Suykens, "Learning with tensors: A framework based on convex optimization and spectral regularization," *Mach. Learn.*, vol. 94, no. 3, pp. 303–351, 2014.

[8] Q. Zhao, G. Zhou, T. Adali, L. Zhang, and A. Cichocki, "Kernelization of tensor-based models for multiway data analysis: Processing of multidimensional structured data," *Signal Processing Magazine, IEEE*, vol. 30, no. 4, pp. 137–148, 2013.

[9] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and Gabor features for gait recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1700–1715, Oct. 2007.

[10] M. Vasilescu and D. Terzopoulos, *Multilinear Analysis of Image Ensembles: TensorFaces*. Berlin, Germany: Springer-Verlag, 2002, pp. 447–460.

[11] X. Zhang, D. Wang, Z. Zhou, and Y. Ma, "Robust low-rank tensor recovery with rectification and alignment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 238–255, Jan. 2021.

[12] R. Bro, "PARAFAC. Tutorial and applications," *Chemometrics and intelligent laboratory systems*, vol. 38, no. 2, pp. 149–171, 1997

[13] Lei Cheng, Yik-Chung Wu, and H. Vincent Poor, "Scaling Probabilistic Tensor Canonical Polyadic Decomposition to Massive Data," *IEEE Trans. on Signal Processing*, vol. 66, no. 21, pp. 5534-5548, Nov 2018.

[14] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966

[15] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2295–2317, Jan. 2011.

[16] Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki, "Tensor ring decomposition," 2016, *arXiv:1606.05535*.

[17] T. G. Kolda, B. W. Bader, "Tensor Decompositions and Applications," *SIAM REVIEW*, Vol. 51, No. 3, pp. 455–500, 2009.

[18] I. V. Oseledets, "Tensor-train decomposition," *SIAM Journal on Scientific Computing*, 33(5), 2295-2317, 2011.

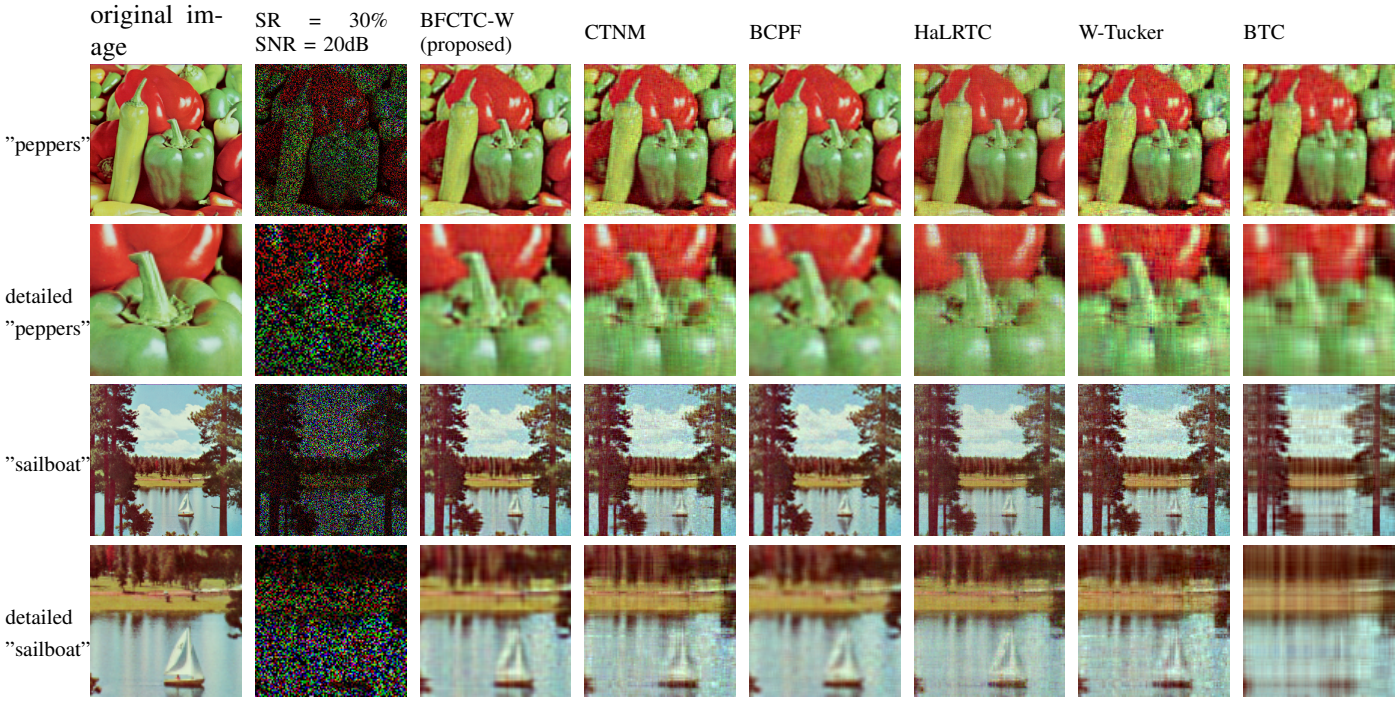

Figure 3: Examples of the recovered images at SR = 30% and SNR = 20dB.

Table VI: RRSE and PSNR of MRI de-noising at SNR = 5dB.

MRI	HOOI		HOOI (BFCTC-W rank)		HaLRTC		ARD-Tucker		BTC		BFCTC-W	
	RRSE	PSNR	RRSE	PSNR	RRSE	PSNR	RRSE	PSNR	RRSE	PSNR	RRSE	PSNR
Block1	0.1348	25.9105	0.1561	24.6363	0.3396	17.8850	0.1647	24.1704	0.1341	25.9557	**0.1205**	**26.8857**
Block2	0.1412	26.3535	0.2060	23.0728	0.3179	19.3044	0.1892	23.8117	0.1431	26.2374	**0.1273**	**27.2560**
Block3	0.1480	26.3605	0.1797	24.6747	0.3015	20.1800	0.1891	24.2319	0.1474	26.3958	**0.1332**	**27.2733**
Block4	0.1496	26.2426	0.1856	24.3697	0.2863	20.6048	0.1971	23.8475	0.1480	26.3360	**0.1346**	**27.1575**
Block5	0.1516	26.1348	0.1856	24.3773	0.2917	20.4501	0.1918	24.0918	0.1491	26.2793	**0.1362**	**27.0672**

Table VII: RRSE and PSNR of MRI completion at SR = 30% and SNR = 20dB.

MRI	CTNM		HaLRTC		W-Tucker		BTC		BFCTC-W	
	RRSE	PSNR	RRSE	PSNR	RRSE	PSNR	RRSE	PSNR	RRSE	PSNR
Block1	0.1106	27.6292	0.1286	26.3195	0.1137	27.3891	0.1436	25.3612	**0.0772**	**30.7507**
Block2	0.1164	28.0311	0.1388	26.5024	0.1215	27.6586	0.1620	25.1599	**0.0806**	**31.2228**
Block3	0.1255	27.7928	0.1447	26.5563	0.1232	27.9535	0.1619	25.5808	**0.0837**	**31.3079**
Block4	0.1244	27.8448	0.1473	26.3772	0.1339	27.2056	0.1677	25.2506	**0.0856**	**31.0924**
Block5	0.1239	27.8874	0.1461	26.4558	0.1313	27.3835	0.1664	25.3257	**0.0863**	**31.0300**

[19] Q. Zhao, G. Zhou, S. Xie, L. Zhang and A. Cichocki, "Tensor ring decomposition," *arXiv preprint arXiv:1606.05535*. 2016.

[20] Le Xu, Lei Cheng, Ngai Wong, and Yik-Chung Wu, "Tensor Train Factorization under Noisy and Incomplete Data with Automatic Rank Estimation," *Pattern Recognition*, vol. 141, 2023.

[21] Z. Xu, F. Yan, and A. Qi, "Infinite Tucker decomposition: Nonparametric Bayesian models for multiway data analysis," *in Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, 2012, pp. 1023–1030.

[22] S. Fang, Z. Wang, Z. Pan, J. Liu and S. Zhe, "Streaming Bayesian Deep Tensor Factorization," *In International Conference on Machine Learning* (pp. 3133-3142), July 2021. PMLR.

[23] J. B. Kruskal, "Rank, decomposition, and uniqueness for 3-way and N-way arrays," *in Multiway Data Analysis, R. Coppi and S. Bolasco, eds., North-Holland, Amsterdam*, 1989, pp. 7–18.

[24] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 24, no. 4, pp. 1253–1278, 2000.

[25] A. Cichocki, R. Zdunek, A. H. Phan, and S. I. Amari, *Nonnegative Matrix and Tensor Factorizations*. John Wiley & Sons, 2009.

[26] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the best rank-1 and rank-(R1,R2, . . . ,RN) approximation of higher-order tensors," *SIAM J. Matrix Anal. Appl.*, pp. 1324–1342, 2000.

[27] Q. Zhao, L. Zhang and A. Cichocki, "Bayesian sparse Tucker models for dimension reduction and tensor completion," in , May 2015, [online] Available: http://arxiv.org/abs/1505.02343.

[28] M. Mørup and L. K. Hansen, "Automatic relevance determination for multi-way models," *Journal of Chemometrics*, vol. 23, no. 7-8, pp. 352–363, 2009.

[29] C. M. Bishop, Pattern Recognition and Machine Learning, New York, NY, USA:Springer-Verlag, 2006.

[30] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3551-3582, Jul. 2017.

[31] M. C. Vanderveen, B. C. Ng, C. B. Papadias and A. Paulraj, "Joint angle and delay estimation (JADE) for signals in multipath environments," *30th Asilomar Conf. on Circuit Systems and Computer*, pp. 1250-1254, November 1996.

[32] V. I. Slyusar, "End products in matrices in radar applications," *Radioelectronics and Communications Systems*, 41 (3): 50–53, December 1996.

[33] L. Yang, J. Fang, H. Duan, H. Li and B. Zeng, "Fast low-rank Bayesian matrix completion with hierarchical gaussian prior models," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2804-2817, Jun. 2018.

[34] E. Candes, M.Wakin, and S. Boyd, "Enhancing sparsity by reweighted

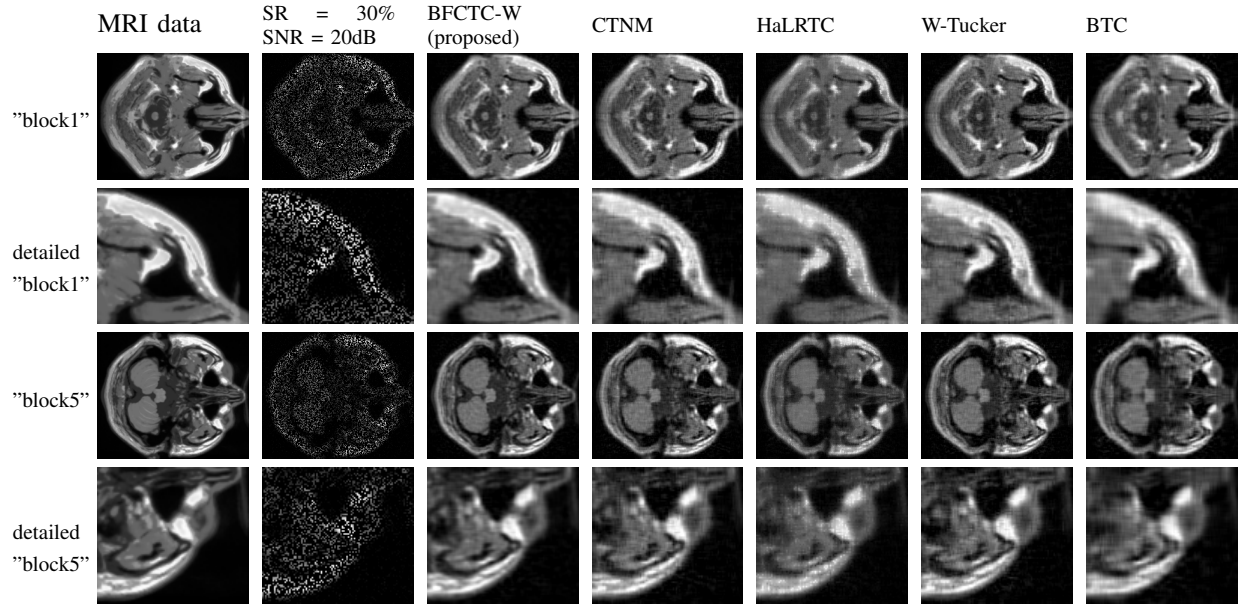

Figure 4: Examples of the recovered MRI data at SR = 30% and SNR = 20dB.

l1 minimization," *J. Fourier Anal. Appl.*, vol. 14, pp. 877–905, Dec. 2008.

[35] Y. Shen, J. Fang, and H. Li, "Exact reconstruction analysis of log-sum minimization for compressed sensing," *IEEE Signal Process. Lett.*, vol. 20, no. 12, pp. 1223–1226, Dec. 2013.

[36] X. Fu, K. Huang, W.-K. Ma, N. D. Sidiropoulos, and R. Bro, "Joint tensor factorization and outlying slab suppression with applications," *IEEE Trans. Signal Process.*, vol. 63, no. 23, pp. 6315–6328, Dec. 2015.

[37] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[38] D. Kong, C. Ding, H. Huang, and F. Nie, "An iterative locally linear embedding algorithm," *in Proceedings of the 29th International Conference on International Conference on Machine Learning*, 2012, pp. 931–938.

[39] Y. Chen, L. Cheng, and Y.-C. Wu, "Bayesian Low-rank Matrix Completion with Dual-graph Embedding: Prior Analysis and Tuning-free Inference," *Signal Process*, 204, 108826 (2023).

[40] Z. Chen, R. Molina, and A. K. Katsaggelos, "Robust recovery of temporally smooth signals from under-determined multiple measurements," *IEEE Trans. Signal Process.*, vol. 63, no. 7, pp. 1779–1791, Apr. 2015.

[41] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P.Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[42] F. R. Chung and F. C. Graham, *Spectral graph theory*. American Mathematical Soc., 1997, no. 92.

[43] De Lathauwer, Lieven, "Decompositions of a higher-order tensor in block terms- Part II: Definitions and uniqueness," *SIAM Journal on Matrix Analysis and Applications* 30.3 (2008): 1033-1066.

[44] P. V. Giampouras, A. A. Rontogiannis and E. Kofidis, "Block-Term Tensor De- composition Model Selection and Computation: The Bayesian Way," *in IEEE Transactions on Signal Processing*, vol. 70, pp. 1704-1717, 2022, doi: 10.1109/TSP.2022.3159029.

[45] F. Roemer and M. Haardt, "A semi-algebraic framework for approximate CP decompositions via simultaneous matrix diagonalizations (SECSI)," *Signal Processing*, vol. 93, no. 9, pp. 2722-2738, 2013.

[46] M. J. Wainwright and M. I. Jordan, "Graphical models exponential families and variational inference," *Found. Trends Mach. Learn.*, vol. 1, no. 1-2, pp. 1-305, Jan. 2008.

[47] C. Zhang, J. Butepage, H. Kjellstrom and S Mandt, "Advances in variational inference," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 2008-2026, Aug. 2019.

[48] L. Gui, Q. Zhao and J. Cao, "Brain image completion by Bayesian tensor decomposition," *Proc. 22nd Int. Conf. Digit. Signal Process. (DSP)*, pp. 1-4, 2017.

[49] L. Gui, X Zhao, Q. Zhao and J. Cao, "Image and Video Completion by Using Bayesian Tensor Decomposition," *International Journal of Computer Science Issues*, Volume 15, Issue 5, September 2018.

[50] M. Filipovic and A. Jukic, "Tucker factorization with missing data with application to low-n-rank tensor completion," *Multidimensional Syst. Signal Process.*, vol. 26, no. 3, pp. 677–692, Jul. 2015.

[51] Y. Liu, F. Shang, W. Fan, J. Cheng, and H. Cheng, "Generalized higher order orthogonal iteration for tensor learning and decomposition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2551–2563, Dec. 2016.

[52] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 208–220, 2013.

APPENDIX A
DERIVATION OF THE ALGORITHM

Derivation for $Q(\boldsymbol{B}_{:,l}^{(k)})$:

Gathering the terms in (11) that are related to $\boldsymbol{B}_{:,l}^{(k)}$, it can be expressed as

$$\begin{aligned} & p(\boldsymbol{\Theta}, \mathcal{Y}_{\boldsymbol{\Omega}}) \\ & \propto \exp\Bigg\{ -\frac{\alpha}{2} \Bigg\| \mathcal{O} * \Bigg(\mathcal{Y} - \sum_{l=1}^{L} \Big(\boldsymbol{B}_{:,l}^{(1)} \circ \boldsymbol{B}_{:,l}^{(2)} \circ \boldsymbol{B}_{:,l}^{(3)} \Big) \Bigg) \Bigg\|_F^2 \\ & \qquad - \frac{1}{2} \boldsymbol{B}_{:,l}^{(k)^T} \boldsymbol{\Sigma}_k \boldsymbol{B}_{:,l}^{(k)} \Bigg\} \\ & \propto \exp\Bigg\{ -\frac{1}{2} \Big(\boldsymbol{B}_{:,l}^{(k)^T} \Big(\boldsymbol{\xi}_l^{(k)} + \boldsymbol{\Sigma}_k \Big) \boldsymbol{B}_{:,l}^{(k)} \\ & \qquad + \boldsymbol{\omega}_l^{(k)} \boldsymbol{B}_{:,l}^{(k)} + \boldsymbol{B}_{:,l}^{(k)^T} \boldsymbol{\omega}_l^{(k)^T} \Big) \Bigg\}. \end{aligned} \tag{20}$$

In (20), $\boldsymbol{\xi}_l^{(k)} = \alpha \cdot \text{diag}\left(\boldsymbol{O}_{(k)} \bigodot_{h \neq k} (\boldsymbol{B}_{:,l}^{(h)} * \boldsymbol{B}_{:,l}^{(h)}) \right) \in \mathbb{R}^{I_k \times I_k}$, where $\boldsymbol{O}_{(k)}$ is the mode-k unfolding of the 3-order tensor $\mathcal{O}$, and $\bigodot_{h \neq k} (\boldsymbol{B}_{:,l}^{(h)} * \boldsymbol{B}_{:,l}^{(h)}) = (\boldsymbol{B}_{:,l}^{(3)} * \boldsymbol{B}_{:,l}^{(3)}) \odot \cdots \odot (\boldsymbol{B}_{:,l}^{(k+1)} * \boldsymbol{B}_{:,l}^{(k+1)}) \odot (\boldsymbol{B}_{:,l}^{(k-1)} * \boldsymbol{B}_{:,l}^{(k-1)}) \odot \cdots \odot (\boldsymbol{B}_{:,l}^{(1)} * \boldsymbol{B}_{:,l}^{(1)})$. Furthermore, $\boldsymbol{\omega}_l^{(k)}$ is a $1 \times I_k$ vector and is defined by

$\boldsymbol{\omega}_l^{(k)} = \alpha\Big\{ \sum_{p\neq l} \text{diag}\Big(\boldsymbol{O}_{(k)} \bigodot_{h\neq k} (\boldsymbol{B}_{:,l}^{(h)} * \boldsymbol{B}_{:,p}^{(h)}) \boldsymbol{B}_{:,p}^{(k)^T}\Big)^T - \Big[\big(\boldsymbol{Y}_{(k)} * \boldsymbol{O}_{(k)}\big)\Big(\bigodot_{h\neq k} \boldsymbol{B}_{:,l}^{(h)}\Big)\Big]^T \Big\}$, where $\boldsymbol{Y}_{(k)}$ is the mode-k unfolding of the 3-order tensor $\mathcal{Y}$, and $\bigodot_{h\neq k} \boldsymbol{B}_{:,l}^{(h)} = \boldsymbol{B}_{:,l}^{(3)} \odot \cdots \odot \boldsymbol{B}_{:,l}^{(k+1)} \odot \boldsymbol{B}_{:,l}^{(k-1)} \odot \cdots \odot \boldsymbol{B}_{:,l}^{(1)}$.

Putting (20) into (13) gives

$$\begin{aligned} Q\left(\boldsymbol{B}_{:,l}^{(k)}\right) &\propto -\frac{1}{2}\exp\Big\{ \boldsymbol{B}_{:,l}^{(k)^T} \mathbb{E}_{\boldsymbol{\Theta}\backslash \boldsymbol{B}_{:,l}^{(k)}}\left[\boldsymbol{\xi}_l^{(k)} + \boldsymbol{\Sigma}_k\right] \boldsymbol{B}_{:,l}^{(k)} \\ &\quad + \mathbb{E}_{\boldsymbol{\Theta}\backslash \boldsymbol{B}_{:,l}^{(k)}}\left[\boldsymbol{\omega}_l^{(k)}\right] \boldsymbol{B}_{:,l}^{(k)} + \boldsymbol{B}_{:,l}^{(k)^T} \mathbb{E}_{\boldsymbol{\Theta}\backslash \boldsymbol{B}_{:,l}^{(k)}}\left[\boldsymbol{\omega}_l^{(k)^T}\right] \Big\} \\ &\propto \mathcal{N}\Bigg(\underbrace{-(\mathbb{E}_{\boldsymbol{\Theta}\backslash \boldsymbol{B}_{:,l}^{(k)}}\left[\boldsymbol{\xi}_l^{(k)} + \boldsymbol{\Sigma}_k\right])^{-1} \mathbb{E}_{\boldsymbol{\Theta}\backslash \boldsymbol{B}_{:,l}^{(k)}}\left[\boldsymbol{\omega}_l^{(k)}\right]^T}_{:=\boldsymbol{m}_l^{(k)}}, \\ &\qquad \underbrace{(\mathbb{E}_{\boldsymbol{\Theta}\backslash \boldsymbol{B}_{:,l}^{(k)}}\left[\boldsymbol{\xi}_l^{(k)} + \boldsymbol{\Sigma}_k\right])^{-1}}_{:=\boldsymbol{\Upsilon}_l^{(k)^{-1}}} \Bigg). \end{aligned} \tag{21}$$

So for each column of $\boldsymbol{B}^{(k)}$, the variational distribution is Gaussian. Putting the definition of $\boldsymbol{\xi}_l^{(k)}$ and $\boldsymbol{\omega}_l^{(k)}$ into $\boldsymbol{\Upsilon}_l^{(k)}$ and $\boldsymbol{m}_l^{(k)}$, and recognizing that $\boldsymbol{B}_{:,l}^{(h)^T} \boldsymbol{B}_{:,p}^{(h)} = [\boldsymbol{B}^{(h)^T} \boldsymbol{B}^{(h)}]_{l,p}$ is a scalar, the covariance and mean of $Q(\boldsymbol{B}_{:,l}^{(k)})$ are given by (note that while not explicitly stated, all the expectations are with respect to the variational distributions of $\boldsymbol{\Theta}\backslash \boldsymbol{B}_{:,l}^{(k)}$):

$$\begin{aligned} \boldsymbol{\Upsilon}_l^{(k)^{-1}} &= \Bigg(\mathbb{E}\left[\boldsymbol{\Sigma}_k\right] \\ &\quad + \mathbb{E}\left[\alpha\right] \text{diag}\left(\boldsymbol{O}_{(k)} \bigodot_{h\neq k} \mathbb{E}\left[\boldsymbol{B}_{:,l}^{(h)} * \boldsymbol{B}_{:,l}^{(h)}\right]\right) \Bigg)^{-1} \\ \boldsymbol{m}_l^{(k)} &= -\boldsymbol{\Upsilon}_l^{(k)^{-1}} \mathbb{E}\left[\alpha\right] \Bigg\{ \\ &\quad \sum_{p\neq l} \text{diag}\Big(\boldsymbol{O}_{(k)} \bigodot_{h\neq k} \mathbb{E}\big[\boldsymbol{B}_{:,l}^{(h)} * \boldsymbol{B}_{:,p}^{(h)}\big] \cdot \mathbb{E}\left[\boldsymbol{B}_{:,p}^{(k)^T}\right]\Big) \\ &\quad - \Bigg[\big(\boldsymbol{Y}_{(k)} * \boldsymbol{O}_{(k)}\big) \cdot \Big(\bigodot_{h\neq k} \mathbb{E}\left[\boldsymbol{B}_{:,l}^{(h)}\right]\Big) \Bigg] \Bigg\}, \end{aligned} \tag{22}$$

where $\bigodot_{h\neq k} \mathbb{E}\left[\boldsymbol{B}_{:,l}^{(h)}\right] = \mathbb{E}\left[\boldsymbol{B}_{:,l}^{(3)}\right] \odot \cdots \odot \mathbb{E}\left[\boldsymbol{B}_{:,l}^{(k+1)}\right] \odot \mathbb{E}\left[\boldsymbol{B}_{:,l}^{(k-1)}\right] \odot \cdots \odot \mathbb{E}\left[\boldsymbol{B}_{:,l}^{(1)}\right]$ and $\bigodot_{h\neq k} \mathbb{E}\left[\boldsymbol{B}_{:,l}^{(h)} * \boldsymbol{B}_{:,l}^{(h)}\right] = \mathbb{E}\left[\boldsymbol{B}_{:,l}^{(3)} * \boldsymbol{B}_{:,l}^{(3)}\right] \odot \cdots \odot \mathbb{E}\left[\boldsymbol{B}_{:,l}^{(k+1)} * \boldsymbol{B}_{:,l}^{(k+1)}\right] \odot \mathbb{E}\left[\boldsymbol{B}_{:,l}^{(k-1)} * \boldsymbol{B}_{:,l}^{(k-1)}\right] \odot \cdots \odot \mathbb{E}\left[\boldsymbol{B}_{:,l}^{(1)} * \boldsymbol{B}_{:,l}^{(1)}\right]$.

In order to compute (22), we need the mean of various columns of $\boldsymbol{B}^{(h)}$, which are simply given by $\mathbb{E}\left[\boldsymbol{B}_{:,p}^{(h)}\right] = \boldsymbol{m}_p^{(h)}$. For $\mathbb{E}\left[\boldsymbol{B}_{:,l}^{(h)} * \boldsymbol{B}_{:,p}^{(h)}\right]$, it is recognized that this is the expectation of elementwise product of $\boldsymbol{B}_{:,l}^{(h)}$ and $\boldsymbol{B}_{:,p}^{(h)}$. This gives $\mathbb{E}\left[\boldsymbol{B}_{:,l}^{(h)} * \boldsymbol{B}_{:,p}^{(h)}\right] = \boldsymbol{m}_l^{(h)} * \boldsymbol{m}_p^{(h)}$, if $l \neq p$. Furthermore, $\mathbb{E}\left[\boldsymbol{B}_{:,l}^{(h)} * \boldsymbol{B}_{:,l}^{(h)}\right] = \boldsymbol{m}_l^{(h)} * \boldsymbol{m}_l^{(h)} + \text{diag}(\boldsymbol{\Upsilon}_l^{(h)^{-1}})$, where diag(.) takes the diagonal of the matrix and put it as a vector. Putting these results into (22), we obtain (15).

Derivation for $Q(\boldsymbol{\Sigma}_k)$:

Gathering the terms in (11) that are related to $\boldsymbol{\Sigma}_k$, (11) can be expressed as

$$\begin{aligned} p(\boldsymbol{\Theta}, \mathcal{Y}_{\boldsymbol{\Omega}}) &\propto \exp\Bigg\{ \frac{L}{2} \ln|\boldsymbol{\Sigma}_k| - \frac{1}{2} \sum_{l=1}^{L} \boldsymbol{B}_{:,l}^{(k)^T} \boldsymbol{\Sigma}_k \boldsymbol{B}_{:,l}^{(k)} \\ &\quad + \frac{v_k - I_k - 1}{2} \ln|\boldsymbol{\Sigma}_k| - \frac{1}{2}\text{Tr}\left(\boldsymbol{\Psi}_k^{-1} \boldsymbol{\Sigma}_k\right) \Bigg\} \\ &= \exp\Bigg\{ \frac{v_k + L - I_k - 1}{2} \ln|\boldsymbol{\Sigma}_k| \\ &\quad - \frac{1}{2}\text{Tr}\Big(\Big(\boldsymbol{\Psi}_k^{-1} + \boldsymbol{B}^{(k)} \boldsymbol{B}^{(k)^T}\Big) \boldsymbol{\Sigma}_k \Big) \Bigg\}. \end{aligned} \tag{23}$$

Putting (23) into (13) gives

$$\begin{aligned} Q\left(\boldsymbol{\Sigma}_k\right) &\propto \exp\Bigg\{ \frac{v_k + L - I_k - 1}{2} \ln|\boldsymbol{\Sigma}_k| \\ &\quad - \frac{1}{2}\text{Tr}\Big(\Big(\boldsymbol{\Psi}_k^{-1} + \mathbb{E}_{\boldsymbol{\Theta}\backslash \boldsymbol{\Sigma}_k}\left[\boldsymbol{B}^{(k)} \boldsymbol{B}^{(k)^T}\right]\Big) \boldsymbol{\Sigma}_k \Big) \Bigg\} \\ &\propto \text{Wishart}\Bigg(\underbrace{v_k + L}_{:=\hat{v}_k}, \underbrace{\Big(\boldsymbol{\Psi}_k^{-1} + \mathbb{E}_{\boldsymbol{\Theta}\backslash \boldsymbol{\Sigma}_k}\left[\boldsymbol{B}^{(k)} \boldsymbol{B}^{(k)^T}\right]\Big)^{-1}}_{:=\hat{\boldsymbol{\Psi}}_k} \Bigg). \end{aligned} \tag{24}$$

Thus, the variational distribution $Q(\boldsymbol{\Sigma}_k)$ is the Wishart distribution

$$Q(\boldsymbol{\Sigma}_k) \propto \text{Wishart}(\hat{v}_k, \hat{\boldsymbol{\Psi}}_k), \tag{25}$$

where $\hat{v}_k$ and $\hat{\boldsymbol{\Psi}}_k$ are updated by

$$\begin{aligned} \hat{v}_k &= v_k + L \\ \hat{\boldsymbol{\Psi}}_k &= \Big(\boldsymbol{\Psi}_k^{-1} + \mathbb{E}_{\boldsymbol{\Theta}\backslash \boldsymbol{\Sigma}_k}\left[\boldsymbol{B}^{(k)} \boldsymbol{B}^{(k)^T}\right]\Big)^{-1}. \end{aligned} \tag{26}$$

For the expectation $\mathbb{E}_{\boldsymbol{\Theta}\backslash \boldsymbol{\Sigma}_k}\left[\boldsymbol{B}^{(k)} \boldsymbol{B}^{(k)^T}\right]$, it is obtained from $Q(\boldsymbol{B}_{:,l}^{(k)})$ and can be shown to be equal to $\boldsymbol{M}^{(k)} \left(\boldsymbol{M}^{(k)}\right)^T + \sum_{l=1}^{L} \boldsymbol{\Upsilon}_l^{(k)^{-1}}$, with $\boldsymbol{M}^{(k)} = \left[\boldsymbol{m}_1^{(k)}, \boldsymbol{m}_2^{(k)}, ..., \boldsymbol{m}_L^{(k)}\right]$.

Derivation for $Q(\alpha)$:

Gathering the terms in (11) that are related to α, it can be expressed as

$$\begin{aligned} &p(\boldsymbol{\Theta}, \mathcal{Y}_{\boldsymbol{\Omega}}) \\ &\propto \exp\Bigg\{ -\frac{\alpha}{2} \Bigg\| \mathcal{O} * \Big(\mathcal{Y} - \sum_{l=1}^{L} \Big(\boldsymbol{B}_{:,l}^{(1)} \circ \boldsymbol{B}_{:,l}^{(2)} \circ \boldsymbol{B}_{:,l}^{(3)}\Big)\Big) \Bigg\|_F^2 \\ &\quad + \frac{\sum_{i_1 i_2 i_3} \mathcal{O}_{i_1 i_2 i_3}}{2} \ln\alpha + (c-1)\ln\alpha - d\alpha \Bigg\} \\ &= \exp\Bigg\{ \Bigg(\underbrace{c + \frac{\sum_{i_1 i_2 i_3} \mathcal{O}_{i_1 i_2 i_3}}{2}}_{:=c'} - 1 \Bigg) \ln\alpha \\ &\quad - \Bigg(\underbrace{d + \frac{1}{2} \Bigg\| \mathcal{O} * \Big(\mathcal{Y} - \sum_{l=1}^{L} \Big(\boldsymbol{B}_{:,l}^{(1)} \circ \boldsymbol{B}_{:,l}^{(2)} \circ \boldsymbol{B}_{:,l}^{(3)}\Big)\Big) \Bigg\|_F^2}_{:=d'} \Bigg) \alpha \Bigg\}. \end{aligned} \tag{27}$$

Putting (27) into (13) gives

$$Q\left(\alpha\right) \propto \exp\left\{ \left(\mathbb{E}_{\boldsymbol{\Theta}\backslash \alpha}\left[c'\right] - 1\right) \ln\alpha - \mathbb{E}_{\boldsymbol{\Theta}\backslash \alpha}\left[d'\right] \alpha \right\}$$

$$\propto \text{Gamma}\Big(\underbrace{\mathbb{E}_{\boldsymbol{\Theta}\backslash\alpha}\left[c^{'}\right]}_{:=\hat{c}}, \underbrace{\mathbb{E}_{\boldsymbol{\Theta}\backslash\alpha}\left[d^{'}\right]}_{:=\hat{d}}\Big). \tag{28}$$

From (28), we obtain that the variational distribution $Q(\alpha)$ is Gamma distribution with parameters updated by

$$\hat{c} = c + \frac{\sum_{i_1 i_2 i_3} \mathcal{O}_{i_1 i_2 i_3}}{2},$$
$$\hat{d} = d + \frac{1}{2}\mathbb{E}_{\boldsymbol{\Theta}\backslash\alpha}\left[\left\|\mathcal{O} * \Big(\mathcal{Y} - \sum_{l=1}^{L}\Big(\boldsymbol{B}^{(1)}_{:,l} \circ \boldsymbol{B}^{(2)}_{:,l} \circ \boldsymbol{B}^{(3)}_{:,l}\Big)\Big)\right\|_F^2\right]. \tag{29}$$

In (29), the expectation of model residuals can be computed using the definition of CPD (the expectation is with respect to $\boldsymbol{\Theta}\backslash\alpha$, but this dependence is not explicitly stated in the expressions below to simplify the notations):

$$\begin{aligned}
&\mathbb{E}\left[\left\|\mathcal{O} * \Big(\mathcal{Y} - \sum_{l=1}^{L}\big(\boldsymbol{B}^{(1)}_{:,l} \circ \boldsymbol{B}^{(2)}_{:,l} \circ \boldsymbol{B}^{(3)}_{:,l}\big)\Big)\right\|_F^2\right] \\
&= \mathbb{E}\Big[\text{Tr}\Big\{\Big(\big(\boldsymbol{B}^{(1)}(\boldsymbol{B}^{(3)} \odot \boldsymbol{B}^{(2)})^T - \boldsymbol{Y}_{(1)}\big) * \boldsymbol{O}_{(1)}\Big) \\
&\qquad \cdot \Big(\boldsymbol{O}_{(1)}{}^T * \big((\boldsymbol{B}^{(3)} \odot \boldsymbol{B}^{(2)})\boldsymbol{B}^{(1)T} - (\boldsymbol{Y}_{(1)})^T\big)\Big)\Big\}\Big] \\
&= \text{Tr}\Big\{(\boldsymbol{Y}_{(1)} * \boldsymbol{O}_{(1)})(\boldsymbol{O}_{(1)} * \boldsymbol{Y}_{(1)})^T\Big\} \\
&\quad - 2\text{Tr}\Big\{(\boldsymbol{Y}_{(1)} * \boldsymbol{O}_{(1)})\Big(\Big(\mathbb{E}\left[\boldsymbol{B}^{(3)}\right] \odot \mathbb{E}\left[\boldsymbol{B}^{(2)}\right]\Big)\mathbb{E}\left[\boldsymbol{B}^{(1)}\right]^T\Big)\Big\} \\
&\quad + \text{Tr}\Big\{\mathbb{E}\Big[\Big(\Big(\boldsymbol{B}^{(1)}\big(\boldsymbol{B}^{(3)} \odot \boldsymbol{B}^{(2)}\big)^T\Big) * \boldsymbol{O}_{(1)}\Big) \\
&\qquad \cdot \Big(\boldsymbol{O}_{(1)}{}^T * \Big(\big(\boldsymbol{B}^{(3)} \odot \boldsymbol{B}^{(2)}\big)\boldsymbol{B}^{(1)T}\Big)\Big)\Big]\Big\},
\end{aligned} \tag{30}$$

where the last euqation is due to the conditional independence of $\boldsymbol{B}^{(k)}$ in variational distribution. Expressing $\boldsymbol{B}^{(k)}$ in terms of its columns, we can further show that (30) is given by

$$\begin{aligned}
&\text{Tr}\Big\{(\boldsymbol{Y}_{(1)} * \boldsymbol{O}_{(1)})(\boldsymbol{O}_{(1)} * \boldsymbol{Y}_{(1)})^T\Big\} \\
&- 2\text{Tr}\Big\{(\boldsymbol{Y}_{(1)} * \boldsymbol{O}_{(1)})\Big(\Big(\mathbb{E}\left[\boldsymbol{B}^{(3)}\right] \odot \mathbb{E}\left[\boldsymbol{B}^{(2)}\right]\Big)\mathbb{E}\left[\boldsymbol{B}^{(1)}\right]^T\Big)\Big\} \\
&+ \text{Tr}\Big\{\sum_{l=1}^{L}\sum_{p=1}^{L}\boldsymbol{T}^{(l,p)} * \mathbb{E}\left[\boldsymbol{B}^{(1)}_{:,l}\boldsymbol{B}^{(1)T}_{:,p}\right]\Big\}.
\end{aligned} \tag{31}$$

where $\boldsymbol{T}^{(l,p)} \in \mathbb{R}^{I_1 \times I_1}$, and $[\boldsymbol{T}^{(l,p)}]_{i,j} = ([\boldsymbol{O}_{(1)}]_{i,:} * [\boldsymbol{O}_{(1)}]_{j,:}) \cdot \bigodot_{h\neq 1}\mathbb{E}\left[\boldsymbol{B}^{(h)}_{:,l} * \boldsymbol{B}^{(h)}_{:,p}\right]$.

Due to $\text{Tr}\left\{\boldsymbol{T}^{(l,p)} * \mathbb{E}\left[\boldsymbol{B}^{(1)}_{:,l}\boldsymbol{B}^{(1)T}_{:,p}\right]\right\} = \sum_{i=1}^{I_1}\left([\boldsymbol{O}_{(1)}]_{i,:} \cdot \bigodot_{h\neq 1}\mathbb{E}\left[\boldsymbol{B}^{(h)}_{:,l} * \boldsymbol{B}^{(h)}_{:,p}\right]\right)\mathbb{E}\left[\boldsymbol{B}^{(1)}_{i,l}\boldsymbol{B}^{(1)}_{i,p}\right] = \mathbb{E}\left[\boldsymbol{B}^{(1)}_{:,l} * \boldsymbol{B}^{(1)}_{:,p}\right]^T \cdot \boldsymbol{O}_{(1)} \cdot \bigodot_{h\neq 1}\mathbb{E}\left[\boldsymbol{B}^{(h)}_{:,l} * \boldsymbol{B}^{(h)}_{:,p}\right]$, we further have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\mathcal{O} * \Big(\mathcal{Y} - \sum_{l=1}^{L}\Big(\boldsymbol{B}^{(1)}_{:,l} \circ \boldsymbol{B}^{(2)}_{:,l} \circ \boldsymbol{B}^{(3)}_{:,l}\Big)\Big)\right\|_F^2\right] \\
&= \text{Tr}\Big\{(\boldsymbol{Y}_{(1)} * \boldsymbol{O}_{(1)})(\boldsymbol{O}_{(1)} * \boldsymbol{Y}_{(1)})^T\Big\} \\
&-2\text{Tr}\Big\{(\boldsymbol{Y}_{(1)} * \boldsymbol{O}_{(1)})\Big(\Big(\mathbb{E}\left[\boldsymbol{B}^{(3)}\right] \odot \mathbb{E}\left[\boldsymbol{B}^{(2)}\right]\Big)\mathbb{E}\left[\boldsymbol{B}^{(1)}\right]^T\Big)\Big\} \\
&+\Big\{\sum_{l=1}^{L}\sum_{p=1}^{L}\mathbb{E}\left[\boldsymbol{B}^{(1)}_{:,l} * \boldsymbol{B}^{(1)}_{:,p}\right]^T \boldsymbol{O}_{(1)} \bigodot_{h\neq 1}\mathbb{E}\left[\boldsymbol{B}^{(h)}_{:,l} * \boldsymbol{B}^{(h)}_{:,p}\right]\Big\}.
\end{aligned} \tag{32}$$

Finally, with the expression of $\mathbb{E}\left[\boldsymbol{B}^{(h)}_{:,l} * \boldsymbol{B}^{(h)}_{:,p}\right]$ derived below (22), (32) is identical to (19).

Appendix B
Efficient algorithm under fully observed tensor

Expression for $Q(\boldsymbol{B}^{(k)}_{:,l})$:

Notice that the square matrix $\text{diag}\left(\boldsymbol{O}_{(k)}\bigodot_{h\neq k}\mathbb{E}\left[\boldsymbol{B}^{(h)}_{:,l} * \boldsymbol{B}^{(h)}_{:,l}\right]\right)$ in (22) is a diagonal matrix and the i^{th} diagonal element is $[\boldsymbol{O}_{(k)}]_{i,:}\bigodot_{h\neq k}\mathbb{E}\left[\boldsymbol{B}^{(h)}_{:,l} * \boldsymbol{B}^{(h)}_{:,l}\right]$, $i = 1, ..., I_k$. In this expression, since the dependence of i only appears in $\mathcal{O}$, if all the elements of $\mathcal{O}$ are one, the diagonal elements do not depend on i, and they all are the same and is given by $\prod_{h\neq k}\mathbb{E}\big[\boldsymbol{B}^{(h)T}_{:,l}\boldsymbol{B}^{(h)}_{:,l}\big] = \prod_{h\neq k}\mathbb{E}\big[\boldsymbol{B}^{(h)T}\boldsymbol{B}^{(h)}\big]_{l,l}$. Furthermore, for the vector $\boldsymbol{O}_{(k)}\bigodot_{h\neq k}\mathbb{E}\left[\boldsymbol{B}^{(h)}_{:,l} * \boldsymbol{B}^{(h)}_{:,p}\right] \in \mathbb{R}^{I_k \times 1}$ in (22), when the elements of $\mathcal{O}$ are all one, all the elements of this vector are the same and is given by $\prod_{h\neq k}\mathbb{E}\big[\boldsymbol{B}^{(h)T}\boldsymbol{B}^{(h)}\big]_{l,p}$. Therefore, under fully observed $\mathcal{Y}$, (22) could be simplified as

$$\begin{aligned}
\boldsymbol{\Upsilon}^{(k)^{-1}}_l &= \Big(\mathbb{E}\left[\boldsymbol{\Sigma}_k\right] + \mathbb{E}\left[\alpha\right]\Big(\prod_{h\neq k}\mathbb{E}\left[\boldsymbol{B}^{(h)T}\boldsymbol{B}^{(h)}\right]_{l,l}\Big)\boldsymbol{I}_{I_k}\Big)^{-1} \\
\boldsymbol{m}^{(k)}_l &= -\boldsymbol{\Upsilon}^{(k)^{-1}}_l\mathbb{E}\left[\alpha\right]\Big\{\sum_{p\neq l}\Big(\prod_{h\neq k}\mathbb{E}\left[\boldsymbol{B}^{(h)T}\boldsymbol{B}^{(h)}\right]_{l,p}\Big) \\
&\qquad \cdot \mathbb{E}\left[\boldsymbol{B}^{(k)}_{:,p}\right] - \left[\boldsymbol{Y}_{(k)}\left(\bigodot_{h\neq k}\mathbb{E}\left[\boldsymbol{B}^{(h)}_{:,l}\right]\right)\right]\Big\},
\end{aligned} \tag{33}$$

where $\bigodot_{h\neq k}\mathbb{E}\left[\boldsymbol{B}^{(h)}_{:,l}\right] = \mathbb{E}\left[\boldsymbol{B}^{(3)}_{:,l}\right] \odot \cdots \odot \mathbb{E}\left[\boldsymbol{B}^{(k+1)}_{:,l}\right] \odot \mathbb{E}\left[\boldsymbol{B}^{(k-1)}_{:,l}\right] \odot \cdots \odot \mathbb{E}\left[\boldsymbol{B}^{(1)}_{:,l}\right]$.

In order to compute (33), we already have $\mathbb{E}\left[\boldsymbol{B}^{(h)}_{:,p}\right] = \boldsymbol{m}^{(h)}_p$. Similarly, we also need the elements of the matrix: $\mathbb{E}\left[\boldsymbol{B}^{(h)T}\boldsymbol{B}^{(h)}\right]_{l,p}$. We could derive that $\mathbb{E}\left[\boldsymbol{B}^{(h)T}\boldsymbol{B}^{(h)}\right]_{l,p} = \mathbb{E}\left[\boldsymbol{B}^{(h)T}_{:,l}\boldsymbol{B}^{(h)}_{:,p}\right] = \boldsymbol{m}^{(h)T}_l\boldsymbol{m}^{(h)}_p$, if $l \neq p$. Furthermore, $\mathbb{E}\left[\boldsymbol{B}^{(h)T}\boldsymbol{B}^{(h)}\right]_{l,l} = \boldsymbol{m}^{(h)T}_l\boldsymbol{m}^{(h)}_l + \text{Tr}(\boldsymbol{\Upsilon}^{(h)^{-1}}_l)$. Putting these results into (33), we obtain

$$\begin{aligned}
\boldsymbol{\Upsilon}^{(k)^{-1}}_l &= \Big(\mathbb{E}\left[\boldsymbol{\Sigma}_k\right] + \mathbb{E}\left[\alpha\right] \\
&\qquad \cdot \prod_{h\neq k}\Big(\boldsymbol{m}^{(h)T}_l\boldsymbol{m}^{(h)}_l + \text{Tr}(\boldsymbol{\Upsilon}^{(h)^{-1}}_l)\Big)\boldsymbol{I}_{I_k}\Big)^{-1} \\
\boldsymbol{m}^{(k)}_l &= -\boldsymbol{\Upsilon}^{(k)^{-1}}_l\mathbb{E}\left[\alpha\right]\Big\{\sum_{p\neq l}\Big(\prod_{h\neq k}(\boldsymbol{m}^{(h)T}_l\boldsymbol{m}^{(h)}_p)\Big) \cdot \boldsymbol{m}^{(k)}_p
\end{aligned}$$

$$- \left[\boldsymbol{Y}_{(k)} \left(\bigodot_{h \neq k} \boldsymbol{m}_l^{(h)} \right) \right] \Bigg\}, \tag{34}$$

Expression for $Q(\alpha)$:

Under fully observed $\mathcal{Y}$, i.e., $\{\mathcal{O}_{i_1 i_2 i_3} = 1\}_{i_1=1, i_2=1, i_3=1}^{I_1, I_2, I_3}$, the expression $\sum_{i_1 i_2 i_3} \mathcal{O}_{i_1 i_2 i_3}$ in (29) equals $I_1 I_2 I_3$. Furthermore, the scalar $\mathbb{E}\left[\boldsymbol{B}_{:,l}^{(1)} * \boldsymbol{B}_{:,p}^{(1)}\right]^T \boldsymbol{O}_{(1)} \bigodot_{h \neq 1} \mathbb{E}\left[\boldsymbol{B}_{:,l}^{(h)} * \boldsymbol{B}_{:,p}^{(h)}\right]$ in (32) with elements of $\boldsymbol{O}_{(1)}$ being all one could be simplified as $\prod_{k=1}^{3} \mathbb{E}\left[\boldsymbol{B}^{(h)^T} \boldsymbol{B}^{(h)}\right]_{l,p}$. Therefore, (29) could be simplified as

$$\hat{c} = c + \frac{I_1 I_2 I_3}{2},$$
$$\hat{d} = d + \frac{1}{2}\mathbb{E}\left[\left\| \mathcal{Y} - \sum_{l=1}^{L} \left(\boldsymbol{B}_{:,l}^{(1)} \circ \boldsymbol{B}_{:,l}^{(2)} \circ \boldsymbol{B}_{:,l}^{(3)} \right) \right\|_F^2 \right] \tag{35}$$

with

$$\begin{aligned} &\mathbb{E}\left[\left\| \mathcal{Y} - \sum_{l=1}^{L} \left(\boldsymbol{B}_{:,l}^{(1)} \circ \boldsymbol{B}_{:,l}^{(2)} \circ \boldsymbol{B}_{:,l}^{(3)} \right) \right\|_F^2 \right] \\ &= \mathrm{Tr}\left\{ \boldsymbol{Y}_{(1)} (\boldsymbol{Y}_{(1)})^T \right\} - 2\mathrm{Tr}\left\{ \boldsymbol{Y}_{(1)} \left(\mathbb{E}\left[\boldsymbol{B}^{(3)}\right] \odot \mathbb{E}\left[\boldsymbol{B}^{(2)}\right] \right) \right. \\ &\quad \left. \cdot \mathbb{E}\left[\boldsymbol{B}^{(1)}\right]^T \right\} + \mathbf{1}_{L\times 1}{}^T \left\{ \circledast_{k=1,2,3} \mathbb{E}\left[\boldsymbol{B}^{(k)^T} \boldsymbol{B}^{(k)}\right] \right\} \mathbf{1}_{L\times 1}, \end{aligned} \tag{36}$$

where $\circledast$ is the Hadamard product among all the matrices $\left\{\mathbb{E}\left[\boldsymbol{B}^{(k)^T} \boldsymbol{B}^{(k)}\right]\right\}_{k=1}^{3}$. It is not difficult to obtain the results that $\mathbb{E}\left[\boldsymbol{B}^{(k)^T} \boldsymbol{B}^{(k)}\right] = \boldsymbol{M}^{(k)^T} \boldsymbol{M}^{(k)} + \mathrm{Diag}\left(\mathrm{Tr}(\boldsymbol{\Upsilon}_1^{(k)^{-1}}), \mathrm{Tr}(\boldsymbol{\Upsilon}_2^{(k)^{-1}}), ... \mathrm{Tr}(\boldsymbol{\Upsilon}_L^{(k)^{-1}})\right)$, where $\boldsymbol{M}^{(k)} = \left[\boldsymbol{m}_1^{(k)}, \boldsymbol{m}_2^{(k)}, ..., \boldsymbol{m}_L^{(k)}\right]$. Putting these results into (36), we obtain

$$\begin{aligned} &\mathbb{E}\left[\left\| \mathcal{Y} - \sum_{l=1}^{L} \left(\boldsymbol{B}_{:,l}^{(1)} \circ \boldsymbol{B}_{:,l}^{(2)} \circ \boldsymbol{B}_{:,l}^{(3)} \right) \right\|_F^2 \right] \\ &= \mathrm{Tr}\left\{ \boldsymbol{Y}_{(1)} (\boldsymbol{Y}_{(1)})^T \right\} - 2\mathrm{Tr}\left\{ \boldsymbol{Y}_{(1)} \left(\boldsymbol{M}^{(3)} \odot \boldsymbol{M}^{(2)} \right) \cdot \boldsymbol{M}^{(1)^T} \right\} \\ &\quad + \mathbf{1}_{L\times 1}{}^T \left\{ \circledast_{k=1,2,3} \left[\boldsymbol{M}^{(k)^T} \boldsymbol{M}^{(k)} \right. \right. \\ &\quad \left. \left. + \mathrm{Diag}\left(\mathrm{Tr}(\boldsymbol{\Upsilon}_1^{(k)^{-1}}), \mathrm{Tr}(\boldsymbol{\Upsilon}_2^{(k)^{-1}}), ... \mathrm{Tr}(\boldsymbol{\Upsilon}_L^{(k)^{-1}})\right) \right] \right\} \mathbf{1}_{L\times 1}. \end{aligned} \tag{37}$$

This gives the $\hat{c}$ and $\hat{d}$ in fully observed tensor case as

$$\begin{aligned} \hat{c} &= c + \frac{I_1 I_2 I_3}{2}, \\ \hat{d} &= d + \frac{1}{2}\mathrm{Tr}\left\{ \boldsymbol{Y}_{(1)} (\boldsymbol{Y}_{(1)})^T \right\} - \mathrm{Tr}\left\{ \boldsymbol{Y}_{(1)} \left(\boldsymbol{M}^{(3)} \odot \boldsymbol{M}^{(2)} \right) \right. \\ &\quad \left. \cdot \boldsymbol{M}^{(1)^T} \right\} + \frac{1}{2} \cdot \mathbf{1}_{L\times 1}{}^T \left\{ \circledast_{k=1,2,3} \left[\boldsymbol{M}^{(k)^T} \boldsymbol{M}^{(k)} \right. \right. \\ &\quad \left. \left. + \mathrm{Diag}\left(\mathrm{Tr}(\boldsymbol{\Upsilon}_1^{(k)^{-1}}), \mathrm{Tr}(\boldsymbol{\Upsilon}_2^{(k)^{-1}}), ... \mathrm{Tr}(\boldsymbol{\Upsilon}_L^{(k)^{-1}})\right) \right] \right\} \mathbf{1}_{L\times 1} \end{aligned} \tag{38}$$

Summary and Complexity analysis

Under fully observed $\mathcal{Y}$, the Algorithm 1 is modified by replacing (15) with (34), and (19) with (38). In this case, the computational complexity of the factor matrices $\{\boldsymbol{B}^{(k)}\}_{k=1}^{3}$, the hyperparameters $\{\boldsymbol{\Sigma}^{(k)}\}_{k=1}^{3}$, and the noise precision α are $O\left(\sum_{k=1}^{3} {I_k}^3 + \prod_{k=1}^{3} I_k\right)$, $O\left(\sum_{k=1}^{3} {I_k}^3\right)$ and $O\left(\prod_{k=1}^{3} I_k\right)$, respectively. Thus the overall computational complexity is $O\left(\sum_{k=1}^{3} {I_k}^3 + \prod_{k=1}^{3} I_k\right)$.

Xueke Tong received the B.Eng. and M.Phil. degrees from South China University of Technology, Guangzhou, China, and the Ph.D. degree from the Department of Electrical and Electronic Engineering, The University of Hong Kong, in 2023. Her research interests include signal processing, regularization analysis in optimization algorithms for tensor decomposition, and Bayesian machine learning for tensor decomposition.

Lei Cheng is currently Assistant Professor (ZJU Young Professor) with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. He received the B.Eng. degree from Zhejiang University in 2013, and the Ph.D. degree from The University of Hong Kong in 2018. He was a Research Scientist in Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong, Shenzhen, from 2018 to 2021. He is the co-author of the book "Bayesian Tensor Decomposition for Signal Processing and Machine Learning: Modeling, Tuning-Free Algorithms, and Applications", Springer, 2023. He was a Tutorial Speaker in ICASSP 2023. His research interests are in Bayesian machine learning for tensor data analytics, and interpretable machine learning for information systems.

Yik-Chung Wu received the B.Eng. (EEE) and M.Phil. degrees from The University of Hong Kong (HKU), in 1998 and 2001, respectively, and the Ph.D. degree from Texas AM University, College Station, in 2005. From 2005 to 2006, he was with Thomson Corporate Research, Princeton, NJ, as a member of the Technical Staff. Since 2006, he has been with HKU, currently as an Associate Professor. He was a Visiting Scholar at Princeton University in 2015 and 2017. His research interests include general areas of signal processing, machine learning, and communication systems. He served as an Editor for the IEEE COMMUNICATIONS LETTERS and the IEEE TRANSACTIONS ON COMMUNICATIONS. He is currently a Senior Area Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING, an Associate Editor for IEEE WIRELESS COMMUNICATION LETTERS, and an Editor of Journal of Communications and Networks.