Bayesian Low-Tubal-Rank Robust Tensor Factorization with Multi-Rank Determination

Yang Zhou and Yiu-Ming Cheung, *Fellow, IEEE*

Abstract—Robust tensor factorization is a fundamental problem in machine learning and computer vision, which aims at decomposing tensors into low-rank and sparse components. However, existing methods either suffer from limited modeling power in preserving low-rank structures, or have difficulties in determining the target tensor rank and the trade-off between the low-rank and sparse components. To address these problems, we propose a fully Bayesian treatment of robust tensor factorization along with a generalized sparsity-inducing prior. By adapting the recently proposed low-tubal-rank model in a generative manner, our method is effective in preserving low-rank structures. Moreover, benefiting from the proposed prior and the Bayesian framework, the proposed method can automatically determine the tensor rank while inferring the trade-off between the low-rank and sparse components. For model estimation, we develop a variational inference algorithm, and further improve its efficiency by reformulating the variational updates in the frequency domain. Experimental results on both synthetic and real-world datasets demonstrate the effectiveness of the proposed method in multi-rank determination as well as its superiority in image denoising and background modeling over state-of-the-art approaches.

Index Terms—Robust PCA, tensor factorization, tubal rank, multi-rank determination, Bayesian inference

1 Introduction

Real-world data such as images, videos, and social networks are often high-dimensional, while considered to be approximately low-rank or lie near a low-dimensional manifold. Finding and exploiting low-rank structures from high-dimensional data is a fundamental problem in many machine learning and computer vision applications, e.g., collaborative filtering [1], face recognition [2], and data mining [3]. Principal Component Analysis (PCA) [4] is a conventional method to seek the best (in the least-squares sense) low-rank representation of given data. It is effective in dealing with the data that is mildly corrupted with small noise, and can be stably computed via *singular value decomposition* (SVD).

However, PCA is very sensitive to outliers, and fails to perform well on data with gross corruptions. Unfortunately, the presence of outliers is ubiquitous in real-world applications such as data mining, image processing, and video surveillance. For instance, moving objects in a video taken by a stationary camera can be viewed as sparse outliers in the static background. To overcome the sensitivity of PCA to outliers, many robust variants of PCA have been proposed [5], [6], [7], [8]. Among them, Robust PCA (RPCA) [6] is arguably the most popular method that enjoys both computational efficiency and theoretical performance guarantees.

RPCA assumes that the observed matrix $\mathbf{Y}$ can be represented as $\mathbf{Y} = \mathbf{X}_0 + \mathbf{S}_0$, where $\mathbf{X}_0$ is a low-rank matrix and $\mathbf{S}_0$ is a sparse matrix with only a small fraction of elements being nonzero and arbitrary in magnitude. It has been proved that, under some broad conditions, $\mathbf{X}_0$ and $\mathbf{S}_0$ can be exactly recovered from $\mathbf{Y}$ by solving the following convex problem:

$$\min_{\mathbf{X},\mathbf{S}} \|\mathbf{X}\|_* + \xi\|\mathbf{S}\|_1 \quad \text{s.t.} \quad \mathbf{Y} = \mathbf{X} + \mathbf{S}, \tag{1}$$

where $\|\cdot\|_*$ and $\|\cdot\|_1$ denote the nuclear norm and ℓ_1 norm, respectively, and $\xi > 0$ is the hyper-parameter balancing the low-rank and sparse terms. RPCA and its extensions have many important applications, such as video denoising [9], subspace clustering [10], and object detection [11], to name a few.

One main limitation of RPCA is that it can only deal with matrix data, while many real-world data are naturally organized as tensors (multidimensional arrays) [12], [13]. For example, a color image is a third-order tensor of $height \times width \times channel$, and a gray-level video can be represented as $height \times width \times time$. When applying RPCA to tensorial data, one has to first reshape the input tensor into a matrix, which often leads to loss of structural information and degraded performance. To address this problem, tensor RPCA (TRPCA) and robust tensor factorization (RTF) methods have been proposed, which directly handle tensors for exploiting their multidimensional structures.

Specifically, given a tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, TRPCA and RTF methods assume $\mathcal{Y} = \mathcal{X}_0 + \mathcal{S}_0$ and seek to recover $\mathcal{X}_0$ from $\mathcal{Y}$, where $\mathcal{X}_0$ is a tensor with certain low-rank structure

• *Y. Zhou is with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong, and also with the School of Computer Science and Software Engineering, East China Normal University, Shanghai 200062, P.R. China. E-mail: youngzhou12@gmail.com.*
• *Y.-M. Cheung is with the Department of Computer Science and Institute of Research and Continuing Education, Hong Kong Baptist University, Kowloon Tong, Hong Kong. E-mail: ymc@comp.hkbu.edu.hk.*

Manuscript received 5 Oct. 2018; revised 6 June 2019; accepted 9 June 2019. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Yiu-Ming Cheung.)
Recommended for acceptance by C. Zhang.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2019.2923240

and $\mathcal{S}_0$ is sparse. Based on different low-rank models and the corresponding tensor rank definitions, there exist three popular frameworks for solving the TRPCA and RTF problems. They are based on the Tucker [14], CANDECOMP/PARAFAC (CP) [15], [16], and low-tubal-rank models [17], [18], respectively.

The Tucker model assumes that the low-rank component $\mathcal{X}_0$ can be well approximated as

$$\mathcal{X}_{\rm tc} = \mathcal{Z} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \cdots \times_N \mathbf{U}^{(N)}, \tag{2}$$

where $\times_n$ denotes the mode-n tensor product, $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ $(n = 1, \ldots, N)$ is the mode-n factor matrix, $\mathcal{Z}$ is the core tensor capturing the correlations among $\{\mathbf{U}^{(n)}\}_{n=1}^N$. The Tucker (multilinear) rank [12] of $\mathcal{Y}$ is defined as $Rank_{\rm tc}(\mathcal{Y}) \equiv (R_1, \ldots, R_N)$ with $R_n = Rank(\mathbf{Y}_{(n)})$, where $\mathbf{Y}_{(n)} \in \mathbb{R}^{I_n \times \prod_{m \neq n} I_m}$ is the mode-n unfolding matrix of $\mathcal{Y}$.

Most Tucker-based TRPCA methods [19], [20] are convex methods. They seek a low-Tucker-rank component by minimizing the Sum of Nuclear Norms (SNN) [21] of $\mathcal{Y}$, which is a convex surrogate of the Tucker rank. Some robust Tucker factorization methods [22], [23], [24] have also been proposed to perform TRPCA by explicitly fitting the Tucker model with a *predetermined* Tucker rank. By alternately solving a (nonconvex) least-squares problem, such RTF methods are generally more efficient and empirically perform better than convex TRPCA approaches, provided that the predetermined Tucker rank matches the input tensor. However, Tucker-based TRPCAs and RTFs require unfolding the input tensor for parameter estimation, and thus fail to fully exploit the correlations among different tensor dimension [19], [25].

The CP model decomposes $\mathcal{X}_0$ into the sum of rank-one tensors as follows:

$$\mathcal{X}_{\rm cp} = \sum_{r=1}^{R} \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \cdots \circ \mathbf{u}_r^{(N)}, \tag{3}$$

where $\circ$ denotes the outer product, and $\mathbf{u}_r^{(n)} \in \mathbb{R}^{I_n}$ $(n = 1, \ldots, N; r = 1, \ldots, R)$ is the rth mode-n factor. The CP rank of $\mathcal{Y}$ is given by $Rank_{\rm cp}(\mathcal{Y}) \equiv R$, defined as the smallest number of the rank-one tensor decomposition [12].

Since the CP rank is difficult to be determined (known as an NP-hard problem) and its convex relaxation is intractable [26], [27], existing CP-based TRPCA and RTF methods resort to the probabilistic framework to estimate the low-rank component and the CP rank. For example, Bayesian Robust Tensor Factorization (BRTF) [28] estimates the CP model in a fully Bayesian manner to recover tensors with both missing values and outliers. By introducing proper priors, it obtains robustness against overfitting and enables automatic CP rank determination. To handle complex noise and outliers, Generalized Weighted Low-Rank Tensor Factorization (GWLRTF) [29] represents the sparse component $\mathcal{S}$ as a mixture of Gaussian, and unifies the Tucker and CP factorization in a joint framework. A key advantage of these probabilistic RTF methods over their non-probabilistic counterparts is that the trade-off between the low-rank and sparse components can be naturally optimized without manually tuning. Nevertheless, the CP model is usually considered as a special case of the Tucker model [12], and may not have enough flexibility in representing tensors with complex low-rank structures.

Recently, Kilmer et al. [17] defined a multiplication operation between tensors, called tensor-tensor product (t-product), and proposed tensor-SVD (t-SVD) associated with two new tensor rank definitions, i.e., tubal rank and multi-rank [18] (see Section 2 for their formal definitions). The *reduced version* [30] of t-SVD for the low-rank component $\mathcal{X}_0$ is given by

$$\mathcal{X}_{\rm t-SVD} = \mathcal{U} * \mathcal{D} * \mathcal{V}^{\dagger}, \tag{4}$$

where $*$ denotes the t-product, $\mathcal{U} \in \mathbb{R}^{I_1 \times R \times I_3}$ and $\mathcal{V} \in \mathbb{R}^{I_2 \times R \times I_3}$ are orthogonal tensors, and $\mathcal{D} \in \mathbb{R}^{R \times R \times I_3}$ is an f-diagonal tensor whose frontal slices are all diagonal matrices. The tubal rank of $\mathcal{X}_0$ is then defined by $Rank_{\rm t}(\mathcal{X}_0) \equiv R$.

The development of t-SVD motivates the low-tubal-rank model for representing tensors of low tubal rank, which has been successfully applied to the tensor completion problem with the state-of-the-art performance achieved [31], [32], [33]. Compared with the conventional Tucker and CP models, the low-tubal-rank model has more expressive modeling power, especially for characterizing tensors that have a fixed orientation or certain "spatial-shifting" properties, such as color images, videos, and multi-channel audio sequences [17].

Based on the low-tubal-rank model, Lu et al. [34], [35] proposed to use the tensor nuclear norm (TNN) [31] as a convex relaxation of the tubal rank, and perform TRPCA by solving a convex problem similar to RPCA (1). They further analyzed the theoretical guarantee for the exact recovery. Outlier-Robust Tensor PCA (OR-TRPCA) combines TNN with the $\ell_{2,1}$ norm to handle sample-specific corruptions, which achieves promising results on outliers detection and classification. However, similar to RPCA, these methods also involve a hyper-parameter as in (1) for adjusting the contributions of the low-rank and sparse components. For good performance, this balancing parameter has to be carefully determined. If the low-rank component contributes too much to the objective function, the outliers will not be completely removed. On the other hand, if the sparse component is dominant, the recovered tensor will lose many details and cannot fully preserve the low-rank structures. Since the trade-off between the low-rank and sparse components should be adjusted according to both the input data and tasks, finding an appropriate value for the balancing parameter is generally difficult and time consuming in practice.

Besides TNN, low-tubal-rank structures can also be introduced by explicitly factorizing a given tensor as the t-product of two smaller tensors [30], [33]. Such low-tubal-rank tensor factorization methods are more efficient and expected to obtain better recovery performance than TNN-based methods. However, in addition to the balancing parameter, they also need to know the target tubal rank in advance. Both over- and under-estimation of the tubal rank will lead to the degraded performance. Although a heuristic rank-decreasing strategy has been proposed in [33], the study on how to discover the underlying tubal rank and multi-rank of a given tensor is still very desirable.

Can we make use of the low-tubal-rank model for RTF without suffering from the difficulties in determining the tubal rank and the balancing parameter? In this paper, we solve this problem

TABLE 1
Convention of Notations

Notation	Description
$\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$	the $I_1 \times I_2 \times I_3$ tensor
$\bar{\mathcal{X}}$	the DFT of $\mathcal{X}$ along the third-dimension
$\overrightarrow{\mathcal{X}}_{i\cdot} \in \mathbb{R}^{1 \times I_2 \times I_3}$	the ith horizontal slice of $\mathcal{X}$
$\overrightarrow{\mathcal{X}}_{\cdot j} \in \mathbb{R}^{I_1 \times 1 \times I_3}$	the jth lateral slice of $\mathcal{X}$
$\mathbf{X}^{(k)} \in \mathbb{R}^{I_1 \times I_2}$	the kth frontal slice of $\mathcal{X}$
$\mathrm{circ}(\mathcal{X}) \in \mathbb{R}^{I_1 I_3 \times I_2 I_3}$	the block circulant matrix of $\mathcal{X}$
$\mathrm{unfold}(\mathcal{X}) \in \mathbb{R}^{I_1 I_3 \times I_2}$	the unfolded matrix of $\mathcal{X}$
$\mathcal{X}^\dagger \in \mathbb{R}^{I_2 \times I_1 \times I_3}$	the conjugate transpose of $\mathcal{X}$
$\mathbf{x}_{ij} \in \mathbb{R}^{I_3}$	the (i,j)th tube of $\mathcal{X}$
$\overrightarrow{\mathbf{x}}_{i\cdot} = \mathrm{unfold}(\overrightarrow{\mathcal{X}}_{i\cdot}^\dagger) \in \mathbb{R}^{I_2 I_3}$	the vector formed by unfolding $\overrightarrow{\mathcal{X}}_{i\cdot}^\dagger$
$\overrightarrow{\mathbf{x}}_{\cdot j} = \mathrm{unfold}(\overrightarrow{\mathcal{X}}_{\cdot j}) \in \mathbb{R}^{I_1 I_3}$	the vector formed by unfolding $\overrightarrow{\mathcal{X}}_{\cdot j}$
$*$	the t-product
$\circ$	the outer product
$\otimes$	the Kronecker product

by introducing low-tubal-rank structures into the Bayesian framework, and propose a fully Bayesian treatment of RTF for third-order tensors, named as **B**ayesian low-**T**ubal-rank **R**obust **T**ensor **F**actorization (BTRTF). To the best of our knowledge, this is the first probabilistic/Bayesian method for low-tubal-rank tensor factorization.

BTRTF equips the low-tubal-rank model with automatic rank determination, and enables implicit trade-off between the low-rank and sparse components via maximizing the (approximated) posterior probability. In addition, it is well known that the Bayesian framework offers unique advantages in capturing data uncertainty, reducing risk of over-fitting, handling missing values, and introducing prior knowledge. These benefits also motivate the development of our BTRTF method. In summary, our contribution is three-fold:

1) We propose a generative model for recovering low-tubal-rank tensors from observations corrupted by both sparse outliers of arbitrary magnitude and dense noise of small magnitude, where the observed tensor is factorized into the t-product of two smaller factor tensors.
2) We consider automatic rank determination for not only the tubal rank but also the multi-rank, which is a more general and challenging problem. To this end, we propose a generalization of the ARD prior [36]. By incorporating this prior into the Bayesian framework, unnecessary low-rank components can be adaptively removed in the frequency domain, leading to automatic multi-rank determination.
3) Since exact inference of the proposed generative model is analytically intractable, we develop an efficient model estimation scheme via variational approximation. By updating the model parameters in the frequency domain instead of the original one, the computational cost of each iteration is greatly reduced from $O(R^3 I_3^3 + R I_1 I_2 I_3^2)$ to $O(R^3 I_3 + R I_1 I_2 I_3)$, when handling a $I_1 \times I_2 \times I_3$ tensor with its tubal rank being R.

2 Preliminaries

This section introduces notations, definitions, and operations used in this paper.

2.1 Notations

We denote vectors, matrices, and tensors by bold lowercase, bold uppercase, and calligraphic letters ($\mathbf{x}$, $\mathbf{X}$, and $\mathcal{X}$), respectively. $\mathbb{R}$ and $\mathbb{C}$ denote the fields of real numbers and complex numbers, respectively. $\langle\cdot\rangle$ denotes the expectation of a certain random variable, $\mathrm{tr}(\cdot)$ denotes the matrix trace, and $\mathbf{I}_I$ denotes the $I \times I$ identity matrix. For a vector $\mathbf{x}$, $\mathrm{diag}(\mathbf{x})$ is the diagonal matrix formed by $\mathbf{x}$. For a third-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, we use the matlab notations to denote the ith horizontal, jth lateral, and kth frontal slices of $\mathcal{X}$ by $\overrightarrow{\mathcal{X}}_{i\cdot} = \mathcal{X}(i,:,:)$, $\overrightarrow{\mathcal{X}}_{\cdot j} = \mathcal{X}(:,j,:)$, and $\mathbf{X}^{(k)} = \mathcal{X}(:,:,k)$, respectively. $\mathbf{x}_{ij} = \mathcal{X}(i,j,:)$ denotes the (i,j)th tube of $\mathcal{X}$. The conjugate transpose and the Frobenius norm of $\mathcal{X}$ are denoted as $\mathcal{X}^\dagger$ and $\|\mathcal{X}\|_F$, respectively. $\mathrm{cir}(\mathcal{X}) \in \mathbb{R}^{I_1 I_3 \times I_2 I_3}$ is the block circulant matrix of $\mathcal{X}$, $\mathrm{unfold}(\mathcal{X}) \in \mathbb{R}^{I_1 I_3 \times I_2}$ is the unfolded matrix of $\mathcal{X}$, $\overrightarrow{\mathbf{x}}_{i\cdot} \in \mathbb{R}^{I_2 I_3}$ is the unfolded vector of $\overrightarrow{\mathcal{X}}_{i\cdot}^\dagger$ with $\overrightarrow{\mathbf{x}}_{i\cdot} = \mathrm{unfold}(\overrightarrow{\mathcal{X}}_{i\cdot}^\dagger)$, and $\overrightarrow{\mathbf{x}}_{\cdot j} \in \mathbb{R}^{I_1 I_3}$ is the unfolded vector of $\overrightarrow{\mathcal{X}}_{\cdot j}$ with $\overrightarrow{\mathbf{x}}_{\cdot j} = \mathrm{unfold}(\overrightarrow{\mathcal{X}}_{\cdot j})$. Table 1 summarizes the notations used in this paper.

2.2 Discrete Fourier Transformation

This subsection introduces Discrete Fourier Transformation (DFT), which plays a key role in the t-product algebraic framework and our BTRTF method. Let $\bar{\mathbf{x}} = \mathbf{F}_I \mathbf{x}$ be the DFT of $\mathbf{x} \in \mathbb{R}^I$. $\mathbf{F}_I \in \mathbb{C}^{I \times I}$ is the DFT matrix defined as

$$\mathbf{F}_I = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{I-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{I-1} & \omega^{2(I-1)} & \cdots & \omega^{(I-1)(I-1)} \end{bmatrix}, \tag{5}$$

where $\omega = \exp(-\frac{2\pi \mathrm{i}}{I})$ and $\mathrm{i} = \sqrt{-1}$ is the imaginary unit. Let $\bar{\mathcal{X}}$ be the DFT of $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ along the *third* dimension, whose (i,j)th tube is given by $\bar{\mathbf{x}}_{ij} = \bar{\mathcal{X}}(i,j,:) = \mathbf{F}_{I_3}\mathcal{X}(i,j,:)$. Using the matlab commands, we have $\bar{\mathcal{X}} = \mathrm{fft}(\mathcal{X}, [], 3)$ and $\mathcal{X} = \mathrm{ifft}(\bar{\mathcal{X}}, [], 3)$ by applying (inverse) Fast Fourier Transform (FFT).

Let $\bar{\mathbf{X}} \in \mathbb{C}^{I_1 I_3 \times I_2 I_3}$ be the block diagonal matrix whose kth diagonal block is given by the kth frontal slice $\bar{\mathbf{X}}^{(k)}$ of $\bar{\mathcal{X}}$, that is

$$\bar{\mathbf{X}} = \mathrm{bdiag}(\bar{\mathcal{X}}) = \begin{bmatrix} \bar{\mathbf{X}}^{(1)} & & & \\ & \bar{\mathbf{X}}^{(2)} & & \\ & & \ddots & \\ & & & \bar{\mathbf{X}}^{(I_3)} \end{bmatrix}, \tag{6}$$

where $\mathrm{bdiag}(\cdot)$ is the operator that transforms $\bar{\mathcal{X}}$ to $\bar{\mathbf{X}}$. We then define $\mathrm{circ}(\mathcal{X}) \in \mathbb{R}^{I_1 I_3 \times I_2 I_3}$ as the block circulant matrix of $\mathcal{X}$ as follows:

$$\mathrm{circ}(\mathcal{X}) = \begin{bmatrix} \mathbf{X}^{(1)} & \mathbf{X}^{(I_3)} & \cdots & \mathbf{X}^{(2)} \\ \mathbf{X}^{(2)} & \mathbf{X}^{(1)} & \cdots & \mathbf{X}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{X}^{(I_3)} & \mathbf{X}^{(I_3-1)} & \cdots & \mathbf{X}^{(1)} \end{bmatrix}. \tag{7}$$

It is well known that block circulant matrices can be block diagonalized by DFT, i.e.,

$$(\mathbf{F}_{I_3} \otimes \mathbf{I}_{I_1})\mathrm{circ}(\mathcal{X})(\mathbf{F}_{I_3}^{-1} \otimes \mathbf{I}_{I_2}) = \bar{\mathbf{X}}, \tag{8}$$

where $\otimes$ denotes the Kronecker product. The above operators and properties will be frequently used in this paper.

2.3 T-Product and T-SVD

This subsection introduces the t-product and its associated algebraic framework [18], which lay the foundation of our BTRTF. Let $\mathrm{unfold}(\cdot)$ and $\mathrm{fold}(\cdot)$ be the unfold operator and its inverse operator, respectively. For a third-order tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times I_3}$, $\mathrm{unfold}(\mathcal{X})$ is the $I_1I_3 \times I_2$ matrix formed by the frontal slices of $\mathcal{X}$, leading to

$$\mathrm{unfold}(\mathcal{X}) = [\mathbf{X}^{(1)}; \ldots ; \mathbf{X}^{(I_3)}], \mathrm{fold}(\mathrm{unfold}(\mathcal{X})) = \mathcal{X}.$$

Definition 2.1 (T-product [18]). *Given $\mathcal{X} \in \mathbb{R}^{I_1\times R\times I_3}$ and $\mathcal{Y} \in \mathbb{R}^{R\times I_2\times I_3}$, the t-product $\mathcal{X} * \mathcal{Y}$ is the $I_1 \times I_2 \times I_3$ tensor*

$$\mathcal{Z} = \mathcal{X} * \mathcal{Y} = \mathrm{fold}(\mathrm{circ}(\mathcal{X})\mathrm{fold}(\mathcal{Y})). \tag{9}$$

The computation of t-product can also be viewed in a tube-wise way

$$\mathbf{z}_{ij} = \mathcal{Z}(i, j, :) = \sum_{r=1}^{R} \mathbf{x}_{ir} * \mathbf{y}_{rj}, \tag{10}$$

where $\mathbf{x}_{ir}$ is the (i,r)th tube of $\mathcal{X}$, $\mathbf{y}_{rj}$ is the (r,j)th tube of $\mathcal{Y}$, and $*$ reduces to the circular convolution between two tubes of the same size. If we consider the tube $\mathbf{z}_{ij} \in \mathbb{R}^{I_3}$ as an "elementary" component, the third-order tensor $\mathcal{Z} \in \mathbb{R}^{I_1\times I_2\times I_3}$ is just a $I_1 \times I_2$ matrix of length-I_3 tubal scalars. From this perspective, the t-product is analogous to the standard matrix multiplication in the sense that the circular convolution of tubes replaces the product of elements.

Remarks. It is also worth noting that when $I_3 = 1$ the t-product reduces to the matrix multiplication. Moreover, the t-product can be viewed as the matrix multiplication in the *Fourier domain*, since $\mathcal{Z} = \mathcal{X} * \mathcal{Y}$ is equivalent to $\bar{\mathbf{Z}} = \bar{\mathbf{X}}\bar{\mathbf{Y}}$ because of (8). This is a key property which provides an efficient way of computing the t-product and greatly facilitates the model estimation of our BTRTF method shown later. In what follows, we further review some definitions related to the t-product.

Definition 2.2 (Identity tensor [17]). *The identity tensor $\mathcal{I} \in \mathbb{R}^{I\times I\times I_3}$ is defined as the tensor whose first frontal slice is the $I \times I$ identity matrix, and other slices are all zeros.*

The identity tensor with appropriate sizes satisfies $\mathcal{X} * \mathcal{I}$ and $\mathcal{I} * \mathcal{X}$. The DFT of $\mathcal{I}$, $\bar{\mathcal{I}} = \mathrm{fft}(\mathcal{I}, [], 3)$, is the tensor with each frontal slice being the identity matrix.

Definition 2.3 (F-diagonal tensor [17]). *A tensor is called f-diagonal if its frontal slices are all diagonal matrices.*

Definition 2.4 (Conjugate transpose [17]). *The conjugate transpose of a tensor is defined as the tensor $\mathcal{X}^\dagger \in \mathbb{R}^{I_2\times I_1\times I_3}$ constructed by conjugate transposing each frontal slice of $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times I_3}$ and then reversing the order of the transposed frontal slices 2 through I_3.*

Definition 2.5 (Orthogonal tensor [17]). *A tensor $\mathcal{Q} \in \mathbb{Q}^{I\times I\times I_3}$ is called orthogonal, provided that $\mathcal{Q}^\dagger * \mathcal{Q} = \mathcal{Q} * \mathcal{Q}^\dagger = \mathcal{I}$ with $\mathcal{I}$ being an $I \times I \times I_3$ identity tensor.*

Definition 2.6 (T-SVD [17]). *Let $\mathcal{X}$ be an $I_1 \times I_2 \times I_3$ real-valued tensor. Then $\mathcal{X}$ can be factored as*

$$\mathcal{X} = \mathcal{U} * \mathcal{D} * \mathcal{V}^\dagger, \tag{11}$$

where $\mathcal{U} \in \mathbb{R}^{I_1\times I_1\times I_3}$, $\mathcal{V} \in \mathbb{R}^{I_2\times I_2\times I_3}$ are orthogonal tensors, and $\mathcal{D} \in \mathbb{R}^{I_1\times I_2\times I_3}$ is an f-diagonal tensor. The factorization (11) is called the t-SVD (i.e., tensor SVD).

The t-SVD provides a way to factorizing any third-order tensor into two orthogonal tensors and a f-diagonal tensor. When the third dimension $I_3 = 1$, it reduces to the classical matrix SVD.

Definition 2.7 (Tensor tubal rank and multi-rank [18]). *The multi-rank of a third-order tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times I_3}$ is a length-I_3 vector defined as*

$$Rank_{\mathrm{m}}(\mathcal{X}) = (Rank(\bar{\mathbf{X}}^{(1)}), \ldots, Rank(\bar{\mathbf{X}}^{(I_3)})),$$

where $\bar{\mathbf{X}}^{(k)}$ is the kth frontal slice of $\bar{\mathcal{X}} = \mathrm{fft}(\mathcal{X}, [], 3)$ and $Rank(\bar{\mathbf{X}}^{(k)})$ is the rank of $\bar{\mathbf{X}}^{(k)}$. The tubal rank of $\mathcal{X}$ is the number of nonzero tubes of $\mathcal{D}$ from the t-SVD of $\mathcal{X} = \mathcal{U} * \mathcal{D} * \mathcal{V}^\dagger$, i.e.,

$$Rank_{\mathrm{t}}(\mathcal{X}) = \#\{i, \mathcal{D}(i, i, :) \neq \mathbf{0}\} = \max_k Rank(\bar{\mathbf{X}}^{(k)}).$$

Lemma 1 (Best rank-R approximation [17], [18]). *Let $\mathcal{X} = \mathcal{U} * \mathcal{D} * \mathcal{V}^\dagger$ be the t-SVD of $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times I_3}$. Then given tubal rank $R < \min(I_1, I_2)$*

$$\begin{aligned}\mathcal{X}_R &= \arg\min_{\hat{\mathcal{X}}\in\mathbb{M}} \|\mathcal{X} - \hat{\mathcal{X}}\|_F \\ &= \sum_{r=1}^{R} \mathcal{U}(:, r, :) * \mathcal{D}(r, r, :) * \mathcal{V}(:, r, :)^\dagger,\end{aligned}$$

is the best approximation of $\mathcal{X}$ with the tubal rank at most R, where $\mathbb{M} = \{\mathcal{C} = \mathcal{A} * \mathcal{B}^\dagger | \mathcal{A} \in \mathbb{R}^{I_1\times R\times I_3}, \mathcal{B} \in \mathbb{R}^{I_2\times R\times I_3}\}$.

3 Bayesian Low-Tubal-Rank Robust Tensor Factorization

This section presents our BTRTF method in three steps. We first provide the detailed Bayesian model specification for BTRTF, and employ the Automatic Relevance Determination (ARD) prior [36] for tubal rank determination. Then we develop a variational inference method for model estimation, and further improve its efficiency by using the properties of the t-product and reformulating the variational updates in the frequency domain. Finally, a generalization of the ARD prior is proposed and incorporated into the BTRTF model to automatically determine both the tubal rank and multi-rank.

3.1 Model Specification

We assume that the observed tensor $\mathcal{Y} \in \mathbb{R}^{I_1\times I_2\times I_3}$ can be decomposed into three parts: the low-rank component $\mathcal{X}$, the sparse component $\mathcal{S}$, and the noise term $\mathcal{E}$, i.e.,

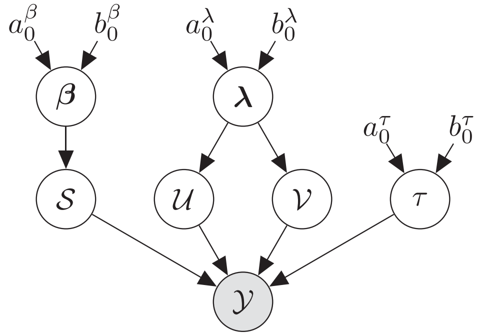


Fig. 1. Graphical illustration of the BTRTF model.

$$\mathcal{Y} = \mathcal{X} + \mathcal{S} + \mathcal{E}, \tag{12}$$

where each element of $\mathcal{E}$ is assumed to be i.i.d Gaussian, leading to $\mathcal{E} \sim \prod_{ijk} \mathcal{N}(E_{ijk}|0, \tau^{-1})$ with the noise precision τ. Given $\mathcal{Y}$, our goal is to recover $\mathcal{X}$ and $\mathcal{S}$. Different from most existing works pursuit $\mathcal{X}$ of low Tucker or CP rank, we preserve the *low-tubal-rank* structure of $\mathcal{X}$ by factorizing it as a t-product of two smaller factor tensors

$$\mathcal{X} = \mathcal{U} * \mathcal{V}^\dagger, \tag{13}$$

where $\mathcal{U} \in \mathbb{R}^{I_1 \times R \times I_3}$, $\mathcal{V} \in \mathbb{R}^{I_1 \times R \times I_3}$, and $R \leq \min(I_1, I_2)$ controls the tubal-rank. According to Lemma 1, any tensor with a tubal rank up to R can be factorized as (13) for some $\mathcal{U}$ and $\mathcal{V}$ satisfying $Rank_t(\mathcal{U}) = Rank_t(\mathcal{V}) = R$ [30], [33]. This means that the low-tubal-rank model (13) is flexible enough to provide good approximation for tensors of low tubal rank.

Conditional Distribution. Based on the above low-tubal-rank factorization, we can obtain the conditional distribution of the observed tensor $\mathcal{Y}$ given the model parameters, which is factorized over each tube of $\mathcal{Y}$ as follows:

$$p(\mathcal{Y}|\mathcal{U}, \mathcal{V}, \mathcal{S}, \tau) = \prod_{ij} \mathcal{N}(\mathbf{y}_{ij} | \overrightarrow{\mathcal{U}}_{i\cdot} * \overrightarrow{\mathcal{V}}_{j\cdot}^\dagger + \mathbf{s}_{ij}, \tau^{-1}\mathbf{I}_{I_3}). \tag{14}$$

Sparse Component. We model the sparse component $\mathcal{S}$ by placing independent Gaussian priors over each element of $\mathcal{S}$, that is

$$p(\mathcal{S}|\boldsymbol{\beta}) = \prod_{ijk} \mathcal{N}(S_{ijk}|0, \beta_{ijk}^{-1}), \tag{15}$$

where $\boldsymbol{\beta} = \{\beta_{ijk}\}$ and β_{ijk} is the precision of the Gaussian distribution for the (i, j, k)th element S_{ijk}. We further place independent Gamma priors for each β_{ijk} and obtain

$$p(\boldsymbol{\beta}) = \prod_{ijk} \mathrm{Ga}(\beta_{ijk}|a_0^\beta, b_0^\beta), \tag{16}$$

where a_0^β and b_0^β are the hyper-parameters, and $\mathrm{Ga}(x|a, b) = \frac{b^a x^{a-1} e^{-bx}}{\Gamma(a)}$ with $\Gamma(a)$ being the Gamma function. Note that as β_{ijk} becomes large, the corresponding S_{ijk} tends to be zero. By encouraging most precision variables to take large values, we can obtain a sparse $\mathcal{S}$ for characterizing outliers.

ARD Prior. For now, we only consider tubal rank determination, while the results below will be generalized for multi-rank determination in Section 3.4. Since the tubal rank of $\mathcal{X}$ is bounded by R, our aim is to introduce lateral-slice sparsity into $\mathcal{U}$ and $\mathcal{V}$, so that the minimum R can be found by removing unnecessary lateral slices from $\mathcal{U}$ and $\mathcal{V}$. To this end, we place the ARD prior [36] over the factor tensors as follows:

$$\begin{aligned} p(\mathcal{U}|\boldsymbol{\lambda}) &= \prod_{i=1}^{I_1} \prod_{r=1}^{R} \mathcal{N}(\mathbf{u}_{ir}|\mathbf{0}, \lambda_r^{-1}\mathbf{I}_{I_3}) \\ &= \prod_{i=1}^{I_1} \mathcal{N}(\overrightarrow{\mathbf{u}}_{i\cdot}|\mathbf{0}, \mathrm{circ}(\Lambda)^{-1}), \end{aligned} \tag{17}$$

$$\begin{aligned} p(\mathcal{V}|\boldsymbol{\lambda}) &= \prod_{j=1}^{I_2} \prod_{r=1}^{R} \mathcal{N}(\mathbf{v}_{jr}|\mathbf{0}, \lambda_r^{-1}\mathbf{I}_{I_3}) \\ &= \prod_{j=1}^{I_2} \mathcal{N}(\overrightarrow{\mathbf{v}}_{j\cdot}|\mathbf{0}, \mathrm{circ}(\Lambda)^{-1}), \end{aligned} \tag{18}$$

$$p(\boldsymbol{\lambda}) = \prod_{r=1}^{R} \mathrm{Ga}(\lambda_r|a_0^\lambda, b_0^\lambda), \tag{19}$$

where $\mathbf{u}_{ir} \in \mathbb{R}^{I_3}$ is the (i, r)th tube of $\mathcal{U}$, $\mathbf{v}_{jr} \in \mathbb{R}^{I_3}$ is the (j, r)th tube of $\mathcal{V}$, $\overrightarrow{\mathbf{u}}_{i\cdot} \in \mathbb{R}^{I_1 I_3} = \mathrm{unfold}(\overrightarrow{\mathcal{U}}_{i\cdot}^\dagger)$, $\overrightarrow{\mathbf{v}}_{j\cdot} \in \mathbb{R}^{I_2 I_3} = \mathrm{unfold}(\overrightarrow{\mathcal{V}}_{j\cdot}^\dagger)$, $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_R]$, and λ_r is the hyper-parameter that controls the rth lateral slices of $\mathcal{U}$ and $\mathcal{V}$. Λ is the $R \times R \times I_3$ tensor whose first frontal slice is the diagonal matrix $\boldsymbol{\Lambda}^{(1)} = \mathrm{diag}(\boldsymbol{\lambda})$ and other slices are all zeros. $\mathrm{circ}(\Lambda)$ is just a diagonal matrix formed by the repeated block $\boldsymbol{\Lambda}^{(1)}$. a_0^λ and b_0^λ are the hyper-parameters of $\boldsymbol{\lambda}$. With the above priors, some elements of $\boldsymbol{\lambda}$ tend to have large values, which in turn pushes the corresponding lateral slices ($\overrightarrow{\mathcal{U}}_{\cdot r}$ and $\overrightarrow{\mathcal{V}}_{\cdot r}$) towards zero. This yields the minimum number of lateral slices required for the low-tubal-rank factorization of $\mathcal{Y}$, and thus determines the tubal rank.

Noise Precision. To complete our fully Bayesian treatment, a conjugate Gamma prior is placed over the noise precision τ, leading to

$$p(\tau) = \mathrm{Ga}(\tau|a_0^\tau, b_0^\tau), \tag{20}$$

where a_0^τ and b_0^τ are commonly set to small values for introducing broad and noninformative priors.

Joint Distribution. Based on the above model specification, we can obtain the joint distribution via $p(\mathcal{Y}, \boldsymbol{\Theta}) = p(\mathcal{Y}|\mathcal{U}, \mathcal{V}, \mathcal{S}, \tau)p(\mathcal{U}|\boldsymbol{\lambda})p(\mathcal{V}|\boldsymbol{\lambda})p(\mathcal{S}|\boldsymbol{\beta})p(\boldsymbol{\lambda})p(\boldsymbol{\beta})p(\tau)$, where $\boldsymbol{\Theta} = \{\mathcal{U}, \mathcal{V}, \boldsymbol{\lambda}, \mathcal{S}, \boldsymbol{\beta}, \tau\}$ is the collection of all the latent variables in the BRTRF model. Fig. 1 shows the graphical model for BTRTF, and the logarithm of $p(\mathcal{D}, \boldsymbol{\Theta})$ is given by

$$\begin{aligned} \ln p(\mathcal{Y}, \boldsymbol{\Theta}) = &-\frac{1}{2}\sum_{ij}\left[\tau||\mathbf{y}_{ij} - \overrightarrow{\mathcal{U}}_{i\cdot} * \overrightarrow{\mathcal{V}}_{j\cdot}^\dagger - \mathbf{s}_{ij}||^2 - I_3 \ln\tau\right] \\ &-\frac{1}{2}\left[\sum_{i=1}^{I_1} \mathrm{tr}(\overrightarrow{\mathbf{u}}_{i\cdot}^\top \mathrm{circ}(\Lambda)\overrightarrow{\mathbf{u}}_{i\cdot}) - \ln|\mathrm{circ}(\Lambda)|\right] \\ &-\frac{1}{2}\left[\sum_{j=1}^{I_2} \mathrm{tr}(\overrightarrow{\mathbf{v}}_{j\cdot}^\top \mathrm{circ}(\Lambda)\overrightarrow{\mathbf{v}}_{j\cdot}) - \ln|\mathrm{circ}(\Lambda)|\right] \\ &+\sum_{r,k}\left[(a_0^\lambda - 1)\ln\lambda_r^{(k)} - b_0^\lambda\lambda_r^{(k)}\right] \\ &-\frac{1}{2}\sum_{ijk}(\beta_{ijk}S_{ijk}^2 - \ln\beta_{ijk}) \\ &+(a_0^\tau - 1)\ln\tau - b_0^\tau\tau + \mathrm{const.} \end{aligned} \tag{21}$$

3.2 Variational Inference

Armed with the above results, the BTRTF model can be learned by estimating the posterior distribution $p(\boldsymbol{\Theta}|\mathcal{Y}) = \frac{p(\mathcal{Y},\boldsymbol{\Theta})}{\int p(\mathcal{Y},\boldsymbol{\Theta})d\boldsymbol{\Theta}}$. Since $p(\boldsymbol{\Theta}|\mathcal{Y})$ is generally intractable, we apply variational inference methods [37], [38] for the model estimation. Specifically, we seek a variational distribution $q(\boldsymbol{\Theta})$ to approximate the true posterior by minimizing the KL divergence $\mathrm{KL}(q(\boldsymbol{\Theta})||p(\boldsymbol{\Theta}|\mathcal{Y})) = \ln p(\mathcal{Y}) - \mathcal{L}(q)$, or equivalently maximizing the *variational lower bound* $\mathcal{L}(q) = \int q(\boldsymbol{\Theta}) \ln\{\frac{p(\mathcal{Y},\boldsymbol{\Theta})}{q(\boldsymbol{\Theta})}\}d\boldsymbol{\Theta}$.

For tractable inference, we use the mean field approximation, and assume that $q(\boldsymbol{\Theta})$ can be factorized as

$$q(\boldsymbol{\Theta}) = q(\mathcal{U})q(\mathcal{V})q(\mathcal{S})q(\boldsymbol{\lambda})q(\boldsymbol{\beta})q(\tau). \tag{22}$$

Then, the optimal distribution of the jth variable set in terms of $\max_{q_j(\boldsymbol{\Theta}_j)} \mathcal{L}(q)$ takes the following form:

$$\ln q_j(\boldsymbol{\Theta}_j) \propto \langle \ln p(\mathcal{Y},\boldsymbol{\Theta})\rangle_{\boldsymbol{\Theta}\backslash\boldsymbol{\Theta}_j}, \tag{23}$$

where $\langle\cdot\rangle_{\boldsymbol{\Theta}\backslash\boldsymbol{\Theta}_j}$ denotes the expectation w.r.t. the variational distributions of all the latent variables in $\boldsymbol{\Theta}$ except $\boldsymbol{\Theta}_j$. By applying the explicit form (23) to the joint distribution (21), we can obtain closed-form solutions for the variational posterior of each variable set $\boldsymbol{\Theta}_j$.

Inference for $\mathcal{U}$ and $\mathcal{V}$. With $\boldsymbol{\Theta}_j = \mathcal{U}$, the posterior $q(\mathcal{U})$ can be obtained as

$$q(\mathcal{U}) = \prod_{i=1}^{I_1} \mathcal{N}(\overrightarrow{\mathbf{u}}_{i\cdot})|\langle\overrightarrow{\mathbf{u}}_{i\cdot}\rangle, \boldsymbol{\Sigma}^u), \tag{24}$$

whose parameters are given by

$$\langle\overrightarrow{\mathbf{u}}_{i\cdot}\rangle = \langle\tau\rangle\boldsymbol{\Sigma}^u \mathrm{circ}(\langle\mathcal{V}\rangle)^\top(\overrightarrow{\mathbf{y}}_{i\cdot} - \langle\overrightarrow{\mathbf{s}}_{i\cdot}\rangle), \tag{25}$$

$$\boldsymbol{\Sigma}^u = \Big(\langle\tau\rangle\langle\mathrm{circ}(\mathcal{V})^\top\mathrm{circ}(\mathcal{V})\rangle + \mathrm{circ}(\langle\Lambda\rangle)\Big)^{-1}. \tag{26}$$

Similarly, the posterior distribution of $\mathcal{V}$ is given by

$$q(\mathcal{V}) = \prod_{j=1}^{I_2} \mathcal{N}(\overrightarrow{\mathbf{v}}_{j\cdot})|\langle\overrightarrow{\mathbf{v}}_{j\cdot}\rangle, \boldsymbol{\Sigma}^v), \tag{27}$$

with the mean and covariance

$$\langle\overrightarrow{\mathbf{v}}_{j\cdot}\rangle = \langle\tau\rangle\boldsymbol{\Sigma}^v \mathrm{circ}(\langle\mathcal{U}\rangle)^\top(\overrightarrow{\mathbf{y}}_{\cdot j} - \langle\overrightarrow{\mathbf{s}}_{\cdot j}\rangle), \tag{28}$$

$$\boldsymbol{\Sigma}^v = \Big(\langle\tau\rangle\langle\mathrm{circ}(\mathcal{U})^\top\mathrm{circ}(\mathcal{U})\rangle + \mathrm{circ}(\langle\Lambda\rangle)\Big)^{-1}. \tag{29}$$

The expectations $\langle\mathrm{circ}(\mathcal{U})^\top\mathrm{circ}(\mathcal{U})\rangle$ and $\langle\mathrm{circ}(\mathcal{V})^\top\mathrm{circ}(\mathcal{V})\rangle$ can be computed as follows:

$$\langle\mathrm{circ}(\mathcal{U})^\top\mathrm{circ}(\mathcal{U})\rangle = I_3\boldsymbol{\Sigma}^u + \mathrm{circ}(\langle\mathcal{U}\rangle)^\top\mathrm{circ}(\langle\mathcal{U}\rangle), \tag{30}$$

$$\langle\mathrm{circ}(\mathcal{V})^\top\mathrm{circ}(\mathcal{V})\rangle = I_3\boldsymbol{\Sigma}^v + \mathrm{circ}(\langle\mathcal{V}\rangle)^\top\mathrm{circ}(\langle\mathcal{V}\rangle). \tag{31}$$

Inference for $\boldsymbol{\lambda}$. Similar to the above derivations, the variational posterior of $\boldsymbol{\lambda}$ is given by

$$q(\boldsymbol{\lambda}) = \prod_{r=1}^{R} \mathrm{Ga}(\lambda_r|a_r^\lambda, b_r^\lambda), \tag{32}$$

where the posterior parameters are

$$a_r^\lambda = a_0^\lambda + \frac{(I_1+I_2)I_3}{2}, b_r^\lambda = b_0^\lambda + \frac{1}{2}\langle\|\overrightarrow{\mathbf{u}}_{\cdot r}\|^2 + \|\overrightarrow{\mathbf{v}}_{\cdot r}\|^2\rangle. \tag{33}$$

The involved expectation can be computed as follows:

$$\langle\|\overrightarrow{\mathbf{u}}_{\cdot r}\|^2\rangle = \sum_{ik}(\boldsymbol{\Sigma}^u + \langle\overrightarrow{\mathbf{u}}_{i\cdot}\rangle\langle\overrightarrow{\mathbf{u}}_{i\cdot}\rangle^\top)_{(k-1)R+r}, \tag{34}$$

$$\langle\|\overrightarrow{\mathbf{v}}_{\cdot r}\|^2\rangle = \sum_{jk}(\boldsymbol{\Sigma}^v + \langle\overrightarrow{\mathbf{v}}_{j\cdot}\rangle\langle\overrightarrow{\mathbf{v}}_{j\cdot}\rangle^\top)_{(k-1)R+r}, \tag{35}$$

where $(\cdot)_{(k-1)R+r}$ denotes the $((k-1)R+r)$th diagonal element of an $RI_3 \times RI_3$ matrix.

From (32) and (33), the expectation of λ_r is given by $\langle\lambda_r\rangle = a_r^\lambda/b_r^\lambda$, which is controlled by the squared ℓ_2 norms of $\overrightarrow{\mathbf{u}}_{\cdot r}$ and $\overrightarrow{\mathbf{v}}_{\cdot r}$. Smaller $\langle\|\overrightarrow{\mathbf{u}}_{\cdot r}\|^2\rangle$ and $\langle\|\overrightarrow{\mathbf{v}}_{\cdot r}\|^2\rangle$ will lead to a larger $\langle\lambda_r\rangle$, which in turn constrains more strongly the corresponding lateral slices towards zero due to (34) and (35).

Inference for $\mathcal{S}$. By applying (23) with $\boldsymbol{\Theta}_j = \mathcal{S}$, the posterior distribution of $\mathcal{S}$ can be obtained as follows:

$$q(\mathcal{S}) = \prod_{ijk} \mathcal{N}(S_{ijk}|\langle S_{ijk}\rangle, \sigma_{ijk}^2), \tag{36}$$

with the parameters

$$\langle S_{ijk}\rangle = \langle\tau\rangle(\langle\beta_{ijk}\rangle + \langle\tau\rangle)z_{ijk}, \tag{37}$$

$$\sigma_{ijk}^2 = (\langle\beta_{ijk}\rangle + \langle\tau\rangle)^{-1}, \tag{38}$$

where z_{ijk} denotes the kth element of $\mathbf{y}_{ij} - \langle\overrightarrow{\mathcal{U}}_{i\cdot}\rangle * \langle\overrightarrow{\mathcal{V}}_{j\cdot}^\dagger\rangle$.

From (37) and (38), $\langle S_{ijk}\rangle$ captures the model residuals from z_{ijk}, and its magnitude is determined by the hyperparameter $\langle\beta_{ijk}\rangle$ and the noise precision $\langle\tau\rangle$. The conceptual meaning of $q(\mathcal{U})$, $q(\mathcal{V})$, and $q(\mathcal{S})$ is that $\mathcal{U} * \mathcal{V}^\dagger$ explains global information of the observed tensor $\mathcal{Y}$ with the minimum tubal rank, while $\mathcal{S}$ explains local information (non-Gaussian outliers) that cannot be well represented by the low-tubal-rank model.

Inference for $\boldsymbol{\beta}$. The posterior distribution of $\boldsymbol{\beta}$ is given by

$$q(\beta_{ijk}) = \mathrm{Ga}(\beta_{ijk}|a_{ijk}^\beta, b_{ijk}^\beta), \tag{39}$$

whose parameters can be updated as follows:

$$a_{ijk}^\beta = a_0^\beta + \frac{1}{2}, b_{ijk}^\beta = b_0^\beta + \frac{1}{2}\langle\beta_{ijk}^2\rangle. \tag{40}$$

Inference for τ. Finally, the noise precision has the following posterior distribution:

$$q(\tau) = \mathrm{Ga}(\tau|a^\tau, b^\tau), \tag{41}$$

whose parameters can be updated as follows:

$$a^\tau = a_\tau^0 + \frac{I}{2}, b^\tau = b_0^\tau + \frac{1}{2}\sum_{ij}\langle||\mathbf{y}_{ij} - \overrightarrow{\mathcal{U}}_{i\cdot} * \overrightarrow{\mathcal{V}}_{j\cdot}^\dagger - \mathbf{s}_{ij}||^2\rangle. \tag{42}$$

The expectation of the model error is given by

$$\begin{aligned}\langle \|\mathbf{y}_{ij} - \overrightarrow{\mathcal{U}}_{i\cdot} * \overrightarrow{\mathcal{V}}_{j\cdot}^{\dagger} - \mathbf{s}_{ij}\|^2 \rangle &= I_1 I_2 I_3 \mathrm{tr}(\boldsymbol{\Sigma}^u \boldsymbol{\Sigma}^v) \\ &+ I_1 I_3 \langle \overrightarrow{\mathbf{v}}_{j\cdot} \rangle^\top \boldsymbol{\Sigma}^u \langle \overrightarrow{\mathbf{v}}_{j\cdot} \rangle + I_2 I_3 \langle \overrightarrow{\mathbf{u}}_{i\cdot} \rangle^\top \boldsymbol{\Sigma}^v \langle \overrightarrow{\mathbf{u}}_{i\cdot} \rangle \\ &+ \|\mathbf{y}_{ij} - \langle \overrightarrow{\mathcal{U}}_{i\cdot} \rangle * \langle \overrightarrow{\mathcal{V}}_{j\cdot} \rangle^\dagger - \langle \mathbf{s}_{ij} \rangle\|^2 + \sum_{ijk} \sigma_{ijk}^2. \end{aligned} \tag{43}$$

3.3 Efficient Updates in Frequency Domain

Although the above variational inference involves only closed-form updates, it is still relatively time consuming. Specifically, the updates for $q(\mathcal{U})$ and $q(\mathcal{V})$ dominate the whole variational inference. They require inversing and multiplying the $RI_3 \times RI_3$ covariance matrices $\boldsymbol{\Sigma}^u$ and $\boldsymbol{\Sigma}^v$, leading to $O(R^3 I_3^3 + RI_1 I_2 I_3^2)$ time complexity. This is impractical when dealing with real-world data with large I_3. Fortunately, such time complexity can be greatly reduced by using *DFT* and reformulating the variational updates in the *frequency domain*. In what follows, we provide efficient variational updates for BTRTF, which not only reduce the time complexity to $O(R^3 I_3 + RI_1 I_2 I_3)$, but also lay the foundation for automatic *multi-rank* determination.

From (25), we can group all the horizontal slices of $\mathcal{U}$ together and obtain

$$\begin{aligned}\mathrm{unfold}(\langle \mathcal{U} \rangle^\dagger) &= \langle (\overrightarrow{\mathbf{u}}_{1\cdot}, \ldots, \overrightarrow{\mathbf{u}}_{I_1\cdot}) \rangle \\ &= \langle \tau \rangle \boldsymbol{\Sigma}^u \mathrm{circ}(\langle \mathcal{V} \rangle)^\top \mathrm{unfold}(\mathcal{Y}^\dagger - \langle \mathcal{S} \rangle^\dagger).\end{aligned}$$

It is worth noting that although $\boldsymbol{\Sigma}^u$ and $\mathrm{circ}(\langle \mathcal{V} \rangle)$ have a large size of $RI_3 \times RI_3$, both of them are *block circulant* matrices and can be block diagonalized by DFT. As a result, their multiplication and inverse can be efficiently computed in the frequency domain.

Let $\hat{\mathbf{F}} = \mathbf{F}_{I_3} \otimes \mathbf{I}_{I_1}$ and $\langle \bar{\mathcal{U}}^\dagger \rangle = \mathrm{fft}(\langle \mathcal{U}^\dagger \rangle, [], 3)$ be the blockwise DFT matrix and the DFT of $\langle \mathcal{U}^\dagger \rangle$, respectively. Then, it is easy to verify that

$$\begin{aligned}&\mathrm{unfold}(\langle \bar{\mathcal{U}} \rangle^\dagger) = \hat{\mathbf{F}} \cdot \mathrm{unfold}(\langle \mathcal{U} \rangle^\dagger) \\ &= \langle \tau \rangle \hat{\mathbf{F}} \boldsymbol{\Sigma}^u \hat{\mathbf{F}}^{-1} \hat{\mathbf{F}} \cdot \mathrm{circ}(\langle \mathcal{V} \rangle)^\top \hat{\mathbf{F}}^{-1} \hat{\mathbf{F}} \cdot \mathrm{unfold}(\mathcal{Y}^\dagger - \langle \mathcal{S} \rangle^\dagger).\end{aligned}$$

This indicates that $\langle \bar{\mathcal{U}} \rangle$ can be computed in a *block-wise* manner by using (7), and similar results hold for $\langle \bar{\mathcal{V}} \rangle$ as well. Therefore, we can infer $q(\mathcal{U})$ and $q(\mathcal{V})$ by equivalently updating the *DFTs* of their parameters instead of the original ones. Specifically, the kth frontal slice of $\langle \bar{\mathcal{U}} \rangle$ and $\langle \bar{\mathcal{V}} \rangle$ can be updated as follows:

$$\langle \bar{\mathbf{U}}^{(k)} \rangle = \langle \tau \rangle (\bar{\mathbf{Y}}^{(k)} - \langle \bar{\mathbf{S}}^{(k)} \rangle) \langle \bar{\mathbf{V}}^{(k)} \rangle \bar{\boldsymbol{\Sigma}}^{u(k)}, \tag{44}$$

$$\bar{\boldsymbol{\Sigma}}^{u(k)} = (\langle \tau \rangle \langle \bar{\mathbf{V}}^{(k)\dagger} \bar{\mathbf{V}}^{(k)} \rangle + \mathrm{diag}(\langle \boldsymbol{\lambda} \rangle))^{-1}, \tag{45}$$

$$\langle \bar{\mathbf{V}}^{(k)} \rangle = \langle \tau \rangle (\bar{\mathbf{Y}}^{(k)} - \langle \bar{\mathbf{S}}^{(k)} \rangle)^\dagger \langle \bar{\mathbf{U}}^{(k)} \rangle \bar{\boldsymbol{\Sigma}}^{v(k)}, \tag{46}$$

$$\bar{\boldsymbol{\Sigma}}^{v(k)} = (\langle \tau \rangle \langle \bar{\mathbf{U}}^{(k)\dagger} \bar{\mathbf{U}}^{(k)} \rangle + \mathrm{diag}(\langle \boldsymbol{\lambda} \rangle))^{-1}, \tag{47}$$

where $\langle \bar{\mathbf{U}}^{(k)} \rangle \in \mathbb{C}^{I_1 \times R}$, $\langle \bar{\mathbf{V}}^{(k)} \rangle \in \mathbb{C}^{I_2 \times R}$, and $\langle \bar{\mathbf{S}}^{(k)} \rangle \in \mathbb{C}^{I_1 \times I_2}$ denote the kth frontal slice of $\langle \bar{\mathcal{U}} \rangle$, $\langle \bar{\mathcal{V}} \rangle$, and $\langle \bar{\mathcal{S}} \rangle$, respectively. The expectations in $\bar{\boldsymbol{\Sigma}}^{u(k)}$ and $\bar{\boldsymbol{\Sigma}}^{v(k)}$ can be computed by

$$\langle \bar{\mathbf{U}}^{(k)\dagger} \bar{\mathbf{U}}^{(k)} \rangle = I_1 I_3 \bar{\boldsymbol{\Sigma}}^{v(k)} + \langle \bar{\mathbf{U}}^{(k)} \rangle^\dagger \langle \bar{\mathbf{U}}^{(k)} \rangle, \tag{48}$$

$$\langle \bar{\mathbf{V}}^{(k)\dagger} \bar{\mathbf{V}}^{(k)} \rangle = I_2 I_3 \bar{\boldsymbol{\Sigma}}^{u(k)} + \langle \bar{\mathbf{V}}^{(k)} \rangle^\dagger \langle \bar{\mathbf{V}}^{(k)} \rangle. \tag{49}$$

With the above results, we avoid directly manipulating the $RI_3 \times RI_3$ covariance matrices in (25) and (28), and turn to updating I_3 much smaller frontal slices in the frequency domain via (44) and (46). Consequently, the computational cost for estimating $q(\mathcal{U})$ and $q(\mathcal{V})$ is reduced from $O(R^3 I_3^3 + RI_1 I_2 I_3^2)$ to $O(R^3 I_3 + RI_1 I_2 I_3)$. The estimation for $\boldsymbol{\lambda}$ and τ can also be accelerated by computing the expectations (34), (35), and (43) in the frequency domain, leading to

$$\langle \|\overrightarrow{\mathbf{u}}_{\cdot r}\|^2 \rangle = \sum_{k=1}^{I_3} \left(I_1 \bar{\boldsymbol{\Sigma}}^{u(k)} + \frac{1}{I_3} \langle \bar{\mathbf{U}}^{(k)} \rangle^\dagger \langle \bar{\mathbf{U}}^{(k)} \rangle \right)_{rr}, \tag{50}$$

$$\langle \|\overrightarrow{\mathbf{v}}_{\cdot r}\|^2 \rangle = \sum_{k=1}^{I_3} \left(I_2 \bar{\boldsymbol{\Sigma}}^{v(k)} + \frac{1}{I_3} \langle \bar{\mathbf{V}}^{(k)} \rangle^\dagger \langle \bar{\mathbf{V}}^{(k)} \rangle \right)_{rr}, \tag{51}$$

$$\begin{aligned}&\sum_{ij} \langle \|\mathbf{y}_{ij} - \overrightarrow{\mathcal{U}}_{i\cdot} * \overrightarrow{\mathcal{V}}_{j\cdot}^{\dagger} - \mathbf{s}_{ij}\|^2 \rangle \\ &= \|\mathcal{Y} - \langle \mathcal{U} \rangle * \langle \mathcal{V} \rangle^\dagger - \langle \mathcal{S} \rangle\|_F^2 + I_1 I_2 I_3 \sum_{k=1}^{I_3} \mathrm{tr}(\bar{\boldsymbol{\Sigma}}^{u(k)} \bar{\boldsymbol{\Sigma}}^{v(k)}) \\ &+ I_1 \sum_{k=1}^{I_3} \mathrm{tr}(\bar{\boldsymbol{\Sigma}}^{u(k)} \langle \bar{\mathbf{V}}^{(k)} \rangle^\dagger \langle \bar{\mathbf{V}}^{(k)} \rangle) \\ &+ I_2 \sum_{k=1}^{I_3} \mathrm{tr}(\bar{\boldsymbol{\Sigma}}^{v(k)} \langle \bar{\mathbf{U}}^{(k)} \rangle^\dagger \langle \bar{\mathbf{U}}^{(k)} \rangle) + \sum_{ijk} \sigma_{ijk}^2,\end{aligned} \tag{52}$$

where $(\cdot)_{rr}$ denotes the rth diagonal element of a $R \times R$ matrix. As $\mathcal{S}$ and $\boldsymbol{\beta}$ are factorized over elements, their updates cannot be further accelerated in the frequency domain, and stay the same.

3.4 Multi-Rank Prior

While the ARD prior achieves automatic tubal rank determination by introducing slice-wise sparsity in $\mathcal{U}$ and $\mathcal{V}$, it is still too restrictive to determine the multi-rank. Recall that the low-tubal-rank model $\mathcal{X} = \mathcal{U} * \mathcal{V}^\dagger$ is equivalent to $\bar{\mathbf{X}} = \bar{\mathbf{U}} \bar{\mathbf{V}}^\dagger$ because of (7), and the kth diagonal block of $\bar{\mathbf{X}}$ is given by $\bar{\mathbf{X}}^{(k)} = \bar{\mathbf{U}}^{(k)} \bar{\mathbf{V}}^{(k)\dagger}$ [35]. From Definition 2.7, the multi-rank of $\mathcal{X}$ is the vector $Rank_{\mathrm{m}}(\mathcal{X}) = (Rank(\bar{\mathbf{X}}^{(1)}), \ldots, Rank(\bar{\mathbf{X}}^{(I_3)}))$, and its kth element $Rank(\bar{\mathbf{X}}^{(k)})$ is controlled by the *number of columns* in $\bar{\mathbf{U}}^{(k)}$ and $\bar{\mathbf{V}}^{(k)}$. Notice that the tubal rank $Rank_{\mathrm{t}}(\mathcal{X}) = \max_k Rank(\bar{\mathbf{X}}^{(k)})$ is just the largest element of $Rank_{\mathrm{m}}(\mathcal{X})$. This indicates that determining multi-rank is a more general and challenging problem.

For automatic multi-rank determination, we need to fit the observed tensor while reducing the effective multi-rank. To this end, we propose a generalized ARD prior, named as multi-rank prior, by imposing sparse constraints on the columns of $\bar{\mathbf{U}}^{(k)}$ and $\bar{\mathbf{V}}^{(k)}$. Similar to (17) and (18), we still place a Gaussian prior over the latent factors $\mathcal{U}$ and $\mathcal{V}$ as follows:

$$\begin{aligned}p(\mathcal{U}|\boldsymbol{\lambda}_{\mathrm{m}}) &= \prod_{i=1}^{I_1} \prod_{r=1}^{R} \mathcal{N}(\mathbf{u}_{ir}|\mathbf{0}, \mathrm{circ}(\boldsymbol{\lambda}_r)^{-1}) \\ &= \prod_{i=1}^{I_1} \mathcal{N}(\overrightarrow{\mathbf{u}}_{i\cdot}|\mathbf{0}, \mathrm{circ}(\Lambda_{\mathrm{m}})^{-1}),\end{aligned} \tag{53}$$

$$\begin{aligned} p(\mathcal{V}|\boldsymbol{\lambda}_{\mathrm{m}}) &= \prod_{j=1}^{I_2}\prod_{r=1}^{R}\mathcal{N}(\mathbf{v}_{jr}|\mathbf{0}, \mathrm{circ}(\boldsymbol{\lambda}_r)^{-1}) \\ &= \prod_{j=1}^{I_2}\mathcal{N}(\overrightarrow{\mathbf{v}}_{j\cdot}|\mathbf{0}, \mathrm{circ}(\Lambda_{\mathrm{m}})^{-1}), \end{aligned} \tag{54}$$

where $\boldsymbol{\lambda}_{\mathrm{m}} = \{\lambda_r^{(k)}\}$, $\boldsymbol{\lambda}_r = [\lambda_r^{(1)}, \ldots, \lambda_r^{(I_3)}]^\top$, $\mathrm{circ}(\boldsymbol{\lambda}_r) \in \mathbb{R}^{I_3\times I_3}$ is the circulant matrix constructed by $\boldsymbol{\lambda}_r$, and Λ_{m} is the $R\times R\times I_3$ f-diagonal tensor whose kth frontal slice is given by $\boldsymbol{\Lambda}_{\mathrm{m}}^{(k)} = \mathrm{diag}([\lambda_1^{(k)}, \ldots, \lambda_R^{(k)}])$. To make sure $\mathrm{circ}(\boldsymbol{\lambda}_r)$ is symmetric as a valid covariance matrix, we define $\lambda_r^{(k)} = \lambda_r^{(I_3-k-2)}$ for $k = 2, \ldots, I_3$.

Compared with (17) and (18), our multi-rank prior has a similar form with the ARD prior, while the precision matrix for each tube is changed from $\lambda_r^{-1}\mathbf{I}_{I_3}$ to $\mathrm{circ}(\boldsymbol{\lambda}_r)$. Essentially, the ARD prior assumes that all the elements in $\mathcal{U}$ and $\mathcal{V}$ are independent, and makes each pair of lateral slices ($\overrightarrow{\mathcal{U}}_{\cdot r}$ and $\overrightarrow{\mathcal{V}}_{\cdot r}$) governed by the same hyper-parameter λ_r. On the other hand, the proposed multi-rank prior takes a more general covariance matrix $\mathrm{circ}(\boldsymbol{\lambda}_r)$ for the tubes of $\overrightarrow{\mathcal{U}}_{\cdot r}$ and $\overrightarrow{\mathcal{V}}_{\cdot r}$, and thus generalizes the ARD prior by characterizing the correlations within each tube of $\mathcal{U}$ and $\mathcal{V}$.

By incorporating (53) and (54) into the BTRTF model, the posterior distributions of $\mathcal{U}$ and $\mathcal{V}$ still follow (24) and (27), respectively, expect that the term $\mathrm{circ}(\langle\Lambda\rangle)$ is *replaced* by $\mathrm{circ}(\langle\Lambda_{\mathrm{m}}\rangle)$ in the covariance matrices (26) and (29). In the frequency domain, the updates for $\langle\overrightarrow{\mathbf{u}}_{i\cdot}\rangle$ and $\langle\overrightarrow{\mathbf{v}}_{j\cdot}\rangle$ are still the same via (44) and (46), repressively, while the updates for $\boldsymbol{\Sigma}^v$ and $\boldsymbol{\Sigma}^u$ become

$$\bar{\boldsymbol{\Sigma}}^{u(k)} = (\langle\tau\rangle\langle\bar{\mathbf{V}}^{(k)\dagger}\bar{\mathbf{V}}^{(k)}\rangle + \langle\bar{\boldsymbol{\Lambda}}_{\mathrm{m}}^{(k)}\rangle)^{-1}, \tag{55}$$

$$\bar{\boldsymbol{\Sigma}}^{v(k)} = (\langle\tau\rangle\langle\bar{\mathbf{U}}^{(k)\dagger}\bar{\mathbf{U}}^{(k)}\rangle + \langle\bar{\boldsymbol{\Lambda}}_{\mathrm{m}}^{(k)}\rangle)^{-1}, \tag{56}$$

where $\langle\bar{\boldsymbol{\Lambda}}_{\mathrm{m}}^{(k)}\rangle = \mathrm{diag}([\langle\bar{\lambda}_1^{(k)}\rangle, \ldots, \langle\bar{\lambda}_R^{(k)}\rangle])$ is the kth frontal slice of $\langle\bar{\Lambda}_{\mathrm{m}}\rangle = \mathrm{fft}(\langle\Lambda_{\mathrm{m}}\rangle, [], 3)$.

Due to the more general precision matrix $\mathrm{circ}(\Lambda_{\mathrm{m}})$, incorporating the multi-rank prior leads to the determinant term $\ln|\mathrm{circ}(\Lambda_{\mathrm{m}})|$. Unlike the ARD case with $\ln|\mathrm{circ}(\Lambda)| = I_3\sum_{r=1}^{R}\ln\lambda_r$, it cannot be decomposed into the sum of $\ln\lambda_r^{(k)}$. Consequently, placing a Gamma distribution over $\lambda_r^{(k)}$ will no longer lead to a tractable variational posterior $q(\lambda_r^{(k)})$. To address this problem, we treat $\bar{\lambda}_r^{(k)}$ rather than $\lambda_r^{(k)}$ as a latent variable and place a Gamma distribution over it, leading to

$$p(\bar{\boldsymbol{\lambda}}_{\mathrm{m}}) = \prod_{r=1}^{R}\prod_{k=1}^{I_3}\mathrm{Ga}(\bar{\lambda}_r^{(k)}|a_0^\lambda, b_0^\lambda), \tag{57}$$

where we have defined $\bar{\boldsymbol{\lambda}}_{\mathrm{m}} = \{\bar{\lambda}_r^{(k)}\}$.

It is worth noting that although the hyper-parameters $\boldsymbol{\lambda}_{\mathrm{m}}$ are coupled, their DFTs $\bar{\boldsymbol{\lambda}}_{\mathrm{m}}$ are decomposable in $\ln|\mathrm{circ}(\Lambda_{\mathrm{m}})| = \sum_{rk}\ln\bar{\lambda}_r^{(k)}$ by applying (7). Due to this fact, we can substitute the prior distributions (53), (54), and (57) into the explicit form (23), and obtain the variational posterior for $\bar{\boldsymbol{\lambda}}_{\mathrm{m}}$ as follows:

$$q(\bar{\boldsymbol{\lambda}}_{\mathrm{m}}) = \prod_{r=1}^{R}\prod_{k=1}^{I_3}\mathrm{Ga}(\bar{\lambda}_r^{(k)}|a_{rk}^\lambda, b_{rk}^\lambda), \tag{58}$$

where the posterior parameters can be updated by

$$a_{rk}^\lambda = a_0^\lambda + \frac{I_1+I_2}{2}, \tag{59}$$

$$b_{rk}^\lambda = b_0^\lambda + \frac{1}{2I_3}(\langle\bar{\mathbf{U}}^{(k)\dagger}\bar{\mathbf{U}}^{(k)}\rangle + \langle\bar{\mathbf{V}}^{(k)\dagger}\bar{\mathbf{V}}^{(k)}\rangle)_{rr}. \tag{60}$$

The involved expectations $\langle\bar{\mathbf{U}}^{(k)\dagger}\bar{\mathbf{U}}^{(k)}\rangle$ and $\langle\bar{\mathbf{V}}^{(k)\dagger}\bar{\mathbf{V}}^{(k)}\rangle$ have been given by (48) and (49), respectively, and the posterior mean is given by $\langle\bar{\lambda}_r^{(k)}\rangle = a_{rk}^\lambda/b_{rk}^\lambda$.

Sparsity in the Frequency Domain. Let $\bar{\mathbf{u}}_r^{(k)}$ and $\bar{\mathbf{v}}_r^{(k)}$ be the rth component (column) of $\bar{\mathbf{U}}^{(k)}$ and $\bar{\mathbf{V}}^{(k)}$. An intuitive interpretation of $q(\bar{\boldsymbol{\lambda}}_{\mathrm{m}})$ (58) is that a_{rk}^λ is related to the number of elements in $\bar{\mathbf{u}}_r^{(k)}$ and $\bar{\mathbf{v}}_r^{(k)}$, and b_{rk}^λ is related to the squared ℓ_2 norms $\langle\|\bar{\mathbf{u}}_r^{(k)}\|^2\rangle = (\langle\bar{\mathbf{U}}^{(k)\dagger}\bar{\mathbf{U}}^{(k)}\rangle)_{rr}$ and $\langle\|\bar{\mathbf{v}}_r^{(k)}\|^2\rangle = (\langle\bar{\mathbf{V}}^{(k)\dagger}\bar{\mathbf{V}}^{(k)}\rangle)_{rr}$. Smaller $\langle\|\bar{\mathbf{u}}_r^{(k)}\|^2\rangle$ and $\langle\|\bar{\mathbf{v}}_r^{(k)}\|^2\rangle$ will lead to a larger $\bar{\lambda}_r^{(k)}$, which in turn pushes the corresponding $\bar{\mathbf{u}}_r^{(k)}$ and $\bar{\mathbf{v}}_r^{(k)}$ towards zero. In this way, the multi-rank prior effectively makes unnecessary components $\bar{\mathbf{u}}_r^{(k)}$ and $\bar{\mathbf{v}}_r^{(k)}$ inactive by constraining them to zero, and thus results in automatic multi-rank determination.

Refinement with Relaxed Regularization. In our experiments, we find the multi-rank prior may lead to premature model and prune most factors before fitting the input data. To address this problem, we propose a refinement trick to relax the regularization effect of the multi-rank prior especially at early iterations. Specifically, we gradually strengthen the regularization effect by making the following modifications in updating $\bar{\boldsymbol{\Sigma}}^{u(k)}$ and $\bar{\boldsymbol{\Sigma}}^{v(k)}$

$$\bar{\boldsymbol{\Sigma}}^{u(k)} = (\langle\tau\rangle\langle\bar{\mathbf{V}}^{(k)\dagger}\bar{\mathbf{V}}^{(k)}\rangle + \frac{Fit}{\gamma}\langle\bar{\boldsymbol{\Lambda}}_{\mathrm{m}}^{(k)}\rangle)^{-1}, \tag{61}$$

$$\bar{\boldsymbol{\Sigma}}^{v(k)} = (\langle\tau\rangle\langle\bar{\mathbf{U}}^{(k)\dagger}\bar{\mathbf{U}}^{(k)}\rangle + \frac{Fit}{\gamma}\langle\bar{\boldsymbol{\Lambda}}_{\mathrm{m}}^{(k)}\rangle)^{-1}, \tag{62}$$

where $\gamma > 0$ is the relaxation parameter that adjusts the overall regularization strength of $\langle\bar{\boldsymbol{\Lambda}}_{\mathrm{m}}^{(k)}\rangle$. $Fit = 1 - \langle||\mathcal{Y} - \mathcal{U}*\mathcal{V}^\dagger - \mathcal{S}||_F\rangle/||\mathcal{Y}||_F$ indicates the goodness of fit for the BTRTF model (12), where $\langle||\mathcal{Y} - \mathcal{U}*\mathcal{V}^\dagger - \mathcal{S}||_F\rangle$ is the square root of (52).

At the first few iterations, the low-tubal-rank model will not fit the observed tensor $\mathcal{Y}$ well, leading to a relatively large model error and small Fit. In this case, the regularization term $\langle\bar{\boldsymbol{\Lambda}}_{\mathrm{m}}^{(k)}\rangle$ does not have much effect on the parameter estimation, and thus no factor will be pruned at early iterations. As the BTRTF model fits $\mathcal{Y}$ better and better, Fit tends to converge to 1 and gradually strengthens the regularization effect. Eventually, the refined updates (61) and (62) return to the original ones (55) and (56) given $\gamma = 1$. In general, the parameter γ could be tuned for different applications, while we find that simply fixing $\gamma = I_3$ is enough to achieve good performance in most cases. Therefore, we set $\gamma = I_3$ in all the experiments unless otherwise specified. Algorithm 1 summaries the variational inference method for BTRTF with multi-rank determination.

3.5 Initialization

Since the variational inference method converges only to a local optimum, it is necessary to select a reasonable initialization to avoid poor local solutions. For BTRTF, we set the top level hyper-parameters a_0^λ, b_0^λ, a_0^β, b_0^β, a_0^τ, and b_0^τ to 10^{-6}

for introducing noninformative priors. We then set the model precision $\langle\tau\rangle = a_0^\tau/b_0^\tau = 1$. The factor tensors $\langle\mathcal{U}\rangle$ and $\langle\mathcal{V}\rangle$ can be initialized randomly by drawing each element from $\mathcal{N}(0,1)$. Another choice is to set $\langle\mathcal{U}\rangle = \mathcal{U}_0 * \mathcal{D}_0^{\frac{1}{2}}$ and $\langle\mathcal{V}\rangle = \mathcal{V}_0 * \mathcal{D}_0^{\frac{1}{2}}$, where $\mathcal{U}_0$, $\mathcal{V}_0$, and $\mathcal{D}_0$ are obtained from the t-SVD of $\mathcal{Y} = \mathcal{U}_0 * \mathcal{D}_0 * \mathcal{V}_0^\dagger$. The covariance matrices $\mathbf{\Sigma}^u$ and $\mathbf{\Sigma}^v$ are set to the identity matrix, and the hyper-parameter $\langle\bar{\lambda}_r^{(k)}\rangle$ for $\bar{\mathbf{u}}_r^{(k)}$ and $\bar{\mathbf{v}}_r^{(k)}$ is set to $a_0^\lambda/b_0^\lambda = 1$. The hyper-parameter $\langle\beta_{ijk}\rangle$ is set to $1/\sigma_0^2$, and the sparse component $\langle S_{ijk}\rangle$ is drawn from the uniform distribution $\mathcal{U}(0,\sigma_0)$, where σ_0^2 is a task-specific constant and serves as the initialized variance of S_{ijk} (see Sections 4.2 and 4.3 for more details).

Algorithm 1. BTRTF with Multi-Rank Determination

1: **Input:** The observed tensor $\mathcal{Y} \in \mathbb{R}^{I_1\times I_2\times I_3}$ and the initialized multi-rank $Rank_{\mathrm{m}}(\hat{\mathcal{X}}^0) \in \mathbb{R}^{I_3}$.
2: Initialize $\mathcal{U}$, $\mathbf{\Sigma}^u$, $\mathcal{V}$, $\mathbf{\Sigma}^v$, $\bar{\boldsymbol{\lambda}}_{\mathrm{m}}$, $\mathcal{S}$, $\boldsymbol{\beta}$, and τ.
3: **repeat**
4: Update the posterior $q(\mathcal{U})$ via (44) and (61);
5: Update the posterior $q(\mathcal{V})$ via (46) and (62);
6: Update the posterior $q(\bar{\boldsymbol{\lambda}}_{\mathrm{m}})$ via (58);
7: Update the posterior $q(\mathcal{S})$ via (36);
8: Update the posterior $q(\boldsymbol{\beta})$ via (39);
9: Update the posterior $q(\tau)$ via (41);
10: Reduce the effective multi-rank by removing zero-components of $\bar{\mathbf{U}}^{(k)}$ and $\bar{\mathbf{V}}^{(k)}$;
11: **until** convergence.

3.6 Connections with Existing Work

In this work, we mainly focus on the TRPCA problem, i.e., recovering tensors corrupted with outliers. One representative TRPCA method is SNN [21], which finds the uncorrupted tensor by minimizing the Tucker rank. KDRSDL [22] also seeks recovering a low-Tucker-rank tensor, while this is achieved by fitting the Tucker model with a predetermined Tucker rank. BRTF [28] formulates CP factorization under the Bayesian framework to obtain probabilistic outputs and automatic CP rank determination. The proposed BTRTF method also takes advantage of the Bayesian framework. Different from BRTF, it represents the uncorrupted tensor with the low-tubal-rank model instead of the CP one, leading to more expressive modeling power and more efficient variational updates.

Except the TRPCA problem, there have been many probabilistic tensor factorization methods for other applications such as tensor completion [33], [39], [40], [41], network analysis [42], [43], feature selection [44], multi-view learning [45], etc. For example, Bayesian Probabilistic Tensor Factorization [40] uses the CP model with the smooth constraints on the time dimension to address the temporal collaborative filtering problem. Infinite Tucker Decomposition [42], [43] introduces tensor-variate Gaussian and t processes into the Tucker model to discover nonlinear interactions among tensor elements. Bayesian multi-tensor factorization [45] proposes a relaxed model to jointly factorize multiple matrices and tensors, which can be viewed as a trade-off between the matrix (Tucker-1) and CP factorization.

Most existing probabilistic tensor factorization methods are based on the Tucker or CP model. In contrast, BTRTF is based on the low-tubal-rank model with very distinct Bayesian formulations. Although BTRTF is developed for the TRPCA problem, its low-tubal-rank model specification and variational inference scheme are general enough and could be extended for other applications such as tensor completion and feature extraction.

4 Experiments

This section evaluates our BTRTF on both synthetic and real-world datasets. We apply BTRTF to image denoising and background modeling, and compare it against several state-of-the-art RPCA methods, including *RPCA baselines*: RPCA [6], VBRPCA [46]; *CP based RTF*: BRTF [28]; *Tucker based TRPCAs*: SNN [47], KDRSDL [22]; and *Low-tubal-rank TRPCAs*: TNN [35], OR-TPCA [48].[1]

4.1 Validation on Synthetic Data

We first validate the effectiveness of BTRTF in tensor recovery and multi-rank determination on synthetic datasets. The synthetic data are generated as follows: Two factor tensors $\mathcal{U} \in \mathbb{R}^{I\times R\times I}$ and $\mathcal{V} \in \mathbb{R}^{I\times R\times I}$ are randomly generated with their elements independently drawn from the standard Gaussian distribution $\mathcal{N}(0,1)$. Then, the low-rank component is constructed by $\mathcal{X}_{gt} \in \mathbb{R}^{I\times I\times I} = \mathcal{U} * \mathcal{V}^\dagger$, and is further truncated by t-SVD to have $Rank_{\mathrm{m}}(\mathcal{X}_{gt}) = (R_{gt}^{(1)},\ldots,R_{gt}^{(I)})$. We generate the sparse component $\mathcal{S}_{gt} \in \mathbb{R}^{I\times I\times I}$ by randomly selecting $\rho\%$ of the I^3 elements to be nonzero, whose values are uniformly drawn from $[-10,10]$. The noise term $\mathcal{E} \in \mathbb{R}^{I\times I\times I}$ is generated by independently sampling its elements from $\mathcal{N}(0,\sigma^2)$ with the noise variance $\sigma^2 = 0$ or $\sigma^2 = 10^{-3}$, where $\sigma^2 = 0$ indicates the noise-free case. Finally, the observed tensor is constructed by $\mathcal{Y} = \mathcal{X}_{gt} + \mathcal{S}_{gt} + \mathcal{E}$. In this experiment, we initialize the sparse component with $\sigma_0^2 = 1$ and set the relaxation parameter $\gamma = 1$, so that their values will have no effect on model estimation. The initialized rank of BTRTF is set to $Rank_{\mathrm{m}}(\hat{\mathcal{X}}^0) = (0.5I,\ldots,0.5I) \in \mathbb{R}^I$. The convergence criterion is $tol = \frac{\|\hat{\mathcal{X}}^t - \hat{\mathcal{X}}^{t-1}\|_F}{\|\hat{\mathcal{X}}^{t-1}\|_F} < 10^{-6}$, where $\hat{\mathcal{X}}^t$ is the estimated low-rank component at the tth iteration.

Table 2 shows the recovery results of BTRTF on the synthetic data, where the rank error is defined as $R_{err} = \sum_{k=1}^{I} \frac{|\hat{R}^{(k)} - R_{gt}^{(k)}|}{I_3}$ and $\hat{R}^{(k)}$ is the estimated rank of the kth frontal slice. As can be seen, BTRTF provides the correct multi-rank in all the cases. It also obtains accurate reconstructions for the low-rank and sparse components on the both noise-free and noisy data. These demonstrate that BTRTF is capable of accurately recovering corrupted tensors and determining the correct multi-rank.

To further test BTRTF in multi-rank determination, we compare BTRTF with Tensor Completion by Tensor Factorization (TCTF) [33], which is a low-tubal-rank tensor completion method equipped with a heuristic multi-rank determination strategy. Since TCTF cannot handle outliers, BTRTF and TCTF are performed on synthetic tensors without outliers ($\rho = 0\%$) for fair comparison. Table 3 shows the

1. Since OR-TPCA is designed mainly for classification and performs worse than TNN in our experiments, its results are not reported for simplicity.

TABLE 2
Recovery Results of BTRTF on the Synthetic Datasets

$Rank_{\mathrm{m}}(\mathcal{X}_{gt}) = \{R, \overbrace{0.5R, \ldots, 0.5R}^{40}, \overbrace{R, \ldots, R}^{I_3-81} \overbrace{0.5R, \ldots, 0.5R}^{40}\}$

I	R	ρ	σ^2	R_{err}	$\frac{\|\|\hat{\mathcal{X}}-\mathcal{X}_{gt}\|\|_F}{\|\|\mathcal{X}_{gt}\|\|_F}$	$\frac{\|\|\hat{\mathcal{S}}-\mathcal{S}_{gt}\|\|_F}{\|\|\mathcal{S}_{gt}\|\|_F}$
100	10	5%	0	0	1.26e−7	2.39e−6
			10^{-3}	0	1.46e−5	6.28e−4
		10%	0	0	1.94e−7	2.62e−6
			10^{-3}	0	1.50e−5	4.46e−4
		20%	0	0	3.90e−7	3.65e−6
			10^{-3}	0	1.60e−5	3.23e−4
200	20	5%	0	0	8.95e−8	3.87e−6
			10^{-3}	0	7.20e−6	5.61e−4
		10%	0	0	1.46e−7	4.41e−6
			10^{-3}	0	7.43e−6	4.04e−4
		20%	0	0	3.42e−7	7.07e−6
			10^{-3}	0	7.97e−5	2.96e−4

$Rank_{\mathrm{m}}(\mathcal{X}_{gt}) = \{0.5R, \overbrace{R, \ldots, R}^{40}, \overbrace{0.5R, \ldots, 0.5R}^{I_3-81}, \overbrace{R, \ldots, R}^{40}\}$

I	R	ρ	σ^2	R_{err}	$\frac{\|\|\hat{\mathcal{X}}-\mathcal{X}_{gt}\|\|_F}{\|\|\mathcal{X}_{gt}\|\|_F}$	$\frac{\|\|\hat{\mathcal{S}}-\mathcal{S}_{gt}\|\|_F}{\|\|\mathcal{S}_{gt}\|\|_F}$
100	10	5%	0	0	1.40e−7	3.33e−6
			10^{-3}	0	1.45e−5	5.71e−4
		10%	0	0	2.29e−7	3.80e−6
			10^{-3}	0	1.48e−5	4.09e−4
		20%	0	0	5.00e−7	5.64e−6
			10^{-3}	0	1.61e−5	3.00e−4
200	20	5%	0	0	8.45e−8	3.43e−6
			10^{-3}	0	7.23e−6	5.82e−4
		10%	0	0	1.47e−7	4.15e−6
			10^{-3}	0	7.43e−6	4.12e−4
		20%	0	0	3.09e−7	6.00e−6
			10^{-3}	0	8.00e−6	3.02e−4

TABLE 3
Rank Determination Results on the Synthetic Datasets with $\rho = 0\%$

$Rank_{\mathrm{m}}(\mathcal{X}_{gt}) = \{R, \overbrace{0.5R, \ldots, 0.5R}^{40}, \overbrace{R, \ldots, R}^{I_3-81} \overbrace{0.5R, \ldots, 0.5R}^{40}\}$

Method			TCTF		BTRTF	
I	R	σ^2	R_{err}	$\frac{\|\|\hat{\mathcal{X}}-\mathcal{X}_{gt}\|\|_F}{\|\|\mathcal{X}_{gt}\|\|_F}$	R_{err}	$\frac{\|\|\hat{\mathcal{X}}-\mathcal{X}_{gt}\|\|_F}{\|\|\mathcal{X}_{gt}\|\|_F}$
100	10	0	0.40	0.7072	0	4.20e−10
		10^{-3}	0.36	0.7075	0	1.41e−5
200	20	0	1.81	0.7071	0	1.36e−10
		10^{-3}	1.82	0.7073	0	6.98e−6

$Rank_{\mathrm{m}}(\mathcal{X}_{gt}) = \{0.5R, \overbrace{R, \ldots, R}^{40}, \overbrace{0.5R, \ldots, 0.5R}^{I_3-81}, \overbrace{R, \ldots, R}^{40}\}$

Method			TCTF		BTRTF	
I	R	σ^2	R_{err}	$\frac{\|\|\hat{\mathcal{X}}-\mathcal{X}_{gt}\|\|_F}{\|\|\mathcal{X}_{gt}\|\|_F}$	R_{err}	$\frac{\|\|\hat{\mathcal{X}}-\mathcal{X}_{gt}\|\|_F}{\|\|\mathcal{X}_{gt}\|\|_F}$
100	10	0	1.04	0.7072	0	4.68e−10
		10^{-3}	1.05	0.7075	0	1.40e−5
200	20	0	1.52	0.7071	0	1.29e−10
		10^{-3}	1.52	0.7073	0	7.02e−6

rank determination results of TCTF and BTRTF on the synthetic datasets with $\rho = 0\%$. BTRTF correctly determines the multi-rank and accurately reconstructs the low-rank component. In contrast, TCTF fails to determine the correct multi-rank and leads to large reconstruction error. This demonstrates the superiority of BTRTF in multi-rank determination.

For comprehensiveness, BTRTF is also tested on the synthetic tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ with $I_1 \neq I_2 \neq I_3$, and still obtains good results. Please refer to the supplementary materials for more details, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2019.2923240.

4.2 Image Denoising

This section considers image denoising for removing random noise from corrupted color images. In this task, clean images are approximated by the low-rank component, while random corruptions are regarded as sparse outliers.

Experimental Setup. We evaluate BTRTF and the competing methods on the Berkeley segmentation datasets (BSD500) [49], which consists of 500 color images represented by $321 \times 481 \times 3$ or $481 \times 321 \times 3$ tensors. We corrupt each color image by setting 10 percent of its elements to random values in [0, 255], so that up to 30 percent pixels are corrupted. Following the common settings, the pixel values of each image are further normalized to [0, 1], and we use peak signal-to-noise ratio (PSNR) to measure the recovery performance. Given the recovered tensor $\hat{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ and the ground truth $\mathcal{X}_{gt} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, PSNR can be computed as follows:

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{\|\mathcal{X}_{gt}\|_\infty^2}{\frac{1}{I_1 I_2 I_3} \|\hat{\mathcal{X}} - \mathcal{X}_{gt}\|_F^2} \right),$$

where $\|\cdot\|_\infty$ is the infinity norm.

Parameter Settings. For RPCA and VBRPCA, we reshape the input tensors into 321×1443 or 481×963 matrices, because they cannot directly deal with tensorial data. For RPCA, VBRPCA, BRTF and KDRSDL, we employ their default parameter settings, which lead to good performance in most cases. For SNN and TNN, we follow the parameter settings suggested in [34], [35]. For BTRTF, we set the initialized multi rank to $Rank_{\mathrm{m}}(\hat{\mathcal{X}}^0) = (150, 150, 150)$, and the convergence criterion to $tol < 10^{-4}$. The sparse component is initialized with $\sigma_0^2 = 10^{-7}$, so that $\hat{\mathcal{S}}^0$ is very close to a zero tensor. This makes BTRTF prefer fitting the input image via the low-rank component rather than the sparse one. Such settings are suitable for image denoising, where only the low-rank component (recovered image) is of interest.

Results and Analysis. Fig. 2 shows the recovered images and PSNR values on 8 sample images of the BSD500 dataset.[2] It can be seen that BTRTF obtains the highest average PSNR value and achieves the best performance on 402 out of the total 500 images from the BSD500 dataset. Specifically, it outperform the second best, TNN, by 1.90 on average. This can be attributed to the BTRTF model in capturing low-tubal-rank structures and the Bayesian framework in estimating sparse outliers. In addition, tensor-based methods such as KDRSDL, TNN and BTRTF often obtain much better results than the matrix-based ones. This is probably because RPCA and VBRPCA are performed on the reshaped images, and fail to capture the correlations across RGB channels. Among tensor-based methods, TNN and BTRTF achieve the top two performance in most cases. This demonstrates that t-SVD based models have an edge over the classical CP and Tucker models in representing color images.

We also compare the average running time of each RPCA method on all 500 images from the BSD500 dataset. From

2. We also provide the normalized mean square error (NMSE) results in the supplementary materials, available online.

(a) Original (b) Noisy (c) RPCA (d) VBRPCA (e) BRTF (f) SNN (g) KDRSDL (h) TNN (i) BTRTF

Image	1	2	3	4	5	6	7	8	Avg. on 500 images	#Best	Avg. Time (s)
RPCA	28.76	25.61	25.44	32.09	27.47	23.34	32.37	27.94	26.00	0	**4.12**
VBRPCA	25.99	21.16	22.55	28.61	23.44	18.89	27.81	23.29	21.79	0	<u>6.56</u>
BRTF	27.73	23.54	25.56	29.92	26.04	27.02	29.10	<u>31.47</u>	24.47	0	54.54
SNN	30.76	28.31	26.79	34.32	30.32	25.20	34.99	29.73	27.91	0	13.44
KDRSDL	30.61	<u>30.72</u>	27.14	33.54	29.17	<u>28.14</u>	32.06	31.38	29.21	32	33.46
TNN	<u>32.75</u>	29.41	<u>28.68</u>	<u>35.50</u>	<u>32.92</u>	27.04	<u>36.51</u>	31.20	<u>29.86</u>	<u>66</u>	15.07
BTRTF	**36.68**	**31.25**	**33.18**	**37.84**	**35.03**	**32.89**	**38.52**	**37.77**	**31.77**	**402**	26.20

(j) PSNR values on the above 8 images (**Best**; <u>Second best</u>).

Fig. 2. Recovery results on the BSD500 dataset. (a) Original image; (b) Corrupted image; (c)-(i) Recovered images by different robust PCA methods; (j) Comparison of PSNR values on the above 8 images. Best viewed in $\times 4$ sized color pdf file.

Fig. 2j, RPCA and VBRPCA are the fastest methods, but they fail to perform well as they cannot fully utilize the tensor structures and tend to obtain an inaccurate low-rank component with the underestimated rank. BTRTF is faster than the non-convex TRPCAs, BRTF and KDRSDL, while slower than the convex methods such as SNN and TNN.

In summary, BTRTF obtains the best recovery results, provides probabilistic outputs, and achieves automatic rank determination, although it takes some computational cost for these benefits. It is worth noting that BTRTF is much faster than BRTF with better performance, despite the fact that both of them are based on variational inference for Bayesian model estimation. This can be attributed to the low-tubal-rank model of BTRTF in better representing color images and enabling the more efficient variational updates via estimating the model parameters in the frequency domain.

4.3 Background Modeling

This section evaluates BTRTF on the background modeling problem, which aims at separating foreground objects and background from a given video sequence. We consider videos recorded by stationary cameras, which are common in video surveillance. In this case, background components of different frames are highly correlated, and thus can be well characterized by low-rank models. On the other hand, foreground objects generally change a lot and can be considered as sparse outliers.

Experimental Setup. We conduct experiments on 15 videos from the I2R [50] and CDnet [51] datasets. The I2R dataset consists of 9 real-world videos (Bootstrap, Campus, Curtain, Escalator, Fountain, Hall, Lobby, ShoppingMall, WaterSurface) in different scenarios including static background, dynamic background, and slow object movement. For each video, 20 frames are labeled with the ground truth. The CDnet dataset consists of 31 videos grouped as 6 categories representing a variety of motion and change detection challenges, where the foreground objects are well annotated for each frame. We test all 6 videos (Boats, Canoe, Fall, Fountain01, Fountain02, Overpass) in the dynamic background category, which is one of the most difficult categories for mounted camera object detection. Since most videos in the I2R and CDnet datasets have different sizes and frame numbers, we extract 300 frames and downsample them to around 160×180, so that the input tensors have similar sizes ($160 \times 180 \times 300$).

TABLE 4
Summary of Precision, Recall, and F-Measure on the I2R and CDnet Datasets (**Best**; Second Best)

Videos	RPCA			VBRPCA			BRTF			SNN			KDRSDL			TNN			BTRTF		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
Bootstrap	0.51	0.26	0.34	0.34	0.30	0.32	0.73	0.42	0.53	0.61	0.33	0.43	0.79	0.45	**0.57**	0.79	0.42	0.55	0.55	0.54	0.55
Campus	0.09	0.29	0.13	0.11	0.28	0.16	0.51	0.61	0.55	0.14	0.67	0.22	0.16	0.27	0.20	0.52	0.83	**0.64**	0.87	0.47	0.61
Curtain	0.52	0.46	0.59	0.40	0.44	0.42	0.72	0.49	0.58	0.64	0.49	0.55	0.71	0.67	0.69	0.88	0.59	0.70	0.94	0.88	**0.91**
Escalator	0.38	0.43	0.40	0.35	0.42	0.38	0.77	0.62	0.69	0.47	0.51	0.50	0.58	0.30	0.39	0.73	0.73	**0.73**	0.85	0.64	**0.73**
Fountain	0.16	0.33	0.22	0.16	0.34	0.22	0.58	0.75	0.66	0.25	0.53	0.34	0.26	0.93	0.40	0.32	0.85	0.47	0.86	0.79	**0.82**
Hall	0.25	0.49	0.33	0.26	0.55	0.35	0.60	0.56	0.58	0.34	0.59	0.43	0.48	0.73	0.58	0.65	0.63	**0.64**	0.71	0.56	0.63
Lobby	0.11	0.24	0.15	0.06	0.18	0.09	0.55	0.50	0.52	0.17	0.35	0.23	0.75	0.89	**0.82**	0.83	0.62	0.71	0.82	0.83	**0.82**
ShoppingMall	0.45	0.44	0.44	0.30	0.40	0.34	0.74	0.73	0.73	0.57	0.58	0.58	0.73	0.82	0.77	0.80	0.78	**0.79**	0.70	0.76	0.73
WaterSurface	0.24	0.20	0.22	0.27	0.25	0.26	0.56	0.27	0.36	0.29	0.26	0.28	0.30	0.31	0.30	0.46	0.29	0.36	0.98	0.81	**0.89**
Boats	0.71	0.37	0.49	0.95	0.53	0.68	0.79	0.29	0.42	0.45	0.44	0.45	0.63	0.19	0.30	0.55	0.12	0.19	0.99	0.54	**0.70**
Canoe	0.33	0.44	0.38	0.47	0.64	0.54	0.55	0.37	0.44	0.31	0.52	0.38	0.12	0.46	0.20	0.29	0.27	0.28	0.99	0.61	**0.75**
Fall	0.25	0.21	0.23	0.20	0.25	0.22	0.69	0.28	0.40	0.52	0.35	0.42	0.49	0.55	0.52	0.75	0.40	0.52	0.89	0.86	**0.88**
Fountain01	0.02	0.23	0.04	0.02	0.31	0.03	0.03	0.33	**0.06**	0.02	0.27	0.03	0.02	0.50	0.03	0.03	0.39	0.05	0.02	0.37	0.04
Fountain02	0.10	0.48	0.17	0.05	0.54	0.10	0.41	0.66	**0.51**	0.26	0.56	0.35	0.07	0.88	0.13	0.19	0.72	0.31	0.19	0.74	0.30
Overpass	0.38	0.27	0.32	0.40	0.37	0.38	0.77	0.40	0.52	0.39	0.46	0.42	0.63	0.65	0.64	0.87	0.42	0.57	0.93	0.61	**0.74**
Average	0.30	0.34	0.30	0.29	0.39	0.30	0.60	0.49	0.47	0.36	0.46	0.37	0.45	0.57	0.44	0.58	0.54	0.50	0.75	0.67	**0.67**

For quantitative evaluation, we compare the estimated sparse component (foreground) $\hat{\mathcal{S}}$ with the ground truth $\mathcal{S}_{gt}$, and regard this as a classification problem. Following the standard settings [11], [52], we evaluate the background subtraction results by precision, recall, and F-measure, which are defined as

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}, \text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}},$$
$$\text{F-measure} = 2\frac{\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}},$$

where TP, FP, and FN represent the number of true positives, false positives, and false negatives, respectively. The higher these three measurements, the better the performance is.

Parameter Settings. For RPCA and VBRPCA, each video is first unfolded along the time dimension into the matrix of size around 28800×300, and then fed into the corresponding RPCA methods. Since there is no training/test partition for the background modeling problem, we empirically select, if necessary, the tuning parameters for the competing methods, so that they can perform well on most video sequences. For BTRTF, we initialize σ_0^2 to a large value 10^7. This allows BTRTF to capture outliers of large magnitude (foreground objects), and often leads to better foreground/background separation. The initialized multi-rank for BTRTF is set to $Rank_{\text{m}}(\hat{\mathcal{X}}^0) = (\min(I_1, I_2) - 1 \ldots, \min(I_1, I_2) - 1) \in \mathbb{R}^{300}$ for the $I_1 \times I_2 \times 300$ video sequence.

4.3.1 Quantitative Evaluation

Table 4 shows the foreground detection results on the I2R and CDnet datasets. It can be seen that BTRTF achieves the top two performance in most cases, and obtains the best average results in precision, recall, and F-measure. TNN is the second best method, while it is still significantly worse than BTRTF by 0.17 in F-measure on average. These demonstrate: 1) t-SVD based methods such as BTRTF and TNN are effective in background reconstruction by exploiting the correlations along the

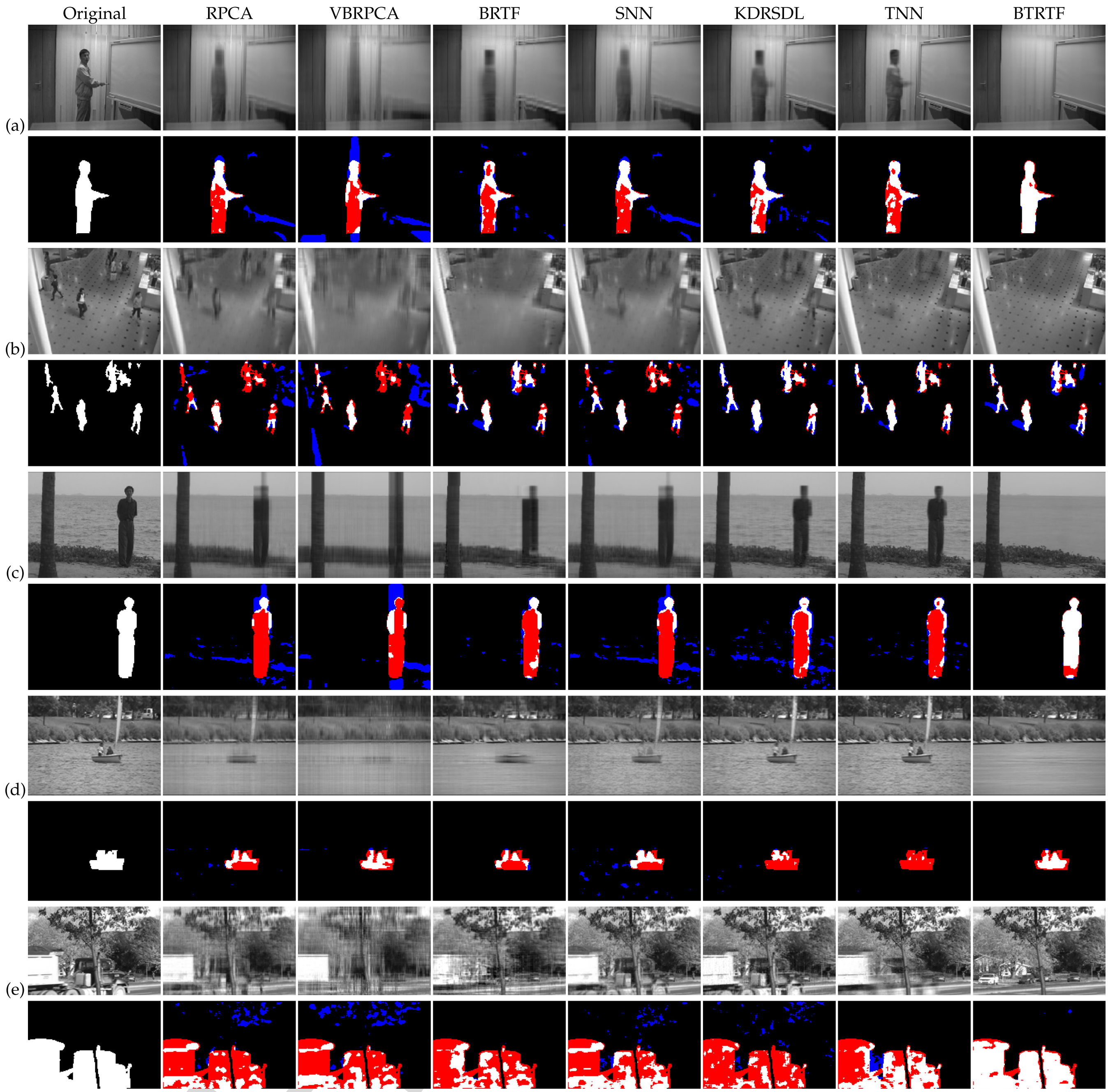


Fig. 3. Detected background and foreground masks on five videos from the I2R and CDnet datasets. (a) Curtain, (b) ShoppingMall, (c) WaterSurface, (d) Boats, (e) Fall. For each video, there are two rows corresponding to background and foreground masks. Blue and red regions in the learned masks indicate false positives and false negatives, respectively.

time dimension. 2) Armed with the Bayesian framework, BTRTF is more advantageous in separating foreground objects especially for those with slow movement. It is worth noting that Fountain01 consists of significant dynamic background elements such as intense water flow, while the foreground objects are relatively small. This makes foreground/background separation much more challenging. As a result, all the methods fail to perform well on this video.

4.3.2 Visual Quality

To visualize the background modeling results, we select five videos from the I2R (Curtain, ShoppingMall, WaterSurface) and CDnet (Boats, Fall) datasets, and show the background and foreground masks learned by different RPCA methods in Fig. 3. It can be seen that only BTRTF obtains coherent foreground masks while constructing clean background in all the cases. Matrix-based methods (RPCA and VBRPCA) can only obtain blurry background with severe ghosting effects. This is because they have to first reshape the input tensors into matrices and thus loss some structural information. On the other hand, tensor-based methods, especially TNN and BTRTF, obtain cleaner background with much more details, showing the capability of t-SVD based models in characterizing low-rank data information.

From (a) Curtain and (c) WaterSurface, all the methods except BTRTF fail to separate the person, who walks through the camera and stands for a while, from the background. This is also the case for (d) Boats and (e) Fall, where the boat moves slowly and the truck is too long to quickly pass through the camera. Because of the slow motion of these foreground objects, the competing methods tend to overfit the low-rank component (background), and thus lead to more false negatives (the red regions) in the foreground masks. In contrast, BTRTF not only completely separates the foreground objects in all the cases, but also has less false positives (the blue regions) by filtering out many dynamic textures, e.g., fluctuations of waves and swaying of leaves. From (b) ShoppingMall, we observe ghosting effects in the background learned by KDRSDL and TNN, although they obtains higher F-measure than BTRTF. BRTF removes not only all the person but also many details such as patterns on the floor from the background. Only our BTRTF achieves good performance on both foreground detection and background construction.

Based on the visual and quantitative results, we summarize that 1) the performance of matrix-based methods is not good enough in background modeling, since they cannot utilize the informative tensor structures. 2) By exploiting the correlations along the time dimension, the low-tubal-rank model can construct the background with higher quality and more details than the classical CP and Tucker models. 3) BTRTF is superior to the competing methods in dealing with dynamic background elements and slow objective movement. This can be attributed to both the more expressive modeling power of the low-tubal-rank model in representing the background and the Bayesian framework in implicitly balancing the low-rank and sparse components.

5 Conclusion and Future Work

In this paper, we have proposed BTRTF, a fully Bayesian method for robust tensor factorization. By incorporating low-tubal-rank structures and a generalized ARD prior into the Bayesian framework, BTRTF features more expressive modeling power than classical Tucker and CP based approaches, automatic multi-rank determination, and implicit trade-off between the low-rank and sparse components. For model estimation, we have developed an efficient variational inference algorithm by updating the model parameters in the frequency domain. Experiments on both synthetic and real-world datasets demonstrated that BTRTF is effective in determining the multi-rank, and outperforms state-of-the-art RPCA methods in image denoising and background modeling.

Since the t-product, tubal rank, and multi-rank are originally defined on third-order tensors [18], we consider dealing with 3D data only in this work. Recently, there have been some attempts to generalize the t-product and t-SVD for higher-order tensors [25]. Along this line, we may also define higher-order extensions of the tubal rank and multi-rank. With these definitions, the BTRTF model along with the variational inference scheme can be naturally generalized for higher-order tensors, which could be the future work.

Acknowledgments

We thank Dr. Jian Lou for helpful discussions. This work was supported by the National Natural Science Foundation of China under Grants: 61672444 and 61272366 and in part by the Faculty Research Grant of Hong Kong Baptist University (HKBU) under Project FRG2/17-18/082, the KTO Grant of HKBU under Project MPCF-004-2017/18, and the SZSTI under Grant JCYJ20160531194006833.

References

[1] Y. Koren, “Factorization meets the neighborhood: A multifaceted collaborative filtering model,” in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 426–434.
[2] M. Pang, Y.M. Cheung, B.H. Wang, and J. Lou, “Synergistic Generic Learning for Face Recognition from a Contaminated Single Sample per Person,” *IEEE Trans. Inf. Forensics Secur.*, doi: 10.1109/TIFS.2019.2919950, 2019.
[3] M. Mørup, “Applications of tensor (multiway array) factorizations and decompositions in data mining,” *Wiley Interdisciplinary Rev.: Data Mining Knowl. Discovery*, vol. 1, no. 1, pp. 24–40, 2011.
[4] I. Jolliffe, *Principal Component Analysis*, 2nd ed. Berlin, Germany: Springer, 2002.
[5] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, “Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 2080–2088.
[6] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis?” *J. ACM*, vol. 58, no. 3, 2011, Art. no. 11.
[7] X. Ding, L. He, and L. Carin, “Bayesian robust principal component analysis,” *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3419–3430, Dec. 2011.
[8] N. Wang and D.-Y. Yeung, “Bayesian robust matrix factorization for image and video processing,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1785–1792.
[9] H. Ji, C. Liu, Z. Shen, and Y. Xu, “Robust video denoising using low rank matrix completion,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1791–1798.
[10] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, “Robust recovery of subspace structures by low-rank representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.
[11] X. Zhou, C. Yang, and W. Yu, “Moving object detection by detecting contiguous outliers in the low-rank representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 597–610, Mar. 2013.
[12] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
[13] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, *Multilinear Subspace Learning: Dimensionality Reduction of Multidimensional Data*. Boca Raton, FL, USA: CRC Press, 2013.
[14] L. R. Tucker, “Some mathematical notes on three-mode factor analysis,” *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.
[15] R. A. Harshman, “Foundations of the PARAFAC procedure: Models and conditions for an “explanatory” multimodal factor analysis,” *UCLA Work. Papers Phonetics*, vol. 16, pp. 1–84, 1970.
[16] J. D. Carroll and J.-J. Chang, “Analysis of individual differences in multidimensional scaling via an N-way generalization of Eckart-Young decomposition,” *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.
[17] M. E. Kilmer and C. D. Martin, “Factorization strategies for third-order tensors,” *Linear Algebra Appl.*, vol. 435, no. 3, pp. 641–658, 2011.
[18] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, “Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging,” *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 1, pp. 148–172, 2013.
[19] C. Mu, B. Huang, J. Wright, and D. Goldfarb, “Square deal: Lower bounds and improved relaxations for tensor recovery,” in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 73–81.
[20] D. Goldfarb and Z. Qin, “Robust low-rank tensor recovery: Models and algorithms,” *SIAM J. Matrix Anal. Appl.*, vol. 35, no. 1, pp. 225–253, 2014.
[21] J. Liu, P. Musialski, P. Wonka, and J. Ye, “Tensor completion for estimating missing values in visual data,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.
[22] M. Bahri, Y. Panagakis, and S. Zafeiriou, “Robust Kronecker-decomposable component analysis for low-rank modeling,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3372–3381.
[23] Q. Shi, Y.-M. Cheung, and Q. Zhao, “Feature extraction for incomplete data via low-rank tucker decomposition,” in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2017, pp. 564–581.

[24] Q. Shi, Y. Cheung, Q. Zhao, and H. Lu, "Feature extraction for incomplete data via low-rank tensor decomposition with feature regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1803–1817, Jun. 2019.

[25] C. D. Martin, R. Shafer, and B. LaRue, "An order-p tensor factorization with applications in imaging," *SIAM J. Sci. Comput.*, vol. 35, no. 1, pp. A474–A490, 2013.

[26] J. Landsberg, *Tensors: Geometry and Applications*, vol. 128. Providence, RI, USA: American Mathematical Society, 2011.

[27] C. J. Hillar and L.-H. Lim, "Most tensor problems are NP-hard," *J. ACM*, vol. 60, no. 6, 2013, Art. no. 45.

[28] Q. Zhao, G. Zhou, L. Zhang, A. Cichocki, and S.-I. Amari, "Bayesian robust tensor factorization for incomplete multiway data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 736–748, Apr. 2016.

[29] X. Chen, Z. Han, Y. Wang, Q. Zhao, D. Meng, L. Lin, and Y. Tang, "A generalized model for robust tensor factorization with noise modeling by mixture of Gaussians," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5380–5393, Nov. 2018.

[30] X.-Y. Liu, S. Aeron, V. Aggarwal, and X. Wang, "Low-tubal-rank tensor completion using alternating minimization," arXiv:1610.01690, 2016.

[31] O. Semerci, N. Hao, M. E. Kilmer, and E. L. Miller, "Tensor-based formulation and nuclear norm regularization for multienergy computed tomography," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1678–1693, Apr. 2014.

[32] Z. Zhang and S. Aeron, "Exact tensor completion using t-SVD," *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1511–1526, Mar. 2017.

[33] P. Zhou, C. Lu, Z. Lin, and C. Zhang, "Tensor factorization for low-rank tensor completion," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1152–1163, Mar. 2018.

[34] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5249–5257.

[35] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis with a new tensor nuclear norm," *IEEE Trans. Pattern Anal. Mach. Intell.*, doi: 10.1109/TPAMI.2019.2891760, 2019.

[36] R. M. Neal, *Bayesian Learning for Neural Networks*, vol. 118. Berlin, Germany: Springer, 2012.

[37] M. J. Beal, et al., *Variational Algorithms for Approximate Bayesian Inference*. London, U.K.: Univ. of London, 2003.

[38] J. Winn and C. M. Bishop, "Variational message passing," *J. Mach. Learn. Res.*, vol. 6, no. Apr., pp. 661–694, 2005.

[39] W. Chu and Z. Ghahramani, "Probabilistic models for incomplete multi-dimensional arrays," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2009, pp. 89–96.

[40] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with Bayesian probabilistic tensor factorization," in *Proc. SIAM Int. Conf. Data Mining*, 2010, pp. 211–222.

[41] P. Rai, Y. Wang, S. Guo, G. Chen, D. Dunson, and L. Carin, "Scalable Bayesian low-rank decomposition of incomplete multiway tensors," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1800–1808.

[42] Z. Xu, F. Yan, and Y. Qi, "Infinite Tucker decomposition: Nonparametric Bayesian models for multiway data analysis," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 1675–1682.

[43] Z. Xu, F. Yan, and Y. Qi, "Bayesian nonparametric models for multiway data analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 2, pp. 475–487, Feb. 2015.

[44] R. Shi and J. Kang, "Thresholded multiscale Gaussian processes with application to Bayesian feature selection for massive neuroimaging data," arXiv:1504.06074, 2015.

[45] S. A. Khan, E. Leppäaho, and S. Kaski, "Bayesian multi-tensor factorization," *Mach. Learn.*, vol. 105, no. 2, pp. 233–253, 2016.

[46] S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos, "Sparse Bayesian methods for low-rank matrix estimation," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 3964–3977, Aug. 2012.

[47] B. Huang, C. Mu, D. Goldfarb, and J. Wright, "Provable low-rank tensor recovery," *Optimization-Online*, vol. 4252, 2014, Art. no. 2.

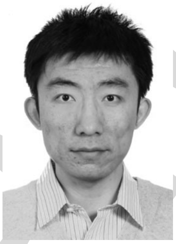
[48] P. Zhou and J. Feng, "Outlier-robust tensor PCA," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3938–3946.

[49] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[50] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1459–1472, Nov. 2004.

[51] N. Goyette, P. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changedetection.net: A new change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 1–8.

[52] W. Hu, Y. Yang, W. Zhang, and Y. Xie, "Moving object detection using tensor-based low-rank and saliently fused-sparse decomposition," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 724–737, Feb. 2017.



Yang Zhou received the PhD degree from the Department of Computer Science, Hong Kong Baptist University, Hong Kong, in 2019. He is currently an assistant professor with the School of Computer Science and Software Engineering, East China Normal University, Shanghai, China. His research interests include probabilistic modeling, tensor analysis, and machine learning.

Yiu-Ming Cheung (F'18) received the PhD degree from the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong. He is currently a full professor with the Department of Computer Science, Hong Kong Baptist University, Hong Kong. His research interests include machine learning, pattern recognition, visual computing, and optimization. He serves as an associate editor of the *IEEE Transactions on Neural Networks and Learning Systems*, the *IEEE Transactions on Cybernetics*, the *Pattern Recognition*, to name a few. He is a fellow of the IEEE, IET, BCS, and RSA, and IETI distinguished fellow. More details can be found at: http://www.comp.hkbu.edu.hk/∼ymc.