Bayesian Robust Tensor Ring Decomposition for Incomplete Multiway Data
Zhenhao Huang, Yuning Qiu, Xinqi Chen, Weijun Sun, and Guoxu Zhou

Abstract—Robust tensor completion (RTC) aims to recover a low-rank tensor from its incomplete observations with outlier corruption. The recently proposed tensor ring (TR) model has demonstrated superiority in solving the TR problem. However, the methods using the TR model either require a preassigned TR rank or aggressively pursue the minimum TR rank, where the latter often leads to biased solutions in the presence of noise. To tackle these bottlenecks, a Bayesian robust TR decomposition (BRTR) method is proposed to give a more accurate solution for the RTC problem, which can avoid excessive selection of the TR rank and penalty parameters. A variational Bayesian (VB) algorithm is developed to infer the probability distribution of posteriors. During the learning process, BRTR can prune off zero components of core tensors, resulting in automatic TR rank determination. Extensive experiments show that BRTR can achieve significantly improved performance than other state-of-the-art methods.

Index Terms—Automatic tensor ring (TR) rank determination, probability distribution, robust tensor completion (RTC), TR, variational Bayesian (VB) algorithm.

I. INTRODUCTION

ANY data in real-world applications are naturally represented as tensors, also known as multidimensional arrays. For instance, a color image forms a third-order tensor with the size of width × height × channel, and a color video clip forms a fourth-order one with an additional dimension of time frame. Tensor data-based learning methods, including CANDECOMP/PARAFAC (CP) decomposition [1] and Tucker decomposition [2], [3], [4], [5], [6], have significant advantages of preserving multilinear structure information of tensor data [7], [8], and have been successfully applied to data mining, clustering and classification [9], [10], brain signal processing [11], object detection [12], [13] and neural architecture search [14].

Most of the time, the observed data are incomplete due to failures in the data collection process or other abnormal conditions. Therefore, tensor completion (TC), which aims to predict missing entries from partial observations, has gained remarkable attention in recent years and has been applied to many fields, such as color image and video recovery [15], [16], [17], knowledge graph completion [18], feature selection [19], [20], and link prediction [21], [22]. In the meanwhile, CP model was first considered for TC due to its simplicity, such as CP weighted optimization (CP-WORT) [23] and CP nonlinear least squares (CPNLSs) [24]. Moreover, to avoid manual rank selection, Zhao et al. introduced a Bayesian approach for the CP model where the CP rank can be automatically determined. Nevertheless, the CP model has some fundamental limitations, including that the CP decomposition of a given tensor can be ill-posed, and its computation often suffers from serious convergence issues [25]. Hence, TC using the Tucker model gained growing interests, such as tucker weighted optimization (Tucker-WORT) [26], Tucker-based overlapped nuclear norm optimization [27], and scalable Tucker factorization method [28], [29]. No matter which model is employed, the associated low-rank feature is the key to achieving satisfactory recovery performance. Tucker model-based methods are often accused that the core tensor grows exponentially and is inefficient to deal with high-order tensors.

Recently, some new tensor ranks have been proposed to solve the TC problem, including tensor tubal rank [30], [31], [32], [33], [34], tensor train (TT) rank [35], and tensor ring (TR) rank [36], [37]. Among them, TT and TR models have achieved remarkable performance [38], [39], [40], [41], [42], [43], [44], [45], [46]. For the TT model, the original tensor is approximated by the tensor contraction of a series of third-order tensors (the first and the last one boil down to matrices). Hence, TT model is more suitable for higher-order tensors and a lot of TT-based TC methods were proposed, such as...
simple low-rank TC via TT (SiLRTC-TT), parallel matrix factorization via TT (TMac-TT) [47], and matrix factorization based on TT rank and total variation (MF-TTTV) [48]. As analyzed in [36], TT model has a strict rank requirement on the border factors, resulting in an unbalanced rank scheme and hampered representation ability. TR model relaxes the strict restriction over TT, which is able to give a more balanced representation. Hence, it is more suitable for arbitrarily higher-order tensors and can better capture the low-rank nature of data. Based on the TR model, Yuan et al. [49] proposed a TR low-rank factors (TRLRF) method by minimizing the nuclear norm of all factor tensors. Yu et al. [50] defined a new circular unfolding operation and proposed a TR nuclear norm minimization (TRNNM) method. Long et al. [51] proposed a Bayesian low-rank TR method for the TC problem. These works of literature show that TR-based TC methods often have better-recovery performance than other methods.

All of the above methods perform well if the partially observed data are free from noise corruption. However, real-life data is not only prone to missing values but also susceptible to noise interference, such as in social media and hyperspectral datasets [52], [53], [54], [55]. Traditional TC-based methods fall short due to their sensitivity to noise when handling with these data. Hence, it is of great interest to predict the intrinsic tensor from noisy and missing observations, which is also known as the robust TC (RTC) problem. Although RTC is more practical and valuable, there is limited work on it. Zhao et al. [56] proposed a Bayesian robust tensor factorization (BRTF) method, which achieved a significant improvement in processing noisy data under Bayesian treatment. Nevertheless, it still suffers from the same issues as the CP model. Song et al. [57] solved RTC problem via transformed tubal nuclear norm (TTNN). TTNN used a new unitary transformation that had a better-recovery performance compared with the use of the Fourier transformation. But it can only serve third-order tensor data and requires a hyperparameter to be manually adjusted. Huang et al. [58] proposed a robust low-rank TR completion (RTRC) method with recovery guarantee. However, RTRC has to unfold a tensor into a matrix, which breaks the structure of the tensor TR decomposition, and it also requires manual adjustment of the penalty parameter.

Generally, existing TR-based RTC methods either require a preassigned TR rank or aggressively pursue the minimum TR rank. For methods that predefine the TR rank, setting an excessively high-TR rank may result in overfitting, where the model excessively fits noise in the data instead of capturing genuine patterns. Conversely, predefining a too low-TR rank may lead to underfitting, with the model failing to capture all critical information in the data [7]. In the case of models pursuing the minimum TR rank, a parameter is required to balance low-rank and sparse components, where the parameter needs to be given in advance and must meet the conditions for precise separation. However, in scenarios characterized by high-noise intensity, the selection of such a parameter is challenging. Under these circumstances, an excessive pursuit of rank minimization may lead to biased solutions, potentially resulting in inaccurate solutions [59], [60], [61], [62], [63].

To tackle these bottlenecks, in this article, we propose a Bayesian robust TR decomposition (BRTR) method for the RTC problem under a fully Bayesian treatment, as shown in Fig. 1. The BRTR method is modeled by multiplicative interactions of core tensors and an additive sparse noise tensor, upon which sparsity-inducing priors are placed. In addition, we develop a variational Bayesian (VB) algorithm for the inference of posteriors. During the learning process, BRTR can prune off zero components of core tensors, resulting in automatic TR rank determination. Extensive experiments show...
that BRTR can achieve significantly improved performance than other state-of-the-art (SOTA) methods.

The remainder of this article is organized as follows. Section II clarifies some basic notations and definitions for TR decomposition. In Section III, we introduce the BRTR method under the Bayesian approach and give a VB algorithm for model learning. In Section IV, we perform extensive experiments to evaluate the proposed BRTR method in competition with other methods. Finally, the conclusion and future work are drawn in Section V.

II. NOTATIONS AND PRELIMINARIES

A. Notations

In this section, we introduce some basic definitions and their description, which are summarized in Table I. In particular, we introduce some concepts about probability theory. \( p(\cdot) \) and \( q(\cdot) \) denote the probability distributions. \( \mathcal{N}(\mu, \sigma^2) \) denotes a Gaussian distribution with mean \( \mu \) and variance \( \sigma^2 \). \( \text{Ga}(\alpha, b) = (b^\alpha x^{\alpha-1} e^{-bx})/\Gamma(\alpha) \) denotes a gamma distribution where \( \Gamma(\alpha) \) is a gamma function.

Definition 1 (Matrix Normal Distribution): Given a matrix \( \mathbf{A} \in \mathbb{R}^{I \times K} \), its matrix normal distribution is defined as \( \mathbf{A} \sim \mathcal{MN}(\mathbf{\Lambda}, \mathbf{U}, \mathbf{V}) \), which can be further expressed by

\[
p(\mathbf{A}|\mathbf{\Lambda}, \mathbf{U}, \mathbf{V}) = \frac{\exp\left(-\frac{1}{2} \text{Tr}(\mathbf{V}^{-1}(\mathbf{A} - \mathbf{\Lambda})^\top\mathbf{U}^{-1}(\mathbf{A} - \mathbf{\Lambda}))\right)}{(2\pi)^{IK/2}|\mathbf{V}|^{1/2}|\mathbf{U}|^{1/2}}
\]

where \( \mathbf{\Lambda} \) denotes the mean, \( \mathbf{U} \) and \( \mathbf{V} \) denote diagonal covariance matrices. And we have \( \text{vec}(\mathbf{A}) \sim \mathcal{N}(\text{vec}(\mathbf{\Lambda}), \mathbf{U} \otimes \mathbf{V}) \).

B. Tensor Ring Decomposition

Given an \( N \)-th order tensor \( \mathbf{L} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \), the TR model decomposes it into a sequence of latent tensors \( \mathbf{Z}^{(n)} \in \mathbb{R}^{R_0 \times \cdots \times R_n} \) \( \forall n \in \{1, N\} \), that is

\[
\mathbf{L}(i_1, i_2, \ldots, i_N) = \text{Tr}(\mathbf{Z}_{i_1}^{(1)} \mathbf{Z}_{i_2}^{(2)} \cdots \mathbf{Z}_{i_N}^{(N)})
\]

where \( \text{Tr}(\cdot) \) denotes the trace operation, and \( \mathbf{Z}_{i_n}^{(n)} \) denotes the \( i_n \)-th slice of \( \mathbf{Z}^{(n)} \). \( \mathbf{Z}^{(0)} = \mathbf{L} \) denotes the TR rank with \( R_0 = R_N \). For simplicity, the TR decomposition of tensor \( \mathbf{L} \) is abbreviated by \( \mathbf{L} = \mathcal{R}(\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}, \ldots, \mathbf{Z}^{(N)}) \).

TABLE I

| Notation | Description |
|----------|-------------|
| \( a, \mathbf{a}, \mathbf{A} \) | A scalar, vector, matrix, tensor, respectively |
| \( \mathbf{A}_{1 \cdots i_N} \) | The \((i_1, \ldots, i_N)\) entry of tensor \( \mathbf{A} \) |
| \( \text{vec}(\mathbf{A}) \) | The vectorization of matrix \( \mathbf{A} \) |
| \( \text{Tr}(\mathbf{A}) \) | The trace operation of square matrix \( \mathbf{A} \) |
| \( \|\mathbf{A}\|_F \) | The Frobenius norm of tensor \( \mathbf{A} \) |
| \( \langle \mathbf{A}, \mathbf{B} \rangle \) | The tensor inner product of two tensors \( \mathbf{A} \) and \( \mathbf{B} \) |
| \( \mathbf{A} \otimes \mathbf{B} \) | The Hadamard product of two tensors \( \mathbf{A} \) and \( \mathbf{B} \) |
| \( \mathbf{A} \otimes \mathbf{B} \) | The Kronecker product of two tensors \( \mathbf{A} \) and \( \mathbf{B} \) |

III. BAYESIAN ROBUST TENSOR RING DECOMPOSITION

In this section, we introduce the proposed Bayesian robust TR model and develop a VB inference algorithm.

A. Model Specification

Given an incomplete tensor \( \mathbf{Y}_\Omega \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \), its observed entries can be defined by \( \{\mathbf{Y}_{i_1 \cdots i_N}|(i_1, i_2, \ldots, i_N) \in \Omega\} \), where \( \Omega \) is a set of indices. In the noisy measurement, we assume that \( \mathbf{Y} \) can be divided into three parts

\[
\mathbf{Y} = \mathbf{L} + \mathbf{S} + \mathbf{M}, \text{ s.t. } \mathcal{P}_\Omega(\mathbf{L} + \mathbf{S} + \mathbf{M}) = \mathcal{P}_\Omega(\mathbf{Y})
\]

where \( \mathbf{L} \) denotes a low-rank component that can be represented using (2), \( \mathbf{S} \) is a sparse tensor that denotes gross noise, and \( \mathbf{M} \) is isotropic Gaussian noise. \( \mathcal{P}_\Omega \) is a linear projection such that the entries in the set \( \Omega \) are kept while the remaining entries are missing. Since \( \mathbf{L} \) is assumed to be a TR structure, a generative model from observed data under a probabilistic framework is defined by

\[
p(\mathbf{Y}_\Omega|\mathbf{Z}^{(n)}_{i_1 \cdots i_N}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{Y}_{i_1 \cdots i_N}|\mathcal{R}(\mathbf{Z}_{i_1}^{(1)}, \ldots, \mathbf{Z}_{i_N}^{(N)}))
\]

where \( \tau \) denotes the noise level and \( \mathcal{O} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \) is the indicator tensor whose entry \( \mathcal{O}_{i_1 \cdots i_N} = 1 \) if \((i_1, i_2, \ldots, i_N) \in \Omega\) or 0 otherwise. Since each core tensor \( \mathbf{Z}^{(n)} \) shares dimensionality with the other two neighbor core tensors, it requires two hyperparameters \( u_{i_1}^{(n)}, u_{i_2}^{(n)} \) to involve in the component in \( \mathbf{Z}^{(n)} \). Hence, for each entry of the core tensor \( \mathbf{Z}^{(n)} \), we
place a Gaussian distribution with zero mean and noise level 

\[ u_{r_{n-1}}^{(n)} \ast u_{r_{n}}^{(n)} \] 

over them

\[
p\left( Z^{(n)} | u^{(n-1)}, u^{(n)} \right) 
= \prod_{i_{n}} \prod_{r_{n-1}} \prod_{r_{n}} \mathcal{N}\left( Z^{(n)} (r_{n-1} - i_{n}, r_{n}) | 0, \left( u_{r_{n-1}}^{(n-1)} \ast u_{r_{n}}^{(n)} \right)^{-1} \right) 
\forall n \in [1, N]
\] 

(8)

where \( u^{(n)} = [u_{1}^{(n)}, \ldots, u_{R_{n}}^{(n)}] \). Further, we can rewrite (8) in a concised form as

\[
p\left( Z^{(n)} | u^{(n-1)}, u^{(n)} \right) 
= \prod_{i_{n}} \mathcal{N}\left( \text{vec}(Z_{n}(i_{n})) | 0, \left( U^{(n)} \otimes U^{(n-1)} \right)^{-1} \right). 
\forall n \in [1, N]
\] 

(9)

where \( U^{(n)} = \text{diag}(u^{(n)}) \) denotes the inverse covariance matrix.

In addition, we place a gamma distribution over each element of \( u^{(n)} \), that is

\[
p\left( u^{(n)} | c^{(n)}, d^{(n)} \right) 
= \prod_{i_{n}} \text{Ga}\left( u_{r_{n}}^{(n)} | c_{r_{n}}^{(n)}, d_{r_{n}}^{(n)} \right) 
\forall n \in [1, N]
\] 

(10)

where parameters \( c^{(n)} = [c_{1}^{(n)}, \ldots, c_{R_{n}}^{(n)}] \) and \( d^{(n)} = [d_{1}^{(n)}, \ldots, d_{R_{n}}^{(n)}] \).

Each entry of the sparse tensor \( S \) is modeled as a Gaussian distribution with zero mean and noise level \( \eta \), which can be represented by

\[
p(S_{\Omega} | \eta) 
= \prod_{i_{1} \cdots i_{N}} \mathcal{N}\left( S_{i_{1} \cdots i_{N}} | 0, \eta_{i_{1} \cdots i_{N}}^{-1} \right) O_{i_{1} \cdots i_{N}}^{-1}. 
\] 

(11)

For the noise level \( \eta \), we place a gamma distribution over each entry of it, that is

\[
p(\eta) = \prod_{i_{1} \cdots i_{N}} \text{Ga}\left( \eta_{i_{1} \cdots i_{N}} | a_{i_{1} \cdots i_{N}}^{\eta}, b_{i_{1} \cdots i_{N}}^{\eta} \right). 
\] 

(12)

In fact, the sparse tensor \( S \) can be viewed as an infinite zero-mean Gaussian distribution with mixture coefficients drawn from a gamma distribution, as known as a student-t distribution. If the noise level \( \eta \) goes to infinity, the corresponding element in \( S \) is forced to be exactly zero, thereby leading to a sparse property. Further, to complete the BRTR model, we place a gamma distribution over the noise level \( \tau \), that is

\[
p(\tau) = \text{Ga}(\tau | a_{\tau}^{0}, b_{\tau}^{0}). 
\] 

(13)

Finally, we can build a full probabilistic Bayesian robust TR model, as shown in Fig. 2. The joint distribution of BRTR model is defined by

\[
p(Y_{\Omega}, H) = p\left( Y_{\Omega} | Z^{(1)}_{n=1}, S_{\Omega}, \tau \right) 
\times \prod_{n=1}^{N} p\left( Z^{(n)} | u^{(n-1)}, u^{(n)} \right) \prod_{n=1}^{N} p(u^{(n)}) p(S_{\Omega} | \eta) p(\eta) p(\tau) 
\] 

(14)

where \( H = \{ (Z^{(n)})_{n=1}^{N}, (u^{(n)})_{n=1}^{N}, S_{\Omega}, \eta, \tau \} \) denotes all unknown parameters. To infer the missing entries \( Y_{\Omega} \) through known entries, the predictive distribution is expressed by

\[
p(Y_{\Omega} | Y_{\Omega}) 
= \int p(Y_{\Omega} | H)p(H | Y_{\Omega})dH
\] 

(15)

where \( p(H | Y_{\Omega}) = p(H, Y_{\Omega}) / \int p(H, Y_{\Omega})dH \).

\[ B. \text{Variational Bayesian Inference for Model Learning} \]

For the BRTR model, it is difficult to obtain an exact solution. Inspired by [65], we develop a VB inference algorithm for the above problem, which provides a closed-form posterior approximation and is computationally efficient. VB inference aims to find a distribution \( q(H) \) that approximates the true distribution \( p(H | Y_{\Omega}) \) by minimizing the Kullback–Leibler (KL) divergence between \( q(H) \) and \( p(H | Y_{\Omega}) \), which can be expressed by

\[
\text{KL}(q(H) \| p(H | Y_{\Omega})) 
= \int q(H) \ln \left( \frac{q(H)}{p(H | Y_{\Omega})} \right)dH
\]

(16)

The problem of minimizing KL divergence is equivalent to maximizing \( \mathcal{L}(q) = \int q(H) \ln [p(H | Y_{\Omega})/q(H)]dH \) since \( \ln p(Y_{\Omega}) \) is a constant. According to mean-field approximation, the approximate distribution \( q(H) \) can be expressed by

\[
q(H) = \prod_{n=1}^{N} q(Z^{(n)}) \prod_{n=1}^{N} q(u^{(n)}) q(S_{\Omega}) q(\eta) q(\tau). 
\] 

(17)

Thus, we can update the \( j \)th factor by

\[
\max \ln q_{j}(H_{j}) = \mathbb{E}_{q(H_{j})} \left[ \ln p(Y_{\Omega}, H_{j}) \right] + \text{const} \]

(18)
where \( E \{ \cdot \} \) denotes an expectation with respect to \( q \) distribution over all variables in \( \mathbf{H} \) expect \( H \). For convenience, we use \( E \{ \cdot \} \) to denote the expectation with respect to \( q(\mathbf{H}) \).

1) Posterior Distribution of Core Tensors \( \{Z^{(n)}\}_{n=1}^{N} \): The posteriors of the core tensor \( Z^{(n)} \) are factorized as independent Gaussian distributions of each lateral slice, which can be expressed by

\[
q(Z^{(n)}) = \prod_{i_n=1}^{I_n} \mathcal{N}(\text{vec}(Z_in(i_n))|\tilde{Z}_{in}^{(n)}, \Sigma_{in}^{n})
\]

(19)

where the mean \( \tilde{Z}_{in}^{(n)} \) and the variance \( \Sigma_{in}^{n} \) can be updated by

\[
\tilde{Z}_{in}^{(n)} = E[\tau]\Sigma_{in}^{n} E\left(\left(\mathbf{Z}_{in}^{(n)}\right)^{\top}\mathbf{Y}_{in} - E[\Sigma_{in}^{n}]\right)
\]

\[
\Sigma_{in}^{n} = \left(E[\tau]E\left(\left(\mathbf{Z}_{in}^{(n)}\right)^{\top}\mathbf{Z}_{in}^{(n)}\right) + E\left(\mathbf{U}^{(n)} \otimes \mathbf{U}^{(n-1)}\right)\right)^{-1}
\]

(20)

where \( \mathbf{Y}_{in} \in \mathbb{R}^{(I_n+1 \times I_n)} \) denotes partial observed data and \( \Sigma_{in}^{n} \in \mathbb{R}^{(I_n+1 \times I_n)} \) denotes the noise of partial observed data \( \mathbf{Z}_{in}^{(n)} \in \mathbb{R}^{(I_n+1 \times I_n)} \) is defined by (5). \( \mathbf{Y}_{in} \) denotes the observed entries in \( \mathbf{Y}_{in} \). As shown in (20), the variance of TR core is affected by \( \mathbf{u}^{(n)} \) and \( \mathbf{u}^{(n-1)} \).

2) Posterior Distribution of Hyperparameters \( \{u^{(n)}\}_{n=1}^{N} \): The posteriors of the hyperparameter \( \mathbf{u}^{(n)} \) are factorized as independent gamma distributions of each entry, which can be expressed by

\[
q(\mathbf{u}^{(n)}) = \prod_{r_n=1}^{R_n} \text{Ga}(u_{r_n}^{(n)}|c_{r_n}^{(n)}, \tilde{d}_{r_n}^{(n)})
\]

(21)

where parameters \( c_{r_n}^{(n)} \) and \( \tilde{d}_{r_n}^{(n)} \) can be updated by

\[
c_{r_n}^{(n)} = c_{r_n}^{(0)} + \frac{1}{2}(I_nR_{n-1} + I_{n+1}R_n)
\]

\[
\tilde{d}_{r_n}^{(n)} = \tilde{d}_{r_n}^{(0)} + \frac{1}{2}\text{Tr}\left(\mathbf{U}^{(n-1)}\mathbf{E}(\mathbf{Z}_n(r_n)\mathbf{Z}_n(r_n))^\top\right)
\]

\[
+ \frac{1}{2}\text{Tr}\left(\mathbf{U}^{(n)}\mathbf{E}((\mathbf{Z}_n(r_n))^\top\mathbf{Z}_n(r_n))\right)
\]

(22)

where \( \mathbf{Z}_n(r_n) = \mathbf{Z}^{(n)}(\cdots; r_n) \in \mathbb{R}^{R_n \times I_{n-1}} \), \( \mathbf{Z}_{n+1}(r_n) = \mathbf{Z}^{(n+1)}(r_n; \cdots) \in \mathbb{R}^{R_{n+1} \times R_n} \), \( \mathbf{U}(\eta_{r_n}) = \mathbf{c}_{r_n}^{(n)} / \tilde{d}_{r_n}^{(n)} \).

3) Posterior Distribution of Sparse Tensor \( \mathbf{S} \): The posteriors of the sparse tensor are factorized as independent Gaussian distributions of each entry, that is

\[
q(\mathbf{S}) = \prod_{(i_1, \ldots, i_N) \in \Omega} \mathcal{N}(S_{i_1 \cdots i_N}|\tilde{S}_{i_1 \cdots i_N}, \Sigma_{i_1 \cdots i_N})
\]

(23)

where the mean and the variance can be updated by

\[
\tilde{S}_{i_1 \cdots i_N} = \sigma^2_{i_1 \cdots i_N} E[\tau]\{|Y_{i_1 \cdots i_N} - E[R(\mathbf{Z}_1(i_1), \ldots, \mathbf{Z}_N(i_N))]\}
\]

\[
\sigma^2_{i_1 \cdots i_N} = (E[\eta_{i_1 \cdots i_N}] + E[\tau])^{-1}
\]

(24)

Algorithm 1 BRTR

Input: An observed tensor \( \mathbf{Y}_{\Omega} \), an indicator tensor \( \mathbf{S} \) and maximum TR rank \( (R_0, R_1, \ldots, R_N) \).

Initialization: \( \mathbf{Z}^{(n)}, \mathbf{V}^{(n)}, \mathbf{u}^{(n)}, \forall n \in [1, N] \), \( \hat{\mathbf{S}}, \sigma^2, \eta, \tau \), top level hyperparameters \( \mathbf{c}^{(0)}, \mathbf{d}^{(0)}, \forall n \in [1, N] \), \( a_0^0, b_0^0, a_0^0 \) and \( b_0^0 \).

while not convergent && t < maxiter do

for \( n = 1 \) to \( N \) do

Update \( q(Z^{(n)}) \) using Eq. (20).

end for

for \( n = 1 \) to \( N \) do

Update \( q(u^{(n)}) \) using Eq. (22).

end for

Update \( q(S) \) using Eq. (24).

Update \( q(\eta) \) using Eq. (26).

Update \( q(\tau) \) using Eq. (28).

Model reduction by eliminating zero-components in \( \{Z^{(n)}\}_{n=1}^{N} \).

Check the convergence condition.

\( t = t + 1 \).

end while

4) Posterior Distribution of Hyperparameter \( \eta \): The posteriors of hyperparameter \( \eta \) are assumed to obey a gamma distribution of each entry, which can be expressed by

\[
q(\eta) = \prod_{(i_1, \ldots, i_N) \in \Omega} \text{Ga}(\eta_{i_1 \cdots i_N}|a_0^0 b_0^0)
\]

(25)

where parameters \( a_0^0 \) and \( b_0^0 \) can be updated by

\[
a_{i_1 \cdots i_N}^0 = a_0^0 + \frac{1}{2}, \quad b_{i_1 \cdots i_N}^0 = b_0^0 + \frac{1}{2}(\hat{\mathbf{S}}_{i_1 \cdots i_N}^2 + \sigma^2_{i_1 \cdots i_N}).
\]

(26)

And \( \mathbb{E}(\eta_{i_1 \cdots i_N}) = a_{i_1 \cdots i_N}^0 b_{i_1 \cdots i_N}^0 \).

5) Posterior Distribution of Hyperparameter \( \tau \): The posterior distribution of hyperparameter \( \tau \) is assumed to obey a gamma distribution, that is

\[
q(\tau) = \text{Ga}(\tau|a^\tau, b^\tau)
\]

(27)

where parameters \( a^\tau \) and \( b^\tau \) can be updated by

\[
a^\tau = a_0^\tau + \frac{1}{2} \sum_{i_1 \cdots i_N} \mathbf{S}_{i_1 \cdots i_N}
\]

\[
b^\tau = b_0^\tau + \frac{1}{2} \mathbb{E}[\|\mathbf{Y} - \mathbb{R}(\mathbf{Z}_1(i_1), \ldots, \mathbf{Z}_N(i_N)) - \mathbf{S})\|_F^2]
\]

(28)

And \( \mathbb{E}(\tau) = a^\tau / b^\tau \). Here, we complete the derivation of the entire variational inference and the whole procedure of the BRTR method is summarized in Algorithm 1. The lower bound of model evidence in can be computed by

\[
\mathcal{L}(q) = \mathbb{E}_{q(\mathbf{S})}[\ln p(\mathbf{Y}_{\Omega}, \mathbf{H})] + H(q(\mathbf{H}))
\]

(29)

where \( \mathbb{E}_{q(\mathbf{S})}[\cdot] \) indicates the posterior expectation while \( H(\cdot) \) represents the entropy. Since \( \mathcal{L}(q) \) should not decrease in each iteration, we can use it to test convergence. A detailed derivation process can be found in the supplementary materials.
C. Complexity Analysis

Assuming that we have an $N$th-order tensor $\mathbf{Y} \in \mathbb{R}^{l_1 \times l_2 \times \cdots \times l_N}$ with TR rank $(R_0, R_1, \ldots, R_N)$ where $\{l_n\}_{n=1}^N = I$ and $\{R_n\}_{n=0}^N = R$. The computation cost of updating core factor $\mathbf{Z}^{(n)}$ is $O((N-1)\|\mathbf{O}\|R^n)$ where $\|\mathbf{O}\|$ is the total observed number of $\mathbf{O}$. The computation costs are $O(2IR^2)$ for hyperparameters $u^{(n)}$, $O((O)NR^3)$ sparse tensor $\mathbf{S}$, $O((O)NR^6)$ for $\tau$ where $|\mathbf{O}| = \sum_{i_1, \ldots, i_N} O_{i_1, \ldots, i_N}$ denotes the number of total observations. Therefore, the total complexity of the proposed algorithm is $O(N(N-1)\|\mathbf{O}\|R^n + 2NR^2 + |\mathbf{O}|NR^3 + |\mathbf{O}|NR^6)$ in each iteration. Note that the running time of the BRTR algorithm decreases significantly as the TR rank is pruned for large numbers of iterations.

D. Robust to Initial Rank Selection

As shown in (20), a larger $u^{(n)}_r$ forces all elements in both frontonal slice $\mathbf{Z}^{(n)}(\cdot, \ldots, r_n)$ and horizontal slice $\mathbf{Z}^{(n+1)}(r_n, \ldots, \cdot)$ to tend to zero. From another perspective, it can be seen from (22) that the update of $u^{(n)}_r$ depends mainly on the weighted powers of the corresponding slices $\mathbf{Z}^{(n)}(\cdot, \ldots, r_n)$ and $\mathbf{Z}^{(n+1)}(r_n, \ldots, \cdot)$. As both slices tend to zero, $u^{(n)}_r$ becomes very large and therefore acts like an indicator for deciding whether to prune slices or not. In practice, rank selection can be achieved by discarding slices where $u^{(n)}_r$ is much larger than other slices. Therefore, the proposed method is also robust to initial TR rank selection.

E. Initialization

To achieve faster convergence, a good initialization is suggested. Therefore, we employ TR approximation [66] to initial core tensors $\{\mathbf{Z}^{(n)}\}_{n=1}^N$. For top level hyperparameters $\{c^{(n)}\}_{n=1}^N$, $\{d^{(n)}\}_{n=1}^N$, $a^0_1$, $b^0_1$, $a^r_0$ and $b^r_0$, we set them to $10^{-6}$ to induce a noninformative prior. In addition, we set the expectation of hyperparameters $E(\tau) = 1$, $E(\eta_{i_1, \ldots, i_N}) = 1$ $\forall n \in N$ and $E(u^{(n)}) = 1$ to achieve stable performance. The sparse tensor $\mathbf{S}_{i_1, \ldots, i_N}$ is sampled from $\mathbf{N}(0, 1)$ while $\sigma^2_{i_1, \ldots, i_N}$ equals $E(u^{(n)}_{i_1, \ldots, i_N})$. The maximum TR rank is usually set to $10^*\text{ones}(1, N+1)$ in the following experiments for the sake of time.

IV. EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the effectiveness of the proposed BRTR method and compare it with several SOTA methods, including BRTF [56], TRLF [49], TRNNM [50], SiLRTC-TT [47], TNN [34], [67], TTNN [57], and RTRC [58]. All experimental settings are based on the default parameters suggested by the authors. All experiments are performed under different missing ratio (MR) and sparse noise ratio (SR). MR is defined by

$$\text{MR} = \frac{M}{\prod_{n=1}^N l_n}$$

(30)

where $M$ is the number of total missing entries. SR is defined by

$$\text{SR} = \frac{S}{|\mathbf{O}|}$$

(31)

where $S$ is the number of sparse noise of observations. In addition, we conduct experiments using MATLAB 2020b on a desktop computer with a 3.40-GHz Inter Core i7-6700 CPU and 16-GB RAM.

A. Experiments on Synthetic Data

1) Exact Recovery: We first perform some experiments on the synthetic data to evaluate the proposed BRTR method. We consider two experimental settings, i.e., $I = [10, 10, 10, 10]$, $R = [3, 3, 3, 3]$ and $I = [10, 10, 10, 10]$, $R = [3, 2, 3, 2]$ where $I$ is the size of dimension and $R$ is the TR rank with $R_0 = R_4$. The initial rank is set to the same dimension as the original data. The low-rank component $\mathbf{L}$ is generated by $\mathbf{L} = \mathcal{R}(\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}, \mathbf{Z}^{(3)}, \mathbf{Z}^{(4)})$ where each core tensor $\mathbf{Z}^{(n)} \in \mathbb{R}_{R^n \times \cdot \times \cdot}$, $n \in [1, 4]$ is sampled from a Gaussian distribution. The sparse component $\mathbf{S}$ is randomly corrupted by outliers, which are drawn from a uniform distribution $\mathcal{U}(-|H|, |H|)$ where $H = \max(\text{vec}(X))$. The precision of noise component $\mathbf{M}$ is controlled by the signal-to-noise ratio (SNR), that is

$$\text{SNR} = 10\log_{10}\left(\frac{\sigma_L}{\sigma_M}\right)$$

(32)

where $\sigma_L$ and $\sigma_M$ denote the variance of $\mathbf{L}$ and $\mathbf{M}$, respectively. Thus, we can obtain the synthetic data $\mathbf{Y} = \mathcal{O} \oplus (\mathbf{L} + \mathbf{S} + \mathbf{M})$. Our goal is to recover the low-rank component $\hat{\mathbf{L}}$ and sparse component $\hat{\mathbf{S}}$ and infer the groundtruth TR rank $\hat{R}$ from the observed data $\mathbf{Y}$ simultaneously. We also defined rank estimation error (REE), that is

$$\text{REE} = \frac{\sum_{n=1}^N |\hat{R}_n - R_n|}{N}$$

(33)

The first experiment is to test the recovery performance and REE at different SNR levels in a fully observed situation. Fig. 3 depicts the experiment results. From Fig. 3, we can observe that the following.

1) The recovery error $\|\hat{\mathbf{L}} - \mathbf{L}\|_F/\|\mathbf{L}\|_F$, $\|\hat{\mathbf{S}} - \mathbf{S}\|_F/\|\mathbf{S}\|_F$, and REE decreases as SNR level increases.
2) BRTR infers the groundtruth TR rank more accurately in the case of balanced rank.
3) When the TR rank is unbalanced, BRTR infers the TR rank with a small error and the recovery performance is unsatisfactory at lower SNR.

However, at higher SNR, BRTR is still able to infer the groundtruth TR rank and has a fine recovery performance. Finally, we can see that BRTR always learns the groundtruth TR rank and has a small recovery error in most cases at higher SNR.

To further verify the effectiveness of the proposed BRTR method, we compare it with the RTRC method, which is a robust low-rank TR completion model in a noise-free environment. Therefore, for the sake of fairness, we consider evaluating the performance of both BRTR and RTRC methods under different MR and SR in a noise-free environment, as shown in Table II. From Table II, we can obtain that both RTRC and BRTR methods have a better-recovery performance when MR = 0%. As MR increases, RTRC has a performance degradation, especially at higher SR. For BRTR, it can always
TABLE II
PERFORMANCE OF BOTH BRTR AND RTRC [58] METHODS UNDER DIFFERENT MR AND SR IN A NOISE-FREE ENVIRONMENT

| Method | R       | MR | SR | \( ||\mathbf{L}-\mathbf{L}||_F\) | \( ||\mathbf{S}-\mathbf{S}||_F\) | REE \( ||\mathbf{L}-\mathbf{S}||_F\) |
|--------|---------|----|----|-----------------|-----------------|-----------------|
| [10,10,10] | [3,3,3] | 0% | 10% | 1.51e-05          | 1.18e-05        | 0.000           |
|        |        | 15%|     | 3.30e-01          | 2.30e-01        | 2.04e-07        |
| [10,10,10] | [3,3,2] | 10%|     | 9.11e-07          | 3.11e-01        | 2.70e-08        |
|        |        | 15%|     | 3.23e-01          | 4.42e-01        | 2.09e-07        |
| [10,10,10] | [3,3,3] | 20%|     | 6.72e-02          | 4.51e-01        | 3.07e-08        |
|        |        | 15%|     | 3.56e-01          | 3.95e-01        | 3.45e-01        |

(continued)

accurately recover low-rank component but has a larger recovery error when reconstructing sparse component. In addition, BRTR is more difficult to infer the groundtruth TR rank as MR increases. In the case of a balanced rank, BRTR has a smaller REE compared to RTRC. In general, BRTR has a superior recovery performance compared to RTRC. It is worth noting that the TR rank of BRTR is automatically determined from the synthetic data whereas the TR rank of RTRC must be set manually.

2) Robust to Initial Rank Selection: In this section, we evaluate the robustness to initial TR rank of the proposed BRTR method. From Fig. 4(a), we can see that relative standard error (RSE) is rather stable as the initial TR rank increases, indicating that BRTR is robust to the selection of initial TR rank. Meanwhile, Fig. 4(b) shows the results of successful estimation of TR rank in 100 experiments, and it can be seen that the proposed BRTR method is able
to correctly estimate TR rank with high probability, which demonstrates the effectiveness of the BRTR method.

B. Experiments on Color Images

In this section, we perform image recovery and denoising experiments to evaluate the proposed BRTR method. We choose some popular images with the size $256 \times 256$, as shown in Fig. 5. In each image, 30% pixels are observed and 10% pixels entries from observations are set to random values in $[0, 255]$.

In addition, we evaluate the recovery performance by comparing RSE, peak SNR (PSNR) and structural similarity index (SSIM). Given a third-order tensor $\mathcal{L} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, RSE can be defined by

$$RSE = \frac{\| \hat{\mathcal{L}} - \mathcal{L} \|_F}{\| \mathcal{L} \|_F}.$$  

(34)

PSNR can be defined by

$$PSNR = 10 \log_{10} \left( \frac{\| \mathcal{L} \|_\infty}{\| \hat{\mathcal{L}} - \mathcal{L} \|_F} \right).$$  

(35)

where $\hat{\mathcal{L}}$ and $\mathcal{L}$ denote the recovered tensor and original tensor, respectively, $\| \cdot \|_\infty$ is the infinity norm. SSIM is denoted by

$$SSIM = \frac{\left(2\mu_{\hat{\mathcal{L}}}\mu_{\mathcal{L}} + c_1\right)\left(2\sigma_{\hat{\mathcal{L}}\mathcal{L}} + c_2\right)}{\left(\mu_{\mathcal{L}}^2 + \mu_{\hat{\mathcal{L}}}^2 + c_1\right)\left(\sigma_{\mathcal{L}}^2 + \sigma_{\hat{\mathcal{L}}}^2 + c_2\right)}.$$  

(36)

where $\mu_{\hat{\mathcal{L}}}$ and $\mu_{\mathcal{L}}$ are the means, $\sigma_{\hat{\mathcal{L}}}$ and $\sigma_{\mathcal{L}}$ are the variances for $\hat{\mathcal{L}}$ and $\mathcal{L}$, respectively. $\sigma_{\hat{\mathcal{L}}\mathcal{L}}$ is the covariance between $\hat{\mathcal{L}}$ and $\mathcal{L}$. $c_1$ and $c_2$ are constants. A smaller RSE and larger PSNR and SSIM imply better-recovery performance. Fig. 6 reports the RSE, PSNR, and SSIM values over eight methods of testing color images. From the experiment results, we can observe that BRTR achieves the best-recovery performance compared to other methods. RTC-based methods, including BRTF, TNN, TTNN, and BRTR, have a better-recovery performance than other methods. In addition, compared with RTRC, BRTR have a smaller RSE as well as larger PSNR and SSIM, which indicates that the Bayesian inference approach helps to recover images from missing and noisy images. Fig. 7 shows the visualization results of partial color images in the case of random missing.

The second experiment is set up to evaluate the recovery performance of various methods in removing noise on incomplete observed images in the case of nonrandom missing. We set four nonrandom missing patterns, including oblique stripe missing, block missing, scratch missing, and letter missing. Similarly, 10% pixels from observations are randomly corrupted. Fig. 8 shows recovery results on four color images under different nonrandom missing patterns. From Fig. 8, we can observe that BRTR has better-recovery performance than other methods in the case of nonrandom missing. Overall, the proposed BRTR method effectively accomplishes RTC problems and achieves better performance in incomplete image denoising tasks.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
C. Experiments on Hyperspectral Data

To further evaluate the effectiveness of the proposed BRTR method, we conduct experiments on hyperspectral image processing, which is a frequent application in the remote sensing scenario. In this section, we evaluate the ability of the proposed BRTR method to remove noise under different MR with SR = 10% on two hyperspectral datasets, including Indian Pines and Salinas A. The Indian Pines dataset has 224 spectral reflectance bands with a spatial extent of 145 × 145 pixels. For the Salinas A dataset, it contains 224 bands with a spatial extent of 83 × 86 pixels. For convenience, we select the first 30 bands of two datasets. Thus, we can reshape the data into the size of 145 × 145 × 30 and 83 × 86 × 30, respectively. Since both datasets have multiple bands, their PSNR and SSIM values are defined as the mean of the PSNR and the SSIM of hyperspectral images of all bands. Figs. 9 and 10 show visual results of RTC over eight methods on the 21st band of the Indian Pines dataset and Salinas A datasets and Table III gives detailed experiment results for both hyperspectral datasets. From Table III, we can observe that: 1) as MR increases, all eight methods suffer some degree of performance degradation, but the impact of BRTR is relatively small compared to the other methods and 2) at a higher MR, BRTR still can recover images from the missing and noisy observations while other methods fail to do it. Overall, the proposed BRTR method achieves the best performance under different MR, which indicates its stronger recovery capability.

D. Experiments on Facial Images

In this section, we evaluate the ability of the proposed BRTR method to remove noise under different SR with MR = 70% on the extended YaleB face dataset. The YaleB face dataset consists of 16128 multipose, multillumination images of 20 individuals, specifically, including nine poses and 64 illumination variations. For convenience, we choose...
TABLE III

| Datasets   | MR   | Metrics           | BRTF | TRLRF | TRNNM | SiLRTC-TT | TNN | TTNN | RTRC | BRTR |
|------------|------|-------------------|------|-------|-------|-----------|-----|------|------|------|
| Indian Pines | 50%  | RSE↑              | 0.140| 0.255| 0.269| 0.274| 0.294| 0.090| 0.089| **0.083** | 0.083 |
|            |      | PSNR↑             | 27.206| 21.172| 20.749| 20.578| 19.899| 31.323| 30.812| **0.827** | **0.827** |
|            |      | SSIM↑             | 0.693| 0.462| 0.416| 0.398| 0.340| 0.845| 0.832| **0.876** | **0.876** |
|            | 60%  | RSE↑              | 0.145| 0.237| 0.258| 0.265| 0.283| 0.094| 0.102| **0.085** | **0.085** |
|            |      | PSNR↑             | 26.853| 21.817| 21.152| 20.917| 20.192| 30.936| 29.593| **0.303** | **0.303** |
|            |      | SSIM↑             | 0.675| 0.489| 0.426| 0.400| 0.337| 0.840| 0.787| **0.873** | **0.873** |
|            | 70%  | RSE↑              | 0.153| 0.220| 0.252| 0.239| 0.278| 0.109| 0.118| **0.089** | **0.089** |
|            |      | PSNR↑             | 26.319| 22.511| 21.445| 21.166| 20.433| 30.191| 28.263| **31.440** | **31.440** |
|            |      | SSIM↑             | 0.638| 0.512| 0.431| 0.395| 0.330| 0.826| 0.730| **0.860** | **0.860** |
|            | 80%  | RSE↑              | 0.170| 0.214| 0.234| 0.261| 0.272| 0.129| 0.145| **0.099** | **0.099** |
|            |      | PSNR↑             | 25.544| 22.993| 21.438| 21.139| 20.687| 28.718| 26.443| **30.415** | **30.415** |
|            |      | SSIM↑             | 0.591| 0.509| 0.425| 0.375| 0.320| 0.791| 0.641| **0.834** | **0.834** |
|            | 90%  | RSE↑              | 0.199| 0.219| 0.284| 0.297| 0.271| 0.174| 0.209| **0.111** | **0.111** |
|            |      | PSNR↑             | 24.144| 22.795| 20.495| 20.012| 20.862| 25.665| 23.409| **29.296** | **29.296** |
|            |      | SSIM↑             | 0.498| 0.444| 0.398| 0.324| 0.309| 0.684| 0.475| **0.794** | **0.794** |

Indian Pines

Salinas A

the first and second subjects and extract their first to 12th, 30th to 34th, and 36th to 50th frames as YaleB1 and YaleB2 datasets, respectively. Each facial image has a size of 192×168 pixels. Thus, we can obtain an input tensor with the size of 192×168×32. Figs. 11 and 12 show the visual results on the first facial images of YaleB1 and YaleB2 datasets and Table IV reports the whole experiment results. From the experiment results, we can observe that as SR increases, all methods are increasingly difficult to remove noise. Both TTNN and RTRC have better-recovery performance at low-SR and poor recovery performance at high SR. BRTF has a stronger ability to remove noise, and it has better-recovery performance than TTNN and RTRC when higher SR. The proposed BRTR method has the best-recovery results in all cases. Through the experimental results of BRTF and BRTR, we can conclude that the Bayesian method can indeed remove the noise better, especially in high SR. In general, the proposed BRTR method has better results in removing noise under different SR from missing and noisy data, which indicates its stronger ability to remove noise for RTC tasks.

E. Experiments on Video Background Modeling

This experiment evaluates BRTR in the real surveillance video datasets, including WaterSurface, Boat, Pedestrian, and

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Table IV

RSE, PSNR, and SSIM over BRTF [56], TRLRF [49], TRNNM [50], SiLRTC-TT [47], TNN [34, 67], TTNN [57], RTRC [58], and BRTR methods under different SR with MR = 50% for face datasets. The highest PSNR and lowest RSE are highlighted in bold.

| Datasets | SR  | BRTF | TRLRF | TRNNM | SiLRTC-TT | TNN | TTNN | RTRC | BRTR |
|----------|-----|------|-------|-------|-----------|-----|------|------|------|
|          | 10% |      |       |       |           |     |      |      |      |
|          | RSE | 0.108| 0.197 | 0.209 | 0.212     | 0.233| 0.065| 0.071| 0.058|
|          | PSNR↑ | 26.800| 21.183| 20.681| 20.555   | 19.734| 30.169| 30.450| 31.958|
|          | SSIM↑ | 0.682| 0.432| 0.398| 0.382     | 0.299| 0.843| 0.838| 0.888|
|          | 15% |      |       |       |           |     |      |      |      |
|          | RSE | 0.133| 0.240| 0.254| 0.237     | 0.282| 0.089| 0.094| 0.059|
|          | PSNR↑ | 25.277| 19.493| 18.993| 18.897   | 18.074| 24.875| 27.567| 31.786|
|          | SSIM↑ | 0.618| 0.328| 0.295| 0.283     | 0.218| 0.722| 0.724| 0.885|
| YaleB1   | 20% |      |       |       |           |     |      |      |      |
|          | RSE | 0.155| 0.277| 0.292| 0.295     | 0.324| 0.128| 0.129| 0.060|
|          | PSNR↑ | 23.682| 18.259| 17.775| 17.705   | 16.881| 24.992| 24.976| 31.639|
|          | SSIM↑ | 0.571| 0.257| 0.229| 0.219     | 0.169| 0.541| 0.559| 0.884|
|          | 25% |      |       |       |           |     |      |      |      |
|          | RSE | 0.179| 0.311| 0.327| 0.329     | 0.362| 0.183| 0.180| 0.061|
|          | PSNR↑ | 22.454| 17.246| 16.791| 16.739   | 15.913| 21.769| 22.028| 31.587|
|          | SSIM↑ | 0.531| 0.206| 0.181| 0.174     | 0.134| 0.360| 0.388| 0.883|
|          | 30% |      |       |       |           |     |      |      |      |
|          | RSE | 0.202| 0.341| 0.358| 0.360     | 0.395| 0.246| 0.233| 0.065|
|          | PSNR↑ | 21.394| 16.427| 16.008| 15.968   | 15.159| 19.294| 19.683| 31.034|
|          | SSIM↑ | 0.487| 0.169| 0.148| 0.143     | 0.111| 0.240| 0.266| 0.874|
| YaleB2   | 10% |      |       |       |           |     |      |      |      |
|          | RSE | 0.104| 0.194| 0.204| 0.207     | 0.225| 0.067| 0.072| 0.058|
|          | PSNR↑ | 26.833| 21.098| 20.597| 20.440   | 19.703| 30.524| 29.976| 31.666|
|          | SSIM↑ | 0.724| 0.443| 0.415| 0.398     | 0.397| 0.856| 0.857| 0.914|
|          | 15% |      |       |       |           |     |      |      |      |
|          | RSE | 0.127| 0.234| 0.248| 0.231     | 0.273| 0.091| 0.095| 0.058|
|          | PSNR↑ | 25.075| 19.437| 18.902| 18.792   | 18.040| 27.764| 27.408| 31.622|
|          | SSIM↑ | 0.662| 0.337| 0.309| 0.297     | 0.235| 0.728| 0.738| 0.913|
|          | 20% |      |       |       |           |     |      |      |      |
|          | RSE | 0.148| 0.268| 0.284| 0.286     | 0.312| 0.131| 0.134| 0.059|
|          | PSNR↑ | 23.723| 18.237| 17.711| 17.629   | 16.884| 24.535| 24.363| 31.508|
|          | SSIM↑ | 0.614| 0.269| 0.243| 0.234     | 0.186| 0.543| 0.561| 0.913|
|          | 25% |      |       |       |           |     |      |      |      |
|          | RSE | 0.169| 0.300| 0.317| 0.319     | 0.348| 0.186| 0.184| 0.061|
|          | PSNR↑ | 22.328| 17.257| 16.754| 16.693   | 15.934| 21.423| 21.515| 31.282|
|          | SSIM↑ | 0.568| 0.218| 0.195| 0.195     | 0.158| 0.515| 0.519| 0.910|
|          | 30% |      |       |       |           |     |      |      |      |
|          | RSE | 0.190| 0.329| 0.346| 0.348     | 0.380| 0.243| 0.237| 0.063|
|          | PSNR↑ | 21.531| 16.445| 15.974| 15.925   | 15.175| 19.082| 19.298| 31.086|
|          | SSIM↑ | 0.532| 0.181| 0.161| 0.156     | 0.125| 0.251| 0.275| 0.907|

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Fig. 13. Visual results for the background modeling tasks over BRTR, RTRC [58], TTNN [57], and BRTF [56] methods on four video datasets with different SR and MR. We select frames 14th to 21st of the WaterSurface dataset and the last eight frames of the other three datasets for visualization. Each method has two recovered results. The left one is the background and the right one is the separated moving object.

V. CONCLUSION

We propose a BRTR method, which can effectively recover target data from missing and noisy observations. BRTR not only automatically learns the TR rank from observations without manual parameter adjustment, but also automatically balances the low-rank and sparse components, thus enabling better-recovery performance. Additionally, we have developed a variational inference algorithm to address this optimization problem, providing detailed derivations and complexity analyses. Extensive experiments on synthetic data, color images, hyperspectral data, video data and face images demonstrate the superiority of BRTR over other SOTA methods. Since the BRTR method becomes computationally expensive for higher-order tensors, we are highly interested in how to reduce its complexity and running time in the future.

REFERENCES

[1] Y. Qiu, G. Zhou, Y. Zhang, and A. Cichocki, “Canonical polyadic decomposition (CPD) of big tensors with low multilinear rank,” Multimedia Tools Appl., vol. 80, no. 15, pp. 22987–23007, 2021.
[2] G. Zhou, A. Cichocki, Q. Zhao, and S. Xie, “Efficient nonnegative Tucker decompositions: Algorithms and uniqueness,” IEEE Trans. Image Process., vol. 24, pp. 4990–5003, 2015.
[3] Z. Huang, G. Zhou, Y. Qiu, Y. Yu, and H. Dai, “A dynamic hypergraph regularized non-negative Tucker decomposition framework for multiway data analysis,” Int. J. Mach. Learn. Cybern., vol. 13, pp. 3691–3710, Sep. 2022.
[4] Y. Qiu, G. Zhou, Y. Wang, Y. Zhang, and S. Xie, “A generalized graph regularized non-negative Tucker decomposition framework for tensor data representation,” IEEE Trans. Cybern., vol. 52, no. 1, pp. 594–607, Jan. 2022.
[5] Y. Qiu, W. Sun, Y. Zhang, X. Gu, and G. Zhou, “Approximately orthogonal nonnegative Tucker decomposition for flexible multiway clustering,” Sci. China Technol. Sci., vol. 64, no. 9, pp. 1872–1880, 2021.
[6] Y. Qiu, G. Zhou, X. Chen, D. Zhang, X. Zhao, and Q. Zhao, “Semi-supervised non-negative Tucker decomposition for tensor data representation,” Sci. China Technol. Sci., vol. 64, no. 9, pp. 1881–1892, 2021.
[7] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” SIAM Rev., vol. 51, no. 3, pp. 455–500, 2009.
[8] Y. Qiu, G. Zhou, Z. Huang, Q. Zhao, and S. Xie, “Efficient tensor robust PCA under hybrid model of Tucker and tensor train,” IEEE Signal Process. Lett., vol. 29, pp. 627–631, Jan. 2022.
[9] H. He and Y. Tan, “Pattern clustering of hysteresis time series with multivalued mapping using tensor decomposition,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 48, no. 6, pp. 993–1004, Jun. 2018.
[10] T. Sun and X.-M. Sun, “New results on classification modeling of noisy tensor datasets: A fuzzy support tensor machine dual model,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 52, no. 8, pp. 5188–5200, Aug. 2022.
[11] Z. Huang, Y. Qiu, and W. Sun, “Recognition of motor imagery EEG patterns based on common feature analysis,” Brain Comput. Interfaces, vol. 8, no. 4, pp. 128–136, 2021.
[12] X. Chang, P. Ren, P. Xu, Z. Li, X. Chen, and A. Hauptmann, “A comprehensive survey of scene graphs: Generation and application,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 1, pp. 1–26, Jan. 2023.
[13] L. Zhang et al., “TN-ZSTAD: Transferable network for zero-shot temporal activity detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 3, pp. 3848–3861, Mar. 2023.
[14] C. Yan et al., “ZeroNAS: Differentiable generative adversarial networks search for zero-shot learning,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 12, pp. 9733–9740, Dec. 2022.
[15] Y. Liu, Z. Long, H. Huang, and C. Zhu, “Low CP rank and Tucker rank tensor completion for estimating missing components in image data,” IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 4, pp. 944–954, Apr. 2020.
A. Wang, Z. Jin, and G. Tang, “Robust tensor decomposition via element-weighted low-rank tensor train with overlapping ket augmentation,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 11, pp. 7286–7300, Nov. 2022.

M. Li, P.-Y. Huang, X. Chang, J. Hu, Y. Yang, and A. Hauptmann, “Video pivoting unsupervised multi-modal machine translation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 3, pp. 3918–3932, Mar. 2023.

P. Shao, D. Zhang, G. Yang, J. Tao, F. Che, and T. Liu, “Tucker decomposition-based temporal knowledge graph completion,” Knowl. Based Syst., vol. 238, Feb. 2022, Art. no. 107841.

H. Kuang, L. Chen, L. L. H. Chan, R. C. Cheung, and H. Yan, “Feature selection based on tensor decomposition and object proposal for night-time surveillance to lowest detection,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 49, no. 1, pp. 71–80, Jan. 2019.

D. Wu, Y. He, X. Luo, and M. Zhou, “A latent factor analysis-based approach to online sparse streaming feature selection,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 52, no. 11, pp. 6744–6758, Nov. 2022.

M. Nickel and V. Tresp, “Tensor factorization for multi-relational learning,” in Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Data Eng. (ICDE), 2018, pp. 1120–1131.

B. Ernus, E. Acar, and A. T. Cemgil, “Link prediction in heterogeneous data via generalized coupled tensor factorization,” Data Min. Knowl. Discovery, vol. 29, no. 1, pp. 203–236, 2015.

E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, “Scalable tensor factorizations for incomplete data,” Chemometr. Intell. Lab. Syst., vol. 106, no. 1, pp. 41–56, 2011.

L. Sorber, M. Van Barel, and L. De Lathauwer, “Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-\((L_1,L_2)\)-terms, and a new generalization,” SIAM J. Optim., vol. 23, no. 2, pp. 695–720, 2013.

V. De Silva and L.-H. Lim, “Tensor rank and the ill-posedness of the best low-rank approximation problem,” SIAM J. Matrix Anal. Appl., vol. 30, no. 3, pp. 1084–1127, 2008.

M. Filipović and A. Jukić, “Tucker factorization with missing data and application to low-rank tensor completion,” Multidimension. Syst. Signal Process., vol. 26, no. 3, pp. 677–692, 2015.

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.

S. Oh, N. Park, S. Lee, and U. Kang, “Scalable tucker factorization for sparse tensors-algorithms and discoveries,” in Proc. IEEE 34th Int. Conf. Databases (ICDE), 2018, pp. 4326–4337.

D. Lee, J. Lee, and H. Yu, “Fast tucker factorization for large-scale tensor completion,” in Proc. IEEE Int. Conf. Data Min. (ICDM), 2018, pp. 1098–1103.

Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, “Novel methods for multilinear data completion and de-noising based on tensor-SVD,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit., 2014, pp. 3842–3849.

C. Lu, X. Peng, and Y. Wei, “Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 5996–6004.

T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, and L.-J. Deng, “Multi-dimensional imaging data recovery via minimizing the partial sum of tubal nuclear norm,” J. Comput. Appl. Math., vol. 372, Jul. 2020, Art. no. 112803.

A. Wang, Z. Jin, and G. Tang, “Robust tensor decomposition via t-SVD: Near-optimal statistical guarantee and scalable algorithms,” Signal Process., vol. 167, Feb. 2020, Art. no. 107319.

A. Wang, C. Li, Z. Jin, and Q. Zhao, “Robust tensor decomposition via orientation invariant tubal nuclear norms,” in Proc. AAAI Conf. Artif. Intell., 2020, pp. 6102–6109.

I. V. Oseledets, “Tensor-train decomposition,” SIAM J. Sci. Comput., vol. 33, no. 5, pp. 2295–2317, 2011.

Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki, “Tensor ring decomposition,” 2016, arXiv:1606.05535.

Z. Huang, Y. Qiu, J. Yu, and G. Zhou, “Multi-aspect streaming tensor ring completion for dynamic incremental data,” IEEE Signal Process. Lett., vol. 29, pp. 2657–2661, Dec. 2022.

L. Xu, L. Cheng, N. Wong, and Y.-C. Wu, “Learning tensor train representation with automatic rank determination from incomplete noisy data,” 2020, arXiv:2010.06564.

H. Huang, Y. Liu, J. Liu, and C. Zhu, “Provable tensor ring completion,” Signal Process., vol. 171, Jun. 2020, Art. no. 107486.
[64] A. Cichocki, “Era of big data processing: A new approach via tensor networks and tensor decompositions,” 2014, arXiv:1403.2048.
[65] M. E. Tipping, “Sparse Bayesian learning and the relevance vector machine,” J. Mach. Learn. Res., vol. 1, pp. 211–244, Sep. 2001.
[66] W. Wang, V. Aggarwal, and S. Aeron, “Efficient low rank tensor ring completion,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 5697–5705.
[67] A. Wang, X. Song, X. Wu, Z. Lai, and Z. Jin, “Robust low-tubal-rank tensor completion,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), 2019, pp. 3432–3436.

Zhenhao Huang received the B.Eng. degree in Internet of Things (IoT) engineering from the Guangdong University of Technology, Guangzhou, China, in 2020, where he is currently pursuing the Ph.D. degree in pattern recognition and intelligent systems. His current research interests include tensor analysis and Bayesian learning.

Yuning Qiu received the B.Eng. degree in mechanical design, manufacturing and automation from South China Agricultural University, Guangzhou, China, in 2016, and the Ph.D. degree in pattern recognition and intelligent systems from the Guangdong University of Technology, Guangzhou, in 2021. He was a Postdoctoral Researcher from the Guangdong University of Technology from 2021 to 2023. He was a Student Intern with the Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Wako, Saitama, Japan, from October 2016 to December 2016. He is currently a Special Postdoctoral Researcher with the School of Automation, Guangdong University of Technology, Guangzhou. His main research interests are tensor-based machine learning algorithms, theory, and applications. Dr. Qiu served as a reviewer for NeurIPS, ICML, ICLR, CVPR, and AISTATS.

Xing Chen received the Ph.D. degree in pattern recognition and intelligent systems from the Guangdong University of Technology, Guangzhou, China, in 2023. He was a Student Intern with the RIKEN Center for Advanced Intelligence Project, Tokyo, Japan, in 2017. He is currently a Postdoctoral Researcher with the School of Automation, Guangdong University of Technology. His research interests include tensor learning, machine learning, and signal processing.

Weijun Sun received the master’s degree in computer application from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2003, and the Ph.D. degree in control science and engineering from the Guangdong University of Technology (GDUT), Guangzhou, China, in 2020. He is a member with the Institute of Intelligent Information Processing, GDUT. His current research interests include pattern recognition and machine learning.

Guoxu Zhou received the Ph.D. degree in intelligent signal and information processing from the South China University of Technology, Guangzhou, China, in 2010. He was a Research Scientist with the Brain Science Institute, RIKEN, Tokyo, Japan. He is currently a Full Professor and the Dean with the School of Automation, Guangdong University of Technology, Guangzhou. He has authored more than 70 peer-reviewed articles that have been published in prestigious journals, such as Proceedings of the IEEE, IEEE Signal Processing Magazine, IEEE Transactions on Signal Processing, IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Cybernetics, and IEEE Transactions on Image Processing. His current research interests include tensor analysis, intelligent information processing, and artificial intelligence. Prof. Zhou serves as an Associate Editor for IEEE Transactions on Neural Networks and Learning Systems and IEEE Transactions on Systems, Man, and Cybernetics: Systems.