Efficient tensor completion: Low-rank tensor train

Ho N. Phien[1], Hoang D. Tuan[1], Johann A. Bengua[1] and Minh N. Do[2]

Abstract—This paper proposes a novel formulation of the tensor completion problem to impute missing entries of data represented by tensors. The formulation is introduced in terms of tensor train (TT) rank which can effectively capture global information of tensors thanks to its construction by a well-balanced matricization scheme. Two algorithms are proposed to solve the corresponding tensor completion problem. The first one called simple low-rank tensor completion via tensor train (SiLRTC-TT) is intimately related to minimizing the TT nuclear norm. The second one is based on a multilinear matrix factorization model to approximate the TT rank of the tensor and called tensor completion by parallel matrix factorization via tensor train (TMac-TT). These algorithms are applied to complete both synthetic and real world data tensors. Simulation results of synthetic data show that the proposed algorithms are efficient in estimating missing entries for tensors with either low Tucker rank or TT rank while Tucker-based algorithms are only comparable in the case of low Tucker rank tensors. When applied to recover color images represented by ninth-order tensors augmented from third-order ones, the proposed algorithms outperforms the Tucker-based algorithms.

Index Terms—Tensor completion, tensor train decomposition, tensor train rank, tensor train nuclear norm, Tucker decomposition.

I. Introduction

Tensors are multi-dimensional arrays, known as higher-order generalizations of matrices and vectors [1]. Tensors provide a natural way to represent multi-dimensional data objects whose entries are indexed by several continuous or discrete variables. Employing tensors and their decompositions to process data objects has become increasingly popular since [2]–[4]. For instance, a color image is a third-order tensor defined by two indices for spatial variables and one index for color mode. A video comprised of color images is a fourth-order tensor with an additional index for a temporal variable. Residing in extremely high-dimensional data spaces, the tensors in practical applications are nevertheless often of *low-rank* [1]. Consequently, they can be effectively projected to much smaller subspaces underlying their decompositions such as the CANDECOMP/PARAFAC (CP) [5], [6], Tucker [7] and tensor train (TT) [8] or matrix product state (MPS) [9]–[11].

Motivated by the success of low rank matrix completion (LRMC) [12]–[14], much recent effort has been made to extend its concept to low rank tensor completion (LRTC). In fact, LRTC has been seen pivotal in computer vision

[1]Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia; Email:ngocphien.ho@uts.edu.au, tuan.hoang@uts.edu.au, johann.a.bengua@student.uts.edu.au.

[2]Department of Electrical and Computer Engineering and the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA; Email: minhdo@illinois.edu

and graphics, signal processing and machine learning [15]–[21]. The common target is to recover missing entries of a tensor from its partially observed entities [22]–[24]. Despite its potential application, LRTC remains a grand challenge due to the fact that minimization of the tensor rank, defined as CP rank [1], is an NP-hard problem. There have been some progress in formulating the LRTC via *Tucker rank* [15], [18]. However, a crucial drawback of Tucker rank is that its components are ranks of matrices constructed based on an unbalanced matricization scheme (one mode versus the rest). Therefore, the upper bound of each individual rank is often small, and may not be suitable for describing global information of the tensor, especially strongly correlated tensors of high orders. In addition, the matrix rank minimization are only efficient when the matrix is more balanced. As the rank of a matrix is not more than $\min\{n, m\}$, where m and n are the number of rows and columns of the matrix, respectively, the high ratio $\max\{m, n\}/\min\{m, n\}$ would effectively rule out the need of matrix rank minimization.

In this paper we introduce a novel formulation of LRTC in the concept of *TT rank* [8] which is different from the Tucker rank. Specifically, the TT rank is constituted by ranks of matrices formed by a well-balanced matricization scheme, i.e. matricize the tensor along one or a few modes. This gives rise to a huge advantage for representing the tensor rank in the sense that its components can have large upper bounds. Consequently, it provides a much better means to capture the global information in the tensor. We will also analyze how the rank of a matrix is closely related to the concept of von Neumann entropy in quantum information theory [25]. Subsequently, it is shown that the weakness of LRTC formulated by Tucker rank can be mitigated by utilizing the TT rank.

Two algorithms are proposed to approximately solve the proposed LRTC formulation, namely SiLRTC-TT and TMac-TT. The former is based on the SiLRTC [15] which employs the block coordinate descent (BCD) for optimization and tools such as singular value thresholding from the matrix rank minimization problem [12], [13]. The SiLRTC-TT mainly solves the *TT nuclear norm* minimization problem that is a convex surrogate for the new LRTC. Here, we define TT nuclear norm of a tensor as a sum of weighted nuclear norms of matrices formed by matricizing the tensor along one or a few modes. The latter adapted from its counterpart, i.e. TMac [21], is related to solving a weighted multilinear matrix factorization model. Although this model is non-convex, it can be solved effectively in the sense that no computationally expensive SVD is needed compared to the former.

The algorithms are applied to complete both synthetic and real world data described by tensors which are assumed to

have low Tucker rank or TT rank. Empirical results simulated by proposed algorithms for synthetic data are much more efficient than the Tucker-based algorithms in the case of low TT rank and are comparable in the case of low Tucker rank tensors. When studying real world data such as color images, we introduce a tensor augmentation scheme called *ket augmentation* (KA) to represent a lower-order tensor by a higher-order one without changing its number of entries. The KA scheme extended from the one introduced in [26] provides a perfect means to obtain higher-order tensor representation of visual data. We apply the proposed algorithms to complete a few color images represented by ninth-order tensors and results show that our algorithms outperforms the Tucker-based ones. Besides, we will show that our proposed algorithms recover augmented tensors more efficiently than those without applying augmentation scheme.

The rest of the paper is organized as follows. Section II provides some notations and preliminaries of tensors. In Section III, we firstly review the conventional formulation of LRTC and then introduce our new formulation in terms of TT rank. The algorithms are then proposed to solve the problem. We introduce the tensor augmentation scheme KA in Section IV. Section V presents simulation results. Finally, we conclude our work in Section VI.

II. NOTATIONS AND PRELIMINARIES OF TENSORS

We adopt some mathematical notations and preliminaries of tensors in [1]. A *tensor* is a multi-dimensional array and its *order* (also known as way or mode) is the number of its dimensions. Scalars are zero-order tensors denoted by lowercase letters $(x, y, z, \ldots)$. Vectors and matrices are the first- and second-order tensors which are denoted by boldface lowercase letters $(\mathbf{x}, \mathbf{y}, \mathbf{z}, \ldots)$ and capital letters $(X, Y, Z, \ldots)$, respectively. A higher-order tensor (the tensor of order three or above) are denoted by calligraphic letters $(\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \ldots)$.

An Nth-order tensor is denoted as $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ where $I_k, \ k = 1, \ldots, N$ is the dimension corresponding to mode k. The elements of $\mathcal{X}$ are denoted as $x_{i_1 \cdots i_k \cdots i_N}$, where $1 \leq i_k \leq I_k, \ k = 1, \ldots, N$.

A mode-n fiber of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is a vector defined by fixing all indices but i_n and denoted by $\mathbf{x}_{i_1 \cdots i_{n-1} : i_{n+1} \cdots i_N}$.

Mode-n matricization (also known as mode-n unfolding or flattening) of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is the process of unfolding or reshaping the tensor into a matrix $X_{(n)} \in \mathbb{R}^{I_n \times (I_1 \cdots I_{k-1} I_{k+1} \cdots I_N)}$ by rearranging the mode-n fibers to be the columns of the resulting matrix. Tensor element $(i_1, \ldots, i_{n-1}, i_n, i_{n+1}, \ldots, i_N)$ maps to matrix element (i_n, j) such that

$$j = 1 + \sum_{k=1, k \neq n}^{N} (i_k - 1) J_k \quad \text{with} \quad J_k = \prod_{m=1, m \neq n}^{k-1} I_m. \quad (1)$$

The mode-n product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a matrix $A \in \mathbb{R}^{J \times I_n}$ results into a new tensor of size $I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$ which is denoted as $\mathcal{X} \times_n A$.

Elementwise, it is described by

$$(\mathcal{X} \times_n A)_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 \cdots i_n \cdots i_N} a_{j i_n}. \quad (2)$$

The Tucker decomposition (TD) is a form of higher-order principle component analysis [1], [7]. It is employed to decompose a tensor into a core tensor multiplied by a matrix along each mode. In general, for a given tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, its TD is written as,

$$\mathcal{X} = \mathcal{G} \times_1 A^{(1)} \times_2 A^{(2)} \cdots \times_N A^{(N)}, \quad (3)$$

where the core tensor $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \cdots \times r_N}$ and the factor matrices $A^{(k)} \in \mathbb{R}^{r_k \times I_k}, k = 1, \ldots, N$. The vector $\mathbf{r} = (r_1, r_2, \ldots, r_N)$, where r_n is the rank of the corresponding matrix $X_{(n)}$ denoted as $r_n = \text{rank}(X_{(n)})$, is called as the *Tucker rank* of the tensor $\mathcal{X}$.

The inner product of two tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is defined as

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} x_{i_1 i_2 \cdots i_N} y_{i_1 i_2 \cdots i_N}. \quad (4)$$

Accordingly, the Frobenius norm of $\mathcal{X}$ is $\|\mathcal{X}\|_F = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}$.

III. THE FORMULATION OF TENSOR COMPLETION AND ALGORITHMS

This section firstly revisits the conventional formulation of LRTC based on the Tucker rank, a generalization of LRMC. Then a new LRTC formulated in terms of the TT rank is introduced with algorithms.

A. Conventional formulation of tensor completion

We give an overview of matrix completion before introducing the formulation for tensor completion. The problem on how to recover missing entries of a low-rank matrix $T \in \mathbb{R}^{m \times n}$ from its partially known entries given by a subset Ω can be studied via the well-known optimization problem [27]:

$$\min_{X} \quad \text{rank}(X) \quad \text{s.t.} \quad X_\Omega = T_\Omega. \quad (5)$$

The missing entries of X are estimated such that the rank of X is as small as possible. Due to the combinational nature of the function $\text{rank}(\cdot)$, problem (5), however, is NP-hard and one needs to look for its surrogates. Minimization of the matrix nuclear norm has proved as an excellent approximation of the matrix rank. This leads to the following convex optimization problem for matrix completion [12], [13], [28]:

$$\min_{X} \quad \|X\|_* \quad \text{s.t.} \quad X_\Omega = T_\Omega, \quad (6)$$

where the nuclear norm $\|X\|_*$ is the summation of the singular values of X. Note that this matrix nuclear norm optimization problem is efficient only when X is balanced which implies that $m \approx n$.

Alternatively, one can also apply the low-rank matrix factorization model to solve the matrix completion problem [29]:

$$\min_{U, V, X} \quad \frac{1}{2} \|UV - X\|_F^2 \quad \text{s.t.} \quad X_\Omega = T_\Omega, \quad (7)$$

where $U \in \mathbb{R}^{m \times r}$, $V \in \mathbb{R}^{r \times n}$ and $X \in \mathbb{R}^{m \times n}$ and the integer r is the estimated rank of the matrix T.

The matrix completion can be generalized for tensor underlying the concept of tensor rank, e.g. the CANDE-COMP/PARAFAC rank (CP-rank), Tucker rank [1], [7]. For instance, in terms of Tucker rank, completing an Nth-order tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ from its known entries given by an index set Ω is related to solving the following optimization problem [15], [18], [20], [21]:

$$\min_{X_{(k)}} \quad \sum_{k=1}^{N} \alpha_k \text{rank}(X_{(k)}) \quad \text{s.t.} \quad \mathcal{X}_\Omega = \mathcal{T}_\Omega. \quad (8)$$

where $\{\alpha_k\}_{k=1}^{N}$ are defined as weights fulfilling condition $\sum_{k=1}^{N} \alpha_k = 1$. The Eq. (8) is a weighted multilinear matrix completion problem which is still NP-hard. Therefore, one needs to switch to an alternative by generalizing the matrix case. For instance, (8) can be converted to the following optimization problem [15]:

$$\min_{X_{(k)}} \quad \sum_{k=1}^{N} \alpha_k ||X_{(k)}||_* \quad \text{s.t.} \quad \mathcal{X}_\Omega = \mathcal{T}_\Omega, \quad (9)$$

where $\sum_{k=1}^{N} \alpha_k ||X_{(k)}||_*$ can be defined as *Tucker nuclear norm* of the tensor. This problem can be then solved by simply applying known methods such as the block coordinate descent (BCD) to alternatively optimize a group of variables while the other groups remain fixed. Besides, one can generalize the low-rank matrix factorization model in (7) as a subsitute for (8) and then apply the BCD method with the nonlinear Gauss-Seidal method to solve it [20], [21], [29].

Although the Tucker-based LRTC problem has become increasingly popular, it is only appropriate for the tensors with either low Tucker rank or low orders and might be less efficient when applying to real world data represented by tensors of orders higher than three. This weakness comes from the fact that each matrix $X_{(k)}$ in (8) is obtained by matricizing the tensor along one single mode. As a consequence, it is unbalanced and the corresponding rank is not large enough to capture the *global correlation* between elements in the tensor. Even when all the modes have the same dimension ($I_1 = \cdots = I_N \equiv I$), these matrices are highly unbalanced. We can clarify this observation via the concept of von Neumann entropy [25] as follows.

Represent $\mathcal{X}$ as a *pure state* in the space $\mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$,

$$\mathcal{X} = \sum_{i_1, i_2 \ldots, i_N} x_{i_1 i_2 \cdots i_N} \mathbf{e}_{i_1} \otimes \mathbf{e}_{i_2} \cdots \otimes \mathbf{e}_{i_N}, \quad (10)$$

where "$\otimes$" denotes a tensor product [1], $\mathbf{e}_{i_k} \in \mathbb{R}^{I_k}$ forms an orthonormal basis in $\mathbb{R}^{I_k}$ for each $k = 1, \ldots, N$. Applying mode-k matricization of $\mathcal{X}$ results into $X_{(k)}$ representing a *pure state* of the composite system AB in the space $\mathcal{H}_{AB} \in \mathbb{R}^{m \times n}$, which is a tensor product of two subspaces $\mathcal{H}_A \in \mathbb{R}^m$ and $\mathcal{H}_B \in \mathbb{R}^n$ of dimensions $m = I_k$ and $n = \prod_{l=1, l \neq k}^{N} I_l$, respectively. The subsystems A and B are seen as two *contigous partitions* consisting of mode k and all

other modes of the tensor, respectively. It follows from (10) that

$$X_{(k)} = \sum_{i_k, j} x_{i_k j} \mathbf{e}_{i_k} \otimes \mathbf{e}_j, \quad (11)$$

where the new index j is defined as in (1), $\mathbf{e}_j = \otimes_{l=1, l \neq k}^{N} \mathbf{e}_{i_l} \in \mathbb{R}^n$. According to the Schmidt decomposition [25], there exist orthonormal bases $\{\mathbf{u}_l^A\}$ in $\mathcal{H}_A$ and $\{\mathbf{v}_l^B\}$ in $\mathcal{H}_B$ such that,

$$X_{(k)} = \sum_{l=1}^{r_k} \lambda_l \mathbf{u}_l^A \otimes \mathbf{v}_l^B, \quad (12)$$

where r_k is the rank of $X_{(k)}$, λ_l are nonvanishing singular values and $\{\mathbf{u}_l^A\}$ and $\{\mathbf{v}_l^B\}$ are columns of orthonormal matrices U and V which are obtained from the SVD $X_{(k)} = U \lambda V^T$, respectively. The correlation between two subsystems A and B can be studied via von Neumann entropy defined as [25]:

$$S^A = -\text{Trace}(\rho^A \log_2(\rho^A)), \quad (13)$$

where ρ^A is called the *reduced density matrix operator* of the composite system and computed by taking the partial trace of the density matrix ρ^{AB} with respect to B. Specifically, we have

$$\begin{aligned} \rho^{AB} &= X_{(k)} \otimes (X_{(k)})^T \\ &= \Big(\sum_{l=1}^{r_k} \lambda_l \mathbf{u}_l^A \otimes \mathbf{v}_l^B \Big) \otimes \Big(\sum_{j=1}^{r_k} \lambda_j \mathbf{u}_j^A \otimes \mathbf{v}_j^B \Big)^T. (14) \end{aligned}$$

Then ρ^A is computed as

$$\begin{aligned} \rho^A &= \text{Trace}_B(\rho^{AB}) \\ &= \sum_{l=1}^{r} \lambda_l^2 \mathbf{u}_l^A \otimes (\mathbf{u}_l^A)^T, \quad (15) \end{aligned}$$

Substituting (15) to (13) yields

$$S^A = -\sum_{l=1}^{r_k} \lambda_l^2 \log_2 \lambda_l^2. \quad (16)$$

Similarly,

$$\begin{aligned} S^B &= -\text{Trace}(\rho^B \log_2(\rho^B)) \\ &= -\sum_{l=1}^{r_k} \lambda_l^2 \log_2 \lambda_l^2, \quad (17) \end{aligned}$$

which is the same with S^A, simply $S^A = S^B = S$. This entropy reflects the correlation or *degree of entanglement* between subsystem A and its complement B [30]. Without loss of generality, the normalization condition $\sum_{l=1}^{r_k} \lambda_l^2 = 1$ can be imposed, so $0 \leq S \leq \log_2 r_k$. Obviously, there is no correlation between subsystems A and B whenever $S = 0$ (where $\lambda_1 = 1$ and the other singular values are zeros). There exists correlation between subsystems A and B whenever $S \neq 0$ with its maxima $S = \log_2 r_k$ (when $\lambda_1 = \cdots = \lambda_{r_k} = 1/\sqrt{r_k}$). If the singular values decay significantly, e.g. exponential decay, we can also keep a few r_k ($r_k \ll m$) largest singular values of λ without considerably losing accuracy in quantifying the amount of correlation between the subsystems. Then r_k is referred to as the approximate low rank of the matrix $X_{(k)}$ which means

that the amount of correlation between the elements in the matrix is small. On the contrary, if two subsystems A and B are highly correlated, i.e. the singular values decay very slowly, then r_k needs to be as large as possible to capture the correlation in the tensor. Therefore, the problem of matrix rank minimization is in fact intimately related to the problem of von Neumann entropy minimization.

From the above analysis, we see that the amount of correlation between elements in the matrix $X_{(k)}$ depends on the rank r_k which is bounded by $m = I_k$. Therefore, when the dimensions of modes are slightly different or the same, that is $I_1 \approx I_2 \approx \cdots \approx I_N \approx I$, the matrix $X_{(k)}$ is essentially unbalanced due to $m \ll n$ when either I or N is large. As a result, the limit of each r_k is too small to describe the correlation of the tensor in case the tensor $\mathcal{X}$ has higher order ($N > 3$) that makes the Tucker-based LRTC no longer appropriate for a highly-correlated tensor. In the next section we introduce a new LRTC problem formulated in terms of *TT rank* defined by more balanced matrices.

B. Tensor completion formulation in the concept of tensor train rank and algorithms

The tensor train (TT) decomposition is applied to decompose a higher-order tensor into a sequence of connected lower-order tensors [8]. Using Vidal's decomposition [31], the TT decomposition of a tensor described by (10) can be written in the following form,

$$\mathcal{X} = \sum_{i_1,\ldots,i_N} \Gamma_{i_1}^{[1]} \lambda^{[1]} \cdots \lambda^{[N-1]} \Gamma_{i_N}^{[N]} \mathbf{e}_{i_1} \otimes \cdots \otimes \mathbf{e}_{i_N} \quad (18)$$

where for $k = 1, \ldots, N$, $\Gamma_{i_k}^{[k]}$ is an $r_{k-1} \times r_k$ matrix and $\lambda^{[k]}$ is the $r_k \times r_k$ diagonal singular matrix, $r_0 = r_{N+1} = 1$. For every k, the following orthogonal conditions are fulfilled:

$$\sum_{i_k=1}^{I_k} \Gamma_{i_k}^{[k]} \lambda^{[k]} (\Gamma_{i_k}^{[k]} \lambda^{[k]})^T = \mathbb{I}^{[k-1]}, \quad (19)$$

$$\sum_{i_k=1}^{I_k} (\lambda^{[k-1]} \Gamma_{i_k}^{[k]})^T \lambda^{[k-1]} \Gamma_{i_k}^{[k]} = \mathbb{I}^{[k]}, \quad (20)$$

where $\mathbb{I}^{[k-1]}$ and $\mathbb{I}^{[k]}$ are the identity matrices of sizes $r_{k-1} \times r_{k-1}$ and $r_k \times r_k$, respectively. Based on the form (18), each component r_k of the so-called *TT rank* of the tensor, simply defined as $\mathbf{r} = (r_1, r_2, \ldots, r_{N-1})$, can be determined directly via the singular matrices $\lambda^{[k]}$. Specifically, to determine r_k, rewrite (18) as

$$\mathcal{X} = \sum_{i_1,i_2\ldots,i_N} \mathbf{u}^{[1\cdots k]i_1\cdots i_k} \lambda^{[k]} \mathbf{v}^{[k+1\cdots N]i_{k+1}\cdots i_N}, \quad (21)$$

where

$$\mathbf{u}^{[1\cdots k]i_1\cdots i_k} = \Gamma_{i_1}^{[1]} \lambda^{[1]} \cdots \Gamma_{i_k}^{[k]} \otimes_{l=1}^{k} \mathbf{e}_{i_l}, \quad (22)$$

and

$$\mathbf{v}^{[k+1\cdots N]i_{k+1}\cdots i_N} = \Gamma_{i_{k+1}}^{[k+1]} \lambda^{[k+1]} \cdots \Gamma_{i_N}^{[N]} \otimes_{l=k+1}^{N} \mathbf{e}_{i_l}. \quad (23)$$

We can also rewrite (21) in terms of the matrix form of an SVD as

$$X_{[k]} = U \lambda^{[k]} V^T, \quad (24)$$

where $X_{[k]} \in \mathbb{R}^{m \times n}$ ($m = \prod_{l=1}^{k} I_l, n = \prod_{l=k+1}^{N} I_l$) is the *mode-*$(1, 2, \ldots, k)$ *matricization* of the tensor $\mathcal{X}$ [8], $U \in \mathbb{R}^{m \times r_k}$ and $V \in \mathbb{R}^{n \times r_k}$ are orthogonal matrices. Obviously, r_k, defined as number of nonvanishing singular values of $\lambda^{[k]}$, is the rank of $X_{[k]}$.

In practice, the mode-$(1, 2, \ldots, k)$ matricization $X_{[k]}$ of tensor $\mathcal{X}$ can be obtained by reshaping the tensor $\mathcal{X}$ in such a way that the first k indices enumerate the rows of $X_{[k]}$, and the last $(N - k)$ enumerate the columns of $X_{[k]}$ [8], [32]. Specifically, the tensor element $(i_1, i_2, \ldots, i_N)$ of $\mathcal{X}$ maps to the element (i, j) of $X_{[k]}$ for

$$i = 1 + \sum_{m=1}^{k} \left((i_m - 1) \prod_{l=1}^{m-1} I_l\right), \quad (25)$$

$$j = 1 + \sum_{m=k+1}^{N} \left((i_m - 1) \prod_{l=k+1}^{m-1} I_l\right). \quad (26)$$

Since matrix $X_{[k]}$ is obtained by matricizing along a few k modes rather than one single mode, its rank r_k is bounded by $\min(\prod_{l=1}^{k} I_l, \prod_{l=k+1}^{N} I_l)$. Therefore TT rank is in general more appropriate than Tucker rank for quantifying correlation of higher-order tensors.

We now propose to formulate the LRTC problem in terms of TT rank as

$$\min_{X_{[k]}} \sum_{k=1}^{N-1} \alpha_k \text{rank}(X_{[k]}) \quad \text{s.t.} \quad \mathcal{X}_\Omega = \mathcal{T}_\Omega, \quad (27)$$

where α_k denotes the weight that the rank of the matrix $X_{[k]}$ contributes to the TT rank that the condition $\sum_{k=1}^{N-1} \alpha_k = 1$ is satisfied. The LRTC problem is now relaxed to the weighted multilinear matrix completion problem which is similar to (8). It is still difficult to directly tackle as rank($\cdot$) is presumably hard. Thus, we will convert this problem into two separate problems. The first one based on the so-called *TT nuclear norm*, defined as

$$||\mathcal{X}||_* = \sum_{k=1}^{N-1} \alpha_k ||X_{[k]}||_*, \quad (28)$$

is given by

$$\min_{\mathcal{X}} \sum_{k=1}^{N-1} \alpha_k ||X_{[k]}||_* \quad \text{s.t.} \quad \mathcal{X}_\Omega = \mathcal{T}_\Omega, \quad (29)$$

The problem (29) is defined similarly to (9) where the Tucker nuclear norm is used instead. Besides, from (29) we can recover the square model [32] by choosing the weights such that $\alpha_k = 1$ if $k = \text{round}(N/2)$ otherwise $\alpha_k = 0$.

The problem (29) can be further converted to the following problem:

$$\min_{\mathcal{X}, M_k} \sum_{k=1}^{N-1} \alpha_k ||M_k||_* + \frac{\beta_k}{2} ||X_{[k]} - M_k||_F^2$$

$$\text{s.t.} \quad \mathcal{X}_\Omega = \mathcal{T}_\Omega, \quad (30)$$

where β_k are positive numbers and can be solved by employing the BCD method for the optimization which will be discussed later in this section.

The second proposed problem is based on the multilinear matrix factorization model. More specifically, given a matrix $X_{[k]} \in \mathbb{R}^{m \times n}$ of rank r_k, it can be factorized as $X_{[k]} = UV$ where $U \in \mathbb{R}^{m \times r_k}$ and $V \in \mathbb{R}^{r_k \times n}$. Therefore, instead of optimizing the nuclear norm of the unfolding matrices $X_{[k]}$, the Frobenius norm is minimized as follows,

$$\min_{U_k, V_k, \mathcal{X}} \quad \sum_{k=1}^{N-1} \frac{\alpha_k}{2} ||U_k V_k - X_{[k]}||_F^2 \tag{31}$$
$$\text{s.t.} \qquad \mathcal{X}_\Omega = \mathcal{T}_\Omega,$$

where $U_k \in \mathbb{R}^{\prod_{j=1}^{k} I_j \times r_k}$ and $V_k \in \mathbb{R}^{r_k \times \prod_{j=k+1}^{N} I_j}$. This model is similar to the one proposed in [20], [21] (which is an extension of the matrix completion model [29]) where the Tucker rank is employed.

To solve the convex but nondifferentiable optimization problem described by (30), one can adapt the TT nuclear norm to the algorithms such as SiLRTC, FaLRTC in [15]. Besides, in order to solve (31), we can apply the alternating least squares (ALS) technique to variationally optimize $U, V, \mathcal{X}$ until a convergence is obtained. Specifically, one can modify the algorithms such as TMac and TC-MLFM in [21] and [20], respectively by incorporating the concept of TT rank into them. The essential advantage of this multilinear matrix factorization model when compared to the model in (30) is that it avoids a lot of SVDs and hence can substantially save the computational time.

Let us propose the first algorithm to solve the optimization problem in (30) which is deeply rooted by the SiLRTC algorithm [15]. We call our algorithm "SiLRTC-TT" which stands for "simple low rank tensor completion via tensor train". The central concept of this algorithm is based on the BCD method to alternatively optimize a group of variables while the other groups remain fixed. More specifically, the variables are divided into two main groups. The first one contains the unfolding matrices $M_1, M_2, \ldots, M_{N-1}$ and the other is tensor $\mathcal{X}$. Computing each matrix M_k is related to solving the following optimization problem:

$$\min_{M_k} \quad \alpha_k ||M_k||_* + \frac{\beta_k}{2} ||X_{[k]} - M_k||_F^2, \tag{32}$$

with fixed $X_{[k]}$. The optimal solution for this problem has the closed form [13] which is determined by

$$M_k = \mathbf{D}_{\gamma_k}(X_{[k]}), \tag{33}$$

where $\gamma_k = \frac{\alpha_k}{\beta_k}$ and $\mathbf{D}_{\gamma_k}(X_{[k]})$ denotes the thresholding SVD of $X_{[k]}$ [12]. Specifically, if the SVD of $X_{[k]} = U\lambda V^T$, its thresholding SVD is defined as:

$$\mathbf{D}_{\gamma_k}(X_{[k]}) = U\lambda_{\gamma_k} V^T, \tag{34}$$

where $\lambda_{\gamma_k} = diag(\max(\lambda_l - \gamma_k, 0))$. After updating all the M_k matrices, we turn into another block to compute the tensor $\mathcal{X}$ which elements are given by

$$x_{i_1 \cdots i_N} = \begin{cases} \left(\frac{\sum_{k=1}^{N} \beta_k \text{fold}(M_k)}{\sum_{k=1}^{N} \beta_k} \right)_{i_1 \cdots i_N} & (i_1 \cdots i_N) \notin \Omega \\ t_{i_1 \cdots i_N} & (i_1 \cdots i_N) \in \Omega \end{cases} \tag{35}$$

The pseudo-code of this algorithm is given in Algorithm I. The convergence condition is reached when the relative error between two successive tensors $\mathcal{X}$ is smaller than a threshold. The algorithm is guaranteed to be converged and gives rise to a global solution since the objective in (30) is a convex and the nonsmooth term is separable. We can also apply this algorithm for the square model [32] by simply choosing the weights such that $\alpha_k = 1$ if $k = \text{round}(N/2)$ otherwise $\alpha_k = 0$. For this particular case, let us call the algorithm as SiLRTC-Square.

Algorithm I: SiLRTC-TT

Input: The observed data $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$, index set Ω.
Parameters: $\alpha_k, \beta_k, k = 1, \ldots, N-1$.

1: **Initialization:** $\mathcal{X}^0$, with $\mathcal{X}_\Omega^0 = \mathcal{T}_\Omega$, $l = 0$.
2: **While not converged do:**
3:   **for** $k = 1$ to $N-1$ **do**
4:     Unfold the tensor $\mathcal{X}^l$ to get $X_{[k]}^l$
5:     $M_k^{l+1} = \mathbf{D}_{\frac{\alpha_k}{\beta_k}}(X_{[k]}^l)$
6:   **end for**
7:   Update $\mathcal{X}^{l+1}$ from M_k^{l+1} by (35)
8: **End while**

Output: The recovered tensor $\mathcal{X}$ as an approximation of $\mathcal{T}$

In order to solve the problem given by (31), we apply the BCD method to alternatively optimize different groups of variables. Specifically, we can first solve the following problem:

$$\min_{U_k, V_k, X_{[k]}} \quad ||U_k V_k - X_{[k]}||_F^2, \tag{36}$$

for $k = 1, 2, \ldots, N-1$. As the problem is convex with respect to each block of variables U_k, V_k and $X_{[k]}$ while the other two are fixed, we have the following updates:

$$U_k^{l+1} = X_{[k]}^l (V_k^l)^T (V_k^l (V_k^l)^T)^\dagger, \tag{37}$$
$$V_k^{l+1} = ((U_k^{l+1})^T U_k^{l+1})^\dagger (U_k^{l+1})^T) X_{[k]}^l \tag{38}$$
$$X_{[k]}^{l+1} = U_k^{l+1} V_k^{l+1}, \tag{39}$$

where "$\dagger$"denotes the Moore-Penrose pseudoinverse. It was shown in [21] that, we can replace (37) by the following one:

$$U_k^{l+1} = X_{[k]}^l (V_k^l)^T, \tag{40}$$

to avoid computing the Moore-Penrose pseudoinverse $(V_k^l (V_k^l)^T)^\dagger$. The rationale behind this is that we only need the product $U_k^{l+1} V_k^{l+1}$ to compute $X_{[k]}^{l+1}$ as in (39) that is the same when either (37) or (40) is used. After updating U_k^{l+1}, V_k^{l+1} and $X_{[k]}^{l+1}$ for all $k = 1, 2, \ldots, N-1$, we compute elements of the tensor $\mathcal{X}^{l+1}$ as follows:

$$x_{i_1 \cdots i_N}^{l+1} = \begin{cases} \left(\sum_{k=1}^{N-1} \alpha_k \text{fold}(X_{[k]}^{l+1}) \right)_{i_1 \cdots i_N} & (i_1 \cdots i_N) \notin \Omega \\ t_{i_1 \cdots i_N} & (i_1 \cdots i_N) \in \Omega \end{cases} \tag{41}$$

Let us name the algorithm as TMac-TT which stands for "tensor completion by parallel matrix factorization in the concept of tensor train" and its pseudo-code is summarized in Algorithm II. Again, the Algorithm II can be applied for the square model [32] by choosing the weights such that $\alpha_k = 1$

if $k = \text{round}(N/2)$ otherwise $\alpha_k = 0$, and we call it as TMac-Square.

Algorithm II: TMac-TT

Input: The observed data $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$, index set Ω.
Parameters: $\alpha_i, r_i, i = 1, \ldots, N-1$.

1: **Initialization:** $U^0, V^0, \mathcal{X}^0$, with $\mathcal{X}_\Omega^0 = \mathcal{T}_\Omega$, $l = 0$.
While not converged do:
2: **for** $k = 1$ to $N-1$ **do**
3:     Unfold the tensor $\mathcal{X}^l$ to get $X_{[k]}^l$
4:     $U_i^{l+1} = X_{[k]}^l (V_k^l)^T$
5:     $V_k^{l+1} = ((U_k^{l+1})^T U_k^{l+1})^\dagger (U_k^{l+1})^T X_{[k]}^l$
6:     $X_{[k]}^{l+1} = U_k^{l+1} V_k^{l+1}$
7: **end**
8: Update the tensor $\mathcal{X}^{l+1}$ using (41)
End while

Output: The recovered tensor $\mathcal{X}$ as an approximation of $\mathcal{T}$

C. Computational complexity of algorithms

We analyze the computational complexity of algorithms applied to complete a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ in the Table I where we assume that $I_1 = I_2 = \cdots = I_N = I$, the Tucker rank and TT rank are the same $r_1 = r_2 = \cdots = r_N = r$.

Table I: Computational complexity of algorithms for one iteration.

Algorithm	Computational complexity
SiLRTC	$O(NI^{N+1})$
SiLRTC-TT	$O(I^{3N/2} + I^{3N/2-1})$
TMac	$O(3NI^N r)$
TMac-TT	$O(3(N-1)I^N r)$

IV. Tensor augmentation

In this section, we introduce the *ket augmentation* (KA) to represent a lower-order tensor by a higher-order one, i.e. to cast an Nth-order tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ into a Kth-order tensor $\tilde{\mathcal{T}} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_K}$, where $K \geq N$ and $\prod_{l=1}^N I_l = \prod_{l=1}^K J_l$. Higher-order representation of the tensor offers some important advantages. For instance, TT decomposition is more efficient for the augmented tensor because the local structure of the data can be exploited effectively in terms of computational resources. Actually, if the tensor is slightly correlated, its augmented tensor can be represented by a low-rank TT [8], [26].

The KA was originally introduced in [26] for casting a grayscale image into *real ket state* of a Hilbert space, which is simply a higher-order tensor, using an appropriate block structured addressing. Here we generalize the KA scheme for third-order tensors $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ that represent color images, where $I_1 \times I_2 = 2^n \times 2^n$ ($n \geq 1 \in \mathbb{Z}$) is the number of pixels in the image and $I_3 = 3$ is the number of colors (red, green and blue). Let us start with an initial block, labeled as i_1, of 2×2 pixels corresponding to a single color j (assume that

the color is indexed by j where $j = 1, 2, 3$ corresponding to red, green and blue colors, respectively). This block can be represented as

$$\mathcal{T}_{[2^1 \times 2^1 \times j]} = \sum_{i_1=1}^4 c_{i_1 j} \mathbf{e}_{i_1}, \tag{42}$$

where $c_{i_1 j}$ is the pixel value corresponding to color j and $\mathbf{e}_{i_1}$ is the orthonormal base which is defined as $\mathbf{e}_1 = (1, 0, 0, 0)$, $\mathbf{e}_2 = (0, 1, 0, 0)$, $\mathbf{e}_3 = (0, 0, 1, 0)$ and $\mathbf{e}_4 = (0, 0, 0, 1)$. The value $i_1 = 1, 2, 3$ and 4 can be considered as labeling the up-left, up-right, down-left and down-right pixels, respectively. For all three colors, we have three blocks which are presented by

$$\mathcal{T}_{[2^1 \times 2^1 \times 3]} = \sum_{i_1=1}^4 \sum_{j=1}^3 c_{i_1 j} \mathbf{e}_{i_1} \otimes \mathbf{u}_j, \tag{43}$$

where $\mathbf{u}_j$ is also an orthonormal base which is defined as $\mathbf{u}_1 = (1, 0, 0)$, $\mathbf{u}_2 = (0, 1, 0)$, $\mathbf{u}_3 = (0, 0, 1)$. We now consider
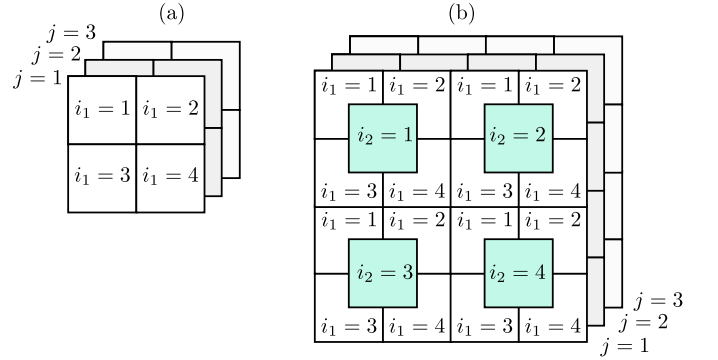

Figure 1: A structured block addressing procedure to cast an image into a higher-order tensor. (a) Example for an image of size $2 \times 2 \times 3$ represented by (43). (b) Illustration for an image of size $2^2 \times 2^2 \times 3$ represented by (44).

a larger block labeled as i_2 make up of four inner sub-blocks for each color j as shown in Fig. 1. In total, the new block is represented by

$$\mathcal{T}_{[2^2 \times 2^2 \times 3]} = \sum_{i_2=1}^4 \sum_{i_1=1}^4 \sum_{j=1}^3 c_{i_2 i_1 j} \mathbf{e}_{i_2} \otimes \mathbf{e}_{i_1} \otimes \mathbf{u}_j. \tag{44}$$

Generally, this block structure can be extended to a size of $2^n \times 2^n \times 3$ after several steps until it can present all the values of pixels in the image. Finally, the image can be cast into an $(n+1)$th-order tensor $\mathcal{C} \in \mathbb{R}^{4 \times 4 \times \cdots \times 4 \times 3}$ containing all the pixel values as follows,

$$\mathcal{T}_{[2^n \times 2^n \times 3]} = \sum_{i_n, \ldots, i_1=1}^4 \sum_{j=1}^3 c_{i_n \cdots i_1 j} \mathbf{e}_{i_n} \otimes \cdots \otimes \mathbf{e}_{i_1} \otimes \mathbf{u}_j. \tag{45}$$

When the image is represented by a real ket state, its entanglement entropy can reflect the correlation between individual pixels as due to their relative positions in the image. Besides, this presentation is suitable for the image processing as it not only preserves the pixels values of the image but also rearrange them in a higher-order tensor such that the richness of textures

in the image can be studied via the correlation between modes of the tensor [26].

V. SIMULATIONS

We apply the proposed algorithms for completing both synthetic data and color images. Simulation results are compared with commonly used algorithms, i.e. SiLRTC [15], SiLRTC-Square [32], TMac [21] and TMac-Square. To measure performance of a LRTC algorithm we compute the relative square error (RSE) between the approximately recovered tensor $\mathcal{X}$ and the original one $\mathcal{T}$, which is defined as,

$$RSE = ||\mathcal{X} - \mathcal{T}||_F / ||\mathcal{T}||_F. \qquad (46)$$

It is hard to choose the parameters in the models so that optimal solutions can be achieved. In experiments, we simply choose the weights α_k for our proposed algorithms as follows:

$$\alpha_k = \frac{\delta_k}{\sum_{k=1}^{N-1} \delta_k} \quad \text{with} \quad \delta_k = \min(\prod_{l=1}^{k} I_l, \prod_{l=k+1}^{N} I_l), (47)$$

where $k = 1, \dots, N-1$. The positive parameters are chosen by $\beta_k = f\alpha_k$, where f is empirically chosen from one of the following values in $[0.01, 0.05, 0.1, 0.5, 1]$ in such a way that the algorithm performs the best. For the algorithms used to compare with ours, i.e. SiLRTC and TMac the weights are chosen as follows:

$$\alpha_k = \frac{I_k}{\sum_{k=1}^{N} I_k}, \qquad (48)$$

where $k = 1, \dots, N$. The positive parameters are chosen such that $\beta_k = f\alpha_k$, where f is empirically chosen from one of the following values in $[0.01, 0.05, 0.1, 0.5, 1]$ which gives the best performance. The convergence criterion of our proposed algorithms is defined by computing the relative error of the tensor $\mathcal{X}$ between two successive iterations as follows:

$$\epsilon = \frac{||\mathcal{X}^{l+1} - \mathcal{X}^l||_F}{||\mathcal{T}||_F} \leq tol, \qquad (49)$$

where we set $tol = 10^{-4}$ and the maximum number of iterations is $maxiter = 1000$.

In what follows, we perform simulations for algorithms with respect to different missing ratios (mr) defined as,

$$mr = \frac{p}{\prod_{k=1}^{N} I_k}, \qquad (50)$$

where p is the number of missing entries which are often chosen randomly from the tensor $\mathcal{T}$ based on a uniform distribution. These simulations are implemented under a Matlab environment using the FEIT cluster from the University of Technology Sydney.

A. Synthetic data completion

We firstly perform the simulation on two different types of low-rank tensors which are generated synthetically in such a way that the Tucker and TT rank are known in advance.

1) Completion of low TT rank tensor: The Nth-order tensors $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ of TT rank $(r_1, r_2, \dots, r_{N-1})$ are generated such that its elements is represented by a TT format [8]. Specifically, its elements is $t_{i_1 i_2 \dots i_N} = A_{i_1}^{[1]} A_{i_2}^{[2]} \cdots A_{i_N}^{[N]}$, where $A^{[1]} \in \mathbb{R}^{I_1 \times r_1}$, $A^{[N]} \in \mathbb{R}^{r_N \times I_N}$ and $\mathcal{A}^{[k]} \in \mathbb{R}^{r_{k-1} \times I_k \times r_k}$ with $k = 2, \dots, N-1$ are generated randomly with respect to the standard Gaussian distribution $\mathcal{N}(0, 1)$. For simplicity, in this paper we set all components of the TT rank the same and so does the dimension of each mode, i.e. $r_1 = r_2 = \cdots = r_{N-1} = r$ and $I_1 = I_2 = \cdots = I_N = I$.

The plots of RSE with respect to mr are shown in the Figure. 2 for tensors of different sizes, $40 \times 40 \times 40 \times 40$ (4D), $20 \times 20 \times 20 \times 20 \times 20$ (5D), $10 \times 10 \times 10 \times 10 \times 10 \times 10$ (6D) and $10 \times 10 \times 10 \times 10 \times 10 \times 10 \times 10$ (7D) and the corresponding TT rank tuples are $(10, 10, 10)$ (4D), $(5, 5, 5, 5)$ (5D), $(4, 4, 4, 4, 4)$ (6D) and $(4, 4, 4, 4, 4, 4)$ (7D). From the plots we can see that TMac-TT shows best performance in most cases. Especially, TMac-TT can recover the tensor successfully despite the high missing ratios. Particularly, in most cases with high missing ratios, e.g. $mr = 0.9$, it can recover the tensor with $RSE \approx 10^{-4}$. More importantly, the proposed algorithms SiLRTC-TT and TMac-TT often performs better than their corresponding counterparts, i.e. SiLRTC and TMac in most cases.
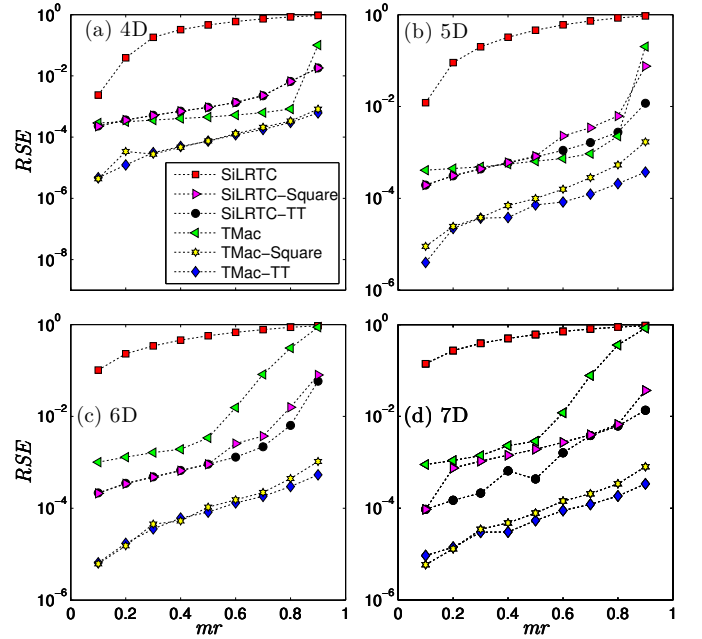

Figure 2: The RSE comparison when applying different LRTC algorithms to synthetic random tensors of low TT rank. Simulation results are shown for different tensor dimensions, 4D, 5D, 6D and 7D.

For a better comparison on the performance of different LRTC algorithms, we present the phase diagrams using the grayscale color to estimate how successfully a tensor can be recovered for a range of different TT rank and missing ratios. If $RSE \leq \epsilon$ where ϵ is a small threshold, we say that the tensor is recovered successfully and is represented by a white block in the phase diagram. Otherwise, if $RSE > \epsilon$, the tensor is recovered partially with a relative error and the block color is

gray. Especially the recovery is completely failed if $RSE = 1$. Concretely, we show in Fig. 3 the phase diagrams for different algorithms applied to complete a 5D tensor of size $20 \times 20 \times 20 \times 20 \times 20$ where the TT rank r varies from 2 to 16 and $\epsilon = 10^{-2}$. We can see that our LRTC algorithms outperform the others. Especially, TMac-TT always recovers successfully the tensor with any TT rank and missing ratio.
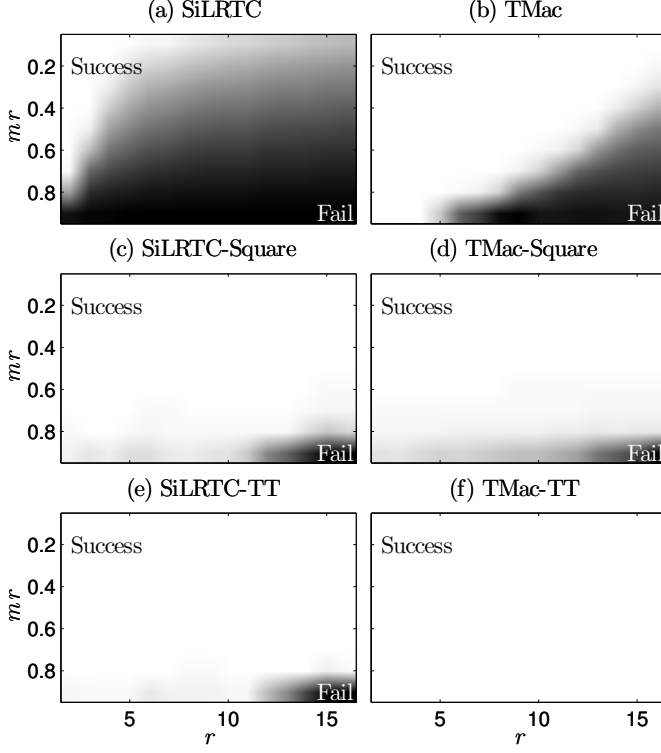

Figure 3: Phase diagrams for low TT rank tensor completion when applying different algorithms to a 5D tensor.

2) Completion of low Tucker rank tensor: Let us now apply our proposed algorithms to synthetic random tensors of low Tucker rank. The Nth-order tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ of Tucker rank $(r_1, r_2, \ldots, r_N)$ is constructed by $\mathcal{T} = \mathcal{G} \times_1 A^{(1)} \times_2 A^{(2)} \cdots \times_N A^{(N)}$, where the core tensor $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \cdots \times r_N}$ and the factor matrices $A^{(k)} \in \mathbb{R}^{r_k \times I_k}, k = 1, \ldots, N$ are generated randomly by using the standard Gaussian distribution $\mathcal{N}(0, 1)$. Here, we choose $r_1 = r_2 = \cdots = r_N = r$ and $I_1 = I_2 = \cdots = I_N = I$ for simplicity. To compare the performance between the algorithms, we show in the Fig. 4 the phase diagrams for different algorithms applied to complete a 5D tensor of size $20 \times 20 \times 20 \times 20 \times 20$ where the Tucker rank r varies from 2 to 16 and $\epsilon = 10^{-2}$. We can see that both TMac and TMac-TT perform much better than the others and. Besides, SiLRTC-TT shows better performance when compared to SiLRTC and SiLRTC-Square. Similarly, TMac-TT is better than its particular case TMac-Square.

In summary, we can see that although the tensors are generated synthetically to have low Tucker ranks, the proposed algorithms are still capable of producing results which are as good as the ones obtained by the Tucker-based algorithms. In order to have a better comparison between algorithms, we show results of applying them to the real world data such as

color images where the ranks of the tensors are not known in advance in the next subsection.
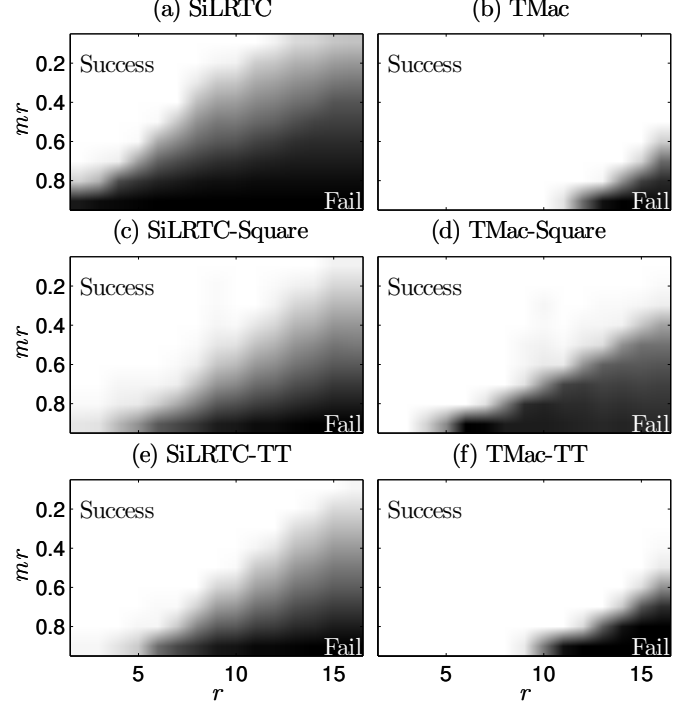

Figure 4: Phase diagrams for low Tucker rank tensor completion when applying different algorithms to a 5D tensor.

B. Image completion

A set of color images, namely "Peppers", "Lenna" and "House" are employed to test the algorithms with different missing ratios. All the images are initially represented by third-order tensors which have same sizes of $256 \times 256 \times 3$. Note
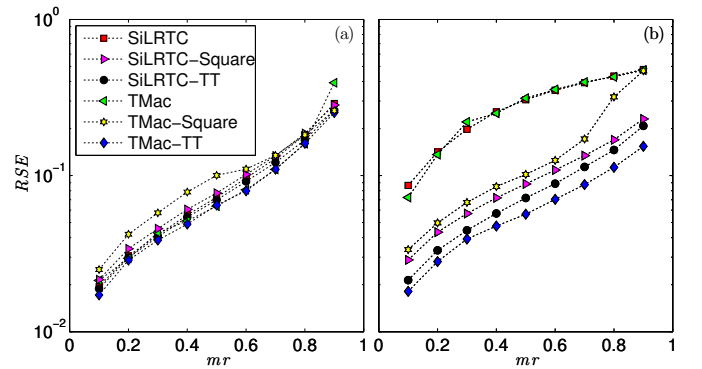

Figure 5: Performance comparison between different tensor completion algorithms based on the RSE vs the missing rate when applied to the Peppers image. (a) Original tensor (no order augmentation). (b) Augmented tensor using KA scheme.

that when completing the third-order tensors, we do not expect that our proposed methods prevail against the conventional ones due to the fact that the TT rank of the tensor is a special case of the Tucker rank. Thus, performance of the algorithms

should be mutually comparable. However, for the purpose of comparing the performance between different algorithms for real data (images) represented in terms of higher-order tensors, we apply tensor augmentation scheme KA mentioned above to reshape third-order tensors to higher-order ones without changing the number of entries in the tensor. Specifically, we start our simulation by casting a third-order tensor $\mathcal{T} \in \mathbb{R}^{256 \times 256 \times 3}$ into a ninth-order $\widetilde{\mathcal{T}} \in \mathbb{R}^{4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 3}$ and then applying the tensor completion algorithms to impute its missing entries. We perform the simulation for the Peppers and Lenna images where missing entries of each image are chosen randomly according to a uniform distribution, the missing ratio mr varies from 0.1 to 0.9. In Fig. 5, we compare performance of algorithms on completing the Peppers image. We can see that, when the image is represented by a third-order tensor, the performance of the algorithms are comparable (The TMac-TT is actually slightly better than the others in most of the missing ratios). However, for the case of the ninth-order tensors, the performance of the algorithms are rigorously distinguished. Specifically, our proposed algorithms (especially TMac-TT) prevail against the others. We also illustrate the recovered images for $mr = 0.7$ in Fig. 6. This shows that our proposed algorithms give really good results in the case of augmented tensors, meanwhile the compared algorithms seem to be inefficient. Furthermore, using the KA scheme to increase the tensor order, SiLRTC-TT and TMac-TT significantly improve the accuracy when compared to the cases without augmentation. More precisely, TMac-TT gives the best results $RSE \approx 0.088$ with respect to the case of using KA scheme. Same experiment is performed on the Lenna image and recovery results are shown in Fig. 7 and Fig. 8. The results also show that TMac-TT gives the best results for the augmented tensor using the KA scheme.
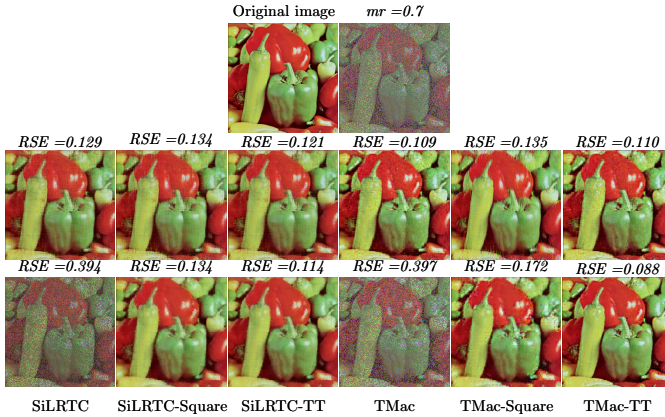

Figure 6: Recover the Peppers image with 70% of missing entries using different algorithms. Top row from left to right: the original image and its copy with 70% of missing entries. Second and third rows represent the recovery results of third-order (no order augmentation) and ninth-order tensors (KA augmentation), using different algorithms: SiLRTC, SiLRTC-Square, SiLRTC-TT, TMac, TMac-Square and TMac-TT from the left to the right, respectively.

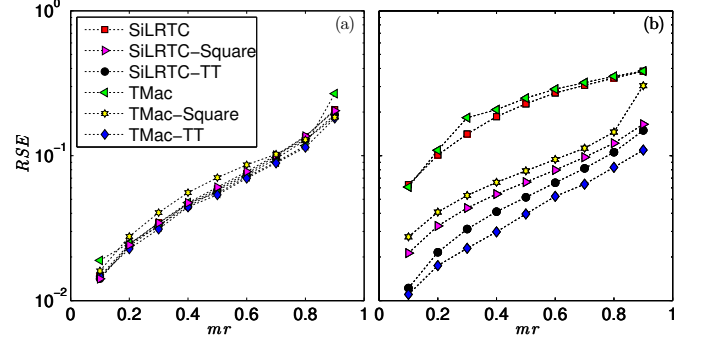We perform the same above experiment on the House image,

Figure 7: Performance comparison between different tensor completion algorithms based on the RSE vs the missing rate when applied to the Lenna image. (a) Original tensor (no order augmentation). (b) Augmented tensor using KA scheme.

Figure 8: Recover the Lenna image with 90% of missing entries using different algorithms. Top row from left to right: the original image and its copy with 90% of missing entries. Second and third rows represent the recovery results of third-order (no order augmentation) and ninth-order tensors (KA augmentation), using different algorithms: SiLRTC, SiLRTC-Square, SiLRTC-TT, TMac, TMac-Square and TMac-TT from the left to the right, respectively.

however, the missing entries are now chosen as the white text, and hence the missing rate is fixed. The result is shown in Fig. 9. In the cases of tensor augmentation, the conventional algorithms SiLRTC and TMac do not perform well meanwhile our proposed algorithms do. Using the KA scheme, better results can be achieved by employing our algorithms when compared to the case without using the augmentation schemes.

To sum up, we see that the TT-based algorithms outperforms the Tucker-based ones when applying to the images represented by ninth-order tensors $\widetilde{\mathcal{T}} \in \mathbb{R}^{4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 3}$. This is because the components of TT rank can approximately vary in a broad range of values (the maximum value it can reach is 256) to capture the global information of the images. On the contrary, the components of Tucker rank can have value up to 4 due to the mode-k matricization $X_k \in \mathbb{R}^{4 \times 49152}$. Consequently, the Tucker-based algorithms are not reliable due to the naturally small Tucker rank.
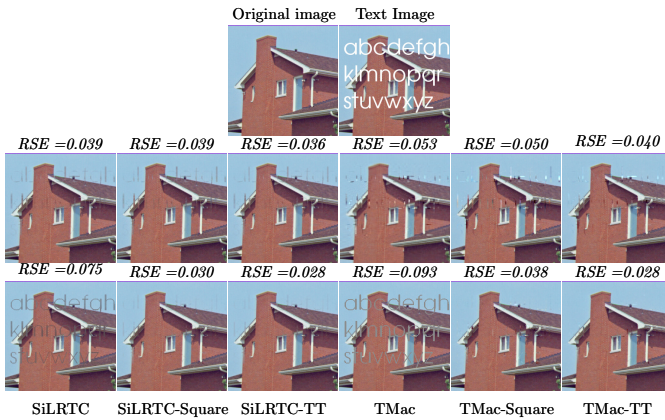
Figure 9: Recover the House image with missing entries described by the white letters using different algorithms. Top row from left to right: the original image and its copy with white letters. Second and third rows represent the recovery results of third-order (no order augmentation) and ninth-order tensors (KA augmentation), using different algorithms: SiLRTC, SiLRTC-Square, SiLRTC-TT, TMac, TMac-Square and TMac-TT from the left to the right, respectively.

VI. CONCLUSION

We have proposed efficient LRTC algorithms based on the concept of the TT decomposition. The SiLRTC-TT algorithm is applied to minimize the TT rank of the tensor by solving the TT nuclear norm optimization. Meanwhile, TMac-TT is based on the multilinear matrix factorization model to minimize the TT-rank. The latter is more computationally efficient due to the fact that it does not need the SVD which is different from the former. The proposed algorithms are employed to simulate with both synthetic and real world data represented by higher-order tensors and their performance are compared with their replicates, which are formulated in terms of Tucker rank. For synthetic data, on the one hand our algorithms prevail the other when the tensors have low TT rank. On the other hand, their performance are comparable in case of low Tucker rank tensors. Therefore, the TT-based algorithms are quite promising and reliable when applying to real world data. To validate this, we apply the algorithms to study the image completion problem. Benchmark results show that when applied to original tensors without order augmentation, all algorithms are comparable to each other. However, in the case of augmented tensors, our proposed algorithms not only outperform the others but also provide better recovery results when compared to the case without tensor order augmentation.

Although the proposed algorithms can potentially be applied to complete tensors with a wide range of low tensor ranks, i.e. Tucker rank or TT rank, the optimal parameters such as weights and TT rank cannot be chosen automatically rather than empirically. We plan to further improve the algorithms by developing a scheme to adaptively choose these parameters. Besides, their applications to data compression, text mining, image classification and video indexing are under our interest.

REFERENCES

[1] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[2] M. Vasilescu and D. Terzopoulos, "Multilinear subspace analysis of image ensembles," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2003. Proceedings.* Institute of Electrical & Electronics Engineers (IEEE), 2003.

[3] J.-T. Sun, H.-J. Zeng, H. Liu, Y. Lu, and Z. Chen, "Cubesvd: A novel approach to personalized web search," in *Proceedings of the 14th International Conference on World Wide Web*, ser. WWW '05. New York, NY, USA: ACM, 2005, pp. 382–390.

[4] T. Franz, A. Schultz, S. Sizov, and S. Staab, "Triplerank: Ranking semantic web data by tensor decomposition," in *The Semantic Web - ISWC 2009*, ser. Lecture Notes in Computer Science, A. Bernstein, D. Karger, T. Heath, L. Feigenbaum, D. Maynard, E. Motta, and K. Thirunarayan, Eds. Springer Berlin Heidelberg, 2009, vol. 5823, pp. 213–228.

[5] J. 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.

[6] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multi-modal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, no. 1, p. 84, 1970.

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

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

[9] M. Fannes, B. Nachtergaele, and R. Werner, "Finitely correlated states on quantum spin chains," *Communications in Mathematical Physics*, vol. 144, no. 3, pp. 443–490, 1992.

[10] A. Klmper, A. Schadschneider, and J. Zittartz, "Matrix product ground states for one-dimensional spin-1 quantum antiferromagnets," *EPL (Europhysics Letters)*, vol. 24, no. 4, p. 293, 1993. [Online]. Available: http://stacks.iop.org/0295-5075/24/i=4/a=010

[11] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac, "Matrix product state representations," *Quantum Info. Comput.*, vol. 7, no. 5, pp. 401–430, 2007.

[12] J.-F. Cai, E. J. Candès, and Z. Shen, "A Singular Value Thresholding Algorithm for Matrix Completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, Jan 2010.

[13] S. Ma, D. Goldfarb, and L. Chen, "Fixed point and Bregman iterative methods for matrix rank minimization," *Mathematical Programming*, vol. 128, no. 1-2, pp. 321–353, Sep 2009.

[14] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization," *SIAM Rev.*, vol. 52, no. 3, pp. 471–501, Jan 2010.

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

[16] M. Signoretto, L. De Lathauwer, and J. A. K. Suykens†, "Nuclear norms for tensors and their use for convex multilinear estimation," ESAT-SISTA, K.U.Leuven (Leuven, Belgium), Tech. Rep., 2010.

[17] M. Signoretto, R. Van de Plas, B. De Moor, and J. Suykens, "Tensor versus matrix completion: A comparison with application to spectral data," *Signal Processing Letters, IEEE*, vol. 18, no. 7, pp. 403–406, Jul 2011.

[18] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inverse Problems*, vol. 27, no. 2, p. 025010, Jan 2011.

[19] T. Ryota, S. Taiji, K. Hayashi, and H. Kashima, "Statistical performance of convex tensor decomposition," in *Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS)*, Dec 2011.

[20] H. Tan, B. Cheng, W. Wang, Y.-J. Zhang, and B. Ran, "Tensor completion via a multi-linear low-n-rank factorization model," *Neurocomputing*, vol. 133, pp. 161–169, Jun 2014.

[21] Y. Xu, R. Hao, W. Yin, and Z. Su, "Parallel matrix factorization for low-rank tensor completion," *IPI*, vol. 9, no. 2, pp. 601–624, Mar 2015.

[22] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '00. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 417–424.

[23] N. Komodakis, "Image Completion Using Global Optimization," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR)*. Institute of Electrical & Electronics Engineers (IEEE), 2006.

[24] T. Korah and C. Rasmussen, "Spatiotemporal Inpainting for Recovering Texture Maps of Occluded Building Facades," *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2262–2271, Sep 2007.

[25] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press (CUP), 2009.

[26] J. I. Latorre, "Image compression and entanglement," *arxiv*, vol. abs/quant-ph/0510031, 2005.

[27] M. Kurucz, A. A. Benczur, and K. Csalogany, "Methods for large scale SVD with missing values," *KDD Cup and Workshop at the 13th ACM SIGKDD Conference*, 2007.

[28] F. R. Bach, "Consistency of trace norm minimization," *J. Mach. Learn. Res.*, vol. 9, pp. 1019–1048, Jun 2008.

[29] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Mathematical Programming Computation*, vol. 4, no. 4, pp. 333–361, Jul 2012.

[30] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, "Mixed-state entanglement and quantum error correction," *Phys. Rev. A*, vol. 54, no. 5, pp. 3824–3851, Nov 1996.

[31] G. Vidal, "Efficient Simulation of One-Dimensional Quantum Many-Body Systems," *Phys. Rev. Lett.*, vol. 93, no. 4, Jul 2004.

[32] C. Mu, B. Huang, J. Wright, and D. Goldfarb, "Square deal: Lower bounds and improved relaxations for tensor recovery," in *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, ser. JMLR Proceedings, vol. 32. JMLR.org, 2014, pp. 73–81.