STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **13**, June 2025, pp 2560–2574.
Published online in International Academic Press (www.IAPress.org)

Randomized Algorithms for Low-Rank Tensor Completion in TT-Format

Yihao Pan [1], Congyi Yu [1], Chaoping Chen [1], Gaohang Yu [2,*]

[1] *Department of Mathematics, Hangzhou Dianzi University, 310018, China*
[2] *School of Science, Zhejiang University of Science and Technology, 310023, China*

Abstract Tensor completion is a crucial technique for filling in missing values in multi-dimensional data. It relies on the assumption that such datasets have intrinsic low-rank properties, leveraging this to reconstitute the dataset using low-rank decomposition or other strategies. Traditional approaches often lack computational efficiency, particularly with singular value decomposition (SVD) for large-scale tensor. Furthermore, fixed-rank SVD methods struggle with determining a suitable initial rank when data are incomplete. This paper introduces two novel randomized algorithms designed for low-rank tensor completion in tensor train (TT) format, named TTrandPI and FPTT. The TTrandPI algorithm integrates randomized tensor train (TT) decomposition with power iteration techniques, thereby enhancing computational efficiency and accuracy by improving spectral decay and minimizing tail energy build-up. Meanwhile, the FPTT algorithm utilizes a fixed-precision low-rank approximation approach that adaptively selects tensor ranks based on error tolerance levels, thus reducing the dependence on a predetermined rank. By conducting numerical experiments on synthetic data, color images, and video sequences, both algorithms exhibit superior performance compared to some existing methods.

Keywords Tensor completion, TT decomposition, randomized SVD, fixed-precision

AMS 2010 subject classifications 15A69, 15A83, 68W20

DOI: 10.19139/soic-2310-5070-2483

1. Introduction

Tensors, which are multidimensional generalizations of vectors and matrices, offer a suitable mathematical framework for representing complex data structures. As a result, high-dimensional tensors find extensive applications in various domains such as signal processing and machine learning[24, 26, 12]. One of the persistent challenges in scientific data analysis is the occurrence of incomplete data due to reasons such as sensor malfunctions, sampling limitations, or transmission errors. Tensor completion tackles this issue by reconstructing missing elements from partial data, thus advancing the established low-rank matrix completion (LRMC)[4] approach to encompass higher-dimensional structures through low-rank tensor completion (LRTC)[13]. The matrix completion problem aims to reconstruct a matrix with the minimal possible rank that exactly matches the observed entries from a partially sampled dataset, that is,

$$\min_{A} \quad \text{rank}(A) \quad \text{s.t.} \quad A_\Omega = M_\Omega. \tag{1}$$

where M is a matrix with missing entries, Ω is the set of indices for the known elements. For the rank minimization issue in tensor completion, we can solve it through low rank decomposition. However, Problem 1 is an NP-hard problem due to the combinational nature of the function rank(). A common approach is to use matrix decomposition

*Correspondence to: Gaohang Yu (E-mail addresses: maghyu@163.com).School of Science, Zhejiang University of Science and Technology, 310023, China

ISSN 2310-5070 (online) ISSN 2311-004X (print)

to solve this problem. For any rank r matrix $A \in \mathbb{R}^{m\times n}$ can be written into a matrix product from $A = XY$ where $X \in \mathbb{R}^{m\times r}$ and $Y \in \mathbb{R}^{r\times n}$. Low-rank factorization model as following:

$$\min_{X,Y,A} \frac{1}{2}\|XY - A\|_F^2 \quad \text{s.t.} \quad A_\Omega = M_\Omega. \tag{2}$$

Tensor completion faces fundamental complexities due to the non-uniqueness and computational intractability of tensor rank definitions. There are many classic decomposition methods, such as CP decomposition[7], Tucker decomposition[8], TT decomposition[16] and others. CP decomposition, which can decompose a tensor into a linear combination of rank-1 tensors, is a popular method for solving low-rank tensor completion problems. Since determining the CP rank is an NP-hard problem, this poses significant limitations for CP decomposition. Some individuals have attempted to utilize Tucker decomposition to address LRTC. Liu et al.[13] employed the tensor trace norm to generalize matrix completion to tensor completion in the Tucker form. Nevertheless, the Tucker rank falls short in capturing the global correlations within tensors, rendering it less than ideal for LRTC. In contrast, the TT rank, which is defined by the ranks of matrices derived from a balanced matricization scheme, proves to be a highly effective tool. Bengua et al.[3] combined the matrix completion algorithm TMac and proposed the tensor completion algorithm TT-TMac based on TT decomposition. Moreover, by utilizing KA augmentation, they transformed low-order tensors into high-order tensors, thereby better utilizing local information. Yu et al.[20] combined multi-modal TT decomposition with spatial spectral smoothing regularization to alleviate the block effect.

Deterministic algorithms often struggle with large datasets due to substantial computational requirements and lengthy execution durations. Randomized approaches offer a viable alternative. Huber et al.[11] significantly enhanced the performance of TT decomposition by incorporating randomized techniques. Chen et al.[6] introduced an innovative least-squares randomized approach for low-rank TT decomposition using TensorSketch. Qin et al.[17] explored the robust high-order tensor completion (RHTC) issue through a randomized low-rank approximation within the tensor singular value decomposition (T-SVD) framework. Ahmadi-Asl et al.[1] examined both adaptive and non-adaptive stochastic algorithms applied to large-scale tensor data in a tensor ring configuration. However, studies on randomized algorithms for tensor completion leveraging TT decomposition remain relatively scarce.

On the other hand, in the face of large-scale data missing, the selection of target rank has always been a significant challenge. Similar issues in matrix low-rank decomposition can be addressed using adaptive rank methods. Martinsson et al.[14] proposed the randomized algorithm for partial decomposition of matrices, $randQB\_b$, achieving fixed-precision low-rank approximation. Building on this, Yu et al.[22] introduced an adaptive rank adjustment mechanism and single-pass data access optimization, significantly reducing computational complexity and memory consumption. Feng et al.[9] further enhanced the computational efficiency and accuracy of large-scale matrix low-rank approximation by replacing QR decomposition with matrix multiplication and inversion, introducing a dynamic error evaluation mechanism, and shifted power iteration technique. In recent years, there have also been some studies on completion related to adaptive rank. For example, Zhang et al.[25] redefined the CP rank and proposed an adaptive low-rank representation model for tensor completion, which can automatically determine the tensor rank. Xu et al.[19] used low-rank matrix decomposition to recover low-rank tensors and adopted an adaptive rank adjustment strategy when the exact rank is unknown. Che et al.[5] explored the application of fixed TT-rank and precision in randomized TT low-rank approximation. However, the research on applying adaptive rank in the field of TT format tensor completion is still relatively limited.

This paper introduces two novel randomized algorithms for low-rank tensor completion in tensor train (TT) format, namely TTrandPI and FPTT. By integration of power iteration schemes with randomized SVD, the TTrandPI algorithm enhances completion, particularly effective under conditions of high data omission. Conversely, the FPTT algorithm addresses inaccuracies from challenging initial rank selection by employing a fixed-precision completion strategy. Numerical tests indicate that these algorithms outperform some existing TT completion methods.

This paper is structured as follows: Section 2 discusses the essential background on tensor TT decomposition and the notion of fixed-precision low-rank approximation. In Section 3, we introduce a randomized tensor completion approach based on tensor-train (TT) decomposition, along with an analysis of the algorithm's convergence. Section

4 presents the fixed-precision tensor completion algorithm. Section 5 evaluates the results of numerical experiments examining the completion effects. Finally, Section 6 summarizes the conclusions.

2. Background

Some common symbols used in this paper are shown in the following Table 1.

a	Scalar
$\mathbf{a}$	Vector
A	Matrix
$\mathcal{A}$	Tensor
$\mathcal{A}(i_1, i_2, \cdots, i_d)$	the $(i_1, i_2, \cdots, i_d) - th$ element of d^{th} order tensor $\mathcal{A}$
$\times_n$	Mode-n Product of tensor and matrix
I_n	Identity matrix with size $n \times n$
$\sigma_i(\mathbf{A})$	the ith largest singular value of $\mathbf{A}$
$\mathbf{A}^\top$	Transpose of $\mathbf{A}$
$\mathbf{A}^\dagger$	Pseudo-inverse of $\mathbf{A}$

Table 1. Common symbols used in this paper

Suppose that two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, the Frobenius norm of a tensor $\mathcal{A}$ is given by $\|\mathcal{A}\|_F = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$ and the scalar product $\langle \mathcal{A}, \mathcal{B} \rangle$ is defined as [8],

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} a_{i_1 i_2 \cdots i_N} b_{i_1 i_2 \cdots i_N} := \mathcal{A} \times_{1,2,\ldots,N}^{1,2,\ldots,N} \mathcal{B}.$$

The mode-α product of tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ by a matrix $B \in \mathbb{R}^{M \times n_\alpha}$ is designated as $\mathcal{A} \times_\alpha B = \mathcal{C} \in \mathbb{R}^{n_1 \times \cdots \times n_{\alpha-1} \times M \times n_{\alpha+1} \times \cdots \times n_d}$, with entries:

$$\mathcal{C}(i_1, \cdots, i_{\alpha-1}, m, i_{\alpha+1}, \cdots, i_d) = \sum_{i_u=1}^{i_\alpha} \mathcal{A}(i_1, \cdots, i_{\alpha-1}, i_u, i_{\alpha+1}, \cdots, i_d) B(M, i_u). \tag{3}$$

The tensor-tensor product of two tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and $\mathcal{B} \in \mathbb{R}^{m_1 \times \cdots \times m_e}$ with equal modes $n_\alpha = m_\beta$ produces an $(d + e - 2)$-th order tensor $\mathcal{C}$, i.e.

$$\begin{aligned} &C(i_1, \ldots, i_{\alpha-1}, i_{\alpha+1}, \ldots, i_d, j_1, \ldots, j_{\beta-1}, j_{\beta+1}, \ldots, j_e) \\ &= \sum_{i_u=1}^{n_\alpha} \mathcal{A}(i_1, \ldots, i_{\alpha-1}, i_u, i_{\alpha+1}, \ldots, i_d) \cdot \mathcal{B}(j_1, \ldots, j_{\beta-1}, i_u, j_{\beta+1}, \ldots, j_e). \end{aligned} \tag{4}$$

where $\mathcal{C} \in \mathbb{R}^{n_1 \times \cdots \times n_{\alpha-1} \times n_{\alpha+1} \times \cdots \times n_d \times m_1 \times \cdots \times m_{\beta-1} \times m_{\beta+1} \times \cdots \times m_e}$.

Definition 1
(Matricization [11]) Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ be a tensor of order d. The α-matricization is defined as $\hat{M}_N(\mathcal{A}) \in \mathbb{R}^{m_\alpha \times m_\beta}$. The matrix dimensions are provided as $m_\alpha = \prod_{j=1}^{N} n_j$ and $m_\beta = \prod_{j=N+1}^{d} n_j$.

Definition 2
(Tensor Train Format [16]) Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ be a tensor of order d. A factorization

$$\mathcal{A} = \mathcal{G}_1 \times_3^1 \mathcal{G}_2 \times_3^1 \cdots \times_3^1 \mathcal{G}_d \tag{5}$$

of $\mathcal{A}$, into core tensors $\mathcal{G}_i \in \mathbb{R}^{r_{i-1} \times n_i \times r_i} (r_0 = r_d = 1)$, is called a tensor train(TT) decomposition of $\mathcal{A}$. The array of the dimensions $\mathbf{r} = (r_1, \cdots, r_{d-1})$ is the tensor train rank(TT-rank) of $\mathcal{A}$ defined as

$$\text{rank}_{\text{TT}}(\mathcal{A}) = (rank(\hat{M}_1(\mathcal{A})), \cdots, rank(\hat{M}_{d-1}(\mathcal{A}))). \tag{6}$$

Definition 3
(Tail Energy [10]). The j-th tail energy of matrix A is defined as

$$\tau_j^2(A) = \min_{\text{rank}(B)<j} \|A - B\|_F^2 = \sum_{i \geq j} \sigma_i^2(A), \tag{7}$$

where $\sigma_i(A)$ is the i-th singular value of A.

3. A randomized algorithm for TT low-rank tensor completion

In practical tensor completion problems, high dimensional data is often low-rank. For the rank minimization issue in tensor completion, we can solve it through low rank decomposition. Let $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ be a known tensor with missing entries, where Ω is the index set for the observed data, $\overline{\Omega}$ is the complementary set of Ω, and M_Ω represents the observed entries. Tensor decomposition is used to describe its low-rankness, and the tensor completion model is as follows:

$$\min_{\mathcal{A}} \quad rank(\mathcal{A}) \quad \text{s.t.} \quad \mathcal{A}_\Omega = \mathcal{M}_\Omega. \tag{8}$$

We use TT low-rank approximation to characterize the low-rank part. The TT low-rank approximation model for solving the tensor $\mathcal{A}$ can be expressed as[5]

$$\min_{\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_d} \|\mathcal{A} - \mathcal{G}_1 \times_3^1 \mathcal{G}_2 \times_3^1 \cdots \times_3^1 \mathcal{G}_d\|_F, \tag{9}$$

where $\mathcal{G}_i \in \mathbb{R}^{r_{i-1} \times n_i \times r_i} (r_0 = r_d = 1, i \in [1, d])$ for $i = 1, 2, \cdots, d-1$, TT core $\mathcal{G}_i$ satisfy

$$G_i^\top G_i = I_{r_i}, G_i = reshape(\mathcal{G}_i, [r_{i-1} n_i, r_i]).$$

Assuming $\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_d$ for a solution of (9). Let $T_i = reshape(\mathcal{T}_i, [r_{i-1} n_i, r_i]) (i = 1, 2, \cdots, d-1)$. Define

$$\begin{cases} \mathcal{A}_0 = \mathcal{A}, \\ \mathcal{A}_1 = \mathcal{T}_1 \times_1^1 \mathcal{A}_0 \in \mathbb{R}^{r_1 \times n_2 \times \cdots \times n_d}, \\ \mathcal{A}_i = \mathcal{T}_i \times_{1,2}^{1,2} \mathcal{A}_{i-1} \in \mathbb{R}^{r_i \times n_{i+1} \times \cdots \times n_d}, i = 2, 3, \cdots, d-1, \\ A^{(i)} = reshape(\mathcal{A}_i, [r_{i-1} n_i, n_{i+1} \cdots n_d]), i = 1, 2, \cdots, d-1. \end{cases}$$

We have

$$\|\mathcal{A} - \mathcal{G}_1 \times_3^1 \mathcal{G}_2 \times_3^1 \cdots \times_3^1 \mathcal{G}_d\|_F \leq \sum_{i=1}^{d-1} \left\| A^{(i)} - T_i T_i^\top A^{(i)} \right\|_F. \tag{10}$$

To derive an approximate result, consider addressing the subsequent $d-1$ subproblem with $r_i \leq \min\{r_{i-1} n_i, n_{i+1}, \cdots, n_d\} (r_0 = 1)$. The objective is to identify an orthogonal matrix $T_i \in \mathbb{R}^{r_{i-1} n_i \times r_i}$ that fulfills

$$T_i = \arg\min_{Q_i} \left\| A^{(i)} - Q_i Q_i^\top A^{(i)} \right\|_F, \quad \text{for} \quad i = 1, 2, \ldots, d-1, \tag{11}$$

wherein $Q_i \in \mathbb{R}^{r_{i-1} n_i \times r_i}$ maintains orthogonality.

Furthermore, solving the LRTC problem through TT decomposition, we have

$$\begin{aligned} \min_{\mathcal{A}, \mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_d} \quad & \|\mathcal{G}_1 \times_3^1 \mathcal{G}_2 \times_3^1 \cdots \times_3^1 \mathcal{G}_d - \mathcal{A}\|_F, \\ \text{s.t.} \quad & \mathcal{A}_\Omega = \mathcal{M}_\Omega. \end{aligned} \tag{12}$$

As a fundamental subtask of LRTC, tensor low-rank decomposition aims to compress large scale tensor data while preserving its essential structural information, thereby significantly reducing storage requirements and computational overhead. A classic algorithm for TT decomposition is TT-SVD[16], which, however, involves the SVD of large-scale matrices and thus has a high computational cost when applied to high-dimensional tensors. To address this challenge, random methods can be leveraged to drastically reduce the time and memory complexity of tensor decomposition. Furthermore, the integration of power iteration schemes accelerates the spectral decay of singular values, which not only suppresses tail energy accumulation but also enhances approximation accuracy through targeted rank truncation. Yu et al.[21] proposed the TT-rSI algorithm, which employs a random approach to accelerate the SVD process of unfolded matrices in TT-SVD, and utilizes subspace power iteration techniques to enhance the algorithm's precision and reduce the impact of noise. The detailed is provided in Algorithm 1. Let $r = max(r_1, \cdots, r_{d-1})$, the arithmetic cost of Algorithm 1 is $\mathcal{O}(n^d(r+s)q + \sum_{i=1}^{d-1} rn^{d-i}(r+s)q)$.

Algorithm 1 TT-rSI

Input: Tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, target rank $\mathbf{r} = (r_1, \cdots, r_{d-1})$, oversampling parameter $s \geq 2$, and $r_0 = 1$;
Output: cores $\mathcal{G}_1, \cdots, \mathcal{G}_d$.

$A^{(1)} := \text{reshape}(\mathcal{A}, [r_0 n_1, \frac{numel(\mathcal{A})}{r_0 n_1}])$.
for $i = 1$ to $d-1$ **do**
  Create random Gaussian matrices $\Psi^{(i)} \in \mathbb{R}^{(n_{i+1}\cdots n_d)\times(r_i+s)}$.
  $Y^{(i)} = A^{(i)}\Psi^{(i)}$.
  $[Q_0^{(i)}, \backsim] = qr(Y^{(i)}, 0)$.
  for $j = 1$ to q **do**
    $\hat{Y}_j^{(i)} = (A^{(i)})^T Q_{j-1}^{(i)}$.
    $(\hat{Q}_j^{(i)}, \backsim) = qr(\hat{Y}_j^{(i)})$.
    $Y_j^{(i)} = A^{(i)}\hat{Q}_j^{(i)}$.
    $(Q_j^{(i)}, \backsim) = qr(Y_j^{(i)})$.
  end for
  $Q^{(i)} = Q_q^{(i)}$ % $Q^{(i)} = Q_0^{(i)}$ when $q = 0$
  $\mathcal{G}_i = \text{reshape}(Q^{(i)}, [r_{i-1}, n_i, r_i])$.
  $A^{(i)} = \text{reshape}((Q^{(i)})^T A^{(i)}, [r_i n_{i+1}, \frac{numel((Q^{(i)})^T A^{(i)})}{r_i n_{i+1}}])$.
end for
$\mathcal{G}_d = \text{reshape}(A^{(i)}, [r_{d-1}, n_d, r_d])$.
return $\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_d$.

Lemma 1
([23], Theorem 2). Let $s \geq 2$ be the oversampling parameter and $\Psi \in \mathbb{R}^{r_i \times n}$ be a Gaussian random matrix. Suppose $Q^{(i)}$ is obtained from Algorithm 1. Then we have

$$\mathbb{E}_\Psi \left\| A^{(i)} - Q^{(i)} {Q^{(i)}}^\top A^{(i)} \right\|_F^2 \leq \left(1 + \frac{r_i}{s-1}\varpi_{r_i}^{4q}\right) \cdot \tau_{r_i+1}^2(A^{(i)}),$$

where $\varpi_k := \sigma_{k+1}/\sigma_k \ll 1$ is the singular value gap.

Combined (10) with Lemma 1, we have the following theorem.

Theorem 1
Let $\hat{\mathcal{A}}$ be the TT approximation of a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ by the Algorithm 1 with target TT-rank $\mathbf{r} = (r_1, \cdots, r_{d-1})$, we have

$$\mathbb{E}\|\mathcal{A} - \hat{\mathcal{A}}\|_F \leq \sum_{i=1}^{d-1} \left(1 + \frac{r_i}{s-1}\varpi_{r_i}^{4q}\right) \cdot \tau_{r_i+1}^2(A^{(i)}). \tag{13}$$

Algorithm 2 TTrandPI

Input: Tensor $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, index set Ω , tolerance ϵ, target rank $\mathbf{r} = (r_1, \cdots, r_{d-1})$, oversampling parameter $s \geq 2$ and integer $q \geq 0$;
Output: Completed tensor $\mathcal{A}_k$;
1: **Initialization:** $k = 0, \hat{\mathcal{A}}_0 = \mathcal{M}_\Omega$.
2: **while** not convergent **do**
3: $\mathcal{A} = \hat{\mathcal{A}}_k$.
4: $(\mathcal{G}_1, \cdots, \mathcal{G}_d) =$ TT-rSI$(\mathcal{A})$ % (Algorithm1).
5: $\hat{\mathcal{A}}_k = \mathcal{G}_1 \times_3^1 \cdots \times_3^1 \mathcal{G}_d$.
6: $\mathcal{A}_{k+1} = \mathcal{M}_\Omega + (\hat{\mathcal{A}}_k)_{\overline{\Omega}}$.
7: $k = k + 1$.
8: **end while**

By combining random strategies with power iteration, we employ the TT low-rank approximation to address the LRTC problem and introduce Algorithm 2. Through Algorithm 1, each step involves updating the TT low-rank approximation $\hat{\mathcal{A}}_k$. The approximate results are then produced. If these do not satisfy the convergence conditions and the iteration count has not yet reached its upper limit, the updating procedure is repeated.

When the estimated rank is reached to the exact rank and the precision error δ tends to 0 , by definition of the tail energy, each $\tau^2_{r_i+1}(A^{(i)})$ will tend to 0, we have :

$$\lim_{\delta \to 0} \mathbb{E}\|\mathcal{A}_k - \hat{\mathcal{A}}_k\|_F^2 \leq \lim_{\delta \to 0} \sum_{i=1}^{d-1} \left(1 + \frac{r_i}{s-1} \varpi_{r_i}^{4q}\right) \cdot \tau^2_{r_i+1}(A^{(i)}) = 0. \tag{14}$$

Theorem 2
The sequence $\{\mathcal{A}_k\}$ generated by Algorithm 2 is convergent.

Proof. From step 6 of Algorithm 2, we have

$$\mathcal{A}_{k+1} = \mathcal{M}_\Omega + (\hat{\mathcal{A}}_k)_{\overline{\Omega}}. \tag{15}$$

So,

$$\mathcal{A}_{k+1} - \mathcal{A}_k = (\hat{\mathcal{A}}_k - \hat{\mathcal{A}}_{k-1})_{\overline{\Omega}}. \tag{16}$$

Then, we obtain

$$\begin{aligned} \mathbb{E}\|\mathcal{A}_{k+1} - \mathcal{A}_k\|_F^2 =& \mathbb{E}\|(\hat{\mathcal{A}}_k - \hat{\mathcal{A}}_{k-1})_{\overline{\Omega}}\|_F^2 \\ \leq& \mathbb{E}\|\hat{\mathcal{A}}_k - \hat{\mathcal{A}}_{k-1}\|_F^2 \\ =& \mathbb{E}\|\hat{\mathcal{A}}_k - \mathcal{A}_k + \mathcal{A}_k - \mathcal{A}_{k-1} + \mathcal{A}_{k-1} - \hat{\mathcal{A}}_{k-1}\|_F^2 \\ \leq& \mathbb{E}\|\hat{\mathcal{A}}_k - \mathcal{A}_k\|_F^2 + \mathbb{E}\|\mathcal{A}_k - \mathcal{A}_{k-1}\|_F^2 + \mathbb{E}\|\mathcal{A}_{k-1} - \hat{\mathcal{A}}_{k-1}\|_F^2. \end{aligned} \tag{17}$$

From (14), we know that

$$\lim_{\delta \to 0} \mathbb{E}\|\mathcal{A}_k - \hat{\mathcal{A}}_k\|_F^2 = \lim_{\delta \to 0} \mathbb{E}\|\mathcal{A}_{k-1} - \hat{\mathcal{A}}_{k-1}\|_F^2 = 0. \tag{18}$$

Since

$$\mathbb{E}\|\mathcal{A}_{k+1} - \mathcal{A}_k\|_F^2 \leq \mathbb{E}\|\mathcal{A}_k - \mathcal{A}_{k-1}\|_F^2, \tag{19}$$

then, the sequence $\{\mathcal{A}_k\}$ generated by Algorithm 2 is convergent. □

Based on Theorem 2, the convergence condition for Algorithm 2 could be set to calculate the relative error of the tensor $\mathcal{A}$ over consecutive iterations: $res = \|\mathcal{A}_k - \mathcal{A}_{k-1}\|_F / \|\mathcal{A}_{k-1}\|_F \leq \epsilon$. In our numerical tests, we set $\epsilon = 10^{-4}$ and restrict the maximum number of iterations to 50.

4. Fixed-precision tensor completion approach based on TT decomposition

When addressing low-rank tensor completion issues, one frequently encounters challenges in choosing the initial rank due to the absence of some original data. Typically, most publications opt to determine the rank directly from either the complete original data or whatever incomplete data is accessible. Nevertheless, this method presents two primary challenges. Firstly, the original data often cannot be directly observed, complicating our efforts to estimate the initial rank. Secondly, incomplete datasets often result in a target rank that considerably diverges from reality. The choice of initial rank critically affects the algorithm's completion performance, leading researchers to invest substantial time and resources in identifying the suitable rank. The farPCA algorithm proposed by Feng et al.[9] is a fast adaptive random PCA algorithm designed to adaptively determine the rank (number of dimensions) of PCA based on a preset error tolerance. The specific algorithm is presented in Algorithm 3. The flop count of Algorithm 3 is

$$\mathrm{FC} = 2C_{\text{mul}} \cdot \mathrm{nnz}(A)k + C_{\text{mul}}(2m+2n)k^2 + q\left[C_{\text{mul}} \cdot \mathrm{nnz}(A)\left(k + \frac{k^2}{b}\right) + C_{\text{mul}} \cdot n(k-b)^2 + 2C_{\text{mul}} \cdot nkb\right] \tag{20}$$

where $2C_{mul} \cdot \mathrm{nnz}(A)k$ reflects the matrix-matrix multiplication on A in Step 11, $C_{mul}(2m+2n)k^2$ reflects the matrix-matrix multiplication in Steps 13 and 20, and $q\left(C_{mul} \cdot \mathrm{nnz}(A)\left(k + \frac{k^2}{b}\right) + C_{mul} \cdot n(k-b)^2 + 2C_{mul} \cdot nkb\right)$ reflects the operations in power iteration in Step 4 through 10.

Algorithm 3 farPCA

Input: $A \in \mathbb{R}^{m\times n}$, error tolerance ε, block size b, power parameter q ;
Output: $\mathbf{U} \in \mathbb{R}^{m\times k}$, $\mathbf{S} \in \mathbb{R}^{k\times k}$, $\mathbf{V} \in \mathbb{R}^{n\times k}$ such that $\|\mathbf{A} - \mathbf{USV}^\top\|_F < \varepsilon$.

1: $\mathbf{Y} \leftarrow [], \mathbf{W} \leftarrow [], E \leftarrow \|\mathbf{A}\|_F^2$, tol $\leftarrow \varepsilon^2$
2: **for** $i = 1, 2, \cdots$ **do**
3: $\quad\Omega_i \leftarrow \mathrm{randn}(n, b)$, $\alpha \leftarrow 0$
4: $\quad$**for** $j = 1, 2, \cdots, q$ **do**
5: $\quad\quad\mathbf{W}_i \leftarrow \mathbf{A}^\top\mathbf{A}\Omega_i - \mathbf{WZ}^{-1}\mathbf{W}^\top\Omega_i - \alpha\Omega_i$
6: $\quad\quad[\Omega_i, \hat{\mathbf{S}}, \sim] \leftarrow \mathrm{eigSVD}(\mathbf{W}_i)$
7: $\quad\quad$**if** ($j > 1$ and $\alpha < \hat{\mathbf{S}}(b,b)$) **then**
8: $\quad\quad\quad\alpha \leftarrow (\alpha + \hat{\mathbf{S}}(b,b))/2$
9: $\quad\quad$**end if**
10: $\quad$**end for**
11: $\quad\mathbf{Y}_i \leftarrow \mathbf{A}\Omega_i$, $\mathbf{W}_i \leftarrow \mathbf{A}^\top\mathbf{Y}_i$
12: $\quad\mathbf{Y} \leftarrow [\mathbf{Y}, \mathbf{Y}_i]$, $\mathbf{W} \leftarrow [\mathbf{W}, \mathbf{W}_i]$
13: $\quad\mathbf{Z} \leftarrow \mathbf{Y}^\top\mathbf{Y}$, $\mathbf{T} \leftarrow \mathbf{W}^\top\mathbf{W}$
14: $\quad$**if** $E - \mathrm{tr}(\mathbf{TZ}^{-1}) <$ tol **then**
15: $\quad\quad$**break**
16: $\quad$**end if**
17: **end for**
18: $[\hat{\mathbf{V}}, \hat{\mathbf{D}}] \leftarrow \mathrm{eig}(\mathbf{Z}), \mathbf{P} \leftarrow \hat{\mathbf{V}}\mathrm{sqrt}(\hat{\mathbf{D}})^{-1}$
19: $[\tilde{\mathbf{V}}, \tilde{\mathbf{D}}] \leftarrow \mathrm{eig}(\mathbf{P}^\top\mathbf{TP}), \mathbf{S} \leftarrow \mathrm{sqrt}(\tilde{\mathbf{D}})$
20: $\mathbf{U} \leftarrow \mathbf{YP}\tilde{\mathbf{V}}$, $\mathbf{V} \leftarrow \mathbf{WP}\tilde{\mathbf{V}}\mathbf{S}^{-1}$

It generates an initial subspace through a random projection matrix and then expands the basis vectors step by step. In each iteration, the orthogonal basis matrix Q and the coefficient matrix B are updated through matrix multiplication and inversion. The current approximation error is dynamically calculated, and the iteration is terminated if it meets the preset tolerance. A dynamic shift parameter is introduced in the power iteration to adjust the direction of the subspace and improve accuracy. The final principal components are quickly obtained through eigenvalue decomposition.

By integrating a fixed-precision low-rank approximation algorithm with a tensor completion algorithm, we introduce the FPTT algorithm(Algorithm 4). This innovative approach adaptively determines the rank based on

Algorithm 4 Fixed-precision TT(FPTT)

Input: Tensor $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, index set Ω , tolerance ϵ, $r_0 = \cdots = r_d = 1$ and integer $q \geq 0$;
Output: Completed tensor $\mathcal{A}_k$;

Initialization: $k = 0, \mathcal{A}_0 = \mathcal{M}_\Omega$.
while not convergent **do**
  $\mathcal{A} = \mathcal{A}_k$.
  for $i = 1$ to $d - 1$ **do**
    $A^{(i)}$ =reshape$(\mathcal{A}, [r_{i-1}n_i, \frac{numel(\mathcal{A})}{r_{i-1}n_i}])$.
    $[U^{(i)}, S^{(i)}, V^{(i)}] = farPCA(A^{(i)})$ % (Algorithm3).
    $\mathcal{G}_i =$ reshape$(U^{(i)}, [r_{i-1}, n_i, r_i])$.
    $A^{(i)}$ =reshape$(S^{(i)}(V^{(i)})^T, [n_1, ..., n_d])$.
  end for
  $\mathcal{G}_d = (U^{(d-1)})^T A^{(d-1)}$.
  $\hat{\mathcal{A}}_k = \mathcal{G}_1 \times_3^1 \cdots \times_3^1 \mathcal{G}_d$.
  $\mathcal{A}_{k+1} = \mathcal{M}_\Omega + (\hat{\mathcal{A}}_k)_{\overline{\Omega}}$.
  $k = k + 1$.
end while

the specified error tolerance, enhancing both accuracy and efficiency.
The low rank approximation error is assumed to be infinitely close to 0, that is, satisfied $\lim_{\delta \to 0} \mathbb{E}\|\mathcal{A}_k - \hat{\mathcal{A}}_k\|_F^2 = 0$. Then the sequence $\{\mathcal{A}_k\}$ generated by Algorithm 4 is convergent. The proof process is the similar as Theorem 2.

5. Numerical Experiments

In this section, a diverse set of experiments has been carried out involving synthetic data, color images, and videos. The performance of the proposed algorithm is compared with that of TMac-TT[3] and TMac-Square[15]. We set the power iteration parameter $q = 1$ in the Algorithm 2 and 4. When Algorithm 2 has no power iteration, that is, $q = 0$, we call it TTrand. The quality of the reconstructed tensor is measured by the peak signal-to-noise ratio (PSNR). For tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and its completed tensor $\hat{\mathcal{A}}$, the PSNR is defined as

$$PSNR = 10 \cdot log_{10} \frac{n_1 n_2 n_3 \|\hat{\mathcal{A}}\|_\infty^2}{\|\mathcal{A} - \hat{\mathcal{A}}\|_F^2}.$$

The PSNR of color video (four-dimensional tensor) is defined as the average value of PSNR of each frame image, i.e.

$$PSNR_v = \frac{1}{F} \sum_{k=1}^{F} 10 \log_{10} \frac{n_1 n_2 n_3 \|\hat{\mathcal{A}}(:,:,:,k)\|_\infty^2}{\|\mathcal{A}(:,:,:,k) - \hat{\mathcal{A}}(:,:,:,k)\|_F^2}.$$

The relative error between the completion result and the original tensor is defined as

$$Relative \quad error = \|\mathcal{A}_k - \mathcal{M}\|_F / \|\mathcal{M}\|_F$$

The mean structural similarity index(SSIM)[18] is defined as

$$\text{SSIM}(x, y) = \frac{(2\mu_x \mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}.$$

where μ_x represents the mean of x, σ_x represents the standard deviation of x, and σ_{xy} represents the covariance between x and y. We set $C_1 = (k_1 L)^2$, where $k_1 = 0.01$, and $C_2 = (k_2 L)^2$, where$k_2 = 0.03$, with $L = 255$.
All experiments were run on a laptop with 2.4 GHz Intel Core i7-8700T CPU and 16GB of RAM. We utilized the MATLAB Tensor Toolbox [2] to perform the experiments.

5.1. TTrandPI

In this section, we first demonstrate the convergence of the proposed algorithm through experiments. The elements of the random tensors with dimensions $10 \times 10 \times 10 \times 10 \times 10 \times 10$, $20 \times 20 \times 20 \times 20 \times 20$, $40 \times 40 \times 40 \times 40$ follow a standard Gaussian distribution $\mathcal{N}(0,1)$. The relative error of the tensor $\mathcal{A}$ between two successive iterations: $res = \|\mathcal{A}_k - \mathcal{A}_{k-1}\|_F / \|\mathcal{A}_{k-1}\|_F$. Where $\mathcal{A}_k$ and $\mathcal{A}_{k-1}$ denote the k^{th} and $k-1^{th}$ iterations of $\mathcal{A}$. We set missing rate to 0.9. The curves of relative error between two successive iterations are shown in Figure 1. The

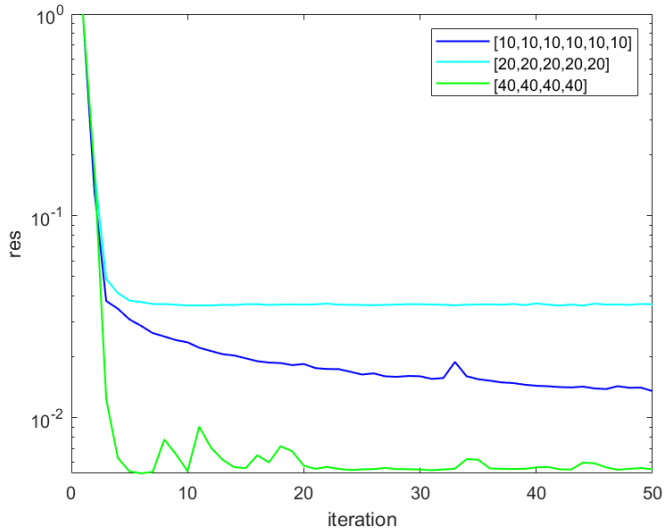

Figure 1. res vs. number of iterations for different dimension random tensor with Algorithm 2.

findings suggest that after several iterations, the relative error of Algorithm 2 diminishes and eventually levels off, illustrating its convergence.

Then, let us compare the recovery performance of TMacTT, TMacTT-Square and TTrandPI on color images. Bengua et al.[3] introduced ket augmentation (KA), which represents a low-order tensor using a higher-order one. This method can better utilize local information, thereby saving computational resources and making it more efficient for TT decomposition. The details of KA augmentation are shown in the Figure 2.

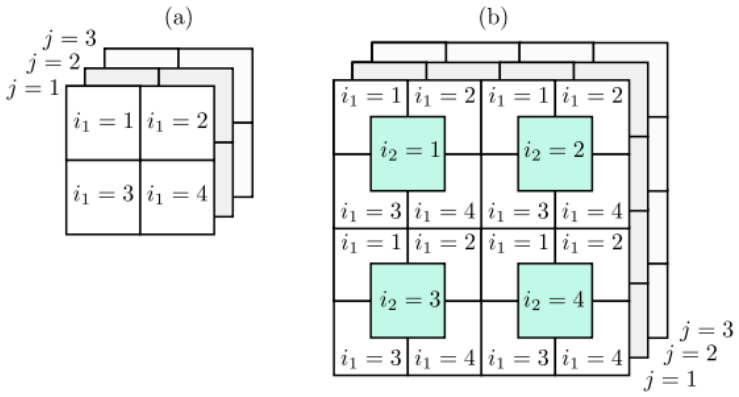

Figure 2. A structured block addressing procedure is employed to transform an image into a higher - order tensor. (a) Example of a $2 \times 2 \times 3$ image. (b) Illustration for an image of size $2^2 \times 2^2 \times 3$.

We take a color image of size $256 \times 256 \times 3$ as the original data. By applying KA to the tensor, it is transformed into a ninth-order tensor of size $4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 3$, and further perform tensor completion on it. The relative error between the recovered image and the original image as well as the PSNR of the completed image is compared when the missing rate varies from 0.1 to 0.9. The initial TT-rank is set to half the smallest size of the unfolded matrix, and re-select the rank after one iteration of completion.

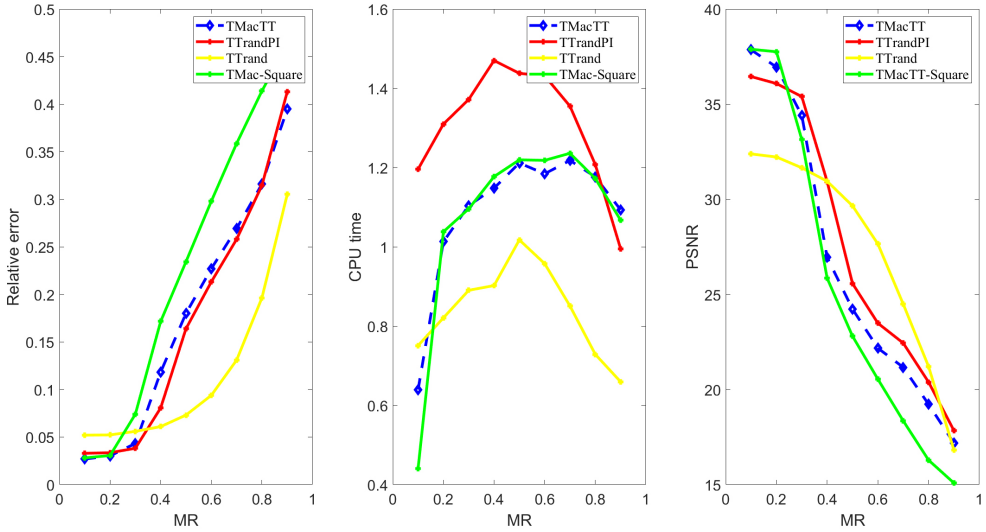

Figure 3. Numerical results for color image pepper.

Table 2. Numerical results of color image restoration by different methods at different sampling rates.

Color image	MR	0.8			0.9			0.95		
	Method	PSNR	Relative error	Time	PSNR	Relative error	Time	PSNR	Relative error	Time
Pepper	TMac-TT	19.72	0.3239	1.2947	16.97	0.4055	1.1139	15.19	0.4749	0.9601
	TTrandPI	21.36	0.4173	1.3140	**17.24**	0.4213	1.0018	**15.45**	0.5478	0.7298
	TMac-Square	16.87	0.3210	1.2656	15.72	0.4701	1.0537	14.21	0.5004	0.9640
	TTrand	**21.77**	**0.1966**	0.8517	16.77	**0.2893**	0.6623	13.32	**0.4476**	0.5722
Baboon	TMac-TT	19.62	0.2666	1.1751	17.34	0.3407	1.0267	**16.05**	0.4073	0.9761
	TTrandPI	**19.81**	0.2675	1.2747	**18.01**	0.3680	0.8927	15.21	0.4509	0.7764
	TMac-Square	16.44	0.3595	1.2140	15.38	0.4044	1.0407	14.51	0.4336	0.9802
	TTrand	19.09	**0.2091**	0.8589	15.45	**0.3102**	0.5994	14.10	**0.3740**	0.5381
Barbara	TMac-TT	20.64	0.2604	1.2603	17.84	0.3522	1.0628	15.62	0.4184	0.9910
	TTrandPI	**21.46**	0.2570	1.2591	**18.13**	0.3826	0.9379	**15.85**	0.4598	0.7337
	TMac-Square	17.48	0.3706	1.2498	14.73	0.4209	1.0271	14.21	0.4483	1.0111
	TTrand	21.25	**0.1934**	0.8654	16.69	**0.3073**	0.6515	14.82	**0.3771**	0.5427
Burano	TMac-TT	18.06	0.3446	1.2834	15.91	0.4287	1.1672	14.12	0.4912	1.0141
	TTrandPI	18.84	0.3543	1.3008	**16.42**	0.4498	1.0522	**14.37**	0.5452	0.8224
	TMac-Square	15.39	0.4326	1.2347	14.41	0.4838	1.1114	13.55	0.5109	1.0219
	TTrand	**19.89**	**0.2350**	0.8975	16.36	**0.3333**	0.7501	13.39	**0.4684**	0.6056
Sailboat	TMac-TT	17.13	0.3842	1.2166	15.58	0.4482	1.0739	14.24	0.5070	1.0895
	TTrandPI	18.01	0.3811	1.2879	15.99	0.4708	0.9895	**14.58**	0.5398	0.8691
	TMac-Square	14.29	0.4645	1.2208	13.60	0.5107	1.0964	12.85	0.5350	1.0131
	TTrand	**20.85**	**0.1833**	0.9234	**16.54**	**0.2890**	0.7595	14.06	**0.4065**	0.5844

In Figure 3, the first, second, and third graphs from left to right illustrate the relative error between the reconstructed and original images at various missing rates, the CPU time, and the PSNR between the reconstructed

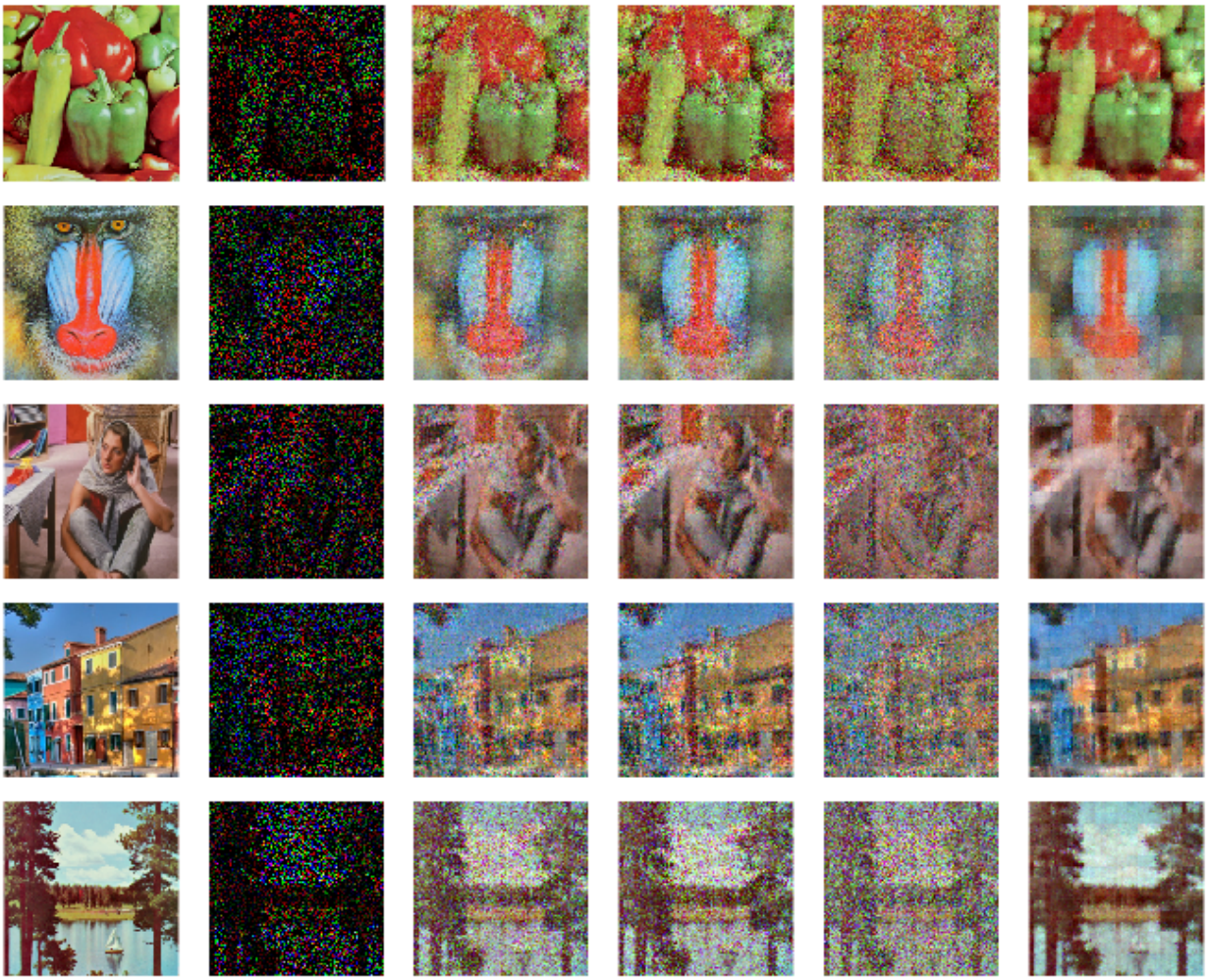

Figure 4. Low-rank completion for color image by different methods with MR=0.8. From left to right: the original image, observed image, completed by TMac-TT, TTrandPI, TMac-Square and TTrand. From up to down: pepper, baoon, barbara, burano and sailboat.

and original images, respectively. Figure 4 presents the completion effect diagrams for a missing rate of 0.8, and Table 2 provides their PSNR, RES, and CPU time for three algorithms at missing rates of 0.8, 0.9, and 0.95. As shown in Figures 3 and 4, the TTrand algorithm is faster and achieves the lowest relative error for $MR \geq 0.4$. The TTrandPI algorithm surpasses the others in PSNR values when the missing rate is high ($MR \geq 0.9$), and it consistently has a lower relative error, suggesting a superior completion effect.

Table 3. Numerical results of color image restoration under different structure damage

Color image	pepper			barbara		
Method	PSNR	Relative error	Time	PSNR	Relative error	Time
TMac-TT	25.73	0.1342	1.0614	23.04	0.1711	0.9547
TTrandPI	**27.62**	**0.0985**	1.2143	**25.84**	0.1266	1.0771
TMac-Square	23.59	0.1718	0.9856	21.97	0.2123	0.9281
TTrand	26.69	0.0970	0.8133	25.40	**0.1144**	0.8180

Figure 5. Structural missing color image restoration results. From left to right: the original image, observed image, completed by TMac-TT, TTrandPI, TMac-Square and TTrand. From up to down: pepper and barbara.

Furthermore, we performed tests on various deletion patterns, with findings presented in Figure 5 and Table 3. The TTrand algorithm is efficient with time and produces satisfactory completion results. TTrandPI achieves optimal performance with a higher PSNR compared to other algorithms and offers notably improved visual quality.

5.2. FPTT

In this section, similarly, we demonstrate the convergence of the proposed FPTT algorithm for tensor completion on random tensors with i.i.d standard Gaussian entries. We set the missing rate to 0.9. The curves of relative error between two successive iterations are shown in Figure 6.

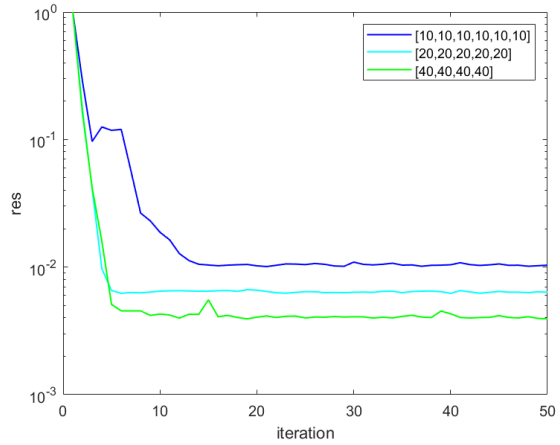

Figure 6. res vs. number of iterations for different dimension random tensor with Algorithm 4.

The results indicate that the relative error of Algorithm 4 remains essentially stable after multiple iterations, demonstrating its convergence.

As for color video completion, we measure TMacTT, TMacTT-Square and FPTT against data named $book$[†] ,$bird$[‡] and $forest$[§]. Resize the video to a tensor of size $480 \times 640 \times 50 \times 3(image\ \ row \times image$

[†]https://pixabay.com/videos/book-pages-literature-beach-ocean-185096/
[‡]https://pixabay.com/videos/robin-bird-forest-nature-spring-21723/
[§]https://pixabay.com/videos/background-clouds-forest-9584/

$colum \times frame \times RGB$). The $image\ row$ mode is merged with the $image\ column$ mode to form a third-order tensor. Therefore, by integrating the information from the frequency domain, we have performed completion for the entire video, directly applying the tensor completion algorithm to the 3rd-order tensor.

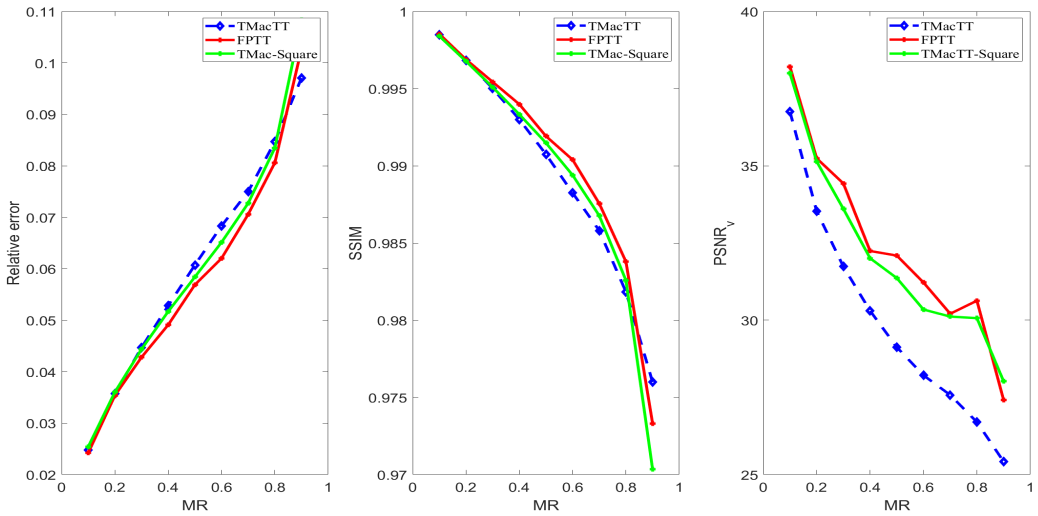

Figure 7. Numerical results for color video book.

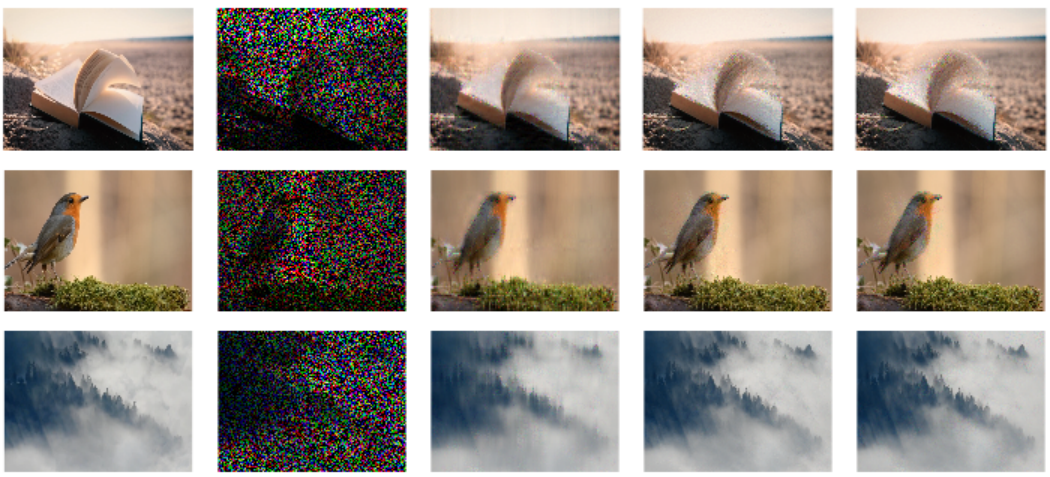

Figure 8. Low-rank completion for the 20-th frame of color video by different methods with MR=0.7. From left to right: the original video, observed video, completed by TMac-TT, FPTT and TMac-Square. From up to down: book ,bird and forest.

Table 4. Numerical results of color videos restoration by different methods at different sampling rates.

Color video	MR	0.7			0.9		
	Method	PSNR$_v$	SSIM	Time	PSNR$_v$	SSIM	Time
book	TMac-TT	27.55	0.9858	55.44	25.31	**0.9763**	55.03
	FPTT	**31.10**	**0.9881**	43.96	**28.03**	0.9731	48.91
	TMac-Square	30.20	0.9867	54.84	27.93	0.9712	54.82
bird	TMac-TT	28.12	0.9736	67.48	26.66	0.9583	59.91
	FPTT	**35.42**	**0.9775**	46.81	**32.01**	**0.9584**	48.47
	TMac-Square	34.72	0.9763	55.07	30.47	0.9509	60.26
forest	TMac-TT	31.15	0.9926	38.80	29.56	0.9894	59.94
	FPTT	**36.84**	**0.9979**	47.89	**33.35**	**0.9949**	43.56
	TMac-Square	32.81	0.9950	28.90	31.30	0.9928	60.94

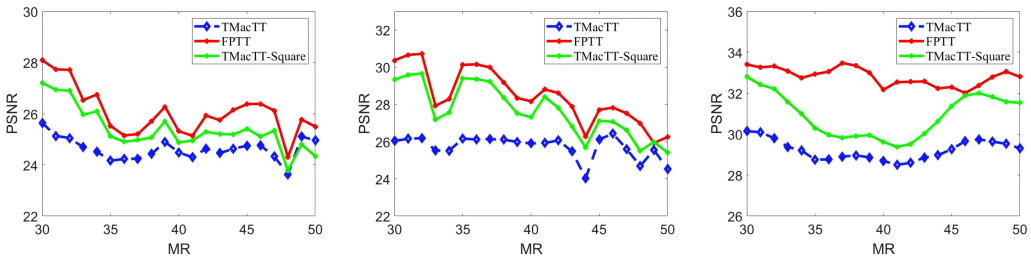

Figure 9. PSNR curve at frames 30-40 of color video with a miss rate of 0.9. From left to right: book, bird and forest

In Figure 7, the three graphs, arranged from left to right, illustrate the relative error, SSIM, and PSNR between the filled-in image and the original one across various missing rates. Figure 8 depicts the effect of image completion when the missing rate is 0.7, while Table 4 presents their PSNR, SSIM, and CPU time for three algorithms at missing rates of 0.7 and 0.9. Figure 9 displays the PSNR for frames 30 to 40 of a color video. As shown in Figures 7, 8, and 9, FPTT surpasses other algorithms in PSNR and generally demonstrates lower CPU time and SSIM, suggesting enhanced completion performance.

6. Conclusion

This study introduces the TTrandPI algorithm, designed for tensor completion utilizing TT decomposition alongside random power iteration. Experimental results indicate that it converges effectively and surpasses some existing algorithms in terms of completion performance, particularly at high rates of missing data. Additionally, to tackle the challenge of rank selection in tensor completion, we integrate a fixed precision matrix approximation technique, presenting the fixed precision TT tensor completion algorithm (FPTT). This approach alleviates the complexity of choosing an initial rank. Experiments demonstrate that FPTT offers superior computational efficiency and improved completion results.

REFERENCES

1. Salman Ahmadi-Asl, Andrzej Cichocki, Anh Huy Phan, Maame G Asante-Mensah, Mirfarid Musavian Ghazani, Toshihisa Tanaka, and Ivan Oseledets. Randomized algorithms for fast computation of low rank tensor ring model. *Machine Learning: Science and Technology*, 2(1):011001, 2020.
2. Brett W Bader and Tamara G Kolda. Tensor toolbox for matlab. *www.tensortoolbox.org*, 2023.
3. Johann A Bengua, Ho N Phien, Hoang Duong Tuan, and Minh N Do. Efficient tensor completion for color image and video recovery: Low-rank tensor train. *IEEE Transactions on Image Processing*, 26(5):2466–2479, 2017.
4. Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization*, 20(4):1956–1982, 2010.

5. Maolin Che, Yimin Wei, and Hong Yan. Randomized algorithms for computing the tensor train approximation and their applications. *arXiv preprint arXiv:2405.07147*, 2024.
6. Zhongming Chen, Huilin Jiang, Gaohang Yu, and Liqun Qi. Low-rank tensor train decomposition using tensorsketch. *arXiv preprint arXiv:2309.08093*, 2023.
7. Pierre Comon. Canonical tensor decompositions. In *Workshop on Tensor Decompositions, Palo Alto, CA*, 2004.
8. Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.
9. Xu Feng and Wenjian Yu. A fast adaptive randomized pca algorithm. In *IJCAI*, pages 3695–3704, 2023.
10. Nicholas J Higham. Matrix nearness problems and applications. *Applications of matrix theory*, 22, 1989.
11. Benjamin Huber, Reinhold Schneider, and Sebastian Wolf. A randomized tensor train singular value decomposition. In *Compressed Sensing and its Applications: Second International MATHEON Conference 2015*, pages 261–290. Springer, 2017.
12. Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.
13. Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):208–220, 2012.
14. Per-Gunnar Martinsson and Sergey Voronin. A randomized blocked algorithm for efficiently computing rank-revealing factorizations of matrices. *SIAM Journal on Scientific Computing*, 38(5):S485–S507, 2016.
15. Cun Mu, Bo Huang, John Wright, and Donald Goldfarb. Square deal: Lower bounds and improved relaxations for tensor recovery. In *International conference on machine learning*, pages 73–81. PMLR, 2014.
16. Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.
17. Wenjin Qin, Hailin Wang, Feng Zhang, Weijun Ma, Jianjun Wang, and Tingwen Huang. Nonconvex robust high-order tensor completion using randomized low-rank approximation. *IEEE Transactions on Image Processing*, 2024.
18. Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
19. Yangyang Xu, Ruru Hao, Wotao Yin, and Zhixun Su. Parallel matrix factorization for low-rank tensor completion. *Inverse Problems and Imaging*, 9(2):601–624, 2015.
20. Gaohang Yu, Chaoping Chen, Shaochun Wan, Liqun Qi, and Yanwei Xu. Multi-mode tensor train factorization with spatial-spectral regularization for third-order tensor completion. *Applied Mathematical Modelling*, 141:115921, 2025.
21. Gaohang Yu, Jinhong Feng, Zhongming Chen, Xiaohao Cai, and Liqun Qi. A randomized block krylov method for tensor train approximation. *arXiv preprint arXiv:2308.01480*, 2023.
22. Wenjian Yu, Yu Gu, and Yaohang Li. Efficient randomized algorithms for the fixed-precision low-rank matrix approximation. *SIAM Journal on Matrix Analysis and Applications*, 39(3):1339–1359, 2018.
23. Jiani Zhang, Arvind K Saibaba, Misha E Kilmer, and Shuchin Aeron. A randomized tensor singular value decomposition based on the t-product. *Numerical Linear Algebra with Applications*, 25(5):e2179, 2018.
24. Lefei Zhang, Liangchen Song, Bo Du, and Yipeng Zhang. Nonlocal low-rank tensor completion for visual data. *IEEE transactions on cybernetics*, 51(2):673–685, 2019.
25. Lei Zhang, Wei Wei, Qinfeng Shi, Chunhua Shen, Anton Van Den Hengel, and Yanning Zhang. Accurate tensor completion via adaptive low-rank representation. *IEEE Transactions on Neural Networks and Learning Systems*, 31(10):4170–4184, 2019.
26. Xi-Le Zhao, Wen-Hao Xu, Tai-Xiang Jiang, Yao Wang, and Michael K Ng. Deep plug-and-play prior for low-rank tensor completion. *Neurocomputing*, 400:137–149, 2020.