Signal, Image and Video Processing (2024) 18:6287–6297
https://doi.org/10.1007/s11760-024-03315-w

ORIGINAL PAPER

Towards efficient and accurate approximation: tensor decomposition based on randomized block Krylov iteration

Yichun Qiu[1] · Weijun Sun[1,2] · Guoxu Zhou[1,3] · Qibin Zhao[1,4,5]

Received: 3 March 2024 / Revised: 24 April 2024 / Accepted: 25 May 2024 / Published online: 15 June 2024

Abstract

Tensor decomposition methods are inefficient when dealing with low-rank approximation of large-scale data. Randomized tensor decomposition has emerged to meet this need, but most existing methods exhibit high computational costs in handling large-scale tensors and poor approximation accuracy in noisy data scenarios. In this work, a Tucker decomposition method based on randomized block Krylov iteration (rBKI-TK) is proposed to reduce computational complexity and guarantee approximation accuracy by employing cumulative sketches rather than randomized initialization to construct a better projection space with fewer iterations. In addition, a hierarchical tensor ring decomposition based on rBKI-TK is proposed to enhance the scalability of the rBKI-TK method. Numerical results demonstrate the efficiency and accuracy of the proposed methods in large-scale and noisy data processing.

Keywords Tensor decomposition · Low-rank approximation · Randomized block Krylov iteration · Noise reduction · Large-scale data processing

1 Introduction

Tensor data are ubiquitous in scientific computing and numerical applications, such as color images, video sequences, and EEG signals [1, 2]. Compared to matrix factorization, tensor decomposition provides a more natural and efficient way to represent, store and compute such data [3–7]. Since observations are often noisy, incomplete, and redundant, low-rank tensor decomposition plays an increasingly important role in many machine learning tasks, such as network compression [8], image imputation [9] and data mining [10]. However, when dealing with large-scale problems, tensor decomposition solved by deterministic methods such as singular value decomposition (SVD) [11] and alternating least squares (ALS) [12] can be computationally expensive and resource-intensive.

Randomized sketching techniques offer the possibility of eliminating this limitation. The basic idea of randomized sketching is to capture a representative subset or subspace of the data in order to efficiently produce a satisfactory approximation. Extensive works have reveal that randomized sketching techniques can effectively reduce computational complexity, thus improve efficiency [13–16]. Without loss of generality, random sketching techniques are divided into non-iterative and iterative sketching in this work. Non-iterative sketching involves sketching the input data in a single pass, while iterative sketching employs power iteration to perform the sketching process.

With the development of sketching techniques, a number of randomized tensor decomposition have emerged to

✉ Weijun Sun
gdutswj@gdut.edu.cn

Yichun Qiu
ycqiu@gdut.edu.cn

Guoxu Zhou
gx.zhou@gdut.edu.cn

Qibin Zhao
qibin.zhao@riken.jp

[1] School of Automation, Guangdong University of Technology, Guangzhou 510006, China

[2] Guangdong-Hong Kong-Macao Joint Laboratory for Smart Discrete Manufacturing, Guangzhou 510006, China

[3] Key Laboratory of Intelligent Detection and The Internet of Things in Manufacturing, Ministry of Education, Guangzhou 510006, China

[4] Joint International Research Laboratory of Intelligent Information Processing and System Integration of IoT, Ministry of Education, Guangzhou 510006, China

[5] Center for Advanced Intelligence Project (AIP), RIKEN, Tokyo 103-0027, Japan

provide more computational efficiency and approximation accuracy. In the non-iterative sketching category, there are tensor decomposition methods based on Gaussian random projection [17–20], CountSketch sampling [21, 22], and leverage score sampling [23, 24]. In the iterative sketching category, there are tensor decomposition methods based on Gaussian random iterations [19, 25], power iteration [26, 27], and rank revealing projection [28, 29]. While existing methods offer effective solutions for handling large-scale data, many of them fail to capture critical information in the presence of noise, leading to reduced approximation accuracy.

The Krylov methods aim to search for more accurate approximations by constructing more efficient projection spaces. The traditional Krylov methods construct the projection space with random initial vectors [30, 31] or matrices [32]. However, when applying these methods directly to tensor decomposition, the iterative orthonormalization required to construct the projection space leads to high computational costs, limiting the applicability of these methods to large tensor, and simple initialization using randomized vectors or matrices may lead to poor convergence [33, 34]. Therefore, the randomized block Krylov iteration (rBKI) method [35] was proposed to improve on the existing Krylov methods by using cumulative sketches instead of random initial data to construct projection spaces, thus obtaining more accurate approximation with fewer iterations.

The aim of this work is to develop more efficient and accurate randomized tensor decomposition methods for large-scale and noisy data. Although the rBKI method has been widely used in matrix analysis [35–37], it has not yet attracted attention in tensor community. The main innovation of this work is the first incorporation of rBKI into tensor decomposition, enabling more efficient and accurate approximation of large-scale and noisy tensors. Specifically, the rBKI method is introduced into the Tucker decomposition (rBKI-TK) for searching significant subspace of each mode, thus capturing the key information when optimizing the multi-linear factors. Besides, a hierarchical tensor ring model is developed to improve the scalability of rBKI-TK method. Yu et al. cited and discussed our methods in their work, which uses the rBKI method in tensor train model [38].

The contribution of our work can be summarised below.

- The rBKI-TK method offers more efficient processing of large-scale data and effectively reduce the noise interference.
- The hierarchical tensor ring decomposition based on rBKI-TK can further reduce the storage cost, which is important for applying large-scale data to various tasks.
- Numerical results demonstrate that the proposed methods exhibit significant efficiency and accuracy in processing large-scale and noisy data.

The following sections are organized as follows. In Sect. 2, some frequently used notations and definitions are outlined. In Sect. 3, main algorithms are presented. In Sect. 4, evaluations on both synthetic and real-world data are conducted. Finally, concluding remarks and future directions are depicted in Sect. 5.

2 Preliminaries

2.1 Notations

Throughout the paper, the terms "order" or "mode" are used to interchangeably describe a tensor data. For example, an Nth-order tensor can be unfolded into a mode-n unfolding matrix on each mode ($n = 1, 2, \dots, N$). Besides, a tensor set to [N = 3, I = 20, R = 3] represents a 3D tensor of dimension 20 and rank 3 on each mode. For better description, some frequently used notations are listed in Table 1.

2.2 Basic tensor decomposition models

Two basic tensor decomposition models studied in this work that are described below, and their corresponding diagrams are shown in Fig. 2.

Definition 1 Tucker decomposition of an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ is expressed as an Nth-order core tensor interacting with N factor matrices. Suppose the Tucker-rank of $\mathcal{X}$ is $(R_1, R_2, \dots, R_N)$, also known as multilinear-rank, then it has the following mathematical expression,

$$\begin{aligned}\mathcal{X} &= \mathcal{C} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_N \mathbf{U}^{(N)} \\ &\doteq [\mathcal{C}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \dots, \mathbf{U}^{(N)}],\end{aligned} \tag{1}$$

where $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ is the mode-n factor matrix, and $\mathcal{C} \in \mathbb{R}^{R_1 \times R_2 \cdots \times R_N}$ is the core tensor reflecting the connections between the factor matrices.

Definition 2 Tensor ring (TR) decomposition of an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ is expressed as a cyclic multilinear product of N third-order core tensors that can be shifted circularly. Given a TR-ranks of $(J_1, J_2, \dots, J_N)$, the TR operator in denoted by $\mathcal{X} \doteq \Re[\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_N]$, and the element-wise TR decomposition is expressed below.

$$\begin{aligned}&\mathcal{X}(i_1, i_2, \dots, i_N) = \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \cdots \sum_{j_N=1}^{J_N} \\ &\mathcal{G}_1^{(i_1)}(j_N, j_1)\mathcal{G}_2^{(i_2)}(j_1, j_2) \cdots \mathcal{G}_N^{(i_N)}(j_{N-1}, j_N),\end{aligned} \tag{2}$$

Table 1 Notations

$\mathbb{R}$	Set of real numbers
$\mathbf{x}, \mathbf{X}, \mathcal{X}$	Vector, matrix, and tensor
$\mathbf{X}^\top$	The transpose of matrix $\mathbf{X}$
$\mathbf{X}_{(n)}$	The standard mode-n unfolding of tensor $\mathcal{X}$
$\mathcal{X}_{[n]}$	The TR mode-n unfolding of tensor $\mathcal{X}$
$\mathcal{X}(i, j, k)$	The (i, j, k)th-entry of tensor $\mathcal{X}$
$\otimes$	The Kronecker product
$\times_n$	The mode-n product
$\lVert\mathbf{X}\rVert_F = \sqrt{\mathbf{X}^\top\mathbf{X}}$	Frobenius norm of matrix $\mathbf{X}$

where $\mathcal{G}_n \in \mathbb{R}^{J_{n-1}\times I_n \times J_n}$ and $\mathcal{G}_n^{(i_n)} \in \mathbb{R}^{J_{n-1}\times J_n}$ denotes the n-th TR-core and the i_n-th slice of the n-th TR-core, respectively.

Note that, each TR-core $\mathcal{G}_n$ contains one dimension-mode (I_n in mode-2) and two rank-modes (R_{n-1} in mode-1 and R_n in mode-3). Besides, adjoin factors $\mathcal{G}_n$ and $\mathcal{G}_{n+1}$ have a shared rank to ensure the contraction operation between them.

2.3 Low-rank approximation for matrix

The Low-rank approximation problem with respect to the Frobenius norm, and the $(1+\epsilon)$ quasi-optimal approximation is described as follows.

Definition 3 Low-rank approximation of any matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ with specific rank$(\mathbf{X}) = r \leq \min(n, d)$, is to find a rank-$k$ ($k \leq r$) approximation $\mathbf{X}_k$ that minimizes the Frobenius norm as

$$\min_{\mathbf{X}_k} \quad \lVert\mathbf{X} - \mathbf{X}_k\rVert_F^2. \tag{3}$$

Generally, the optimal rank-k approximation $\mathbf{X}_k$ can be obtained from the truncated SVD of $\mathbf{X}$, which is known as the Eckart-Young-Mirsky theorem [39]. Since SVD is prohibitive for large-scale data, randomized SVD addresses this limitation by using a variety of sketching techniques aimed at obtaining satisfactory approximation accuracy while significantly reducing storage and computational costs [13–15].

For a matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, the low-rank approximation based on randomized SVD consists of three main steps: 1) draw a sketch matrix $\mathbf{Y} = \mathbf{X}\boldsymbol{\Omega} \in \mathbb{R}^{n\times s}$ by employing a random sketching matrix $\boldsymbol{\Omega} \in \mathbb{R}^{d\times s}$ with a sketch-size $k \leq s \leq \min(n, d)$; 2) compute the orthonormal basis $\mathbf{Q} \in \mathbb{R}^{n\times s}$ of $\mathbf{Y}$ to project the data onto the sketching space $\mathbf{Q}^\top\mathbf{X} \in \mathbb{R}^{s\times d}$; 3) compute the low-rank approximation: $\tilde{\mathbf{X}} = \mathbf{U}\mathbf{U}^\top\mathbf{X}$ via truncated SVD of $\mathbf{Z} = \mathbf{Q}^\top\mathbf{X}$, i.e., $[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \text{svd}(\mathbf{Z})$.

Definition 4 The $(1 + \epsilon)$ quasi-optimal approximation can be described as the Frobenius norm error bound for the randomized rank-k approximation holds with high probability that

$$\lVert\mathbf{X} - \tilde{\mathbf{X}}_k\rVert_F^2 \leq (1 + \epsilon)\lVert\mathbf{X} - \mathbf{X}_k\rVert_F^2. \tag{4}$$

As $\tilde{\mathbf{X}}_k$ denotes the $(1 + \epsilon)$ rank-k approximation in a s-dimension subspace spaned by $\mathbf{Q} \in \mathbb{R}^{n\times s}$, $s \geq k$, sketching can further speed up the low-rank approximation when $s > k$. More theoretical analysis can be found in the work [40].

2.4 Low multilinear-rank approximation for tensor

Definition 5 The low multilinear-rank approximation [17] of an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\cdots\times I_N}$ with multilinear-rank $(R_1, R_2, \ldots, R_N)$ can be expressed as

$$\begin{aligned}\tilde{\mathcal{X}} &= \mathcal{C} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_N \mathbf{U}^{(N)}\\ &= \mathcal{X}\times_1(\mathbf{U}^{(1)}\mathbf{U}^{(1)\top})\times_2(\mathbf{U}^{(2)}\mathbf{U}^{(2)\top}) \cdots \times_N (\mathbf{U}^{(N)}\mathbf{U}^{(N)\top}),\end{aligned} \tag{5}$$

where $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n\times R_n}$ is the left singular matrix of the mode-n unfolding $\mathbf{X}_{(n)}$.

Note that, Eq. (5) can be defined as a randomized low multilinear-rank approximation when $\mathbf{U}^{(n)}$ is obtained from a randomized SVD. More details can be found in work [17].

3 Methodology

3.1 Randomized block Krylov iteration (rBKI)

The Krylov methods have been extensively investigated in matrix approximation. Traditional Krylov methods tended to capture the key information of data by simply constructing a Krylov subspace in a vector-by-vector paradigm [30, 31], i.e.,

$$\mathbf{K} \doteq [\mathbf{v}, \mathbf{X}\mathbf{v}, \mathbf{X}^2\mathbf{v}, \ldots, \mathbf{X}^{q-1}\mathbf{v}], \tag{6}$$

where $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the input and $\mathbf{v} \in \mathbb{R}^n$ is a randomized initial vector. To improve the efficiency, block Krylov method was proposed to construct a Krylov subspace in a block-by-block paradigm [32], i.e.,

$$\mathbf{K} \doteq [\mathbf{V}, \mathbf{X}\mathbf{V}, \mathbf{X}^2\mathbf{V}, \ldots, \mathbf{X}^{q-1}\mathbf{V}], \tag{7}$$

where $\mathbf{V} \in \mathbb{R}^{n\times s}$ is a random initial matrix.

However, when extended to large-scale data, these Krylov methods with random initial vectors or matrices may result in poor convergence [37], and the repeated orthonormalization required for constructing subspace results in high computational costs, limiting the applicability of these methods to large-scale data [33, 34]. Accordingly, randomized block Krylov iteration (rBKI) [35] was proposed to construct a Krylov subspace using sketches $\mathbf{W} = \mathbf{X}\Omega$ in a cumulative way instead of random initial data to capture more information about the original data with fewer iterations, i.e.,

$$\mathbf{K} \doteq [\mathbf{W}, (\mathbf{X}\mathbf{X}^\top)\mathbf{W}, \ldots, (\mathbf{X}\mathbf{X}^\top)^q\mathbf{W}], \tag{8}$$

where $\mathbf{\Omega} \in \mathbb{R}^{d\times s}$ is a random initial matrix. Empirically, $q = 2$ is sufficient to achieve a satisfactory approximate accuracy [35].

Compared with the existing popular randomized sketching methods: non-iterative method (e.g., Gaussian random projection), and power iteration method ($\mathbf{K} \doteq (\mathbf{X}\mathbf{X}^\top)^q\mathbf{W}$), rBKI performs better in reducing the effect of tail singular values. Although the power iteration method construct the sketching space in an iterative way, $(\mathbf{X}\mathbf{X}^\top)^q\mathbf{W}$ focuses solely on computing the dominant eigenspace. In contrast, rBKI method accumulates important information of the original data when constructing $[\mathbf{W}, (\mathbf{X}\mathbf{X}^\top)\mathbf{W}, \ldots, (\mathbf{X}\mathbf{X}^\top)^q\mathbf{W}]$, minimizing the loss of information during the projection process. With a more accurate projection subspace, rBKI facilitates higher accuracy approximation.

For better understanding, given a tensor with setting of [N = 3, I = 10, R = 5], the clean data is drawn from $\mathcal{N}(0, 1)$ and the noisy data is corrupted by SNR = 0dB Gaussian random noise. The pseudocolor plot of singular value matrices of the clean data, noisy data, and projections based on three representative randomized SVD methods, namely Gaussian random projection (GR-SVD), Gaussian random power iteration (GRpi-SVD), and rBKI (rBKI-SVD), are presented in Fig. 1. As shown, rBKI-SVD performed better in eliminating tail singular values (i.e., noise interference) than the non-iterative GR-SVD and the GRpi-SVD method.

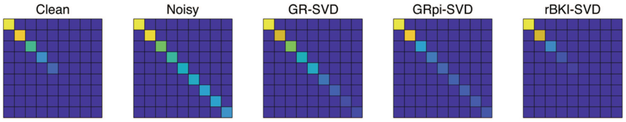

Fig. 1 Comparison of Gaussian random projection (GR-SVD), Gaussian random power iteration (GRpi-SVD) and rBKI (rBKI-SVD) methods in removing tail singular values of an [N = 3, I = 10, R = 5] tensor

Besides, since rBKI constructs the subspace through a lower q-degree polynomials (8), it requires only $q = \Theta(\log d/\sqrt{\epsilon})$ iterations to achieve the $(1+\epsilon)$ approximation to an $n \times d$ matrix, which is more efficient than the power iteration that requires $q = \Theta(\log d/\epsilon)$ iterations (Theorem 1 in [35]). Particularly, when the data have a rapid decay or a heavy-tail of singular value, rBKI can offer faster convergence performance, which is validated in the simulation section.

Moreover, the time complexity of several typical compared methods is presented in Table 2. Given a matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ with sketch-size s, according to the cost of each stage, the runtime of SVD is $\mathcal{O}(n^2d)$ while that of the GR-SVD is $\mathcal{O}(nds)$. With the number of iteration for rBKI of $q = \Theta(\log d/\sqrt{\epsilon})$ and for power iteration method of $q = \Theta(\log d/\epsilon)$, the corresponding runtime is $\mathcal{O}(nds \log d/\sqrt{\epsilon} + ns^2 \log^2 d/\epsilon + s^3 \log^3 d/\epsilon^{\frac{3}{2}})$ and $\mathcal{O}(nds \log d/\epsilon + ns^2 \log d/\epsilon)$, respectively.

3.2 Tucker decomposition based on rBKI (rBKI-TK)

Tucker decomposition can be regarded as a higher-order SVD (HOSVD) model [3], and randomized sketching techniques can be used to achieve efficient low multilinear-rank approximation of large-scale tensors. Besides, the challenges posed by low approximation accuracy in noisy data scenarios can be addressed by leveraging the rBKI method. Therefore, the rBKI-based Tucker decomposition (rBKI-TK) is proposed to search for crucial projection subspace with fewer iterations, thus improving the optimization of multilinear factors.

When computing the Tucker decomposition (Eq. 1), the standard mode-n unfolding on $\mathcal{X}$ as $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n\times\prod_{p\neq n} I_p}$ is

Table 2 Comparison of time complexity between the deterministic SVD and randomized SVD based on Gaussian random projection (GR-SVD), power iteration (GRpi-SVD) and rBKI (rBKI-SVD), on an $(n \times d)$ matrix with sketch-size s

Methods	SVD	GR-SVD	GRpi-SVD	rBKI-SVD
Sketch **W**	–	$\mathcal{O}(nds)$	$\mathcal{O}(nds)$	$\mathcal{O}(nds)$
Construct **K**	–	–	$\mathcal{O}(ns^2q + ndsq)$	$\mathcal{O}(ns^2q + ndsq)$
Compute **Q**	–	$\mathcal{O}(ns^2)$	$\mathcal{O}(ns^2)$	$\mathcal{O}(n(sq)^2)$
Compute **Z**	–	$\mathcal{O}(nds)$	$\mathcal{O}(nds + ns^2)$	$\mathcal{O}(nd(sq) + nd(sq)^2)$
Factorization	$\mathcal{O}(n^2d)$	$\mathcal{O}(s^2d)$	$\mathcal{O}(s^3)$	$\mathcal{O}((sq)^3)$

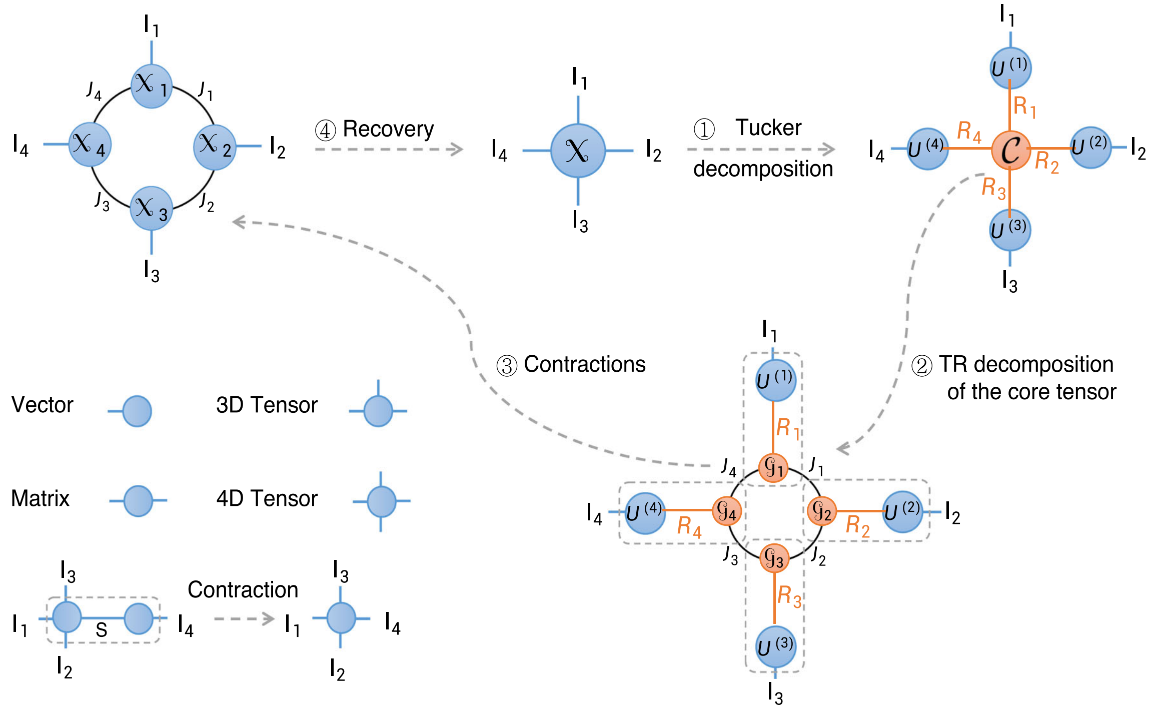

Fig. 2 A tensor network diagram of Tucker decomposition (TK), tensor ring decomposition (TR), and hierarchical TK-TR decomposition for a 4th-order tensor. The dimension size and rank are marked on the edges

necessary to be performed first and then optimize factors in a form of matrix product as follows.

$$\min_{\mathbf{U}^{(n)},\mathbf{G}_{(n)}} \left\| \mathbf{X}_{(n)} - \mathbf{U}^{(n)}\mathbf{C}_{(n)} \left(\bigotimes\nolimits_{p\neq n} \mathbf{U}^{(p)} \right)^{\top} \right\|_F^2, \tag{9}$$

where $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R_n}$, and $\mathbf{C}_{(n)} \left(\bigotimes_{p\neq n} \mathbf{U}^{(p)} \right)^{\top} \in \mathbb{R}^{R_n \times \prod_{p\neq n} I_p}$ can be computed by SVD [11] or ALS [12] algorithms.

Given an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ with sketch-size $(S_1, S_2, \ldots, S_N)$ and Tucker-rank $(R_1, R_2, \ldots, R_N)$, the rBKI-TK method provides a low multilinear-rank approximation of $\mathcal{X}$ through the following steps:

1) Draw a sketch $\mathbf{W}^{(n)} = \mathbf{X}_{(n)}\Omega^{(n)}$ to construct the mode-n block Krylov subspace

$$\mathbf{K}^{(n)} \doteq [\mathbf{W}^{(n)}, (\mathbf{X}_{(n)}\mathbf{X}_{(n)}^{\top})\mathbf{W}^{(n)}, \ldots, (\mathbf{X}_{(n)}\mathbf{X}_{(n)}^{\top})^{q}\mathbf{W}^{(n)}], \tag{10}$$

where $\Omega^{(n)} \in \mathbb{R}^{\prod_{p\neq n} I_p \times S_n}$ is a Gaussian random matrix.

2) Compute the orthonormal basis $\mathbf{Q}^{(n)}$ via QR of $\mathbf{K}^{(n)}$ to project data $\mathbf{X}_{(n)}$ onto the sketch subspace $\mathbf{Z}^{(n)} = \mathbf{Q}^{(n)\top}\mathbf{X}_{(n)}$.
3) Compute the factor matrices via $\mathbf{U}^{(n)} = \mathbf{Q}^{(n)}\tilde{\mathbf{U}}^{(n)}(:, 1 : R_n)$, where $\tilde{\mathbf{U}}^{(n)}$ is the left singular basis of the projection data $\mathbf{Z}^{(n)}$.
4) Compute the core tensor $\mathcal{C} \leftarrow \mathcal{X} \times_1 \mathbf{U}^{(1)\top} \times_2 \mathbf{U}^{(2)\top} \cdots \times_N \mathbf{U}^{(N)\top}$, and obtain the low multilinear-rank approximation by $\tilde{\mathcal{X}} \leftarrow \mathcal{C} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_N \mathbf{U}^{(N)}$.

The proposed rBKI-TK method offers the advantage of lower computational complexity and better denoising ability, which can be applied to large-scale and noisy data, thus has the potential for a wider range of applications. Details of the rBKI-TK method are presented in Algorithm 1.

Algorithm 1 The rBKI-based Tucker Decomposition (rBKI-TK)

```
Require: Tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$, sketch-size $(S_1, S_2, \ldots, S_N)$ and Tucker-rank $(R_1, R_2, \ldots, R_N)$, $q = 2$.
Ensure: $\tilde{\mathcal{X}} = [\mathcal{C}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]$.
for $n = 1, 2, \ldots, N$ do
    Draw a Gaussian random matirx $\Omega^{(n)} \in \mathbb{R}^{\prod_{p\neq n} I_p \times S_n}$;
    Draw a sketch $\mathbf{W}^{(n)} = \mathbf{X}_{(n)}\Omega^{(n)}$;
    Construct the block Krylov subspace $\mathbf{K}^{(n)}$ via Eq. (10);
    Compute the orthonormal basis $[\mathbf{Q}^{(n)}, \sim, \sim] = \mathrm{qr}(\mathbf{K}^{(n)})$;
    Project data onto the sketch subspace $\mathbf{Z}^{(n)} = \mathbf{Q}^{(n)\top}\mathbf{X}_{(n)}$;
    Compute the SVD of $\mathbf{Z}^{(n)}$: $[\tilde{\mathbf{U}}^{(n)}, \sim, \sim] = \mathrm{svd}(\mathbf{Z}^{(n)})$;
    Compute the $n$th factor matrix $\mathbf{U}^{(n)} = \mathbf{Q}^{(n)}\tilde{\mathbf{U}}^{(n)}(:, 1 : R_n)$;
end for
Compute core tensor $\mathcal{C} \leftarrow \mathcal{X} \times_1 \mathbf{U}^{(1)\top} \times_2 \mathbf{U}^{(2)\top} \ldots \times_N \mathbf{U}^{(N)\top}$;
Return Tucker factors $[\mathcal{C}, \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]$.
```

3.3 Extensions to hierarchical tensor decomposition

Tensor Ring (TR) decomposition is a powerful tool for higher-order (three or more) tensor analysis [6]. When computing the TR decomposition, Eq. (2) can be rewritten in a TR mode-n unfolding form as

$$\min_{\mathbf{G}_{n(2)}} \left\| \mathbf{X}_{[n]} - \mathbf{G}_{n(2)} (\mathbf{G}^{\neq n}_{[2]})^{\top} \right\|_F^2, \quad n = 1, 2, \ldots, N, \tag{11}$$

where $\mathbf{G}_{n(2)} \in \mathbb{R}^{I_n \times J_{n-1} J_n}$ and $\mathbf{G}^{\neq n}_{[2]} \in \mathbb{R}^{\prod_{p \neq n} I_p \times J_{n-1} J_n}$ denotes the mode-2 unfolding of the nth core and the subchain except the nth core, respectively. Note that, the difference between the TR mode-n unfolding $\mathbf{X}_{[n]} \in \mathbb{R}^{I_n \times I_{n+1} \cdots I_N I_1 \cdots I_{n-1}}$ and the standard mode-n unfolding $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 \cdots I_{n-1} I_{n+1} \cdots I_N}$ is the order of TR-cores in the subchain.

TR-ALS is a typical algorithm for updating TR-cores, which updates the nth TR-core by first computing the unfolding $\mathbf{G}_{n(2)}$ via Eq. (11), and then reshape $\mathbf{G}_{n(2)}$ into a third-order tensor $\mathcal{G}_n \in \mathbb{R}^{J_{n-1} \times I_n \times J_n}$, for $n = 1, 2, \ldots, N$, and repeated until the relative error $\delta = \left\| \mathbf{X}_{[n]} - \mathbf{G}_{n(2)} (\mathbf{G}^{\neq n}_{[2]})^{\top} \right\|_F / \left\| \mathbf{X}_{[n]} \right\|_F$ reaches the set tolerance δ_p or reaches the maximum number of iteration p [6].

However, the direct application of TR decomposition to large-scale data processing often results in inefficiency. Therefore, hierarchical tensor decomposition methods with randomized algorithm were extensively studied for reducing the computation and storage cost [17, 19, 20, 29]. In this subsection, a hierarchical tensor decomposition based on rBKI method is developed to improve the scalability of the rBKI-TK method and the efficiency of TR method. In the rBKI-TK-TR model, the rBKI-TK can efficiently obtain factors and reduce noise interference, and the TR decomposition can further reduce the data storage cost when the TR-ranks are much smaller than the Tucker-rank.

Generally, such hierarchical framework can be extended to other tensor decomposition methods. The proposed rBKI-TK-TR method is detailed in Algorithm 2, and a tensor network diagram of TK, TD, hierarchical TK-TR decomposition and contraction operations among them are shown in Fig. 2.

4 Experiments

This section evaluates the efficiency and accuracy of the proposed method for large-scale processing and noise reduction. All evaluations were conducted on a computer with an Intel Core i5 3 GHz CPU and 8GB of RAM. The algorithms were coded in MATLAB 2020r.

Algorithm 2 The rBKI-TK-based Tensor Ring Decomposition (rBKI-TK-TR)

```
Require: Tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$, sketch-size $(S_1, S_2, \ldots, S_N)$, Tucker-rank $(R_1, R_2, \ldots, R_N)$, and TR-ranks $(J_1, J_2, \ldots, J_N)$, tolerance $\delta_p$, max-iteration $p$.
Ensure: $\tilde{\mathcal{X}} = \mathfrak{R}[\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_N]$.
while $\delta > \delta_p$, or the max-iteration $p$ is not reached do
    for $n = 1, 2, \ldots, N$ do
        Compute Tucker factors $[\mathcal{C}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]$ via rBKI-TK($\mathcal{X}$);
        Compute unfolding TR-core $\mathbf{G}_{n(2)} \leftarrow \arg\min \left\| \mathbf{C}_{[n]} - \mathbf{G}_{n(2)} (\mathbf{G}^{\neq n}_{[2]})^{\top} \right\|_F^2$;
        Normalize colums of $\mathbf{G}_{n(2)}$, if $n < N$;
        Reshape TR-core $\mathcal{G}_n \leftarrow$ permute(reshape($\mathbf{G}_{n(2)}$, $[R_n, J_{n-1}, J_n]$), [2, 1, 3]);
        Contraction: $\mathcal{X}_n \leftarrow \mathcal{G}_n \times_2 \mathbf{U}^{(n)}$;
    end for
    Compute relative error $\delta \leftarrow ||\mathcal{X} - \mathfrak{R}[\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_N]||_F / ||\mathcal{X}||_F$;
end while
Return TR-cores $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_N$.
```

4.1 Evaluation metric

The following metrics were employed for evaluations. Note that all results were the average of 10 runs.

- Fitting rate, Fit $= (1 - \|\mathcal{X} - \tilde{\mathcal{X}}_k\|_F / \|\mathcal{X}\|_F) \times 100\%$, where $\mathcal{X}$ and $\tilde{\mathcal{X}}_k$ is the original tensor and the sketched rank-k approximation, respectively. A larger value indicates higher recovery accuracy.
- Frobenius norm error (Eq. 4), FErr $= \|\mathcal{X} - \tilde{\mathcal{X}}_k\|_F / \|\mathcal{X} - \mathcal{X}_k\|_F$, where $\mathcal{X}_k$ is the deterministic rank-k approximation obtained by truncated SVD (T-SVD). When FErr = 1, the randomized rank-k approximation $\tilde{\mathcal{X}}_k$ captures the main information as well as the deterministic one $\mathcal{X}_k$. The larger the ratio, the worse the results [35].
- Peak-Signal-Noise-Ratio, PSNR $= 10 \log_{10}(255^2/\text{MSE})$, where MSE is the Mean-Square-Error denoted by $\|\mathcal{X} - \tilde{\mathcal{X}}_k\|_F / \text{num}(\mathcal{X})$, and num$(\cdot)$ denotes the number of components.
- Running time in seconds. For the TR-ALS, the iterations were repeated until the maximum number of iterations 50 is reached, or the relative error $\delta < \delta_p = 10^{-3}$.

4.2 Datasets

4.2.1 Synthetic data

The generation of tensor data followed $\mathcal{X} + \lambda\mathcal{N}$, where the original data $\mathcal{X}$ was corrupted by a noise term $\lambda\mathcal{N}$, and λ was a parameter for tuning the noise level of $\mathcal{N}$ via Signal-to-Noise-Ratio, SNR(dB) $= 20\log_{10}(\|\mathcal{X}\|_F / \|\lambda\mathcal{N}\|_F)$. A relatively comprehensive test was provided that considers higher and lower rank tensors with Gaussian random noise.

Table 3 Comparison in terms of FErr, Fit(%) and Time(s) of variant methods on the [N = 3,I = 200,R_n = 10] tensor of power functional (PF) and Gaussian random (GR) at different Gaussian noise level.

Algorithm	PF-data, SNR = 5 dB			GR-data, SNR = −5 dB			GR-data, SNR = 5 dB		
	FErr	Fit	Time	FErr	Fit	Time	FErr	Fit	Time
GR-SVD [19]	12.73	62.91	5.42	17.69	2.03	4.17	46.75	22.75	8.89
CS-SVD [21]	12.64	63.20	5.55	17.68	2.11	5.87	47.07	22.21	11.09
LS-SVD [23]	13.98	59.28	21.41	17.68	2.07	14.48	47.80	21.00	33.74
IS-SVD [13]	10.78	68.60	3.86	17.59	2.56	4.81	43.71	27.77	8.69
SRHT-SVD [13]	12.78	62.78	105.05	17.68	2.08	188.01	46.43	23.27	447.71
GR3i-SVD [25]	20.58	40.05	28.03	15.74	12.83	39.12	21.39	64.65	33.97
GR2i-SVD [19]	2.19	93.62	3.11	8.97	50.31	11.74	16.18	73.27	10.65
GRrr-SVD [29]	2.53	92.63	9.14	5.84	67.67	23.84	1.04	98.27	5.67
GRpi-SVD [27]	2.12	93.83	1.97	2.72	84.94	12.54	**1.00**	98.34	1.57
rBKI-SVD [ours]	1.10	96.80	**0.85**	**1.00**	**94.46**	**1.68**	**1.00**	**98.35**	**1.47**
T-SVD [11]	**1.00**	**97.09**	2.48	**1.00**	**94.46**	4.30	**1.00**	**98.35**	3.74

Compared to the deterministic LRA method (T-SVD), the proposed method achieve the best results among the randomized LRA methods

Table 4 Comparison of the denoising performance in terms of PSNR(dB), FErr, and Time(s) of variant methods on three YUV video data with Gaussian noise (SNR = 5dB).

Algorithm	News			M&D			Hall		
	PSNR	FErr	Time	PSNR	FErr	Time	PSNR	FErr	Time
GR-SVD [19]	20.34	2.89	127.02	21.48	2.68	124.82	20.10	2.82	63.44
CS-SVD [21]	20.30	2.90	130.72	21.32	2.70	126.10	20.32	2.79	75.49
IS-SVD [13]	20.58	2.76	86.95	21.56	2.58	62.15	22.05	2.89	42.85
GRrr-SVD [29]	24.88	1.59	90.06	25.70	1.45	65.36	25.32	1.45	55.92
GRpi-SVD [27]	25.22	1.52	32.46	25.99	1.41	33.26	25.12	1.49	25.66
rBKI-SVD [ours]	28.69	1.01	**18.50**	28.99	1.01	**18.83**	28.40	1.01	**16.78**
T-SVD [11]	**28.73**	**1.00**	51.85	**29.07**	**1.00**	56.55	**28.50**	**1.00**	38.26

Compared to the deterministic T-SVD, the proposed method achieve the best results among the randomized methods

Bold values indicates the best results. The deterministic method T-SVD serves as the benchmark algorithm and our method achieves the best results among the other randomized algorithms (except for TSVD).

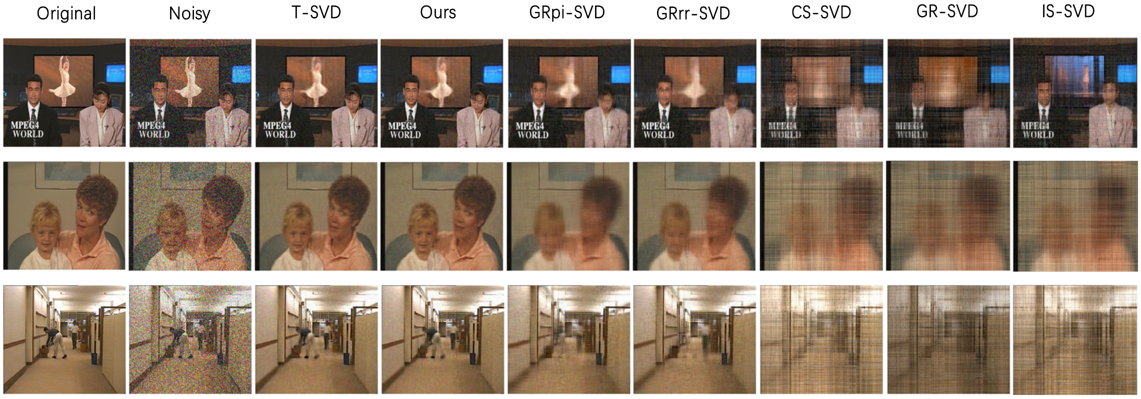

Fig. 3 Examples of video denoising, including original data, noisy data and recovered data. Our method achieves accurate approximations comparable to T-SVD, while in a more time-efficient fashion

Table 5 Comparison in terms of Fit(%) and Time (s) of variant hierarchical TR models on two large-scale image datasets. The TR-ranks is set ranged from R = [3,3,3,3], R = [6,6,3,6] to R = [9,9,3,9].

Algorithm	COIL-100					
	R = [3, 3, 3, 3]		R = [6, 6, 3, 6]		R = [9, 9, 3, 9]	
	Fit	Time	Fit	Time	Fit	Time
GR-TK-TR [20]	57.33	15.40	65.25	19.79	69.20	25.14
RR-TK-TR [29]	64.63	12.90	72.60	16.15	76.04	20.14
rBKI-TK-TR [ours]	64.68	**4.27**	72.69	**5.51**	76.25	**7.23**
TR-ALS [6]	**64.85**	111.69	**73.80**	277.18	**77.78**	474.53
Algorithm	CIFAR-10					
	R = [3, 3, 3, 3]		R = [6, 6, 3, 6]		R = [9, 9, 3, 9]	
	Fit	Time	Fit	Time	Fit	Time
GR-TK-TR [20]	64.94	23.62	71.03	29.28	73.63	30.51
RR-TK-TR [29]	71.78	19.78	76.77	24.21	78.99	24.88
rBKI-TK-TR [ours]	71.80	**7.11**	76.91	**8.96**	79.00	**9.38**
TR-ALS [6]	**71.91**	156.68	**77.46**	390.90	**80.00**	802.87

Compared to the deterministic TR-ALS, our method achieve the best results among the randomized methods, which are shown in bold. All results are the average of 10 runs

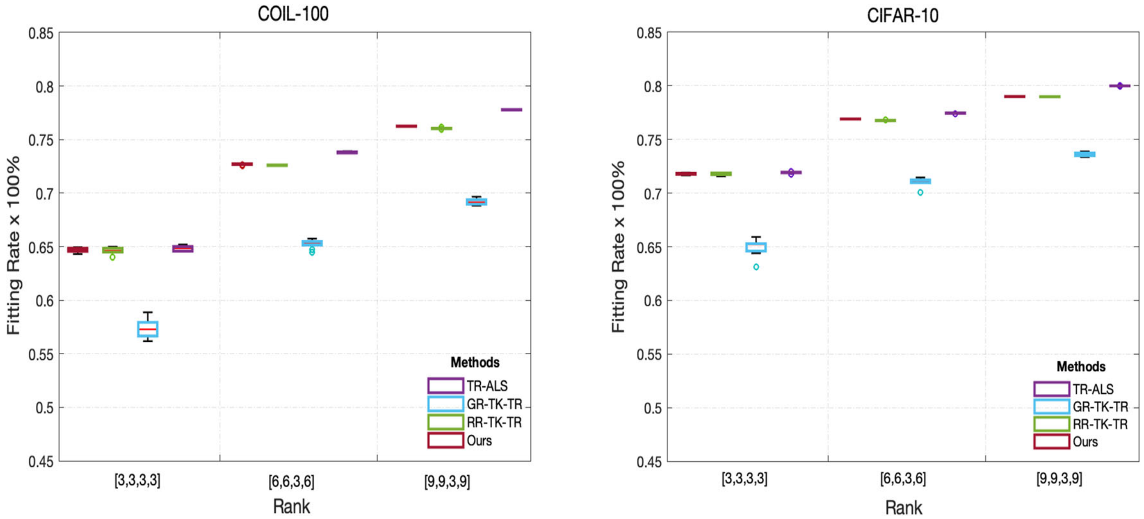

Fig. 4 Comparison in terms of Fit (×100%) of the compared methods on COIL-100 and CIFAR-10 datasets. The TR-ranks is set ranged from R = [3,3,3,3], R = [6,6,3,6] to R = [9,9,3,9]

Specifically, in the higher rank case, 3D tensors $\mathcal{X}$ with dimension of 200 and rank of 10 were given by Tucker model (1), where the entries of latent factor matrices $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R_n}$, $n = 1, 2, 3$ and core tensor $\mathcal{C} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ were drawn from $\mathcal{N}(0, 1)$. In the lower rank case, the singular values of the data decay rapidly. The 3D tensors $\mathcal{X}$ were given by a power function $x_{i,j,k} = 1/\sqrt[p]{i^p + j^p + k^p}$, where p is a power parameter controlling the singular values of $\mathcal{X}$ (here, p = 10). It is noted that, for the additional noise, SNR = 0dB indicates that the power of signal is equal to the volume of noise, while SNR>0 and SNR<0 represents a less noisy and a much noisier case, respectively.

4.2.2 Real-world datasets

Three YUV video sequences (“News”, “Hall”, “M&D”) [1] were utilized for denoising evaluations. The sequences were converted into fourth-order RGB tensors with a size of 288 pixels × 352 pixels × 3 RGB × 300 frames. The 123th-frame of “News” video, the 15th-frame of “M&D” video, and the 120th-frame of “Hall” video were randomly selected for demonstration. Moreover, two large-scale image datasets were used for comparison of efficiency: COIL-100 dataset of size 32 × 32 × 3 × 7200, and a subset of CIFAR-10 dataset of size 32 × 32 × 3 × 10000. Both datasets contain a number of elements at the 10^7 level.

[1] http://trace.eas.asu.edu/yuv/index.html.

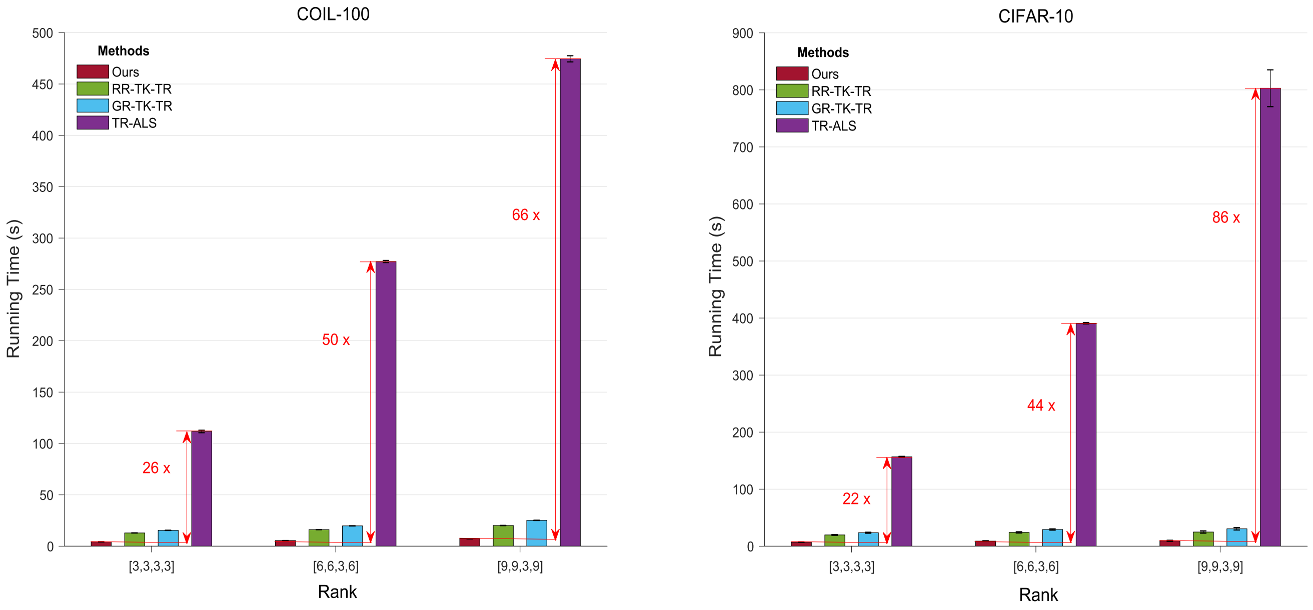

Fig. 5 Comparison in terms of Time consumption of the compared methods on COIL-100 and CIFAR-10 datasets. The TR-ranks is set ranged from R = [3,3,3,3], R = [6,6,3,6] to R = [9,9,3,9]

4.3 Evaluation of rBKI-TK

In this section, the proposed rBKI-TK is compared with the deterministic truncated higher-order SVD (T-SVD) [11] and existing randomized tensor decomposition methods based on different sketching techniques. The non-iterative series includes Gaussian random projection (GR-SVD) [19], CountSketch sampling (CS-SVD) [21], leverage score sampling [23], importance sampling (IS-SVD) [13] and subsampled randomized Hadamard transform [13]. The iterative series includes Gaussian random power iteration (GRpi-SVD) [27], Gaussian random iteration (GR3i-SVD, GR2i-SVD) [19, 25], and Gaussian random rank revealing (GRrr-SVD) [29]. Besides, the sketch size was set by $S_n = R_n + 1/\gamma$, where the overlapping parameter γ was set to 0.2, as suggested in [15].

The experimental results of synthetic data were detailed in Table 3, where the the best results were shown in bold. Specifically, iterative methods obtained higher accuracy than non-iterative methods, except GR3i-SVD [25] in the lower rank case (PF-data). With different noise setting, non-iterative methods and two Gaussian random iteration methods (GR3i-SVD, GR2i-SVD) were severely invalidated by noise interference, while our method remained valid as T-SVD in terms of FErr and Fit in most cases. These results show that the proposed method was superior in terms of accuracy and efficiency in both lower and higher rank data types, as well as in severe and slight noise interference.

Moreover, to evaluate the denoising performance of rBKI-TK in real-world case, Gaussian noise with an SNR of 5dB were added to video sequences. The approximate rank was set to $85 \times 90 \times 3 \times 30$, which was selected by T-SVD and yielded satisfactory results in terms of both accuracy and efficiency. The T-SVD, two competitive iterative methods: GRpi-SVD, GRrr-SVD, and three non-iterative methods: GR-SVD, CS-SVD, IS-SVD were adopted in comparisons. The average results were given in Table 4, and a visual comparison of denoising performance was shown in Fig. 3.

As shown, iterative methods also outperform non-iterative methods in terms of approximation accuracy in real-world data, which produced a clear foreground (e.g., the letters “MPEG4 WORLD” in the “News” video). Besides, the proposed rBKI-TK and T-SVD both provided more details of contents (e.g., moving objects in the “Hall” video) that were not captured by the other methods, which are important information.

4.4 Evaluation of rBKI-TK-TR

In the subsection, the proposed rBKI-TK-TR model were compared with the deterministic TR-ALS [6] and two existing hierarchical TR models, i.e., Gaussian random TK-TR model (GR-TK-TR) [20] and rank revealing TK-TR model (RR-TK-TR) [29] on two large-scale image datasets. For large-scale data, higher rank leads to huge computational costs, we empirically chose several sets of TR-ranks for comparison and evaluation. The TR-ranks J_n was set ranged from [3,3,3,3], [6,6,3,6] to [9,9,3,9], while the Tucker-rank was set to $R_n = \min(5J_n, I_n)$, and in order to minimize information loss without reducing efficiency, the Tucker-rank R_n and sketch-size S_n were set to the same value in the hierarchical TK-TR model.

The efficiency and recovery accuracy of comparison models were detailed in Table 5. Moreover, the visual comparison in terms of fitting rate and running time were provided in Figs. 4 and 5, respectively. Comparable to the deterministic TR-ALS, our method achieved the best recovery accuracy and time efficiency among randomized methods. In all cases, the Fit of non-iterative GR-TK-TR was lower than the other methods and slightly unstable, while iterative methods, i.e., RR-TK-TR and the proposed rBKI-TK-TR performed stably and better (see Fig. 4). Besides, the consumed time by the TR-ALS was extremely larger than the other randomized methods. Our method was the most time efficient, and as the rank increases the efficiency became more prominent, even 86 times faster than TR-ALS (see Fig. 5).

5 Conclusion

In this work, a randomized Tucker decomposition method is proposed by leveraging the strength of rBKI algorithm in reducing computational complexity and noise inferences, which achieves efficient and accurate low-rank approximation for large-scale and noisy tensors. Besides, a hierarchical rBKI-TK-TR method is developed for further reducing the storage cost. Experimental results promisingly validate the superior performance of the proposed methods in large-scale processing and noise reduction. Recently, adversarial attacks in visual data has attracted much attention. Therefore, randomized tensor decomposition and its application to machine learning for improving adversarial robustness will be explored in future work.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant 62073087 and Grant U1911401.

Author Contributions 1. Yichun Qiu (First author): Conceptualization, Investigation, Methodology, Software, Writing - Original Draft. 2. Weijun Sun (Corresponding author): Resources, Funding Acquisition, Writing - Review & Editing. 3. Guoxu Zhou: Supervision, Writing - Review & Editing. 4. Qibin Zhao: Supervision, Writing - Review & Editing.

Data availability Data is provided within the manuscript or supplementary information files.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

1. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. **51**, 455–500 (2009)
2. Cichocki, A., Mandic, D.P., Lathauwer, L.D., Zhou, G., Zhao, Q., Caiafa, C.F., Phan, A.: Tensor decompositions for signal processing applications: from two-way to multiway component analysis. IEEE Signal Process. Mag. **32**, 145–163 (2015)
3. Tucker, L.R.: Some mathematical notes on three-mode factor analysis. Psychometrika **31**, 279–311 (1966)
4. Harshman, R.A., et al.: Foundations of the parafac procedure: models and conditions for an "explanatory" multi-modal factor analysis. UCLA Working Papers Phonet. **16**(1), 84 (1970)
5. Oseledets, I.: Tensor-train decomposition. SIAM J. Sci. Comput. **33**, 2295–2317 (2011)
6. Zhao, Q., Zhou, G., Xie, S., Zhang, L., Cichocki, A.: Tensor ring decomposition. arXiv:1606.05535 (2016)
7. Zheng, Y., Huang, T., Zhao, X., Zhao, Q., Jiang, T.-X.: Fully-connected tensor network decomposition and its application to higher-order tensor completion. In: Proceedings of the AAAI conference on artificial intelligence, vol. 35, pp. 11071–11078 (2021)
8. Novikov, A., Podoprikhin, D., Osokin, A., Vetrov, D.P.: Tensorizing neural networks. In: Proceedings of the 28th international conference on neural information processing systems (2015)
9. Yu, J., Zhou, G., Li, C., Zhao, Q., Xie, S.: Low tensor-ring rank completion by parallel matrix factorization. IEEE Trans. Neural Netw. Learn. Syst. **32**, 3020–3033 (2021)
10. Qiu, Y., Sun, W., Zhang, Y., Gu, X., Zhou, G.: Approximately orthogonal nonnegative tucker decomposition for flexible multiway clustering. Sci. China Technol. Sci. **64**, 1872–1880 (2021)
11. Lathauwer, L.D., Moor, B.D., Vandewalle, J.: A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. **21**, 1253–1278 (2000)
12. Lathauwer, L.D., Moor, B.D., Vandewalle, J.: On the best rank-1 and rank-(r1 , r2, ... , rn) approximation of higher-order tensors. SIAM J. Matrix Anal. Appl. **21**, 1324–1342 (2000)
13. Wang, S.: A practical guide to randomized matrix computations with matlab implementations. CoRR arXiv:1505.07570 (2015)
14. Tropp, J.A., Yurtsever, A., Udell, M., Cevher, V.: Practical sketching algorithms for low-rank matrix approximation. SIAM J. Matrix Anal. Appl. **38**(4), 1454–1485 (2017)
15. Halko, N., Martinsson, P.G., Tropp, J.A.: Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev. **53**(2), 217–288 (2011)
16. Szlam, A., Tulloch, A., Tygert, M.: Accurate low-rank approximations via a few iterations of alternating least squares. SIAM J. Matrix Anal. Appl. **38**(2), 425–433 (2017)
17. Che, M., Wei, Y.-M.: Randomized algorithms for the approximations of tucker and the tensor train decompositions. Adv. Comput. Math. **45**, 395–428 (2019)
18. Feng, Y., Zhou, G.: Orthogonal random projection for tensor completion. IET Comput. Vis. **14**, 233–240 (2020)
19. Qiu, Y., Zhou, G., Zhang, Y., Cichocki, A.: Canonical polyadic decomposition (cpd) of big tensors with low multilinear rank. Multimed. Tools Appl. **80**(15), 22987–23007 (2020)
20. Yuan, L., Li, C., Cao, J., Zhao, Q.: Randomized tensor ring decomposition and its application to large-scale data reconstruction. In: IEEE international conference on acoustics, speech and signal processing (ICASSP), pp. 2127–2131 (2019)
21. Malik, O.A., Becker, S.: Low-rank tucker decomposition of large tensors using tensorsketch. In: Proceedings of the 32nd international conference on neural information processing systems. NIPS'18, pp. 10117–10127. Curran Associates Inc., Red Hook, NY, USA (2018)
22. Wang, Y., Tung, H.-Y., Smola, A., Anandkumar, A.: Fast and guaranteed tensor decomposition via sketching. In: Proceedings of the 28th international conference on neural information processing systems - Volume 1. NIPS'15, pp. 991–999. MIT Press, Cambridge, MA, USA (2015)
23. Larsen, B.W., Kolda, T.G.: Practical leverage-based sampling for low-rank tensor decomposition. arXiv:2006.16438 (2022)

24. Malik, O.A.: More efficient sampling for tensor decomposition with worst-case guarantees. In: International conference on machine learning, pp. 14887–14917 (2022). PMLR
25. Ma, L., Solomonik, E.: Fast and accurate randomized algorithms for low-rank tensor decompositions. Adv. Neural. Inf. Process. Syst. **34**, 24299–24312 (2021)
26. Erichson, N.B., Manohar, K., Brunton, S.L., Kutz, J.N.: Randomized cp tensor decomposition. Mach. Learn.: Sci. Technol. **1**(2), 025012 (2020)
27. Ahmadi-Asl, S., Abukhovich, S., Asante-Mensah, M.G., Cichocki, A., Phan, A., Tanaka, T., Oseledets, I.: Randomized algorithms for computation of tucker decomposition and higher order svd (hosvd). IEEE Access **9**, 28684–28706 (2021)
28. Minster, R., Saibaba, A.K., Kilmer, M.E.: Randomized algorithms for low-rank tensor decompositions in the tucker format. arXiv:1905.07311 (2020)
29. Ahmadi-Asl, S., Cichocki, A., Phan, A.H., Asante-Mensah, M.G., Ghazani, M.M., Tanaka, T., Oseledets, I.: Randomized algorithms for fast computation of low rank tensor ring model. Mach. Learn.: Sci. Technol. **2**(1), 011001 (2020)
30. Lanczos, C.: An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. J. Res. Natl. Bur. Stand. **45**, 255–282 (1950)
31. Golub, G., Kahan, W.: Calculating the singular values and pseudo-inverse of a matrix. J. Soc. Indus. Appl. Math. Series B Numer. Anal. **2**(2), 205–224 (1965)
32. Cullum, J., Donath, W.E.: A block lanczos algorithm for computing the q algebraically largest eigenvalues and a corresponding eigenspace of large, sparse, real symmetric matrices. In: 1974 IEEE conference on decision and control including the 13th symposium on adaptive processes, pp. 505–509 (1974). IEEE
33. Savas, B., Eldén, L.: Krylov-type methods for tensor computations i. Linear Algebra Appl. **438**(2), 891–918 (2013)
34. Eldén, L., Dehghan, M.: A krylov-schur-like method for computing the best rank-(r1, r2, r3) approximation of large and sparse tensors. Numer. Algorithms **91**, 1315–1347 (2022)
35. Musco, C., Musco, C.: Randomized block krylov methods for stronger and faster approximate singular value decomposition. In: Proceedings of the 28th international conference on neural information processing systems. NIPS'15, pp. 1396–1404. MIT Press, Cambridge, MA, USA (2015)
36. Tropp, J.A.: Randomized block krylov methods for approximating extreme eigenvalues. Numer. Math. **150**, 217–255 (2021)
37. Wang, C., Yi, Q., Liao, X., Wang, Y.: An improved frequent directions algorithm for low-rank approximation via block krylov iteration. IEEE Transactions on Neural Networks and Learning Systems (2023)
38. Yu, G., Feng, J., Chen, Z., Cai, X., Qi, L.: A randomized block krylov method for tensor train approximation. arXiv preprint arXiv:2308.01480 (2023)
39. Eckart, C., Young, G.M.: The approximation of one matrix by another of lower rank. Psychometrika **1**, 211–218 (1936)
40. Woodruff, D.P.: Sketching as a tool for numerical linear algebra. Found. Trends Theor. Comput. Sci. **10**(1–2), 1–157 (2014)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.