SIAM J. MATRIX ANAL. APPL.
Vol. 45, No. 4, pp. 1929–1953
© 2024 Society for Industrial and Applied Mathematics

A MULTILINEAR NYSTRÖM ALGORITHM FOR LOW-RANK APPROXIMATION OF TENSORS IN TUCKER FORMAT*

ALBERTO BUCCI† AND LEONARDO ROBOL†

Abstract. The Nyström method offers an effective way to obtain low-rank approximation of SPD matrices and has been recently extended and analyzed to nonsymmetric matrices (leading to the generalized Nyström method). It is a randomized, single-pass, streamable, cost-effective, and accurate alternative to the randomized SVD, and it facilitates the computation of several matrix low-rank factorizations. In this paper, we take these advancements a step further by introducing a higher-order variant of Nyström's methodology tailored to approximating low-rank tensors in the Tucker format: the multilinear Nyström technique. We show that, by introducing appropriate small modifications in the formulation of the higher-order method, strong stability properties can be obtained. This algorithm retains the key attributes of the generalized Nyström method, positioning it as a viable substitute for the randomized higher-order SVD algorithm.

Key words. low-rank approximation, Nyström method, randomized linear algebra, tensors, Tucker decomposition

MSC codes. 15A69, 65F55

DOI. 10.1137/23M1599343

See reproducibility of computational results at end of the article.

1. Introduction.

Multilinear arrays, or tensors, offer a natural way to model higher-order structures, such as multivariate functions, models depending on several parameters, or high-dimensional PDEs. For this reason, they are often encountered in the numerical treatment of such applications. On one hand, this allows a seamless description of such structures. On the other hand, problems dealing with multidimensional arrays suffer the so-called *curse of dimensionality* [3], which makes them computationally intractable due to memory requirements. More specifically, the storage requirements and the complexity of the numerical tools grow exponentially with the number of dimensions d.

There have been several efforts to reduce the cost associated with dealing with tensors, and the most promising techniques leverage the use of low-rank properties [15]. It turns out that giving an appropriate definition of tensor rank can be a challenging task. In contrast to the 2-dimensional case (i.e., when dealing with matrices), where the definition of rank is essentially unique (and is linked with the dimensions of the subspaces spanned by rows and columns, which are the same), for tensors, we have several alternatives. In the matrix case, the rank and the closest rank k approximant

*Received by the editors September 6, 2023; accepted for publication (in revised form) by M. Tygert June 28, 2024; published electronically October 15, 2024.
https://doi.org/10.1137/23M1599343
Funding: The authors are members of the INdAM Research group GNCS (Gruppo Nazionale di Calcolo Scientifico) and have been supported by the PRIN 2022 Project "Low-Rank Structures and Numerical Methods in Matrix and Tensor Computations and Their Application," and the MIUR Excellence Department Project awarded to the Department of Mathematics, University of Pisa, CUP I57G22000700001. The work of the second author was also supported by the National Research Center in High Performance Computing, Big Data and Quantum Computing (CN1 Spoke 6).
†Department of Mathematics, University of Pisa, Pisa 56127 Italy (alberto.bucci@phd.unipi.it, leonardo.robol@unipi.it).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

can be efficiently and stably computed with the singular value decomposition (SVD) [12].

The closest algebraic concept for $d > 2$ is finding the shortest decomposition in terms of outer products of d vector (which we define as rank 1 tensors). This is usually referred to as canonical polyadic decomposition (CP or CPD) [19] and does not lead to favorable properties for numerical computations (for instance, the set of tensors with rank at most k is not closed). An analogue of the SVD is not available in this setting. For this reason, it is convenient to look for other definitions of rank, such as the multilinear rank and the related Tucker decomposition (for which a higher-order SVD (HOSVD) is available [9]) or decompositions such as tensor trains [29] or hierarchical Tucker [14].

In this work, we focus on the Tucker decomposition [17] of low-rank tensors, and we address the problem of finding a randomized algorithm for the low-rank approximation in this format exploiting only tensor mode-j products or contractions.

It has been recently shown that randomization can be a powerful and highly successful tool in numerical linear algebra [20, 22, 28], especially for large-scale problems, where parallelization and limited access to data are needed. The main example of this situation is to find a near-optimal low-rank approximation to a matrix $A \in \mathbb{R}^{m\times n}$.

The underlying idea of randomized low-rank approximation algorithms is that the rows of a numerically low-rank matrix are almost linearly dependent, and they can be embedded into a low-dimensional space without substantially altering their geometric properties. In particular, it has been observed that using random sketch matrices to construct a dimensionality reduction map (DRM) is often an efficient and nonadaptive way to achieve this.

In this regard, consider Algorithm 1.1, described by Halko, Martinsson, and Tropp (HMT) in [16], to determine an orthogonal matrix $Q \in \mathbb{R}^{m\times(r+\ell)}$ such that $QQ^TA \approx A$ is a rank $r+\ell$ factorization of A.

Despite its simplicity, this strategy is efficient and comes with attractive theoretical guarantees: It provides a near-optimal low-rank approximation to A. When the matrix X is chosen with independent Gaussian distributed entries (with zero mean and unit variance), one can prove the following upper bound for the expected value of the approximation error:

$$\mathbb{E}\|A - QQ^TA\|_F \le \sqrt{1 + \frac{r}{\ell - 1}} \cdot \sqrt{\sum_{j>r} \sigma_j^2(A)}.$$

Since $\sqrt{\sum_{j>r} \sigma_j^2(A)}$ is the distance of A from the set of rank r matrices with respect to the Frobenius norm, the result is quasi-optimal up to a moderate constant. To better characterize the behavior of the randomized algorithm, one can describe the tail of the distribution. Under mild assumptions on ℓ (the oversampling parameter), we have

Algorithm 1.1. Randomized rangefinder (HMT).

Input $A \in \mathbb{R}^{m\times n}$, rank $r \le \min\{m, n\}$, oversampling parameter $\ell \ge 2$.

Output $Q \in \mathbb{R}^{m\times(r+\ell)}$ with $Q^TQ = I$ and such that $QQ^TA \approx A$.

Draw a random matrix $X \in \mathbb{R}^{n\times(r+\ell)}$;

Compute AX (sketching);

Compute Q, orthogonal factor of an economy-size QR of AX.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

$$\mathbb{P}\left\{\|A-QQ^TA\|_F > \left(1+\ell\sqrt{\frac{3r}{\ell+1}}+e\sqrt{2\ell(r+\ell)\log\ell}\right)\sqrt{\sum_{j>r}\sigma_j^2(A)}\right\} \le 3\ell^{-\ell}.$$

This and other results are discussed in [16] and the references therein.[1]

While very effective, the HMT method requires orthogonalization steps, which, in specific situations, may be relatively expensive or not available because of the use of particular architectures (e.g., GPUs) [25]. In addition, the method requires two passes, where the result of the first matrix-vector multiplication needs to be processed and then used in another matrix-vector multiplication. To reduce communication, single-pass algorithms are more attractive in several environments and have driven the interest in the design of "streamable" algorithms [21]. To mitigate this issue, one may rely on the Nyström method; originally developed for SPD matrices, it has been recently extended to general matrices in [35]. In [25], where the method is called generalized Nyström, or GN, a few tricks to improve stability are proposed and a detailed error analysis of the method is given.

The scheme of the GN method is the following: First, two DRMs $X \in \mathbb{R}^{n\times r}$ and $Y \in \mathbb{R}^{m\times(r+\ell)}$ for some $\ell \ge 1$ are generated; then, the low-rank approximation $\widehat{A}$ is obtained by the formula

$$\widehat{A} = AX(Y^TAX)^\dagger Y^TA. \tag{1.1}$$

The analysis in [25] shows that, as for HMT, the quality of the approximation provided by GN is near optimal and that the method can be implemented in a numerically stable fashion despite the presence of a (potentially) ill-conditioned pseudoinverse.

In scenarios involving sparsity or structured data, one may wish to have factors with the same structure, a feature that GN might not have. In this context, alternative approaches such as the CUR factorization proposed in [13] may be more appealing.

As regards tensors, several different randomized approaches for the Tucker approximation have been proposed in the literature [1, 8, 10, 23, 24, 33]. In [5, 11, 30, 32], for instance, the authors propose CUR-type algorithms for the decomposition of tensors.

Another straightforward approach is to replace every truncated SVD inside the HOSVD [9] or the sequential truncated HOSVD (STHOSVD) [36] algorithms with the HMT method [37]. However, such methods and the variants mentioned above are not streamable and may require (large and expensive) QR decompositions.

In this work, we present an extension of GN to tensors that recovers a Tucker decomposition; being a higher-order generalization of the Nyström method, we refer to the new version as *multilinear Nyström* (MLN).

The resulting method has near-optimal approximation quality and delivers results of comparable accuracy to other competing methods; the computational cost is near optimal for dense tensors, with small hidden constants, and, similarly to GN, it can be implemented in a numerically stable fashion. The method avoids QR factorizations of large matrices and only uses "advanced" linear algebra techniques (i.e., operations that are not matrix-vector products or matrix-matrix products) on small matrices. Hence, it is amenable to the implementation on various architectures with minimal requirements.

Theoretical results and numerical experiments show that the algorithm outperforms state-of-the-art methods in terms of memory requirements, computational cost, and number of accesses to the original tensor data.

[1] To obtain the tail bound for the approximation error in the Frobenius norm, we used $t = \ell$ and $u = \sqrt{2\ell\log\ell}$ in Theorem 10.7 of [16].

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

2. Preliminary concepts and notation. In this section, we introduce a few concepts and notations used throughout the paper. A *tensor* $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_d}$ is a d dimensional array with entries

$$a_{i_1 i_2 \ldots i_d}, \quad 1 \le i_k \le I_k, \quad k = 1, \ldots, d.$$

The symbols used for transposition, Moore–Penrose inverse, and Kronecker products of matrices are T, $\dagger$, and $\otimes$, respectively. We use $\|\cdot\|_F$ for the Frobenius norm and $\|\cdot\|_2$ for the spectral norm.

We will repeatedly use the unfolding operation, which consists of reshaping tensors into matrices. The operation is sometimes called matricization or flattening. More specifically, when $\mathcal{A}$ is a tensor, its *mode-k matricization* is denoted by $\mathcal{A}_k \in \mathbb{R}^{I_k \times \prod_{j \neq k} I_j}$ and satisfies

$$(\mathcal{A}_k)_{i_k, j} = \mathcal{A}_{i_1, \ldots, i_d},$$

where

$$j = 1 + \sum_{t=1, t \neq k}^{d} (i_t - 1) J_t, \quad J_t = \prod_{s=1, s \neq k}^{t-1} I_s.$$

The *mode-k product* of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_d}$ and a matrix $X \in \mathbb{R}^{J \times I_k}$ is denoted by $\mathcal{A} \times_k X$ and is such that

$$(\mathcal{A} \times_k X)_{i_1 \ldots i_{k-1} j i_{k+1} \ldots i_d} = \sum_{s=1}^{I_k} \mathcal{A}_{i_1 \ldots i_{k-1} s i_{k+1} \ldots i_d} X_{js}.$$

The mode-k product along all dimensions (i.e., for $k = 1, \ldots, d$) can be effectively expressed by leveraging a mix of matricizations and Kronecker products as follows:

$$(\mathcal{A} \times_1 X_1 \times \cdots \times X_d)_k = X_k \mathcal{A}_k (X_d \otimes \cdots \otimes X_{k+1} \otimes X_{k-1} \otimes \cdots \otimes X_1)^T.$$

The focus of this work is on Tucker decompositions, which are closely related with the concept of multilinear rank. Given a d-dimensional tensor, its multilinear rank is a tuple denoted by $rk^{ML}(\mathcal{A}) = (r_1, \ldots, r_d)$, where r_k is the matrix rank of $\mathcal{A}_k$, its mode-k matricization [9].

When we need to compare multilinear ranks of different tensors, we say that $rk^{ML}(\mathcal{A}) \le rk^{ML}(\mathcal{B})$ if the multilinear rank of $\mathcal{A}$ is componentwise smaller than the one of $\mathcal{B}$.

Several tensors of interest have some low-rank properties [15], which often only hold in an approximate sense (i.e., they are not low-rank, but they are close to a low-rank tensor in an appropriate metric). We introduce the class of ϵ-approximable tensors, which makes this idea more precise.

DEFINITION 2.1. *Given a tuple $(r_1, \ldots, r_d)$ and $\epsilon > 0$, we define $\mathcal{T}_\epsilon(r_1, \ldots, r_d)$ as the set*

$$\mathcal{T}_\epsilon(r_1, \ldots, r_d) := \{\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d} \mid \exists \|\mathcal{E}\|_F \le \epsilon \text{ with } rk^{ML}(\mathcal{A} + \mathcal{E}) \le (r_1, \ldots, r_d)\}.$$

When multiplying a tensor with mode-k products along different modes (for instance, for $k = 1, \ldots, d$), the order of the multiplication is irrelevant since the operations commute. Hence, we often shorten formulas involving this kind of product as follows:

$$\mathcal{A} \times_{i=1}^{d} X_i = \mathcal{A} \times_1 X_1 \cdots \times_d X_d,$$
$$X_{\otimes \check{k}} = X_d \otimes \cdots \otimes X_{k+1} \otimes X_{k-1} \otimes \cdots \otimes X_1.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

We also remark that the following properties hold and are easily verified using the formulations of mode-k products by means of matricization:

$$\mathcal{A} \times_k X_i \times_k X_j = \mathcal{A} \times_k X_j X_i, \qquad (\mathcal{A} \times_{i=1}^{d} X_i)_k = X_k \mathcal{A}_k X_{\otimes \check{k}}^T.$$

We denote by $Q = \mathrm{orth}(X)$ the Q factor of an economy-size QR factorization of a matrix X with more rows than columns.

3. Randomized matrix low-rank approximation. The analysis of the MLN method is based on results from the matrix case, which are briefly reviewed in this section. More specifically, we consider the approximant obtained by the HMT scheme for finding an orthogonal basis of the column span (as in [16]) and the GN scheme from [25].

Given a matrix A of size $m \times n$, the approximants obtained by the HMT and GN methods are given, respectively, by

$$\widehat{A}_{HMT} = Q(Q^T A) \quad \text{and} \quad \widehat{A}_{GN} = AX(Y^T AX)^\dagger Y^T A,$$

where $Q = \mathrm{orth}(AX)$ and $X \in \mathbb{R}^{n \times r}$, $Y \in \mathbb{R}^{m \times (r+\ell)}$ are two DRM matrices.

We denote by E_{HMT} the error of the approximation in the Frobenius norm of the HMT approximant and with E_{GN} that of GN; we have the following upper bounds:

$$E_{HMT} \le \|\widetilde{\Sigma}\|_F \sqrt{1 + \|\widetilde{V}_\perp^T X\|_2^2 \|(\widetilde{V}^T X)^\dagger\|_2^2}, \tag{3.1}$$

$$E_{GN} \le E_{HMT} \sqrt{1 + \|Q_\perp^T Y\|_2^2 \|(Q^T Y)^\dagger\|_2^2}, \tag{3.2}$$

where, for any $\widehat{r} < r$, $\widetilde{\Sigma}$ is the diagonal term in the SVD of A with a 0 in place of the first $\widehat{r}$ singular values and $\widetilde{V}$ is the orthogonal matrix with the first $\widehat{r}$ right singular vectors of A.

The term $\|\widetilde{\Sigma}\|_F$ is the optimal error that would be obtained by a truncated SVD; hence, it is clear that it is important to choose the DRM in a way that makes the other terms as small as possible (with high probability). At the same time, we wish to maintain the cost of taking the matrix-vector products small, so it makes sense to use DRMs drawn from a set of structured matrices that have fast matrix-vector products routines available.

A few choices of random samplings that allow for fast multiplications arise from subsampling trigonometric transforms. Examples include the subsampled randomized Hadamard transform (denoted by SRHT) [4, 34], and the subsampled randomized Fourier transform (SRFT) [31]. These approaches reduce the cost of forming AX to $\mathcal{O}(mn \log n)$, where n is the number of columns of X and m the number of rows of A. The theory for these subsampled transforms can be more complex than the one for more "classical" choices, such as Gaussian matrices; the latter are deeply understood and have sharp error bounds available (see [22] and the references therein).

The use of GN has a few advantages with respect to the HMT scheme; it avoids orthogonalizations and can be used as a single-pass approximation method.

However, without a proper implementation, the stability of GN can be cause for concern. The pseudocode in Algorithm 3.1 reports the implementation suggested in [25].

In practice, a slight oversample parameter ℓ makes this implementation stable, but no theoretical assessments have been conducted. While stability cannot be established for (1.1) as is, there is an inexpensive modification that guarantees stability:

$$\widehat{A} = AX(Y^T AX)_\epsilon^\dagger Y^T A, \tag{3.3}$$

which is the stabilized generalized Nyström (SGN) method.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Algorithm 3.1. GN.

Input $A \in \mathbb{R}^{m\times n}$, rank $r \le \min\{m,n\}$, oversampling parameter $\ell \ge 1$.

Output Low-rank approximant $\widehat{A}$ of A.

Draw two random matrices $X \in \mathbb{R}^{n\times r}$ and $Y \in \mathbb{R}^{m\times(r+\ell)}$;
Compute AX, Y^TA;
Compute an economy-size QR factorization $Y^TAX = ZR$;
Compute $\widehat{A} = ((AX)R^{-1})(Z^T(Y^TA))$.

Here, $(Y^TAX)_\epsilon$ denotes the ϵ-pseudoinverse; that is, if

$$Y^TAX = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1 & V_2 \end{bmatrix}^T$$

is the SVD, where Σ_1 contains singular values larger than ϵ, then $(Y^TAX)_\epsilon^\dagger = V_1\Sigma_1^{-1}U_1^T$. In the following, ϵ will be chosen as a modest multiple of the unit roundoff u times $\|A\|_F$; that is, $\epsilon = \mathcal{O}(u\|A\|_F)$.

Different strategies to implement SGN in a stable manner can be found in [25].

4. MLN. This section is devoted to extending GN to tensors and, in particular, laying the ground for determining whether the stability analysis and the related guarantees that are explored in [25] for matrices carry over to the tensor setting.

To simplify the analysis, it is convenient to rewrite the classical Nyström approximant (1.1) in the matrix setting in a slightly different way. Leveraging the properties of the Moore–Penrose pseudoinverse, we have the following identity:

$$AX(Y^TAX)^\dagger Y^TA = AX(Y^TAX)^\dagger Y^TAX(Y^TAX)^\dagger Y^TA.$$

The above reformulation identifies a structure in the approximant, which is formed by a small core matrix Y^TAX and two matrices of the form $AX(Y^TAX)^\dagger$ and $(Y^TAX)^\dagger Y^TA = (A^TY(X^TA^TY)^\dagger)^T$ that invert the sketching procedure. Notice that the matrices $\mathcal{P}_1 := AX(Y^TAX)^\dagger Y^T$ and $\mathcal{P}_2 := A^TY(X^TA^TY)^\dagger X^T$ are approximate oblique projections on the column space of A and A^T, respectively.

This formulation can be replicated in the tensor setting. Given a tensor $\mathcal{A} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ and sketch matrices $X_i \in \mathbb{R}^{n_i\times r_i}$ $i = 1,\dots,d$, we form a core tensor $\mathcal{A} \times_{i=1}^d X_i^T$ by sketching $\mathcal{A}$ in all the modes, and we construct the matrices $(\mathcal{A} \times_{i\ne k} X_i^T)_k(\mathcal{A} \times_{i=1}^d X_i^T)_k^\dagger$ that give an approximate basis of the column span of the mode-k matricization.

This leads to a randomized algorithm for finding a low-rank representation of a tensor in Tucker format. Formally, we define the approximant $\widehat{\mathcal{A}}$ of $\mathcal{A}$ by

$$\begin{aligned} \widehat{\mathcal{A}} &= (\mathcal{A} \times_{i=1}^d X_i^T) \times_{k=1}^d \left((\mathcal{A} \times_{i\ne k} X_i^T)_k (\mathcal{A} \times_{i=1}^d X_i^T)_k^\dagger \right) \\ &= (\mathcal{A} \times_{i=1}^d X_i^T) \times_{k=1}^d \left(\mathcal{A}_k X_{\otimes\check{k}} (X_k^T \mathcal{A}_k X_{\otimes\check{k}})^\dagger \right). \end{aligned} \tag{4.1}$$

Note that, when $d = 2$, the matrix $\widehat{A}_1$, flattening along the first index of (4.1), is indeed the reformulation of the GN approximant proposed above. In fact, we have

$$\widehat{A}_1 = AX_2(X_1^TAX_2)^\dagger(X_1^TAX_2)(X_1^TAX_2)^\dagger X_1^TA = AX_2(X_1^TAX_2)^\dagger X_1^TA. \tag{4.2}$$

It is convenient to define, for each $k = 1,\dots,d$, the projection matrix

$$\mathcal{P}_k := \left((\mathcal{A} \times_{i\ne k} X_i^T)_k (\mathcal{A} \times_{i=1}^d X_i^T)_k^\dagger \right) X_k^T = \mathcal{A}_k X_{\otimes\check{k}} (X_k^T \mathcal{A}_k X_{\otimes\check{k}})^\dagger X_k^T. \tag{4.3}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

The matrix $\mathcal{P}_k$ is an oblique projection onto the column space of $\mathcal{A}_k$ and allows us to rewrite (4.1) in the compact form

$$\widehat{\mathcal{A}} = \mathcal{A} \times_{k=1}^{d} \mathcal{P}_k.$$

Note that the formula we mentioned above is not new and was already present in Caiafa and Cichocki [6]. This paper was published before the formulation of GN in [35, 25]. To the best of our knowledge, the connection between these two works has not been emphasized so far.

An important feature of (4.1) is that, when the tensor $\mathcal{A}$ is expressed in Tucker format (that is, $\mathcal{A} = \mathcal{C} \times_{k=1}^{d} U_k$ with small core tensor $\mathcal{C}$), the cost of computing the different mode products drops dramatically; we can first compute the products $\Psi_k = X_k^T U_k$ and then the mode products between $\mathcal{C}$ and the Ψ_k. This suggests that (4.1) can be used as an effective method for recompression of tensors in Tucker format.

For the matrix case ($d = 2$) in the formulation of (3.3), the key property that allows us to prove some form of stability under floating-point inaccuracies is to have unbalanced dimensions in X and Y; the fact that Y has $r + \ell$ rows and X only r plays the same role of the oversampling in Algorithm 1.1 and allows us to stabilize the least-square solution. This property is lost when we factor out the projections as in (4.1). Hence, the reformulation of (4.1) has two opposite effects on the design of the low-rank approximation scheme:

- On one hand, it makes extending the approach to $d > 2$ much easier because it suffices to define the oblique projectors $\mathcal{P}_k$ and apply them on all modes from $k = 1, \ldots, d$.
- On the other hand, it makes a stability analysis more difficult (or impossible without further modifications) because any immediate bound will depend on the norms of $\mathcal{P}_k$, which are hard to control.

We have already discussed how the reformulation allows for an easy extension for a generic d. In the next section, we will show that we can introduce further degrees of freedom in the choice of samplings, and this will greatly help in slightly modifying the method to make it stable. The core idea is that, instead of fixing only d sampling matrices $X_1, \ldots, X_d$, we can choose $2d$ by introducing additional samplings $Y_1, \ldots, Y_d$. This will allow us to reintroduce the unbalanced dimension (and the stabilized least-square solvers) into the picture.

4.1. Introducing more sketching matrices. This section introduces a small variant of the sketching procedure described in (4.1) that allows us to select more than d sketching matrices ($2d$ instead of d). The aim of this generalization is to design an approximation method with better stability properties.

To make it easier to follow the discussion, we start by showing how this generalization can be formulated in the matrix case.

Recall that, according to the previous discussion, the approximation provided by GN can be written as

$$AX(Y^T AX)^\dagger Y^T AX(Y^T AX)^\dagger Y^T A.$$

Instead of using the same X and Y for the two projections $AX(Y^T AX)^\dagger Y^T$ and $X(Y^T AX)^\dagger Y^T A$, we may achieve more generality by using different sketching matrices ($X_1 \in \mathbb{R}^{n_1 \times r_1}$ and $Y_1 \in \mathbb{R}^{m_1 \times (r_1 + \ell_1)}$ for the first projection and $X_2 \in \mathbb{R}^{n_2 \times r_2}$ and $Y_2 \in \mathbb{R}^{m_2 \times (r_2 + \ell_2)}$ for the second), still ensuring that, whenever A is of low rank, an exact representation is retrieved (at least theoretically, if no floating-point errors are considered). The resulting approximant would be as follows:

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

$$\widehat{A} = AX_1(Y_1^T AX_1)^\dagger Y_1^T AY_2(X_2^T AY_2)^\dagger X_2^T A. \tag{4.4}$$

Note that, in the second projection, not only have we substituted X and Y with X_2 and Y_2, but we have also swapped them for reasons of symmetry.

In addition, by looking at $\widehat{A}$ as a 2-dimensional tensor, this approximation can be obtained by applying oblique projectors $\mathcal{P}_1$ and $\mathcal{P}_2$ along modes 1 and 2, respectively, with the projectors $\mathcal{P}_1 = AX_1(Y_1^T AX_1)^\dagger Y_1^T$ and $\mathcal{P}_2 = A^T X_2(Y_2^T A^T X_2)^\dagger Y_2^T$. We use the same notation $\mathcal{P}_k$ used in (4.3) for these more general projectors since there is no risk of confusion.

Going back to the notation of mode-j products, we may rewrite the approximation as follows:

$$\widehat{A} = A \times_1 \mathcal{P}_1 \times_2 \mathcal{P}_2.$$

In the matrix case $d = 2$, this idea is not particularly appealing since it requires more matrix-vector products.

For the "standard" GN, only one of two projections needs to be computed since we may write

$$\widehat{A} = A \times_1 \mathcal{P}_1 \times_2 \mathcal{P}_2 = A \times_1 \mathcal{P}_1 = A \times_2 \mathcal{P}_2.$$

The idea will, however, be valuable in the tensor setting, where we have already anticipated that the simplification of the projections that happens for $d = 2$ is not easily obtainable. The next section is devoted to analyzing (4.4) in detail; we anticipate that its accuracy is just slightly less than GN but still near optimal, its cost is about twice as expensive as GN, and crucially, it can be implemented in a numerically stable fashion.

This more general approximant is expressed in a form that can be readily extended to the tensor setting. Given a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and sketch matrices $X_i \in \mathbb{R}^{\prod_{j \neq i} n_j \times r_i}$ and $Y_i \in \mathbb{R}^{n_i \times (r_i + \ell_i)}$, $i = 1, \ldots, d$, we define the MLN approximant $\widehat{\mathcal{A}}$ of $\mathcal{A}$ as follows:

$$\widehat{\mathcal{A}} = (\mathcal{A} \times_{k=1}^d Y_k^T) \times_{k=1}^d \mathcal{A}_k X_k (Y_k^T \mathcal{A}_k X_k)^\dagger. \tag{4.5}$$

The approximant is obtained by sketching with matrices Y_k and using oblique projections built with the matrices X_k in order to construct a low-rank Tucker factorization. The oblique projectors are easy to write explicitly; we have $\mathcal{P}_k := \mathcal{A}_k X_k (Y_k^T \mathcal{A}_k X_k)^\dagger Y_k^T$, and this will be helpful for our analysis later on. In (4.5), we assumed Y_k with more columns than X_k. Nevertheless, it is worth noting that, by relaxing this assumption, the resulting reformulation is an extension of the original one in (4.1), which can be recovered by substituting Y_k with X_k and X_k with $X_{\otimes \check{k}}$ in (4.5).

The pseudocode in Algorithm 4.1 describes the method for computing the approximant in (4.5). The QR factorization in the pseudocode is the standard economy-size QR factorization without pivoting.

We remark that, even though we obtained (4.5) with a different approach, the MLN approximant turns out to be mathematically equivalent to the one provided by Algorithm 4.3 in [33]. There are, however, algorithmic differences; our implementation avoids the expensive QR factorization of $A_k X_k$ and, as we prove in section 5.3, has the same stability properties of GN.

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Algorithm 4.1. MLN.

Input $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, multilinear rank $r = (r_1, \ldots, r_d) \leq (n_1, \ldots, n_d)$, oversampling vector $\ell = (\ell_1, \ldots, \ell_d)$.

Output Low-rank Tucker approximant $\widehat{\mathcal{A}}$ of $\mathcal{A}$.

for $k = 1, \ldots, d$

Draw random matrices $X_k \in \mathbb{R}^{\prod_{i \neq k} n_i \times r_k}$ and $Y_k \in \mathbb{R}^{n_k \times (r_k + \ell_k)}$;

Compute $\mathcal{A}_k X_k$, $Y_k^T \mathcal{A}_k$;

Compute an economy-size QR factorization $Y_k^T \mathcal{A}_k X_k = Z_k R_k$;

Compute $\widehat{\mathcal{A}} = ((\mathcal{A} \times_{k=1}^{d} Y_k^T) \times_{k=1}^{d} Z_k^T) \times_{k=1}^{d} (\mathcal{A}_k X_k R_k^{-1})$.

end

5. Properties of MLN. In this section, we prove that the accuracy of MLN is near optimal and that the version of Algorithm 4.1 with the ε-pseudoinverse, stabilized multilinear Nyström (SMLN) may be implemented in a stable way.

Concerning accuracy, our objective is to show that, choosing appropriate sketchings, the performances attained by MLN are close to the one of randomized HOSVD (RHOSVD) (a tensor version of HMT), which is, in turn, close to the HOSVD with high probability [37].

For what concerns the stabilization, we will show how some ideas from [25] can be generalized from the matrix to the tensor setting, allowing stronger stability guarantees. This will be obtained thanks to the particular choices of sketchings that we made in the previous section, introducing the matrices Y_k, and is not easy to obtain otherwise (such as in the first MLN proposed by Caiafa and Cichocki [6], where some form of stability has been shown only in the matrix case $d = 2$).

5.1. Accuracy of MLN. In order to prove the results related to the accuracy of the MLN scheme, we now introduce a few auxiliary lemmas.

We denote by $\mathcal{P}_k$ the oblique projections $\mathcal{P}_k := \mathcal{A}_k X_k (Y_k^T \mathcal{A}_k X_k)^\dagger Y_k^T$. In this way, we may write the approximation obtained by MLN with the compact notation $\widehat{\mathcal{A}} = \mathcal{A} \times_{k=1}^{d} \mathcal{P}_k$.

LEMMA 5.1. *Let $\mathcal{A}$ be a d-dimensional tensor and $\mathcal{P}_k := \mathcal{A}_k X_k (Y_k^T \mathcal{A}_k X_k)^\dagger Y_k^T$ for sketching matrices X_k, Y_k of compatible dimensions. Then, the following inequality holds:*

$$\|\mathcal{A} - \widehat{\mathcal{A}}\|_F \leq \sum_{k=1}^{d} \|\mathcal{A} \times_{i=1}^{k-1} \mathcal{P}_i - \mathcal{A} \times_{i=1}^{k} \mathcal{P}_i\|_F. \tag{5.1}$$

Proof. We expand the approximation error $\mathcal{A} - \widehat{\mathcal{A}}$ in the telescopic sum

$$\mathcal{A} - \widehat{\mathcal{A}} = \sum_{k=1}^{d} \left(\mathcal{A} \times_{i=1}^{k-1} \mathcal{P}_i - \mathcal{A} \times_{i=1}^{k} \mathcal{P}_i \right).$$

The result follows by taking Frobenius norms on both sides and using the subadditivity property. ∎

The terms in the summation above are of the form $\|\mathcal{B} - \mathcal{B} \times_k \mathcal{P}_k\|_F$, where $\mathcal{B} = \mathcal{A} \times_{i=1}^{k-1} \mathcal{P}_i$. This fact allows us to relate the approximation error of MLN with the approximation error of a GN obtained by appropriately flattening the tensors.

The next lemma is a step in this direction; it relates the approximation error along mode-1 with the error in the projection used with HMT on the matricization.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

The result is in line with the one proved in [25] for the matrix case when analyzing GN.

LEMMA 5.2. *Let $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and $X_1 \in \mathbb{R}^{\prod_{k \neq 1} n_k \times r_1}$, and let $Q_1 R_1$ denote an economy-size QR of $\mathcal{A}_1 X_1 \in \mathbb{R}^{n_1 \times r_1}$. If Y_1 has $r_1 + \ell_1$ columns and $Y_1^T Q_1$ has rank r_1, then*

$$\|\mathcal{A} - \mathcal{A} \times_1 \mathcal{P}_1\|_F \leq \|Q_{1\perp}^T \mathcal{A}_1\|_F \sqrt{1 + \|(Y_1^T Q_1)^\dagger\|_2^2 \|Y_1^T Q_{1\perp}\|_2^2}. \tag{5.2}$$

Proof. The Frobenius norm of a tensor coincides with the one of any of its matricizations; hence, we may write

$$\begin{aligned}\|\mathcal{A} - \mathcal{A} \times_1 \mathcal{P}_1\|_F &= \|\mathcal{A}_1 - Q_1 Q_1^T \mathcal{A}_1 + Q_1 Q_1^T \mathcal{A}_1 - \mathcal{A}_1 X_1 (Y_1^T \mathcal{A}_1 X_1)^\dagger Y_1^T \mathcal{A}_1\|_F \\ &= \|\mathcal{A}_1 - Q_1 Q_1^T \mathcal{A}_1 + Q_1 (Q_1^T - (Y_1^T Q_1)^\dagger Y_1^T) \mathcal{A}_1\|_F.\end{aligned}$$

Let us denote with $Q_{1\perp}$ a matrix whose columns span the orthogonal space to the columns of Q_1. Then, the following two identities hold:

$$Q_1 Q_1^T + Q_{1\perp} Q_{1\perp}^T = I, \tag{5.3}$$

$$(Q_1^T - (Y_1^T Q_1)^\dagger Y_1^T) Q_1 = 0. \tag{5.4}$$

The first relation is simply the decomposition of the identity as the projections over the column span of Q_1 and its orthogonal space; the second follows from our assumption that $Y_1^T Q_1$ is of full column rank. We make use of these two identities to further simplify the previous expression for $\|\mathcal{A} - \mathcal{A} \times_1 \mathcal{P}_1\|_F$:

$$\begin{aligned}\|\mathcal{A} - \mathcal{A} \times_1 \mathcal{P}_1\|_F &= \|\mathcal{A}_1 - Q_1 Q_1^T \mathcal{A}_1 + Q_1 (Q_1^T - (Y_1^T Q_1)^\dagger Y_1^T) \mathcal{A}_1\|_F, \\ [(5.3) + (5.4)] \to &= \|Q_{1\perp} Q_{1\perp}^T \mathcal{A}_1 + Q_1 (Q_1^T - (Y_1^T Q_1)^\dagger Y_1^T) Q_{1\perp} Q_{1\perp}^T \mathcal{A}_1\|_F, \\ Q_1^T Q_{1\perp} = 0 \to &= \|Q_{1\perp} Q_{1\perp}^T \mathcal{A}_1 - Q_1 (Y_1^T Q_1)^\dagger (Y_1^T Q_{1\perp}) Q_{1\perp}^T \mathcal{A}_1\|_F.\end{aligned}$$

To conclude, recall that, whenever two matrices A, B have orthogonal columns, we have $\|A + B\|_F^2 = \|A\|_F^2 + \|B\|_F^2$. Therefore,

$$\begin{aligned}\|\mathcal{A} - \mathcal{A} \times_1 \mathcal{P}_1\|_F^2 &= \|Q_{1\perp} Q_{1\perp}^T \mathcal{A}_1\|_F^2 + \|Q_1 (Y_1^T Q_1)^\dagger (Y_1^T Q_{1\perp}) Q_{1\perp}^T \mathcal{A}_1\|_F^2 \\ &\leq \|Q_{1\perp}^T \mathcal{A}_1\|_F^2 + \|(Y_1^T Q_1)^\dagger\|_2^2 \|Y_1^T Q_{1\perp}\|_2^2 \|Q_{1\perp}^T \mathcal{A}_1\|_F^2 \\ &\leq \|Q_{1\perp}^T \mathcal{A}_1\|_F^2 \cdot \left(1 + \|(Y_1^T Q_1)^\dagger\|_2^2 \|Y_1^T Q_{1\perp}\|_2^2\right).\end{aligned}$$

Taking the square root on both sides of the inequality gives us the claim. ∎

Lemma 5.2 is stated for $\mathcal{P}_1$ but clearly holds for any $\mathcal{P}_i$ with $i = 1, \ldots, d$ up to permuting the indices. Hence, a rough bound for the accuracy of the low-rank approximation may be obtained by bounding the terms in (5.1) as follows:

$$\|\mathcal{A} \times_{i=1}^{k-1} \mathcal{P}_i - \mathcal{A} \times_{i=1}^{k} \mathcal{P}_i\|_F \leq \|\mathcal{A} - \mathcal{A} \times_k \mathcal{P}_k\|_F \prod_{i=1}^{k-1} \|\mathcal{P}_i\|_2.$$

We may then proceed by finding upper bounds for $\|\mathcal{A} - \mathcal{A} \times_k \mathcal{P}_k\|_F$ and the norms of the projections $\mathcal{P}_k$ separately. However, as discussed in [25, section 3.3], this would lead to a large overestimate.

To obtain more predictive bounds, we follow another approach and consider the tensor $\mathcal{A} \times_{i=1}^{k} \mathcal{P}_i$ as the tensor $\mathcal{A} \times_{i=1}^{k-1} \mathcal{P}_i$ projected along the kth mode. This yields the following result. Since the proof follows similar steps to the one of Lemma 5.2, some details are omitted.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

LEMMA 5.3. *Let* $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ *and* $X_k \in \mathbb{R}^{\prod_{i \neq k} n_i \times r_k}$, *and let* $Q_k R_k$ *denote an economy-size QR of* $\mathcal{A}_k X_k \in \mathbb{R}^{n_k \times r_k}$. *If* Y_k *has* $r_k + \ell_k$ *columns and* $Y_k^T Q_k$ *has rank* r_k, *then, setting* $E_k := \|\mathcal{A} \times_{i=1}^{k-1} \mathcal{P}_i - \mathcal{A} \times_{i=1}^{k} \mathcal{P}_i\|_F$, *we have*

$$E_k \leq \Big(\|Q_{k\perp}^T \mathcal{A}_k\|_F + \|\mathcal{A} - \mathcal{A} \times_{i=1}^{k-1} \mathcal{P}_i\|_F \Big) \sqrt{1 + \|(Y_k^T Q_k)^\dagger\|_2^2 \|Y_k^T Q_{k\perp}\|_2^2}.$$

Proof. Let us define $B_1 = I$ and $B_k = I \otimes \cdots \otimes I \otimes \mathcal{P}_{k-1}^T \otimes \cdots \otimes \mathcal{P}_1^T$ for $k > 1$ so we may write

$$E_k = \|\mathcal{A} \times_{i=1}^{k-1} \mathcal{P}_i - \mathcal{A} \times_{i=1}^{k} \mathcal{P}_i\|_F = \|\mathcal{A}_k B_k - \mathcal{P}_k \mathcal{A}_k B_k\|_F.$$

We now follow the analogous steps to the proof for Lemma 5.2 but taking into account the effect of B_k, which yields

$$\begin{aligned}
E_k^2 &= \|\mathcal{A}_k B_k - Q_k Q_k^T \mathcal{A}_k B_k + Q_k Q_k^T \mathcal{A}_k B_k - \mathcal{A}_k X_k (Y_k^T \mathcal{A}_k X_k)^\dagger Y_k^T \mathcal{A}_k B_k\|_F^2 \\
&= \|Q_{k\perp} Q_{k\perp}^T \mathcal{A}_k B_k + Q_k (Q_k^T - (Y_k^T Q_k)^\dagger Y_k^T) \mathcal{A}_k B_k\|_F^2 \\
&= \|Q_{k\perp} Q_{k\perp}^T \mathcal{A}_k B_k - Q_k (Y_k^T Q_k)^\dagger (Y_k^T Q_{k\perp}) Q_{k\perp}^T \mathcal{A}_k B_k\|_F^2 \\
&= \|Q_{k\perp} Q_{k\perp}^T \mathcal{A}_k B_k\|_F^2 + \|Q_k (Y_k^T Q_k)^\dagger (Y_k^T Q_{k\perp}) Q_{k\perp}^T \mathcal{A}_k B_k\|_F^2 \\
&\leq \|Q_{k\perp}^T \mathcal{A}_k B_k\|_F^2 + \|(Y_k^T Q_k)^\dagger\|_2^2 \|Y_k^T Q_{k\perp}\|_2^2 \|Q_{k\perp}^T \mathcal{A}_k B_k\|_F^2 \\
&\leq \|Q_{k\perp}^T \mathcal{A}_k B_k\|_F^2 \Big(1 + \|(Y_k^T Q_k)^\dagger\|_2^2 \|Y_k^T Q_{k\perp}\|_2^2 \Big) \\
&= \|Q_{k\perp}^T (\mathcal{A}_k + \mathcal{A}_k B_k - \mathcal{A}_k)\|_F^2 \Big(1 + \|(Y_k^T Q_k)^\dagger\|_2^2 \|Y_k^T Q_{k\perp}\|_2^2 \Big) \\
&\leq \Big(\|Q_{k\perp}^T \mathcal{A}_k\|_F + \|\mathcal{A}_k - \mathcal{A}_k B_k\|_F \Big)^2 \Big(1 + \|(Y_k^T Q_k)^\dagger\|_2^2 \|Y_k^T Q_{k\perp}\|_2^2 \Big).
\end{aligned}$$

As in the previous result, the thesis follows by taking the square root on both sides of the identity. ∎

Remark 5.4. The term $\|Q_{k\perp}^T \mathcal{A}_k\|_F$ is the approximation error of HMT of the matrix $\mathcal{A}_k$ with sketch matrix X_k. Thus, by (3.1), we have

$$\|Q_{k\perp}^T \mathcal{A}_k\|_F \leq \|\widetilde{\Sigma}_k\|_F \sqrt{1 + \|\widetilde{V}_{k\perp}^T X_k\|_2^2 \|(\widetilde{V}_k^T X_k)^\dagger\|_2^2},$$

where, for any $\widehat{r}_k < r_k$, $\widetilde{V}_k$ is the orthogonal matrix with the leading $\widehat{r}_k$ right singular vectors of $\mathcal{A}_k$ and $\widetilde{\Sigma}_k$ is the diagonal term in the SVD of $\mathcal{A}_k$ with a 0 in place of the first $\widehat{r}_k$ singular values.

We now combine these results to state a deterministic accuracy bound for MLN. Here, deterministic means that the bound is exact as long as the sketchings X_k, Y_k have been fixed. When these sketches are instead random variables with a known distribution, the result yields probabilistic estimates.

Recall that we denote by $\mathcal{T}_\varepsilon(r_1, \ldots, r_d)$ the set of tensors that admit an approximation with multilinear rank at most $(r_1, \ldots, r_d)$ and an approximation error at most ϵ in the Frobenius norm (see Definition 2.1).

THEOREM 5.5 (deterministic accuracy bound). *Let* $\mathcal{A} \in \mathcal{T}_\varepsilon(r_1, \ldots, r_d)$, *let* $\widehat{\mathcal{A}}$ *be the approximant in* (4.5), *and set*

$$\tau_k := \sqrt{1 + \|(Y_k^T Q_k)^\dagger\|_2^2 \|Y_k^T Q_{k\perp}\|_2^2} \quad \textit{and} \quad \rho_k := \sqrt{1 + \|\widetilde{V}_{k\perp}^T X_k\|_2^2 \|(\widetilde{V}_k^T X_k)^\dagger\|_2^2},$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

where $\widetilde{V}_k$ is an orthogonal matrix with the first r_k right singular vectors of $\mathcal{A}_k$ and $Q_k = \operatorname{orth}(\mathcal{A}_k X_k)$. Then, denoting with $\tau = \max_k \tau_k$ and $\rho = \max_k \rho_k$, the following bound holds:

$$\|\mathcal{A} - \widehat{\mathcal{A}}\|_F \le \varepsilon\rho((1+\tau)^d - 1).$$

Proof. Note that τ_k and ρ_k are defined in terms of Q_k and $\widetilde{V}_k$, which are uniquely determined only up to right multiplication by appropriate unitary matrices; we start by verifying that τ_k and ρ_k do not depend on the specific choice of Q_k and $\widetilde{V}_k$ and are therefore well-defined.

Since $\widetilde{V}_k^T X_k$ is square and $Y_k^T Q_k$ has more rows than columns, for any unitary matrices Z, W,

$$(Z^T \widetilde{V}_k^T X_k)^\dagger = (\widetilde{V}_k^T X_k)^\dagger Z, \qquad (W^T Y_k^T Q_k)^\dagger = (Y_k^T Q_k)^\dagger W.$$

Hence, thanks to the invariance of the spectral norm under unitary transformation, we conclude that τ_k and ρ_k do not depend on the specific choice of Q_k and $\widetilde{V}_k$, as desired.

We now prove that the sought inequality holds. Thanks to (5.1), to obtain the claim, it is sufficient to bound terms of the form $E_k = \|\mathcal{A} \times_{i=1}^{k-1} \mathcal{P}_i - \mathcal{A} \times_{i=1}^{k} \mathcal{P}_i\|_F$, which then leads to the upper bound

$$\|\mathcal{A} - \widehat{\mathcal{A}}\|_F \le \sum_{k=1}^{d} E_k. \tag{5.5}$$

We use Lemma 5.2 and Remark 5.4 to obtain the upper bound

$$E_1 \le \varepsilon_1 \rho_1 \tau_1,$$

where ε_k denotes the best possible error of approximation in the Frobenius norm of rank r_k of $\mathcal{A}_k$. Similarly, we make use of Lemma 5.3 for all the remaining modes, which yields for $k = 1, \dots, d-1$ the recurrence relation

$$E_{k+1} \le (\varepsilon_k \rho_k + \|\mathcal{A} - \mathcal{A} \times_{i \le k} \mathcal{P}_i\|_F)\tau_k \le \left(\varepsilon_k \rho_k + \sum_{i \le k} E_i a\right)\tau_k \le \left(\varepsilon\rho + \sum_{i \le k} E_i\right)\tau.$$

In the last inequality, we used $\max_k \varepsilon_k \le \varepsilon$, which holds thanks to $\mathcal{A} \in \mathcal{T}_\varepsilon(r_1, \dots, r_d)$. Then, the E_k for $k = 1, \dots, d$ satisfy the vector inequality

$$\begin{bmatrix} \tau^{-1} & 0 & \cdots & \cdots & 0 \\ -1 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ -1 & \cdots & \cdots & -1 & \tau^{-1} \end{bmatrix} \begin{bmatrix} E_1 \\ \vdots \\ \vdots \\ \vdots \\ E_d \end{bmatrix} \le \begin{bmatrix} \varepsilon\rho \\ \vdots \\ \vdots \\ \vdots \\ \varepsilon\rho \end{bmatrix}. \tag{5.6}$$

Let T_k be the $k \times k$ principal minor of the lower triangular matrix in (5.6); both T_k and its inverse are lower triangular Toeplitz matrices, and we have the explicit formula

$$(T_k^{-1})_{ij} = \begin{cases} 0 & \text{if } i < j, \\ \tau & \text{if } i = j, \\ \tau\left[(1+\tau)^{i-j} - (1+\tau)^{i-j-1}\right] & \text{if } i > j. \end{cases}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

In particular, T_k^{-1} is nonnegative since $\tau > 0$, so we can left-multiply inequality (5.6) by $e_k^T T_k^{-1}$ and obtain an upper bound for E_k:

$$E_k \leq e_k^T T_k^{-1} \begin{bmatrix} \varepsilon\rho \\ \vdots \\ \varepsilon\rho \end{bmatrix}. \tag{5.7}$$

Using the explicit expression of the entries in the row vector $e_k^T T_k^{-1}$, we finally obtain the upper bound $E_k \leq \varepsilon\rho\tau(1+\tau)^{k-1}$. We conclude using (5.5):

$$\|\mathcal{A} - \widehat{\mathcal{A}}\|_F \leq \sum_{k=1}^{d} \varepsilon\rho\tau(1+\tau)^{k-1} \leq \varepsilon\rho((1+\tau)^d - 1). \qquad \square$$

5.2. Accuracy of SMLN. We now consider a modification of the proposed approach that improves the stability properties: we replace any pseudoinverse $M^\dagger$ appearing in the formulas with its regularized counterpart $M_\epsilon^\dagger$, which consists in treating the singular values below ϵ in M as zeros (see the definition of ϵ-pseudoinverse at the end of section 3). We refer to such a modification as SMLN. In practice, this amounts to replacing the projections $\mathcal{P}_k$ with

$$\widetilde{\mathcal{P}}_k := \mathcal{A}_k X_k (Y_k^T \widetilde{\mathcal{A}}_k X_k)_\epsilon^\dagger Y_k^T.$$

This modification is motivated by the following observations made in the analysis of the matrix case in [25]:

1. For the matrix GN, this change does not substantially change the attainable accuracy.
2. This modification makes the method reliable in the presence of inexact floating-point arithmetic.

This and section 5.3 investigate whether the same results hold in the tensor case for the generalization discussed in this paper. This section covers the first item (the accuracy), whereas section 5.3 discusses the second item (the stability).

Concerning the accuracy, we prove that the computed approximant $\widehat{\mathcal{A}} = \mathcal{A} \times_{k=1}^{d} \widetilde{\mathcal{P}}_k$ attains, in exact arithmetic, an error estimate of the form

$$\|\mathcal{A} - \widehat{\mathcal{A}}\|_F \leq \frac{(1+\tilde{\tau})^d - 1}{\tilde{\tau}} \max_{k=1,\ldots,d} \|E_{SGN}^{(k)}\|_F,$$

where $E_{SGN}^{(k)}$ is the error of the matrix SGN approximation computed in exact arithmetic for the mode-k matricization $\mathcal{A}_k$ with sketchings X_k, Y_k and where $\widetilde{\tau}$ is chosen so that $\|\widetilde{\mathcal{P}}_k\|_2 \leq \widetilde{\tau}$.

In relation to the stability result, to obtain a similar estimate, it will be necessary to take into account the floating-point error of approximation. In doing so, we will make the simplifying assumption that $\mathcal{A}_k X_k$ and $Y_k^T \mathcal{A}_k X_k$ are computed exactly. On one hand, this is obviously unrealistic. On the other hand, any sketching low-rank approximation method will use matrices of this form and will thus incur a similar approximation error; in this work, we prefer to focus on the error introduced by the algorithmic choice in SMLN that happens after the sketching.

The upcoming theorem proves the accuracy result discussed above.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

THEOREM 5.6. *Let* $\widetilde{\mathcal{P}}_k := \mathcal{A}_k X_k (Y_k^T \widetilde{\mathcal{A}}_k X_k)_\epsilon^\dagger Y_k^T$, *where* $\widetilde{\mathcal{A}}_k = \mathcal{A}_k + \delta\mathcal{A}_k$, *and set*

$$\widetilde{\varepsilon}_k := \|\mathcal{A} - \mathcal{A} \times_k \widetilde{\mathcal{P}}_k\|_F \quad \textit{and} \quad \widetilde{\tau}_k := 1 + \|\widetilde{\mathcal{P}}_k\|_2.$$

Then, denoting with $\widetilde{\varepsilon} := \max_k \widetilde{\varepsilon}_k$ *and* $\widetilde{\tau} := \max_k \widetilde{\tau}_k$, *we have*

$$\|\mathcal{A} - \mathcal{A} \times_{k=1}^d \widetilde{\mathcal{P}}_k\|_F \le \frac{\widetilde{\varepsilon}}{\widetilde{\tau}}((1+\widetilde{\tau})^d - 1).$$

Proof. Since

$$\mathcal{A} - \mathcal{A} \times_{k=1}^d \widetilde{\mathcal{P}}_k = \sum_{k=1}^d \mathcal{A} \times_{i=1}^{k-1} \widetilde{\mathcal{P}}_i - \mathcal{A} \times_{i=1}^{k} \widetilde{\mathcal{P}}_i,$$

by the subadditivity of the Frobenius norm, we can write

$$\|\mathcal{A} - \mathcal{A} \times_{k=1}^d \widetilde{\mathcal{P}}_k\|_F \le \sum_{k=1}^d \|\mathcal{A} \times_{i=1}^{k-1} \widetilde{\mathcal{P}}_i - \mathcal{A} \times_{i=1}^{k} \widetilde{\mathcal{P}}_i\|_F.$$

Observe that $E_{SGN}^{(1)} := \|\mathcal{A} - \mathcal{A} \times_1 \widetilde{\mathcal{P}}_1\|_F$ is exactly the error of the SGN approximant of $\mathcal{A}_1$ with sketches X_1 and Y_1. For $k \ge 2$, let $(\mathcal{A} \times_{i=1}^{k-1} \widetilde{\mathcal{P}}_i)_k := \mathcal{A}_k \widetilde{B}_k$ with $\widetilde{B}_k = I \otimes \cdots \otimes I \otimes \widetilde{\mathcal{P}}_{k-1}^T \otimes \cdots \otimes \widetilde{\mathcal{P}}_1^T$. We have

$$\begin{aligned}\|\mathcal{A} \times_{i=1}^{k-1} \widetilde{\mathcal{P}}_i - \mathcal{A} \times_{i=1}^{k} \widetilde{\mathcal{P}}_i\|_F &= \|\mathcal{A}_k \widetilde{B}_k - \widetilde{\mathcal{P}}_k \mathcal{A}_k \widetilde{B}_k\|_F = \|(I - \widetilde{\mathcal{P}}_k)\mathcal{A}_k \widetilde{B}_k\|_F \\ &= \|(I - \widetilde{\mathcal{P}}_k)(\mathcal{A}_k - \mathcal{A}_k + \mathcal{A}_k \widetilde{B}_k)\|_F \\ &\le \|(I - \widetilde{\mathcal{P}}_k)\mathcal{A}_k\|_F + \|(I - \widetilde{\mathcal{P}}_k)(\mathcal{A}_k - \mathcal{A}_k \widetilde{B}_k)\|_F.\end{aligned}$$

Again, $E_{SGN}^{(k)} := \|(I - \widetilde{\mathcal{P}}_k)\mathcal{A}_k\|_F$ is the error of the SGN approximant $\mathcal{A}_k$ with sketches X_k and Y_k.

It remains to bound the term $\|(I - \widetilde{\mathcal{P}}_k)(\mathcal{A}_k - \mathcal{A}_k \widetilde{B}_k)\|_F$. We can write the following chain of inequalities:

$$\begin{aligned}\|(I - \widetilde{\mathcal{P}}_k)(\mathcal{A}_k - \mathcal{A}_k \widetilde{B}_k)\|_F &\le \|I - \widetilde{\mathcal{P}}_k\|_2 \|\mathcal{A}_k - \mathcal{A}_k \widetilde{B}_k\|_F \le (1 + \|\widetilde{\mathcal{P}}_k\|_2)\|\mathcal{A}_k - \mathcal{A}_k \widetilde{B}_k\|_F \\ &= \widetilde{\tau}_k \|\mathcal{A}_k - \mathcal{A}_k \widetilde{B}_k\|_F = \widetilde{\tau}_k \|\mathcal{A} - \mathcal{A} \times_{i=1}^{k-1} \widetilde{\mathcal{P}}_i\|_F \\ &\le \widetilde{\tau}_k \sum_{s=1}^{k-1} \|\mathcal{A} \times_{i=1}^{s-1} \widetilde{\mathcal{P}}_i - \mathcal{A} \times_{i=1}^{s} \widetilde{\mathcal{P}}_i\|_F.\end{aligned}$$

Summarizing and denoting with $E^{(k)} := \|\mathcal{A} \times_{i=1}^{k-1} \widetilde{\mathcal{P}}_i - \mathcal{A} \times_{i=1}^{k} \widetilde{\mathcal{P}}_i\|_F$, we can write the following recurrence relation for an upper bound to the approximation error:

$$\begin{cases} E^{(1)} = E_{SGN}^{(1)} \le \widetilde{\varepsilon}, \\ E^{(k)} \le E_{SGN}^{(k)} + \widetilde{\tau}_k \sum_{i=1}^{k-1} E^{(i)} \le \widetilde{\varepsilon} + \widetilde{\tau} \sum_{i=1}^{k-1} E^{(i)}. \end{cases}$$

The latter system of inequalities is similar to that in (5.6), and analogous steps lead to the sought bound. ∎

The structure of the bound for SMLN is similar to the one of MLN. However, in this second case, $\widetilde{\tau}_k$ depends on the norm of $\widetilde{\mathcal{P}}_k$, which, in general, can be nonnegligible (for instance, it grows as $\mathcal{O}(\sqrt{n_k})$ in the Gaussian case [25]).

In practice, the proof of Theorem 5.6 can be modified to obtain a sharper bound. We avoided this change in Theorem 5.6 for the sake of clarity, but we give some details here. The key idea is to modify the bound for $\|(I - \widetilde{\mathcal{P}}_k)(\mathcal{A}_k - \mathcal{A}_k \widetilde{B}_k)\|_F$ in a way that changes the $\mathcal{O}(\sqrt{n_k})$ term into an $\mathcal{O}(\sqrt{r_k})$. We may write

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

$$\|(I-\widetilde{\mathcal{P}}_k)(\mathcal{A}_k-\mathcal{A}_k\widetilde{B}_k)\|_F \le \|\mathcal{A}_k-\mathcal{A}_k\widetilde{B}_k\|_F + \|\widetilde{\mathcal{P}}_k(\mathcal{A}_k-\mathcal{A}_k\widetilde{B}_k)\|_F.$$

The second term in the formula satisfies

$$\begin{aligned}\|\widetilde{\mathcal{P}}_k(\mathcal{A}_k-\mathcal{A}_k\widetilde{B}_k)\|_F &= \|\mathcal{A}_kX_k(Y_k^T\mathcal{A}_kX_k)_\epsilon^\dagger Y_k^T(\mathcal{A}_k-\mathcal{A}_k\widetilde{B}_k)\|_F\\ &\le \|\mathcal{A}_kX_k(Y_k^T\mathcal{A}_kX_k)_\epsilon^\dagger\|_2\|Y_k^T(\mathcal{A}_k-\mathcal{A}_k\widetilde{B}_k)\|_F.\end{aligned}$$

We refer the reader to [25, Theorem 3.3] for a detailed discussion on how to proceed from here to derive a sharper bound since the remaining steps coincide for $d=2$ or $d>2$. At a high level, we need to check two facts: The term $\|\mathcal{A}_kX_k(Y_k^T\mathcal{A}_kX_k)_\epsilon^\dagger\|_2$ is $\mathcal{O}(1)$, while $\|Y_k^T(\mathcal{A}_k-\mathcal{A}_k\widetilde{B}_k)\|_F$ is $\mathcal{O}(\sqrt{r_k})\|\mathcal{A}_k-\mathcal{A}_k\widetilde{B}_k\|_F$. The proof requires Gaussianity of X_k, Y_k. However, in practical applications, any class of random matrices such that the entries are $\mathcal{O}(1)$ and a rectangular realization is well-conditioned would work well, including the SRFT and SRHT matrices.

5.3. Stability of SMLN. So far, we have analyzed the accuracy of MLN and SMLN without taking into account roundoff errors in floating-point arithmetic. In this section, we address this potential issue.

Regarding the instability of MLN, the situation is similar to that of GN analyzed in [25]: Stability cannot be established, but the instability is usually benign, and one obtains satisfactory results. In this section, we establish the numerical stability of SMLN.

Consider the stabilized MLN approximation $(\mathcal{A}\times_{k=1}^d Y_k^T)\times_{k=1}^d \mathcal{A}_kX_k(Y_k^T\mathcal{A}_kX_k)_\epsilon^\dagger$ obtained in finite precision arithmetic. We assume that we can compute each Tucker factor $\widetilde{W}_k := \mathcal{A}_kX_k(Y_k^T\mathcal{A}_kX_k)_\epsilon^\dagger$ and $\mathcal{C}=\mathcal{A}\times_{k=1}^d Y_k^T$ separately. Then, we compare the floating representation with the original tensor $\mathcal{A}$: $\|\mathcal{A}-\mathcal{C}\times_{k=1}^d \mathrm{fl}(\widetilde{W}_k)\|_F$, where we used the notation $\mathrm{fl}(\cdot)$ to denote the outcome of an arithmetic operation in floating-point arithmetic.

As we have anticipated in section 5.2, and in line with the analysis in [25], we will assume that each row of $\widetilde{W}_k$ is computed by a backward stable underdetermined linear solver and that $\mathcal{A}_kX_k$ and $Y_k^T\mathcal{A}_kX_k$ are computed exactly.

We also borrow some notation from [25]; we use $\emptyset(1)$ to suppress terms involving dimensions of the problem or the ranks (like n_k, r_k), but not $1/\epsilon, \sigma_r^{-1}(\mathcal{A}_k)$. We use ϵ_* to denote a tensor, a matrix, or a scalar such that $\|\epsilon_*\|_F = \emptyset(u\|A\|_F)$. The precise value of ϵ_* may change from appearance to appearance. See [25] and [26] for the motivation behind this notation, which is standard practice in stability analysis.

Let us denote with $[M]_i$ the ith row of M. The following proofs are heavily based on the results from [25] in the matrix case, for which we give precise references.

LEMMA 5.7. *Let $\mathcal{A}$, X_k, Y_k be such that X_k, Y_k are Gaussian, $\mathcal{A}_kX_k$ is full column rank, and $Y_k^T\mathcal{A}_kX_k$ is tall. Suppose also that $\epsilon = \varnothing(u\|A\|_F)$ and that each row of $\mathcal{A}_kX_k(Y_k^T\mathcal{A}_kX_k)_\epsilon^\dagger$ is computed by a backward stable underdetermined linear solver. Then, with an exponentially high probability,*

$$\|\mathrm{fl}(\mathcal{A}_kX_k(Y_k^T\mathcal{A}_kX_k)_\epsilon^\dagger)\|_F \sim \varnothing(1). \tag{5.8}$$

Proof. We can assume, without loss of generality, that we can perform a preliminary scaling to have $\|\mathcal{A}_k\|_F=1$. Following the proof of [25, Theorem 4.1], we know that

$$\|[\mathrm{fl}(\mathcal{A}_kX_k(Y_k^T\mathcal{A}_kX_k)_\epsilon^\dagger)]_i\|_2 = \|[\mathcal{A}_kX_k+\epsilon_*]_i(Y_k^T\mathcal{A}_kX_k+\epsilon_*)_\epsilon^\dagger\|_2 \tag{5.9}$$

$$\le \|[\mathcal{A}_kX_k]_i(Y_k^T\mathcal{A}_kX_k+\epsilon_*)_\epsilon^\dagger\|_2 + \emptyset(1). \tag{5.10}$$

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Let $U\Sigma V^T$ be the SVD of $\mathcal{A}_k X_k$; we have

$$\begin{aligned}\|\mathcal{A}_k X_k (Y_k^T \mathcal{A}_k X_k + \epsilon_*)_\epsilon^\dagger\|_2 &= \|\Sigma V^T (Y_k^T \mathcal{A}_k X_k + \epsilon_*)_\epsilon^\dagger\|_2 \\ &= \|(Y_k^T U)^\dagger (Y_k^T \mathcal{A}_k X_k)(Y_k^T \mathcal{A}_k X_k + \epsilon_*)_\epsilon^\dagger\|_2.\end{aligned}$$

Let us denote by Ξ_i the error matrix in the above expression; we fix $(Y_k^T \mathcal{A}_k X_k + \epsilon_*)_\epsilon^\dagger = (Y_k^T \mathcal{A}_k X_k + \Xi_i)_\epsilon^\dagger$, and we have $\|\Xi_i\|_2 \sim \emptyset(u)$. Then,

$$\begin{aligned}\|\mathcal{A}_k X_k (Y_k^T \mathcal{A}_k X_k + \epsilon_*)_\epsilon^\dagger\|_2 &\le \|(Y_k^T U)^\dagger\|_2 \|(Y_k^T \mathcal{A}_k X_k + \Xi + \epsilon_*)(Y_k^T \mathcal{A}_k X_k + \Xi)_\epsilon^\dagger\|_2 \\ &\le \|(Y_k^T U)^\dagger\|_2 \|(Y_k^T \mathcal{A}_k X_k + \Xi_i)(Y_k^T \mathcal{A}_k X_k + \Xi_i)_\epsilon^\dagger\|_2 \\ &\quad + \|(Y_k^T U)^\dagger\|_2 \|\epsilon_* (Y_k^T \mathcal{A}_k X_k + \epsilon_*)_\epsilon^\dagger\|_2 \sim \emptyset(1).\end{aligned}$$

In the last equality, we used that $\|(Y_k^T U)^\dagger\|_2 \sim \emptyset(1)$, which follows from the fact that $Y_k^T U$ is tall-Gaussian and hence well-conditioned; that $\|(Y_k^T \mathcal{A}_k X_k + \Xi_i)(Y_k^T \mathcal{A}_k X_k + \Xi_i)_\epsilon^\dagger\|_2 = 1$; and that $\|\epsilon_* (Y_k^T \mathcal{A}_k X_k + \Xi_i)_\epsilon^\dagger\|_2 \le \|\epsilon_*\|_2/\epsilon \sim \emptyset(1)$.

Then, the rows of the computed matrix satisfy $\|[\mathrm{fl}(\mathcal{A}_k X_k (Y_k^T \mathcal{A}_k X_k)_\epsilon^\dagger)]_i\|_2 = \emptyset(1)$. As a consequence, the Frobenius norm of the computed matrix is also $\emptyset(1)$. ∎

We are now ready to prove the main stability result.

THEOREM 5.8. *Let the assumptions in Lemma* 5.7 *be satisfied, and suppose that we can form* $\mathcal{C} \times_{k=1}^d \widetilde{W}_k = (\mathcal{A} \times_{k=1}^d Y_k^T) \times_{k=1}^d \mathcal{A}_k X_k (Y_k^T \mathcal{A}_k X_k)_\epsilon^\dagger$ *by first computing the core tensor* $\mathcal{C}$ *and the matrices* $\widetilde{W}_k$ *and then performing the mode products between* $\mathcal{C}$ *and the* $\widetilde{W}_k$. *Set*

$$\widetilde{\varepsilon}_k := \|\mathcal{A} - \mathcal{A} \times_k \mathrm{fl}(\widetilde{W}_k) Y_k^T)\|_F \qquad \text{and} \qquad \widetilde{\tau}_k := 1 + \|\mathrm{fl}(\widetilde{W}_k)\|_2 \|Y_k^T\|_2.$$

Then, denoting with $\widetilde{\varepsilon} := \max_k \widetilde{\varepsilon}_k$ *and* $\widetilde{\tau} := \max_k \widetilde{\tau}_k$, *we have*

$$\|\mathcal{A} - \mathcal{C} \times_{k=1}^d \mathrm{fl}(\widetilde{W}_k)\|_F \le \frac{\widetilde{\varepsilon}}{\widetilde{\tau}}((1+\widetilde{\tau})^d - 1).$$

Proof. The error of the approximation, in view of the subadditivity of the Frobenius norm, satisfies

$$\begin{aligned}\|\mathcal{A} - \mathcal{C} \times_{k=1}^d \mathrm{fl}(\widetilde{W}_k)\|_F &= \|\mathcal{A} - \mathcal{A} \times_{k=1}^d \mathrm{fl}(\widetilde{W}_k) Y_k^T\|_F \\ &\le \sum_{k=1}^d \|\mathcal{A} \times_{i=1}^{k-1} \mathrm{fl}(\widetilde{W}_i) Y_i^T - \mathcal{A} \times_{i=1}^{k} \mathrm{fl}(\widetilde{W}_i) Y_i^T\|_F.\end{aligned}$$

The Frobenius norm is invariant under matricization, so it suffices to bound terms of the form $\|(I - \mathrm{fl}(\widetilde{W}_k) Y_k^T) \mathcal{A}_k B_k\|_F$ for $k = 1, \dots, d$, where the matrices B_k are given by $B_k := (I \otimes \cdots \otimes I \otimes \mathrm{fl}(\widetilde{W}_{k-1}) Y_{k-1}^T \otimes \cdots \otimes \mathrm{fl}(\widetilde{W}_1) Y_1^T)^T$. Then,

$$\begin{aligned}\|(I - \mathrm{fl}(\widetilde{W}_k) Y_k^T) \mathcal{A}_k B_k\|_F &= \|(I - \mathrm{fl}(\widetilde{W}_k) Y_k^T)(\mathcal{A}_k + \mathcal{A}_k B_k - \mathcal{A}_k)\|_F \\ &\le \|(I - \mathrm{fl}(\widetilde{W}_k) Y_k^T) \mathcal{A}_k\|_F + \|(I - \mathrm{fl}(\widetilde{W}_k) Y_k^T)(\mathcal{A}_k - \mathcal{A}_k B_k)\|_F \\ &\le \|\mathcal{A} - \mathcal{A} \times_k \mathrm{fl}(\widetilde{W}_k) Y_k^T\|_F + (1 + \|\mathrm{fl}(\widetilde{W}_k)\|_2 \|Y_k^T\|_2) \|\mathcal{A}_k - \mathcal{A}_k B_k\|_F \\ &\le \|\mathcal{A}_k - \mathrm{fl}(\mathcal{A}_k X_k (Y^T \mathcal{A}_k X_k)_\epsilon^\dagger) Y_k^T \mathcal{A}_k\|_F + \widetilde{\tau}_k \|\mathcal{A}_k - \mathcal{A}_k B_k\|_F \\ &\le \widetilde{\varepsilon}_k + \widetilde{\tau}_k \sum_{k=1}^d \|\mathcal{A} \times_{i=1}^{k-1} \mathrm{fl}(\widetilde{W}_i) Y_i^T - \mathcal{A} \times_{i=1}^{k} \mathrm{fl}(\widetilde{W}_i) Y_i^T\|_F.\end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

We obtained a recurrence inequality with the same structure encountered in the proof of Theorem 5.6. Thus, similar steps lead to the sought inequality. ∎

We emphasize that the parameter $\widetilde{\varepsilon}_k = \|\mathcal{A} - \mathcal{A} \times_k \mathrm{fl}(\widetilde{W}_k) Y_k^T\|_F$ is the error of approximation of the SGN method in floating-point arithmetic performed on $\mathcal{A}_k$ with sketch matrices X_k, Y_k and has been extensively analyzed in [25, section 4.1]. The parameter $\widetilde{\tau}_k$ is bounded thanks to Lemma 5.7. To facilitate the understanding of the proof, we set $\widetilde{\tau}_k$ equal to $1 + \|\mathrm{fl}(\widetilde{Q}_k)\|_2 \|Y_k^T\|_2$; however, a sharper estimate may be obtained following the strategies in [25, section 4.1]. Again, the Gaussian hypothesis is not strictly necessary, and any class of random matrices such that the entries are $\mathcal{O}(1)$ and a rectangular realization is well-conditioned would work well.

6. Sketch selection for MLN. An important aspect of MLN is the choice of the sketch matrices. Indeed, in the deterministic bound for MLN (5.5), conditioning terms that depend on the probability distribution of the sketching appear.

If the tensor $\mathcal{A}$ is not structured, several options are available: Gaussian, SRHT, SRFT, and Discrete Cosine Transform (DCT), just to name a few. We recommend the recent survey [22] for an excellent overview of randomized algorithms in numerical linear algebra.

When $\mathcal{A}$ is structured instead, specific DRMs should be used. Of particular interest is the case where the tensor $\mathcal{A}$ is given in Tucker format; that is, $\mathcal{A} = \mathcal{C} \times_{i=1}^{d} U_i$, where $\mathcal{C} \in \mathbb{R}^{k_1 \times \cdots \times k_d}$ is a small tensor and the U_i have size $n_i \times k_i$, and we would compress it to obtain a Tucker tensor of smaller dimensions.

In this case, the flattening along the kth index is given by $U_k \mathcal{C}_k U_{\otimes \check{k}}^T$, and an appropriate right DRM could exploit the Kronecker structure of $U_{\otimes \check{k}}$ to accelerate the sketching procedure. For example, one may choose as sketching matrix a Kronecker product of small independent DRMs $\Omega_i \in \mathbb{R}^{n_i \times r_i}$, r_i, like the ones described for the unstructured case, in order to compute $U_{\otimes \check{k}} \Omega_{\otimes \check{k}}$ in a very cheap way.

This strategy has the drawback that, to obtain a satisfactory approximation of a matrix with rank r with high probability, each Ω_i should have at least r columns for a total of r^d columns. Another possibility, which is what we suggest, is to use a random selection of columns from the Kronecker product of the Ω_i. These types of sketchings have been extensively analyzed in the literature; in particular, in [18], Kronecker Fast Johnson–Lindenstrauss Transforms (KFJLT) are proposed. KFJLTs drastically reduce the embedding cost to an exponential factor of the standard fast Johnson–Lindenstrauss transform (FJLT)'s cost when applied to vectors with Kronecker structure, and, above all, the computational gain comes with only a small price in embedding power. A related strategy is selecting the right DRMs with Khatri–Rao structure, as done in [7] in a similar context.

Our results allow us to analyze the accuracy attained from the approximation methods obtained by a specific choice of sketchings Y_k, X_k in terms of the singular values and the condition number of the matrices $Y_k^T Q_k$ with an orthogonal Q_k (see Theorem 5.5).

7. Experiments. The aim of this section is to illustrate the performances of MLN and SMLN and to show that the theoretical bounds provided in Theorem 5.5 are sharp. The code used for the numerical experiments is available at https://github.com/alb95/MLN.

Implementation. In our implementation of (S)MLN, the core tensor $\mathcal{A} \times_{k=1}^{d} Y_k^T$ and the matrices $\mathcal{A}_k X_k (Y_k^T \mathcal{A}_k X_k)^\dagger$ are computed separately. Regarding the implementation of the pseudoinverse and of the ϵ-truncated pseudoinverse, see [25, section 5].

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Numerical illustration. In the experiments, we will compare the performances of the MLN and SMLN methods with other popular methods for the low-rank tensor approximation in Tucker format. In particular, we will compare the method with the truncated HOSVD, the RHOSVD, and the randomized sequential truncated HOSVD (RSTHOSVD).

All numerical experiments were performed in MATLAB version 2023b on a laptop with 16 GB of system memory, and the tensor operations (mode-k products, unfoldings, and the computation of the HOSVD) have been performed by means of the Tensor Toolbox for MATLAB v3.5 [2] and the `tensorprod` function in MATLAB.

We recall that, in the HOSVD method, the SVD of each of the d matricizations of the tensor $\mathcal{A}$ is computed; i.e., $U_k S_k V_k^T = \mathcal{A}_k$, and then, setting $r = (r_1, \dots, r_d)$ as the desired multilinear rank of the approximant, the tensor $\widehat{\mathcal{A}} = (\mathcal{A} \times_{k=1}^{d} U_k^{(r_k)T}) \times_{k=1}^{d} U_k^{(r_k)}$ is formed, where $U_k^{(r_k)}$ are the first r_k columns of U_k.

The RHOSVD is similar; once the multilinear rank r of the approximant is fixed, we draw d random sketchings X_k of size $\prod_{j \neq k} n_j \times r_k$, and then we compute an economy-size SVD of the matrices $U_k S_k V_k^T = \mathcal{A}_k X_k$. The approximant is given by $\widehat{\mathcal{A}} = (\mathcal{A} \times_{k=1}^{d} U_k^T) \times_{k=1}^{d} U_k$.

The RSTHOSVD differs from previous methods because it truncates the tensor while processing each mode. More specifically, if the target multilinear rank r of the approximant is chosen, d random sketchings X_k of size $(\prod_{j<k} r_j)(\prod_{j>k} n_j) \times r_k$ are drawn. Then, an economy-size SVD of the matrices $U_k S_k V_k^T = \mathcal{B}_k^{(k)} X_k$ is sequentially computed, where $\mathcal{B}^{(k)}$ denotes the partially truncated core tensor $\mathcal{B}^{(k)} = \mathcal{A} \times_{j=1}^{k} U_j^T$. In this case, the approximant is given by $\widehat{\mathcal{A}} = (\mathcal{A} \times_{k=1}^{d} U_k^T) \times_{k=1}^{d} U_k$.

Even if the truncated HOSVD is often prohibitively expensive, it is an important benchmark because it provides an almost-optimal Tucker approximant; i.e., if $\widehat{\mathcal{A}}$ is the truncated HOSVD and $\mathcal{A}_*$ is the optimal solution to the best low multilinear rank approximation problem, then

$$\|\mathcal{A} - \widehat{\mathcal{A}}\|_F \leq \sqrt{d} \|\mathcal{A} - \mathcal{A}_*\|_F.$$

In the majority of the experiment, to describe different decays, we construct the tensors by fixing their CPD. That is, we fix a sequence of n decreasing positive numbers σ_i; we generate d matrices $Q_i \in \mathbb{R}^{n \times n}$, where Q_i is a random orthogonal matrix (Q-factor in the QR factorization of a square Gaussian matrix); and we set $\mathcal{T} = \mathcal{S} \times_{i=1}^{d} Q_i$, where $\mathcal{S}$ is the superdiagonal tensor with the σ_i in the superdiagonal. Notice that, in this way, the singular values of each matricization are the σ_i.

Unless explicitly specified, the sketch matrices are assumed to be Gaussian.

In the first experiment, Figure 1, we compare the performances of MLN and SMLN. To do so, we test the algorithms on a numerically low-rank tensor $\mathcal{T}$ of size $70 \times 70 \times 70$ with exponential decay in the σ_i of rate 0.1 (i.e., $\sigma_i = 0.1^i$).

The tests show that, when there is no oversampling, SMLN performs better than MLN, but both methods produce unsatisfactory results. Instead, even with a small oversampling ($\ell = 3$), both methods show significant improvement and in practice result equivalent. This confirms what was observed in [25] for GN: Stability cannot be established for plain MLN, but oversampling makes its instability benign, and one usually obtains satisfactory results.

Given that both theoretical and experimental results support oversampling's fundamental impact, we recommend its consistent implementation; moreover, due to the equivalence of results achieved with oversampling by MLN and SMLN, we will not include SMLN in the experiments below.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

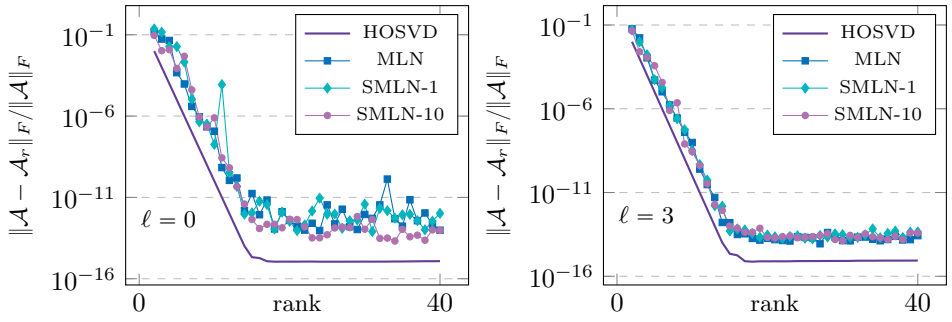

FIG. 1. *Accuracy of MLN and SMLN with* $\epsilon = u\|A\|_F$ *(SMLN-*1*) and* $\epsilon = 10u\|A\|_F$ *(SMLN-*10*).*

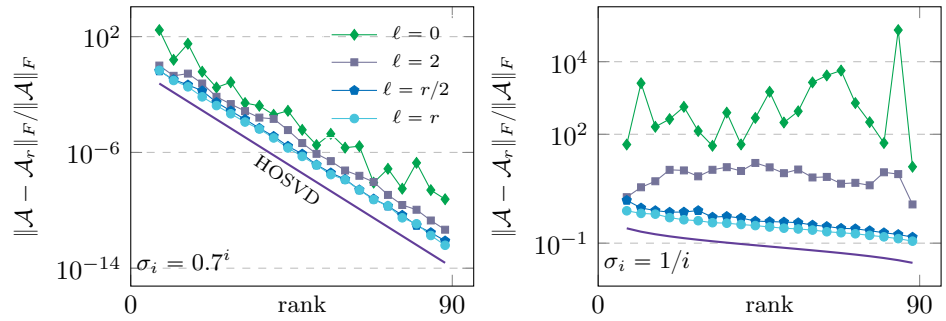

FIG. 2. *Performance comparison of MLN with varying values of the oversampling parameter* ℓ *on two tensors of size* $100 \times 100 \times 100$*: one with* $\sigma_i = 0.7^i$ *(left) and another with* $\sigma_i = 1/i$ *(right)."*

We notice that oversampling, other than stabilizing MLN, serves to improve its accuracy. Hence, we conduct experiments to determine the optimal choice for the oversampling parameter ℓ.

The results are shown in Figure 2. As observed for GN in [25], MLN with fixed ℓ gets further from optimal as r increases, whereas choosing $\ell = cr$ avoids this issue. In particular, the examples show that choosing $c = \frac{1}{2}$ and therefore $\ell = \frac{r}{2}$ yields a robust implementation in all cases.

In the upcoming experiment, Figure 3, we compare the performances of MLN (with $\ell = r/2$), RHOSVD, and HOSVD. We test the algorithms on 4 tensors of size $100 \times 100 \times 100$ with different decays: linear ($\sigma_i = 1/i$), quadratic ($\sigma_i = 1/i^2$), cubic ($\sigma_i = 1/i^3$), and exponential ($\sigma_i = 0.5^i$). The plots show that, up to a small constant, the accuracy of MLN and RHOSVD are the same and that both achieve near-optimal accuracy.

The same occurs when we test the algorithms on 3-dimensional and 4-dimensional Hilbert tensors; see Figure 4. We remark that the Tucker approximations are only practical for such values of d since, for larger values, the storage cost for the core tensor can easily become the bottleneck.

To evaluate the execution time of the MLN method, in the next experiments, we compare it against the RHOSVD and RSTHOSVD. In Figure 5, we show the performances of these 3 methods on 3D tensors of size $100 \times 100 \times 100$ and 4D tensors of size $70 \times 70 \times 70 \times 70$ varying the multilinear rank of the approximants, while in Figure 6, we fix the multilinear rank of the approximants and increase the size of the problem.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

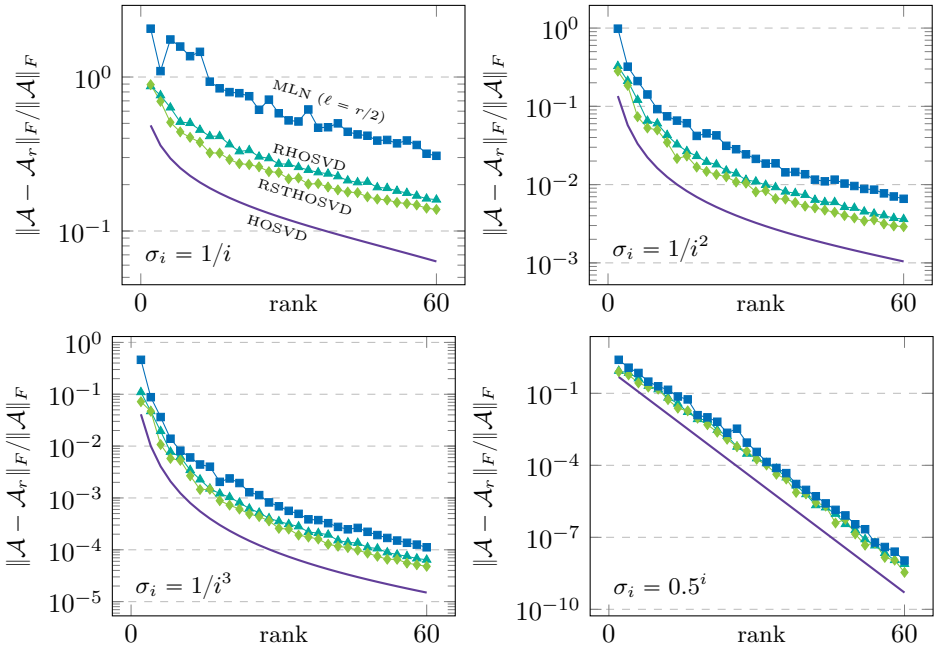

FIG. 3. *Comparison of MLN with oversampling parameter $\ell = r/2$, HOSVD, RHOSVD, and RSTHOSVD on tensors with different decays.*

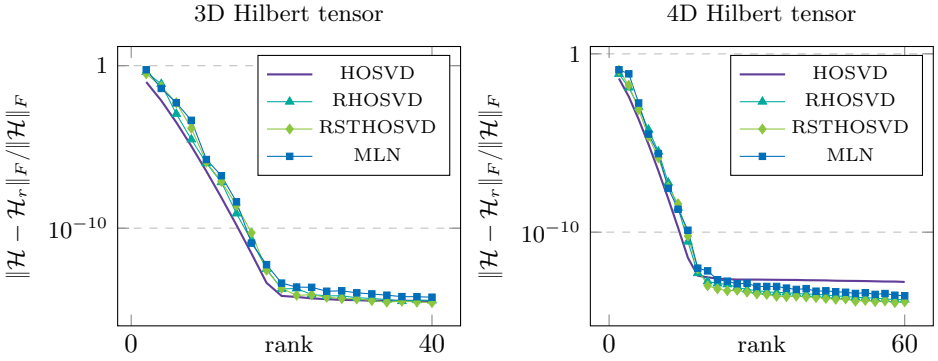

FIG. 4. *Frobenius error of approximation of the 3D Hilbert tensor $\mathcal{H}(i,j,k) = \frac{1}{i+j+k-2}$ (left) and 4D Hilbert tensor $\mathcal{H}(i,j,k,\ell) = \frac{1}{i+j+k+\ell-3}$ (right).*

As we can see, the computing time of MLN is slightly higher than that of RHOSVD and RSTHOSVD. This is because the bulk of the algorithm lies in the sketching procedure for not-structured tensors and MLN requires oversampling.

In the next experiment, we investigate the effect of d on the accuracy of the approximation. In terms of accuracy, our analysis suggests a linear correlation between algorithmic error and the singular values of the tensor's matricizations and an exponential relationship with the norm of the projections $\mathcal{P}_k$, as shown for instance in Theorem 5.8. In particular, since we do not expect the norm of the projectors $\widetilde{\mathcal{P}}_k$ to be influenced by d, the bound would predict an exponential growth of the constant with respect to d, because of the term $(1+\tilde{\tau})^d$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

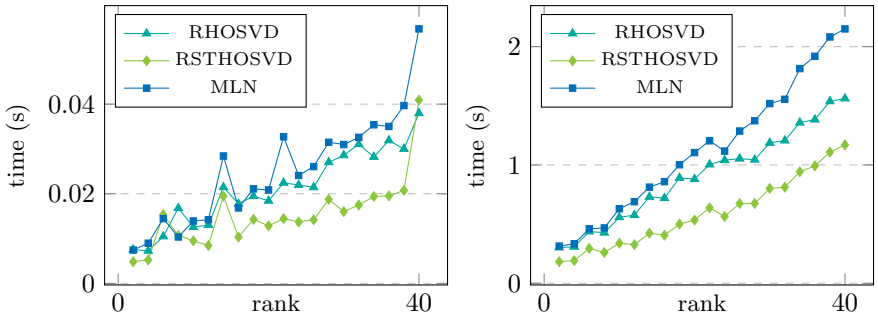

FIG. 5. *Comparison of Tucker approximation methods in terms of computing time on 3D tensors (left) and 4D tensors (right) of fixed size.*

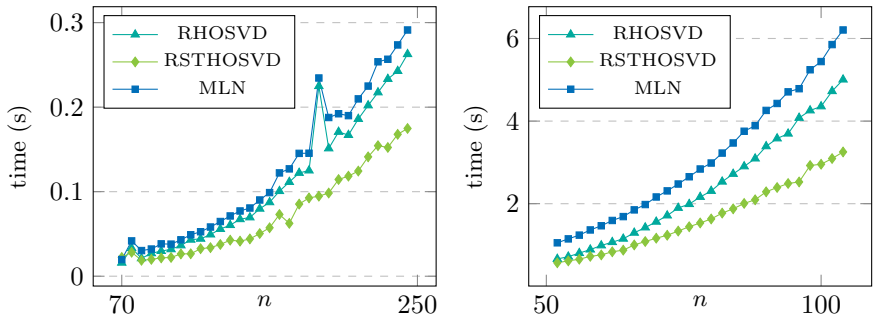

FIG. 6. *Comparison of Tucker approximation methods in terms of computing time on 3D tensors (left) and 4D tensors (right) with a fixed multilinear rank of approximation.*

Figure 7 shows a slight degradation of the quality of the approximation as d increases, but this does not severely impact the performances (in particular, in the case of exponential decay of the σ_i).

In fact, the growth is far from the exponential one predicted by Theorem 5.5. We now perform another experiment aimed at understanding whether Theorem 5.5 is descriptive of the worst-case behavior. To accomplish this, we select non-Gaussian sketching in an unfavorable setting. Specifically, we use SRHT matrices for the sketchings. The tensor $\mathcal{A} = \mathcal{S} \times_{i=1}^{d} Q_i^T$ is constructed as follows: Instead of employing different Haar-distributed orthogonal matrices for the Q_i, each Q_i is set equal to the same 2×2 block diagonal matrix $\mathrm{diag}(I_7, U)$, where I_7 is the 7×7 identity matrix and $U \in \mathbb{R}^{25\times 25}$ is a Haar-distributed orthogonal matrix. These Q_i are known to be a difficult example for SRHT matrices [4, 27]. Regarding $\mathcal{S}$, we use a 4-dimensional superdiagonal tensor with exponential decay in the σ_i of rate 0.3. Note that, by construction, such a tensor is symmetric with respect to each mode. Therefore, setting the SRHT matrices X_i and Y_i equal to the same X and Y, we expect that each projection contributes almost equally to the total error.

We report the numerical results in Table 1 and consistently use the following notation for $k = 1, \ldots, 4$:

$$E_k = \frac{||\mathcal{T} - \mathcal{T} \times_{i=1}^{k} \mathcal{P}_i||_F}{||\mathcal{T} - \mathcal{T}_r||_F}.$$

In view of Theorem 5.5, we would expect that $E_{j+1} \approx (\tau + 1)E_j$ (assuming a worst-case behavior is encountered for all modes). The experiment demonstrates that the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

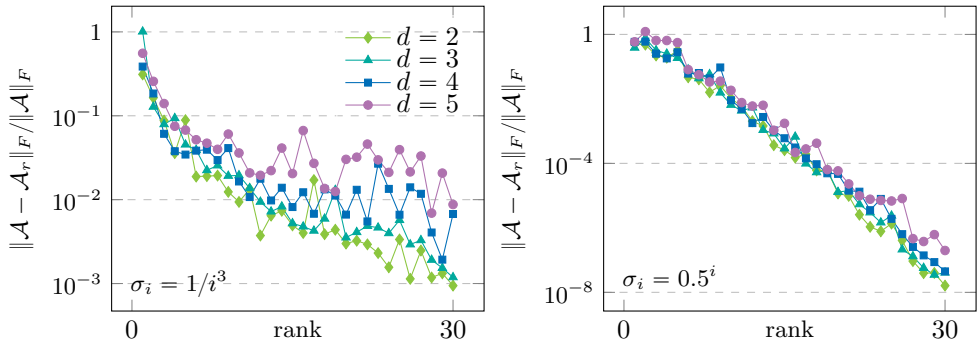

FIG. 7. *MLN tested for tensors of varying dimension $d = 2, 3, 4, 5$ but same decay rate: $\sigma_i = 1/i^3$ (left) and $\sigma_i = 0.5^i$ (right).*

TABLE 1

Frobenius relative error of approximation obtained by increasing the number of oblique projections used for the approximation. Each row of the table represents a different experiment; in this example, the sketchings are chosen in a particularly unfavorable manner in order to trigger the worst-case growth of the errors described in Theorem 5.5.

ρ	τ	$\|\|\mathcal{T}-\mathcal{T}_r\|\|_F$	E_1	E_2	E_3	E_4
1.20×10^{-8}	1.04×10^{2}	4.51×10^{-9}	9.67×10^{0}	4.16×10^{1}	6.38×10^{2}	2.02×10^{4}
1.45×10^{-3}	1.49×10^{3}	4.51×10^{-9}	3.63×10^{7}	6.93×10^{9}	2.45×10^{12}	9.27×10^{14}
1.01×10^{-8}	1.12×10^{7}	4.51×10^{-9}	4.16×10^{6}	2.27×10^{12}	6.12×10^{18}	1.94×10^{25}
1.46×10^{-3}	1.04×10^{2}	4.51×10^{-9}	1.22×10^{6}	4.38×10^{6}	3.66×10^{7}	3.33×10^{8}
2.40×10^{-8}	2.23×10^{3}	4.51×10^{-9}	3.92×10^{3}	5.71×10^{5}	2.98×10^{8}	2.22×10^{11}

TABLE 2

The table represents the ratio of the errors obtained projecting on the first $j+1$ modes and the first j. Each row of the table represents a different experiment; the sketchings are chosen in a particularly unfavorable manner to trigger the worst-case growth of the errors described in Theorem 5.5.

$1+\tau$	E_2/E_1	E_3/E_2	E_4/E_3
1.05×10^{2}	4.33×10^{0}	1.54×10^{1}	3.17×10^{1}
1.49×10^{3}	1.90×10^{2}	3.54×10^{2}	3.78×10^{2}
1.12×10^{7}	5.46×10^{5}	2.72×10^{7}	3.17×10^{6}
1.05×10^{2}	3.61×10^{0}	8.35×10^{0}	9.10×10^{0}
2.23×10^{3}	1.96×10^{2}	5.22×10^{2}	7.45×10^{2}

error growth is indeed exponential in d; in addition, the results in Table 1 show that $1+\tau$ describes the order of magnitude of E_{j+1}/E_j, as expected. To further highlight this, we report in Table 2 the factors $1+\tau$ and the error amplifications E_{j+1}/E_j. We remark that this example is not particularly meaningful from the low-rank approximation perspective; the errors are large and the low-rank approximations obtained of little practical use. We only include it to discuss whether Theorem 5.5 gives an accurate description of the worst-case scenario.

Recall that this is not a limitation in practice; the method and the analysis are only of interest for moderate d, whereas for situations involving high dimensions, we suggest looking for alternatives that completely avoid the curse of dimensionality. Among these alternatives, the streaming tensor train approximation (STTA) [21] emerges as the closest in methodology to ours. It remains grounded in the GN framework and maintains streamability and one-pass capability, yet delivers an approximant in tensor train format.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

8. Conclusions. This paper presents and analyzes MLN, an innovative algorithm for the low-rank compression of a tensor in Tucker format. The method is based on the matrix Nyström method, and in particular on the GN presented in [25].

Two key distinctions set our method apart from existing techniques. First, MLN is a single-pass and streamable algorithm as it requires only two-sided sketchings of the original tensor. Second, the MLN algorithm eliminates the need for costly orthogonalizations because it is based on the GN method for matrices, a crucial advantage over traditional approaches based on the randomized SVD.

In terms of accuracy, the method exhibits only a marginal deviation from the RHOSVD while still maintaining its near-optimal approximation quality. This assertion is reinforced not only by our rigorous theoretical analysis, but also by the results obtained through extensive experiments.

Another crucial aspect of the method is its stability. Even though similar ideas have been proposed in the past (see, for instance, [6]), we propose suitable modifications that can ensure the stability of the methodology; this is attained by carefully handling the pseudoinverses involved in the process. This stability further strengthens the applicability and practicality of our proposed approach. Several research lines remain open and will be investigated in the future. Our work characterizes the quality of the approximants based on the norm of certain matrices (see Theorem 5.5).

In principle, finding a priori probabilistic bounds for such matrices allows us to select the best sketching in any given scenario. While deriving an a priori bound proves to be relatively easy in certain instances (such as when dealing with Gaussian matrices, as discussed in our paper), the complexity of the analysis increases when adopting structured sketching techniques. These techniques, while highly beneficial for structured tensors, present a more intricate analytical challenge. Notably, substantial efforts are already underway in this direction. Another intriguing avenue of research involves expanding the methodology outlined in this study to encompass a broader spectrum of tensor networks. Such an extension could potentially unlock novel insights and applications across a wider range of contexts within the realm of tensor-based computations.

Reproducibility of computational results. This paper has been awarded the "SIAM Reproducibility Badge: Code and data available" as a recognition that the authors have followed reproducibility principles valued by SIMAX and the scientific computing community. Code and data that allow readers to reproduce the results in this paper are available at https://github.com/alb95/MLN and in the supplementary materials (MLN-main.zip [local/web 20KB]).

REFERENCES

[1] S. Ahmadi-Asl, S. Abukhovich, M. G. Asante-Mensah, A. Cichocki, A. H. Phan, T. Tanaka, and I. Oseledets, *Randomized algorithms for computation of Tucker decomposition and higher order SVD (HOSVD)*, IEEE Access, 9 (2021), pp. 28684–28706, https://doi.org/10.1109/ACCESS.2021.3058103.

[2] B. W. Bader and T. G. Kolda, *Algorithm 862: MATLAB tensor classes for fast algorithm prototyping*, ACM Trans. Math. Software, 32 (2006), pp. 635–653, https://doi.org/10.1145/1186785.1186794.

[3] R. E. Bellman, *Adaptive Control Processes: A Guided Tour*, Princeton University Press, Princeton, NJ, 1961.

[4] C. Boutsidis and A. Gittens, *Improved matrix algorithms via the subsampled randomized Hadamard transform*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 1301–1340, https://doi.org/10.1137/120874540.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[5] C. F. Caiafa and A. Cichocki, *Generalizing the column–row matrix decomposition to multi-way arrays*, Linear Algebra Appl., 433 (2010), pp. 557–573, https://doi.org/10.1016/j.laa.2010.03.020.

[6] C. F. Caiafa and A. Cichocki, *Stable, robust, and super fast reconstruction of tensors using multi-way projections*, IEEE Trans. Signal Process., 63 (2015), pp. 780–793, https://doi.org/10.1109/TSP.2014.2385040.

[7] M. Che and Y. Wei, *Randomized algorithms for the approximations of Tucker and the tensor train decompositions*, Adv. Comput. Math., 45 (2019), pp. 395–428, https://doi.org/10.1007/s10444-018-9622-8.

[8] M. Che, Y. Wei, and H. Yan, *The computation of low multilinear rank approximations of tensors via power scheme and random projection*, SIAM J. Matrix Anal. Appl., 41 (2020), pp. 605–636, https://doi.org/10.1137/19M1237016.

[9] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278, https://doi.org/10.1137/S0895479896305696.

[10] W. Dong, G. Yu, L. Qi, and X. Cai, *Practical sketching algorithms for low-rank Tucker approximation of large tensors*, J. Sci. Comput., 95 (2023), 52, https://doi.org/10.1007/s10915-023-02172-y.

[11] P. Drineas and M. W. Mahoney, *A randomized algorithm for a tensor-based generalization of the singular value decomposition*, Linear Algebra Appl., 420 (2007), pp. 553–571, https://doi.org/10.1016/j.laa.2006.08.023.

[12] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[13] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin, *A theory of pseudoskeleton approximations*, Linear Algebra Appl., 261 (1997), pp. 1–21, https://doi.org/10.1016/S0024-3795(96)00301-1.

[14] L. Grasedyck, *Hierarchical singular value decomposition of tensors*, SIAM J. Matrix Anal. Appl., 10 (2009), pp. 2029–2054, https://doi.org/10.1137/090764189.

[15] L. Grasedyck, D. Kressner, and C. Tobler, *A literature survey of low-rank tensor approximation techniques*, GAMM-Mitt., 36 (2013), pp. 53–78, https://doi.org/10.1002/gamm.201310004.

[16] N. Halko, P. G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288, https://doi.org/10.1137/090771806.

[17] F. L. Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, J. Math. Phys., 6 (1927), pp. 164–189, https://doi.org/10.1002/sapm192761164.

[18] R. Jin, T. G. Kolda, and R. Ward, *Faster Johnson–Lindenstrauss transforms via Kronecker products*, Inf. Inference, 10 (2021), pp. 1533–1562, https://doi.org/10.1093/imaiai/iaaa028.

[19] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), pp. 455–500, https://doi.org/10.1137/07070111X.

[20] D. Kressner and B. Plestenjak, *Analysis of a Class of Randomized Numerical Methods for Singular Matrix Pencils*, preprint, arXiv:2305.13118, 2023.

[21] D. Kressner, B. Vandereycken, and R. Voorhaar, *Streaming Tensor Train Approximation*, preprint, arXiv:2208.02600, 2022.

[22] P.-G. Martinsson and J. A. Tropp, *Randomized numerical linear algebra: Foundations and algorithms*, Acta Numer., 29 (2020), pp. 403–572, https://doi.org/10.1017/s0962492920000021.

[23] R. Minster, Z. Li, and G. Ballard, *Parallel Randomized Tucker Decomposition Algorithms*, preprint, arXiv:2211.13028, 2022.

[24] R. Minster, A. K. Saibaba, and M. E. Kilmer, *Randomized algorithms for low-rank tensor decompositions in the Tucker format*, SIAM J. Math. Data Sci., 2 (2020), pp. 189–215, https://doi.org/10.1137/19M1261043.

[25] Y. Nakatsukasa, *Fast and Stable Randomized Low-Rank Matrix Approximation*, preprint, arXiv:2009.11392, 2020.

[26] Y. Nakatsukasa and N. J. Higham, *Backward stability of iterations for computing the polar decomposition*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 460–479, https://doi.org/10.1137/110857544.

[27] Y. Nakatsukasa and T. Park, *Randomized low-rank approximation for symmetric indefinite matrices*, SIAM J. Matrix Anal. Appl., 44 (2023), pp. 1370–1392, https://doi.org/10.1137/22M1538648.

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[28] Y. Nakatsukasa and J. A. Tropp, *Fast & Accurate Randomized Algorithms for Linear Systems and Eigenvalue Problems*, preprint, arXiv:2111.00113, 2021.
[29] I. V. Oseledets, *Tensor-train decomposition*, SIAM J. Sci. Comput., 33 (2011), pp. 2295–2317, https://doi.org/10.1137/090752286.
[30] I. V. Oseledets, D. Savostianov, and E. E. Tyrtyshnikov, *Tucker dimensionality reduction of three-dimensional arrays in linear time*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 939–956, https://doi.org/10.1137/060655894.
[31] V. Rokhlin and M. Tygert, *A fast randomized algorithm for overdetermined linear least-squares regression*, Proc. Natl. Acad. Sci. USA, 105 (2008), pp. 13212–13217, https://doi.org/10.1073/pnas.0804869105.
[32] A. K. Saibaba, *HOID: Higher order interpolatory decomposition for tensors based on Tucker representation*, SIAM J. Matrix Anal. Appl., 37 (2016), pp. 1223–1249, https://doi.org/10.1137/15M1048628.
[33] Y. Sun, Y. Guo, C. Luo, J. Tropp, and M. Udell, *Low-rank Tucker approximation of a tensor from streaming data*, SIAM J. Math. Data Sci., 2 (2020), pp. 1123–1150, https://doi.org/10.1137/19M1257718.
[34] J. A. Tropp, *Improved analysis of the subsampled randomized Hadamard transform*, Adv. Adapt. Data Anal., 3 (2011), pp. 115–126, https://doi.org/10.1142/S1793536911000787.
[35] J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher, *Practical sketching algorithms for low-rank matrix approximation*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1454–1485, https://doi.org/10.1137/17M1111590.
[36] N. Vannieuwenhoven, R. Vandebril, and K. Meerbergen, *A new truncation strategy for the higher-order singular value decomposition*, SIAM J. Sci. Comput., 34 (2012), pp. A1027–A1052, https://doi.org/10.1137/110836067.
[37] G. Zhou, A. Cichocki, and S. Xie, *Decomposition of Big Tensors with Low Multilinear Rank*, preprint, arXiv:1412.1885, 2014.

Downloaded 10/28/25 to 146.48.83.26 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.