Tensor Train Completion from Fiberwise Observations Along a Single Mode

Shakir Showkat Sofi[1,2*] and Lieven De Lathauwer[1,2*]

[1]Group Science, Engineering and Technology, KU Leuven Kulak, Kortrijk, Belgium.

[2]Dept. Electrical Engineering (ESAT), KU Leuven, Leuven, Belgium.

*Corresponding author(s). E-mail(s): shakirshowkat.sofi@kuleuven.be; lieven.delathauwer@kuleuven.be;

Abstract

Tensor completion is an extension of matrix completion aimed at recovering a multiway data tensor by leveraging a given subset of its entries (observations) and the pattern of observation. The low-rank assumption is key in establishing a relationship between the observed and unobserved entries of the tensor. The low-rank tensor completion problem is typically solved using numerical optimization techniques, where the rank information is used either implicitly (in the rank minimization approach) or explicitly (in the error minimization approach). Current theories concerning these techniques often study probabilistic recovery guarantees under conditions such as random uniform observations and incoherence requirements. However, if an observation pattern exhibits some low-rank structure that can be exploited, more efficient algorithms with deterministic recovery guarantees can be designed by leveraging this structure. This work shows how to use only standard linear algebra operations to compute the tensor train decomposition of a specific type of "fiber-wise" observed tensor, where some of the fibers of a tensor (along a single specific mode) are either fully observed or entirely missing, unlike the usual entry-wise observations. From an application viewpoint, this setting is relevant when it is easier to sample or collect a multiway data tensor along a specific mode (e.g., temporal). The proposed completion method is fast and is guaranteed to work under reasonable deterministic conditions on the observation pattern. Through numerical experiments, we showcase interesting applications and use cases that illustrate the effectiveness of the proposed approach[†]

Keywords: low-rank approximation, tensor train decomposition, completion, fiber-wise, subspace techniques

---

[†]This paper is an extended version of our paper published in EUSIPCO 2024 [1]. See the contribution statement for the new additions.

1 Introduction

We live in a data-driven world where massive amounts of data are generated and collected daily. In addition to the huge volume and high velocity, the structural complexity of the data is becoming so high that it renders standard techniques inadequate. Multidimensional arrays (tensors) are data structures that offer a better way to organize and analyze the data with a higher-order structure. As the number of dimensions of the data increases, the memory required to store it (and the computational effort required for its analysis) increases exponentially—an obstacle known as the "curse of dimensionality." Tensor decompositions offer efficient approaches to analyzing higher-order datasets, allowing for the retention of intrinsic information within the data while taming the curse of dimensionality [2–11].

Real-world datasets are often noisy and incomplete for various reasons, including sensor malfunctions, recording errors, constraints imposed by privacy regulations, delays in obtaining access permissions, and deliberate incomplete sampling to meet memory and time requirements. Analyzing incomplete datasets is a big challenge, as the missing information may affect the accuracy and reliability of the findings and, therefore, limit subsequent applications [12]. *Completion* of a partially observed dataset may be thought of as a particular specification of its unobserved entries. Completion is more crucial for higher-order datasets as they are larger, increasing the chance of missing or unreliable entries. Many problems can be framed as instances of tensor completion, e.g., image and video inpainting, gene expression imputation, weather and traffic data imputation (see, e.g., [13–18]). Without any constraints on the completed tensor, there are infinitely many ways to specify the missing entries. Therefore, to make the estimation meaningful, it is necessary to assume that the completed tensor satisfies specific properties (e.g., low rank, minimum volume) — that lower the degrees of freedom and enable a unique solution. These properties constrain the unobserved entries and help establish their relationships with the observations. The principle of parsimony (Occam's razor) serves as a (heuristic) guideline for model selection, stating that if multiple competing models explain the same data, the model with lower complexity is the best. The rank can be considered a kind of natural measure of the complexity; therefore, *low rankness* is a reasonable perspective for promoting parsimonious models with only a few parameters explaining the data.

The literature outlines two approaches to employing low-rank constraints in tensor completion: the rank minimization method (implicit method) and the error minimization method (explicit method). In the former approach, rank is used implicitly as an optimization objective to minimize, with observed entries as constraints to be satisfied. As the rank function is non-convex and NP-hard [19, 20], convex surrogates of the rank function are used to relax this problem, allowing for an approximate but efficient solution. In the latter approach, a hypothesis (tensor decomposition) model is explicitly imposed on a partially observed tensor with a specific, fixed low rank. The loss function—typically continuous and differentiable—is minimized to fit the model parameters (see, e.g., [13–15, 21–24]). In this case, the flexibility in choosing a differentiable objective function enables the use of gradient-based optimization approaches (e.g., first- and/or second-order methods). Moreover, this approach explicitly uses rank information, which can be meaningful in applications where the rank has a physical

significance; therefore, tweaking it may not be allowed. See [11] for an overview of matrix and tensor completion approaches.

Tensor completion problems are typically solved by numerical optimization algorithms (see, e.g., [13–15, 17, 23–25]). An essential aspect of a reliable completion algorithm is its recovery guarantees—specifically, the conditions under which it can uniquely recover unobserved entries from partial observations. Existing theories generally study probabilistic guarantees for recovery based on conditions such as entries being observed uniformly at random and satisfying incoherence requirements [21, 22, 26]. However, if an observation pattern has some structure, better and faster algorithms with deterministic guarantees can be designed by exploiting the structure. Note that, depending on the application, the observation pattern may be structured rather than random; it may even be fixed, for instance, when an incomplete dataset is given "as such", without the possibility of acquiring more entries. There are many interesting problems where conditions like entries being observed uniformly at random may not be explicitly satisfied. In this work, we discuss one such interesting observation pattern where fibers of a tensor (along a single specific mode) are either fully observed or entirely missing, unlike the usual entry-wise observations. This observation pattern is interesting because: (i) it occurs in many real-life applications; (ii) it makes an intriguing distinction in the uniqueness of the completion between matrix and tensor settings. Specifically, if some fibers (rows or columns) are entirely missing in a matrix, the completion problem becomes underdetermined, whereas completion is still possible in a higher-order tensor, even if some fibers are completely missing along a specific mode (see, e.g., [27–30]). In fact, there are many applications where it is easier to collect data (or sample a multivariate function) along one mode (variable) than along others. For example, consider weather time series (e.g., temperature and humidity) collected across various locations [31]. Think of collecting the data (temperature, latitude, latitude) at specific combinations of geospatial locations. Another example of a fiber-wise observation pattern arises when recording traffic speed data (road segment, day, time window) for specific combinations of road segments and days (see, e.g., [27] where this particular dataset is studied). Other examples include chemical reaction data with "time" and "concentration" modes. Obtaining samples along the temporal mode may be easier than varying the concentrations (which may require conducting new experiments).

Algebraic algorithms exploit the low-rank structure in a specific algebraic manner to design completion algorithms that rely solely on standard numerical linear algebra (NLA) techniques. These algorithms are fast and are guaranteed to work under reasonable deterministic conditions on observation pattern. In this line, a novel algebraic method for fitting a low-rank matrix to a matrix with missing entries was proposed [30, 32, 33]. More detailed results on low-rank matrix completion, with extensive discussions on recovery guarantees, have also been studied (see, e.g., [30, 34, 35]). Previous studies have shown that the canonical polyadic decomposition (CPD) and multilinear singular value decomposition (MLSVD) of an incomplete tensor, observed along a single mode, can be computed using only standard NLA by exploiting the fiber-wise observation pattern [28–30]. As big data becomes more prevalent, the need for stable

and scalable algorithms has become more pressing. The tensor train (TT) decomposition is stable, like the MLSVD and breaks the curse of dimensionality, like the CPD, since it has asymptotically the same number of parameters [2, 36]. In light of these advantages, we presented some of the results, including an extension of the algebraic algorithm to the TT format [1]. Note that there is an important difference with the popular technique of (TT) cross approximation, in the sense that the latter obtains TT decomposition from a subset of fibers sampled across *all* modes [37]. In our method, fibers are sampled along a *single* mode.

1.1 Contributions

In this paper, we present more detailed results regarding our approach, with the following main additions:

- We provide more insights into piecewise subspace learning, specifically the conditions for determining the column space of a low-rank matrix, where only some pieces (submatrices) are observed.
- We include the subspace intersection approach, in addition to the subspace constraint approach, for computing the column space of a low-rank matrix from an *informationally complete* set of observed submatrices.
- We utilize the piecewise subspace learning approaches to compute TT approximation using only standard NLA operations.
- Convincing numerical experiments have been included to show that the proposed method is practically fast and reliable.
- In line with [38], we show that the TT decomposition obtained through algebraic completion can serve as a "proxy" for efficient subsequent computations (e.g., a constrained CPD is fitted to the TT approximation rather than to the actual tensor).

1.2 Preliminaries and notation

We use lower-case, bold lower-case, bold capital, and calligraphic letters to denote scalars, vectors, matrices, and tensors, respectively, i.e., $x, \mathbf{x}, \mathbf{X}$, and $\mathcal{X}$, respectively. The *order* of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is the number of its *modes* or *ways*. Matlab-like indexing is used to specify a particular part of a tensor, employing commas, colons, and semicolons. For instance, the specific entry of the Nth-order tensor is denoted by $x_{i_1 i_2 \cdots i_N} = \mathcal{X}(i_1, i_2, \ldots, i_N)$ with $i_n \in \{1, 2, 3, \ldots, I_n\}$ for $n \in \{1, 2, 3, \ldots, N\}$. A mode-$n$ *fiber* of the Nth-order tensor is obtained by fixing every index except the nth. For a third-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, the mode-1 fibers $\mathbf{x}_{:i_2 i_3}$, mode-2 fibers $\mathbf{x}_{i_1 : i_3}$, and mode-3 fibers $\mathbf{x}_{i_1 i_2 :}$ are also known as column, row, and tube fibers, respectively. Similarly, the mode-1 slices $\mathbf{X}_{i_1 ::}$, mode-2 slices $\mathbf{X}_{:i_2 :}$, and mode-3 slices $\mathbf{X}_{::i_3}$, are also known as horizontal, lateral, and frontal slices, respectively. A permutation and reshaping of a tensor are reflected in the ordering of its indices and the position of the semicolon (;), where the semicolon indicates a new mode, and the ordering of indices indicates the order in which the entries are stacked. In the nth matrix unfolding that is denoted by $\mathbf{X}_{[1,\ldots,n;n+1,\ldots,N]} \in \mathbb{R}^{I_1 \cdots I_n \times I_{n+1} \cdots I_N}$, the first n indices enumerate rows and the remaining indices enumerate columns. The rank, column space (also called the range) and kernel of a matrix $\mathbf{X}$ are denoted by $\operatorname{rank}(\mathbf{X})$, $\operatorname{col}(\mathbf{X})$, $\ker(\mathbf{X})$, respectively.

The dimension of subspace S is denoted by $\dim(S)$. Let $\mathbf{e}_i^{(I)} \in \{0,1\}^I$ denote a vector with a unit entry at index i and zeros elsewhere. We denote a set by a lowercase Greek letter. The cardinality of a set α is denoted by $|\alpha|$. The union and intersection of a sequence of sets are denoted by $\cup_{l=1}^{L}\alpha_l$ and $\cap_{l=1}^{L}\alpha_l$, respectively.

Furthermore, we introduce the following notation to represent a submatrix of a matrix $\mathbf{X} \in \mathbb{R}^{J\times K}$. Let $\alpha_l = \{j_1, \ldots, j_{J_l}\} \subseteq \{1, \ldots, J\}$ and $\beta_l = \{k_1, \ldots, k_{K_l}\} \subseteq \{1, \ldots, K\}$ be the indices of the J_l selected rows and K_l selected columns, respectively. We define the corresponding row and column selection matrices as $\mathbf{S}_r^{(l)} = \left[\mathbf{e}_{j_1}^{(J)} \ \ldots \ \mathbf{e}_{j_{J_l}}^{(J)}\right]^\top \in \{0,1\}^{J_l\times J}$ and $\mathbf{S}_c^{(l)} = \left[\mathbf{e}_{k_1}^{(K)} \ \ldots \ \mathbf{e}_{k_{K_l}}^{(K)}\right] \in \{0,1\}^{K\times K_l}$, respectively. The matrix $\mathbf{S}_r^{(l)}$ selects the J_l rows indexed by α_l upon left multiplication, and $\mathbf{S}_c^{(l)}$ selects the K_l columns indexed by β_l upon right multiplication. For example, the submatrix $\mathbf{X}_{obs}^{(l)} = \mathbf{S}_r^{(l)}\mathbf{X}\mathbf{S}_c^{(l)} \in \mathbb{R}^{J_l\times K_l}$ contains elements of $\mathbf{X}$ observed at $(j,k) \in \alpha_l \times \beta_l$.

Definition 1 (Isorank submatrix) A submatrix $\mathbf{X}_{obs}^{(l)}$ of a matrix $\mathbf{X}$ is called an isorank submatrix or rank-preserving submatrix iff:

$$\operatorname{rank}\left(\mathbf{X}_{obs}^{(l)}\right) = \operatorname{rank}\left(\mathbf{X}\right),$$

where $\mathbf{X}_{obs}^{(l)} = \mathbf{S}_r^{(l)}\mathbf{X}\mathbf{S}_c^{(l)}$. Here, $\mathbf{S}_r^{(l)}$ and $\mathbf{S}_c^{(l)}$ are row and column selection matrices defined by sets α_l and β_l, respectively, such that $\min(|\alpha_l|, |\beta_l|) \geq \operatorname{rank}(\mathbf{X})$.

Definition 2 (Row overlap) Two submatrices $\mathbf{X}_{obs}^{(1)}$ and $\mathbf{X}_{obs}^{(2)}$ of a matrix $\mathbf{X}$, with the corresponding row selection index sets α_1 and α_2, are said to have a row overlap of size k if there exist k indices that are common to both α_1 and α_2, i.e., $|\alpha_1 \cap \alpha_2| = k$.

Definition 3 (Contraction) The product between the last mode of tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$ and the first mode of tensor $\mathcal{Y} \in \mathbb{R}^{J_1\times J_2\times\ldots\times J_M}$, where $I_N = J_1 = K$, yields an $(N+M-2)$-order tensor $\mathcal{Z} = \mathcal{X} \bullet \mathcal{Y}$, the elements of which are given by:

$$z_{i_1\ldots i_{N-1}j_2\ldots j_M} = \sum_{k=1}^{K} x_{i_1\ldots i_{N-1}k} y_{kj_2\ldots j_M}.$$

1.3 Organization

In Section 2, we provide a brief overview of the TT decomposition of a fully observed tensor. Following that, in Section 3, we outline our method for obtaining the TT decomposition of a tensor observed fiber-wise along a single mode. In Sections 3.2 and 3.3, we discuss the algorithm and uniqueness conditions, respectively. Finally, we present convincing numerical experiments in Section 4 and a brief conclusion in Section 5.

2 TT decomposition of fully observed tensor

A TT decomposition of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ corresponds to a contraction of a sequence of third-order core tensors (TT cores) $\mathcal{G}^{(n)} \in \mathbb{R}^{R_{n-1} \times I_n \times R_n} (1 \leq n \leq N)$ with $R_0 = R_N = 1$ such that each entry of $\mathcal{X}$ can be expressed as the sequence of matrix products [36, 39]:

$$\mathcal{X} = \mathcal{G}^{(1)} \bullet \mathcal{G}^{(2)} \bullet \cdots \bullet \mathcal{G}^{(N-1)} \bullet \mathcal{G}^{(N)}, \tag{1}$$

or entry-wise, we can write:

$$x_{i_1 i_2 \cdots i_{N-1} i_N} = \mathbf{G}^{(1)}_{:i_1:} \mathbf{G}^{(2)}_{:i_2:} \cdots \mathbf{G}^{(N-1)}_{:i_{N-1}:} \mathbf{G}^{(N)}_{:i_N:}, \tag{2}$$

where the matrix $\mathbf{G}^{(n)}_{:i:} \in \mathbb{R}^{R_{n-1} \times R_n}$ is the ith mode-2 slice of the core tensor $\mathcal{G}^{(n)}$. The tuple of minimal integers $(R_0, R_1, \ldots, R_N)$ for which equality in Equations (1) and (2) holds is the TT rank of $\mathcal{X}$, denoted by $\text{rank}_{TT}(\mathcal{X})$. The dense tensor $\mathcal{X}$ has a total storage complexity of $\prod_{n=1}^{N} I_n$, whereas in the TT format, the storage complexity is $\sum_{n=1}^{N} R_{n-1} I_n R_n$. Hence, a low TT rank can greatly reduce storage complexity. A visualization of the TT decomposition of a fifth-order tensor is shown in Figure 1.

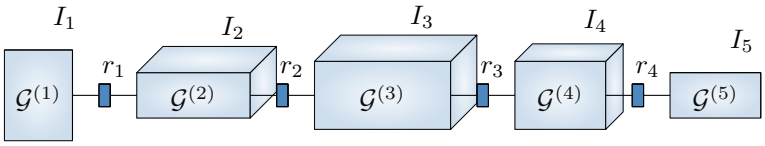


Fig. 1: TT decomposition of 5th order tensor as a train of five core tensors, where $\mathcal{G}^{(1)} \in \mathbb{R}^{1 \times I_1 \times r_1}$ and $\mathcal{G}^{(5)} \in \mathbb{R}^{r_4 \times I_5 \times 1}$.

Additionally, we introduce a shorthand notation for TT decomposition using partial products. A tensor $\mathcal{X}$ can be represented as:

$$\mathcal{X} = \mathcal{G}^{(<n)} \bullet \mathcal{G}^{(n:m)} \bullet \mathcal{G}^{(>m)}, \quad 1 \leq n \leq m \leq N, \tag{3}$$

where the left partial product $\mathcal{G}^{(<n)} = \mathcal{G}^{(1)} \bullet \cdots \bullet \mathcal{G}^{(n-1)}$, the right partial product $\mathcal{G}^{(>m)} = \mathcal{G}^{(m+1)} \bullet \cdots \bullet \mathcal{G}^{(N)}$ and $\mathcal{G}^{(n:m)} = \mathcal{G}^{(n)} \bullet \cdots \bullet \mathcal{G}^{(m)}$.

TT decomposition of a given tensor $\mathcal{X}$ is computed by a sequence of truncated SVDs (TT-SVD) [36]. The TT-SVD algorithm sequentially computes TT cores and essentially alternates between the SVD computation step and projection step, which makes it difficult to parallelize. Inspired by the natural parallelizability of MLSVD [40], a similar method was proposed that computes the orthonormal basis for the column spaces of all matrix unfoldings independently and then utilizes them to compute the TT cores (Parallel-TTSVD) [41]. It should be noted that independently computing the orthonormal basis for the column spaces of each matrix unfolding is more expensive than the basis computations in TT-SVD, where the projection step reduces the complexity of subsequent computations. One can reduce this computational burden by replacing SVDs with randomized SVDs [42, 43]. The algorithm for computing

the TT decomposition with parallel SVDs (Parallel-TTSVD) is shown in Algorithm 1 (for more in-depth details, refer to [41, Algorithm 3.1]).

Algorithm 1: Parallel-TTSVD

Input: A tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, and TT rank $(R_0, \ldots, R_N)$
Output: TT cores $\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(N)}$

/*compute orthonormal bases for the ranges of the nth unfoldings*/
for $1 \leq n \leq N-1$ **do**
  Compute $\mathbf{U}^{(n)}, \mathbf{\Sigma}^{(n)}, \mathbf{V}^{(n)\top} = \texttt{svd}\left(\mathbf{X}_{[1,\ldots,n;n+1,\ldots,N]}, R_n\right)$
  Set $\mathbf{A}^{(n)} = \mathbf{U}^{(n)}$, and $\mathbf{B}^{(n)} = \mathbf{\Sigma}^{(n)}\mathbf{V}^{(n)\top}$
end
/*compute the TT cores*/
Set $\mathcal{G}^{(1)} = \texttt{reshape}\left(\mathbf{A}^{(1)}, [1, I_1, R_1]\right)$
for $1 \leq n \leq N-2$ **do**
  Compute $\mathbf{W}^{(n)} = \mathbf{A}^{(n)\top}\left(\mathbf{A}^{(n+1)}\right)_{[1,\ldots,n;n+1,(n+1)+1]}$
  Set $\mathcal{G}^{(n+1)} = \texttt{reshape}\left(\mathbf{W}^{(n)}, [R_n, I_{n+1}, R_{n+1}]\right)$
end
Set $\mathcal{G}^{(N)} = \texttt{reshape}\left(\mathbf{B}^{(N-1)\top}, [R_{N-1}, I_N, 1]\right)$

3 Tensor train completion of a tensor observed along one mode

This section introduces an algebraic algorithm analogous to the Parallel-TTSVD to compute the TT decomposition of an incomplete tensor with fibers observed along a single mode. Assume a given tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ with $\text{rank}_{TT}(\mathcal{X}) = (R_0, \ldots, R_N)$ of which only some of the mode-N (last mode) fibers are observed. One essential step to compute the TT cores is to find orthonormal bases for the ranges of the matrix unfoldings. However, since the matrix unfoldings have missing entries, it is not directly possible to utilize, e.g., SVD or QR factorization to obtain the ranges. Note that under fiber-wise observations, two types of observation patterns arise in matrix unfoldings. In the $(N-1)$th matrix unfolding, i.e., $\mathbf{X}_{[1,\ldots,N-1;N]}$, the rows are either fully observed or entirely missing. If the number of observed rows is greater than or equal to R_{N-1}, we can generically[1] obtain the last TT core ($\mathcal{G}^{(N)} \in \mathbb{R}^{R_{N-1} \times I_N \times 1}$) by computing the top R_{N-1} right-singular vectors of the matrix formed by the observed rows. The other matrix unfoldings can be seen as the horizontal stack of mode-2 slices $\tilde{\mathbf{X}}_{:i:}$ of a third-order reshaping $\tilde{\mathcal{X}} = \mathcal{X}_{[1,\ldots,n;n+1,\ldots,N-1;N]} \in \mathbb{R}^{\prod_{i=1}^{n} I_i \times \prod_{i=n+1}^{N-1} I_i \times I_N}$ of the tensor $\mathcal{X}$. That is,

$$\mathbf{X}_{[1,\ldots,n;n+1,\ldots,N]} = \left[\tilde{\mathbf{X}}_{:1:} \ \cdots \ \tilde{\mathbf{X}}_{:\prod_{i=n+1}^{N-1} I_i:}\right]. \tag{4}$$

[1] Here, "generic" refers to the property that holds with probability 1 when the entries of the matrix are drawn from a continuous distribution.

In this case, the rows of the lateral slices (submatrices of the unfolding) of $\tilde{\mathcal{X}}$ are either fully observed or entirely missing. However, different slices may have observed rows at different row indices. A visual representation of such a matrix unfolding is shown in Figure 2.



Fig. 2: The nth matrix unfolding of a tensor observed through fibers along a single mode is characterized by a structured observation pattern.

The following section will discuss how to determine the orthonormal basis for the column space of such a partially observed matrix unfolding.

3.1 Piecewise subspace learning

In this section, we will discuss how to determine the overall column space of a low-rank matrix, of which only some pieces (submatrices) are observed. Moreover, we will characterize the sampling of pieces such that the overall subspace is guaranteed to be unique (i.e., the subspace is identifiable). Subspace identifiability is closely related to the matrix completion problem. The conditions required to identify the subspace (e.g., column space) are necessary but not sufficient for matrix completion [30, 44]. We will first explore conditions for unique matrix completion, a problem widely studied in the literature, and then discuss how these conditions are relaxed in the context of subspace identification.

For simplicity, we use $\mathbf{M} \in \mathbb{R}^{J \times K}$ to denote a rank-R matrix unfolding of which we need to determine the R-dimensional column space. In the literature, unique minimal rank completions for various types of observation patterns have been studied (see, e.g., [45–47]). This work will focus on an observation pattern shown in Figure 2. Before proceeding to the general case, we will begin with a simple matrix formed by stacking two slices of a tensor, as explained next. Denote the observed submatrices of $\mathbf{M}$ by $\mathbf{M}_{obs}^{(1)} \in R^{(J_1+J_2)\times K_1}$ and $\mathbf{M}_{obs}^{(2)} \in R^{(J_2+J_3)\times K_2}$ that have J_2 overlapping rows, and $J = J_1 + J_2 + J_3$ and $K = K_1 + K_2$. While the overlap can occur anywhere among the rows, let us assume, without loss of generality, that the row overlap occurs in the middle block, as shown below:

$$\mathbf{M} = \left[\begin{array}{c|c} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \\ \mathbf{M}_{33} & \mathbf{M}_{32} \end{array}\right], \quad \mathbf{M}_{obs}^{(1)} = \begin{bmatrix} \mathbf{M}_{11} \\ \mathbf{M}_{21} \end{bmatrix} \text{ and } \mathbf{M}_{obs}^{(2)} = \begin{bmatrix} \mathbf{M}_{22} \\ \mathbf{M}_{32} \end{bmatrix}. \tag{5}$$

It has been proven that there is a unique rank-R completion for $\mathbf{M}$ if and only if the following condition holds (see, e.g., Corollary 2.3 in [45]):

$$\operatorname{rank}\left(\mathbf{M}_{obs}^{(1)}\right) = \operatorname{rank}\left(\mathbf{M}_{21}\right) = \operatorname{rank}\left([\mathbf{M}_{21}\ \mathbf{M}_{22}]\right) = \operatorname{rank}\left(\mathbf{M}_{22}\right) = \operatorname{rank}\left(\mathbf{M}_{obs}^{(2)}\right) = R. \tag{6}$$

As noted above, subspace identifiability is less restrictive than matrix completion. To illustrate, consider an incomplete matrix formed by concatenating a matrix $\mathbf{M}$ that satisfies condition (6), and a vector $\mathbf{z}$. Let this concatenated matrix be denoted as $\hat{\mathbf{M}} = [\mathbf{M} \mid \mathbf{z}]$, and assume that $\operatorname{rank}\left(\hat{\mathbf{M}}\right) = \operatorname{rank}\left(\mathbf{M}\right) = R$. The subspace identifiability of $\hat{\mathbf{M}}$ is ensured by $\mathbf{M}$ since the latter satisfies the row overlapping condition. As a result, $\operatorname{col}\left(\hat{\mathbf{M}}\right)$ can be determined solely from $\mathbf{M}$ [30, 35, 44]. However, ensuring the unique completion of $\hat{\mathbf{M}}$ requires an additional condition: the vector $\mathbf{z}$ should also have at least R observed entries. Without this, a unique recovery is not possible.

The study in [30] investigates the subspace identifiability of a partially observed low-rank matrix formed by stacking multiple slices (submatrices) of an incomplete tensor as shown in Figure 2. The authors show that subspace identifiability also requires a row-overlapping condition as stated in (6). However, this is not very restrictive as only some, rather than all, pairs of observed submatrices need to satisfy it. As noted earlier, for the full completion to be unique, we also need to ensure that every column contains at least R observed entries. For further details, see [30, 35, 44].

In the following example of a partially observed rank-1 matrix, we provide a geometrical interpretation of the conditions required to identify the column space and set out the basis for the algebraic algorithm for computing the desired column space more generally.

Example 1 Given is a partially observed rank-1 matrix $\mathbf{M} = \begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \\ z_1 & z_2 \end{bmatrix}$. Suppose we are required to determine its 1-dimensional range in $\mathbb{R}^3$.

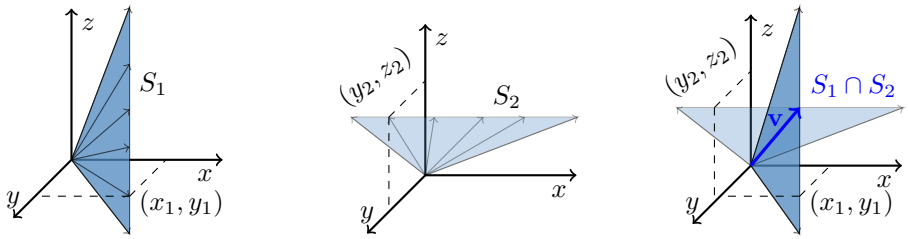


Fig. 3: (a) Affine subspace S_1 corresponding to the first column $\mathbf{m}_{:1}$, (b) Affine subspace S_2 corresponding to the second column $\mathbf{m}_{:2}$, and (c) Intersection of the affine subspaces $S_1 \cap S_2$.

We discuss the subspace identifiability of the rank-1 matrix $\mathbf{M}$ in terms of subspaces associated with partially observed columns. In the first column, z_1 is missing and may lay anywhere on the line parallel to the z-axis passing through (x_1, y_1), as shown in Figure 3 (a). The subspace S_1 (shaded region) accounts for all possible completions of the first column. Similarly, in the second column, the entry x_2 is missing and may lay anywhere on the line parallel to the x-axis passing through (y_2, z_2) and the corresponding subspace that accounts for all possible completions of the second column is S_2 (shaded region), as shown in Figure 3 (b). Given that the unknown 1-dimensional subspace $\text{col}\,(\mathbf{M})$ spans both columns, it must lie in the intersection of these affine subspaces, i.e., $\text{col}\,(\mathbf{M}) \subseteq S_1 \cap S_2$. Note that there is a subset relationship between $\text{col}\,(\mathbf{M})$ and $S_1 \cap S_2$, not an equality. However, if $\dim(S_1 \cap S_2) = 1$, i.e., if $S_1 \cap S_2$ forms a line, then the desired $\text{col}\,(\mathbf{M})$ is necessarily equal to $S_1 \cap S_2$. Therefore, any nonzero vector $\mathbf{v} \in S_1 \cap S_2$ spans $\text{col}\,(\mathbf{M})$, as shown in Figure 3 (c). In this case, any technique that computes the intersection of subspaces can be employed to find the desired subspace $\text{col}\,(\mathbf{M})$.

One approach for determining the intersecting subspace is via the null spaces. Specifically, find $\mathbf{n_1}$ and $\mathbf{n_2}$ such that $\langle \mathbf{n_1}, \mathbf{m}_{:1} \rangle = 0$, and $\langle \mathbf{n_2}, \mathbf{m}_{:2} \rangle = 0$. Define $\mathbf{N} = [\mathbf{n_1}\ \mathbf{n_2}] \in \mathbb{R}^{3\times 2}$. If $\dim(\ker(\mathbf{N}^\top)) = 1$, then the subspace S is given by $\ker(\mathbf{N}^\top)$. The condition $\dim(\ker(\mathbf{N}^\top)) = 1$ represents the dual of the condition $\dim(S_1 \cap S_2) = 1$. Several other approaches have been proposed to find the intersection of subspaces (see, e.g., [48]).

Before proceeding to the rank-R case, for ease of notation, we use $\mathbf{M} \in \mathbb{R}^{J\times K}$ to denote a low-rank matrix $\mathbf{X}_{[1,\ldots,n;n+1,\ldots,N]}$ of which we need to determine the R-dimensional column space. Let $\mathbf{M}_{obs}^{(l)} = \mathbf{S}_r^{(l)}\mathbf{M}\mathbf{S}_c^{(l)} \in \mathbb{R}^{J_l\times K_l}$ represent the lth fully observed isorank submatrix of $\mathbf{M}$. We denote a matrix that holds an orthonormal basis for $\text{col}\,(\mathbf{M})$ by $\mathbf{A} \in \mathbb{R}^{J\times R}$. Below, we discuss two approaches to determine $\mathbf{A}$: (1) the *subspace constraint* approach, which computes it via null spaces of observed submatrices, and (2) the *subspace intersection* method, which computes it via column spaces of observed submatrices.

3.1.1 Subspace constraint approach

The main idea of the subspace constraint method is to determine the overall range of a matrix from a set of constraints derived from the submatrices [32]. This method involves identifying an informationally complete set of fully observed isorank submatrices $\{\mathbf{M}_{obs}^{(l)}\}_{l=1}^{L}$ of $\mathbf{M}$ [29]. By *informationally complete,* we mean that the set of constraints derived from the submatrices should fully characterize the desired subspace $\mathrm{col}\,(\mathbf{A})$. As we assume $\mathbf{M}_{obs}^{(l)}$ is an isorank submatrix of $\mathbf{M}$, we can write $\mathbf{M}_{obs}^{(l)} = \mathbf{A}_{obs}^{(l)}\mathbf{B}_{obs}^{(l)\top}$, where matrices $\mathbf{A}_{obs}^{(l)} \in \mathbb{R}^{J_l \times R}$ and $\mathbf{B}_{obs}^{(l)} \in \mathbb{R}^{K_l \times R}$ are full column rank. Stacking a basis for the orthogonal complement of $\mathrm{col}\left(\mathbf{M}_{obs}^{(l)}\right)$ in a matrix $\mathbf{N}_{obs}^{(l)} \in \mathbb{R}^{J_l \times (J_l - R)}$, each column of the resulting matrix after zero padding, i.e., $\mathbf{S}_r^{(l)\top}\mathbf{N}_{obs}^{(l)} \in \mathbb{R}^{J \times (J_l - R)}$, represents a vector that is orthogonal to the desired subspace $\mathrm{col}\,(\mathbf{A})$, thus imposing constraints on what $\mathrm{col}\,(\mathbf{A})$ may be. To obtain the desired $\mathrm{col}\,(\mathbf{A})$, we find orthogonal complements for L such submatrices $\mathbf{M}_{obs}^{(l)}$ and concatenate them in a matrix $\mathbf{N} = \left[\mathbf{S}_r^{(1)\top}\mathbf{N}_{obs}^{(1)}, \cdots, \mathbf{S}_r^{(L)\top}\mathbf{N}_{obs}^{(L)}\right] \in \mathbb{R}^{J \times (\sum_{l=1}^{L}(J_l - R))}$. If the L orthogonal complements are enough to ensure the $\ker(\mathbf{N}^\top)$ is of minimal dimension, i.e., $\dim(\ker(\mathbf{N}^\top)) = R$, where the very existence of $\mathrm{col}\,(\mathbf{A})$ implies that the dimension is at least R, then the desired $\mathrm{col}\,(\mathbf{A})$ is necessarily given by $\ker(\mathbf{N}^\top)$. In other words, the number of independent constraints must be at least $J - R$, which leads to the inequality $\sum_{l=1}^{L}(J_l - R) \geq J - R$. For example, if we assume that each observed submatrix imposes one (independent) constraint and we have $J_l = R + 1$, then the minimum number of the observed submatrices required is $L = \lceil J - R \rceil$. This can be seen as the generalization of the rank-1 case mentioned in Example 1. In what follows, we assume that our working assumptions hold, i.e., $\{\mathbf{M}_{obs}^{(l)}\}_{l=1}^{L}$ is a set of isorank submatrices of $\mathbf{M}$ and $\dim(\ker(\mathbf{N}^\top)) = R$. Then, this set of submatrices is *informationally complete*. In the noisy case, we estimate $\mathbf{N}_{obs}^{(l)}$ from the left-singular vectors corresponding to the smallest singular values of $\mathbf{M}_{obs}^{(l)}$. Let the SVD of $\mathbf{M}_{obs}^{(l)}$ be $\mathbf{M}_{obs}^{(l)} = \mathbf{U}^{(l)}\mathbf{\Sigma}^{(l)}\mathbf{V}^{(l)\top}$. Then $\mathbf{N}_{obs}^{(l)} = \left[\mathbf{u}_{:R+1}^{(l)} \cdots \mathbf{u}_{:J_l}^{(l)}\right]$. In the noisy case, we also estimate the desired $\mathrm{col}\,(\mathbf{A})$ to be the subspace that is most orthogonal to the space spanned by the columns of $\mathbf{N}$, i.e., we estimate $\mathrm{col}\,(\mathbf{A})$ as the (approximate) kernel of $\mathbf{N}^\top$. Let the SVD of $\mathbf{N}$ be $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. Then $\mathbf{A} = [\mathbf{u}_{:J-R+1} \cdots \mathbf{u}_{:J}]$.

3.1.2 Subspace intersection approach

Let us first define a binary matrix $\tilde{\mathbf{S}}_r^{(l)} \in \{0,1\}^{(J-J_l)\times J}$, of which the rows correspond to standard unit vectors that indicate which rows of $\mathbf{M}$ are missing in $\mathbf{M}_{obs}^{(l)}$. (The matrix $\tilde{\mathbf{S}}_r^{(l)}$ is such that $\mathbf{I}_J = \begin{bmatrix}\mathbf{S}_r^{(l)\top} & \tilde{\mathbf{S}}_r^{(l)\top}\end{bmatrix}\mathbf{\Pi} \in \mathbb{R}^{J\times J}$ for some permutation matrix $\mathbf{\Pi} \in \mathbb{R}^{J\times J}$.) Additionally, let S_l denote a subspace that accounts for all possible completions of rows that are missing in $\mathbf{M}_{obs}^{(l)}$. Specifically, let the SVD of $\mathbf{M}_{obs}^{(l)}$ be $\mathbf{M}_{obs}^{(l)} = \mathbf{U}^{(l)}\mathbf{\Sigma}^{(l)}\mathbf{V}^{(l)\top}$. Then, the subspace S_l is given by $S_l = \mathrm{span}\,(\mathbf{Q}_{S_l})$, where

$\mathbf{Q}_{S_l} := \begin{bmatrix} \mathbf{S}_r^{(l)\top}[\mathbf{u}_{:1}^{(l)} \cdots \mathbf{u}_{:R}^{(l)}] & \tilde{\mathbf{S}}_r^{(l)\top} \end{bmatrix} \in \mathbb{R}^{J\times(R+J-J_l)}$; note that the columns of $\mathbf{Q}_{S_l}$ form an orthonormal basis for the subspace S_l.

As illustrated in Example 1, the desired subspace $\mathrm{col}\,(\mathbf{M})$ satisfies $\mathrm{col}\,(\mathbf{M}) \subseteq \left(\cap_{l=1}^{L} S_l\right)$. However, since we assume the subspaces are associated with an informationally complete set of fully observed isorank submatrices $\{\mathbf{M}_{obs}^{(l)}\}_{l=1}^{L}$, i.e., $\dim\left(\cap_{l=1}^{L} S_l\right) = R \Leftrightarrow \dim(\ker(\mathbf{N}^\top)) = R$, it follows that the R-dimensional subspace $\mathrm{col}\,(\mathbf{M})$ is necessarily equal to $\cap_{l=1}^{L} S_l$. Hence, we simply need to find the intersection of the subspaces.

A closed-form solution for computing the intersection of $L \geq 2$ subspaces in finite-dimensional spaces is discussed in [48], and a low-complexity implementation of these formulas based on SVD is presented in [30, 49]. The procedure to compute the intersection is as follows. Given a set of matrices $\{\mathbf{Q}_{S_l}\}_{l=1}^{L}$, the intersection of the corresponding subspaces can be computed as $\cap_{l=1}^{L} S_l = \ker\left(L\mathbf{I}_J - \sum_{l=1}^{L} \mathbf{Q}_{S_l}\mathbf{Q}_{S_l}^\top\right)$. It has been shown that if $\dim\left(\cap_{l=1}^{L} S_l\right) = R$, then the solution can be computed more efficiently via the SVD of a concatenated matrix $\mathbf{Q} = [\mathbf{Q}_{S_1}, \cdots, \mathbf{Q}_{S_L}] \in \mathbb{R}^{J\times\left(LR+\sum_{l=1}^{L}(J-J_l)\right)}$, without first calculating the orthogonal projectors $\left\{\mathbf{Q}_{S_l}\mathbf{Q}_{S_l}^\top\right\}$. Specifically, let the SVD of $\mathbf{Q}$ be $\mathbf{Q} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. Then $\mathbf{A} = [\mathbf{u}_{:1}, \cdots, \mathbf{u}_{:R}]$ forms an orthonormal basis for the desired subspace $\mathrm{col}\,(\mathbf{M})$. See [30, 48, 50] for an in-depth discussion. In case the matrix $\mathbf{Q}$ is too large to fit in memory, one can use an incremental SVD to find the dominant left singular subspace [51, 52].

3.2 Algorithm

This section presents our algorithm for computing the TT decomposition of a tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ observed along the Nth mode. Our algorithm is similar to Algorithm 1, with the key subspace computation step in line 2 adapted to handle a tensor observed through mode-N fibers. The key steps of the algorithm are as follows. The piecewise subspace learning approach is employed to compute orthonormal bases for the column spaces of the partially observed matrix unfoldings $\mathbf{X}_{[1,\ldots,n;\,n+1,\ldots,N]}$ for $1 \leq n \leq N-2$, assuming that the corresponding observed submatrices satisfy the informationally complete condition. These orthonormal bases are used to compute the TT cores $\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(N-2)}$. The last core $\mathcal{G}^{(N)}$ is set to an orthonormal basis for the observed rows of the $(N-1)$th matrix unfolding, and the penultimate core $\mathcal{G}^{(N-1)}$ is computed in a least-squares sense as explained next. The pseudocode for the proposed method is outlined in Algorithm 2.

Computing the next-to-last TT core $\mathcal{G}^{(N-1)}$

In the fully observed case, the last core can be computed using the SVD of the $(N-1)$th unfolding matrix $\mathbf{X}_{[1,\ldots,N-1;N]}$. Let its SVD be $\mathbf{X}_{[1,\ldots,N-1;N]} = \mathbf{U}^{(N-1)}\,\mathbf{\Sigma}^{(N-1)}\,\mathbf{V}^{(N-1)\top}$. Then, the last TT core is obtained as $\mathcal{G}^{(N)} = \mathbf{\Sigma}^{(N-1)}\mathbf{V}^{(N-1)\top}$. However, for fiber-wise observed tensors, this is not directly possible. Indeed, since $\mathcal{X}$ is observed through mode-N fibers, some rows of $\mathbf{X}_{[1,\ldots,N-1;N]}$ are now completely observed while other rows are entirely missing. Consequently, under the informationally complete condition, $\mathbf{V}^{(N-1)\top}$ can be estimated but $\mathbf{U}^{(N-1)}$

and $\mathbf{\Sigma}^{(N-1)}$ cannot. To address this issue, we set the last core $\mathcal{G}^{(N)} = \mathbf{V}^{(N-1)\top}$, i.e., the orthonormal basis for the observed rows $(\mathbf{S}_r\mathbf{X}_{[1,\ldots,N-1;N]})$. Here, $\mathbf{S}_r \in \{0,1\}^{|\alpha|\times\prod_{i=1}^{N-1} I_i}$ represents the row selection matrix for the $(N-1)$th matrix unfolding, where $|\alpha|$ denotes the number of observed mode-N fibers (rows) indexed by α. We then fix the scaling indeterminacies by computing the next-to-last TT core in a least-squares sense. For the third-order tensor $\tilde{\mathcal{X}} = \mathcal{X}_{[1,\ldots,N-2;N-1;N]}$, the slice-wise representation, using (3), can be written as:

$$\tilde{\mathbf{X}}_{:i:} = \mathbf{G}^{(<N-1)}\mathbf{G}^{(N-1)}_{:i:}\mathbf{G}^{(>N-1)},\ i \in \{1,\ldots,I_{N-1}\}, \tag{7}$$

where the matrices $\mathbf{G}^{(<N-1)} \in \mathbb{R}^{\prod_{i=1}^{N-2} I_i \times R_{N-2}}$ and $\mathbf{G}^{(>N-1)} = \mathbf{G}^{(N)} \in \mathbb{R}^{R_{N-1}\times I_N}$ have orthonormal columns and orthonormal rows, respectively. Let $\{\mathbf{S}_r^{(i)}\}_{i=1}^{I_{N-1}}$ be the row selection matrices for the mode-2 slices of $\tilde{\mathcal{X}}$. The mode-2 slices $\mathbf{G}^{(N-1)}_{:i:}$ of $\mathcal{G}^{(N-1)}$ can now be computed by solving the following linear systems:

$$\mathbf{S}_r^{(i)}\left(\tilde{\mathbf{X}}_{:i:}\mathbf{G}^{(N)\top}\right) = \left(\mathbf{S}_r^{(i)}\mathbf{G}^{(<N-1)}\right)\mathbf{G}^{(N-1)}_{:i:},\ i \in \{1,\ldots,I_{N-1}\}. \tag{8}$$

Algorithm 2: TT mode-N fiber-wise

Input: A tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ observed through mode-N fibers, TT rank $(R_0,\ldots,R_N)$ and row selection matrix $\mathbf{S}_r$

Output: TT cores $\mathcal{G}^{(1)},\ldots,\mathcal{G}^{(N)}$

/*compute orthonormal bases for the ranges of the nth unfoldings*/

for $1 \le n \le N-2$ **do**
  Compute $\mathbf{A}^{(n)} \in \mathbb{R}^{\prod_{i=1}^{n} I_i \times R_n}$ using the subspace constraint method, as discussed in Section 3.1.1, or the subspace intersection method, as discussed in Section 3.1.2.
end

/*compute the TT cores*/

Set $\mathcal{G}^{(1)} = \texttt{reshape}\left(\mathbf{A}^{(1)}, [1, I_1, R_1]\right)$
for $1 \le n \le N-3$ **do**
  Compute $\mathbf{W}^{(n)} = \mathbf{A}^{(n)\top}\left(\mathbf{A}^{(n+1)}\right)_{[1,\ldots,n;n+1,(n+1)+1]}$
  Set $\mathcal{G}^{(n+1)} = \texttt{reshape}\left(\mathbf{W}^{(n)}, [R_n, I_{n+1}, R_{n+1}]\right)$
end
Compute $[\sim,\sim,\mathbf{V}^{(N-1)}] = \texttt{svd}\left(\mathbf{S}_r\mathbf{X}_{[1,\ldots,N-1;N]}, R_{N-1}\right)$
Set $\mathcal{G}^{(N)} = \texttt{reshape}\left(\mathbf{V}^{(N-1)\top}, [R_{N-1}, I_N, 1]\right)$
Compute $\mathcal{G}^{(N-1)}$ by solving the linear systems in (8)

3.3 Uniqueness conditions

This section will summarize the conditions for uniqueness, up to basis transformation, of the TT cores in the noiseless case.

In Section 3.1, we showed that an orthonormal basis for the range of the nth unfolding matrix, as shown in (4), can be computed from a subset of its fully observed submatrices that is informationally complete. For this set of submatrices to be informationally complete, the following conditions need to be met. (i) The observed submatrices $\mathbf{S}_r^{(i_l)}\tilde{\mathbf{X}}_{:i_l:} \in \mathbb{R}^{|\alpha_l| \times I_N}$ are isorank, i.e., rank $\left(\mathbf{S}_r^{(i_l)}\tilde{\mathbf{X}}_{:i_l:}\right) =$ rank $\left(\mathbf{X}_{[1,\ldots,n;n+1,\ldots,N]}\right) = R_n$. This implies that the submatrices in Figure 2 have at least R_n observed rows. (ii) We should have $\dim(\ker(\mathbf{N}^\top)) = R \Leftrightarrow \dim\left(\cap_{l=1}^{L} S_l\right) = R$. As noted earlier, if there exist some pairs of slices, say $\tilde{\mathbf{X}}_{:i_{l_1}:}$ and $\tilde{\mathbf{X}}_{:i_{l_2}:}$, which overlap with at least R_n observed rows, and if every row appears in at least one of the submatrices, i.e., $\cup_{\ell=1}^{L}\alpha_\ell = \{1,\ldots,\prod_{i=1}^{n} I_i\}$, then, generically, condition (ii) is satisfied. For a detailed discussion, refer to [30, 35] and the references therein. Conditions (i) and (ii) are required to retrieve the TT cores $\mathcal{G}^{(1)},\ldots,\mathcal{G}^{(N-2)}$. The next-to-last TT core $\mathcal{G}^{(N-1)}$ is obtained by solving a set of linear systems, as in (8). To uniquely solve (8), the matrix $\left(\mathbf{S}_r^{(i)}\mathbf{G}^{(<N-1)}\right)$ must have full column rank for all i. This implies that the mode-2 slices of the third-order reshaping $\tilde{\mathcal{X}} = \mathcal{X}_{[1,\ldots,N-2;N-1;N]}$ should have at least R_{N-2} observed rows. The last TT core $\mathcal{G}^{(N)}$ can be retrieved iff rank $\left(\mathbf{S}_r\mathbf{X}_{[1,\ldots,N-1;N]}\right) \geq R_{N-1}$, i.e., generically, if at least R_{N-1} mode-N fibers are observed.

In the noiseless case, the proposed method computes the exact TT decomposition of the fiber-wise observed tensor if the uniqueness conditions are satisfied. In the presence of noise, each step can be computed in a least-squares sense, and thus, the estimated TT decomposition is expected to be close to the true TT decomposition.

4 Numerical experiments

This section evaluates the performance of the proposed method across three setups. First, we experiment with synthetic data to show that our algebraic method is accurate and computationally efficient, comparing it with the tensor train weighted optimization (TT-WOPT) from [15] and the simple low-rank tensor completion (SiLRTC-TT) from [25]. Second, we showcase two real-world applications, namely multidimensional harmonic retrieval (MHR) and spatiotemporal weather data completion. Finally, we show the use of the proposed algebraic approach as a "proxy" for efficiently performing subsequent computations. In line with [38], we fit a non-negative CPD to the TT approximation instead of the actual tensor.

Baselines:

- TT-WOPT is an error minimization method that fits the parameters of a fixed rank TT model by minimizing a weighted least-squares loss using the gradient descent approach; see [15] for details.
- SiLRTC-TT is a rank minimization method that minimizes a nuclear norm relaxation of the TT rank of the tensor. The resulting convex problem is solved by a singular value thresholding algorithm; see [25, 53] for more details.

The experiments are performed on an HP EliteBook 845 G8 Notebook PC with an AMD Ryzen 7 PRO 5850U CPU and 32GB RAM.

4.1 Synthetic data

4.1.1 Completion from noisy observations

A 5th-order tensor in TT format $\mathcal{X} \in \mathbb{R}^{15\times15\times15\times15\times15}$ is generated with $\text{rank}_{TT}(\mathcal{X}) = (1, 3, 3, 3, 4, 1)$, by sampling entries of the TT cores from a standard normal distribution. Random Gaussian i.i.d. noise $\mathcal{N}$ is added to the generated tensor, resulting in a noisy tensor $\mathcal{X}_{noisy} = \mathcal{X} + \mathcal{N}$ with a fixed signal-to-noise ratio (SNR), defined as $\text{SNR} = 20\log_{10}\frac{\|\mathcal{X}\|}{\|\mathcal{N}\|}$. Forty percent of the mode-5 fibers are randomly removed from the noisy tensor. Algorithm 2 is run along with the reference algorithms to compute the low-rank TT approximations with SNR varying from -10 to 50 dB. The reference algorithms are run with default parameters (see [15, 25]), except for the parameter α in SiLRTC-TT, which is determined through an extensive grid search. The median of the relative error, defined as $\text{Relative Error} = \frac{\|\mathcal{X}-\hat{\mathcal{X}}\|_F}{\|\mathcal{X}\|_F}$, is plotted across 30 trials in Figure 4 (left). The relative error of the proposed algorithm is observed to be lower than that of SiLRTC-TT algorithm, while the lowest error is noted for TT-WOPT. This is expected as the latter explicitly tries to minimize the error. In comparison, the accuracy of the proposed algorithm is very good despite the fact that it only relies on standard NLA operations. The time required to compute the approximation is shown in Figure 4 (right), which depicts the proposed method is more than a magnitude faster, and the effect of the SNR on computation time is not significant.

4.1.2 Scalability

This experiment compares the scalability of the proposed approach with the reference algorithms. The reference algorithms are based on optimization and are terminated when the relative change in the function value falls below 10^{-7}. If this level of precision is not achieved, the algorithms terminate upon reaching the maximum number of iterations (maximum iterations $= 7 \times 10^2$). A 4th-order tensor in TT format $\mathcal{X} \in \mathbb{R}^{I\times I\times I\times I}$ is generated with a fixed $\text{rank}_{TT}(\mathcal{X}) = (1, 4, 4, 4, 1)$, where I is varied from 10 to 50. Random Gaussian i.i.d. noise is added to the generated tensor to achieve an SNR of 25 dB, and 35% of the mode-4 fibers are randomly removed. Figure 5 (left) shows the median relative error computed over 30 trials and plotted as a function of I. The TT-WOPT algorithm is shown to achieve the highest accuracy, while a

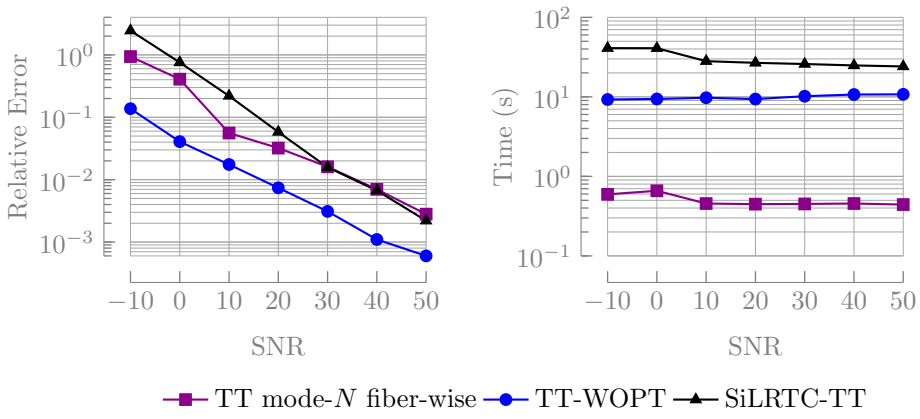


Fig. 4: (left) Accuracy of our method is slightly lower than that of TT-WOPT, which is expected due to its reliance on only standard NLA operations. (right) However, it provides a significant computational speedup.

good but not optimal accuracy is achieved by our method. Notably, for the proposed approach, accuracy consistently improves as the problem size increases. This behaviour is expected because, for a fixed missing rate and TT rank, an increase in I results in relatively more observed data being used to estimate the model parameters (TT cores). Similar reasoning applies to the reference optimization methods; see [25, 53] for more details. In Figure 5 (right), it is shown that the computation time for the proposed method is the lowest, in line with expectations. Specifically, in the current experimental settings, as I increases from 30 to 50, the time increases by a factor of 4.2 in our approach, while the time increases by factors of 15.8 and 47.8 for TT-WOPT and SiLRTC-TT algorithms, respectively.

4.2 Real-life applications

4.2.1 Multidimensional harmonic retrieval

Harmonic retrieval is a classical problem in signal processing. The goal is to estimate the parameters of a signal that is modelled as a sum of complex exponentials. MHR is the natural multidimensional extension. An MHR data tensor sampled over the $(D+1)$th-order tensor grid can be modelled as:

$$x_{i_1 i_2 \cdots i_d k} = \sum_{r=1}^{R} s_r(k) \prod_{d=1}^{D} \mathrm{e}^{j(i_d-1)\mu_r^{(d)}} + n_{i_1 i_2 \cdots i_D k}, \tag{9}$$

where $j^2 = -1$ and $s_r(k)$ is the kth complex symbol carried by the rth multidimensional harmonic. The noise $n_{i_1 i_2 \cdots i_D k}$ is modelled as zero-mean i.i.d additive Gaussian noise.

A 5th-order data tensor $\mathcal{X} \in \mathbb{C}^{10\times10\times10\times10\times25}$ is generated by a CPD model of rank $R = 4$, using (9). The parameter D is set to 4, and binary phase shift keying sources

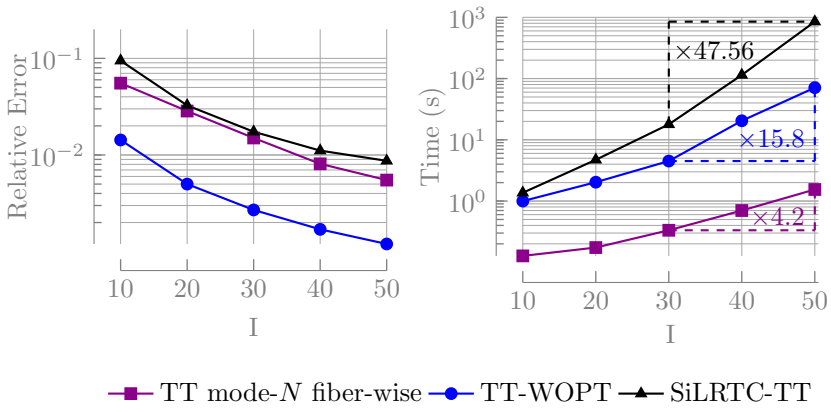


Fig. 5: (left) Accuracy improves as I increases since relatively more data becomes available per parameter (with the missing rate and TT rank held constant). TT-WOPT achieves the highest accuracy. (right) Meanwhile, our method outperforms both reference algorithms in terms of computation time.

$s_r(k) \in \{-1, 1\}$ of length $K = 25$ are used. The parameter vectors are set as follows: $\boldsymbol{\mu}^{(1)} = [1, -0.5, 0.1, -0.8]$, $\boldsymbol{\mu}^{(2)} = [-0.5, 1, -0.9, 0.2]$, $\boldsymbol{\mu}^{(3)} = [-0.2, -0.6, 1.0, 0.4]$, and $\boldsymbol{\mu}^{(4)} = [-0.8, 0.4, 0.3, -0.1]$. The parameter settings and data generation are performed as described in [54]. In the first experiment, Gaussian noise is added to the generated tensor with SNR varying from -10 dB to 40 dB. Forty percent of fibers in mode-5 are randomly removed. The estimated TT approximations are subsequently used to compute the parameter vectors $\hat{\boldsymbol{\mu}}^{(d)}$ using the classical ESPRIT algorithm [55]. Root mean square error (RMSE), which is defined as $\text{RMSE} = \sqrt{\frac{1}{RD}\sum_{r=1}^{R}\sum_{d=1}^{D}\left(\mu_r^{(d)} - \hat{\mu}_r^{(d)}\right)^2}$, is used to assess the accuracy. The results are summarized in the left column of Figure 6. Next, the SNR is fixed to 25 dB, and the effect of the missing rate on the accuracy is studied. The RMSE and computation time are plotted as a function of the missing fiber rate in the right column of Figure 6.

4.2.2 Spatiotemporal weather data imputation

Spatiotemporal weather data are typically collected at fixed spatial coordinates, with measurements at each location varying over time. In practice, however, it is often not feasible to gather (or store) data at every location—especially when the spatial resolution is high. In such cases, time series data are recorded only for a subset of spatial coordinates across selected time windows. These data may be organized in a tensor of size—for example, "longitudes × latitudes × year × day of year"—with observations in a fiber-wise pattern. In this experiment, such a dataset consisting of the maximum temperature time series (TMAX in °C) from the NASA POWER database[2] is used. The dataset comprises 5 478 daily observations from 01 January 2005 through

[2]These data are obtained from the NASA Langley Research Center (LaRC) POWER Project funded through the NASA Earth Science/Applied Science Program: https://power.larc.nasa.gov

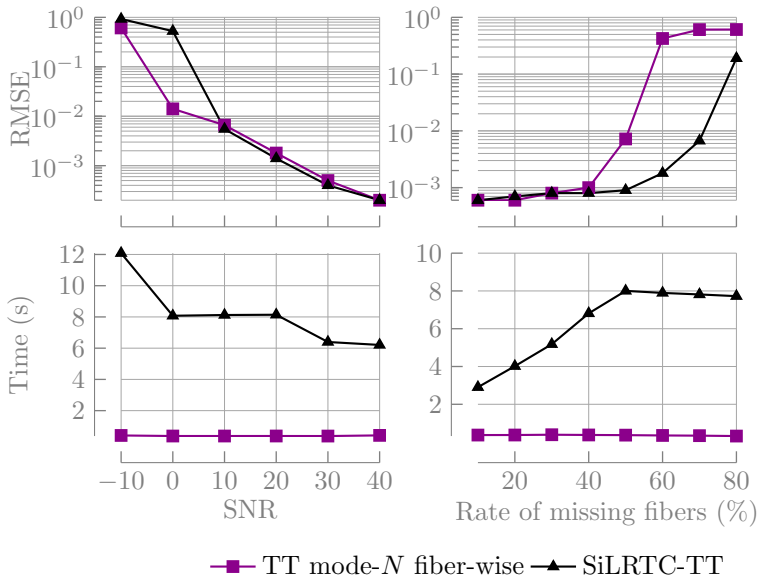


Fig. 6: (top left) The proposed method achieves lower RMSE at low SNRs and performs comparably to SiLRTC-TT at higher SNRs. (bottom left) It also requires less computation time, while SiLRTC-TT converges slowly at low SNRs. (top right) Our approach remains accurate even with up to a 50% missing rate, but beyond that, performance degrades sharply as its working conditions no longer hold. (bottom right) The proposed method is faster overall, while SiLRTC-TT slows down at high missing rates due to increased optimization challenges.

31 December 2019, and spans the region bounded by 4.0°E to 50.5°E longitude and 30.0°N to 54.5°N latitude, on a regular $0.5° \times 0.5°$ grid. The code to download the dataset is provided in [31]. The data are reshaped into a 4th-order tensor, organized yearly, with each year comprising 366 days[3], resulting in a shape of $94 \times 50 \times 15 \times 366$. We simulate a scenario in which time series data are observed only for a subset of spatial coordinates during specific years. This results in a spatiotemporal tensor whose fibers along the last mode are either fully observed or entirely missing.

In the first experiment, the mode-4 fibers of the data tensor are randomly removed, with the rate of missing fibers varying from 40% to 65%. The approximation is computed using our method for different TT ranks, and the median relative error between the ground truth and the estimated tensor (i.e., the overall error, which includes both prediction error and reconstruction error) is recorded over 30 trials. In Table 1, it is observed that the approximation is improved by increasing the TT rank. A reasonably good approximation is obtained even when up to 65% of the mode-4 fibers are completely missing. However, the error is observed to rise once the missing fibers exceed this rate. A sharp rise in relative error is noted when the approximation rank is increased under a high missing rate (see, e.g., the value highlighted in Table 1). With

[3]Day-of-year alignment is performed, with missing entries imputed via nearest neighbor averaging.

such a high TT rank $\left(\text{i.e., } \text{rank}_{TT}(\hat{\mathcal{X}}) = (1, 42, 42, 46, 1)\right)$, the overlapping conditions are no longer satisfied, resulting in the estimated approximation being rendered inaccurate. Nevertheless, a valid approximation can still be obtained if the TT rank is low enough to satisfy the overlapping conditions.

Next, the rate of missing fibers is set to 50%, and mode-4 fibers are randomly removed. The approximation is computed using $\text{rank}_{TT}(\hat{\mathcal{X}}) = (1, 15, 15, 50, 1)$, which is determined by analyzing the singular values of the matrix unfoldings. The median relative error across 10 trials is found to be below 9.7%. Figure 7 visualizes the observed, estimated, and residual values (i.e., the absolute differences between the ground truth and the estimates) for segments of the time series between 01 January 2013 and 12 December 2018 at four locations. It is observed that the maximum temperature (TMAX) exhibits a sinusoidal pattern, with peaks corresponding to summer temperatures and valleys corresponding to winter temperatures.

Table 1: For a fixed missing rate, the completion accuracy improves with increasing the TT rank. For a fixed TT rank, the error increases only slightly, showing that the method can achieve reasonably good approximations even at higher missing rates if the working conditions are met. If not (as in the highlighted case), the approximation becomes inaccurate.

	Rate of missing fibers (%)					
$\text{rank}_{TT}(\hat{\mathcal{X}})$	40	45	50	55	60	65
(1, 10, 10, 14, 1)	0.1393	0.1395	0.1396	0.1396	0.1396	0.1397
(1, 18, 18, 22, 1)	0.1174	0.1176	0.1177	0.1180	0.1181	0.1180
(1, 26, 26, 30, 1)	0.1044	0.1044	0.1045	0.1046	0.1049	0.1050
(1, 34, 34, 38, 1)	0.0943	0.0944	0.0946	0.0947	0.0956	0.0956
(1, 42, 42, 46, 1)	0.0869	0.0869	0.0874	0.0873	0.0881	**0.6304**

4.3 TT approximation as a prior for efficient computation

One can use our algebraic method as an initialization for optimization-based methods, improving computational speed and potentially reducing the risk of convergence to local minima, since the algebraic method—which works under deterministic conditions—yields a solution already close to the true one. Moreover, the solution obtained from the proposed method can also be directly used for downstream tasks, particularly in low-noise settings. In line with [38], our experiments show that the estimated TT approximation can also serve as a proxy for efficiently computing other tensor decompositions.

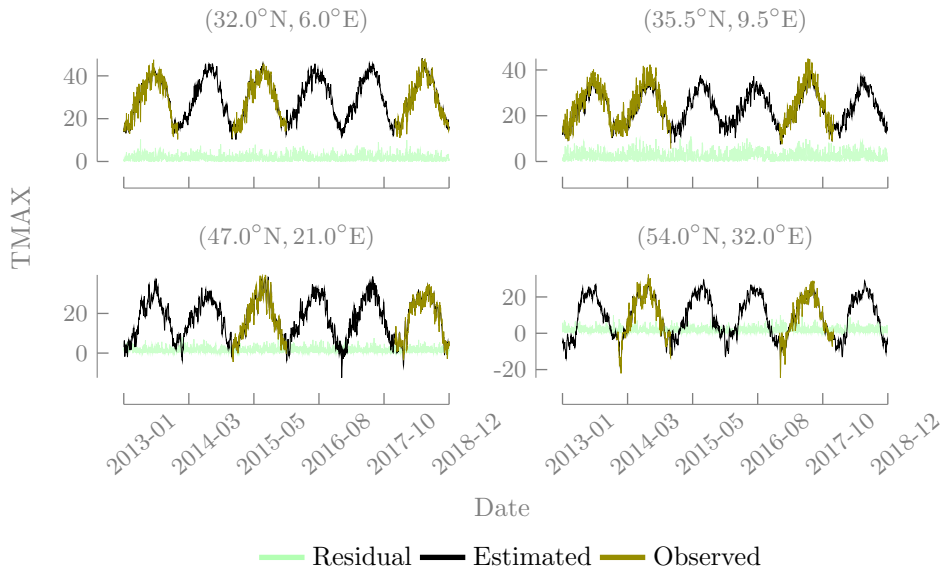


Fig. 7: The estimated time series closely matches the observations. Even when data are completely missing for consecutive years, our estimation remains reasonably accurate, as evidenced by small residuals. This is due to the fact that the data exhibit a low-rank structure and well-identifiable patterns.

4.3.1 Initialization of optimization methods

In a noiseless setting, the proposed algorithm computes the exact TT decomposition when the uniqueness conditions are satisfied. However, in noisy settings, the approximation is good but not optimal; further refinement can be performed using optimization methods. This experiment compares TT-WOPT—run with different *random* initializations—to a hybrid approach in which the *algebraic* method serves as an initialization for TT-WOPT.

A 4th-order random tensor in TT format $\mathcal{X} \in \mathbb{R}^{40\times40\times40\times40}$ is generated with a fixed $\text{rank}_{TT}(\mathcal{X}) = (1, 5, 5, 5, 1)$. Random Gaussian i.i.d. noise is added to the generated tensor at SNR levels of 100 dB, 75 dB, 50 dB, and 25 dB to simulate low and moderate noise conditions. Sixty percent of the mode-4 fibers are randomly removed. Completion is then performed with two initialization strategies: one using 10 different random initializations, and a hybrid approach where the TT approximation computed by the algebraic algorithm is used as the initialization for (one run of) TT-WOPT. Figure 8 compares the number of iterations required to reach the same convergence criterion for the algebraic and random initialization strategies. Each dot indicates the number of iterations required to reach the convergence criterion in a single experiment. For the random strategy, each dot corresponds to the median number of iterations across 10 different initializations. A total of 100 experiments is conducted. The (median) accuracies achieved at SNR levels of 100 dB, 75 dB, 50 dB, and 25 dB are 4.51×10^{-6}, 3.27×10^{-5}, 2.54×10^{-4}, and 2.76×10^{-3}, respectively, for the

hybrid strategy, and 1.28×10^{-6}, 3.01×10^{-5}, 2.52×10^{-4}, and 2.76×10^{-3}, respectively, for the random initialization strategy; thus, both methods achieve comparable accuracies over the considered SNR levels. Meanwhile, with algebraic initialization, the TT-WOPT method reaches this accuracy in significantly fewer iterations, as the algebraic initialization is already close to the optimal solution; see Figure 8 (left).

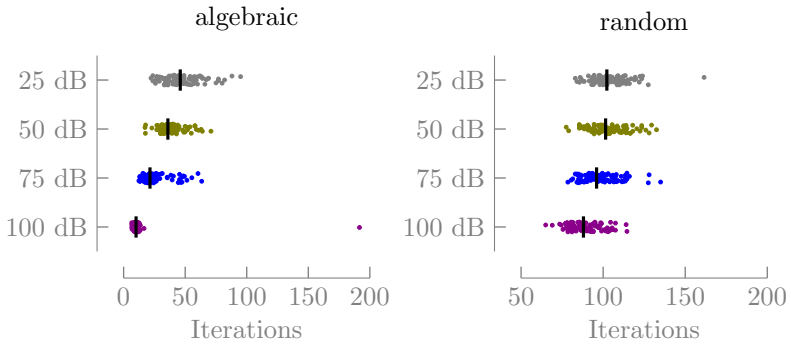


Fig. 8: Both strategies achieve comparable accuracy, but with algebraic initialization, the TT-WOPT method reaches it in significantly less time or fewer iterations. Under low noise settings, the (TT) algebraic initialization is already close to the optimal solution and thus does not require many iterations to converge. The median number of iterations is indicated by vertical bars.

Next, we investigate the success of the two approaches in accurately estimating the underlying decomposition when the rate of missing fibers is high. A 4th-order random tensor in TT format $\mathcal{X} \in \mathbb{R}^{20\times20\times20\times20}$ is generated with a fixed $\text{rank}_{TT}(\mathcal{X}) = (1, 4, 4, 4, 1)$. Random Gaussian i.i.d. noise is added to the generated tensor at SNR levels of 75 dB, 50 dB, and 25 dB. Approximately 20% of the mode-4 fibers are sampled in such a way that the working conditions, as discussed in Section 3.3, on the observation pattern are satisfied. Completion of the tensor is performed similarly to the method described in the previous experiment. Figure 9 shows the distribution of relative error across three different noise levels. Each dot represents one of 100 experiments, with medians calculated over 10 initializations for the random strategy. It is observed that the TTWOPT becomes highly successful when initialized with the algebraic method. To assess the accuracy, we define thresholds for SNR values of 25 dB, 50 dB, and 75 dB as 0.015, 0.00125, and 0.000175, respectively. A completion is considered successful if the relative error is below the threshold associated with the specified noise level. Figure 10 shows the success rate at each noise level, defined as the proportion of 100 experiments that satisfy the accuracy criterion. The success rate of the algebraic strategy is defined as the fraction of successful trials, whereas the success rate of the random strategy is computed by first calculating the fraction of successful trials for each of the 10 initializations and then averaging these fractions to obtain the final success rate. Note that each noise level has its corresponding threshold, meaning comparisons are made within the same SNR level rather than across different levels.

At an SNR of 75 dB, the algebraic strategy successfully completed 96 out of 100 experiments. In comparison, the random strategy has an average success rate of 76.2 out of 100 over 10 initializations. Overall, the algebraic strategy consistently demonstrated a higher success rate than the random strategy under different noise conditions.

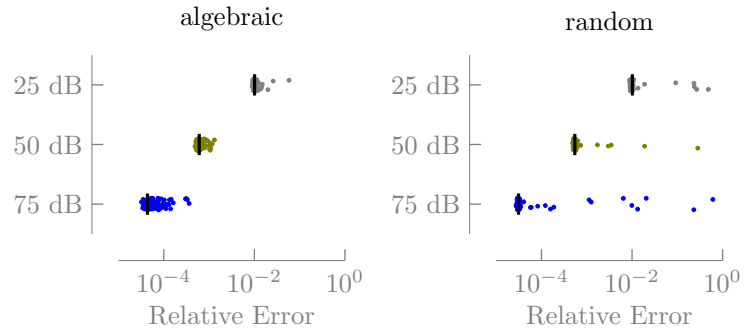


Fig. 9: Performance comparison of TTWOPT initialization methods at 25 dB, 50 dB, and 75 dB SNR levels. While median relative errors (indicated by vertical bars) for random initialization (0.0100, 5.50×10^{-4}, and 3.11×10^{-5}) are comparable to algebraic initialization (0.0101, 6.06×10^{-4}, and 4.36×10^{-5}), respectively, random initialization exhibits significantly lower success rates in accurate TT completion. Algebraic initialization achieves reliable convergence in all trials, while random initialization fails to converge in a substantial fraction of cases, as evidenced by the wider error distributions.

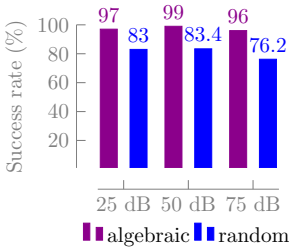


Fig. 10: Each noise level has its threshold value: a completion is deemed successful if the relative error is below 0.015, 0.00125, and 0.000175 for SNR values of 25 dB, 50 dB, and 75 dB, respectively. The algebraic strategy exhibits a higher success rate than the average success rate of the random strategy. This is also evident in Figure 9, where the standard deviation of the relative errors is lower for the algebraic strategy, indicating more consistent performance.

4.3.2 Proxy for non-negative CPD

In this experiment, we focus on computing a non-negative CPD of a non-negative data tensor that is observed through mode-N fibers. The constrained CPD is computed using two strategies: the *full* strategy, where partial data in a dense format is directly used to compute the non-negative CPD, and the *compressed* strategy, where we first compute the TT approximation using our method and then use the compressed representation to compute the non-negative CPD. We use projected Gauss–Newton algorithm to compute the CPD (`cpd_nls` with `nlsb_gndl` solver), as discussed in [38].

A 4th-order tensor in CPD format $\mathcal{X} \in \mathbb{R}^{I \times I \times I \times I}$ is generated with a fixed $\text{rank}_{CP}(\mathcal{X}) = 4$ by sampling the factor matrices from a uniform distribution $\mathcal{U}(0, 1)$. Random Gaussian i.i.d. noise is added to the generated tensor to achieve an SNR of 40 dB. Fifty percent of the mode-4 fibers are randomly removed from the noisy tensor. The constrained rank-4 CPD is computed using both the full and compressed strategies for I varying from 30 to 50. The median of the computation time is plotted across 50 trials in Figure 11 (left). It is observed that the use of the TT approximation as a prior significantly improves speed. The overall computation time in the compressed strategy, which includes the proxy step and the computation of the constrained CPD from the compressed tensor, remains lower than that of the full strategy. The small accuracy gap that arises in noisy settings can be eliminated by refining the proxy over a few iterations. Moreover, the computational cost of the proxy step can be further reduced by using parallel implementations or randomized SVD algorithms.

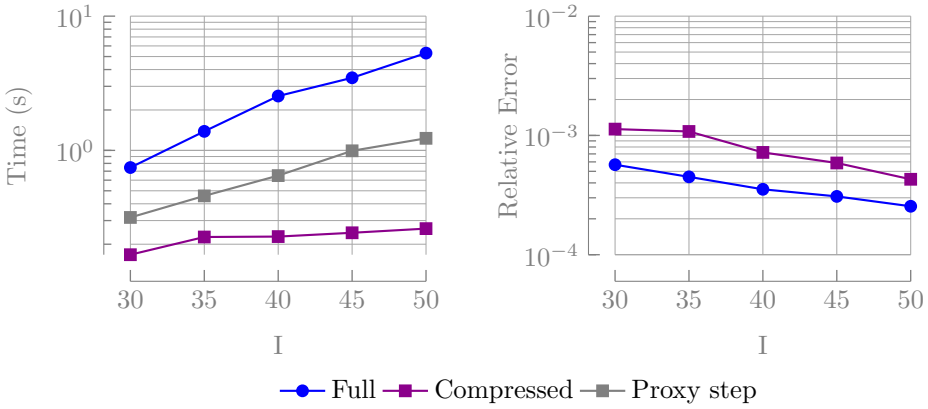


Fig. 11: The time required to compute the non-negative CPD, given a TT approximation as a prior, increases very slowly with problem size. On the other hand, the time required to compute the (TT) proxy increases, as indicated by the grey curve. However, the overall computation time in the compressed strategy remains significantly lower than that of the full strategy. (right) The trade-off is that the compressed strategy is slightly less accurate, but this can be resolved by refining the proxy for a few iterations.

Note on accuracy gap between the algebraic and optimization methods

The observed accuracy gap between the algebraic and optimization methods stems not only from the algebraic method's reliance solely on standard NLA operations—which makes it slightly less accurate in noisy cases—but also from the difference in the proportion of data points that each method uses. When 70% of the fibers along a specific mode are randomly removed, the optimization method uses all of the remaining 30% of the observations to compute the completion. In contrast, the algebraic method only uses the observed submatrices that satisfy the working conditions for recovering the column spaces of the matrix unfoldings, discarding the rest. Consequently, the algebraic method uses fewer than 30% of the observed fibers. Under deterministic sampling, where all 30% of the observed fibers satisfy the working conditions, the accuracy gap is expected to be small.

Another key point is that only the observed rows of the (individual) mode-2 slices, as in Equation 4, are used in the above experiments. The observed submatrices spanning multiple slices, satisfying the working conditions, can also be utilized. Let us consider an example of an observed submatrix spanning two mode-2 slices, $\mathbf{S}_r^{(l_1)}\tilde{\mathbf{X}}_{:l_1:} \in \mathbb{R}^{|\alpha_{l_1}|\times I_N}$ and $\mathbf{S}_r^{(l_2)}\tilde{\mathbf{X}}_{:l_2:} \in \mathbb{R}^{|\alpha_{l_2}|\times I_N}$, which have overlapping rows indexed by $\alpha_{l_1\cap l_2} = \alpha_{l_1} \cap \alpha_{l_2}$, where $|\alpha_{l_1\cap l_2}| = K \geq R$. Then, the concatenated matrix $\mathbf{S}_r^{(l_1\cap l_2)}\begin{bmatrix}\tilde{\mathbf{X}}_{:l_1:} & \tilde{\mathbf{X}}_{:l_2:}\end{bmatrix} \in \mathbb{R}^{K\times 2I_N}$, where the row selection matrix $\mathbf{S}_r^{(l_1\cap l_2)}$ selects rows indexed by $\alpha_{l_1\cap l_2}$, can also serve as an additional observed submatrix in the subspace computation approaches discussed in Section 3.1.1 and Section 3.1.2. As a result, the column subspaces can be estimated more accurately by incorporating these additional submatrices. However, this approach comes with additional computational cost as it requires identifying the overlapping slices. Computing the SVDs of these larger matrices can also be relatively expensive, but the results are expected to be close to those obtained by optimization methods. In the remainder of this section, we demonstrate this through an experiment. We compute the (TT) approximation of an incomplete tensor observed fiber-wise using the algebraic algorithm with two different approaches: one that utilizes only individual slices and another that incorporates both individual mode-2 slices and the largest observed submatrix from each pair of slices.

A 4th-order tensor in TT format $\mathcal{X} \in \mathbb{R}^{16\times 16\times 16\times 16}$ is generated with a fixed $\text{rank}_{TT}(\mathcal{X}) = (1, 3, 3, 4, 1)$. Random Gaussian i.i.d. noise is then added to the generated tensor with SNR varying from 0 to 45 dB. Sixty percent of the mode-4 fibers are randomly removed from the noisy tensor. Completion is then performed using the algebraic algorithm with the two strategies. Figure 12 shows that using the additional observed submatrices significantly improves accuracy, bringing it close to that of TT-WOPT. However, this approach is slightly slower due to the extra computational overhead as mentioned previously. Overall, the computational cost remains low compared to TT-WOPT, while the accuracy gap is very small. It's worth noting that, in this experiment, we have only combined pairs of slices; however, it may be possible to achieve accuracy more closer to TT-WOPT by selecting triplets or more slices.

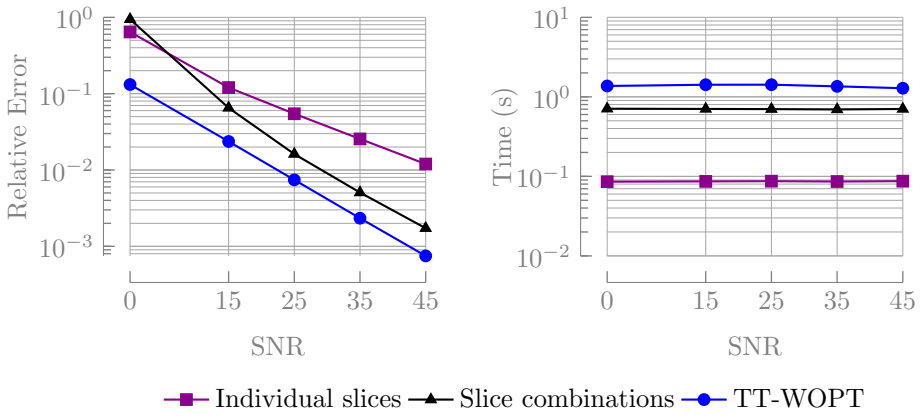


Fig. 12: (left) Accuracy of the algebraic method increases as observed submatrices spanning pairs of slices are included in the computation of the column spaces of the matrix unfoldings, and it becomes close to that of TT-WOPT. (right) However, this improvement comes with additional computational cost.

5 Conclusion

We introduced an algebraic framework for computing the TT decomposition of an incomplete tensor observed fiber-wise (along a single specific mode). This framework builds on established algebraic techniques known for CPD and MLSVD [28, 30]. The TT decomposition combines the features of MLSVD and CPD, making the extension of such algebraic methods in TT format highly consequential. The proposed approach relies solely on standard NLA operations and is fast, guaranteed to work under reasonable deterministic conditions on the observation pattern.

We have provided theoretical insights into piecewise subspace learning, an essential ingredient of our method, and discussed both the algebraic and generic uniqueness conditions for retrieving the TT cores (up to basis transformation).

Convincing numerical experiments demonstrate that our proposed approach is practical and valuable for real-life applications. When compared with state-of-the-art methods, our method has been observed to be fast, in line with expectations, while achieving competitive accuracy in recovering partially (fiber-wise) observed tensors. Moreover, our experiments show that the solution obtained from the proposed method can also serve as a proxy for efficient subsequent computations, including the initialization of optimization-based methods, which are typically more computationally expensive.

Declarations

- **Funding:** This work was supported by the Flemish Government's AI Research Program and KU Leuven Internal Funds (iBOF/23/064, C14/22/096, IDN/19/014). Lieven De Lathauwer and Shakir Showkat Sofi are affiliated with Leuven.AI - KU Leuven institute for AI, B-3000, Leuven, Belgium.

- **Code availability:** The source code for the proposed algorithm, along with all data necessary to reproduce the results, will be made available at https://www.tensorlabplus.net/.
- **Conflict of interest:** Not applicable.

References

[1] Sofi, S.S., Hendrikx, S., De Lathauwer, L.: Tensor train completion of multi-way data observed along one mode. In: Proceedings of the 32nd EUSIPCO, pp. 1067–1071 (2024)

[2] Oseledets, I.V., Tyrtyshnikov, E.E.: Breaking the curse of dimensionality, or how to use SVD in many dimensions. SIAM J. Sci. Comput. **31**, 3744–3759 (2009)

[3] Hackbusch, W.: Tensor Spaces and Numerical Tensor Calculus vol. 42. Springer, Heidelberg (2012)

[4] Grasedyck, L., Kressner, D., Tobler, C.: A literature survey of low-rank tensor approximation techniques. GAMM Mitt. **36**(1), 53–78 (2013)

[5] Sidiropoulos, N.D., De Lathauwer, L., Fu, X., Huang, K., Papalexakis, E.E., Faloutsos, C.: Tensor decomposition for signal processing and machine learning. IEEE Trans. Signal Process. **65**(13), 3551–3582 (2017)

[6] Khoromskij, B.N.: Tensor Numerical Methods in Scientific Computing vol. 19. De Gruyter, Berlin/Boston (2018)

[7] Khoromskaia, V., Khoromskij, B.N.: Tensor Numerical Methods in Quantum Chemistry. De Gruyter, Berlin/Boston (2018)

[8] Lim, L.-H.: Tensors in computations. Acta Numerica **30**, 555–764 (2021)

[9] Franc, A.: Tensor ranks for the pedestrian for dimension reduction and disentangling interactions. arXiv preprint arXiv:2201.07473 (2022)

[10] Ballard, G., Kolda, T.G.: Tensor Decompositions for Data Science. Cambridge University Press, Cambridge (2025)

[11] Tokcan, N., Sofi, S.S., Pham, V.T., Prévost, C., Kharbech, S., Magnier, B., Nguyen, T.P., Zniyed, Y., De Lathauwer, L.: Tensor decompositions for signal processing: Theory, advances, and applications. Signal Process., 110191 (2025)

[12] Little, R.J., Rubin, D.B.: Statistical Analysis with Missing Data, 3rd edn. Wiley, Hoboken, NJ (2019)

[13] Liu, J., Musialski, P., Wonka, P., Ye, J.: Tensor completion for estimating missing values in visual data. IEEE Trans. Pattern Anal. Mach. Intell. **35**, 208–220 (2013)

[14] Song, Q., Ge, H., Caverlee, J., Hu, X.: Tensor completion algorithms in big data analytics. ACM Trans. Knowl. Discov. Data **13**, 1–48 (2019)

[15] Yuan, L., Zhao, Q., Gui, L., Cao, J.: High-order tensor completion via gradient-based optimization under tensor train format. Signal Process. Image Commun. **73**, 53–61 (2019)

[16] Qiu, P.: Embracing the dropouts in single-cell rna-seq analysis. Nat. Commun. **11**(1), 1169 (2020)

[17] Pan, X., Li, Z., Qin, S., Yu, M., Hu, H.: Sclrtc: imputation for single-cell rna-seq data via low-rank tensor completion. BMC Genom. **22**, 1–19 (2021)

[18] Chen, X., Yang, J., Sun, L.: A nonconvex low-rank tensor completion model for spatiotemporal traffic data imputation. Transp. Res. Part C Emerg. Technol. **117**, 102673 (2020)

[19] Vandenberghe, L., Boyd, S.: Semidefinite programming. SIAM Rev. **38**(1), 49–95 (1996)

[20] Fazel, M., Hindi, H., Boyd, S.: Rank minimization and applications in system theory. In: Proceedings of the 2004 ACC, vol. 4, pp. 3273–3278 (2004). IEEE

[21] Candès, E.J., Tao, T.: The power of convex relaxation: Near-optimal matrix completion. IEEE Trans. Inf. Theory **56**(5), 2053–2080 (2010)

[22] Candès, E., Recht, B.: Exact matrix completion via convex optimization. Commun. ACM **55**, 111–119 (2012)

[23] Kressner, D., Steinlechner, M., Vandereycken, B.: Low-rank tensor completion by Riemannian optimization. BIT Numer. Math. **54**, 447–468 (2014)

[24] Gandy, S., Recht, B., Yamada, I.: Tensor completion and low-n-rank tensor recovery via convex optimization. Inverse Probl. **27**, 025010 (2011)

[25] Bengua, J.A., Phien, H.N., Tuan, H.D., Do, M.N.: Efficient tensor completion for color image and video recovery: Low-rank tensor train. IEEE Trans. Image Process. **26**, 2466–2479 (2017)

[26] Chen, Y., Bhojanapalli, S., Sanghavi, S., Ward, R.: Completing any low-rank matrix, provably. J. Mach. Learn. Res. **16**(94), 2999–3034 (2015)

[27] Chen, X., He, Z., Wang, J.: Spatial-temporal traffic speed patterns discovery and incomplete data recovery via svd-combined tensor decomposition. Transp. Res. Part C Emerg. Technol. **86**, 59–77 (2018)

[28] Sørensen, M., De Lathauwer, L.: Fiber sampling approach to canonical polyadic decomposition and application to tensor completion. SIAM J. Matrix Anal. Appl. **40**, 888–917 (2019)

[29] Hendrikx, S., Sørensen, M., De Lathauwer, L.: Multilinear singular value decomposition of a tensor with fibers observed along one mode. In: Proceedings of the 2023 SSP, pp. 566–570 (2023). IEEE

[30] Sørensen, M., Hendrikx, S., De Lathauwer, L.: Multilinear singular value decomposition–based completion with fibers observed in a single mode. SIAM J. Matrix Anal. Appl. **46**(2), 1061–1090 (2025)

[31] Sofi, S.S., Oseledets, I.: A case study of spatiotemporal forecasting techniques for weather forecasting. GeoInformatica **29**(2), 275–298 (2025)

[32] Jacobs, D.W.: Linear fitting with missing data for structure-from-motion. Comput. Vis. Image Underst. **82**, 57–81 (2001)

[33] Jia, H., Martinez, A.M.: Low-rank matrix fitting based on subspace perturbation analysis with applications to structure from motion. IEEE Trans. Pattern Anal. Mach. Intell. **31**(5), 841–854 (2009)

[34] J.Király, F., Theran, L., Tomioka, R.: The algebraic combinatorial approach for low-rank matrix completion. J. Mach. Learn. Res. **16**, 1391–1436 (2015)

[35] Pimentel-Alarcón, D.L., Boston, N., Nowak, R.D.: A characterization of deterministic sampling patterns for low-rank matrix completion. IEEE J. Sel. Top. Signal Process. **10**, 623–636 (2016)

[36] Oseledets, I.: Tensor-train decomposition. SIAM J. Sci. Comput. **33**, 2295–2317 (2011)

[37] Oseledets, I.V., Tyrtyshnikov, E.: TT-cross approximation for multidimensional arrays. Linear Algebra Appl. **432**, 70–88 (2010)

[38] Vervliet, N., Debals, O., De Lathauwer, L.: Exploiting efficient representations in large-scale tensor decompositions. SIAM J. Sci. Comput. **41**, 789–815 (2019)

[39] Perez-Garcia, D., Verstraete, F., Wolf, M.M., Cirac, J.I.: Matrix product state representations. Quantum Inf. Comput. **7**, 401–430 (2007)

[40] De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. **21**, 1253–1278 (2000)

[41] Shi, T., Ruth, M., Townsend, A.: Parallel algorithms for computing the tensor-train decomposition. SIAM J. Sci. Comput. **45**, 101–130 (2023)

[42] Mahoney, M.W.: Randomized algorithms for matrices and data. Found. Trends Mach. Learn. **3**, 123–224 (2010)

[43] Halko, N., Martinsson, P.G., Tropp, J.A.: Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev. **53**, 217–288 (2011)

[44] Pimentel-Alarcón, D.L., Boston, N., Nowak, R.D.: Deterministic conditions for subspace identifiability from incomplete sampling. In: 2015 IEEE International Symposium on Information Theory, pp. 2191–2195 (2015)

[45] Bostian, A.A., Woerdeman, H.J.: Unicity of minimal rank completions for tri-diagonal partial block matrices. Linear Algebra Appl. **325**(1), 23–55 (2001)

[46] Kaashoek, M.A., Woerdeman, H.J.: Unique minimal rank extensions of triangular operators. J. Math. Anal. Appl **131**(2), 501–516 (1988)

[47] Epperly, E.N., Govindarajan, N., Chandrasekaran, S.: Minimal rank completions for overlapping blocks. Linear Algebra Appl. **627**, 185–198 (2021)

[48] Ben-Israel, A.: Projectors on intersections of subspaces. Contemp. Math. **636**, 41–50 (2015)

[49] Yan, F., Wang, J., Liu, S., Jin, M., Shen, Y.: Svd-based low-complexity methods for computing the intersection of $k \geq 2$ subspaces. Chinese Journal of Electronics **28**(2), 430–436 (2019)

[50] Sørensen, M., Kanatsoulis, C.I., Sidiropoulos, N.D.: Generalized canonical correlation analysis: A subspace intersection approach. IEEE Trans. Signal Process. **69**, 2452–2467 (2021)

[51] Brand, M.: Incremental singular value decomposition of uncertain data with missing values. In: Computer Vision – ECCV 2002, pp. 707–720. Springer, Berlin/Heidelberg (2002)

[52] Matthew, B.: Fast low-rank modifications of the thin singular value decomposition. Linear Algebra Appl. **415**(1), 20–30 (2006)

[53] Cai, J.-F., Candès, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM J. Optim. **20**(4), 1956–1982 (2010)

[54] Vervliet, N., Debals, O., Sorber, L., De Lathauwer, L.: Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis. IEEE Signal Process. Mag. **31**, 71–79 (2014)

[55] Roy, R., Kailath, T.: Esprit-estimation of signal parameters via rotational invariance techniques. IEEE Trans. Acoust. Speech Signal Process. **37**, 984–995 (1989)