

High-order tensor estimation via trains of coupled third-order CP and Tucker decompositions

Yassine Zniyed, Remy Boyer, André de Almeida, Gérard Favier

► **To cite this version:**

Yassine Zniyed, Remy Boyer, André de Almeida, Gérard Favier. High-order tensor estimation via trains of coupled third-order CP and Tucker decompositions. *Linear Algebra and its Applications*, Elsevier, 2020, 588, pp.304-337. hal-02354634

HAL Id: hal-02354634

<https://hal.univ-lille.fr/hal-02354634>

Submitted on 7 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

High-order tensor estimation via trains of coupled third-order CP and Tucker decompositions

Yassine Zniyed^a, Rémy Boyer^a, André L. F. de Almeida^b, Gérard Favier^c

^a*Laboratoire des Signaux et Systèmes (L2S), Université Paris-Sud (UPS), CNRS, CentraleSupélec, Gif-sur-Yvette, France.*

^b*Department of Teleinformatics Engineering, Federal University of Fortaleza, Brazil.*

^c*Laboratoire I3S, Université Côte d'Azur, CNRS, Sophia Antipolis, France.*

Abstract

In this work, equivalence relations between a Tensor Train (TT) decomposition and the Canonical Polyadic Decomposition (CPD)/Tucker Decomposition (TD) are investigated. It is shown that a Q -order tensor following a CPD/TD with $Q > 3$ can be written using the graph-based formalism as a train of Q tensors of order at most 3 following the same decomposition as the initial Q -order tensor. This means that for any practical problem of interest involving the CPD/TD, it exists an equivalent TT-based formulation. This equivalence allows us to overcome the curse of dimensionality when dealing with the big data tensors. In this paper, it is shown that the native difficult optimization problems for CPD/TD of Q -order tensors can be efficiently solved using the TT decomposition according to flexible strategies that involve $Q - 2$ optimization problems with 3-order tensors. This methodology hence involves a number of free parameters linear with Q , and thus allows to mitigate the exponential growth of parameters for Q -order tensors. Then, by capitalizing on the TT decomposition, we also formulate several robust and fast algorithms to accomplish Joint Dimensionality Reduction And Factors rEtirement (JIRAFE) for the CPD/TD. In particular, based on the TT-SVD algorithm, we show how to exploit the existing coupling between two successive TT-cores in the graph-based formalism. The advantages of the proposed solutions in terms of storage cost, computational complexity and factor estimation accuracy are also discussed.

Keywords: Canonical polyadic decomposition, Tucker decomposition, HOSVD, tensor trains, structured tensors.

^{*}Conference version of part of this work appeared in [45]

Email addresses: yassine.zniyed@l2s.centralesupelec.fr (Yassine Zniyed), remy.boyer@l2s.centralesupelec.fr (Rémy Boyer), andre@gtel.ufc.br (André L. F. de Almeida), favier@i3s.unice.fr (Gérard Favier)

1. Introduction

Low rank tensor approximation is one of the major challenges in the information scientific community, see for instance [35, 8]. It is useful to extract relevant information confined into a small dimensional subspace from a massive volume of data while reducing the computational cost. In the matrix case, the approximation of a full rank matrix by a (fixed) low rank matrix is a well posed problem. The famous Eckart-Young theorem [16] provides both theoretical guarantees on the existence of a solution and a convenient way to compute it. Specifically, the set of low rank matrices is closed and the best low rank approximation (in the sense of the Frobenius norm) is obtained via the truncated Singular Value Decomposition (SVD). This fundamental result is at the origin of the Principal Component Analysis (PCA) [18], for instance. PCA is based on the decomposition of a set of observations into a set of uncorrelated variables. When the measurements are naturally modeled according to more than two axes of variations, *i.e.*, in the case of high-order tensors, the problem of obtaining a low rank approximation faces a number of practical and fundamental difficulties. Indeed, even if some aspects of tensor algebra can be considered as mature, several algebraic concepts such as decomposition uniqueness, rank determination, or the notions of singular values and eigenvalues remain challenging research topics [2, 12]. To illustrate these conceptual difficulties and without being too exhaustive, we will address the non-uniqueness of the rank through the description of the canonical rank and of the multilinear rank.

A natural generalization to high-order tensors of the usual concept of matrix rank leads to the canonical polyadic decomposition (CPD) [22, 20, 4, 15, 39]. The canonical rank of a Q -order tensor is equal to the minimal number, say R , of rank-one tensors that must be linearly combined to reach a perfect recovery of the initial tensor. A rank-one tensor of order Q is given by the outer product of Q vectors. In the context of massive data processing and analysis, this decomposition and its variants [5, 6] are attractive in terms of compactness thanks to the minimality constraint on R . In addition, the CPD has remarkable uniqueness properties [39] and involves only QNR free parameters for a Q -order rank- R tensor of size $N \times \dots \times N$. Unfortunately, unlike the matrix case, the set of low-rank tensors is not closed [21]. This singularity implies that the problem of computing the CPD is mathematically ill-posed. The consequence is that its numerical computation remains non trivial and is usually done using suboptimal iterative algorithms [28]. Note that this problem can sometimes be avoided by exploiting some natural hidden structures in the physical model [38].

The Tucker decomposition and the HOSVD (High-Order SVD) [11, 41] are two popular decompositions being an alternative to the CPD. In this case, the notion of canonical rank is no longer relevant and a new rank definition has to be introduced. Specifically, the multilinear rank of a tensor is defined as the set of Q positive integers : $\{T_1, \dots, T_Q\}$ where each T_q is the usual (in the matrix sense) rank of the q -th mode unfolding of this tensor. Its practical construction is algebraic, non-iterative and optimal in the sense of the Eckart-Young theorem, applied to each matrix unfolding. This approach became popular because

it can be computed in real time or adaptively, using very standard algebraic methods [3]. However, a low (multilinear) rank tensor based on the HOSVD is generally not optimal regarding the approximation in the Frobenius norm sense. In other words, there is no generalization of the Eckart-Young theorem

 50 for tensors of order strictly greater than two. This decomposition has a poor compactness property compared to the CPD. For a multilinear rank $\{T, \dots, T\}$, the number of free parameters is about $QNT + T^Q$ for a Q -order tensor of size $N \times \dots \times N$, and therefore grows exponentially with the order Q . In the context of the massive data processing, this decomposition is irrelevant and cannot

 55 break the curse of dimensionality [33]. Indeed, a generally not sparse core tensor must be stored in the Tucker/HOSVD decomposition, leading to a term in T^Q . A new family of tensor decompositions called Tensor Train (TT) [32, 30] has been recently introduced. This tensor decomposition has a twofold benefit: first it increases representation compactness with respect to the Tucker model

 60 and exploits a stable numerical estimation procedure which avoids the iterative algorithms used to compute the CPD. Second, the TT decomposition breaks the curse of dimensionality by operating on lower (≤ 3) order tensors. The TT decomposition relies on the estimation of the TT-ranks denoted by $\{R_1, \dots, R_{Q-1}\}$, and the number of free parameters is $(Q-2)NR^2 + 2NR$ for

 65 $R_1 = \dots = R_{Q-1} = R$. Hence, its storage cost is now linear in Q . Unfortunately, as for the canonical or multilinear ranks, the estimation of the TT-ranks is a very delicate task.

In this work, we study the equivalence between the TT decomposition and both CPD and TD for an estimation purpose. Seminal works [32, 7, 19] have shown

 70 some links between these decompositions. Our work is different, since we are not aiming to provide algebraic equivalences from a modeling point of view. Our results are mainly constructive and focused on the TT-ranks and TT-cores structure when the TT-SVD algorithm is applied to CPD and TD, depending on the rank conditions of the initial tensors, so they can be exploitable in an

 75 estimation scheme. This work brings the following original contributions:

- A different TT-based strategy for multilinear projection estimation applied to TD/HOSVD is proposed.
- We provide an algebraic analysis of the TT-SVD algorithm and propose a new methodology called JIRAFE for Joint dDimensionality Reduction And Factors rEtrieval to solve a sum of coupled least-square criteria for each

 80 TT-core.
- New theorems on the analysis of the application of the TT-SVD algorithm to high-order CPD and TD, regarding the resulting TT-cores structures and the rank conditions, are given.
- Constructive proofs of the structure of the TT-cores are developed, while

 85 proposing fast, stable and parallel algorithms which exploit this structure for both the CPD and TD. A better algorithmic stability refers to either proposing fully closed-form solutions replacing the existing iterative

solutions, or breaking high order iterative solutions by 3-order iterative ones.

90

It is worth noting that this work is essentially centered on the algebraic analysis of the TT-SVD algorithm and the development of the JIRAFE methodology, unlike the publications [32, 7], that discuss the problem in the context of tensor modeling and not in terms of estimation algorithms. Moreover, our methodology is different from the one presented in [31, 14] where two (or more) tensor models are associated to propose new tensor factorizations. In this work, new algorithms for breaking the problem of decomposing a large-scale tensor into coupled small-order tensor factorizations are proposed for standard tensor models, *i.e.*, for CPD and TD/HOSVD, based on constructive proofs. Also, one may note that the algebraic rewritings (Theorems 1 and 3 in this work) of CPD/TD as TT do not allow to guarantee the minimality of the considered TT-ranks, and thus the structure of the TT-cores when a decomposition algorithm is applied. On the contrary, the new constructive results, Theorems 2 and 4 in this work, provide guarantees on the minimality of the TT-ranks and the TT-cores structure, considering additional rank conditions on the initial tensors, and based on the Eckart-Young theorem when the constructive proofs are detailed.

95

100

105

All applications using high-order CPD/TD/HOSVD models can be processed using the proposed JIRAFE framework. For example, but not limited to, the JIRAFE framework can be applied to the problem of multilinear harmonic retrieval (MHR) [23], high-order CPD probability mass function tensors [25] in machine learning, and for channel estimation for massive MIMO systems [24] in wireless communications.

110

The rest of this paper is organized as follows. Section 2 introduces notations and some algebraic background that will be used throughout the article. Section 3 recalls the Tensor Train decomposition and the TT-SVD algorithm. In Section 4, we present the equivalence between TD and TT decompositions, showing that a high order TD is equivalent to a train of 3-order TDs. In Section 5, the equivalence between the CPD and the TT decomposition is studied, and the structure of the new CPD-Train cores is established. Section 6 presents different applications of the TD-Train and CPD-Train models. Stable, fast and scalable algorithms adapted to these new models are also proposed in this section. Finally, Section 7 contains the concluding remarks.

115

120

2. Notations and algebraic background

The notations used throughout this paper are now defined. The symbols $(\cdot)^T$, $(\cdot)^\dagger$ and $\text{rank}(\cdot)$ denote, respectively, the transpose, the pseudo-inverse, and the rank. The Kronecker, Khatri-Rao and n -mode products are denoted to by \otimes , \odot and \times_n . The symbols $\langle \cdot \rangle$, \triangleq and $O(\cdot)$ denote spanned subspace, an equality by definition, and a dominant term in a complexity expression. The operator $\text{diag}(\cdot)$ forms a diagonal matrix from its vector argument. The operator $\text{vec}(\cdot)$ forms a vector by stacking the columns of its matrix argument,

while $\text{unvec}_{I \times J}(\cdot)$ is the inverse operator. $\mathbf{e}_n^{(N)}$ denotes the n th basis vector of \mathbb{R}^N . We denote by $\mathbf{A}(:, i)$ (*resp.* $\mathbf{A}(i, :)$) the i th column (*resp.* row) of the matrix \mathbf{A} . $\mathcal{X}(i, :, :)$, $\mathcal{X}(:, j, :)$ and $\mathcal{X}(:, :, k)$ are the i -th horizontal, j -th lateral and k -th frontal slices of sizes $N_2 \times N_3$, $N_1 \times N_3$ and $N_1 \times N_2$, respectively, of the tensor \mathcal{X} of size $N_1 \times N_2 \times N_3$. Scalars, vectors, matrices and tensors are represented by x , \mathbf{x} , \mathbf{X} and \mathcal{X} , respectively. $\text{unfold}_q \mathcal{X}$ refers to the unfolding of tensor \mathcal{X} over its q -th mode. $\mathcal{I}_{k,R}$ denotes the k -order identity tensor of size $R \times \dots \times R$. It is a hypercubic tensor of order k , with ones on its diagonal and zeros otherwise, and we have $\mathcal{I}_{2,R} = \mathbf{I}_R$.

Let $\mathbf{A} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \dots \ \mathbf{a}_R]$ and $\mathbf{B} = [\mathbf{b}_1 \ \mathbf{b}_2 \ \dots \ \mathbf{b}_R]$. Their Khatri-Rao product is defined as

$$\mathbf{A} \odot \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1 \ \mathbf{a}_2 \otimes \mathbf{b}_2 \ \dots \ \mathbf{a}_R \otimes \mathbf{b}_R].$$

125 We now introduce some standard lemmas and definitions that will be useful in the sequel, especially for the proofs of our main Theorems 2 and 4.

Definition 1. Let $R \leq N$. If matrices \mathbf{A} and \mathbf{B} of size $N \times R$ span the same R -dimensional column space, then it exists a nonsingular $R \times R$ matrix \mathbf{M} , usually called a *change-of-basis* matrix, such as

$$\mathbf{B} = \mathbf{A}\mathbf{M} \quad \text{or equivalently} \quad \mathbf{B}\mathbf{M}^{-1} = \mathbf{A}.$$

In this case, we have the following equality in terms of orthogonal projectors

$$\mathbf{B}\mathbf{B}^\dagger = \mathbf{A}\mathbf{M}\mathbf{M}^{-1}\mathbf{A}^\dagger = \mathbf{A}\mathbf{A}^\dagger.$$

Lemma 1. Let \mathbf{A} and \mathbf{B} be two matrices of respective sizes $I \times J$ and $K \times J$, where $I \geq J$, and $K \geq J$. The Khatri-Rao product of both matrices does not decrease the rank [36]:

$$\text{rank}(\mathbf{A} \odot \mathbf{B}) \geq \max\{\text{rank}(\mathbf{A}), \text{rank}(\mathbf{B})\},$$

which implies that, if \mathbf{A} or \mathbf{B} has full rank, $\mathbf{A} \odot \mathbf{B}$ also has full rank, i.e.,

$$\text{rank}(\mathbf{A}) = J \quad \text{or} \quad \text{rank}(\mathbf{B}) = J \Rightarrow \text{rank}(\mathbf{A} \odot \mathbf{B}) = J.$$

Definition 2. The contraction product \times_q^p between two tensors \mathcal{A} and \mathcal{B} of size $N_1 \times \dots \times N_Q$ and $M_1 \times \dots \times M_P$, where $N_q = M_p$, is a tensor of order $Q + P - 2$ such as [5]

$$\begin{aligned} & [\mathcal{A} \times_q^p \mathcal{B}]_{n_1, \dots, n_{q-1}, n_{q+1}, \dots, n_Q, m_1, \dots, m_{p-1}, m_{p+1}, \dots, m_P} \\ &= \sum_{k=1}^{N_q} [\mathcal{A}]_{n_1, \dots, n_{q-1}, k, n_{q+1}, \dots, n_Q} [\mathcal{B}]_{m_1, \dots, m_{p-1}, k, m_{p+1}, \dots, m_P}. \end{aligned}$$

In tensor-based data processing, it is standard to unfold a tensor into matrices. We refer to Eq. (5) in [17], for a general matrix unfolding formula, also called tensor reshaping. The q -th generalized unfolding $\mathbf{X}_{(q)}$, of size $(\prod_{s=1}^q N_s) \times$

($\prod_{s=q+1}^Q N_s$), of the tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times \dots \times N_Q}$ using the native reshape function of the MATLAB software, is defined by:

$$\mathbf{X}_{(q)} = \text{reshape} \left(\mathcal{X}; \prod_{s=1}^q N_s, \prod_{s=q+1}^Q N_s \right).$$

Another type of reshaping is the tensor unfolding, that transforms the tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times \dots \times N_Q}$ into a 3-order tensor \mathcal{X}_q of size $(N_1 \cdots N_{q-1}) \times N_q \times (N_{q+1} \cdots N_Q)$ where:

$$\mathcal{X}_q = \text{reshape} \left(\mathcal{X}; \prod_{s=1}^{q-1} N_s, N_q, \prod_{s=q+1}^Q N_s \right).$$

The last transformation is the square matrix tensorization, that transforms a square matrix $\mathbf{I} \in \mathbb{R}^{N \times N}$, with $N = N_1 N_2$, to a 3-order tensor according to:

$$\mathcal{X} = \text{reshape}(\mathbf{I}; N_1, N_2, N), \quad (1)$$

or,

$$\bar{\mathcal{X}} = \text{reshape}(\mathbf{I}; N, N_1, N_2), \quad (2)$$

where $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N}$ and $\bar{\mathcal{X}} \in \mathbb{R}^{N \times N_1 \times N_2}$. These two last reshapings will be used in Theorem 1 to define a one-to-one mapping between an identity matrix and a sparse tensor filled with binary values.

130 2.1. Tucker decomposition

The Tucker decomposition was proposed in [41]. It decomposes a tensor into a core tensor of same order, multiplied by a factor matrix along each mode. It can be seen as a generalization of the CPD [22, 20, 4].

Definition 3. A Q -order tensor of size $N_1 \times \dots \times N_Q$ that follows a Tucker decomposition can be written as:

$$\mathcal{X} = \mathcal{C} \times_1 \mathbf{F}_1 \times_2 \mathbf{F}_2 \times_3 \dots \times_Q \mathbf{F}_Q \quad (3)$$

135 where \mathbf{F}_q is of size $N_q \times T_q$, $1 \leq q \leq Q$, and \mathcal{C} is the core tensor of size $T_1 \times \dots \times T_Q$. The multilinear rank [11] of the tensor \mathcal{X} is defined as the Q -uplet $\{T_1, \dots, T_Q\}$, such that $T_1 \times \dots \times T_Q$ is the minimal possible size of the core tensor \mathcal{C} . The storage cost of a Tucker decomposition is $O(QNT + T^Q)$, where $N = \max\{N_1, \dots, N_Q\}$, and $T = \max\{T_1, \dots, T_Q\}$.

2.2. HOSVD

Definition 4. The HOSVD [11, 1] is a special case of the Tucker decomposition (3), where the factors \mathbf{F}_q are column-orthonormal. Generally, the HOSVD is not optimal in terms of low rank approximation, but it relies on the stable (*i.e.*, non-iterative), SVD-based algorithm. In the real case, the core tensor can be expressed as

$$\mathbf{C} = \mathcal{X} \times_1 \mathbf{F}_1^T \times_2 \mathbf{F}_2^T \times_3 \dots \times_Q \mathbf{F}_Q^T$$

140 Note that the core tensor \mathbf{C} obeys to the “all-orthogonality” property [11].

Lemma 2. *The q -th generalized unfolding, denoted by $\mathbf{X}_{(q)}$, of a tensor \mathcal{X} that follows (3), of size $(N_1 \dots N_q) \times (N_{q+1} \dots N_Q)$, admits the following expression:*

$$\begin{aligned} \mathbf{X}_{(q)} &= \text{reshape}(\mathcal{X}; N_1 \dots N_q, N_{q+1} \dots N_Q) \\ &= (\mathbf{F}_q \otimes \dots \otimes \mathbf{F}_1) \cdot \mathbf{C}_{(q)} \cdot (\mathbf{F}_Q \otimes \dots \otimes \mathbf{F}_{q+1})^T, \end{aligned}$$

where $\mathbf{C}_{(q)} = \text{reshape}(\mathbf{C}; T_1 \dots T_q, T_{q+1} \dots T_Q)$.

2.3. CPD

The CPD expresses a Q -order tensor \mathcal{X} into a sum of rank-one Q -order tensors [22, 20]. The canonical rank of a CPD is the minimum number of rank-
145 one tensors needed for a perfect representation of \mathcal{X} .

Definition 5. A Q -order tensor of size $N_1 \times \dots \times N_Q$ belonging to the family of rank- R CPD admits the following decomposition:

$$\mathcal{X} = \mathcal{I}_{Q,R} \times_1 \mathbf{P}_1 \times_2 \mathbf{P}_2 \times_3 \dots \times_Q \mathbf{P}_Q \quad (4)$$

where the q -th factor \mathbf{P}_q is of size $N_q \times R$, $1 \leq q \leq Q$. The q -mode unfolded matrix $\text{unfold}_q \mathcal{X}$, of size $N_q \times \frac{N_1 \dots N_Q}{N_q}$, is given by:

$$\text{unfold}_q \mathcal{X} = \mathbf{P}_q \cdot (\mathbf{P}_Q \odot \dots \odot \mathbf{P}_{q+1} \odot \mathbf{P}_{q-1} \odot \dots \odot \mathbf{P}_1)^T.$$

Lemma 3. *For the CPD (4), the q -th unfolding, denoted by $\mathbf{X}_{(q)}$, of size $(N_1 \dots N_q) \times (N_{q+1} \dots N_Q)$, admits the following expression:*

$$\begin{aligned} \mathbf{X}_{(q)} &= \text{reshape}(\mathcal{X}; N_1 \dots N_q, N_{q+1} \dots N_Q) \\ &= (\mathbf{P}_q \odot \dots \odot \mathbf{P}_1) \cdot (\mathbf{P}_Q \odot \dots \odot \mathbf{P}_{q+1})^T. \end{aligned}$$

3. Tensor Train decomposition

The Tensor-Train decomposition was proposed in [32, 30]. It transforms a high Q -order tensor into a set of 3-order core tensors. It is one of the simplest tensor networks that allows to break the “curse of dimensionality”. Indeed, it
150 has a low storage cost, and the number of its parameters is linear in Q .

Definition 6. Let $\{R_1, \dots, R_{Q-1}\}$ be the TT-ranks with bounding conditions $R_0 = R_Q = 1$. A Q -order tensor of size $N_1 \times \dots \times N_Q$ admits a decomposition into a train of tensors if

$$\boldsymbol{\mathcal{X}} = \mathbf{G}_1 \times_2^1 \mathcal{G}_2 \times_3^1 \mathcal{G}_3 \times_4^1 \dots \times_{Q-1}^1 \mathcal{G}_{Q-1} \times_Q^1 \mathbf{G}_Q, \quad (5)$$

with the TT-cores of dimensions

$$\begin{aligned} \mathbf{G}_1 &: N_1 \times R_1, \\ \mathbf{G}_Q^T &: N_Q \times R_{Q-1}, \\ \mathcal{G}_q &: R_{q-1} \times N_q \times R_q, \text{ for } 2 \leq q \leq Q-1, \end{aligned}$$

where $\text{rank}(\mathbf{G}_1) = R_1$, $\text{rank}(\mathbf{G}_Q) = R_{Q-1}$ and for $2 \leq q \leq Q-1$,

$$\begin{aligned} \text{rank}(\mathcal{G}_{q(1)}) &= R_{q-1}, \\ \text{rank}(\mathcal{G}_{q(2)}) &= R_q. \end{aligned}$$

A graph-based representation of the TT decomposition of a Q -order tensor is given in Fig. 1. The TT-cores can be computed using the TT-SVD algo-

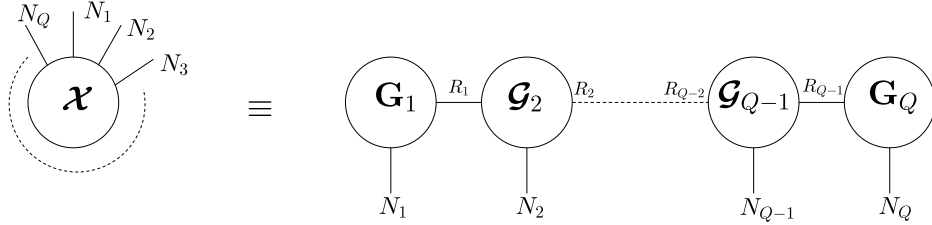


Figure 1: TT decomposition of a Q -order tensor

rithm [32]. It is a sequential algorithm based on the extraction of the dominant subspaces of sequential matrix-based reshapings. The complete algorithmic description is given in [32]. It is also worth noting that the TTD of $\boldsymbol{\mathcal{X}}$ in (5) is not unique, since we can always replace two successive TT-cores \mathcal{G}_q and \mathcal{G}_{q+1} , respectively, by \mathcal{G}'_q and \mathcal{G}'_{q+1} such that

$$\begin{aligned} \mathcal{G}'_q &= \mathcal{G}_q \times_3^1 \mathbf{M}_q^{-1}, \\ \mathcal{G}'_{q+1} &= \mathbf{M}_q \times_2^1 \mathcal{G}_{q+1}, \end{aligned}$$

to recover the same tensor $\boldsymbol{\mathcal{X}}$, where \mathbf{M}_q is a nonsingular matrix of size $R_q \times R_q$. This means that the multiplicative ambiguities in the TTD correspond to post- and pre-multiplications by nonsingular matrices.

4. TD-Train model: Equivalence between a high-order TD and a train of low-order TDs

In Theorem 1, we present a new algebraic equivalence between the Tucker and TT decompositions. We show how the matrix factors and the core tensor of a TD can be recast into the TT format.

Theorem 1. *Assume that tensor \mathcal{X} follows a Q -order Tucker model of multilinear rank- (T_1, \dots, T_Q) , given by eq. (3). A TT decomposition (5) of \mathcal{X} is then given by*

$$\begin{aligned}
\mathbf{G}_1 &= \mathbf{F}_1, \\
\mathcal{G}_q &= \mathcal{T}_q \times_2 \mathbf{F}_q, (1 < q < \bar{q}) \\
&\quad \text{with } \mathcal{T}_q = \text{reshape}(\mathbf{I}_{R_q}; T_1 \cdots T_{q-1}, T_q, T_1 \cdots T_q) \\
\mathcal{G}_{\bar{q}} &= \mathcal{C}_{\bar{q}} \times_2 \mathbf{F}_{\bar{q}}, \\
&\quad \text{with } \mathcal{C}_{\bar{q}} = \text{reshape}(\mathcal{C}; R_{\bar{q}-1}, T_{\bar{q}}, R_{\bar{q}}) \\
\mathcal{G}_q &= \bar{\mathcal{T}}_q \times_2 \mathbf{F}_q, (\bar{q} < q < Q) \\
&\quad \text{with } \bar{\mathcal{T}}_q = \text{reshape}(\mathbf{I}_{R_{q-1}}; T_q \cdots T_Q, T_q, T_{q+1} \cdots T_Q) \\
\mathbf{G}_Q &= \mathbf{F}_Q^T,
\end{aligned}$$

where \mathcal{T}_q and $\bar{\mathcal{T}}_q$ result from the tensorization as defined in (1) and (2), respectively, and

- \bar{q} is the smallest q that verifies $\prod_{i=1}^q T_i \geq \prod_{i=q+1}^Q T_i$,
- The TT-ranks verify: $R_q = \min(\prod_{i=1}^q T_i, \prod_{i=q+1}^Q T_i)$.

PROOF. It is straightforward to verify that the TT decomposition of the Q -order Tucker core \mathcal{C} takes the following expression:

$$\mathcal{C} = \mathbf{I}_{R_1} \times_2^1 \mathcal{T}_2 \times_3^1 \cdots \times_{\bar{q}-1}^1 \mathcal{T}_{\bar{q}-1} \times_{\bar{q}}^1 \mathcal{C}_{\bar{q}} \times_{\bar{q}+1}^1 \bar{\mathcal{T}}_{\bar{q}+1} \times_{\bar{q}+2}^1 \cdots \times_{Q-1}^1 \bar{\mathcal{T}}_{Q-1} \times_Q^1 \mathbf{I}_{R_{Q-1}} \quad (6)$$

where tensors \mathcal{T}_q and $\bar{\mathcal{T}}_q$ have been defined in the Theorem and thanks to the reshaping eq. (1) and eq. (2), respectively. Replacing the TT decomposition of \mathcal{C} in eq. (3), the q -th 3-order tensor in eq. (6) is multiplied in its second mode by its corresponding factor \mathbf{F}_q . By identifying the final TT-cores, the theorem is proved.

Note that all the TT-cores follow a 3-order Tucker1 model, whose the second factor is the q -th factor of the original TD, hence the name TD-Train. More specifically, for $q < \bar{q}$ and $q > \bar{q}$, the corresponding TT cores have Tucker1 structures whose core tensors have fixed 1's and 0's patterns. We recall that

a Tucker1 model is a 3-order Tucker decomposition with two factor matrices equal to identity matrices. For $q = \bar{q}$, the associated TT-core has a Tucker1 structure with a core tensor $\mathbf{C}_{\bar{q}}$ obtained from the core tensor of the original TD. To illustrate this result, consider the TT decomposition of the core tensor \mathbf{C} in eq. (3). Note that \bar{q} corresponds to the smallest q such that $\mathbf{C}_{(q)}$ has at least as many rows as columns. For example, if \mathbf{C} is a 5-order tensor of size $2 \times 3 \times 2 \times 4 \times 2$, then $\bar{q} = 3$, since $2 \times 3 \times 2 > 4 \times 2$. Another example where \mathbf{C} is of size $2 \times 2 \times 3 \times 4 \times 4$ corresponds to $\bar{q} = 4$.

The cores of the TD-Train can be recovered by using the TT-SVD algorithm [32] in practice. However, the application of this algorithm will recover the cores up to nonsingular transformation matrices according to definition 1. In the following theorem, the structures of the TT-cores associated with a TD-Train are given.

Theorem 2. Applying the TT-SVD algorithm on the tensor (3), under the assumptions that

- \mathbf{F}_q is a full column rank matrix of size $N_q \times T_q$,
- $\mathbf{C}_{(q)} = \text{reshape}(\mathbf{C}; \prod_{i=1}^q T_i, \prod_{i=q+1}^Q T_i)$ has full rank,

allows to recover TT-cores according to

$$\begin{aligned}
\mathbf{G}_1 &= \mathbf{F}_1 \mathbf{M}_1^{-1}, \\
\mathbf{G}_q &= \mathcal{T}_q \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{F}_q \times_3 \mathbf{M}_q^{-T}, \quad (1 < q < \bar{q}) \\
\mathbf{G}_{\bar{q}} &= \mathbf{C}_{\bar{q}} \times_1 \mathbf{M}_{\bar{q}-1} \times_2 \mathbf{F}_{\bar{q}} \times_3 \mathbf{M}_{\bar{q}}^{-T}, \\
\mathbf{G}_q &= \bar{\mathcal{T}}_q \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{F}_q \times_3 \mathbf{M}_q^{-T}, \quad (\bar{q} < q < Q) \\
\mathbf{G}_Q &= \mathbf{M}_{Q-1} \mathbf{F}_Q^T,
\end{aligned} \tag{7}$$

where \mathbf{M}_q is a $R_q \times R_q$ nonsingular change-of-basis matrix, and the quantities \mathcal{T}_q , $\bar{\mathcal{T}}_q$, $\mathbf{C}_{\bar{q}}$ and R_q are defined in Theorem 1.

PROOF. See Appendix 8.1

Note that the TT-cores follow Tucker models with nonsingular transformation matrices along their first and third modes. These matrices compensate each other due to the train format. The proof relies on the TT-SVD algorithm applied to a Q -order Tucker decomposition.

Remark 1. One may note that, in Theorem 1, no assumptions on the rank of the Tucker core nor the factors are made, thus there is no guarantee that the considered TT-ranks are minimal, *i.e.*, applying a decomposition algorithm such as TT-SVD may estimate different (lower) TT-ranks and provide different TT-cores' structure. In Theorem 2, additional rank assumptions are made, especially on the Tucker core \mathbf{C} , allowing to guarantee the minimality of the given TT-ranks and to provide constructive/exploitable results from an estimation point of view.

205 **5. CPD-Train model: Equivalence between a high-order CPD and a train of low-order CPD(s)**

Following the same reasoning as in the previous section, we will draw results about the relation between the CPD and the TT decomposition. The CPD-Train equivalence turns out to be very useful and of broad interest, due to the various applications of the CPD. Note that the idea of rewriting a CPD into the TT format was briefly introduced in seminal works [32, 7]. In [7], Section 4.3 presents a similar result as Theorem 3 without discussing the TT-cores structure after the application of a decomposition algorithm such as the TT-SVD algorithm. In this paper, we exploit this idea to propose a new CPD factor retrieval algorithm. Indeed, in Theorem 4, which is the main result to derive the structure of all the proposed algorithms, we discuss and expose the TT-cores structure after the application of the TT-SVD algorithm on a CPD tensor when the factor have full column rank. Moreover, we give a rigorous demonstration for the relation between TT and CPD in an algebraic point of view, and more important, we expose, for the first time, as a constructive proof an useful relation between TT and CPD in the context of the TT-SVD algorithm. Specifically, the coupling factors between consecutive TT-cores takes its explication in the TT-SVD. This important property totally inspires the architecture of the proposed algorithms. To the best of the authors' knowledge, this is the first CPD algorithm that exploits such a link. In the following theorem, we provide an algebraic equivalence between the two decompositions.

Theorem 3. *If the tensor \mathcal{X} follows a Q -order CPD of rank- R according to (4), then a TT decomposition (5) is given by [32, 7]:*

$$\begin{aligned} \mathbf{G}_1 &= \mathbf{P}_1, \\ \mathcal{G}_q &= \mathcal{I}_{3,R} \times_2 \mathbf{P}_q \quad (3\text{-order CPD}), \text{ where } 2 \leq q \leq Q-1, \\ \mathbf{G}_Q &= \mathbf{P}_Q^T, \end{aligned}$$

and the TT-ranks are all identical and equal to the canonical rank R .

PROOF. The TT decomposition of the Q -order identity tensor $\mathcal{I}_{Q,R}$ of size $R \times \dots \times R$ involved in the CPD given by (4) is

$$\begin{aligned} \mathcal{I}_{Q,R}(i_1, i_2, \dots, i_Q) &= \sum_{r_1, \dots, r_{Q-1}=1}^R \mathbf{I}_R(i_1, r_1) \mathcal{I}_{3,R}(r_1, i_2, r_2) \mathcal{I}_{3,R}(r_2, i_3, r_3) \dots \\ &\quad \dots \mathcal{I}_{3,R}(r_{Q-2}, i_{Q-1}, r_{Q-1}) \mathbf{I}_R(r_{Q-1}, i_Q), \end{aligned}$$

$i_1, \dots, i_Q, \in [1, R]$, which can be rewritten in a train format using identity matrices and tensors as

$$\mathcal{I}_{Q,R} = \mathbf{I}_R \times_2^1 \mathcal{I}_{3,R} \times_3^1 \dots \times_{Q-1}^1 \mathcal{I}_{3,R} \times_Q^1 \mathbf{I}_R.$$

Replacing the above TT decomposition in (4), and identifying the TT-cores, we can deduce the result of Theorem 3.

It is worth noting that both decompositions have the same number of free parameters. All the TT-cores follow a 3-order rank- R CPD structure, whose the second factor is the q -th factor of the original CPD, the other factors being equal to the identity matrix \mathbf{I}_R . The result of Theorem 3 means that computing a CPD via its associated TT decomposition maintains the model structure while reducing its complexity. In addition, we know the whole structure of the CPD-Train cores, as well as the ranks of the associated 3-order CPDs.

In the same perspective, the application of the TT-SVD algorithm to (4) allows to recover TT-cores up to nonsingular matrices following Definition 1, as established in Theorem 4. It is very important to note that the non-uniqueness property of the TT decomposition suggest that there exists some indeterminate invertible matrices, but in the following theorem, we rigorously, show that these matrices in the context of the TT-SVD algorithm are in fact a set of change-of-basis rank- R matrices essentially linked to the fundamental problem of singular subspace estimation via the SVD. Moreover, it is also important to note that the structure of the TT-cores presented in Theorem 4 is essentially related to the assumption that the factors have full column rank. For instance, in the case of full row rank factors, the structure of the TT-cores will completely change, and we may have 2 factors that are absorbed in the same TT-core, unlike the case we consider and demonstrate in Theorem 4, which shows that the structure of the resultant TT-cores is not trivial and that each separate case needs a new analysis.

Theorem 4. *Applying the TT-SVD algorithm to the Q -order CPD (4) allows to recover TT-cores of (4) when the factors \mathbf{P}_q have full column rank, yielding*

$$\begin{aligned}\mathbf{G}_1 &= \mathbf{P}_1 \mathbf{M}_1^{-1}, \\ \mathcal{G}_q &= \mathcal{I}_{3,R} \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{P}_q \times_3 \mathbf{M}_q^{-T}, \text{ where } 2 \leq q \leq Q-1 \\ \mathbf{G}_Q &= \mathbf{M}_{Q-1} \mathbf{P}_Q^T\end{aligned}$$

where \mathbf{M}_q is a nonsingular $R \times R$ change-of-basis matrix.

PROOF. See Appendix 8.2

From Theorem 4, we can see that all TT-cores have a 3-order CPD structure, whose two matrix factors are nonsingular transformation matrices. Note that the CPD does not belong to the cases of Theorem 2, since the generalized unfolding $\mathcal{I}_{(q)}$ of the CPD core tensor \mathcal{I} is not a full rank matrix (as supposed in Theorem 2), due to the sparsity of the core tensor. In other words, Theorem 4 is not a special case of Theorem 2. Figure 2 depicts the CPD of the TT-core \mathcal{G}_q .

Remark 2. Once again, we should mention that the algebraic equivalence of Theorem 3 does not guarantee the minimality of the given TT-ranks, *i.e.*, the algebraic equivalence of Theorem 3 remains true even if some factors are rank deficient ($\text{rank} \mathbf{P}_q < R$), but the estimated TT-ranks and the TT-cores' structure will not be the same when the TT-SVD is applied in that case. Thus,

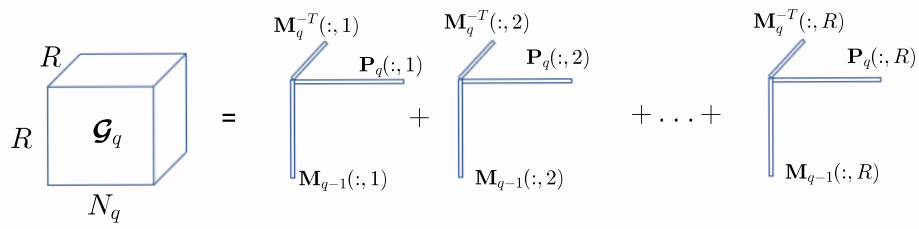


Figure 2: CPD of the q -th 3-order core of the associated Q -order CPD-Train.

265 additional rank assumptions on the factors are made in Theorem 4 guaranteeing
the minimality of the given TT-ranks.

5.1. Symmetric CPD

A special case of the CPD is the fully symmetric CPD tensor, which has
found several applications, e.g. in signal processing using high-order statistics
270 [9].

Definition 7. A Q -order symmetric tensor of size $N \times \dots \times N$ belonging to
the family of rank- R CPD admits the following decomposition:

$$\mathcal{X} = \mathcal{I}_{Q,R} \times_1 \mathbf{P} \times_2 \mathbf{P} \times_3 \dots \times_Q \mathbf{P}$$

where \mathbf{P} is of size $N \times R$.

Theorem 5. If a Q -order symmetric tensor admits a rank- R CPD, and \mathbf{P} has
full column rank, we have:

$$\begin{aligned} \mathcal{X} &\stackrel{\text{CPD}}{=} \mathcal{I}_{Q,R} \times_1 \mathbf{P} \times_2 \mathbf{P} \times_3 \dots \times_Q \mathbf{P} & (9) \\ &\stackrel{\text{TT}}{=} \mathbf{G}_1 \times_2^1 \mathcal{G}_2 \times_3^1 \mathcal{G}_3 \times_4^1 \dots \times_{Q-1}^1 \mathcal{G}_{Q-1} \times_Q^1 \mathbf{G}_Q, \end{aligned}$$

where

$$\mathbf{G}_1 = \mathbf{P} \mathbf{M}_1^{-1} \quad (10)$$

$$\mathcal{G}_q = \mathcal{I}_{3,R} \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{P} \times_3 \mathbf{M}_q^{-T} \quad \text{for } 2 \leq q \leq Q-1 \quad (11)$$

$$\mathbf{G}_Q = \mathbf{M}_{Q-1} \mathbf{P}^T, \quad (12)$$

and \mathbf{M}_q is a $R \times R$ matrix that follows Definition 1 ($1 \leq q \leq Q-1$). We can
then conclude that:

$$\langle \mathbf{P} \rangle = \langle \mathbf{G}_1 \rangle = \langle \mathcal{G}_q(:, :, i) \rangle = \langle \mathcal{G}_{q'}(j, :, :) \rangle = \langle \mathbf{G}_Q^T \rangle \quad (13)$$

for $2 \leq q \leq Q-1$, $2 \leq q' \leq Q-1$, $1 \leq i \leq R$, $1 \leq j \leq R$, and we have:

$$\text{rank}(\mathbf{P}) = \text{rank}(\mathbf{G}_1) = \text{rank}(\mathcal{G}_q(:, :, i)) = \text{rank}(\mathcal{G}_{q'}(j, :, :)) = \text{rank}(\mathbf{G}_Q^T) \quad (14)$$

which means that, in the symmetric case, the horizontal and frontal slices of the
TT-cores span the same subspaces.

PROOF. See Appendix 8.3.

275 *5.2. Permutation and scaling ambiguities*

It is known that matrix decompositions are not unique unless structural constraints are imposed on the factor matrices. The usual approach is to assume orthonormality by means of the SVD. However, the CPD enjoys essential uniqueness under mild conditions. Specifically, the factors of the 3-order CPD
 280 can be identified in a unique manner up to trivial (column permutation and scaling) ambiguities [39].

Theorem 6. *The factors of the Q CPDs associated with a CPD-Train are unique up to the following ambiguities:*

1. *common column permutation matrix denoted by $\mathbf{\Pi}$;*
2. *diagonal scaling matrices satisfying the following relation:*

$$\mathbf{\Lambda}_1 \mathbf{\Lambda}_2 \cdots \mathbf{\Lambda}_{Q-1} \mathbf{\Lambda}_Q = \mathbf{I}_R, \quad (15)$$

285 *where $\mathbf{\Lambda}_k$ is the scaling ambiguity for the k -th factor \mathbf{P}_k .*

PROOF. 1. It is straightforward to check that if a *change-of-basis* matrix \mathbf{M}_k^{-T} can be determined up to a column permutation matrix $\mathbf{\Pi}$ and a diagonal scaling matrix $\mathbf{\Gamma}_k$, then \mathbf{M}_k can also be determined up to the same permutation matrix and inverse scaling $\mathbf{\Gamma}_k^{-1}$. We can deduce that the column permutation matrix is unique and common to all the TT-cores due to the recursion property. This proves the first point of the above theorem.

290

2. We have

$$\begin{aligned} \mathbf{G}_1 &= \mathbf{P}_1 \mathbf{\Pi} \mathbf{\Lambda}_1 (\mathbf{M}_1 \mathbf{\Pi} \mathbf{\Gamma}_1^{-1})^{-1} \\ \mathcal{G}_2 &= \mathcal{I}_{3,R} \times_1 \mathbf{M}_1 \mathbf{\Pi} \mathbf{\Gamma}_1^{-1} \times_2 \mathbf{P}_2 \mathbf{\Pi} \mathbf{\Lambda}_2 \times_3 \mathbf{M}_2^{-T} \mathbf{\Pi} \mathbf{\Gamma}_2 \end{aligned} \quad (16)$$

$$\mathcal{G}_3 = \mathcal{I}_{3,R} \times_1 \mathbf{M}_2 \mathbf{\Pi} \mathbf{\Gamma}_2^{-1} \times_2 \mathbf{P}_3 \mathbf{\Pi} \mathbf{\Lambda}_3 \times_3 \mathbf{M}_3^{-T} \mathbf{\Pi} \mathbf{\Gamma}_3 \quad (17)$$

\vdots

$$\mathcal{G}_{Q-1} = \mathcal{I}_{3,R} \times_1 \mathbf{M}_{Q-2} \mathbf{\Pi} \mathbf{\Gamma}_{Q-2}^{-1} \times_2 \mathbf{P}_{Q-1} \mathbf{\Pi} \mathbf{\Lambda}_{Q-1} \times_3 \mathbf{M}_{Q-1}^{-T} \mathbf{\Pi} \mathbf{\Gamma}_{Q-1}$$

$$\mathbf{G}_Q = \mathbf{M}_{Q-1} \mathbf{\Pi} \mathbf{\Gamma}_{Q-1}^{-1} (\mathbf{P}_Q \mathbf{\Pi} \mathbf{\Lambda}_Q)^T.$$

Based on the above expressions, we have $\mathbf{\Gamma}_1^{-1} \mathbf{\Lambda}_2 \mathbf{\Gamma}_2 = \mathbf{I}_R$ from (16) and $\mathbf{\Gamma}_2 = \mathbf{\Lambda}_3 \mathbf{\Gamma}_3$ from (17). From these relations, we deduce $\mathbf{\Gamma}_1^{-1} \mathbf{\Lambda}_2 \mathbf{\Lambda}_3 \mathbf{\Gamma}_3 = \mathbf{I}_R$. Following the same reasoning for the k -th step we have

$$\mathbf{\Gamma}_1^{-1} \mathbf{\Lambda}_2 \cdots \mathbf{\Lambda}_k \mathbf{\Gamma}_k = \mathbf{I}_R,$$

$$\mathbf{\Gamma}_k^{-1} \mathbf{\Lambda}_{k+1} \mathbf{\Gamma}_{k+1} = \mathbf{I}_R.$$

As $\mathbf{\Gamma}_1^{-1} = \mathbf{\Lambda}_1$, and $\mathbf{\Gamma}_{Q-1} = \mathbf{\Lambda}_Q$, combining the above relations for a Q -order tensor allows to obtain (15).

The result of Theorem 6 is important from an estimation viewpoint. It means that the CPD-Train offers a way to retrieve the factors of a CPD under the same uniqueness properties but at lower complexity, thanks to dimensionality reduction and the *change-of-basis* matrices \mathbf{M}_q . This is particularly important for high-order tensors, where the direct computation of the factor matrices by means of traditional algorithms such as ALS may be impractical due to processing and storage limitations. Otherwise, not considering the matrices \mathbf{M}_q in the estimation will not guarantee the same permutation for the estimated factors, nor verify the equation (15) between the scaling matrices. Moreover, these matrices will be used to reduce the complexity by assuming the knowledge/pre-estimation of \mathbf{M}_{q-1} when decomposing \mathcal{G}_q in an estimation scheme. Hence the importance of these matrices.

6. Estimation algorithms for the CPD-and TD-Trains

In this section, several algorithms for Joint dimensionality Reduction And Factors rEtrieval (JIRAFE) are presented. These solutions are based on the results of Theorem 4 for the CPD-Train and of Theorem 2 for the TD-Train. So, the JIRAFE methodology can be described by the following two step procedure.

1. Reduce the dimensionality of the original factor retrieval problem by breaking the difficult multidimensional optimization problem into a collection of simpler optimization problems on small-order tensors. This step is carried out using the TT-SVD algorithm.
2. Design a factor retrieval strategy by exploiting (or not) the coupled structure existing in the 1st and 3rd factors for two consecutive TT-cores. Here, the goal is to minimize a sum of coupled least-squares criteria.

6.1. Fast Multilinear Projection (FMP) based on TD-Train

In many important applications, it is crucial to extract the dominant singular subspaces associated to the factors while the computation of the core tensor is not a primary goal. This scenario is usually known as a multilinear projection as described in [29] for instance. Indeed, in the context of multilinear analysis for facial recognition [43], the physical informations, *i.e.* people \times expression \times view \times illumination are encoded in the factors. Specifically, assume that we dispose of Q matrices $\hat{\mathbf{F}}_q$ ($1 \leq q \leq Q$), with $\hat{\mathbf{F}}_q^T \hat{\mathbf{F}}_q = \mathbf{I}$. The tensor-to-tensor multilinear projection is formulated according to

$$\mathcal{X}_{\text{proj}} = \mathcal{X}_{\text{data}} \times \hat{\mathbf{F}}_1^T \times_1 \cdots \times_Q \hat{\mathbf{F}}_Q^T.$$

6.1.1. Algorithmic description

A fast computation of the orthonormal factors $\hat{\mathbf{F}}_1, \dots, \hat{\mathbf{F}}_Q$ is presented in this section. Our scheme is based on an alternative interpretation of equations

(7)-(8) such as

$$\begin{aligned}\text{unfold}_2 \mathcal{G}_q &= \mathbf{F}_q \text{unfold}_2 \mathcal{T}'_q, \quad (1 < q < \bar{q}), \\ \text{unfold}_2 \mathcal{G}_{\bar{q}} &= \mathbf{F}_{\bar{q}} \text{unfold}_2 \mathcal{C}'_{\bar{q}}, \\ \text{unfold}_2 \mathcal{G}_q &= \mathbf{F}_q \text{unfold}_2 \bar{\mathcal{T}}'_q, \quad (\bar{q} < q < Q)\end{aligned}$$

where

$$\begin{aligned}\mathcal{T}'_q &= \mathcal{T}_q \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{I} \times_3 \mathbf{M}_q^{-T}, \quad (1 < q < \bar{q}), \\ \mathcal{C}'_{\bar{q}} &= \mathcal{C}_{\bar{q}} \times_1 \mathbf{M}_{\bar{q}-1} \times_2 \mathbf{I} \times_3 \mathbf{M}_{\bar{q}}^{-T}, \quad (q = \bar{q}), \\ \bar{\mathcal{T}}'_q &= \bar{\mathcal{T}}_q \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{I} \times_3 \mathbf{M}_q^{-T}, \quad (\bar{q} < q < Q).\end{aligned}$$

320 In other words, after the dimensionality reduction based on the TT-SVD algorithm, each projector matrix can be recovered thanks to a truncated SVD of the 2nd unfolding of the corresponding TT-core. A pseudo-code of this new approach is given in Algorithm 1.

Algorithm 1 Fast Multilinear Projection algorithm based on JIRAFE

Input: Q -order tensor \mathcal{X}

Output: Estimated orthonormal factors: $\hat{\mathbf{F}}_1, \dots, \hat{\mathbf{F}}_Q$.

1: Dimensionality reduction:

$$[\mathbf{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_{Q-1}, \mathbf{G}_Q] = \text{TT-SVD}(\mathcal{X}).$$

Orthonormal Factors retrieval:

2: **for** $q = 2 \dots Q - 1$ **do**

3: $\hat{\mathbf{F}}_q =$ Matrix of the left singular vectors of $\text{SVD}(\text{unfold}_2 \mathcal{G}_q)$ {This step can be done in a parallel way}

4: **end for**

6.1.2. On the difference with the HOSVD

It is important to note that using Definition 4, the core tensor $\hat{\mathcal{C}}$ generated by the above algorithm is given by $\hat{\mathcal{C}} = \mathcal{X} \times_1 \hat{\mathbf{F}}_1^T \times_2 \dots \times_Q \hat{\mathbf{F}}_Q^T$. Let be the HOSVD of a tensor $\mathcal{X} = \mathcal{S} \times_1 \mathbf{U}_1 \times_2 \dots \times_Q \mathbf{U}_Q$ with $\mathbf{U}_q^T \mathbf{U}_q = \mathbf{I}$. Using Definition 4, the core tensor for the HOSVD is given by $\mathcal{S} = \mathcal{X} \times_1 \mathbf{U}_1^T \times_2 \dots \times_Q \mathbf{U}_Q^T$. It is well-known that the core tensor, \mathcal{S} , of the HOSVD satisfies the all-orthogonality and pseudo-diagonality properties [11]. On the contrary, we have no guaranty that $\hat{\mathcal{C}}$ satisfies these properties. Indeed, the TT-SVD algorithm provides an orthonormal matrix $\hat{\mathbf{F}}_q$ which span the same subspace as the factor \mathbf{U}_q . This means that there exists a change-of-basis matrix (see Definition 2.1), denoted by \mathbf{J}_q for instance, such as $\mathbf{J}_q = \mathbf{U}_q^T \hat{\mathbf{F}}_q$. The link between the two core tensors is given by

$$\mathcal{S} = \mathcal{X} \times_1 \mathbf{U}_1^T \times_2 \dots \times_Q \mathbf{U}_Q^T = \hat{\mathcal{C}} \times_1 \mathbf{J}_1 \times_2 \dots \times_Q \mathbf{J}_Q.$$

325 *6.1.3. Analysis of the computational cost and execution time*

The computation of the native HOSVD for a Q -order Tucker tensor \mathcal{X} of size $N \times \dots \times N$ involves the computation of Q dominant left singular basis thanks to truncated-SVDs on the unfolding matrices of size $N \times N^{Q-1}$, other methods, such as ST-HOSVD [42], may use different truncation strategies, while
 330 the proposed strategy consists of computing Q left and right dominant singular basis in the TT-SVD algorithm and $Q - 2$ left dominant singular basis of the TT-cores. We recall that based on the QR-orthogonal iteration (QR-OI) [18], the r -truncated SVD computational cost for a single iteration and for the left dominant singular basis for a $n \times m$ matrix is evaluated to $2rmn + 2r^2 \min(n, m)$
 335 flops. The computation of the left and the right dominant singular basis is evaluated to $2rmn + 2r^2(n + m)$ flops.

As an illustrative example, the computational cost of the HOSVD factors of (T, \dots, T) -multilinear rank for a 4-order tensor is $8TN^4 + 8T^2N$. For the TT-SVD, the TT-ranks are (T, T^2, T) and $\bar{q} = 2$. Note that the TT-ranks may be large with respect to the multilinear rank so as shown on Fig. 3, it makes sense to consider the case where $N \gg T$. Note that this is a standard assumption when using the HOSVD. The complexity cost for the TT-SVD is $2TN^4 + 2T^2(N + N^3)$ for the first SVD of a $N \times N^3$ matrix of rank T . The complexity of the second
 340 SVD of a $(TN) \times N^2$ matrix of rank T^2 is $2T^3N^3 + 2T^4(TN + N^2)$. Finally, the last SVD of a $(T^2N) \times N$ matrix of rank T is $2T^3N^2 + 2T^2(T^2N + N)$. Finally, we have to perform two SVDs of the 2nd unfolding matrices of \mathcal{G}_2 and \mathcal{G}_3 of size $T \times N \times T^2$ and $T^2 \times N \times T$, respectively for the extraction of the left dominant basis. The total cost is $4NT^4 + 4T^2 \min(N, T^3)$.

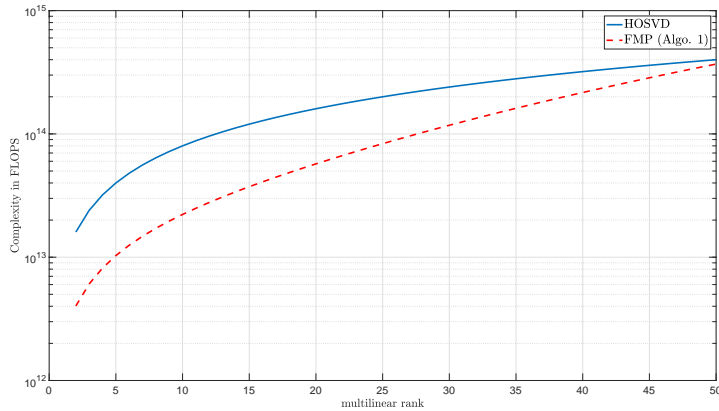


Figure 3: Number of flops vs the multilinear rank 4-order tensor with $N = 500$.

350 If we generalize this analysis to a Q -order tensor, we find $2QTN^Q$ flops for the factors computation of the HOSVD and $2TN^Q$ for the proposed algorithm. So, the proposed algorithm computational cost is reduced by a factor Q . In

other words, the cost of the computation of the Q factors by Algo.1 is comparable to the cost of a single factor computation for the HOSVD.

355 Although the proposed method remains less complex than the HOSVD. The computation of the TT-SVD algorithm still requires the computation of the SVD of a $N \times N^{Q-1}$ matrix, with a complexity $O(TN^Q)$, which can be expensive for large Q . This means that the computation of the TT decomposition of an unstructured tensor is a challenging problem in high dimension.

360

In Table 1, an hypercubic tensor with $N = 10$ was generated, with a multilinear rank given by $T_1 = \dots = T_Q = T = 2$. Its entries are randomly drawn from a Gaussian distribution with zero mean and unit variance for several tensor orders in a noiseless scenario. The average execution time was evaluated for 20 Monte Carlo realisations. Note that in these simulations, we compute the thin SVD [18].

Table 1: Computation times for ($T = 2, N = 10$)

Tensor order	Fast Multilinear Projection	Q -th order HOSVD	Gain
$Q = 6$	0.14 (s)	0.66 (s)	4.71
$Q = 7$	1.31 (s)	7.34 (s)	5.6
$Q = 8$	15.25 (s)	101.87 (s)	6.68

Note that the gain increases when the order increases and is of the order of Q as remarked before. It is also worth noting that the TT-ranks can actually be large, but, as demonstrated by the complexity cost analysis, Algorithm 1 is less complex to a native implementation of the HOSVD and faster as demonstrated in the simulations.

370

The robustness to an i.i.d. Gaussian noise is evaluated thanks to the Normalized MSE defined according to $NMSE = \frac{\|\hat{\mathcal{X}} - \mathcal{X}\|_F^2}{\|\mathcal{X}\|_F^2}$, where $\hat{\mathcal{X}}$ refers to the estimated tensor. In Fig. 4, the NMSE measurements averaged over 300 noise realisations are given for the proposed Fast Multilinear Projection and the native HOSVD algorithms for a 6-order hypercubic tensor, with $N = 4$, and $T = 2$. Note that for a computational gain approximately equal to the tensor order Q , the same robustness to noise is observed for both algorithms. This result is important since the HOSVD is intensively exploited in numerous applications.

375

380 6.2. Fast CPD with CPD-Train

In this section, we provide several algorithms adapted to the CPD-Train. Before presenting the new algorithms, we give hereafter a list of advantages of the proposed CPD-Train equivalence:

385

1. The CPD-Train has the same number of free parameters as the CPD for large Q .
2. The TT-cores follow a 3-order CPD with canonical rank equal to the rank of the initial Q -order CPD.

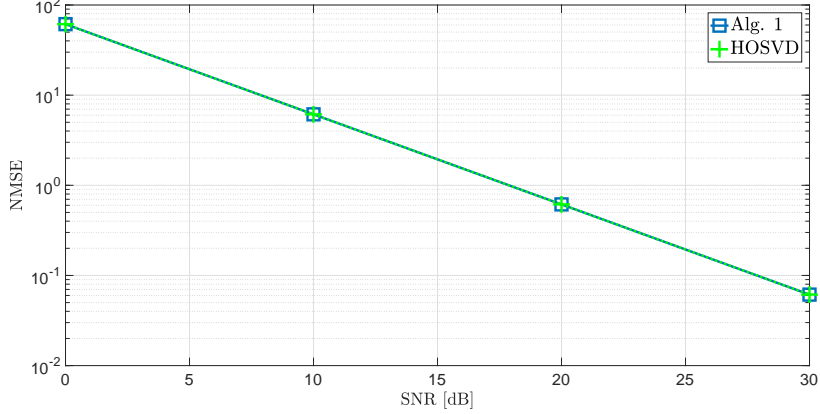


Figure 4: NMSE vs SNR in dB with Fast Multilinear Projection for a 6-order Tucker with $N = 4$, $T = 2$, 300 runs

3. Based on the TT-SVD algorithm, a dimensionality reduction is carried out thanks to a non-iterative SVD-based algorithm instead of the standard iterative ALS algorithm.
4. Due to the CPD structure of the TT-cores, the optimization in a high dimensional (Q) space is replaced by a collection of much simpler optimization problems, *i.e.*, we have to solve $(Q-3)$ optimizations in bi-dimensional spaces and a single optimization in a tri-dimensional space.

6.2.1. CPD-Train based on low-order ALS algorithm

The goal of the JIRAFE approach is to optimize the following criterion:

$$\min_{\mathbb{M}, \mathbb{P}} \left\{ \|\mathbf{G}_1 - \mathbf{P}_1 \mathbf{M}_1^{-1}\| + \|\mathbf{G}_Q - \mathbf{M}_{Q-1} \mathbf{P}_Q^T\| + \sum_{q=2}^{Q-2} \|\mathcal{G}_q - \mathcal{I}_{3,R} \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{P}_q \times_3 \mathbf{M}_q^{-T}\| \right\}$$

where $\mathbb{M} = \{\mathbf{M}_1, \dots, \mathbf{M}_Q\}$ and $\mathbb{P} = \{\mathbf{P}_1, \dots, \mathbf{P}_Q\}$.

Our first proposition is based on the popular ALS-CPD algorithm [4, 20] applied in a sequential way on the CPD-Train cores to jointly retrieve the CPD factors and the *change-of-basis* matrices \mathbb{M} . A pseudo-code is presented in Algorithm 2, where Tri-ALS stands for the ALS algorithm applied to a 3-order tensor, while Bi-ALS denotes the ALS algorithm applied to a 3-order tensor using *a priori* knowledge of one factor. Note that the ALS approach fixes all but one factor to estimate this latter solving the following least squares problem.

$$\min_{\mathbf{P}_q} \left\| \text{unfold}_q \mathcal{X} - \mathbf{P}_q \cdot (\mathbf{P}_Q \odot \dots \odot \mathbf{P}_{q+1} \odot \mathbf{P}_{q-1} \odot \dots \odot \mathbf{P}_1)^T \right\|^2$$

This procedure is repeated several times until a convergence criterion is satisfied. When one of the 3-order tensor's factors is known, the ALS algorithm only has two steps, which we referred to with the Bi-ALS.

Algorithm 2 JIRAFE based on CPD-Train

Input: Q -order rank- R CPD tensor \mathcal{X}

Output: Estimated factors: $\hat{\mathbf{P}}_1, \dots, \hat{\mathbf{P}}_Q$.

1: Dimensionality reduction:

$$[\mathbf{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_{Q-1}, \mathbf{G}_Q] = \text{TT-SVD}(\mathcal{X}, R).$$

2: Factor retrieval:

$$[\hat{\mathbf{M}}_1, \hat{\mathbf{P}}_2, \hat{\mathbf{M}}_2^{-T}] = \text{Tri-ALS}(\mathcal{G}_2, R).$$

3: **for** $k = 3 \dots Q - 1$ **do**

4: $[\hat{\mathbf{P}}_k, \hat{\mathbf{M}}_k^{-T}] = \text{Bi-ALS}(\mathcal{G}_k, \hat{\mathbf{M}}_{k-1}, R)$

5: **end for**

6: $\hat{\mathbf{P}}_1 = \mathbf{G}_1 \hat{\mathbf{M}}_1$, and $\hat{\mathbf{P}}_Q = \mathbf{G}_Q^T \hat{\mathbf{M}}_{Q-1}^{-T}$

The usual strategy is based on the brute force exploitation of a Q -order ALS-CPD on the original tensor. We note that $\text{TT-SVD}(\mathcal{X}, R)$ refers to the conventional TT-SVD algorithm with all TT-ranks set to R . For the complexity analysis, it is assumed that the pseudo-inverse and the SVD have the same complexity order. Note that a single iteration of the Q -order ALS-CPD requires Q SVDs of rank- R matrices of size $N \times N^{Q-1}$. The complexity is evaluated as $Q \cdot O(R^2 \cdot N^{Q-1})$, whereas the TT-SVD applied to the original Q -order tensor ($Q \gg 1$) has a complexity $O(R^2 \cdot N^{Q-1})$.

In addition, the R -truncated SVD [18] using the orthogonal iteration algorithm is faster to compute than the full rank SVD. The complexity of a R -truncated SVD for a $m \times n$ matrix is $O(R^2 \max(m, n))$. Thus, we have

$$\begin{aligned} \kappa(1 \cdot \text{iteration of } Q\text{-order ALS}) &\gg \kappa(Q\text{-order TT-SVD}) = O(R^2 \cdot N^{Q-1}) \\ &\gg \kappa(1 \cdot \text{iteration of 3-order ALS}) = O(3 \cdot R^3 \cdot N). \end{aligned}$$

400 This means that the proposed strategy is approximately (Q .number of iterations)-
times less complex than the Q -order ALS-CPD. In Table 2, the computation
times of the two methods are given for $N = 6$ and for different tensor or-
ders. To manage the convergence of the ALS-CPD, the stopping criterion is
 $\frac{|f(\hat{\mathbf{x}}^{(t)}) - f(\hat{\mathbf{x}}^{(t+1)})|}{f(\hat{\mathbf{x}}^{(t)})} < \epsilon$, where $f(\hat{\mathbf{x}}^{(t)}) = \|\mathcal{X} - \hat{\mathbf{x}}^{(t)}\|_F$, in which $\hat{\mathbf{x}}^{(t)}$ denotes
405 the estimated tensor at the t -th iteration and ϵ is the convergence threshold. In

addition, the number of iterations is limited to 1000. It is clear that the gain strongly increases when the order of the initial tensor grows.

Table 2: Computation times for ($R = 3, N = 6$)

Tensor order	Alg. 2	Q -order ALS-CPD	Gain
$Q = 6$	0, 63 (s)	7, 48 (s)	11, 87
$Q = 7$	1, 07 (s)	62, 57 (s)	58, 47
$Q = 8$	1, 31 (s)	431, 95 (s)	329, 73

In Table 3, we fix the order at $Q = 8$ and vary the tensor dimensions. Here again, the gain of Alg. 2 over ALS-CPD in terms of computation time is measured.

Table 3: Computation times for ($R = 3, Q = 8$)

Tensor dimension	Alg. 2	Q -order ALS-CPD	Gain
$N = 4$	0, 79 (s)	19, 26 (s)	24, 37
$N = 5$	0, 91 (s)	114, 16 (s)	125, 45
$N = 6$	1, 31 (s)	431, 95 (s)	329, 73

One may note that in both Tables 2 and 3 the results are given in the case where the tensor dimensions are larger than its rank. This usually happens in big data applications, where the data tensor is of high order, and its rank is smaller than its dimensions ($N > R$). This assumption is valid in a number of applications in wireless communications, spectral analysis, MIMO radar and image compression. Note also that the result of Theorem 4 can not be directly applied in the case $R > N$, since the structure of the TT-cores and the TT-ranks are totally different according to the ranks of the factors \mathbf{P}_q , as mentioned before.

To evaluate the noise robustness, an 8-order, rank-2 CPD tensor of size $N \times \dots \times N$ is generated with $N = 3$. In Fig. 5, we plot the NMSE obtained with Algorithm 2, the Q -order ALS algorithm and the popular HOSVD-based preprocessed ALS for 8-order hypercubic CPDs, with $N = 3$ and $R = 2$. The preprocessing step is done thanks to the HOSVD with multilinear ranks $T = 2$. The association of a preprocessing (HOSVD) to the ALS algorithm generally improves its robustness to noise while increasing the convergence speed. The NMSEs were averaged over 300 i.i.d. realisations of a Gaussian noise. To eliminate ill-convergence experiments and outliers from our results, 5% of the worst and 5% of the best NMSE values are discarded.

According to the NMSE results, the Q -order ALS algorithm is the least robust scheme for SNR lower than 20 dB for a high computational cost. The ALS scheme with the preprocessing step shows an improved noise robustness for a

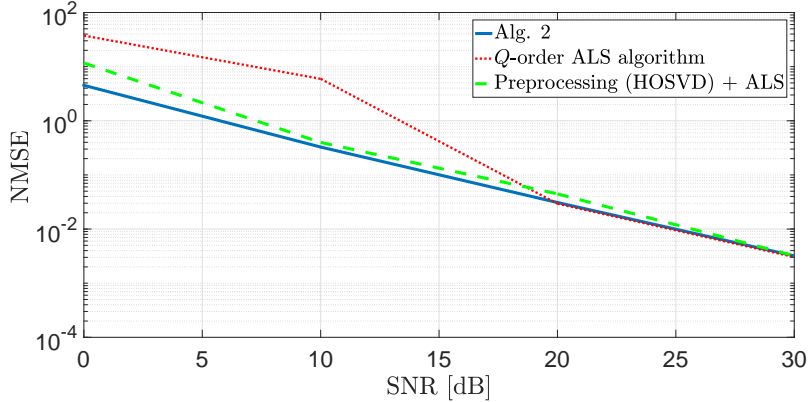


Figure 5: NMSE vs SNR in dB with JIRAFE for an 8-order CPD

very high computational cost. Finally, the proposed algorithm has the highest accuracy, or equivalently, the best noise robustness with the lowest computational cost.

6.2.2. Improved CPD-Train

In this section, an improved CPD-Train based factor retrieval algorithm is proposed. In Algorithm 2, by exploiting the repetition of the matrices $\{\hat{\mathbf{M}}_1, \dots, \hat{\mathbf{M}}_{Q-1}\}$ in the TT-cores, it is possible to replace the iterative Bi-ALS algorithm by the non-iterative Khatri-Rao factorization (KRF) algorithm proposed in [27]. The KRF algorithm is a closed-form algorithm that recovers 3-order CPD factors assuming that one factor is known and has a full-column rank. It computes R SVDs of rank-one matrices to recover the remaining two other factors. These assumptions match exactly to our case. Note that the matrix $\hat{\mathbf{M}}_q$ by definition is invertible and thus satisfies the full-column rank assumption. A pseudo-code of the proposed strategy is given in Algorithm 3. Note that Algorithm 3 is thus less sensitive to potential ill-convergence problems than Algorithm 2. In addition, it is worth noting that the complexity of the KRF corresponds to $\kappa(\text{KRF}) = O(R \cdot N) \ll \kappa(1 \cdot \text{iteration of Bi-ALS}) = O(R^3 \cdot N)$, which means that the non-iterative KRF is much less complex than the iterative Bi-ALS algorithm.

In Fig. 6, we plot the NMSE of Algorithm 3 for an 8-order hypercubic CPD with $N = 3$ and $R = 2$. The NMSEs were averaged in the same way as in Fig. 5. It is worth noting that Algorithm 3 has an equivalent robustness to noise as Algorithm 2, using a closed-form, non sensitive to ill-convergence problems, KRF method instead of the iterative Bi-ALS algorithm.

Algorithm 3 Improved CPD-Train algorithm

Input: Q -order rank- R CPD tensor \mathcal{X} ,**Output:** Estimated factors: $\hat{\mathbf{P}}_1, \dots, \hat{\mathbf{P}}_Q$.1: Dimensionality reduction:

$$[\mathbf{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_{Q-1}, \mathbf{G}_Q] = \text{TT-SVD}(\mathcal{X}, R).$$

2: Factor retrieval:

$$[\hat{\mathbf{M}}_1, \hat{\mathbf{P}}_2, \hat{\mathbf{M}}_2^{-T}] = \text{Tri-ALS}(\mathcal{G}_2, R).$$

3: **for** $q = 3 \dots Q - 1$ **do**4: $[\hat{\mathbf{P}}_q, \hat{\mathbf{M}}_q^{-T}] = \text{KRF}(\mathcal{G}_q, \hat{\mathbf{M}}_{q-1}, R)$ 5: **end for**6: $\hat{\mathbf{P}}_1 = \mathbf{G}_1 \hat{\mathbf{M}}_1$, and $\hat{\mathbf{P}}_Q = \mathbf{G}_Q^T \hat{\mathbf{M}}_{Q-1}^{-T}$

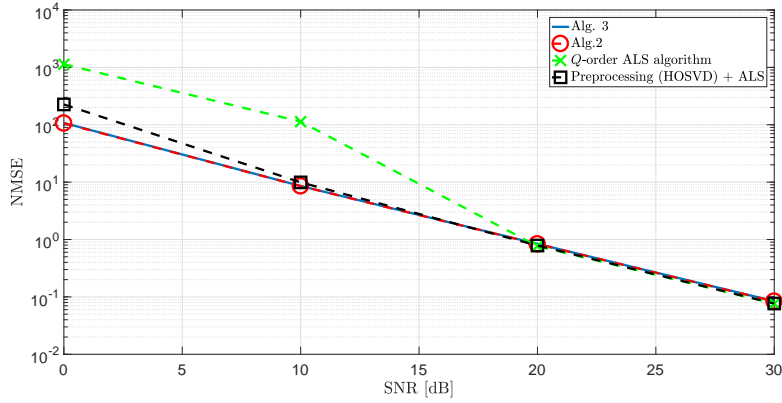


Figure 6: NMSE vs SNR in dB with improved CPD-Train for an 8-order CPD

6.2.3. Non-iterative CPD-Train algorithm in the case of a known factor

In some applications, a factor among Q can be supposed to be known [17, 44]. In this case, a fully non-iterative algorithm can be derived. Indeed, assume
 460 without loss of generality¹ that the first factor, \mathbf{P}_1 , is known. The recovery of \mathbf{M}_1 is straightforward thanks to the pseudo-inverse of the first TT-core. The rest of the method remains identical as for Algorithm 3. A pseudo-code is given in Algorithm 4.

Algorithm 4 Non-iterative CPD-Train algorithm in case of a known factor

Input: Q -order rank- R CPD tensor \mathcal{X} , \mathbf{P}_1 .

Output: Estimated factors: $\hat{\mathbf{P}}_2, \dots, \hat{\mathbf{P}}_Q$.

1: Dimensionality reduction:

$$[\mathbf{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_{Q-1}, \mathbf{G}_Q] = \text{TT-SVD}(\mathcal{X}, R).$$

2: Factor retrieval:

$$\hat{\mathbf{M}}_1 = \mathbf{G}_1^\dagger \mathbf{P}_1$$

3: **for** $q = 2 \dots Q - 1$ **do**

4: $[\hat{\mathbf{P}}_q, \hat{\mathbf{M}}_q^{-T}] = \text{KRF}(\mathcal{G}_q, \hat{\mathbf{M}}_{q-1}, R)$

5: **end for**

6: $\hat{\mathbf{P}}_Q = \mathbf{G}_Q^T \hat{\mathbf{M}}_{Q-1}^{-T}$

6.2.4. Parallel and non-iterative CPD-Train algorithm in case of Toeplitz factors

465 For structured tensors, several algorithms have been proposed by exploiting the structure of the factors in [13, 1]. In some applications [26, 37], the factors of the CPD are Toeplitz. Recall that for a vector $\mathbf{a} = (a_0 \dots a_{L-1})^T$ of length L , we denote by $\mathbf{T}(\mathbf{a})$ the $(L + R - 1) \times R$ Toeplitz matrix that verifies $\mathbf{T}(\mathbf{a})_{i,j} = a_{i-j}$,
 470 where by convention $a_{i-j} = 0$ if $i - j < 0$ or $i - j > L - 1$. The choice of this convention is related to applications like Wiener-Hammerstein systems [27], the generalization to the general case is straightforward. Thanks to Theorem 5.2, the TT-cores inherit from this structure. Thus, it makes sense to exploit the TOMFAC algorithm, for TOeplitz Matrix Factor Computation, proposed in
 475 [27]. This leads to the proposition of Algorithm 5. TOMFAC is an algorithm that allows to recover a CPD factor in a closed-form way, considering that it has a Toeplitz structure. As a consequence, Algorithm 5 is fully non-iterative and thus the convergence problem due to the use of an alternated estimation scheme

¹Remark that the choice of the index of the factors is totally arbitrary and is meaningless relatively to the model.

is completely avoided. In addition, the estimation of the Toeplitz factors can
 480 be done via a *parallel* processing.

Algorithm 5 Parallel and non-iterative CPD-Train algorithm in case of Toeplitz factors

Input: Q -order rank- R CPD tensor \mathcal{X}

Output: Estimated factors: $\hat{\mathbf{P}}_1, \dots, \hat{\mathbf{P}}_Q$.

1: Dimensionality reduction:

$$[\mathbf{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_{Q-1}, \mathbf{G}_Q] = \text{TT-SVD}(\mathcal{X}, R).$$

2: **for** $q = 1 \dots Q$ **do**

3: $\hat{\mathbf{P}}_q = \text{TOMFAC}(\mathcal{G}_q, R)$ {This step can be done in a parallel way}

4: **end for**

In Table 4, the computation times of the native TOMFAC method applied to a Q -order CPD tensor with Toeplitz factors and of Algorithm 5 are compared. Note that the complexity of the Q -order TOMFAC is of the same order as the complexity of a Q -order HOSVD. Here again, interesting gains in terms of
 485 computation times are obtained.

Table 4: Computation times for ($R = 3, N = 4$)

Tensor order	CPD-Train-TOMFAC	TOMFAC	Gain
$Q = 6$	0.0296 (s)	0.1206 (s)	4.0743
$Q = 7$	0.2906 (s)	1.6447 (s)	5.6593
$Q = 8$	5.1253 (s)	39.7964 (s)	7.7647

Figure 7 depicts the NMSE performance for the Q -order TOMFAC and Algorithm 5, referred to here as CPD-Train-TOMFAC, for an 8-order hypercubic CPD with $N = 6$ and $R = 3$. The NMSE curves are plotted with respect to the SNR in dB. Note that thanks to the dimensionality reduction step, the CPD-
 490 Train-TOMFAC algorithm shows a comparable robustness, for severe SNRs, to the native TOMFAC for a much smaller computational complexity cost.

7. Conclusion

We have discussed the joint dimensionality reduction and factor retrieval problem for high-order tensors. We have shown that a Q -order tensor following a CPD/TD ($Q > 3$) can be written as a train of Q 3-order tensors with a coupled
 495 structure. Exploiting this model equivalence property, we have introduced new tensor models, namely the CPD-Train and the Tucker Decomposition (TD)-Train. A two-step JIRAFE methodology has been proposed to overcome the

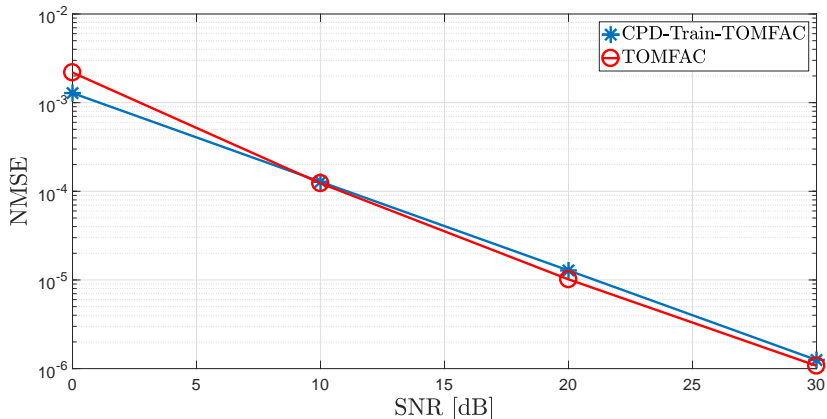


Figure 7: NMSE vs SNR in dB with CPD-Train-TOMFAC for an 8-order CPD with Toeplitz factors, $N = 6$, and $R = 3$

“curse of dimensionality” for high-order tensors. The initial step, called “dimensionality reduction”, consists of splitting the high-order tensor into a collection of graph-connected core tensors of lower orders, at most equal to three. The second step consists of factors retrieval. Several algorithms specialized for the CPD-Train/TD-Train have been proposed, which have a low computational cost and a storage cost that grows linearly with respect to the order of the data tensor. The advantages in terms of storage cost, computational complexity, algorithmic stability, and factor estimation robustness to noise have been demonstrated by means of Monte Carlo simulations. JIRAFE can then be seen as a new concept of a more general framework for joint dimensionality reduction and factor retrieval, where different solutions for the factor retrieval step can be considered, such as Gauss-Newton [40] for fitting the CPD, gradient descent [10], or improved versions of ALS such as those based on enhanced line search [34], for instance. Perspective for future works include the application of JIRAFE to big data tensors for multilinear harmonic retrieval problem, massive MIMO systems and coupled matrix-tensor factorizations, besides the investigation of the TT-cores structures when the rank exceeds the dimensions.

8. Appendix

8.1. Proof of theorem 2

In this constructive proof, the TT-SVD algorithm is applied “step by step” to a TD presented in (3), under the two assumptions that

- all the factors \mathbf{F}_q have full column rank, and
- $\mathbf{C}_{(q)} = \text{reshape}(\mathcal{C}; \prod_{i=1}^q T_i, \prod_{i=q+1}^Q T_i)$ has full rank.

The first unfolding $\mathbf{X}_{(1)}$ of size $N_1 \times (N_2 \cdots N_Q)$, using Lemma 2, is given by:

$$\begin{aligned}\mathbf{X}_{(1)} &= \text{reshape}(\mathcal{X}; N_1, \prod_{q=2}^Q N_q) \\ &\triangleq \mathbf{F}_1 \mathbf{C}_{(1)} (\mathbf{F}_Q \otimes \cdots \otimes \mathbf{F}_2)^T \\ &\stackrel{\text{SVD}}{=} \mathbf{U}^{(1)} \mathbf{V}^{(1)}.\end{aligned}$$

Note that $\text{rank}(\mathbf{X}_{(1)}) = \text{rank}(\mathbf{F}_1) = R_1 = T_1$ (R_1 is the first TT-rank), which means that $\mathbf{U}^{(1)}$ is of size $N_1 \times R_1$, and $\mathbf{V}^{(1)}$ is of size $R_1 \times (N_2 \cdots N_Q)$. It exists a $R_1 \times R_1$ nonsingular *change-of-basis* matrix \mathbf{M}_1 that verifies

$$\mathbf{U}^{(1)} = \mathbf{F}_1 \mathbf{M}_1^{-1} \quad (18) \quad \mathbf{V}^{(1)} = \mathbf{M}_1 \mathbf{C}_{(1)} (\mathbf{F}_Q \otimes \cdots \otimes \mathbf{F}_2)^T \quad (19)$$

Following the methodology of the TT-SVD algorithm, the first TT-core \mathbf{G}_1 is given by eq. (18), *i.e.*,

$$\mathbf{G}_1 = \mathbf{U}^{(1)} = \mathbf{F}_1 \mathbf{M}_1^{-1}. \quad (20)$$

Applying the same methodology to a reshaped version, denoted by $\mathbf{V}_{(2)}^{(1)}$, of $\mathbf{V}^{(1)}$, of size $(R_1 N_2) \times (N_3 \cdots N_Q)$, provides from eq. (19)

$$\begin{aligned}\mathbf{V}_{(2)}^{(1)} &= \text{reshape}(\mathbf{V}^{(1)}; R_1 N_2, \prod_{q=3}^Q N_q) \\ &\triangleq (\mathbf{F}_2 \otimes \mathbf{M}_1) \mathbf{C}_{(2)} (\mathbf{F}_Q \otimes \cdots \otimes \mathbf{F}_3)^T \\ &\stackrel{\text{SVD}}{=} \mathbf{U}^{(2)} \mathbf{V}^{(2)}\end{aligned} \quad (21)$$

Once again, we have $\text{rank}(\mathbf{V}_{(2)}^{(1)}) = \text{rank}(\mathbf{F}_2 \otimes \mathbf{M}_1) = R_2 = T_1 T_2$. In addition, $\mathbf{U}^{(2)}$ of size $(R_1 N_2) \times R_2$, and $\mathbf{V}^{(2)}$ of size $R_2 \times (N_3 \cdots N_Q)$ are defined by

$\mathbf{U}^{(2)} = (\mathbf{F}_2 \otimes \mathbf{M}_1) \mathbf{M}_2^{-1}$, (22) $\mathbf{V}^{(2)} = \mathbf{M}_2 \mathbf{C}_{(2)} (\mathbf{F}_Q \otimes \cdots \otimes \mathbf{F}_3)^T$, where \mathbf{M}_2 is a $R_2 \times R_2$ nonsingular *change-of-basis* matrix. From eq.(22), the 2nd TT-core \mathbf{G}_2 is expressed as

$$\mathbf{G}_2 = \text{reshape}(\mathbf{U}^{(2)}; R_1, N_2, R_2) = \mathcal{T}_2 \times_1 \mathbf{M}_1 \times_2 \mathbf{F}_2 \times_3 \mathbf{M}_2^{-T}, \quad (23)$$

where $\mathcal{T}_2 = \text{reshape}(\mathbf{I}_{R_2}; R_1, T_2, R_2)$.

This strategy can be continued to obtain the q -th TT-cores for $3 \leq q \leq \bar{q} - 1$ since $\mathbf{C}_{(q)}$, defined in Theorem 1, has always a full row rank and can always be absorbed in $\mathbf{V}^{(q)}$, when applying the SVD. Let us now see what happens when we express $\mathbf{V}_{(2)}^{(\bar{q}-1)}$ of size $(R_{\bar{q}-1} N_{\bar{q}}) \times (N_{\bar{q}+1} \cdots N_Q)$. From (21), we obtain

$$\begin{aligned}\mathbf{V}_{(2)}^{(\bar{q}-1)} &= \text{reshape}(\mathbf{V}^{(\bar{q}-1)}; R_{\bar{q}-1} N_{\bar{q}}, \prod_{q=\bar{q}+1}^Q N_q) \\ &\triangleq (\mathbf{F}_{\bar{q}} \otimes \mathbf{M}_{\bar{q}-1}) \mathbf{C}_{(\bar{q})} (\mathbf{F}_Q \otimes \cdots \otimes \mathbf{F}_{\bar{q}+1})^T \\ &\stackrel{\text{SVD}}{=} \mathbf{U}^{(\bar{q})} \mathbf{V}^{(\bar{q})}\end{aligned}$$

Note that here we have $\text{rank}(\mathbf{V}_{(2)}^{(\bar{q}-1)}) = \text{rank}((\mathbf{F}_{\bar{q}} \otimes \mathbf{M}_{\bar{q}-1})\mathbf{C}_{(\bar{q})}) = R_{\bar{q}} = T_{\bar{q}+1} \cdots T_Q$, since $\mathbf{C}_{(\bar{q})}$ has a full column rank, and we can write

$$\mathbf{U}^{(\bar{q})} = (\mathbf{F}_{\bar{q}} \otimes \mathbf{M}_{\bar{q}-1})\mathbf{C}_{(\bar{q})}\mathbf{M}_{\bar{q}}^{-1} \quad (24) \quad \mathbf{V}^{(\bar{q})} = \mathbf{M}_{\bar{q}}(\mathbf{F}_Q \otimes \cdots \otimes \mathbf{F}_{\bar{q}+1})^T \quad (25)$$

From eq.(24), the \bar{q} -th TT-core can be expressed as:

$$\mathcal{G}_{\bar{q}} = \text{reshape}(\mathbf{U}^{(\bar{q})}; R_{\bar{q}-1}, N_{\bar{q}}, R_{\bar{q}}) = \mathbf{C}_{\bar{q}} \times_1 \mathbf{M}_{\bar{q}-1} \times_2 \mathbf{F}_{\bar{q}} \times_3 \mathbf{M}_{\bar{q}}^{-T}, \quad (26)$$

where $\mathbf{C}_{\bar{q}} = \text{reshape}(\mathbf{C}; R_{\bar{q}-1}, T_{\bar{q}}, R_{\bar{q}})$ is absorbed in $\mathcal{G}_{\bar{q}}$.

Applying the same computation to the reshaping of $\mathbf{V}^{(\bar{q})}$, denoted by $\mathbf{V}_{(2)}^{(\bar{q})}$ of size $(R_{\bar{q}}N_{\bar{q}+1}) \times (N_{\bar{q}+2} \cdots N_Q)$, and considering an identity matrix $\mathbf{I}_{R_{\bar{q}} \times R_{\bar{q}}}$ in (25) such as $\mathbf{V}^{(\bar{q})} = \mathbf{M}_{\bar{q}}\mathbf{I}_{R_{\bar{q}} \times R_{\bar{q}}}(\mathbf{F}_Q \otimes \cdots \otimes \mathbf{F}_{\bar{q}+1})^T$, we obtain

$$\begin{aligned} \mathbf{V}_{(2)}^{(\bar{q})} &= \text{reshape}(\mathbf{V}^{(\bar{q})}; R_{\bar{q}}N_{\bar{q}+1}, \prod_{k=\bar{q}+2}^Q N_k) \\ &\stackrel{\Delta}{=} (\mathbf{F}_{\bar{q}+1} \otimes \mathbf{M}_{\bar{q}})\mathbf{I}_{(\bar{q}+1)}(\mathbf{F}_Q \otimes \cdots \otimes \mathbf{F}_{\bar{q}+2})^T \\ &\stackrel{\text{SVD}}{=} \mathbf{U}^{(\bar{q}+1)}\mathbf{V}^{(\bar{q}+1)} \end{aligned}$$

where $\mathbf{I}_{(\bar{q}+1)} = \text{reshape}(\mathbf{I}_{R_{\bar{q}} \times R_{\bar{q}}}; R_{\bar{q}} \cdot T_{\bar{q}+1}, \frac{R_{\bar{q}}}{T_{\bar{q}+1}})$. Matrices $\mathbf{U}^{(\bar{q}+1)}$ and $\mathbf{V}^{(\bar{q}+1)}$ are then expressed as

$$\begin{aligned} \mathbf{U}^{(\bar{q}+1)} &= (\mathbf{F}_{\bar{q}+1} \otimes \mathbf{M}_{\bar{q}})\mathbf{I}_{(\bar{q}+1)}\mathbf{M}_{\bar{q}+1}^{-1}, \quad (27) \\ \mathbf{V}^{(\bar{q}+1)} &= \mathbf{M}_{\bar{q}+1}(\mathbf{F}_Q \otimes \cdots \otimes \mathbf{F}_{\bar{q}+2})^T. \end{aligned}$$

From eq.(27), $\mathcal{G}_{\bar{q}+1}$ is expressed as:

$$\begin{aligned} \mathcal{G}_{\bar{q}+1} &= \text{reshape}(\mathbf{U}^{(\bar{q}+1)}; R_{\bar{q}}, N_{\bar{q}+1}, R_{\bar{q}+1}) \\ &= \bar{\mathcal{T}}_{\bar{q}+1} \times_1 \mathbf{M}_{\bar{q}} \times_2 \mathbf{F}_{\bar{q}+1} \times_3 \mathbf{M}_{\bar{q}+1}^{-T}, \quad (28) \end{aligned}$$

where $\bar{\mathcal{T}}_{\bar{q}+1} = \text{reshape}(\mathbf{I}_{R_{\bar{q}} \times R_{\bar{q}}}; R_{\bar{q}}, T_{\bar{q}+1}, R_{\bar{q}+1})$. From (28), the result can be generalized to the $Q - \bar{q}$ last TT-cores giving

$$\mathcal{G}_q = \bar{\mathcal{T}}_q \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{F}_q \times_3 \mathbf{M}_q^{-T}, (\bar{q} < q < Q). \quad (29)$$

Given the expressions of the TT-cores in (20), (23), (26) and (29), we can therefore deduce the result of Theorem 2.

8.2. Proof of theorem 4

525 To prove Theorem 4, a constructive proof based on the TT-SVD algorithm is given for (4). We assume that the factors have full column rank. For that aim, we will follow step by step the methodology of the TT-SVD algorithm.

- The first unfolding $\mathbf{X}_{(1)}$ of size $N_1 \times (N_2 \cdots N_Q)$, using Lemma 3, is given by:

$$\mathbf{X}_{(1)} = \text{reshape}(\mathcal{X}; N_1, N_2 \cdots N_Q) = \mathbf{P}_1(\mathbf{P}_Q \odot \mathbf{P}_{Q-1} \odot \cdots \odot \mathbf{P}_2)^T.$$

Note that, according to Lemma 1, $(\mathbf{P}_Q \odot \mathbf{P}_{Q-1} \odot \cdots \odot \mathbf{P}_2)^T$ is a full row rank matrix of size $R \times (N_2 \cdots N_Q)$, since the factors \mathbf{P}_q are assumed to be full column rank.

- Applying the SVD to $\mathbf{X}_{(1)}$, the following expression holds:

$$\mathbf{X}_{(1)} = \mathbf{U}^{(1)}\mathbf{V}^{(1)} = \mathbf{G}_1\mathbf{V}^{(1)}$$

where $\mathbf{G}_1 = \mathbf{U}^{(1)}$ contains the left singular vectors, and is the first TT-core of size $N_1 \times R_1$. $\mathbf{V}^{(1)}$ contains the singular values and the the right singular vectors. It is of size $R_1 \times (N_2 \cdots N_Q)$. From the above two relations, we can conclude that

$$\text{rank}(\mathbf{X}_{(1)}) = \text{rank}(\mathbf{P}_1) = \text{rank}(\mathbf{G}_1) = R = R_1.$$

and we can express \mathbf{G}_1 and $\mathbf{V}^{(1)}$ as:

$$\begin{aligned} \mathbf{G}_1 &= \mathbf{P}_1\mathbf{M}_1^{-1}, \\ \mathbf{V}^{(1)} &= \mathbf{M}_1(\mathbf{P}_Q \odot \mathbf{P}_{Q-1} \odot \cdots \odot \mathbf{P}_2)^T \end{aligned} \quad (30)$$

where \mathbf{M}_1 is a $R \times R$ matrix that follows definition 1.

- Applying the TT-SVD algorithm, we have to reshape the matrix $\mathbf{V}^{(1)}$ as a matrix of size $(RN_2) \times (N_3 \cdots N_Q)$, according to Lemma 3, which gives:

$$\mathbf{V}_{(2)}^{(1)} = \text{reshape}(\mathbf{V}^{(1)}; RN_2, N_3 \cdots N_Q) = (\mathbf{P}_2 \odot \mathbf{M}_1)(\mathbf{P}_Q \odot \mathbf{P}_{Q-1} \odot \cdots \odot \mathbf{P}_3)^T. \quad (31)$$

Note that, according to Lemma 1, the matrix $\mathbf{P}_2 \odot \mathbf{M}_1$ of size $(RN_2) \times R$ is full column rank, and $(\mathbf{P}_Q \odot \mathbf{P}_{Q-1} \odot \cdots \odot \mathbf{P}_3)^T$ of size $R \times (N_3 \cdots N_Q)$ is full row rank.

- The SVD of $\mathbf{V}_{(2)}^{(1)}$ gives:

$$\mathbf{V}_{(2)}^{(1)} = \mathbf{U}^{(2)}\mathbf{V}^{(2)} \quad (32)$$

where $\mathbf{U}^{(2)}$ of size $(RN_2) \times R_2$ is the reshaping of the 2nd TT-core \mathcal{G}_2 . From (31) and (32), we can conclude that:

$$\text{rank}\mathbf{V}_{(2)}^{(1)} = \text{rank}(\mathbf{P}_2 \odot \mathbf{M}_1) = \text{rank}\mathbf{U}_2 \stackrel{\text{Lemma 1}}{=} R \stackrel{\text{TT}}{=} R_2$$

and we can write $\mathbf{U}^{(2)}$ and $\mathbf{V}^{(2)}$ as:

$$\begin{aligned}\mathbf{U}^{(2)} &= \mathbf{P}_2 \odot \mathbf{M}_1 \mathbf{M}_2^{-1}, \\ \mathbf{V}^{(2)} &= \mathbf{M}_2 (\mathbf{P}_Q \odot \mathbf{P}_{Q-1} \odot \cdots \odot \mathbf{P}_3)^T\end{aligned}\quad (33)$$

where \mathbf{M}_2 is a $R \times R$ matrix that follows definition 1. Reshaping $\mathbf{U}^{(2)}$, and from (33), we obtain the expression of \mathcal{G}_2 according to

$$\mathcal{G}_2 = \text{reshape}(\mathbf{U}^{(2)}; R, N_2, R) = \mathcal{I}_{3,R} \times_1 \mathbf{M}_1 \times_2 \mathbf{P}_2 \times_3 \mathbf{M}_2^{-T}$$

- Based on the same methodology, we find at the q -th step:

$$\mathbf{V}_{(2)}^{(q-1)} \stackrel{\text{Reshaping}}{=} (\mathbf{P}_q \odot \mathbf{M}_{q-1}) (\mathbf{P}_Q \odot \mathbf{P}_{Q-1} \odot \cdots \odot \mathbf{P}_{q+1})^T \stackrel{\text{SVD}}{=} \mathbf{U}^{(q)} \mathbf{V}^{(q)} \quad (34)$$

where $\mathbf{V}_{(2)}^{(q-1)}$ is of size $(RN_q) \times (N_{q+1} \cdots N_Q)$, and according to Lemma 1, $(\mathbf{P}_q \odot \mathbf{M}_{q-1})$ is a full column rank matrix of size $RN_q \times R$, $(\mathbf{P}_Q \odot \mathbf{P}_{Q-1} \odot \cdots \odot \mathbf{P}_{q+1})^T$ is a full row rank matrix of size $R \times (N_{q+1} \cdots N_Q)$, and \mathbf{M}_{q-1} is a $R \times R$ matrix that follows definition 1.

From (34), we can conclude that the relation between the factor matrices \mathbf{P}_q and the TT-cores \mathcal{G}_q is given by:

$$\mathbf{U}^{(q)} \mathbf{M}_q = \mathbf{P}_q \odot \mathbf{M}_{q-1}$$

or equivalently

$$\mathbf{U}^{(q)} = (\mathbf{P}_q \odot \mathbf{M}_{q-1}) (\mathbf{M}_q^{-T})^T \quad \text{for } 2 \leq q \leq Q-1 \quad (35)$$

where \mathbf{M}_q is a $R \times R$ matrix that follows definition 1, and $\mathbf{U}^{(q)} = \text{reshape}(\mathcal{G}_q; RN_q, R_q)$. We can write:

$$\text{rank} \mathbf{V}_{(2)}^{(q-1)} = \text{rank}(\mathbf{P}_q \odot \mathbf{M}_{q-1}) = \text{rank} \mathbf{U}^{(q)} \stackrel{\text{Lemma 1}}{=} R \stackrel{\text{TT}}{=} R_q,$$

for $2 \leq q \leq Q-1$.

From (35) and considering $\mathbf{U}^{(q)} = \text{reshape}(\mathcal{G}_q; RN_q, R_q)$, we can see that \mathcal{G}_q follows a 3-order CPD according to:

$$\mathcal{G}_q = \mathcal{I}_{3,R} \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{P}_q \times_3 \mathbf{M}_q^{-T} \quad (36)$$

- At the last step, we have:

$$\mathbf{V}_{(2)}^{(Q-2)} \stackrel{\text{Reshaping}}{=} (\mathbf{P}_{Q-1} \odot \mathbf{M}_{Q-2}) \mathbf{P}_Q^T \stackrel{\text{SVD}}{=} \mathbf{U}^{(Q-1)} \mathbf{V}^{(Q-1)} = \mathbf{U}^{(Q-1)} \mathbf{G}_Q.$$

We then have for the last core:

$$\mathbf{G}_Q = \mathbf{M}_{Q-1} \mathbf{P}_Q^T. \quad (37)$$

535 Given (30), (36) and (37), we then have a proof for Theorem 4.

8.3. Proof of theorem 5

The first part of the proof of Theorem 5 is obtained by the use of Theorem 4. In other words, assuming the factors in (9) have full column rank, expressions (10), (11), and (12) are obtained in a straightforward way thanks to Theorem 4.

To prove that two matrices span the same range space, it is equivalent to show the equality of their orthogonal projectors. This is the methodology used to prove (13) and (14). More precisely, assuming a full column rank for factor \mathbf{P} the following equalities hold:

$$\begin{aligned}\mathbf{G}_1 \mathbf{G}_1^\dagger &= (\mathbf{P} \mathbf{M}_1^{-1}) (\mathbf{P} \mathbf{M}_1^{-1})^\dagger \\ &= \mathbf{P} \mathbf{P}^\dagger \\ \mathcal{G}_q(:, :, i)^\dagger \mathcal{G}_q(:, :, i) &= \left((\mathbf{M}_q^{-T}(i, :) \odot \mathbf{M}_{q-1}) \mathbf{P}^T \right)^\dagger \left((\mathbf{M}_q^{-T}(i, :) \odot \mathbf{M}_{q-1}) \mathbf{P}^T \right) \quad (38) \\ &= \mathbf{P}^{T\dagger} \left((\mathbf{M}_q^{-T}(i, :) \odot \mathbf{M}_{q-1})^\dagger (\mathbf{M}_q^{-T}(i, :) \odot \mathbf{M}_{q-1}) \right) \mathbf{P}^T \quad (39) \\ &= (\mathbf{P} \mathbf{P}^\dagger)^T \\ &= \mathbf{P} \mathbf{P}^\dagger\end{aligned}$$

We recall that the i -th frontal slice of the q -th TT-core can be expressed as $\mathcal{G}_q(:, :, i) = (\mathbf{M}_q^{-T}(i, :) \odot \mathbf{M}_{q-1}) \mathbf{P}^T$. Note that, using Lemma 1, we can see that the product $(\mathbf{M}_q^{-T}(i, :) \odot \mathbf{M}_{q-1})$ of size $R \times R$, is of rank R , since all \mathbf{M}_q are of rank R . This justifies the passage from (38) to (39). The same argument is used for the following equalities. Considering the j -th horizontal slice of the q' -th TT-core $\mathcal{G}_{q'}(j, :, :) = \mathbf{P} (\mathbf{M}_{q'}^{-T} \odot \mathbf{M}_{q'-1}(j, :))^T$, we have

$$\begin{aligned}\mathcal{G}_{q'}(j, :, :)^{\dagger} \mathcal{G}_{q'}(j, :, :)^{\dagger} &= \left(\mathbf{P} (\mathbf{M}_{q'}^{-T} \odot \mathbf{M}_{q'-1}(j, :))^T \right)^\dagger \left(\mathbf{P} (\mathbf{M}_{q'}^{-T} \odot \mathbf{M}_{q'-1}(j, :))^T \right)^\dagger \\ &= \mathbf{P} \left((\mathbf{M}_{q'}^{-T} \odot \mathbf{M}_{q'-1}(j, :))^T (\mathbf{M}_{q'}^{-T} \odot \mathbf{M}_{q'-1}(j, :))^T \right)^\dagger \mathbf{P}^\dagger \\ &= \mathbf{P} \mathbf{P}^\dagger \\ \mathbf{G}_Q^T \mathbf{G}_Q^{T\dagger} &= (\mathbf{P} \mathbf{M}_{Q-1}) (\mathbf{P} \mathbf{M}_{Q-1})^\dagger \\ &= \mathbf{P} \mathbf{P}^\dagger\end{aligned}$$

which justifies the equalities

$$\langle \mathbf{P} \rangle = \langle \mathbf{G}_1 \rangle = \langle \mathcal{G}_q(:, :, i) \rangle = \langle \mathcal{G}_{q'}(j, :, :) \rangle = \langle \mathbf{G}_Q^T \rangle,$$

and

$$\text{rank} \mathbf{P} = \text{rank} \mathbf{G}_1 = \text{rank} \mathcal{G}_q(:, :, i) = \text{rank} \mathcal{G}_{q'}(j, :, :) = \text{rank} \mathbf{G}_Q^T,$$

for $2 \leq q \leq Q-1$, $2 \leq q' \leq Q-1$, $1 \leq i \leq R$, $1 \leq j \leq R$.

References

- 540 [1] Badeau, R. and Boyer, R. (2008). Fast multilinear singular value decomposition for structured tensors. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1008–1021.
- [2] Berge, J. T. (2000). The k-rank of a Khatri-Rao product. Unpublished Note, Heijmans Institute of Psychological Research, University of Groningen, The Netherlands.
- 545 [3] Boyer, R. and Badeau, R. (2006). Adaptive multilinear svd for structured tensors. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toulouse, France*.
- [4] Carroll, J. and Chang, J.-J. (1970). Analysis of individual differences in multidimensional scaling via an n -way generalization of 'Eckart-Young' decomposition. *Psychometrika*, 35(3):283–319.
- 550 [5] Cichocki, A. (2014a). Era of big data processing: A new approach via tensor networks and tensor decompositions. *arXiv:1403.2048*.
- [6] Cichocki, A. (2014b). Tensor networks for big data analytics and large-scale optimization problems. *arXiv:1407.3124*.
- 555 [7] Cichocki, A., Lee, N., Oseledets, I., Phan, A.-H., Zhao, Q., and Mandic, D. P. (2016). Tensor networks for dimensionality reduction and large-scale optimization part 1 low-rank tensor decompositions. *Foundations and Trends in Machine Learning*, 9(4-5):249–429.
- [8] Cichocki, A., Mandic, D., De Lathauwer, L., Zhou, G., Zhao, Q., Caiafa, C., and Phan, H. A. (2015). Tensor decompositions for signal processing applications. *IEEE Signal Processing Magazine*, 32(2):145–163.
- 560 [9] Comon, P., Golub, G., Lim, L.-H., and Mourrain, B. (2008). Symmetric tensors and symmetric tensor rank. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1254–1279.
- 565 [10] Comon, P., Luciani, X., and de Almeida, A. L. (2009). Tensor decompositions, alternating least squares and other tales. *Journal of Chemometrics*, 23(7-8):393–405.
- [11] De Lathauwer, L., De Moor, B., and Vandewalle, J. (2000). A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278.
- 570 [12] De Lathauwer, L., De Moor, B., and Vandewalle, J. (2006). On the best rank-1 and rank- (r_1, r_2, \dots, r_n) approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1324–1342.

- [13] De Morais Goulart, J. H., Boyer, M. B. R., Favier, G., and Comon, P. (2016). Tensor CP Decomposition with structured factor matrices: Algorithms and performance. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):757–769.
- [14] Dolgov, S. and Khoromskij, B. (2013). Two-level QTT-Tucker format for optimized tensor calculus. *SIAM Journal on Matrix Analysis and Applications*, 34(2):593–623.
- [15] Domanov, I. and De Lathauwer, L. (2017). Canonical polyadic decomposition of third-order tensors: Relaxed uniqueness conditions and algebraic algorithm. *Linear Algebra and its Applications*, 513(Supplement C):342 – 375.
- [16] Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.
- [17] Favier, G. and de Almeida, A. L. F. (2014). Tensor space-time-frequency coding with semi-blind receivers for mimo wireless communication systems. *IEEE Transactions on Signal Processing*, 62(22):5987–6002.
- [18] Golub, G. H. and Van Loan, C. F. (2013). *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 4th edition.
- [19] Handschuh, S. (2014). *Numerical methods in Tensor Networks*. PhD thesis, Leipzig university.
- [20] Harshman, R. A. (1970). Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84.
- [21] Hillar, C. J. and Lim, L.-H. (2013). Most tensor problems are NP-hard. *Journal of the ACM*, 60(6):45:1–45:39.
- [22] Hitchcock, F. L. (1928). Multiple invariants and generalized rank of a p-way matrix or tensor. *Journal of Mathematics and Physics*, 7(1-4):39–79.
- [23] Jiang, T., Sidiropoulos, N., and ten Berge, J. (2001). Almost sure identifiability of multidimensional harmonic retrieval. *IEEE Transactions on Signal Processing*, 49(9):1849–1859.
- [24] Kammoun, A., Khanfir, H., Altman, Z., Debbah, M., and Kamoun, M. (2014). Preliminary results on 3d channel modeling: From theory to standardization. *IEEE Journal on Selected Areas in Communications*, 32(6):1219–1229.
- [25] Kargas, N. and Sidiropoulos, N. (2017). Completing a joint PMF from projections: a low-rank coupled tensor factorization approach. *Information Theory and Applications Workshop (ITA), San Diego, CA*.

- [26] Kibangou, A. Y. and Favier, G. (2007). Toeplitz-Vandermonde matrix factorization with application to parameter estimation of Wiener-Hammerstein systems. *IEEE Signal Processing Letters*, 14(2):141–144.
- [27] Kibangou, A. Y. and Favier, G. (2009). Non-iterative solution for
615 PARAFAC with a Toeplitz matrix factor. In *EUSIPCO, Glasgow, Scotland*.
- [28] Li, N., Kindermann, S., and Navasca, C. (2013). Some convergence results on the regularized Alternating Least-Squares method for tensor decomposition. *Linear Algebra and its Applications*, 438(2):796–812.
- [29] Lu, H., Plataniotis, K. N., and Venetsanopoulos, A. N. (2011). A survey of multilinear subspace learning for tensor data. *Pattern Recognition*,
620 44(7):1540–1551.
- [30] Oseledets, I. and Tyrtshnikov, E. (2010). TT-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*, 432(1):70 – 88.
- [31] Oseledets, I. and Tyrtshnikov, E. E. (2009a). Tensor tree decomposition
625 does not need a tree. *Unpublished preprint*.
- [32] Oseledets, I. V. (2011). Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317.
- [33] Oseledets, I. V. and Tyrtshnikov, E. E. (2009b). Breaking the curse of
630 dimensionality, or how to use SVD in many dimensions. *SIAM Journal on Scientific Computing*, 31(5):3744–3759.
- [34] Rajih, M., Comon, P., and Harshman, R. A. (2008). Enhanced line search: A novel method to accelerate parafac. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1128–1147.
- [35] Sidiropoulos, N., De Lathauwer, L., Fu, X., Huang, K., Papalexakis, E., and
635 Faloutsos, C. (2017). Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582.
- [36] Sidiropoulos, N. D. and Bro, R. (2000). On the uniqueness of multilinear decomposition of N-way arrays. *Journal of Chemometrics*, 14(3):229–239.
- [37] Sørensen, M. and De Lathauwer, L. (2011). Tensor decompositions with
640 block-Toeplitz structure and applications in signal processing. In *ASILOMAR, Pacific Grove, CA, USA*.
- [38] Sørensen, M. and De Lathauwer, L. (2016). Multidimensional harmonic retrieval via coupled canonical polyadic decomposition - part I: Model and identifiability. *IEEE Transactions on Signal Processing*, 65(2):517 – 527.
- [39] Stegeman, A. and Sidiropoulos, N. D. (2007). On Kruskal’s uniqueness
645 condition for the candecomp/parafac decomposition. *Linear Algebra and its Applications*, 420(2):540 – 552.

- [40] Tomasi, G. and Bro, R. (2006). A comparison of algorithms for fitting the parafac model. *Computational statistics & Data Analysis*, 50(7):1700–1734.
- 650 [41] Tucker, L. R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311.
- [42] Vannieuwenhoven, N., Vandebril, R., and Meerbergen, K. (2012). A new truncation strategy for the higher-order singular value decomposition. *SIAM Journal on Scientific Computing*, 34(2):1027–1052.
- 655 [43] Vasilescu, M. and Terzopoulos, D. (2003). Multilinear subspace analysis of image ensembles. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, WI, USA*.
- [44] Ximenes, L. R., Favier, G., and de Almeida, A. L. F. (2016). Closed-form semi-blind receiver for mimo relay systems using double Khatri-Rao space-time coding. *IEEE Signal Processing Letters*, 23(3):316–320.
- 660 [45] Zniyed, Y., Boyer, R., Almeida, A. L. D., and Favier, G. (2018). High-order CPD estimator with dimensionality reduction using a tensor train model. In *EUSIPCO, Rome, Italy*.