Citation/Reference	Sorensen M., Domanov I., De Lathauwer L., ``Coupled Canonical Polyadic Decompositions and (Coupled) Decompositions in Multilinear rank-(L_r,n,L_r,n,1) terms --- Part II: Algorithms", *SIAM Journal on Matrix Analysis and Applications*, vol. 36, no. 3, Jul. 2015, pp. 1015-1045
Archived version	Author manuscript: Final publisher's version / pdf
Journal homepage	insert link to the journal homepage of your paper . http://epubs.siam.org/journal/sjmael
Author contact	your email mikael.sorensen@kuleuven.be
IR	url in Lirias https://lirias.kuleuven.be/handle/123456789/463244

(article begins on next page)

SIAM J. MATRIX ANAL. APPL.
Vol. 36, No. 3, pp. 1015–1045

COUPLED CANONICAL POLYADIC DECOMPOSITIONS AND (COUPLED) DECOMPOSITIONS IN MULTILINEAR RANK-($L_{r,n}, L_{r,n}, 1$) TERMS—PART II: ALGORITHMS*

MIKAEL SØRENSEN†, IGNAT DOMANOV†, AND LIEVEN DE LATHAUWER†

Abstract. The coupled canonical polyadic decomposition (CPD) is an emerging tool for the joint analysis of multiple data sets in signal processing and statistics. Despite their importance, linear algebra based algorithms for coupled CPDs have not yet been developed. In this paper, we first explain how to obtain a coupled CPD from one of the individual CPDs. Next, we present an algorithm that directly takes the coupling between several CPDs into account. We extend the methods to single and coupled decompositions in multilinear rank-($L_{r,n}, L_{r,n}, 1$) terms. Finally, numerical experiments demonstrate that linear algebra based algorithms can provide good results at a reasonable computational cost.

Key words. coupled decompositions, higher-order tensor, polyadic decomposition, parallel factor, canonical decomposition, canonical polyadic decomposition, coupled matrix-tensor factorization

AMS subject classifications. 15A22, 15A23, 15A69

DOI. 10.1137/140956865

1. Introduction. In recent years the coupled canonical polyadic decomposition (CPD) and its variants have found many applications in science and engineering, ranging from psychometrics, chemometrics, data mining, and bioinformatics to biomedical engineering and signal processing. For an overview and references to concrete applications we refer the reader to [35, 33]. For a more general background on tensor decompositions, we refer the reader to the review papers [22, 4, 6] and references therein. It was demonstrated in [35] that improved uniqueness conditions can be obtained by taking the coupling between several coupled CPDs into account. We can expect that it is also advantageous to take the coupling between the tensors into account in the actual computation.

There are two main approaches to computing a tensor decomposition, namely, linear algebra (e.g., [24, 9, 14]) and optimization based methods (e.g., [37, 5, 30]). For many exact coupled decomposition problems an explicit solution can be obtained by means of linear algebra. However, in practice data are noisy, and consequently the estimates are inexact. In many cases the explicit solution obtained by linear algebra is still accurate enough. If not, then the explicit solution may be used to initialize an optimization based method. On the other hand, optimization based methods for

*Received by the editors February 12, 2014; accepted for publication (in revised form) by D. P. O'Leary April 27, 2015; published electronically July 21, 2015. This research was supported by Research Council KU Leuven, GOA/10/09 MaNet, CoE PFV/10/002 (OPTEC); F.W.O. project G.0830.14N, G.0881.14N; and the Belgian Federal Science Policy Office, IUAP P7 (DYSCO II, Dynamical Systems, Control and Optimization, 2012–2017). The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC Advanced Grant: BIOTENSORS (339804). This paper reflects only the authors' views, and the Union is not liable for any use that may be made of the contained information.
http://www.siam.org/journals/simax/36-3/95686.html

†Group Science, Engineering and Technology, KU Leuven - Kulak, 8500 Kortrijk, Belgium, and E.E. Department (ESAT) - STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, and iMinds Medical IT Department, KU Leuven, B-3001 Leuven-Heverlee, Belgium (Mikael.Sorensen@kuleuven-kulak.be, Ignat.Domanov@kuleuven-kulak.be, Lieven.DeLathauwer@kuleuven-kulak.be).

coupled decompositions may work well in the case of noisy data but are not formally guaranteed to find the decomposition (i.e., the global optimum), even in the exact case.

So far, mainly optimization based methods for computing the coupled CPD have been proposed (e.g., [1, 32]). The goal of this paper is to develop algebraic methods for computing coupled CPDs. In contrast to optimization based methods, algebraic methods are under certain working conditions guaranteed to find the decomposition in the exact case. We first explain how to compute a coupled CPD by first computing one of the individual CPDs, and then handling the remaining ones as CPDs with a known factor. Next, we present an algorithm that simultaneously takes the coupling between the different CPDs into account. In signal processing polyadic decompositions (PDs) may contain factor matrices with collinear columns, known as block term decompositions (BTDs) [10, 11, 12]. For a further motivation, see [35, 33] and references therein. Consequently, we also extend the algebraic framework to single or coupled decompositions in multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ terms. This also leads to a new uniqueness condition for single/coupled decompositions in multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ terms.

The paper is organized as follows. The rest of the introduction presents our notation. Sections 2 and 3 briefly define the coupled CPD without and with a common factor matrix with collinear components, respectively. Next, in section 4 we present algorithms for computing the coupled CPD. Section 5 considers CPD models where the common factor matrix contains collinear components. Numerical experiments are reported in section 6. We end the paper with a conclusion in section 7. We also mention that in the supplementary materials an efficient implementation of the iterative alternating least squares (ALS) method for coupled decompositions is reported.

1.1. Notation. Vectors, matrices, and tensors are denoted by lowercase bold, uppercase bold, and uppercase calligraphic letters, respectively. The rth column vector of $\mathbf{A}$ is denoted by $\mathbf{a}_r$. The symbols $\otimes$ and $\odot$ denote the Kronecker and Khatri–Rao product, defined as

$$\mathbf{A} \otimes \mathbf{B} := \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}, \quad \mathbf{A} \odot \mathbf{B} := \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_2 \otimes \mathbf{b}_2 & \dots \end{bmatrix},$$

in which $(\mathbf{A})_{mn} = a_{mn}$. The Hadamard product is given by $(\mathbf{A} * \mathbf{B})_{ij} = a_{ij}b_{ij}$. The outer product of N vectors $\mathbf{a}^{(n)} \in \mathbb{C}^{I_n}$ is denoted by $\mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \dots \circ \mathbf{a}^{(N)} \in \mathbb{C}^{I_1 \times I_2 \times \dots \times I_N}$, such that

$$\left(\mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \dots \circ \mathbf{a}^{(N)}\right)_{i_1,i_2,\dots,i_N} = a^{(1)}_{i_1} a^{(2)}_{i_2} \cdots a^{(N)}_{i_N}.$$

The identity matrix, all-zero matrix, and all-zero vector are denoted by $\mathbf{I}_M \in \mathbb{C}^{M\times M}$, $\mathbf{0}_{M,N} \in \mathbb{C}^{M\times N}$, and $\mathbf{0}_M \in \mathbb{C}^M$, respectively. The all-ones vector is denoted by $\mathbf{1}_R = [1,\dots,1]^T \in \mathbb{C}^R$. Dirac's delta function is defined as

$$\delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

The cardinality of a set S is denoted by $\mathrm{card}\,(S)$.

The transpose, conjugate, conjugate-transpose, inverse, Moore–Penrose pseudo-inverse, Frobenius norm, determinant, range, and kernel of a matrix are denoted by

$(\cdot)^T$, $(\cdot)^*$, $(\cdot)^H$, $(\cdot)^{-1}$, $(\cdot)^\dagger$, $\|\cdot\|_F$, $|\cdot|$, range $(\cdot)$, and ker $(\cdot)$, respectively. The orthogonal sum of subspaces is denoted by $\oplus$.

MATLAB index notation will be used for submatrices of a given matrix. For example, $\mathbf{A}(1:k,:)$ represents the submatrix of $\mathbf{A}$ consisting of the rows from 1 to k of $\mathbf{A}$. $D_k(\mathbf{A}) \in \mathbb{C}^{J\times J}$ denotes the diagonal matrix holding row k of $\mathbf{A} \in \mathbb{C}^{I\times J}$ on its diagonal. Similarly, $\mathrm{Diag}(\mathbf{a}) \in \mathbb{C}^{I\times I}$ denotes the diagonal matrix holding the elements of the vector $\mathbf{a} \in \mathbb{C}^I$ on its main diagonal. Given $\mathcal{X} \in \mathbb{C}^{I_1\times I_2\times\cdots\times I_N}$, $\mathrm{Vec}(\mathcal{X}) \in \mathbb{C}^{\prod_{n=1}^N I_n}$ denotes the column vector

$$\mathrm{Vec}(\mathcal{X}) = \left[x_{1,\ldots,1,1},\ x_{1,\ldots,1,2}, \ldots, x_{I_1,\ldots,I_{N-1},I_N}\right]^T.$$

The reverse operation is $\mathrm{Unvec}(\mathrm{Vec}(\mathcal{X})) = \mathcal{X}$. Let $\mathbf{A} \in \mathbb{C}^{I\times I}$; then $\mathrm{Vecd}(\mathbf{A}) \in \mathbb{C}^I$ denotes the column vector defined by $(\mathrm{Vecd}(\mathbf{A}))_i = (\mathbf{A})_{ii}$.

The matrix that orthogonally projects on the orthogonal complement of the column space of $\mathbf{A} \in \mathbb{C}^{I\times J}$ is denoted by

$$\mathbf{P}_{\mathbf{A}} = \mathbf{I}_I - \mathbf{F}\mathbf{F}^H \in \mathbb{C}^{I\times I},$$

where the column vectors of $\mathbf{F}$ constitute an orthonormal basis for range $(\mathbf{A})$.

The rank of a matrix $\mathbf{A}$ is denoted by $r(\mathbf{A})$ or $r_{\mathbf{A}}$. The k-rank of a matrix $\mathbf{A}$ is denoted by $k(\mathbf{A})$. It is equal to the largest integer $k(\mathbf{A})$ such that every subset of $k(\mathbf{A})$ columns of $\mathbf{A}$ is linearly independent. Let $C_n^k = \frac{n!}{k!(n-k)!}$ denote the binomial coefficient. The kth compound matrix of $\mathbf{A} \in \mathbb{C}^{m\times n}$ is denoted by $\mathcal{C}_k(\mathbf{A}) \in \mathbb{C}^{C_m^k \times C_n^k}$, and its entries correspond to the k-by-k minors of $\mathbf{A}$ ordered lexicographically. See [20, 13] for a discussion of compound matrices.

2. Coupled canonical polyadic decomposition. We say that $\mathbf{a}\circ\mathbf{b}\circ\mathbf{c} \in \mathbb{C}^{I\times J\times K}$ is a rank-1 tensor if it is equal to the outer product of some nonzero vectors $\mathbf{a} \in \mathbb{C}^I$, $\mathbf{b} \in \mathbb{C}^J$, and $\mathbf{c} \in \mathbb{C}^K$. The decomposition of a tensor $\mathcal{X} \in \mathbb{C}^{I\times J\times K}$ into a minimal number of rank-1 tensors is called the canonical polyadic decomposition (CPD). We say that a set of tensors $\mathbf{a}^{(n)}\circ\mathbf{b}^{(n)}\circ\mathbf{c} \in \mathbb{C}^{I_n\times J_n\times K}$, $n \in \{1,\ldots,N\}$, is a coupled rank-1 tensor if at least one of the involved tensors $\mathbf{a}^{(n)}\circ\mathbf{b}^{(n)}\circ\mathbf{c}$ is nonzero, where "coupled" means that the set of tensors $\{\mathbf{a}^{(n)}\circ\mathbf{b}^{(n)}\circ\mathbf{c}\}$ share the third-mode vector $\mathbf{c}$. A decomposition of a set of tensors $\mathcal{X}^{(n)} \in \mathbb{C}^{I_n\times J_n\times K}$, $n \in \{1,\ldots,N\}$, into a sum of coupled rank-1 tensors of the form

$$\mathcal{X}^{(n)} = \sum_{r=1}^{R} \mathbf{a}_r^{(n)}\circ\mathbf{b}_r^{(n)}\circ\mathbf{c}_r, \quad n \in \{1,\ldots,N\}, \tag{2.1}$$

is called a coupled polyadic decomposition (PD). The factor matrices in the first and second modes are

$$\mathbf{A}^{(n)} = \left[\ \mathbf{a}_1^{(n)},\ \ \ldots\ \ ,\mathbf{a}_R^{(n)}\ \right] \in \mathbb{C}^{I_n\times R}, \quad n \in \{1,\ldots,N\},$$
$$\mathbf{B}^{(n)} = \left[\ \mathbf{b}_1^{(n)},\ \ \ldots\ \ ,\mathbf{b}_R^{(n)}\ \right] \in \mathbb{C}^{J_n\times R}, \quad n \in \{1,\ldots,N\}.$$

The factor matrix in the third mode,

$$\mathbf{C} = \left[\ \mathbf{c}_1,\ \ \ldots\ \ ,\mathbf{c}_R\ \right] \in \mathbb{C}^{K\times R},$$

is common to all terms. Note that the columns of $\mathbf{C}$ are nonzero, while columns of $\mathbf{A}^{(n)}$ and $\mathbf{B}^{(n)}$ can be zero. We define the coupled rank of $\{\mathcal{X}^{(n)}\}$ as the minimal number of coupled rank-1 tensors $\mathbf{a}_r^{(n)} \circ \mathbf{b}_r^{(n)} \circ \mathbf{c}_r$ that yield $\{\mathcal{X}^{(n)}\}$ in a linear combination. Since each third-mode vector is shared across a coupled rank-1 tensor, the coupled CPD of $\{\mathcal{X}^{(n)}\}$ leads to a different decomposition compared to ordinary CPDs of the individual tensors in $\{\mathcal{X}^{(n)}\}$. If R in (2.1) equals the coupled rank of $\{\mathcal{X}^{(n)}\}$, then (2.1) is called a coupled CPD. The coupled rank-1 tensors in (2.1) can be arbitrarily permuted, and the vectors within the same coupled rank-1 tensor can be arbitrarily scaled provided the overall coupled rank-1 term remains the same. We say that the coupled CPD is unique when it is only subject to these trivial indeterminacies. Uniqueness conditions for the coupled CPD have been derived in [35].

A special case of (2.1) is the coupled matrix-tensor factorization

$$\begin{cases} \mathcal{X}^{(1)} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{b}_r^{(1)} \circ \mathbf{c}_r, \\ \mathbf{X}^{(2)} = \sum_{r=1}^{R} \mathbf{a}_r^{(2)} \circ \mathbf{c}_r. \end{cases} \tag{2.2}$$

2.1. Matrix representation. Let $\mathbf{X}^{(i\cdot\cdot,n)} \in \mathbb{C}^{J_n \times K}$ denote the matrix slice for which $(\mathbf{X}^{(i\cdot\cdot,n)})_{jk} = x_{ijk}^{(n)}$; then $\mathbf{X}^{(i\cdot\cdot,n)} = \mathbf{B}^{(n)} D_i(\mathbf{A}^{(n)}) \mathbf{C}^T$ and

$$\mathbb{C}^{I_n J_n \times K} \ni \mathbf{X}_{(1)}^{(n)} := \left[\mathbf{X}^{(1\cdot\cdot,n)T}, \ldots, \mathbf{X}^{(I_n\cdot\cdot,n)T}\right]^T = \left(\mathbf{A}^{(n)} \odot \mathbf{B}^{(n)}\right) \mathbf{C}^T. \tag{2.3}$$

Similarly, let $\mathbf{X}^{(\cdot\cdot k,n)} \in \mathbb{C}^{I_n \times J_n}$ be such that $(\mathbf{X}^{(\cdot\cdot k,n)})_{ij} = x_{ijk}^{(n)}$; then $\mathbf{X}^{(\cdot\cdot k,n)} = \mathbf{A}^{(n)} D_k(\mathbf{C}) \mathbf{B}^{(n)T}$ and

$$\mathbb{C}^{I_n K \times J_n} \ni \mathbf{X}_{(3)}^{(n)} := \left[\mathbf{X}^{(\cdot\cdot 1,n)T}, \ldots, \mathbf{X}^{(\cdot\cdot K,n)T}\right]^T = \left(\mathbf{C} \odot \mathbf{A}^{(n)}\right) \mathbf{B}^{(n)T}. \tag{2.4}$$

By stacking expressions of the type (2.3), we obtain the following overall matrix representation of the coupled PD of $\{\mathcal{X}^{(n)}\}$:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_{(1)}^{(1)} \\ \vdots \\ \mathbf{X}_{(1)}^{(N)} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^{(1)} \odot \mathbf{B}^{(1)} \\ \vdots \\ \mathbf{A}^{(N)} \odot \mathbf{B}^{(N)} \end{bmatrix} \mathbf{C}^T = \mathbf{F}\mathbf{C}^T \in \mathbb{C}^{(\sum_{n=1}^{N} I_n J_n) \times K}, \tag{2.5}$$

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{A}^{(1)} \odot \mathbf{B}^{(1)} \\ \vdots \\ \mathbf{A}^{(N)} \odot \mathbf{B}^{(N)} \end{bmatrix} \in \mathbb{C}^{(\sum_{n=1}^{N} I_n J_n) \times R}. \tag{2.6}$$

3. Coupled BTD.

We consider PDs of the following form:

$$\mathcal{X} = \sum_{r=1}^{R} \sum_{l=1}^{L_r} \mathbf{a}_l^{(r)} \circ \mathbf{b}_l^{(r)} \circ \mathbf{c}^{(r)} = \sum_{r=1}^{R} \left(\mathbf{A}^{(r)} \mathbf{B}^{(r)T}\right) \circ \mathbf{c}^{(r)}. \tag{3.1}$$

Equation (3.1) can be seen as a PD with collinear columns $\mathbf{c}^{(r)}$ in the third factor matrix. We say that $(\mathbf{A}\mathbf{B}^T) \circ \mathbf{c}$ is a multilinear rank-$(L, L, 1)$ tensor if $\mathbf{A}\mathbf{B}^T$ has rank L and $\mathbf{c}$ is a nonzero vector. If the matrices $\mathbf{A}^{(r)}\mathbf{B}^{(r)T}$ in (3.1) have rank L_r, then (3.1) corresponds to a decomposition into multilinear rank-$(L_r, L_r, 1)$ terms [10]. Uniqueness conditions for the decomposition of $\mathcal{X}$ into multilinear rank-$(L_r, L_r, 1)$ terms can, for instance, be found in [10, 11, 27].

We say that a set of tensors $(\mathbf{A}^{(n)}\mathbf{B}^{(n)T}) \circ \mathbf{c} \in \mathbb{C}^{I_n \times J_n \times K}$, $n \in \{1,\ldots,N\}$, is a coupled multilinear rank-$(L_n, L_n, 1)$ tensor if at least one of the involved tensors $(\mathbf{A}^{(n)}\mathbf{B}^{(n)T}) \circ \mathbf{c}$ is a multilinear rank-$(L_n, L_n, 1)$ tensor, where again "coupled" means that the set of tensors $\{(\mathbf{A}^{(n)}\mathbf{B}^{(n)T}) \circ \mathbf{c}\}$ shares the third-mode vector $\mathbf{c}$. In this paper we consider a decomposition of a set of tensors $\mathcal{X}^{(n)} \in \mathbb{C}^{I_n \times J_n \times K}$, $n \in \{1,\ldots,N\}$, into a sum of coupled multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ tensors of the following form:

$$\mathcal{X}^{(n)} = \sum_{r=1}^{R} \sum_{l=1}^{L_{r,n}} \mathbf{a}_l^{(r,n)} \circ \mathbf{b}_l^{(r,n)} \circ \mathbf{c}^{(r)} = \sum_{r=1}^{R} \left(\mathbf{A}^{(r,n)}\mathbf{B}^{(r,n)T}\right) \circ \mathbf{c}^{(r)}. \tag{3.2}$$

We call the coupled multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ term decomposition (3.2) a coupled block term decomposition (BTD) for brevity.

The coupled multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ tensors in (3.2) can be arbitrarily permuted without changing the decomposition. The vectors or matrices within the same coupled multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ tensor can also be arbitrarily scaled or transformed, provided that the overall coupled multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ term remains the same (e.g., $(\mathbf{A}^{(r,n)}\mathbf{B}^{(r,n)T}) \circ \mathbf{c}^{(r)} = (2 \cdot \mathbf{A}^{(r,n)}\mathbf{N})(3 \cdot \mathbf{B}^{(r,n)}\mathbf{N}^{-T})^T \circ \frac{1}{6}\mathbf{c}^{(r)}$, where $\mathbf{N}$ is an arbitrary nonsingular matrix). We say that the coupled BTD is unique when it is only subject to the mentioned indeterminacies. Uniqueness conditions for the coupled BTD are given in [35].

3.1. Matrix representations.

Denote $R_{\text{tot},n} = \sum_{r=1}^{R} L_{r,n}$, and define

$$\mathbf{A}^{(r,n)} = \left[\ \mathbf{a}_1^{(r,n)}, \ \ldots \ , \mathbf{a}_{L_{r,n}}^{(r,n)}\ \right] \in \mathbb{C}^{I_n \times L_{r,n}},$$

$$\mathbf{A}^{(n)} = \left[\ \mathbf{A}^{(1,n)}, \ \ldots \ , \mathbf{A}^{(R,n)}\ \right] \in \mathbb{C}^{I_n \times R_{\text{tot},n}}, \quad n \in \{1,\ldots,N\},$$

$$\mathbf{B}^{(r,n)} = \left[\ \mathbf{b}_1^{(r,n)}, \ \ldots \ , \mathbf{b}_{L_{r,n}}^{(r,n)}\ \right] \in \mathbb{C}^{J_n \times L_{r,n}},$$

$$\mathbf{B}^{(n)} = \left[\ \mathbf{B}^{(1,n)}, \ \ldots \ , \mathbf{B}^{(R,n)}\ \right] \in \mathbb{C}^{J_n \times R_{\text{tot},n}}, \quad n \in \{1,\ldots,N\},$$

$$\mathbf{C}^{(\text{red})} = \left[\ \mathbf{c}^{(1)}, \ \ldots \ , \mathbf{c}^{(R)}\ \right] \in \mathbb{C}^{K \times R}, \tag{3.3}$$

$$\mathbf{C}^{(n)} = \left[\mathbf{1}_{L_{r,n}}^T \otimes \mathbf{c}^{(1)}, \ldots, \mathbf{1}_{L_{R,n}}^T \otimes \mathbf{c}^{(R)}\right] \in \mathbb{C}^{K \times R_{\text{tot},n}}, \tag{3.4}$$

where "red" stands for reduced. We have the following analogues of (2.3)–(2.4):

$$\mathbb{C}^{I_n J_n \times K} \ni \mathbf{X}_{(1)}^{(n)} = \left[\mathbf{X}^{(1\cdot\cdot,n)T}, \ldots, \mathbf{X}^{(I_n\cdot\cdot,n)T}\right]^T = \left(\mathbf{A}^{(n)} \odot \mathbf{B}^{(n)}\right)\mathbf{C}^{(n)T}, \tag{3.5}$$

$$\mathbb{C}^{I_n K \times J_n} \ni \mathbf{X}_{(3)}^{(n)} = \left[\mathbf{X}^{(\cdot\cdot 1,n)T}, \ldots, \mathbf{X}^{(\cdot\cdot K,n)T}\right]^T = \left(\mathbf{C}^{(n)} \odot \mathbf{A}^{(n)}\right)\mathbf{B}^{(n)T}. \tag{3.6}$$

Similar to (2.5), we have the following matrix representation of (3.2):

$$\mathbf{X} = \left[\mathbf{X}_{(1)}^{(1)T}, \ldots, \mathbf{X}_{(1)}^{(N)T}\right]^T = \mathbf{F}^{(\text{red})}\mathbf{C}^{(\text{red})T} \in \mathbb{C}^{(\sum_{n=1}^{N} I_n J_n) \times K}, \tag{3.7}$$

where $\mathbf{F}^{(\text{red})} \in \mathbb{C}^{(\sum_{n=1}^{N} I_n J_n) \times R}$ is given by

$$\mathbf{F}^{(\text{red})} = \begin{bmatrix} \text{Vec}\left(\mathbf{B}^{(1,1)}\mathbf{A}^{(1,1)T}\right) & \cdots & \text{Vec}\left(\mathbf{B}^{(R,1)}\mathbf{A}^{(R,1)T}\right) \\ \vdots & \ddots & \vdots \\ \text{Vec}\left(\mathbf{B}^{(1,N)}\mathbf{A}^{(1,N)T}\right) & \cdots & \text{Vec}\left(\mathbf{B}^{(R,N)}\mathbf{A}^{(R,N)T}\right) \end{bmatrix}. \tag{3.8}$$

4. Algorithms for computing the coupled CPD. So far, for the computation of the coupled CPD, mainly optimization based methods have been proposed (e.g., [1, 32]). Standard unconstrained optimization methods proposed for ordinary CPDs (e.g., nonlinear least squares methods) can be adapted to coupled CPDs; see [1, 32] and references therein for details. A linear algebra based method for the computation of the coupled CPD of two tensors has been suggested in [17]. However, the method requires that each individual CPD be unique and have a full column rank factor matrix. We also mention that in the case where all factor matrices $\{\mathbf{A}^{(n)}\}$ and $\mathbf{C}$ in (2.1) have full column rank, it is possible to transform the coupled CPD problem into an ordinary CPD problem via a joint similarity transform [2]. As in [17], a drawback of this approach is that it basically requires the individual CPDs to be unique. In contrast, we first present in subsection 4.1 a linear algebra inspired method for coupled CPD problems in which only one of the involved CPDs is required to be unique. Next, in subsection 4.2 we present a linear algebra inspired method for coupled CPD problems which only requires that the common factor matrix have full column rank (i.e., none of the individual CPDs is required to be unique).

4.1. Coupled CPD via ordinary CPD. Consider the coupled CPD of the third-order tensors $\mathcal{X}^{(n)}$, $n \in \{1, \ldots, N\}$, in (2.1). Under the conditions in [35, Theorem 4.4] the coupled CPD inherits uniqueness from one of the individual CPDs. Assume that the CPD of $\mathcal{X}^{(p)}$ with matrix representation

$$\mathbf{X}^{(p)}_{(1)} = \left(\mathbf{A}^{(p)} \odot \mathbf{B}^{(p)}\right) \mathbf{C}^T \tag{4.1}$$

is unique for some $p \in \{1, \ldots, N\}$. We first compute this CPD. Linear algebra based methods for the computation of the CPD can be found in [24, 9, 36, 14]. For instance, if $\mathbf{A}^{(p)}$ and $C_2(\mathbf{B}^{(p)}) \odot C_2(\mathbf{C})$ have full column rank, then the simultaneous diagonalization (SD) method in [9, 14], reviewed in subsection 4.2.1, can be applied. Optimization based methods can also be used to compute the CPD of $\mathcal{X}^{(p)}$; see [22, 30] and references therein. Next, the remaining CPDs may be computed as "CPDs with a known factor matrix" (i.e., matrix $\mathbf{C}$):

$$\mathbf{X}^{(n)}_{(1)} = \left(\mathbf{A}^{(n)} \odot \mathbf{B}^{(n)}\right) \mathbf{C}^T, \quad n \in \{1, \ldots, N\} \setminus p.$$

If $\mathbf{C}$ has full column rank, then the remaining factor matrices of the coupled CPD of $\{\mathcal{X}^{(n)}\}$ follow from the well-known fact that the columns of $\mathbf{Y}^{(n)}_{(1)} = \mathbf{X}^{(n)}_{(1)}(\mathbf{C}^T)^\dagger = \mathbf{A}^{(n)} \odot \mathbf{B}^{(n)}$, $n \in \{1, \ldots, N\} \setminus p$, correspond to vectorized rank-1 matrices. For the case where $\mathbf{C}$ does not have full column rank, a dedicated algorithm is discussed in [34]. The results may afterward be refined by an optimization algorithm such as ALS, discussed in the supplementary materials. The extension to coupled CPDs of M_nth-order tensors with $M_n \geq 4$ for one or more $n \in \{1, \ldots, N\}$ is straightforward.

For the coupled matrix-tensor factorization problem (2.2), the factor matrix $\mathbf{C}$ is required to have full column rank in order to guarantee uniqueness of $\mathbf{A}^{(2)}$ [34]. Consequently, we may first compute the CPD of the tensor $\mathcal{X}^{(1)}$ in (2.2) and thereafter obtain the remaining factor as $\mathbf{A}^{(2)} = \mathbf{X}^{(2)}(\mathbf{C}^T)^\dagger$.

4.2. Simultaneous diagonalization (SD) method for coupled CPDs. In [9] the computation of a CPD of a third-order tensor was reduced to a matrix generalized eigenvalue decomposition (GEVD) in cases where only one of the factor matrices has full column rank. This generalizes the more common use of GEVD in cases where at least two of the factor matrices have full column rank [24]. In this subsection, first we briefly recall the result from [9], following the notation of [14]. For simplicity we will explain the result for the noiseless case and assume that the third factor matrix is square. Then we present a generalization for coupled CPDs. For this contribution, we will consider the noisy case, and we will just assume that the third factor matrix has full column rank.

4.2.1. Single CPD. Let $\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$ be an $I \times J \times R$ tensor with frontal slices $\mathbf{X}(:,:,1), \ldots, \mathbf{X}(:,:,R)$. The basic idea behind the SD procedure is to consider the tensor decomposition problem of $\mathcal{X}$ as a structured matrix decomposition problem of the form

$$\mathbf{X}_{(1)} = \mathbf{F}\mathbf{C}^T, \tag{4.2}$$

where $\mathbf{F}$ is subject to a constraint depending on the decomposition under consideration. In the single CPD case, $\mathbf{F}$ is subject to the Khatri–Rao product constraint $\mathbf{F} = \mathbf{A} \odot \mathbf{B}$; i.e., the columns of $\mathbf{F}$ are assumed to be vectorized rank-1 matrices. The other way around, we can interpret a rank constrained matrix decomposition problem of the form (4.2) as a CPD problem. By capitalizing on the structure of $\mathbf{F}$, the SD method transforms the constrained decomposition problem in (4.2) into an SD problem involving a congruence transform, as will be explained in this section. The advantage of the SD method is that in the exact case it reduces a tensor decomposition problem into a generalized eigenvalue problem, which in turn can be solved by means of standard numerical linear algebra methods (e.g., [16]). We assume that

$$\begin{cases} \mathbf{C} \ \text{ has full column rank,} \\ \mathcal{C}_2(\mathbf{A}) \odot \mathcal{C}_2(\mathbf{B}) \ \text{ has full column rank.} \end{cases} \tag{4.3}$$

If condition (4.3) is satisfied, then the rank of $\mathcal{X}$ is R, the CPD of $\mathcal{X}$ is unique, and the factor matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ can be determined via the SD method [9, 13]. In other words, condition (4.3) ensures that scaled versions of $\mathbf{a}_r \otimes \mathbf{b}_r$, $r \in \{1, \ldots, R\}$, are the only Kronecker-structured vectors in range$\left(\mathbf{X}_{(1)}\right)$.

We define a $C_I^2 C_J^2 \times R^2$ matrix $\mathbf{R}_2(\mathcal{X})$ that has columns

$$\operatorname{Vec}\left(\mathcal{C}_2(\mathbf{X}(:,:,r_1) + \mathbf{X}(:,:,r_2)) - \mathcal{C}_2(\mathbf{X}(:,:,r_1)) - \mathcal{C}_2(\mathbf{X}(:,:,r_2)) \right), \quad 1 \leq r_1, r_2 \leq R, \tag{4.4}$$

where $\mathcal{C}_2(\cdot)$ denotes the second compound matrix of its argument and is defined in subsection 1.1. We also define an $R^2 \times C_R^2$ matrix $\mathcal{R}_2(\mathbf{C})$ that has columns

$$\frac{1}{2}(\mathbf{c}_{r_1} \otimes \mathbf{c}_{r_2} + \mathbf{c}_{r_2} \otimes \mathbf{c}_{r_1}), \quad 1 \leq r_1 < r_2 \leq R.$$

So the columns of $\mathbf{R}_2(\mathcal{X})$ (resp., $\mathcal{R}_2(\mathbf{C})$) can be enumerated by means of R^2 (resp., C_R^2) pairs (r_1, r_2). For both matrices we follow the convention that the column associated with the pair (r_1, r_2) is preceding the column associated with the pair (r_1', r_2') if and only if either $r_1' > r_1$ or $r_1' = r_1$ and $r_2' > r_2$.

Expression (4.4) implies the following entrywise definition of $\mathbf{R}_2(\mathcal{X})$: if $1 \leq i_1 <$

$i_2 \le I$, $1 \le j_1 < j_2 \le J$, and $1 \le r_1, r_2 \le R$, then

$$\text{the } \left(\frac{(j_1(2j_2-j_1-1)-2)I(I-1)}{4} + \frac{i_1(2i_2-i_1-1)}{2}, (r_2-1)R+r_1\right)\text{th entry of the matrix } \mathbf{R}_2(\mathcal{X}) \text{ is equal to}$$
$$\begin{vmatrix} x_{i_1j_1r_1}+x_{i_1j_1r_2} & x_{i_1j_2r_1}+x_{i_1j_2r_2} \\ x_{i_2j_1r_1}+x_{i_2j_1r_2} & x_{i_2j_2r_1}+x_{i_2j_2r_2}\end{vmatrix} - \begin{vmatrix} x_{i_1j_1r_1} & x_{i_1j_2r_1} \\ x_{i_2j_1r_1} & x_{i_2j_2r_1}\end{vmatrix} - \begin{vmatrix} x_{i_1j_1r_2} & x_{i_1j_2r_2} \\ x_{i_2j_1r_2} & x_{i_2j_2r_2}\end{vmatrix}$$
$$= x_{i_1j_1r_1}x_{i_2j_2r_2} + x_{i_1j_1r_2}x_{i_2j_2r_1} - x_{i_1j_2r_1}x_{i_2j_1r_2} - x_{i_1j_2r_2}x_{i_2j_1r_1}. \tag{4.5}$$

Since (4.5) is invariant under permutation of r_1 and r_2, $\mathbf{R}_2(\mathcal{X})$ only consists of C_{R+1}^2 distinct columns (i.e., switching r_1 and r_2 in (4.5) will not change $\mathbf{R}_2(\mathcal{X})$).

Let $\pi_S : \mathbb{C}^{R^2} \to \mathbb{C}^{R^2}$ be a symmetrization mapping:

$$\pi_S(\operatorname{Vec}(\mathbf{F})) = \operatorname{Vec}\left((\mathbf{F}+\mathbf{F}^T)/2\right), \quad \mathbf{F} \in \mathbb{C}^{R\times R};$$

i.e., π_S is the vectorized version of the mapping that sends an arbitrary $R \times R$ matrix to its symmetric part. It is clear that $\dim \operatorname{range}(\pi_S) = R(R+1)/2$ (dimension of the subspace of the symmetric $R \times R$ matrices) and that

$$\pi_S(\mathbf{x}\otimes\mathbf{y}) = \pi_S(\operatorname{Vec}(\mathbf{y}\mathbf{x}^T)) = \operatorname{Vec}\left((\mathbf{y}\mathbf{x}^T + \mathbf{x}\mathbf{y}^T)/2\right) = \mathbf{x}\otimes\mathbf{y} + \mathbf{y}\otimes\mathbf{x}, \quad \mathbf{x},\mathbf{y} \in \mathbb{C}^R.$$

Hence, $\operatorname{range}(\mathcal{R}_2(\mathbf{C}))$ is a subspace of $\operatorname{range}(\pi_S)$. Let W denote the orthogonal complement to $\operatorname{range}(\mathcal{R}_2(\mathbf{C})^*)$ in $\operatorname{range}(\pi_S)$,

$$\operatorname{range}(\pi_S) = \operatorname{range}(\mathcal{R}_2(\mathbf{C})^*) \oplus W \quad \text{or} \quad W = \ker(\mathcal{R}_2(\mathbf{C})^T) \cap \operatorname{range}(\pi_S). \tag{4.6}$$

It was shown in [14] that if $\mathbf{C}$ has full column rank, then

$$\dim \operatorname{range}(\mathcal{R}_2(\mathbf{C})^*) = R(R-1)/2, \quad \dim W = R, \tag{4.7}$$

and that

$$\begin{aligned}&[\mathbf{x}_1\ \dots\ \mathbf{x}_R] \text{ coincides with } \mathbf{C}^{-T} \text{ up to permutation and column scaling} \quad \Leftrightarrow \\ &\mathbf{x}_1\otimes\mathbf{x}_1, \dots, \mathbf{x}_R\otimes\mathbf{x}_R \quad \text{form a basis of } W.\end{aligned} \tag{4.8}$$

If one can find the subspace W (from $\mathcal{X}$), then one can reconstruct the columns of $\mathbf{C}$ up to permutation and column scaling by SD techniques. Indeed, if the vectors $\mathbf{m}_1 = \operatorname{Vec}(\mathbf{M}_1), \dots, \mathbf{m}_R = \operatorname{Vec}(\mathbf{M}_R)$ form a basis of W (yielding that $\mathbf{M}_1, \dots, \mathbf{M}_R$ are symmetric matrices), then by (4.8), there exists a nonsingular $R \times R$ matrix $\mathbf{L} = [\mathbf{l}_1\ \dots\ \mathbf{l}_R]$ such that

$$(\mathbf{C}^{-T} \odot \mathbf{C}^{-T})[\mathbf{l}_1\ \dots\ \mathbf{l}_R] = [\mathbf{m}_1\ \dots\ \mathbf{m}_R],$$

or, in matrix form,

$$\mathbf{C}^{-1}\operatorname{Diag}(\mathbf{l}_1)\mathbf{C}^{-T} = \mathbf{M}_1, \dots, \mathbf{C}^{-1}\operatorname{Diag}(\mathbf{l}_R)\mathbf{C}^{-T} = \mathbf{M}_R. \tag{4.9}$$

Thus, the matrices $\mathbf{M}_1, \dots, \mathbf{M}_R$ can be reduced simultaneously to diagonal form by congruence. It is well known that the solution $\mathbf{C}$ of (4.9) is unique (up to permutation and column scaling); see, for instance, [19, 24]. The matrices $\mathbf{A}$ and $\mathbf{B}$ can now be easily found from $\mathbf{X}_{(1)}\mathbf{C}^{-T} = \mathbf{A} \odot \mathbf{B}$.

The following algebraic identity was obtained in [14]:

$$(\mathcal{C}_2(\mathbf{A}) \odot \mathcal{C}_2(\mathbf{B}))\mathcal{R}_2(\mathbf{C})^T = \mathbf{R}_2(\mathcal{X}). \tag{4.10}$$

Since by assumption the matrix $\mathcal{C}_2(\mathbf{A}) \odot \mathcal{C}_2(\mathbf{B})$ has full column rank, it follows from (4.6) and (4.10) that

$$W = \ker(\mathcal{R}_2(\mathbf{C})^T) \cap \operatorname{range}(\pi_S) = \ker(\mathbf{R}_2(\mathcal{X})) \cap \operatorname{range}(\pi_S). \tag{4.11}$$

Hence, a basis $\mathbf{m}_1, \ldots, \mathbf{m}_R$ for W can be found directly from $\mathcal{X}$, which in turn means that $\mathbf{C}$ can be recovered via SD techniques (cf. (4.9)).

Algorithm 1 summarizes what we have discussed about the link between CPD and SD (for more details and proofs, see [9] and [14]).

The computational cost of Algorithm 1 is dominated by the construction of $\mathbf{R}_2(\mathcal{X})$ given by (4.5), the determination of a basis $\mathbf{m}_1, \ldots, \mathbf{m}_R$ for the subspace $\ker(\mathbf{R}_2(\mathcal{X})) \cap \operatorname{range}(\pi_S)$, and solving the SD problem (4.9). The following paragraphs discuss the complexity of the mentioned steps.

From (4.5) we conclude that the construction of $\mathbf{R}_2(\mathcal{X})$ requires $7C_I^2 C_J^2 C_{R+1}^2$ flops[1] (four multiplications and three additions/subtractions per distinct entry of $\mathbf{R}_2(\mathcal{X})$).

Once $\mathbf{R}_2(\mathcal{X})$ has been constructed, we can find a basis $\{\mathbf{m}_r\}$ for W. Since the rows of $\mathbf{R}_2(\mathcal{X})$ are vectorized symmetric matrices, we have that $\operatorname{range}\left(\mathbf{R}_2(\mathcal{X})^T\right) \subseteq \operatorname{range}(\pi_S)$. Consequently, a basis $\{\mathbf{m}_r\}$ for W can be obtained from a $C_I^2 C_J^2 \times C_{R+1}^2$ submatrix of $\mathbf{R}_2(\mathcal{X})$, which we denote by $\mathbf{P}$. More precisely, let $\mathbf{P} = \mathbf{R}_2(\mathcal{X})\mathbf{S}$, where $\mathbf{S}$ is an $R^2 \times C_{R+1}^2$ column selection matrix that selects the C_{R+1}^2 distinct columns of $\mathbf{R}_2(\mathcal{X})$ indexed by the elements in the set $\{(i-1)R + j \mid 1 \le i \le j \le R\}$.

We choose the R right singular vectors associated with the R smallest singular values of $\mathbf{P}$ as the basis $\{\mathbf{m}_r\}$ for W. The cost of finding this basis via an SVD is of order $6C_I^2 C_J^2 (C_{R+1}^2)^2$ when the SVD is implemented via the R-SVD method [16]. Note that the complexity of the R-SVD is proportional to $I^2 J^2 R^4$, making it the most expensive step. If the dimensions $\{I, J\}$ are large, then we may find the basis $\{\mathbf{m}_r\}$ for W via $\mathbf{P}^H\mathbf{P}$. (This squares the condition number.) Without taking the structure of $\mathbf{P}^H\mathbf{P}$ into account, the matrix product $\mathbf{P}^H\mathbf{P}$ requires $(2C_I^2 C_J^2 - 1)C_{R+1}^2$ flops, while, on the other the hand, the complexity of the determination of the basis $\{\mathbf{m}_r\}$ for W via the R-SVD method is now only proportional to $(C_{R+1}^2)^3$.

Note that for large dimensions $\{I, J\}$ the complexity of the construction of $\mathbf{R}_2(\mathcal{X})$ and $\mathbf{P}^H\mathbf{P}$ is proportional to $(IJR)^2$. By taking the structure of $\mathbf{P}^H\mathbf{P}$ into consideration, a procedure for determining a basis $\mathbf{m}_1, \ldots, \mathbf{m}_R$ for the subspace $\ker(\mathbf{R}_2(\mathcal{X})) \cap \operatorname{range}(\pi_S)$ with a complexity proportional to $\max\left(IJ^2, J^2R^2\right) R^2$ is described in the supplementary materials. This makes it more suitable for large dimensions $\{I, J\}$. We also note that the complexity of Algorithm 1 in the case of large dimensions $\{I, J\}$ can be reduced by an initial dimensionality reduction step, as will be briefly discussed in subsection 4.3.

The SD problem (4.9) can in the exact case be solved by means of a generalized Schur decomposition (GSD) of a pair $(\mathbf{M}_r, \mathbf{M}_s)$. According to [16], the complexity of the GSD implemented via the QZ step is of order $30R^2$. However, in the inexact case, there does not exist a simple algebraic method for solving the SD problem

[1] Complexity is measured here in terms of floating point operations (flops). Each multiplication, addition, and subtraction corresponds to a flop [38]. Furthermore, as in [38], no distinction between complex and real data is made.

(4.9). An iterative procedure that simultaneously tries to diagonalize the matrices $\{\mathbf{M}_r\}$ is applied in practice. A well-known method for the latter problem is the ALS method with a complexity of order $8R^4$ flops per iteration [30]; see also [30] for other optimization based methods.

Algorithm 1 SD procedure for a single CPD (noiseless case) assuming that condition (4.3) is satisfied.

Input: Tensor $\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$ such that (4.3) holds
Step 1: Estimate $\mathbf{C}$
Construct the matrix $\mathbf{R}_2(\mathcal{X})$ by (4.5)
Find a basis $\mathbf{m}_1, \ldots, \mathbf{m}_R$ of the subspace $\ker(\mathbf{R}_2(\mathcal{X})) \cap \text{range}(\pi_S)$
Denote $\mathbf{M}_1 = \text{Unvec}(\mathbf{m}_1), \ldots, \mathbf{M}_R = \text{Unvec}(\mathbf{m}_R)$
Solve simultaneous matrix diagonalization problem

$$\mathbf{C}^{-1}\ \text{Diag}(\mathbf{l}_1)\mathbf{C}^{-T} = \mathbf{M}_1,\ \ldots,\ \mathbf{C}^{-1}\ \text{Diag}(\mathbf{l}_R)\mathbf{C}^{-T} = \mathbf{M}_R$$

(the vectors $\mathbf{l}_1, \ldots, \mathbf{l}_R$ are a by-product).
Step 2: Estimate $\mathbf{A}$ and $\mathbf{B}$
Compute $\mathbf{Y} = \mathbf{X}_{(1)}\mathbf{C}^{-T}$
Find $\mathbf{a}_r$ and $\mathbf{b}_r$ from $\mathbf{y}_r = \mathbf{a}_r \otimes \mathbf{b}_r$, $r = 1, \ldots, R$,
Output: $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$

4.2.2. Coupled CPD. We now present a generalization of Algorithm 1 for the coupled PDs of the tensors $\mathcal{X}^{(n)} \in \mathbb{C}^{I_n \times J_n \times K}$, $n \in \{1, \ldots, N\}$, with matrix representation (2.5) and, repeated below,

$$\mathbf{X} = \mathbf{F}\mathbf{C}^T, \tag{4.12}$$

where $\mathbf{F} \in \mathbb{C}^{(\sum_{n=1}^{N} I_n J_n) \times R}$ now takes the form (2.6). Comparing (4.2) with (4.12) it is clear that the only difference between SD for single CPD and coupled CPD is that now $\mathbf{F}$ is subject to a blockwise Khatri–Rao structural constraint.

Define

$$\mathbf{E} = \begin{bmatrix} C_2\left(\mathbf{A}^{(1)}\right) \odot C_2\left(\mathbf{B}^{(1)}\right) \\ \vdots \\ C_2\left(\mathbf{A}^{(N)}\right) \odot C_2\left(\mathbf{B}^{(N)}\right) \end{bmatrix} \in \mathbb{C}^{\left(\sum_{n=1}^{N} C_{I_n}^2 C_{J_n}^2\right) \times C_R^2}, \tag{4.13}$$

and assume that

$$\begin{cases} \mathbf{C} \text{ has full column rank,} \\ \mathbf{E} \text{ has full column rank.} \end{cases} \tag{4.14}$$

(Compare to (4.3).) Then by [35, Corollary 4.11], the coupled rank of $\{\mathcal{X}^{(n)}\}$ is R, and the coupled CPD of $\{\mathcal{X}^{(n)}\}$ is unique. In other words, condition (4.14) guarantees that only scaled versions of $[(\mathbf{a}_r^{(1)} \otimes \mathbf{b}_r^{(1)})^T, \ldots, (\mathbf{a}_r^{(N)} \otimes \mathbf{b}_r^{(N)})^T]^T$, $r \in \{1, \ldots, R\}$, are contained in range$(\mathbf{X})$.

We will now extend the SD method to coupled CPDs for the case where condition (4.14) is satisfied. First we reduce the dimension of the third mode. By [35, Proposition 4.2], the matrix $\mathbf{F} = [(\mathbf{A}^{(1)} \odot \mathbf{B}^{(1)})^T\ \ldots\ (\mathbf{A}^{(N)} \odot \mathbf{B}^{(N)})^T]^T$ has full column rank.

Hence, $\mathbf{X} = \mathbf{F}\mathbf{C}^T$ is a rank-R matrix. If $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^H$ is the compact SVD of $\mathbf{X}$, then by (2.5)

$$\mathbf{U}\Sigma = \begin{bmatrix} \widetilde{\mathbf{X}}_{(1)}^{(1)} \\ \vdots \\ \widetilde{\mathbf{X}}_{(1)}^{(N)} \end{bmatrix} = \mathbf{F}\widetilde{\mathbf{C}}^T, \quad \widetilde{\mathbf{C}} := \mathbf{V}^T\mathbf{C} \in \mathbb{C}^{R\times R}, \tag{4.15}$$

where $\widetilde{\mathbf{X}}_{(1)}^{(n)} := \mathbf{X}_{(1)}^{(n)}\mathbf{V}$ and where $\widetilde{\mathcal{X}}^{(n)} := \sum_{r=1}^{R} \mathbf{a}_r^{(n)} \circ \mathbf{b}_r^{(n)} \circ \widetilde{\mathbf{c}}_r$ has matrix representation $\widetilde{\mathbf{X}}_{(1)}^{(n)}$. Applying (4.10) to tensors $\widetilde{\mathcal{X}}^{(n)}$ for $n \in \{1,\ldots,N\}$, we obtain

$$\mathbf{E}\cdot\mathcal{R}_2(\widetilde{\mathbf{C}})^T = \begin{bmatrix} \mathbf{R}_2(\widetilde{\mathcal{X}}^{(1)}) \\ \vdots \\ \mathbf{R}_2(\widetilde{\mathcal{X}}^{(N)}) \end{bmatrix} =: \mathbf{R}_2(\widetilde{\mathcal{X}}^{(1)},\ldots,\widetilde{\mathcal{X}}^{(N)}). \tag{4.16}$$

Since the matrix $\mathbf{E}$ has full column rank, it follows that

$$W = \ker(\mathcal{R}_2(\widetilde{\mathbf{C}})^T) \cap \operatorname{range}(\pi_S) = \ker(\mathbf{R}_2(\widetilde{\mathcal{X}}^{(1)},\ldots,\widetilde{\mathcal{X}}^{(N)})) \cap \operatorname{range}(\pi_S).$$

Thus, the matrix $\widetilde{\mathbf{C}}$ can be found from W using SD techniques as before.

Since the matrix $\mathbf{F}$ has full column rank, it follows that $\operatorname{range}(\mathbf{V}^*) = \operatorname{range}(\mathbf{X}^T) = \operatorname{range}(\mathbf{C}\mathbf{F}^T) = \operatorname{range}(\mathbf{C})$, and the matrix $\mathbf{C}$ can be recovered from $\widetilde{\mathbf{C}}$ as $\mathbf{C} = \mathbf{V}^*\widetilde{\mathbf{C}}$.

Finally, the factor matrices $\mathbf{A}^{(n)}$ and $\mathbf{B}^{(n)}$ can be easily obtained from the PD of $\mathcal{X}^{(n)}$ taking into account that the third factor matrix $\mathbf{C}$ is known. An outline of the SD procedure for computing a coupled CPD is presented as Algorithm 2.

Comparing Algorithm 1 for a single CPD with Algorithm 2 for a coupled CPD, we observe that the increased computational cost is dominated by the construction of $\mathbf{R}_2(\widetilde{\mathcal{X}}^{(1)},\ldots,\widetilde{\mathcal{X}}^{(N)})$ given by (4.16) and the determination of a basis $\mathbf{m}_1,\ldots,\mathbf{m}_R$ for the subspace $\ker(\mathbf{R}_2(\widetilde{\mathcal{X}}^{(1)},\ldots,\widetilde{\mathcal{X}}^{(N)}) \cap \operatorname{range}(\pi_S)$.

From (4.5) and (4.16) we conclude that the construction of the distinct elements of $\mathbf{R}_2(\widetilde{\mathcal{X}}^{(1)},\ldots,\widetilde{\mathcal{X}}^{(N)})$ requires $7(\sum_{n=1}^{N} C_{I_n}^2 C_{J_n}^2) C_{R+1}^2$ flops.

Since the rows of $\mathbf{R}_2(\widetilde{\mathcal{X}}^{(1)},\ldots,\widetilde{\mathcal{X}}^{(N)})$ are vectorized symmetric matrices, we have that $\operatorname{range}(\mathbf{R}_2(\widetilde{\mathcal{X}}^{(1)},\ldots,\widetilde{\mathcal{X}}^{(N)})^T) \subseteq \operatorname{range}(\pi_S)$. As in Algorithm 1, a basis $\{\mathbf{m}_r\}$ for W can be obtained from a $(\sum_{n=1}^{N} C_{I_n}^2 C_{J_n}^2) \times C_{R+1}^2$ submatrix of $\mathbf{R}_2(\widetilde{\mathcal{X}}^{(1)},\ldots,\widetilde{\mathcal{X}}^{(N)})$, which we denote by $\mathbf{P} = \mathbf{R}_2(\widetilde{\mathcal{X}}^{(1)},\ldots,\widetilde{\mathcal{X}}^{(N)})\mathbf{S}$, where $\mathbf{S}$ is an $R^2 \times C_{R+1}^2$ column selection matrix that selects C_{R+1}^2 distinct columns of $\mathbf{R}_2(\widetilde{\mathcal{X}}^{(1)},\ldots,\widetilde{\mathcal{X}}^{(N)})$. The R right singular vectors associated with the R smallest singular values of $\mathbf{P}$ are then chosen as the basis $\{\mathbf{m}_r\}$ for W. The cost of finding a basis of $\mathbf{P}$ via the R-SVD method is now in order of $6(\sum_{n=1}^{N} C_{I_n}^2 C_{J_n}^2)(C_{R+1}^2)^2$ flops. If the dimensions $\{I_n, J_n\}$ are large, then we may find the basis $\{\mathbf{m}_r\}$ for W via $\mathbf{P}^H\mathbf{P}$. Without taking the structure of $\mathbf{P}^H\mathbf{P}$ into account, the matrix product $\mathbf{P}^H\mathbf{P}$ requires $\sum_{n=1}^{N}(2C_{I_n}^2 C_{J_n}^2 - 1)C_{R+1}^2$ flops, while, on the other the hand, the complexity of the determination of the basis $\{\mathbf{m}_r\}$ for W via the R-SVD now is only proportional to $(C_{R+1}^2)^3$ flops.

For large dimensions $\{I_n, J_n\}$ the complexity of building $\mathbf{R}_2(\widetilde{\mathcal{X}}^{(1)},\ldots,\widetilde{\mathcal{X}}^{(N)})$ and $\mathbf{P}^H\mathbf{P}$ is proportional to $(\sum_{n=1}^{N} I_n^2 J_n^2)R^2$. By taking the structure of $\mathbf{P}^H\mathbf{P}$ into account, a procedure for finding a basis $\{\mathbf{m}_r\}$ for the subspace $\ker(\mathbf{R}_2(\widetilde{\mathcal{X}}^{(1)},\ldots,\widetilde{\mathcal{X}}^{(N)}) \cap \operatorname{range}(\pi_S)$ with a complexity proportional to $\max((\sum_{n=1}^{N} I_n J_n^2), (\sum_{n=1}^{N} J_n^2)R^2)R^2$ is described in the supplementary materials. This makes it more suitable for large dimensions $\{I_n, J_n\}$. As in Algorithm 1, the complexity of Algorithm 2 can in the case of large dimensions $\{I_n, J_n\}$ be reduced by an initial dimensionality reduction step, as will be briefly discussed in subsection 4.3.

Algorithm 2 SD procedure for coupled CPDs assuming that condition (4.14) is satisfied.

Input: Tensors $\mathcal{X}^{(n)} = \sum_{r=1}^{R} \mathbf{a}_r^{(n)} \circ \mathbf{b}_r^{(n)} \circ \mathbf{c}_r$, $n \in \{1, \ldots, N\}$.
Step 1: Estimate $\mathbf{C}$
Build $\mathbf{X}$ given by (2.5)
Compute SVD $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^H$
Build $\widetilde{\mathcal{X}}^{(1)}, \ldots, \widetilde{\mathcal{X}}^{(N)}$ by (4.15)
Build $\mathbf{R}_2(\widetilde{\mathcal{X}}^{(1)}), \ldots, \mathbf{R}_2(\widetilde{\mathcal{X}}^{(N)})$ by (4.5) and $\mathbf{R}_2(\widetilde{\mathcal{X}}^{(1)}, \ldots, \widetilde{\mathcal{X}}^{(N)})$ by (4.16)
Find a basis $\mathbf{m}_1, \ldots, \mathbf{m}_R$ of $\ker(\mathbf{R}_2(\widetilde{\mathcal{X}}^{(1)}, \ldots, \widetilde{\mathcal{X}}^{(N)})^*) \cap \operatorname{range}(\pi_S)$
Denote $\mathbf{M}_1 = \operatorname{Unvec}(\mathbf{m}_1), \ldots, \mathbf{M}_R = \operatorname{Unvec}(\mathbf{m}_R)$
Solve simultaneous matrix diagonalization problem

$$\widetilde{\mathbf{C}}^{-1} \operatorname{Diag}(\mathbf{l}_1)\widetilde{\mathbf{C}}^{-T} = \mathbf{M}_1, \ \ldots, \ \widetilde{\mathbf{C}}^{-1} \operatorname{Diag}(\mathbf{l}_R)\widetilde{\mathbf{C}}^{-T} = \mathbf{M}_R.$$

(the vectors $\mathbf{l}_1, \ldots, \mathbf{l}_R$ are a by-product)
Set $\mathbf{C} = \mathbf{V}^*\widetilde{\mathbf{C}}$
Step 2: Estimate $\{\mathbf{A}^{(n)}\}$ and $\{\mathbf{B}^{(n)}\}$
Compute

$$\mathbf{Y}_{(1)}^{(n)} = \mathbf{X}_{(1)}^{(n)} \left(\mathbf{C}^T\right)^\dagger, \quad n \in \{1, \ldots, N\}.$$

Solve rank-1 approximation problems

$$\min_{\mathbf{a}_r^{(n)}, \mathbf{b}_r^{(n)}} \left\| \mathbf{y}_{(1)}^{(n)} - \mathbf{a}_r^{(n)} \otimes \mathbf{b}_r^{(n)} \right\|_F^2, \quad r \in \{1, \ldots, R\}, \ n \in \{1, \ldots, N\}.$$

Output: $\{\mathbf{A}^{(n)}\}$, $\{\mathbf{B}^{(n)}\}$, and $\mathbf{C}$

4.2.3. Higher-order tensors. The SD procedure summarized as Algorithm 2 can also be extended to coupled CPDs of tensors of arbitrary order. More precisely, as explained in [35, subsection 4.5], the coupled CPD of

$$\mathbb{C}^{I_{1,n} \times \cdots \times I_{M_n,n} \times K} \ni \mathcal{X}^{(n)} = \sum_{r=1}^{R} \mathbf{a}_r^{(1,n)} \circ \cdots \circ \mathbf{a}_r^{(M_n,n)} \circ \mathbf{c}_r, \quad n \in \{1, \ldots, N\}, \tag{4.17}$$

can be reduced to a coupled CPD of a set of third-order tensors, which may be computed by means of Algorithm 2. An efficient implementation of the SD method for coupled CPDs of tensors of arbitrary order is also discussed in the supplementary materials. In short, the SD method addresses the coupled CPD problem (4.17) as a low-rank constrained structured matrix decomposition problem of the form

$$\mathbf{X} = \mathbf{F}\mathbf{C}^T, \tag{4.18}$$

where $\mathbf{F}$ is now subject to the blockwise higher-order Khatri–Rao constraint

$$\mathbf{F} = \begin{bmatrix} \mathbf{A}^{(1,1)} \odot \cdots \odot \mathbf{A}^{(M_1,1)} \\ \vdots \\ \mathbf{A}^{(1,N)} \odot \cdots \odot \mathbf{A}^{(M_N,1)} \end{bmatrix}.$$

Comparing (4.2) and (4.12) with (4.18) it is clear that the only difference between SD for single/coupled CPDs and single/coupled CPDs for tensors of arbitrary order is that $\mathbf{F}$ is now subject to a blockwise higher-order Khatri–Rao structural constraint.

4.2.4. Coupled matrix-tensor factorization. Due to its simplicity, the coupled matrix-tensor factorization (2.2) is frequently used; see [35] for references and a brief motivation. Note also that the SD procedure can be used to compute the coupled matrix-tensor decomposition (2.2) in the case where the common factor $\mathbf{C}$ has full column rank. Recall that the latter assumption is actually necessary in the uniqueness of $\mathbf{A}^{(2)}$ in the coupled matrix-tensor decomposition [35]. More precisely, let $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^H$ denote the compact SVD of

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}^{(1)}_{(1)} \\ \mathbf{X}^{(2)} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^{(1)} \odot \mathbf{B}^{(1)} \\ \mathbf{A}^{(2)} \end{bmatrix} \mathbf{C}^T.$$

Partition $\mathbf{U}$ as follows: $\mathbf{U} = [\mathbf{U}^{(1)T}, \mathbf{U}^{(2)T}]^T \in \mathbb{C}^{I_1 I_2 \times R}$ in which $\mathbf{U}^{(n)} \in \mathbb{C}^{I_n \times R}$. Then $\mathbf{A}^{(1)}$, $\mathbf{B}^{(1)}$, and $\mathbf{C}$ can be obtained from $\mathbf{U}^{(1)}\Sigma$ via the ordinary SD method [9]. Once $\mathbf{C}$ is known, $\mathbf{A}^{(2)}$ immediately follows from $\mathbf{A}^{(2)} = \mathbf{X}^{(2)}(\mathbf{C}^T)^\dagger$.

4.3. Remark on large tensors. Consider the tensors $\mathcal{X}^{(n)} \in \mathbb{C}^{I_{1,n}\times\cdots\times I_{M_n,n}\times K}$, $n \in \{1,\ldots,N\}$, for which the coupled CPD admits the matrix representation

$$\mathbb{C}^{\prod_{m=1}^{M_n} I_{m,n}\times K} \ni \mathbf{X}^{(n)} = \left(\mathbf{A}^{(1,n)} \odot \cdots \odot \mathbf{A}^{(M_n,n)}\right)\mathbf{C}^T, \quad n \in \{1,\ldots,N\}. \tag{4.19}$$

For large dimensions $\{I_{m,n}, K\}$ it is not feasible to directly apply the discussed SD methods. However, in data analysis applications the coupled rank R is usually very small compared to the large dimensions $\{I_{m,n}, K\}$. In such cases it is common to compress the data in a preprocessing step [29, 23]. Many different types of Tucker compression schemes for coupled tensor decompositions can be developed based on the existing literature, ranging from methods based on alternating subspace based projections (e.g., [3, 7, 8, 39]) and manifold optimization (e.g., [28, 21]) to randomized projections (e.g., [15, 18]). Briefly, a Tucker compression method looks for columnwise orthonormal projection matrices $\mathbf{U}^{(m,n)} \in \mathbb{C}^{I_{m,n}\times J_{m,n}}$ and $\mathbf{V} \in \mathbb{C}^{K\times L}$, where $J_{m,n} \leq I_{m,n}$ and $L \leq K$ denote the compression factors. This leads to the compressed tensors $\mathcal{Y}^{(n)} \in \mathbb{C}^{J_{1,n}\times\cdots\times J_{M_n,n}\times L}$, $n \in \{1,\ldots,N\}$, for which the coupled CPD admits the matrix representation

$$\begin{aligned} \mathbb{C}^{\prod_{m=1}^{M_n} J_{m,n}\times L} \ni \mathbf{Y}^{(n)} &= \left(\mathbf{U}^{(1,n)H} \otimes \cdots \otimes \mathbf{U}^{(M_n,n)H}\right)\mathbf{X}^{(n)}\mathbf{V}^* \\ &= \left(\mathbf{B}^{(1,n)} \odot \cdots \odot \mathbf{B}^{(M_n,n)}\right)\mathbf{D}^T, \quad n \in \{1,\ldots,N\}, \end{aligned} \tag{4.20}$$

in which $\mathbf{B}^{(m,n)} = \mathbf{U}^{(m,n)H}\mathbf{A}^{(m,n)}$ and $\mathbf{D} = \mathbf{V}^H\mathbf{C}$. Once the coupled CPD of the smaller tensors $\{\mathcal{Y}^{(n)}\}$ has been found, then the coupled CPD factor matrices of $\{\mathcal{X}^{(n)}\}$ follow immediately via $\mathbf{A}^{(m,n)} = \mathbf{U}^{(m,n)}\mathbf{B}^{(m,n)}$ and $\mathbf{C} = \mathbf{V}^*\mathbf{D}$.

5. Algorithms for computing the coupled BTD. In this section we adapt the methods described in the previous section to coupled BTD.

5.1. Coupled BTD via ordinary BTD. Consider the coupled BTD of the third-order tensors $\mathcal{X}^{(n)} \in \mathbb{C}^{I_n \times J_n \times K}$, $n \in \{1, \ldots, N\}$, in (3.2). Under the conditions stated in Theorem 5.2 in [35], the coupled BTD may be computed as follows. First we compute one of the individual multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ term decompositions

$$\mathbf{X}^{(p)}_{(1)} = \left(\mathbf{A}^{(p)} \odot \mathbf{B}^{(p)}\right) \mathbf{C}^{(p)T} \quad \text{for some } p \in \{1, \ldots, N\}.$$

For multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ term decomposition algorithms, see [25, 26, 30] and references therein. Next, the remaining multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ term decompositions may be computed as "multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ term decompositions with a known factor matrix" (i.e., matrix $\mathbf{C}^{(\text{red})}$):

$$\begin{aligned} \mathbf{X}^{(n)}_{(1)} &= \left(\mathbf{A}^{(n)} \odot \mathbf{B}^{(n)}\right) \mathbf{C}^{(n)T} \\ &= \left[\operatorname{Vec}\left(\mathbf{B}^{(1,n)}\mathbf{A}^{(1,n)T}\right), \ldots, \operatorname{Vec}\left(\mathbf{B}^{(R,n)}\mathbf{A}^{(R,n)T}\right)\right] \mathbf{C}^{(\text{red})T}, \end{aligned} \tag{5.1}$$

where $n \in \{1, \ldots, N\} \setminus p$. The results may afterward be refined by an optimization algorithm, such as the ALS algorithm discussed in the supplementary materials. The extension of the procedure to coupled M_nth-order tensors with $M_n \geq 4$ for one or more $n \in \{1, \ldots, N\}$ is straightforward. In the case where $\mathbf{C}^{(\text{red})}$ in (5.1) additionally has full column rank, the overall decomposition of $\mathbf{X}^{(n)}_{(1)}$ is obviously unique. Indeed, from $\mathbf{Y}^{(n)} = \mathbf{X}^{(n)}_{(1)}(\mathbf{C}^{(\text{red})T})^{\dagger}$, the factor matrices $\mathbf{A}^{(r,n)}$ and $\mathbf{B}^{(r,n)}$ follow from the best rank-$L_{r,n}$ approximation of $\|\operatorname{Unvec}(\mathbf{y}^{(n)}_r) - \mathbf{B}^{(r,n)}\mathbf{A}^{(r,n)T}\|^2_F$. In the rest of this subsection we will discuss a uniqueness condition and an algorithm for the case where $\mathbf{C}^{(\text{red})}$ does not have full column rank. Proposition 5.1 below presents a uniqueness condition for the case where $\mathbf{C}^{(\text{red})}$ in (5.1) is known but does not necessarily have full column rank.

PROPOSITION 5.1. *Consider the PD of $\mathcal{X} \in \mathbb{C}^{I \times J \times K}$ in* (3.1), *and assume that $\boldsymbol{C}^{(\text{red})}$ is known. Let S denote a subset of $\{1, \ldots, R\}$, and let $S^c = \{1, \ldots, R\} \setminus S$ denote the complementary set. Define $s := card(S)$ and $s^c := card(S^c)$. Stack the columns of $\boldsymbol{C}^{(\text{red})}$ with index in S in $\boldsymbol{C}^{(S)} \in \mathbb{C}^{K \times s}$, and stack the columns of $\boldsymbol{C}^{(\text{red})}$ with index in S^c in $\boldsymbol{C}^{(S^c)} \in \mathbb{C}^{K \times s^c}$. Let the elements of S be indexed by $\sigma(1), \ldots, \sigma(s)$, and let the elements of S^c be indexed by $\mu(1), \ldots, \mu(s^c)$. The corresponding partitions of $\boldsymbol{A}^{(n)}$ and $\boldsymbol{B}^{(n)}$ are then given by*

$$\begin{aligned} \boldsymbol{A}^{(S)} &= \left[\boldsymbol{A}^{(\sigma(1))}, \ldots, \boldsymbol{A}^{(\sigma(s))}\right] \in \mathbb{C}^{I \times \left(\sum_{p \in S} L_p\right)}, \\ \boldsymbol{A}^{(S^c)} &= \left[\boldsymbol{A}^{(\mu(1))}, \ldots, \boldsymbol{A}^{(\mu(s^c))}\right] \in \mathbb{C}^{I \times \left(\sum_{p \in S^c} L_p\right)}, \\ \boldsymbol{B}^{(S)} &= \left[\boldsymbol{B}^{(\sigma(1))}, \ldots, \boldsymbol{B}^{(\sigma(s))}\right] \in \mathbb{C}^{J \times \left(\sum_{p \in S} L_p\right)}, \\ \boldsymbol{B}^{(S^c)} &= \left[\boldsymbol{B}^{(\mu(1))}, \ldots, \boldsymbol{B}^{(\mu(s^c))}\right] \in \mathbb{C}^{J \times \left(\sum_{p \in S^c} L_p\right)}. \end{aligned}$$

If there exists a subset $S \subseteq \{1,\ldots,R\}$ with $0 \le s \le r_{\boldsymbol{C}^{(\mathrm{red})}}$ such that[2] [3]

$$\begin{cases} \boldsymbol{C}^{(S)} \text{ has full column rank (i.e., } r_{\boldsymbol{C}^{(S)}} = S), \\ \boldsymbol{B}^{(S^c)} \text{ has full column rank (i.e., } r_{\boldsymbol{B}^{(S^c)}} = \sum_{p \in S^c} L_p), \\ r([(\boldsymbol{P}_{\boldsymbol{C}^{(S)}} \widetilde{\boldsymbol{C}}^{(S^c)}) \odot \boldsymbol{A}^{(S^c)}, (\boldsymbol{P}_{\boldsymbol{C}^{(S)}} \boldsymbol{c}_{\mu(r)}^{(S^c)}) \otimes \boldsymbol{I}_I]) = I + \sum_{p \in S^c} L_p - L_r \quad \forall r \in S^c, \end{cases} \tag{5.2}$$

where $\widetilde{\boldsymbol{C}}^{(S^c)} = [\boldsymbol{1}_{L_{\mu(1)}}^T \otimes \boldsymbol{c}_{\mu(1)}^{(S^c)}, \ldots, \boldsymbol{1}_{L_{\mu(s^c)}}^T \otimes \boldsymbol{c}_{\mu(s^c)}^{(S^c)}]$, *then the decomposition of $\mathcal{X}$ in* (3.1) *is unique.*

Proof. The result is a variant of [34, Theorem 4.8] to the case where $\mathbf{C}$ contains collinear columns. A derivation is provided in the supplementary materials. ∎

The proof of Proposition 5.1 admits a constructive interpretation that is summarized as Algorithm 3. To avoid the construction of the tall matrix $\widetilde{\mathbf{D}}^{(S^c)} \odot \mathbf{A}^{(S^c)}$, we exploited the relation (see supplementary materials for details)

$$\begin{aligned} \mathbf{D}_{\mathbf{B}^{(S^c)}} &= (\widetilde{\mathbf{D}}^{(S^c)} \odot \mathbf{A}^{(S^c)})^H \mathbf{Y}_{(3)} \\ &= \begin{bmatrix} \mathbf{A}^{(\mu(1))*} \cdot \mathbf{f}^{(1,1)} & \cdots & \mathbf{A}^{(\mu(1))*} \cdot \mathbf{f}^{(1,J)} \\ \vdots & \ddots & \vdots \\ \mathbf{A}^{(\mu(s^c))*} \cdot \mathbf{f}^{(s^c,1)} & \cdots & \mathbf{A}^{(\mu(s^c))*} \cdot \mathbf{f}^{(s^c,J)} \end{bmatrix}, \end{aligned} \tag{5.3}$$

in which $\mathbf{f}^{(r,j)} = \mathbf{Y}^{(\cdot j \cdot)T} \mathbf{P}_{\mathbf{C}^{(S)}} \mathbf{c}_{\mu(r)}^{(S^c)*}$.

5.2. SD method for coupled BTD. In this section we explain how to extend the SD method discussed in subsection 4.2 to the decomposition in multilinear rank-$(L,L,1)$ terms and to the coupled decomposition in multilinear rank-$(L,L,1)$ terms. Note that we limit ourselves to the case $L_1 = \cdots = L_R = L$. The notation in section 3 simplifies to

$$\mathcal{X} = \sum_{r=1}^{R} (\mathbf{A}^{(r)} \mathbf{B}^{(r)T}) \circ \mathbf{c}^{(r)} = \sum_{r=1}^{LR} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r = \sum_{r=1}^{LR} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}^{(\lceil r/L \rceil)},$$

where

$$\mathbf{A} = [\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(R)}] \in \mathbb{C}^{I \times RL}, \qquad \mathbf{A}^{(r)} = [\mathbf{a}_{r(L-1)+1}, \ldots, \mathbf{a}_{rL}], \tag{5.4}$$

$$\mathbf{B} = [\mathbf{B}^{(1)}, \ldots, \mathbf{B}^{(R)}] \in \mathbb{C}^{J \times RL}, \qquad \mathbf{B}^{(r)} = [\mathbf{b}_{r(L-1)+1}, \ldots, \mathbf{b}_{rL}], \tag{5.5}$$

$$\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_{LR}] = [\mathbf{1}_L^T \otimes \mathbf{c}^{(1)}, \ldots, \mathbf{1}_L^T \otimes \mathbf{c}^{(R)}] \in \mathbb{C}^{K \times RL}, \tag{5.6}$$

$$\mathbf{C}^{(\mathrm{red})} := [\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(R)}] \in \mathbb{C}^{K \times R}, \tag{5.7}$$

and $r(\mathbf{A}^{(r)}) = r(\mathbf{B}^{(r)}) = L$, $\mathbf{c}^{(r)} \neq \mathbf{0}_K$.

Recall that the basic idea behind the SD procedure is to consider the tensor decomposition problem of $\mathcal{X}$ as a low-rank constrained matrix decomposition problem

[2] The last condition means that $\mathbf{M}_r = [(\mathbf{P}_{\mathbf{C}^{(S)}} \widetilde{\mathbf{C}}^{(S^c)}) \odot \mathbf{A}^{(S^c)}, (\mathbf{P}_{\mathbf{C}^{(S)}} \mathbf{c}_{\mu(r)}^{(S^c)}) \otimes \mathbf{I}_I]$ has an L_r-dimensional kernel for every $r \in S^c$, which is minimal since for every $p \in \{1,\ldots,L_r\}$ the vector $[\mathbf{n}_r^T, \mathbf{a}_p^{(\mu(r))T}]^T \in \ker(\mathbf{M}_r)$ for some $\mathbf{n}_r \in \mathbb{C}^{\mathrm{card}(S^c)}$.

[3] Note that the set S in Proposition 5.1 may be empty; i.e., $\mathrm{card}(S) = 0$ such that $S = \emptyset$. This corresponds to the case where $\mathbf{P}_{\mathbf{C}^{(S)}} = \mathbf{I}_K$.

Algorithm 3 Computation of BTD of $\mathcal{X}$ with known $\mathbf{C}^{(\text{red})}$ assuming that condition (5.2) is satisfied.

Input: $\mathbf{X}_{(1)} = \left[\text{Vec}\left(\mathbf{B}^{(1)}\mathbf{A}^{(1)T}\right), \ldots, \text{Vec}\left(\mathbf{B}^{(R)}\mathbf{A}^{(R)T}\right)\right] \mathbf{C}^{(\text{red})T}$ and $\mathbf{C}^{(\text{red})}$.

Choose sets $S \subseteq \{1, \ldots, R\}$ and $S^c = \{1, \ldots, R\} \setminus S$.

Build $\mathbf{C}^{(S)} = \left[\mathbf{c}^{(\sigma(1))}, \ldots, \mathbf{c}^{(\sigma(\text{card}(S)))}\right]$ and $\mathbf{C}^{(S^c)} = \left[\mathbf{c}^{(\mu(1))}, \ldots, \mathbf{c}^{(\mu(\text{card}(S^c)))}\right]$.

Find $\mathbf{Q}$ whose column vectors constitute an orthonormal basis for range $\left(\mathbf{C}^{(S)}\right)$.

Build $\widetilde{\mathbf{C}}^{(S^c)} = \left[\mathbf{1}^T_{L_{\mu(1)}} \otimes \mathbf{c}^{(S^c)}_{\mu(1)}, \ldots, \mathbf{1}^T_{L_{\mu(\text{card}(S^c))}} \otimes \mathbf{c}^{(S^c)}_{\mu(\text{card}(S^c))}\right]$.

Compute $\mathbf{P}_{\mathbf{C}^{(S)}} = \mathbf{I}_K - \mathbf{Q}\mathbf{Q}^H$.

Step 1. Find $\mathbf{A}^{(S^c)}$ and $\mathbf{B}^{(S^c)}$:

Compute $\mathbf{Y}_{(1)} = \mathbf{X}_{(1)}\mathbf{P}^T_{\mathbf{C}^{(S)}}$ and $\widetilde{\mathbf{D}}^{(S^c)} = \mathbf{P}_{\mathbf{C}^{(S)}}\widetilde{\mathbf{C}}^{(S^c)}$.

Reformatting: $\mathbf{Y}_{(3)} \leftarrow \mathbf{Y}_{(1)}$.

Compute SVD $\mathbf{Y}_{(3)} = \mathbf{U}\Sigma\mathbf{V}^H$.

Solve $\left[\mathbf{U}, -\left(\mathbf{P}_{\mathbf{C}^{(S)}}\mathbf{c}^{(S^c)}_{\mu(r)}\right) \otimes \mathbf{I}_I\right] \mathbf{X}_r = \mathbf{0}_{KI,L_r}, \quad r \in S^c$.

Set $\mathbf{A}^{(r)} = \mathbf{X}_r \left(\sum_{p \in S^c} L_p + 1 : \sum_{p \in S^c} L_p + J, 1 : L_r\right), \quad r \in S^c$.

Build $\mathbf{D}_{\mathbf{B}^{(S^c)}}$ in (5.3).

Compute $\mathbf{B}^{(S^c)T} = \left(\left(\widetilde{\mathbf{C}}^{(S^c)H}\widetilde{\mathbf{D}}^{(S^c)}\right) * \left(\mathbf{A}^{(S^c)H}\mathbf{A}^{(S^c)}\right)\right)^{-1} \mathbf{D}_{\mathbf{B}^{(S^c)}}$.

Step 2. Find $\mathbf{A}^{(S)}$ and $\mathbf{B}^{(S)}$:

Build $\mathbf{F}^{(S^c)} = \left[\text{Vec}\left(\mathbf{B}^{(\mu(1))}\mathbf{A}^{(\mu(1))T}\right), \ldots, \text{Vec}\left(\mathbf{B}^{(\mu(\text{card}(S^c)))}\mathbf{A}^{(\mu(\text{card}(S^c)))T}\right)\right]$.

Compute $\mathbf{Z}_{(1)} = \mathbf{Y}_{(1)} - \mathbf{F}^{(S^c)}\mathbf{C}^{(S^c)T}$.

Compute $\mathbf{H} = \mathbf{Z}_{(1)} \left(\mathbf{C}^{(S)T}\right)^{\dagger}$.

Solve $\min_{\mathbf{A}^{(\sigma(r))}, \mathbf{B}^{(\sigma(r))}} \left\| \mathbf{h}_{\sigma(r)} - \text{Vec}\left(\mathbf{B}^{(\sigma(r))}\mathbf{A}^{(\sigma(r))T}\right)\right\|_F^2, \quad r \in S$.

Output: $\mathbf{A}$ and $\mathbf{B}$.

(and vice versa). In the case of the multilinear rank-$(L, L, 1)$ term decomposition, the associated low-rank constrained matrix decomposition is

$$\mathbf{X}_{(1)} = \left[\text{Vec}\left(\mathbf{B}^{(1)}\mathbf{A}^{(1)T}\right) \ \cdots \ \text{Vec}\left(\mathbf{B}^{(R)}\mathbf{A}^{(R)T}\right)\right] \mathbf{C}^{(\text{red})T} = \mathbf{F}^{(\text{red})}\mathbf{C}^{(\text{red})T}, \tag{5.8}$$

where the columns of $\mathbf{F}^{(\text{red})}$ are subject to a low-rank constraint. More precisely, the columns of $\mathbf{F}^{(\text{red})}$ are assumed to be vectorized rank-L matrices. The other way around, we can interpret a rank constrained matrix decomposition problem of the form (5.8) as a multilinear rank-$(L, L, 1)$ term decomposition problem. This section explains how to adapt the SD method to low-rank constrained matrix decomposition problems of the form (5.8).

Our derivation is based on the following identity [14] (we assume that $K = R$):

$$\left[\mathcal{C}_m(\mathbf{A}) \odot \mathcal{C}_m(\mathbf{B})\right] \mathcal{R}_m(\mathbf{C})^T = \mathbf{R}_m(\mathcal{X}). \tag{5.9}$$

For $m = 1$ and $m = 2$, (5.9) coincides with (3.5) ($n = 1$) and with (4.10), respectively. All the matrices are well defined if $m \le \min(I, J, LR)$.

Subsection 5.2.1 further discusses (5.9). For simplicity we assume that $K = R$. First, we briefly recall the construction of $\mathcal{R}_m(\mathbf{C})$ and $\mathbf{R}_m(\mathcal{X})$ (see [14] for details). Then we present a version of identity (5.9) for $m = L + 1$ and for $\mathbf{C}$ given by (5.6). In subsections 5.2.2 and 5.2.3 we only assume that the matrix $\mathbf{C}^{(\text{red})}$ has full column rank ($K \geq R$) and derive algorithms for the actual computation of the decomposition in multilinear rank-$(L, L, 1)$ terms and the coupled decomposition in multilinear rank-$(L, L, 1)$ terms, respectively.

5.2.1. Auxiliary results related to identity (5.9).

Definition of matrix $\boldsymbol{\mathcal{R}_m}(\mathbf{C})$. Let P_m denote the set of all permutations of the set $\{1, \ldots, m\}$. The symmetrization π_S is a linear mapping that sends a rank-1 tensor $\mathbf{t}_1 \otimes \cdots \otimes \mathbf{t}_m$ to its symmetric part by

$$\pi_S(\mathbf{t}_1 \otimes \cdots \otimes \mathbf{t}_m) = \frac{1}{m!} \sum_{(l_1,\ldots,l_m) \in P_m} \mathbf{t}_{l_1} \otimes \cdots \otimes \mathbf{t}_{l_m}, \qquad \mathbf{t}_1, \ldots, \mathbf{t}_m \in \mathbb{C}^R. \tag{5.10}$$

Let $\mathbf{C} = \begin{bmatrix} \mathbf{c}_1 & \ldots & \mathbf{c}_{LR} \end{bmatrix} \in \mathbb{C}^{R \times LR}$. We define the R^m-by-C^m_{LR} matrix $\mathcal{R}_m(\mathbf{C})$ as the matrix consisting of the columns

$$m!\pi_S(\mathbf{c}_{i_1} \otimes \cdots \otimes \mathbf{c}_{i_m}), \quad 1 \leq i_1 < \cdots < i_m \leq LR. \tag{5.11}$$

We follow the convention that the column associated with the m-tuple $(i_1, \ldots, i_m)$ is preceding the column associated with the m-tuple $(j_1, \ldots, j_m)$ if and only if either $i_1 < j_1$ or there exists a $k \in \{1, \ldots, LR - 1\}$ such that $i_1 = j_1, \ldots, i_k = j_k$ and $i_{k+1} < j_{k+1}$. In what follows, such ordering of m-tuples is called lexicographical ordering. Thus,

$$\mathcal{R}_m(\mathbf{C}) = m! \begin{bmatrix} \pi_S(\mathbf{c}_1 \otimes \cdots \otimes \mathbf{c}_m) & \ldots & \pi_S(\mathbf{c}_{LR-m+1} \otimes \cdots \otimes \mathbf{c}_{LR}) \end{bmatrix}. \tag{5.12}$$

Construction of matrix $\boldsymbol{R_m}(\boldsymbol{\mathcal{X}})$. Let $\mathcal{X}$ be an $I \times J \times R$ tensor with frontal slices $\mathbf{X}(:, :, 1), \ldots, \mathbf{X}(:, :, R)$ and $2 \leq m \leq \min(I, J)$. By definition, the

$$\big((r_m - 1)R^{m-1} + (r_{m-1} - 1)R^{m-2} + \cdots + (r_2 - 1)R + r_1\big) \text{ th column}$$

of the $C^m_I C^m_J$-by-R^m matrix $\mathbf{R}_m(\mathcal{X})$ equals

$$\text{Vec}\left(\sum_{k=1}^{m} (-1)^{m-k} \sum_{1 \leq p_1 < p_2 < \cdots < p_k \leq m} \mathcal{C}_m\big(\mathbf{X}(:, :, r_{p_1}) + \cdots + \mathbf{X}(:, :, r_{p_k})\big)\right). \tag{5.13}$$

One can easily check that expression (5.13) is invariant under permutation of $r_1, \ldots, r_m$. Since the number of m-combinations with repetitions from the set $\{1, \ldots, R\}$ equals C^m_{R+m-1}, the matrix $\mathbf{R}_m(\mathcal{X})$ has exactly C^m_{R+m-1} distinct columns. Moreover, the rows of $\mathbf{R}_m(\mathcal{X})$ represent vectorized symmetric tensors.

For instance, if $m = R = 3$, then the $(1-1)3^2 + (2-1)3^1 + 3 = 6$th column of the $C^3_I C^3_J$-by-27 matrix $\mathbf{R}_3(\mathcal{X})$ equals

$$\begin{aligned} \text{Vec}\Big(&\mathcal{C}_3(\mathbf{X}(:, :, 1)) + \mathcal{C}_3(\mathbf{X}(:, :, 2)) + \mathcal{C}_3(\mathbf{X}(:, :, 3)) \\ &- \mathcal{C}_3(\mathbf{X}(:, :, 1) + \mathbf{X}(:, :, 2)) - \mathcal{C}_3(\mathbf{X}(:, :, 1) + \mathbf{X}(:, :, 3)) - \mathcal{C}_3(\mathbf{X}(:, :, 2) + \mathbf{X}(:, :, 3)) \\ &+ \mathcal{C}_3(\mathbf{X}(:, :, 1) + \mathbf{X}(:, :, 2) + \mathbf{X}(:, :, 3))\Big), \end{aligned}$$

and the columns of $\mathbf{R}_3(\mathcal{X})$ with indices 6, 8, 12, 15, 20, and 22 are the same.

A version of identity (5.9) for $m = L+1$ and for C given by (5.6). Let the matrix $\mathcal{R}_{L+1}^{(\text{dis})}(\mathbf{C})$ be formed by all distinct columns of $\mathcal{R}_{L+1}(\mathbf{C})$. By (5.6) and (5.11) these columns are

$$(L+1)!\pi_S(\mathbf{c}^{(j_1)}\otimes\cdots\otimes\mathbf{c}^{(j_{L+1})}),\quad 1\le j_1\le j_2\le\cdots\le j_{L+1}\le R,\quad j_1\ne j_{L+1}. \tag{5.14}$$

Indeed, the columns $\mathbf{c}^{(j_1)},\dots,\mathbf{c}^{(j_{L+1})}$ in (5.14) are obtained by choosing with repetition $L+1$ out of R columns $\mathbf{c}^{(1)},\dots,\mathbf{c}^{(R)}$ in such a way that at least one inequality in $j_1\le j_2\le\cdots\le j_{L+1}$ is strict. Hence, the columns of $\mathcal{R}_{L+1}^{(\text{dis})}(\mathbf{C})$ can be enumerated by means of $(L+1)$-tuples of the set

$$\Omega := \{(j_1,\dots,j_{L+1}):\ 1\le j_1\le j_2\le\cdots\le j_{L+1}\le R\}\setminus\{(j,\dots,j):\ 1\le j\le R\}. \tag{5.15}$$

Thus, Ω is obtained from the set of all $(L+1)$ combinations with repetitions from the set $\{1,\dots,R\}$ by removing the R combinations $(1,\dots,1),\dots,(R,\dots,R)$. Hence, $\text{card}(\Omega) = C_{R+(L+1)-1}^{L+1} - R = C_{R+L}^{L+1} - R$ and $\mathcal{R}_{L+1}^{(\text{dis})}(\mathbf{C})$ is an R^{L+1}-by-$(C_{R+L}^{L+1}-R)$ matrix. We will assume that the elements of Ω (and, hence, the columns of $\mathcal{R}_{L+1}^{(\text{dis})}(\mathbf{C})$) are ordered lexicographically.

It is clear that

$$\mathcal{R}_{L+1}(\mathbf{C}) = \mathcal{R}_{L+1}^{(\text{dis})}(\mathbf{C})\mathbf{P}^T, \tag{5.16}$$

where $\mathbf{P}^T$ is a $(C_{R+L}^{L+1}-R)$-by-C_{RL}^{L+1} matrix of which the entries are “0” or “1,” such that each column of $\mathbf{P}^T$ has exactly one entry “1.” Thus, the matrix $\mathbf{P}^T$ “expands” $\mathcal{R}_{L+1}^{(\text{dis})}(\mathbf{C})$ to $\mathcal{R}_{L+1}(\mathbf{C})$ by adding copies of columns. Formally, if we enumerate the rows and columns of $\mathbf{P}^T$ by means of the elements of Ω and $\Sigma := \{(i_1,\dots,i_{L+1}) : 1\le i_1<\cdots<i_{L+1}\le RL\}$, respectively, and assume that the elements of Σ are ordered lexicographically, then

$$\begin{gathered}\text{the entry of }\ \mathbf{P}^T\ \text{associated with}\ \ ((j_1,\dots,j_{L+1}),(i_1,\dots,i_{L+1}))\\ \text{is equal to}\ \begin{cases}1 & \text{if}\ \ (j_1,\dots,j_{L+1}) = (\lceil i_1/L\rceil,\dots,\lceil i_{L+1}/L\rceil),\\ 0 & \text{otherwise.}\end{cases}\end{gathered} \tag{5.17}$$

Thus, by (5.16), identity (5.9) for $m = L+1$ and for $\mathbf{C}$ given by (5.6) takes the form

$$[(\mathcal{C}_{L+1}(\mathbf{A})\odot\mathcal{C}_{L+1}(\mathbf{B}))\,\mathbf{P}]\,\mathcal{R}_{L+1}^{(\text{dis})}(\mathbf{C})^T = \mathbf{R}_{L+1}(\mathcal{X}). \tag{5.18}$$

In the remaining part of this subsection we prove an analogue of (4.6)–(4.9) for the decomposition in multilinear rank-$(L,L,1)$ terms.

Denote by Π_s a subspace of vectorized $\underbrace{R\times\cdots\times R}_{L+1}$ symmetric tensors:

$$\Pi_s = \text{span}\{\pi_S(\mathbf{t}_1\otimes\cdots\otimes\mathbf{t}_{L+1}):\ \mathbf{t}_1,\dots,\mathbf{t}_{L+1}\in\mathbb{C}^R\},$$

where π_s is defined in (5.10). The following result is well known.

LEMMA 5.2. *Let $\mathbf{t}_1,\dots,\mathbf{t}_R$ be a basis of $\mathbb{C}^R$. Then the vectors*

$$\pi_S(\mathbf{t}_{i_1}\otimes\cdots\otimes\mathbf{t}_{i_{L+1}}),\quad 1\le i_1\le\cdots\le i_{L+1}\le R,$$

form a basis of Π_s. In particular, $\dim\Pi_s = C_{R+L}^{L+1}$.

The following lemma makes the link between the subspace

$$W := \ker(\mathcal{R}_{L+1}^{(\mathrm{dis})}(\mathbf{C})^T) \cap \Pi_s \subset \mathbb{C}^{R^{L+1}}$$

and columns of the matrix $(\mathbf{C}^{(\mathrm{red})})^{-T}$.

LEMMA 5.3. *Let the matrices* $\mathbf{C}$ *and* $\mathbf{C}^{(\mathrm{red})}$ *be defined as in* (5.6)–(5.7) *and* $\mathbf{C}^{(\mathrm{red})}$ *be nonsingular. Then*

(i) $\dim(\ker(\mathcal{R}_{L+1}^{(\mathrm{dis})}(\mathbf{C})^T) \cap \Pi_s) = R$*;*

(ii) *a nonzero vector* $\mathbf{x} \in \mathbb{C}^R$ *is a solution of*

$$\mathcal{R}_{L+1}^{(\mathrm{dis})}(\mathbf{C})^T(\underbrace{\mathbf{x} \otimes \cdots \otimes \mathbf{x}}_{L+1}) = \mathbf{0} \tag{5.19}$$

if and only if $\mathbf{x}$ *is proportional to a column of* $(\mathbf{C}^{(\mathrm{red})})^{-T}$*;*

(iii) *the matrix* $\mathcal{R}_{L+1}^{(\mathrm{dis})}(\mathbf{C})$ *has full column rank; that is,* $r(\mathcal{R}_{L+1}^{(\mathrm{dis})}(\mathbf{C})) = C_{R+L}^{L+1} - R$.

Proof. By Lemma 5.2, the columns of $\mathcal{R}_{L+1}^{(\mathrm{dis})}(\mathbf{C})$ can be extended to a basis of Π_s by adding R vectors $\mathbf{c}^{(r)} \otimes \cdots \otimes \mathbf{c}^{(r)}$, $r = 1, \ldots, R$. This proves (i) and (iii). To prove (ii), we note that by (5.14), equality (5.19) holds for a nonzero vector $\mathbf{x} \in \mathbb{C}^R$ if and only if

$$(\mathbf{c}^{(j_1)T}\mathbf{x}) \cdots (\mathbf{c}^{(j_{L+1})T}\mathbf{x}) = 0 \quad \text{for all} \quad (j_1, \ldots, j_{L+1}) \in \Omega.$$

This is possible if and only if

$$(\mathbf{c}^{(j_1)T}\mathbf{x})(\mathbf{c}^{(j_2)T}\mathbf{x}) = 0 \quad \text{for all} \quad j_1, j_2 \text{ such that } 1 \le j_1 < j_2 \le R,$$

which in turn is possible if and only if $\mathbf{x}$ is proportional to a column of $(\mathbf{C}^{(\mathrm{red})})^{-T}$. ∎

5.2.2. SD method for the decomposition in multilinear rank-$(L, L, 1)$ terms. We consider decomposition (3.1) and assume that

$$\begin{cases} (\mathcal{C}_{L+1}(\mathbf{A}) \odot \mathcal{C}_{L+1}(\mathbf{B}))\,\mathbf{P} \text{ has full column rank,} \\ \mathbf{C}^{(\mathrm{red})} \text{ has full column rank.} \end{cases} \tag{5.20}$$

(Compare with (4.3).) First we show that if (5.20) holds, then $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ can be recovered from $\mathcal{X}$ using Algorithm 4. Then we show that the decomposition is unique (i.e., we show that there are no decompositions that cannot be found via Algorithm 4).

SD procedure. If the matrix $\mathbf{C}^{(\mathrm{red})}$ is not square, then we first reduce the dimension of the third mode. We use the fact that

$$\mathbf{F}^{(\mathrm{red})} = \left[\mathrm{Vec}\left(\mathbf{B}^{(1)}\mathbf{A}^{(1)T}\right) \;\cdots\; \mathrm{Vec}\left(\mathbf{B}^{(R)}\mathbf{A}^{(R)T}\right)\right]$$

has full column rank (see Lemma S.1.1 in supplementary materials).

Hence, $\mathbf{X}_{(1)} = \mathbf{F}^{(\mathrm{red})}\mathbf{C}^{(\mathrm{red})T}$ is a rank-R matrix. Let $\mathbf{X}_{(1)} = \mathbf{U}\Sigma\mathbf{V}^H$ be the compact SVD of $\mathbf{X}_{(1)}$, where $\mathbf{U} \in \mathbb{C}^{IJ\times R}$, $\mathbf{V} \in \mathbb{C}^{K\times R}$, and $\Sigma \in \mathbb{C}^{R\times R}$. Then

$$\overline{\mathbf{X}}_{(1)} := \mathbf{X}_{(1)}\mathbf{V} = \mathbf{U}\Sigma = \mathbf{F}^{(\mathrm{red})}\overline{\mathbf{C}}^T, \quad \overline{\mathbf{C}} := \mathbf{V}^T\mathbf{C}^{(\mathrm{red})} \in \mathbb{C}^{R\times R}, \tag{5.21}$$

where $\overline{\mathbf{X}}_{(1)}$ is the matrix unfolding of the $I \times J \times R$ tensor $\overline{\mathcal{X}} := \sum_{r=1}^{R} (\mathbf{A}^{(r)} \mathbf{B}^{(r)T}) \circ \overline{\mathbf{c}}_r$. Hence, all results of subsection 5.2.1 hold for $\mathcal{X}$ and $\mathbf{C}^{(\text{red})}$ replaced by $\overline{\mathcal{X}}$ and $\overline{\mathbf{C}}$, respectively. In particular, by (5.18),

$$W = \ker(\mathcal{R}_{L+1}^{(\text{dis})}(\overline{\mathbf{C}})^T) \cap \Pi_s = \ker(\mathbf{R}_{L+1}(\overline{\mathcal{X}})) \cap \Pi_s,$$

which means that the subspace W can be found directly from $\overline{\mathcal{X}}$.

Let us show how to reconstruct the columns of $\widetilde{\mathbf{C}}$ up to permutation and column scaling from the subspace W by means of SD techniques. By Lemma 5.3 (i), $\dim(W) = R$. Let the vectors $\mathbf{m}_1 = \text{Vec}(\mathcal{M}_1), \dots, \mathbf{m}_R = \text{Vec}(\mathcal{M}_R)$ form a basis of W (implying that $\mathcal{M}_1, \dots, \mathcal{M}_R$ are symmetric tensors). Then by Lemma 5.3 (ii), there exists a nonsingular $R \times R$ matrix $\mathbf{L} = [\mathbf{l}_1 \ \dots \ \mathbf{l}_R]$ such that

$$(\underbrace{\overline{\mathbf{C}}^{-T} \odot \cdots \odot \overline{\mathbf{C}}^{-T}}_{L+1})[\mathbf{l}_1 \ \dots \ \mathbf{l}_R] = [\mathbf{m}_1 \ \dots \ \mathbf{m}_R]$$

or, in tensor form,

$$\begin{cases} \mathcal{L}_1 \bullet_1 \overline{\mathbf{C}}^{-1} \bullet_2 \cdots \bullet_{L+1} \overline{\mathbf{C}}^{-1} = \mathcal{M}_1, \\ \qquad \vdots \\ \mathcal{L}_R \bullet_1 \overline{\mathbf{C}}^{-1} \bullet_2 \cdots \bullet_{L+1} \overline{\mathbf{C}}^{-1} = \mathcal{M}_R, \end{cases} \tag{5.22}$$

where $\mathcal{L}_r$ denotes a diagonal $(L+1)$th order tensor with the elements of the vector $\mathbf{l}_r$ on the main diagonal, and $\mathcal{L}_r \bullet_n \overline{\mathbf{C}}^{-1}$ denotes the n-mode product, defined as the summation over the nth index:

$$(\mathcal{L}_r \bullet_n \overline{\mathbf{C}}^{-1})_{m_1,\dots,m_{l-1},p,m_{l+1},\dots,m_{L+1}} = \sum_{s=1}^{R} (\mathcal{L}_r)_{m_1,\dots,m_{l-1},s,m_{l+1},\dots,m_{L+1}} (\overline{\mathbf{C}}^{-1})_{p,s}.$$

Thus, the tensors $\mathcal{M}_1, \dots, \mathcal{M}_R$ can be reduced simultaneously to diagonal form. It is well known that the solution $\overline{\mathbf{C}}$ of (5.22) is unique (up to permutation and column scaling). Indeed, the set of R equations in (5.22) can, for instance, be expressed similarly to (4.9) in terms of the matrix slices of $\mathcal{M}_1, \dots, \mathcal{M}_R$, after which $\overline{\mathbf{C}}^{-1}$ can be found by solving a simultaneous matrix diagonalization problem of a set of R^L matrices.

Since $\mathbf{F}^{(\text{red})}$ has full column rank, it follows that $\text{range}(\mathbf{V}^*) = \text{range}(\mathbf{X}_{(1)}^T) = \text{range}(\mathbf{C}^{(\text{red})} \mathbf{F}^{(\text{red})T}) = \text{range}(\mathbf{C}^{(\text{red})})$. Hence the matrix $\mathbf{C}^{(\text{red})}$ can be recovered from $\overline{\mathbf{C}}$ as $\mathbf{C}^{(\text{red})} = \mathbf{V}^* \overline{\mathbf{C}}$. The matrices $\mathbf{A}^{(r)}$ and $\mathbf{B}^{(r)}$ can now be easily found from $\mathbf{X}_{(1)} (\mathbf{C}^{(\text{red})})^\dagger = \mathbf{F}^{(\text{red})} = [\text{Vec}(\mathbf{B}^{(1)} \mathbf{A}^{(1)T}) \ \dots \ \text{Vec}(\mathbf{B}^{(R)} \mathbf{A}^{(R)T})]$.

Algorithm 4 summarizes what we have discussed about the link between the decomposition in multilinear rank-$(L, L, 1)$ terms and SD.

The complexity of Algorithm 4 is dominated by the cost of building $\mathbf{R}_{L+1}(\overline{\mathcal{X}})$ as in (5.13) with $m = L+1$ and $K = R$. From (5.13) we observe that the computation of $\mathbf{R}_{L+1}(\overline{\mathcal{X}})$ involves

1. $C_{R+(L+1)-1}^{L+1}$ matrices of the form $C_{L+1}\big(\overline{\mathbf{X}}(:,:,r_{p_1}) + \cdots + \overline{\mathbf{X}}(:,:,r_{p_{L+1}})\big)$,
2. $C_{R+(L+1)-2}^{(L+1)-1}$ matrices of the form $C_{L+1}\big(\overline{\mathbf{X}}(:,:,r_{p_1}) + \cdots + \overline{\mathbf{X}}(:,:,r_{p_L})\big)$,

$\vdots$

L+1. $R = C^{(L+1)-L}_{R+(L+1)-(L+1)}$ matrices of the form $C_{L+1}\big(\overline{\mathbf{X}}(:,:,r_{p_1})\big)$.

Recall that each entry of an $(L+1)$th-order compound matrix is equal to the determinant of an $(L+1)\times(L+1)$ matrix. The complexity of computing the determinant of an $(L+1)\times(L+1)$ matrix via the LU factorization is of order $(L+1)^3$ [16]. Since the $(C_I^{L+1}C_J^{L+1}) \times R^{L+1}$ matrix $\mathbf{R}_{L+1}(\overline{\mathcal{X}})$ has C^{L+1}_{R+L} distinct columns, we conclude that the complexity of Algorithm 4 is of order $(\sum_{m=0}^{L} C^{L+1-m}_{R+L-m})(C_I^{L+1}C_J^{L+1}C^{L+1}_{R+L}(L+1)^3)$.

Uniqueness. We prove that (5.20) implies the uniqueness of decomposition (3.1). We have already shown that if $\mathcal{X} = \sum_{r=1}^{R}(\mathbf{A}^{(r)}\mathbf{B}^{(r)T}) \circ \mathbf{c}^{(r)}$ with factor matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}^{(\mathrm{red})}$ that satisfy condition (5.20), then $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}^{(\mathrm{red})}$ can be recovered by Algorithm 4. This does not yet exclude the existence of alternative decompositions that cannot be found via Algorithm 4. To prove the overall uniqueness it is sufficient to show that any alternative decomposition

$$\mathcal{X} = \sum_{r=1}^{\widehat{R}}\sum_{l=1}^{L_r} \widehat{\mathbf{a}}_l^{(r)} \circ \widehat{\mathbf{b}}_l^{(r)} \circ \widehat{\mathbf{c}}^{(r)} = \sum_{r=1}^{\widehat{R}} \left(\widehat{\mathbf{A}}^{(r)}\widehat{\mathbf{B}}^{(r)T}\right) \circ \widehat{\mathbf{c}}^{(r)}$$

with $\widehat{R} \le R$ satisfies

$$\widehat{R} = R,$$

$$\begin{cases} \left(\mathcal{C}_{L+1}(\widehat{\mathbf{A}}) \odot \mathcal{C}_{L+1}(\widehat{\mathbf{B}})\right)\mathbf{P} \text{ has full column rank,} \\ \widehat{\mathbf{C}}^{(\mathrm{red})} := [\widehat{\mathbf{c}}^{(1)}, \dots, \widehat{\mathbf{c}}^{(R)}] \text{ has full column rank,} \end{cases} \tag{5.23}$$

which implies that in all cases Algorithm 4 can be used. Since $\mathbf{F}^{(\mathrm{red})}$ and $\mathbf{C}^{(\mathrm{red})}$ have full column rank and

$$\mathbf{X}_{(1)} = \mathbf{F}^{(\mathrm{red})}\mathbf{C}^{(\mathrm{red})T} = \left[\mathrm{Vec}\left(\widehat{\mathbf{B}}^{(1)}\widehat{\mathbf{A}}^{(1)T}\right) \;\cdots\; \mathrm{Vec}\left(\widehat{\mathbf{B}}^{(\widehat{R})}\widehat{\mathbf{A}}^{(\widehat{R})T}\right)\right]\widehat{\mathbf{C}}^{(\mathrm{red})T},$$

it follows that $\widehat{R} = R$ and that $\widehat{\mathbf{C}}^{(\mathrm{red})T}$ has full column rank. By (5.18),

$$\begin{aligned}[(\mathcal{C}_{L+1}(\mathbf{A}) \odot \mathcal{C}_{L+1}(\mathbf{B}))\,\mathbf{P}]\mathcal{R}^{(\mathrm{dis})}_{L+1}(\mathbf{C})^T &= \mathbf{R}_{L+1}(\mathcal{X}) \\ &= \left[\left(\mathcal{C}_{L+1}(\widehat{\mathbf{A}}) \odot \mathcal{C}_{L+1}(\widehat{\mathbf{B}})\right)\mathbf{P}\right]\mathcal{R}^{(\mathrm{dis})}_{L+1}(\widehat{\mathbf{C}})^T.\end{aligned} \tag{5.24}$$

From Lemma 5.2 (iii), (5.20), and (5.24) it follows that $(\mathcal{C}_{L+1}(\widehat{\mathbf{A}}) \odot \mathcal{C}_{L+1}(\widehat{\mathbf{B}}))\mathbf{P}$ has full column rank.

5.2.3. SD method for the coupled decomposition in multilinear rank-$(L, L, 1)$ terms. We consider coupled decomposition (3.2) subject to

$$\max_{1\le n\le N} r(\mathbf{A}^{(r,n)}\mathbf{B}^{(r,n)T}) = L \text{ and } \mathbf{c}^{(r)} \neq \mathbf{0}_K \quad \text{for all } r \in \{1,\dots,R\}. \tag{5.25}$$

Note that condition (5.25) does not prevent that $r(\mathbf{A}^{(r,n)}\mathbf{B}^{(r,n)T}) = L_{r,n} < L$ for some pairs (r,n). Since we are interested in the matrices $\mathbf{A}^{(r,n)}\mathbf{B}^{(r,n)T}$ and vectors $\mathbf{c}^{(r)}$, and since $\mathbf{A}^{(r,n)}\mathbf{B}^{(r,n)T} = [\mathbf{A}^{(r,n)}\ \mathbf{0}_{J_n,L-L_{r,n}}][\mathbf{B}^{(r,n)}\ \mathbf{0}_{I_n,L-L_{r,n}}]^T$, we can w.l.o.g. assume that $L_{r,n} = L$.

Let the matrices $\mathbf{C}^{(\mathrm{red})}$ and $\mathbf{P}$ be defined by (5.7) and (5.17), respectively, and let

$$\mathbf{E} := \begin{bmatrix} C_{L+1}\left(\mathbf{A}^{(1)}\right) \odot C_{L+1}\left(\mathbf{B}^{(1)}\right) \\ \vdots \\ C_{L+1}\left(\mathbf{A}^{(N)}\right) \odot C_{L+1}\left(\mathbf{B}^{(N)}\right) \end{bmatrix} \mathbf{P} \in \mathbb{C}^{\left(\sum_{n=1}^{N} C^{L+1}_{I_n}C^{L+1}_{J_n}\right)\times(C^{L+1}_{R+L}-R)},$$

Algorithm 4 SD procedure for the decomposition in multilinear rank-$(L, L, 1)$ terms assuming that condition (5.20) is satisfied.

Input: Tensor $\mathcal{X} = \sum_{r=1}^{LR} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}^{(\lceil r/L \rceil)}$.
  Step 1: Estimate $\mathbf{C}^{(\text{red})}$
    Compute SVD $\mathbf{X}_{(1)} = \mathbf{U}\Sigma\mathbf{V}^H$
    Stack $\mathbf{U}\Sigma$ in $\overline{\mathcal{X}}$ as in (5.21)
    Construct the matrix $\mathbf{R}_{L+1}(\overline{\mathcal{X}})$ by (5.13)
    Find a basis $\mathbf{m}_1, \ldots, \mathbf{m}_R$ of the subspace $\ker(\mathbf{R}_{L+1}(\overline{\mathcal{X}})) \cap \Pi_s$
      (Π_s denotes the subspace of vectorized symmetric tensors of order $L+1$)
    Denote $\mathcal{M}_1 = \text{Unvec}(\mathbf{m}_1), \ldots, \mathcal{M}_R = \text{Unvec}(\mathbf{m}_R)$
    Solve simultaneous tensor diagonalization problem (5.22)
      (the diagonal tensors $\mathcal{L}_1, \ldots, \mathcal{L}_R$ are a by-product)
    Set $\mathbf{C}^{(\text{red})} = \mathbf{V}^*\overline{\mathbf{C}}$
  Step 2: Estimate $\mathbf{A}$ and $\mathbf{B}$
    Compute $\mathbf{Y} = \mathbf{X}_{(1)} \left(\mathbf{C}^{(\text{red})}\right)^\dagger$
    Solve rank-L approximation problems

$$\min_{\mathbf{A}^{(r)},\mathbf{B}^{(r)}} \left\| \text{Unvec}(\mathbf{y}_r) - \mathbf{B}^{(r)}\mathbf{A}^{(r)T} \right\|_F^2, \quad r \in \{1, \ldots, R\}.$$

Output: $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}^{(\text{red})}$

where $\mathbf{A}^{(n)} = [\mathbf{A}^{(1,n)} \ \cdots \ \mathbf{A}^{(R,n)}] \in \mathbb{C}^{I_n \times RL}$ and $\mathbf{B}^{(n)} = [\mathbf{B}^{(1,n)} \ \cdots \ \mathbf{B}^{(R,n)}] \in \mathbb{C}^{J_n \times RL}$. We assume that

$$\begin{cases} \mathbf{E} \ \text{ has full column rank,} \\ \mathbf{C}^{(\text{red})} \ \text{ has full column rank.} \end{cases} \tag{5.26}$$

(Compare with (4.14) and (5.20).) In this subsection we first present a generalization of Algorithms 2 and 4 for the coupled decomposition (3.2). Then we prove that decomposition (3.2) is unique.

SD procedure. First we reduce the dimension of the third mode. We use the fact that the matrix $\mathbf{F}^{(\text{red})}$, given by (3.8), has full column rank (see Lemma S.1.1 in supplementary materials). Hence $\mathbf{X} = \mathbf{F}^{(\text{red})}\mathbf{C}^{(\text{red})T}$ is a rank-R matrix. Let $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^H$ be the compact SVD of $\mathbf{X}$, where $\mathbf{U} \in \mathbb{C}^{(\sum_{n=1}^N I_n J_n) \times R}$, $\mathbf{V} \in \mathbb{C}^{K \times R}$, and $\Sigma \in \mathbb{C}^{R \times R}$. Then by (3.7),

$$\mathbf{U}\Sigma := \begin{bmatrix} \overline{\mathbf{X}}_{(1)}^{(1)} \\ \vdots \\ \overline{\mathbf{X}}_{(1)}^{(N)} \end{bmatrix} = \mathbf{F}^{(\text{red})}\overline{\mathbf{C}}^T, \quad \overline{\mathbf{C}} := \mathbf{V}^T\mathbf{C}^{(\text{red})} \in \mathbb{C}^{R \times R}, \tag{5.27}$$

where $\overline{\mathbf{X}}_{(1)}^{(n)} = \mathbf{X}_{(1)}^{(n)}\mathbf{V}$. Then $\overline{\mathcal{X}}^{(n)} := \sum_{r=1}^{R}(\mathbf{A}^{(r,n)}\mathbf{B}^{(r,n)T}) \circ \overline{\mathbf{c}}_r$ has the matrix unfolding $\overline{\mathbf{X}}_{(1)}^{(n)}$. Applying (5.18) to tensors $\overline{\mathcal{X}}^{(n)}$ for $n = 1, \ldots, N$, we obtain

$$\mathbf{E} \cdot \mathcal{R}_{L+1}^{(\mathrm{dis})}(\overline{\mathbf{C}})^T = \begin{bmatrix} \mathbf{R}_{L+1}(\overline{\mathcal{X}}^{(1)}) \\ \vdots \\ \mathbf{R}_{L+1}(\overline{\mathcal{X}}^{(N)}) \end{bmatrix} =: \mathbf{R}_{L+1}(\overline{\mathcal{X}}^{(1)}, \ldots, \overline{\mathcal{X}}^{(N)}). \tag{5.28}$$

Since the matrix $\mathbf{E}$ has full column rank, it follows that

$$W = \ker(\mathcal{R}_{L+1}^{(\mathrm{dis})}(\overline{\mathbf{C}})^T) \cap \Pi_s = \ker(\mathbf{R}_{L+1}(\overline{\mathcal{X}}^{(1)}, \ldots, \overline{\mathcal{X}}^{(N)})) \cap \Pi_s.$$

Hence, a basis $\mathbf{m}_1, \ldots, \mathbf{m}_R$ for W can be found directly from $\mathcal{X}$. This in turn means that we proceed as in subsection 5.2.2: we find the matrix $\overline{\mathbf{C}}$ from W by means of SD techniques (cf. (5.22)), then set $\mathbf{C}^{(\mathrm{red})} = \mathbf{V}^*\overline{\mathbf{C}}$, and, finally, obtain the factor matrices $\mathbf{A}^{(r,n)}$ and $\mathbf{B}^{(r,n)}$ from $\mathbf{X}_{(1)}^{(n)}(\mathbf{C}^{(\mathrm{red})})^{\dagger}$. An outline of the SD procedure for computing coupled decomposition in multilinear rank-$(L, L, 1)$ terms is presented as Algorithm 5.

Algorithm 5 SD procedure for the coupled decomposition in multilinear rank-$(L, L, 1)$ terms assuming that condition (5.26) is satisfied.

Input: Tensors $\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(N)}$.
Step 1: Estimate $\mathbf{C}^{(\mathrm{red})}$
Build $\mathbf{X} = \left[\mathbf{X}_{(1)}^{(1)T}, \ldots, \mathbf{X}_{(1)}^{(N)T}\right]^T$
Compute SVD $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^H$
Build $\widetilde{\mathcal{X}}^{(1)}, \ldots, \widetilde{\mathcal{X}}^{(N)}$ by (5.27)
Build $\mathbf{R}_{L+1}(\overline{\mathcal{X}}^{(1)}, \ldots, \overline{\mathcal{X}}^{(N)})$ by (5.28)
Find a basis $\mathbf{m}_1, \ldots, \mathbf{m}_R$ of $\ker(\mathbf{R}_{L+1}(\overline{\mathcal{X}}^{(1)}, \ldots, \overline{\mathcal{X}}^{(N)})) \cap \Pi_s$
(Π_s denotes the subspace of vectorized symmetric tensors of order $L+1$)
Denote $\mathcal{M}_1 = \mathrm{Unvec}(\mathbf{m}_1), \ldots, \mathcal{M}_R = \mathrm{Unvec}(\mathbf{m}_R)$
Solve simultaneous tensor diagonalization problem (5.22)
(the diagonal tensors $\mathcal{L}_1, \ldots, \mathcal{L}_R$ are a by-product)
Set $\mathbf{C}^{(\mathrm{red})} = \mathbf{V}^*\overline{\mathbf{C}}$
Step 2: Estimate $\{\mathbf{A}^{(n)}\}$ and $\{\mathbf{B}^{(n)}\}$
Compute

$$\mathbf{Y}^{(n)} = \mathbf{X}_{(1)}^{(n)}\left(\mathbf{C}^{(\mathrm{red})T}\right)^{\dagger}, \quad n \in \{1, \ldots, N\}.$$

Solve rank-$L_{r,n}$ approximation problems

$$\min_{\mathbf{A}^{(r,n)}, \mathbf{B}^{(r,n)}} \left\| \mathrm{Unvec}\left(\mathbf{y}_r^{(n)}\right) - \mathbf{B}^{(r,n)}\mathbf{A}^{(r,n)T} \right\|_F^2, \quad r \in \{1, \ldots, R\}, n \in \{1, \ldots, N\}.$$

Output: $\{\mathbf{A}^{(n)}\}$, $\{\mathbf{B}^{(n)}\}$, and $\mathbf{C}^{(\mathrm{red})}$

The SD procedure summarized as Algorithm 5 can also be extended to coupled BTD of tensors of arbitrary order. More precisely, as explained in the supplementary material of [35], the problem of computing the coupled BTD of

$$\mathbb{C}^{I_{1,n}\times\cdots\times I_{M_n,n}\times K} \ni \mathcal{X}^{(n)} = \sum_{r=1}^{R} \mathbf{a}_r^{(1,n)} \circ \cdots \circ \mathbf{a}_r^{(M_n,n)} \circ \mathbf{c}_r^{(n)}, \quad n \in \{1,\ldots,N\},$$

in which $\mathbf{C}^{(n)} = [\mathbf{1}_{L_{r,n}}^T \otimes \mathbf{c}^{(1)}, \ldots, \mathbf{1}_{L_{R,n}}^T \otimes \mathbf{c}^{(R)}]$ and $L = \max_{1\leq n\leq N} L_{r,n}$ for all $r \in \{1,\ldots,R\}$, can be reduced to the computation of a coupled BTD of a set third-order tensors.

Uniqueness. One can assume that there exists another coupled decomposition

$$\mathcal{X}^{(n)} = \sum_{r=1}^{R}\sum_{l=1}^{L_{r,n}} \mathbf{a}_l^{(r,n)} \circ \mathbf{b}_l^{(r,n)} \circ \mathbf{c}^{(r)} = \sum_{r=1}^{R} \left(\mathbf{A}^{(r,n)}\mathbf{B}^{(r,n)T}\right) \circ \mathbf{c}^{(r)}$$

with $\widehat{R} \leq R$ and prove that $\widehat{R} = R$ and that (5.26) holds for $\mathbf{A}$, $\mathbf{B}$, $\mathbf{c}_r, \ldots$ replaced by $\widehat{\mathbf{A}}$, $\widehat{\mathbf{B}}$, $\widehat{\mathbf{c}}_r, \ldots$. Since the proof is very similar to that in subsection 5.2.2 (namely, (5.28) and (5.26) are used instead of (5.18) and (5.20), respectively), we omit it.

6. Numerical experiments. We compare the algorithms discussed in this paper, the ALS algorithm in the supplementary materials, and the iterative nonlinear least squares (NLS) solver `sdf_nls.m` in [31] on synthetic data in MATLAB. The tensors $\mathcal{X}^{(n)} \in \mathbb{C}^{I_n\times J_n\times K}$, $n \in \{1,\ldots,N\}$, are given by (2.1) or (3.2) depending on the experiment. The goal is to estimate the factor matrices from the observed tensors $\mathcal{T}^{(n)} = \mathcal{X}^{(n)} + \beta\mathcal{N}^{(n)}$, $n \in \{1,\ldots,N\}$, where $\mathcal{N}^{(n)}$ is an unstructured perturbation tensor and $\beta \in \mathbb{R}$ controls the noise level. The real and imaginary entries of all the involved factor matrices and perturbation tensors are randomly drawn from a Gaussian distribution with zero mean and unit variance.

The following signal-to-noise ratio (SNR) measure will be used:

$$\text{SNR [dB]} = 10\log\left(\sum_{n=1}^{N}\left\|\mathbf{X}_{(1)}^{(n)}\right\|_F^2 \Big/ \sum_{n=1}^{N}\left\|\beta\mathbf{N}_{(1)}^{(n)}\right\|_F^2\right).$$

The performance evaluation will be based on the distance between a factor matrix, say, $\mathbf{C}$, and its estimate, $\widehat{\mathbf{C}}$. The distance is measured according to the following criterion:

$$\mathrm{P}\left(\mathbf{C}\right) = \min_{\Pi\Lambda}\left\|\mathbf{C} - \widehat{\mathbf{C}}\Pi\Lambda\right\|_F / \left\|\mathbf{C}\right\|_F\ ,$$

where Π and Λ denote a permutation matrix and a diagonal matrix, respectively. The distance measure is numerically computed by means of the function `cpd_err.m` in [31]. To measure the time in seconds needed to execute the algorithms in MATLAB, the built-in functions `tic.m` and `toc.m` are used.

Let $f(\{\widehat{\mathbf{T}}_{(1)}^{(n,k)}\}) = \sum_{n=1}^{N}\|\mathbf{T}_{(1)}^{(n)} - \widehat{\mathbf{T}}_{(1)}^{(n,k)}\|_F$, where $\widehat{\mathbf{T}}_{(1)}^{(n,k)}$ denotes the estimate of tensor $\mathcal{T}^{(n)}$ at iteration k; then we decide that the ALS method has converged when $f(\{\widehat{\mathbf{T}}_{(1)}^{(n,k)}\}) - f(\{\widehat{\mathbf{T}}_{(1)}^{(n,k+1)}\}) < \epsilon_{\text{ALS}} = 1e-8$. Denote $g(\{\widehat{\mathbf{T}}_{(1)}^{(n,k)}\}) = \sum_{n=1}^{N}\|\mathbf{T}_{(1)}^{(n)} - \widehat{\mathbf{T}}_{(1)}^{(n,k)}\|_F^2$; then the stopping threshold $\left(g(\{\widehat{\mathbf{T}}_{(1)}^{(n,k)}\}) - g(\{\widehat{\mathbf{T}}_{(1)}^{(n,k+1)}\})\right)/g(\{\widehat{\mathbf{T}}_{(1)}^{(n,0)}\}) < \epsilon_{\text{NLS}}$ used for the NLS method `sdf_nls.m` in [31] will depend on the experiment under consideration. The conclusions do not critically depend on the chosen threshold values. We also terminate the ALS and NLS methods if the number of iterations exceeds 5000. Randomly initialized ALS or NLS methods will simply be referred to as ALS and NLS, respectively. We also consider the ALS method in which the best out of ten random initializations is retained, referred to as ALS-10.

In the case where the common factor matrix $\mathbf{C}$ (resp., $\mathbf{C}^{(\text{red})}$) has full column rank, the coupled CPD (resp., coupled BTD) will also be computed by means of the SD Algorithm 2 (resp., Algorithm 5) described in section 4.2 (resp., section 5.2.3). We numerically solve the simultaneous matrix diagonalization step in the SD procedure by means of a simultaneous GSD method [36]. In the case where the common factor matrix does not have full column rank, but one of the individual CPDs has a full column rank factor matrix, we compute the coupled CPD via the SD procedure for ordinary CPDs [9] followed by CPD problems with a known factor, as described in subsection 4.1. When the SD method is refined by at most 500 ALS iterations it will be referred to as SD-ALS.

6.1. Coupled CPD.

Case 1. In many signal processing applications the dimension K corresponds to the number of observations, such that $\mathbf{C}$ is often tall (e.g., [33]). The model parameters are $N = 2$, $I_1 = I_2 = J_1 = J_2 = 5$, $K = 50$, and $R = 10$. We set $\epsilon_{\text{NLS}} = 1e - 8$. The mean $P(\mathbf{C})$ and time values over 500 trials as a function of SNR can be seen in Figure 1. Above 15 dB SNR the algebraic SD method yields a good estimate of $\mathbf{C}$ at a low computational cost, and only below 15 dB SNR the algebraic SD method provides a poor estimate of $\mathbf{C}$. The reason for this behavior is that in the noise-free case SD yields the exact solution, while at low SNR values the noise-free assumption is violated. In the former case no fine-tuning is needed, while in the latter case a fine-tuning step may be necessary. However, by comparing the computational times of SD and SD-ALS we also remark that almost no fine-tuning is needed. For this particular case we observe that a reinitialization of ALS and NLS was not necessary. ALS has a lower complexity than NLS in this simple example. Overall, SD-ALS yields a good performance at a relatively low computational cost.

Case 2. The model parameters are $N = 2$, $I_1 = 3$, $J_1 = 4$, $I_2 = 4$, $J_2 = 5$, $K = 10$, and $R = 5$. To demonstrate that the coupled CPD framework may work even if none of the individual CPDs are unique, we set $\mathbf{b}_1^{(1)} = \mathbf{b}_2^{(1)}$, $\mathbf{a}_1^{(1)} = \mathbf{a}_2^{(1)}$, and $\mathbf{b}_3^{(2)} = \mathbf{b}_4^{(2)}$; that is, $r(\mathbf{A}^{(1)} \odot \mathbf{B}^{(1)}) < R$ and $k(\mathbf{B}^{(2)}) = 1$. We set $\epsilon_{\text{NLS}} = 1e - 8$. The mean $P(\mathbf{C})$ and time values over 500 trials as a function of SNR can be seen in Figure 2. In contrast to SD and SD-ALS, we notice that at high SNR the optimization-based ALS and NLS methods do not always find the solution with high accuracy. The main reason is that compared to Case 1 the problem addressed here is more difficult, which can make the iterative ALS and NLS methods more sensitive w.r.t. their initializations. For this reason a proper initialization of an optimization method is beneficial. We also observe that above 25 dB SNR the algebraic SD method performs well at a low computational cost, while below 25 dB SNR the algebraic SD method performs worse than the ALS and NLS methods. The main reason for the performance degradation of the SD method compared to Case 1 is that the problem is more difficult and the fact that K has gone from 50 to 10, implying a worse estimate of range$(\mathbf{X})$. However, we again notice that SD-ALS yields a good overall performance at a relatively low computational cost.

Case 3. The model parameters are $N = 2$, $I_1 = I_2 = 6$, $J_1 = J_2 = 4$, $K = 4$, and $R = 6$. Note that the common factor matrix does not have full column rank, but one of the individual CPDs has a full column rank factor matrix. The SD method now follows the "coupled CPD via ordinary CPD" approach described in subsection

4.1. More precisely, we first compute the coupled CPD of $\mathcal{T}^{(1)}$ via the SD procedure for ordinary CPDs [9] in which the first mode ($I_1 = R = 6$) is considered as the long mode. Thereafter, we compute the CPD of $\mathcal{T}^{(2)}$ with known factor $\mathbf{C}$ by the method in [34]. We set $\epsilon_{\text{NLS}} = 1e - 8$. The mean $P(\mathbf{C})$ and time values over 500 trials as a function of SNR can be seen in Figure 3. We observe that the SD method performs worse than the ALS and NLS methods. The reasons are that only one of the involved CPDs is used when computing the common factor, and additionally the dimension of I_1 is not large compared to the rank. However, at high SNR the SD-ALS method performs almost as well as the ALS and NLS methods but at a significantly lower computational cost.

Case 4. The model parameters are $N = 3$, $I_1 = I_2 = I_3 = J_1 = J_2 = J_3 = 20$, $K = 50$, and $R = 3$. To demonstrate that the coupled CPD framework may work in the presence of unshared components we set $\mathbf{a}_n^{(n)} = 0_{I_n}$ and $\mathbf{b}_n^{(n)} = 0_{J_n}$ for all $n \in \{1, 2, 3\}$. The maximal number of iterations for the ALS method is increased to 6000. We also relax the threshold of the NLS method to $\epsilon_{\text{NLS}} = 1e - 9$. In this experiment the iterative ALS and NLS methods turned out to be sensitive against outliers. For this reason we only plot the median $P(\mathbf{C})$ and time values over 500 trials as a function of SNR in Figure 4. We first observe that since the dimensions $\{I_n, J_n, K\}$ are large compared to the coupled rank R, all methods yield a good estimate of $\mathbf{C}$. More precisely, above 20 dB SNR, all methods perform the same, while below 20 dB SNR, SD performs slightly worse. By also taking the complexity of the methods into account, the SD-ALS seems to be the method of choice.

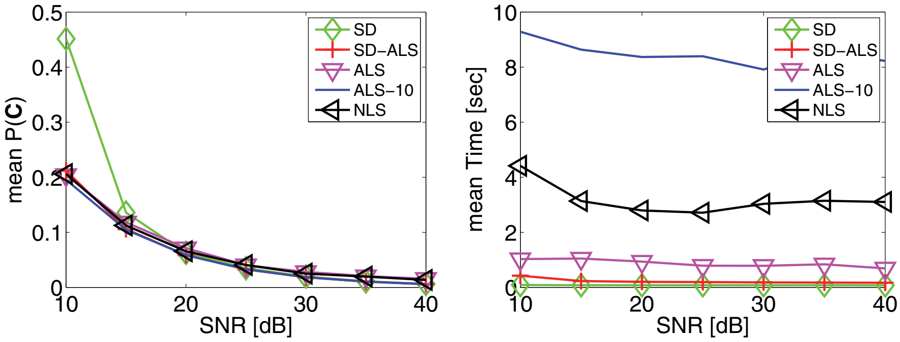
6.2. Coupled BTD.

Case 5. We now consider a coupled BTD problem with model parameters $N = 2$, $L_{1,1} = L_{1,2} = L_{2,1} = L_{2,2} = 2$, $I_1 = I_2 = 3$, $J_1 = J_2 = 4$, and $K = 50$. We set $\epsilon_{\text{NLS}} = 1e - 8$. The mean and median $P(\mathbf{C}^{(\text{red})})$ and time values over 500 trials as a function of SNR can be seen in Figure 5. We notice that all the methods find the solution except for ALS and NLS, which in some cases may require a proper initialization. (The difficult cases explain the difference between mean and median performance.) By exploiting both the coupled and BTD structure of the problem we note that SD does not require any fine-tuning, not even at low SNR. We also observe that SD and SD-ALS have very low cost.

Case 6. As our final example we consider a coupled BTD problem with model parameters $N = 2$, $L_{1,1} = 2$, $L_{1,2} = 3$, $L_{2,1} = 3$, $L_{2,2} = 2$, $I_1 = I_2 = J_1 = J_2 = 5$, and $K = 50$. We limit the comparison to the SD, ALS-10, and NLS methods. We also set threshold $\epsilon_{\text{NLS}} = 1e - 10$. In this experiment the iterative ALS and NLS methods turned out to be sensitive against outliers. For this reason we only plot the median $P(\mathbf{C})$ and time values over 500 trials as a function of SNR in Figure 6. We observe that NLS performs worse, illustrating the sensitivity of an iterative method w.r.t. initialization in the case of difficult problems. The SD method performed almost as well as the ALS-10 method but at a much lower computational cost.

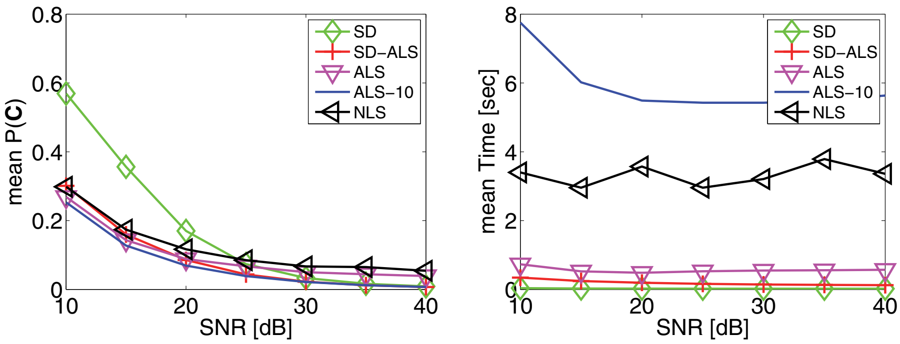
7. Conclusion. The coupled tensor decomposition framework is a natural and important extension of the framework of tensor decompositions. We demonstrated in [35] that improved uniqueness conditions can be obtained by taking the coupling between the involved decompositions into account. This observation suggests that it is also important to take the coupling into account in the actual computation.

So far, mainly iterative methods for coupled tensor decompositions have been presented that may be sensitive w.r.t. local minima, slow convergence, or improper



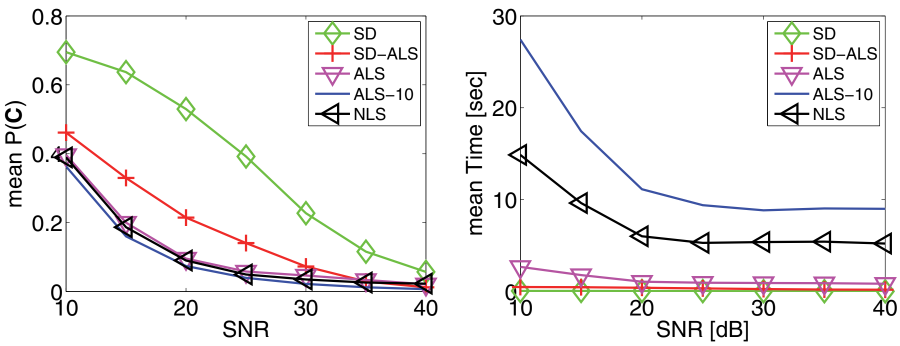
(a) Mean P (**C**). (b) Mean time.

FIG. 1. *Mean $P(\boldsymbol{C})$ and time values over* 500 *trials for varying SNR, case* 1.



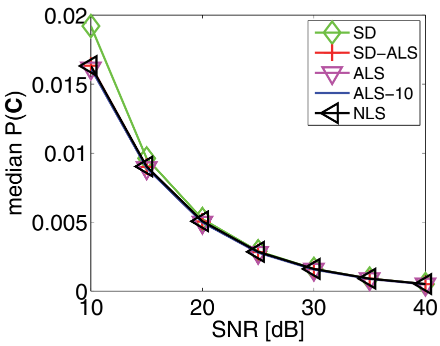
(a) Mean P (**C**). (b) Mean time.

FIG. 2. *Mean $P(\boldsymbol{C})$ and time values over* 500 *trials for varying SNR, case* 2.

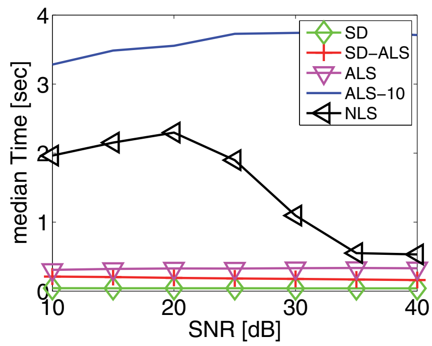


(a) Mean P (**C**). (b) Mean time.

FIG. 3. *Mean $P(\boldsymbol{C})$ and time values over* 500 *trials for varying SNR, case* 3.

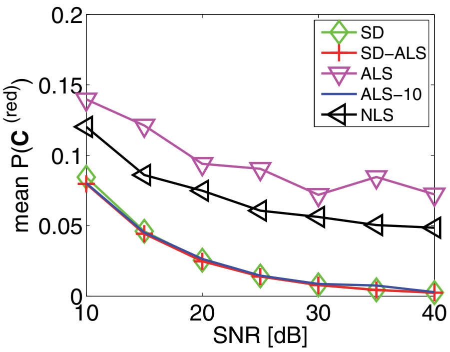


(a) Median P $(\mathbf{C})$.

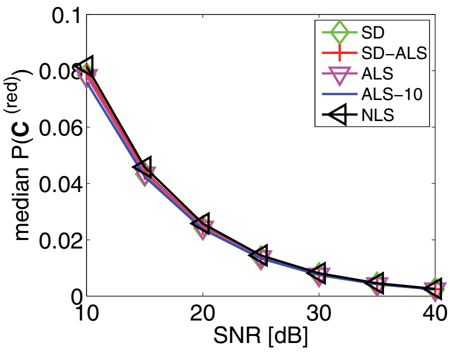


(b) Median time.

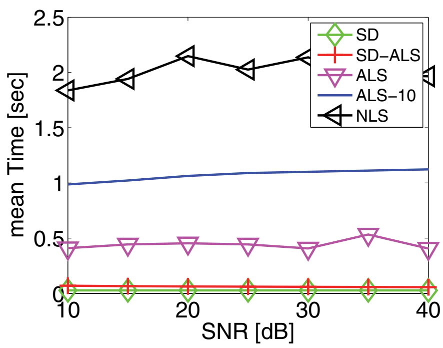
FIG. 4. *Median* $P(\boldsymbol{C})$ *and time values over* 500 *trials for varying SNR, case* 4.



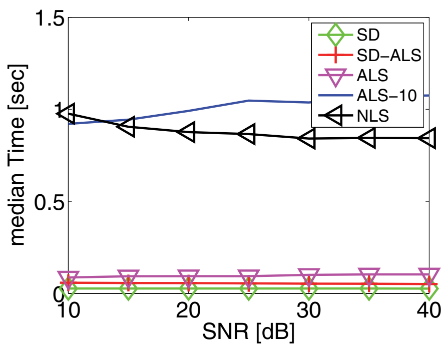
(a) Mean $\mathrm{P}(\mathbf{C}^{(\mathrm{red})})$.



(b) Median $\mathrm{P}(\mathbf{C}^{(\mathrm{red})})$.

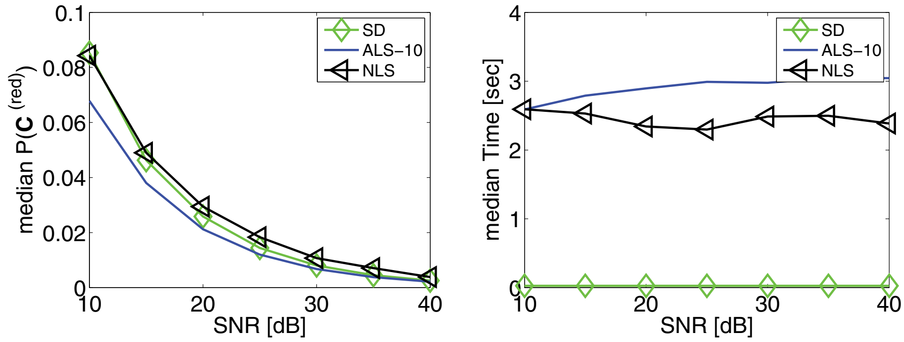


(c) Mean time.



(d) Median time.

FIG. 5. *Mean and median* $P(\boldsymbol{C}^{(red)})$ *and time values over* 500 *trials for varying SNR, case* 5.



(a) Median P (**C**).

(b) Median time.

FIG. 6. *Median* $P(\mathbf{C}^{(red)})$ *and time values over* 500 *trials for varying SNR, case* 6.

initialization. To alleviate this problem we first explained how to compute a coupled CPD via the algebraic SD framework [9] in cases where one of the individual CPDs is unique or where the coupled CPD is unique and the common factor has full column rank. By incorporating the results in [14] it is also possible to extend the SD method to cases where none of the individual CPDs are unique and the common factor does not have full column rank. These methods are guaranteed to find the exact solution in the noise-free case and are expected to find a good approximation at high SNR.

In the second part of the paper we extended the SD framework to tensor decompositions into multilinear rank-$(L, L, 1)$ terms and coupled decompositions into multilinear rank-$(L, L, 1)$ terms for the cases where the column reduced common factor matrix has full column rank. We also presented a uniqueness condition and an algorithm for the decomposition of multilinear rank-$(L_r, L_r, 1)$ terms in the case where the common factor matrix is known.

Numerical experiments demonstrated that in the case of high SNR values, the linear algebra based methods have a good performance at a relatively low computational cost. The numerical experiments also revealed that in the case of low SNR values, linear algebra based methods such as SD can provide a good initialization for an optimization method at a relatively low computational cost. Numerical experiment Case 3 confirmed that in the presence of noise it is better to fully exploit the coupled CPD/BTD structure in the actual computation.

REFERENCES

[1] E. ACAR, T. G. KOLDA, AND D. M. DUNLAVY, *All-at-once optimization for coupled matrix and tensor factorizations*, in the 9th KDD Workshop on Mining and Learning with Graphs, San Diego, CA, 2011.

[2] H. BECKER, P. COMON, AND L. ALBERA, *Tensor-based preprocessing of combined EEG/MEG data*, in Proceedings of Eusipco 2012, Bucharest, Romania, 2012.

[3] R. BRO AND C. A. ANDERSON, *Improving the speed of multiway algorithms. Part* II: *Compression*, Chemometrics Intelligent Laboratory Syst., 42 (1998), pp. 105–113.

[4] A. CICHOCKI, D. MANDIC, C. CAIFA, A.-H. PHAN, G. ZHOU, AND L. DE LATHAUWER, *Tensor decompositions for signal processing applications: From two-way to multiway component analysis*, IEEE Signal Process. Mag., 32 (2015), pp. 145–163.

[5] P. COMON, X. LUCIANI, AND A. L. F. DE ALMEIDA, *Tensor decompositions, alternating least squares and other tales*, J. Chemometrics, 23 (2009), pp. 393–405.

[6] P. Comon, *Tensors: A brief survey*, IEEE Signal Process. Mag., 31 (2014), pp. 44–53.
[7] L. De Lathauwer, B. De Moor, and J. Vandewalle, *On the best rank-1 and rank-$(R_1, R_2, \dots, R_N)$ approximation of higher-order tensors*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1324–1342.
[8] L. De Lathauwer and J. Vandewalle, *Dimensionality reduction in higher-order signal processing and rank-$(R_1, R_2, \dots, R_N)$ reduction in multilinear algebra*, Linear Algebra Appl., 391 (2004), pp. 31–55.
[9] L. De Lathauwer, *A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 642–666.
[10] L. De Lathauwer, *Decomposition of a higher-order tensor in block terms—part* II: *Definitions and uniqueness*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1033–1066.
[11] L. De Lathauwer, *Blind separation of exponential polynomials and the decomposition of a tensor in rank-$(L_r, L_r, 1)$ terms*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 1451–1474.
[12] L. De Lathauwer, *Block component analysis, a new concept for blind source separation*, in Proceedings of LVA/ICA 2012, Tel-Aviv, Israel, 2012.
[13] I. Domanov and L. De Lathauwer, *On the uniqueness of the canonical polyadic decomposition of third-order tensors—part* I: *Basic results and uniqueness of one factor matrix*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 855–875.
[14] I. Domanov and L. De Lathauwer, *Canonical polyadic decomposition of third-order tensors: Reduction to generalized eigenvalue decomposition*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 636–660.
[15] P. Drineas and M. W. Mahoney, *A randomized algorithm for a tensor-based generalization of the singular value decomposition*, Linear Algebra Appl., 420 (2007), pp. 553–571.
[16] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed., John Hopkins University Press, Baltimore, MD, 2013.
[17] X.-F. Gong, Y.-N. Hao, and Q.-H. Lin, *Joint canonical polyadic decomposition of two tensors with one shared loading matrix*, in Proceedings of the 23rd IEEE International Workshop on Machine Learning for Signal Processing (MLSP), Southampton, UK, 2013, pp. 1–6.
[18] N. Halko, P. G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288.
[19] R. A. Harshman, *Foundations of the PARAFAC procedure: Models and conditions for an explanatory multimodal factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.
[20] R. A. Horn and C. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
[21] M. Ishteva, P.-A. Absil, S. Van Huffel, and L. De Lathauwer, *Best low multilinear rank approximation of higher-order tensors, based on the Riemannian trust-region scheme*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 115–135.
[22] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), pp. 455–500.
[23] P. M. Kroonenberg, *Applied Multiway Data Analysis*, Wiley, Hoboken, NJ, 2008.
[24] S. E. Leurgans, R. T. Ross, and R. B. Abel, *A decomposition for three-way arrays*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 1064–1083.
[25] D. Nion and L. De Lathauwer, *A tensor-based blind DS-CDMA receiver using simultaneous matrix diagonalization*, in Proceedings of the 8th IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Helsinki, Finland, 2007, pp. 1–5.
[26] D. Nion and L. De Lathauwer, *A link between the decomposition of a third-order tensor in rank-$(L, L, 1)$ terms and joint block diagonalization*, in Proceedings of CAMSAP 2009, Aruba, Dutch Antilles, 2009.
[27] D. Nion and L. De Lathauwer, *A Study of the Decomposition of a Third-Order Tensor in Rank-$(L, L, 1)$ Terms*, Tech. Rep. 11-239, ESAT-STADIUS, KU Leuven, Leuven, Belgium, 2011.
[28] B. Savas and L.-H. Lim, *Quasi-Newton methods on Grassmannians and multilinear approximations of tensors*, SIAM J. Sci. Comput., 32 (2010), pp. 3352–3393.
[29] A. Smilde, R. Bro, and P. Geladi, *Multi-Way Analysis: Applications in the Chemical Sciences*, Wiley, New York, 2004.
[30] L. Sorber, M. Van Barel, and L. De Lathauwer, *Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-$(L_r, L_r, 1)$ terms, and a new generalization*, SIAM J. Optim., 23 (2013), pp. 695–720.
[31] L. Sorber, M. Van Barel, and L. De Lathauwer, *Tensorlab v.* 2.0, http://www.tensorlab.net, 2014.

[32] L. Sorber, M. Van Barel, and L. De Lathauwer, *Structured Data Fusion*, Tech. Rep. 13-177, ESAT-STADIUS, KU Leuven, Leuven, Belgium, 2013.
[33] M. Sørensen and L. De Lathauwer, *Coupled tensor decompositions for applications in array signal processing*, in Proceedings of CAMSAP 2013, Saint Martin, Dutch Antilles, 2013.
[34] M. Sørensen and L. De Lathauwer, *New Uniqueness Conditions for the Canonical Polyadic Decomposition of Third-Order Tensors*, Internal Report 13-05, ESAT-STADIUS, KU Leuven, Leuven, Belgium.
[35] M. Sørensen and L. De Lathauwer, *Coupled canonical polyadic decompositions and (coupled) decompositions in multilinear rank-*$(L_{r,n}, L_{r,n}, 1)$ *terms—part* I: *Uniqueness*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 496–522.
[36] M. Sørensen and L. De Lathauwer, *New simultaneous generalized Schur decomposition methods for computing the canonical polyadic decomposition*, in Proceedings of Asilomar 2010, Pacific Grove, CA, 2010.
[37] G. Tomasi and R. Bro, *A comparison of algorithms for fitting the Parafac model*, Comput. Statist. Data Anal., 50 (2006), pp. 1700–1734.
[38] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
[39] L. R. Tucker, *Some mathematical notes on the three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.

COUPLED CANONICAL POLYADIC DECOMPOSITIONS AND (COUPLED) DECOMPOSITIONS IN MULTILINEAR RANK-$(L_{r,n}, L_{r,n}, 1)$ TERMS — PART II: ALGORITHMS SUPPLEMENTARY MATERIAL

MIKAEL SØRENSEN *†, IGNAT DOMANOV *†, AND LIEVEN DE LATHAUWER *†

S.1. An indirect SD method for coupled CPD suitable for large dimensions $\{I_n, J_n\}$. Consider the R-term coupled PD of the tensors $\mathcal{X}^{(n)} \in \mathbb{C}^{I_n \times J_n \times K}$, $n \in \{1, \ldots, N\}$ given by

$$\mathcal{X}^{(n)} = \sum_{r=1}^{R} \mathbf{a}_r^{(n)} \circ \mathbf{b}_r^{(n)} \circ \mathbf{c}_r \,, \quad n \in \{1, \ldots, N\}, \tag{S.1.1}$$

with factor matrices $\mathbf{A}^{(n)} = \left[\mathbf{a}_1^{(n)}, \ldots, \mathbf{a}_R^{(n)}\right] \in \mathbb{C}^{I_n \times R}$, $\mathbf{B}^{(n)} = \left[\mathbf{b}_1^{(n)}, \ldots, \mathbf{b}_R^{(n)}\right] \in \mathbb{C}^{J_n \times R}$ and $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_R] \in \mathbb{C}^{K \times R}$. The coupled PD of $\{\mathcal{X}^{(n)}\}$ given by (S.1.1) has the following matrix representation:

$$\mathbf{X} = \mathbf{F}\mathbf{C}^T \in \mathbb{C}^{(\sum_{n=1}^{N} I_n J_n) \times K}, \tag{S.1.2}$$

where

$$\mathbf{F} = \left[\left(\mathbf{A}^{(1)} \odot \mathbf{B}^{(1)}\right)^T, \ldots, \left(\mathbf{A}^{(N)} \odot \mathbf{B}^{(N)}\right)^T \right]^T. \tag{S.1.3}$$

We concluded in [4, Subsection 4.2.2] that the complexity of the SD method for coupled CPD is proportional to $(\sum_{n=1}^{N} I_n^2 J_n^2) R^2$ flops. In this section we will discuss an indirect implementation of the SD procedure for coupled CPD with complexity proportional to $\max\left(\left(\sum_{n=1}^{N} I_n J_n^2\right), \left(\sum_{n=1}^{N} J_n^2\right) R^2\right) R^2$ flops. This makes the indirect SD method more suitable for the case of large dimensions $\{I_n, J_n\}$, in particular if $\max\left(\left(\sum_{n=1}^{N} I_n J_n^2\right), \left(\sum_{n=1}^{N} J_n^2\right) R^2\right)$ is significantly smaller than $(\sum_{n=1}^{N} I_n^2 J_n^2)$.

Subsection S.1.1 reviews the the SD procedure [1] and its extensions to coupled CPD [5]. Based on the reviewed results we will in Subsection S.1.2 present an indirect but more efficient version of the SD procedure for the case of large dimensions $\{I_n, J_n\}$.

S.1.1. Direct SD. Let the columns of $\mathbf{U} \in \mathbb{C}^{(\sum_{n=1}^{N} I_n J_n) \times R}$ constitute a basis for range$\,(\mathbf{X})$ obtained via for instance an SVD of $\mathbf{X}$. Consider the bilinear mappings $\Phi^{(n)} : \mathbb{C}^{I_n \times J_n} \times \mathbb{C}^{I_n \times J_n} \to \mathbb{C}^{I_n^2 J_n^2}$ defined by

$$\left(\Phi^{(n)}(\mathbf{X}, \mathbf{Y})\right)_{(i-1)I_n J_n^2 + (j-1)J_n^2 + (k-1)J_n + l} = x_{ik} y_{jl} + y_{ik} x_{jl} - x_{il} y_{jk} - y_{il} x_{jk}.$$

*Group Science, Engineering and Technology, KU Leuven - Kulak, E. Sabbelaan 53, 8500 Kortrijk, Belgium, KU Leuven - E.E. Dept. (ESAT) - STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, and iMinds Medical IT Department, Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium. {Mikael.Sorensen, Ignat.Domanov, Lieven.DeLathauwer}@kuleuven-kulak.be.

†Research supported by: (1) Research Council KU Leuven: GOA/10/09 MaNet, CoE PFV/10/002 (OPTEC), (2) F.W.O.: project G.0830.14N, G.0881.14N, (3) the Belgian Federal Science Policy Office: IUAP P7 (DYSCO II, Dynamical systems, control and optimization, 2012-2017), (4) EU: The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Advanced Grant: BIOTENSORS (no. 339804). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information.

Note that the number of multiplications and additions required for the construction of $\Phi^{(n)}(\mathbf{X},\mathbf{Y})$ is $4I_n^2J_n^2$.

Partition $\mathbf{U}$ as follows

$$\mathbf{U}=\left[\mathbf{U}^{(1)T},\ldots,\mathbf{U}^{(N)T}\right]^T,\quad \mathbf{U}^{(n)}\in\mathbb{C}^{I_nJ_n\times R}. \tag{S.1.4}$$

Define $\mathbf{U}^{(n,r)}=\text{Unvec}\left(\mathbf{u}_r^{(n)}\right)\in\mathbb{C}^{I_n\times J_n}$ and $\Phi_{r,s}^{(n)}=\Phi^{(n)}\left(\mathbf{U}^{(n,r)},\mathbf{U}^{(n,s)}\right)$. Construct

$$\begin{aligned}
\mathbf{P}^{(1,n)}&=\left[\Phi_{1,1}^{(n)},\Phi_{2,2}^{(n)},\ldots,\Phi_{R,R}^{(n)}\right]\in\mathbb{C}^{I_n^2J_n^2\times R},\\
\mathbf{P}^{(2,n)}&=\left[\Phi_{1,2}^{(n)},\Phi_{1,3}^{(n)},\Phi_{2,3}^{(n)},\ldots,\Phi_{R-1,R}^{(n)}\right]\in\mathbb{C}^{I_n^2J_n^2\times C_R^2},\\
\mathbf{P}^{(n)}&=\left[\mathbf{P}^{(1,n)},2\cdot\mathbf{P}^{(2,n)}\right]\in\mathbb{C}^{I_n^2J_n^2\times C_{R+1}^2}.
\end{aligned}$$

It can be verified that the SD problem boils down to finding the kernel of

$$\mathbf{Pm}=\mathbf{0}_{\left(\sum_{n=1}^N I_n^2J_n^2\right)}, \tag{S.1.5}$$

where

$$\mathbf{P}=\begin{bmatrix}\mathbf{P}^{(1)}\\ \vdots\\ \mathbf{P}^{(N)}\end{bmatrix}=\begin{bmatrix}\mathbf{P}^{(1,1)} & 2\cdot\mathbf{P}^{(2,1)}\\ \vdots & \vdots\\ \mathbf{P}^{(1,N)} & 2\cdot\mathbf{P}^{(2,N)}\end{bmatrix}\in\mathbb{C}^{\left(\sum_{n=1}^N I_n^2J_n^2\right)\times C_{R+1}^2}, \tag{S.1.6}$$

and

$$\mathbf{m}=\left[m_{11},m_{22},\ldots,m_{RR},m_{12},m_{13},\ldots,m_{R-1R}\right]^T.$$

Once the kernel of $\mathbf{P}$ has been determined, the coupled CPD of $\{\mathcal{X}^{(n)}\}$ can be found via a SD. The matrix $\mathbf{P}^{(n)}\in\mathbb{C}^{I_nJ_n\times R}$ contains identical row-vectors. After removing the redundant row-vectors of $\mathbf{P}^{(n)}$ we obtain the matrix $\widetilde{\mathbf{P}}^{(n)}\in\mathbb{C}^{C_{I_n}^2C_{J_n}^2\times C_{R+1}^2}$, corresponding to the matrix $\mathbf{R}_2(\widetilde{\mathcal{X}}^{(1)},\ldots,\widetilde{\mathcal{X}}^{(N)})\mathbf{S}$ in [4, Subsection 4.2.2].

From the kernel of $\mathbf{P}$ in (S.1.5) we obtain R symmetric matrices $\{\mathbf{M}^{(r)}\}$, admitting the factorizations (see [4, Subsection 4.2.2] for details):

$$\mathbf{M}^{(r)}=\mathbf{G}\Lambda^{(r)}\mathbf{G}^T,\quad r\in\{1,\ldots,R\}, \tag{S.1.7}$$

where $\mathbf{G}=\mathbf{C}^{-1}$ and $\Lambda^{(r)}\in\mathbb{C}^{R\times R}$ are diagonal matrices. In other words, the coupled CPD problem (S.1.1) has been reduced to a generalized eigenvalue problem. In the exact case, the SD problem (S.1.7) can for instance be solved by means of a Generalized Schur Decomposition (GSD) of a pair $(\mathbf{M}^{(r)},\mathbf{M}^{(s)})$.

S.1.2. Indirect SD. In cases where the dimensions $\{I_n,J_n\}$ are large such that $(\sum_{n=1}^N J_n^2)R^2$ is significantly smaller than $(\sum_{n=1}^N I_n^2J_n^2)$, we may determine the kernel of $\mathbf{P}$ via the Hermitian matrix

$$\begin{aligned}
\mathbf{P}^H\mathbf{P}&=\sum_{n=1}^N\mathbf{P}^{(n)H}\mathbf{P}^{(n)}\\
&=\sum_{n=1}^N\begin{bmatrix}\mathbf{P}^{(1,n)H}\mathbf{P}^{(1,n)} & 2\cdot\mathbf{P}^{(1,n)H}\mathbf{P}^{(2,n)}\\ 2\cdot\mathbf{P}^{(2,n)H}\mathbf{P}^{(1,n)} & 4\cdot\mathbf{P}^{(2,n)H}\mathbf{P}^{(2,n)}\end{bmatrix}\in\mathbb{C}^{C_{R+1}^2\times C_{R+1}^2},
\end{aligned} \tag{S.1.8}$$

where $\mathbf{P}^{(1,n)H}\mathbf{P}^{(1,n)} \in \mathbb{C}^{R\times R}$, $\mathbf{P}^{(1,n)H}\mathbf{P}^{(2,n)} \in \mathbb{C}^{R\times C_R^2}$ and $\mathbf{P}^{(2,n)H}\mathbf{P}^{(2,n)} \in \mathbb{C}^{C_R^2\times C_R^2}$ are given by

$$\mathbf{P}^{(1,n)H}\mathbf{P}^{(1,n)} = \begin{bmatrix} \Phi_{1,1}^{(n)H}\Phi_{1,1}^{(n)} & \Phi_{1,1}^{(n)H}\Phi_{2,2}^{(n)} & \cdots & \Phi_{1,1}^{(n)H}\Phi_{R,R}^{(n)} \\ \Phi_{2,2}^{(n)H}\Phi_{1,1}^{(n)} & \Phi_{2,2}^{(n)H}\Phi_{2,2}^{(n)} & \cdots & \Phi_{2,2}^{(n)H}\Phi_{R,R}^{(n)} \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{R,R}^{(n)H}\Phi_{1,1}^{(n)} & \Phi_{R,R}^{(n)H}\Phi_{2,2}^{(n)} & \cdots & \Phi_{R,R}^{(n)H}\Phi_{R,R}^{(n)} \end{bmatrix},$$

$$\mathbf{P}^{(1,n)H}\mathbf{P}^{(2,n)} = \begin{bmatrix} \Phi_{1,1}^{(n)H}\Phi_{1,2}^{(n)} & \Phi_{1,1}^{(n)H}\Phi_{1,3}^{(n)} & \Phi_{1,1}^{(n)H}\Phi_{2,3}^{(n)} & \cdots & \Phi_{1,1}^{(n)H}\Phi_{R-1,R}^{(n)} \\ \Phi_{2,2}^{(n)H}\Phi_{1,2}^{(n)} & \Phi_{2,2}^{(n)H}\Phi_{1,3}^{(n)} & \Phi_{2,2}^{(n)H}\Phi_{2,3}^{(n)} & \cdots & \Phi_{2,2}^{(n)H}\Phi_{R-1,R}^{(n)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Phi_{R,R}^{(n)H}\Phi_{1,2}^{(n)} & \Phi_{R,R}^{(n)H}\Phi_{1,3}^{(n)} & \Phi_{R,R}^{(n)H}\Phi_{2,3}^{(n)} & \cdots & \Phi_{R,R}^{(n)H}\Phi_{R-1,R}^{(n)} \end{bmatrix},$$

$$\mathbf{P}^{(2,n)H}\mathbf{P}^{(2,n)} = \begin{bmatrix} \Phi_{1,2}^{(n)H}\Phi_{1,2}^{(n)} & \Phi_{1,2}^{(n)H}\Phi_{1,3}^{(n)} & \Phi_{1,2}^{(n)H}\Phi_{2,3}^{(n)} & \cdots & \Phi_{1,2}^{(n)H}\Phi_{R-1,R}^{(n)} \\ \Phi_{1,3}^{(n)H}\Phi_{1,2}^{(n)} & \Phi_{1,3}^{(n)H}\Phi_{1,3}^{(n)} & \Phi_{1,3}^{(n)H}\Phi_{2,3}^{(n)} & \cdots & \Phi_{1,3}^{(n)H}\Phi_{R-1,R}^{(n)} \\ \Phi_{2,3}^{(n)H}\Phi_{1,2}^{(n)} & \Phi_{2,3}^{(n)H}\Phi_{1,3}^{(n)} & \Phi_{2,3}^{(n)H}\Phi_{2,3}^{(n)} & \cdots & \Phi_{2,3}^{(n)H}\Phi_{R-1,R}^{(n)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Phi_{R-1,R}^{(n)H}\Phi_{1,2}^{(n)} & \Phi_{R-1,R}^{(n)H}\Phi_{1,3}^{(n)} & \Phi_{R-1,R}^{(n)H}\Phi_{2,3}^{(n)} & \cdots & \Phi_{R-1,R}^{(n)H}\Phi_{R-1,R}^{(n)} \end{bmatrix}.$$

Note that the submatrices $\mathbf{P}^{(1,n)H}\mathbf{P}^{(1,n)}$ and $\mathbf{P}^{(2,n)H}\mathbf{P}^{(2,n)}$ of $\mathbf{P}^H\mathbf{P}$ are Hermitian. Consequently, only the upper C_{R+1}^2 entries of $\mathbf{P}^{(1,n)H}\mathbf{P}^{(1,n)}$, the upper $C_{R(R-1)/2}^2$ entries of $\mathbf{P}^{(2,n)H}\mathbf{P}^{(2,n)}$ and all the $R{\cdot}C_R^2$ entries of $\mathbf{P}^{(1,n)H}\mathbf{P}^{(2,n)}$ need to be computed.

In the following subsection we will explain how to efficiently compute $\mathbf{P}^H\mathbf{P}$ by capitalizing on the structure of the vector products $\Phi_{r_1,r_2}^{(n)H}\Phi_{s_1,s_2}^{(n)}$. This will be particularly useful in the case of large dimensions $\{I_n, J_n\}$. The complexity of the construction of $\mathbf{P}^H\mathbf{P}$ will measured in terms of flops. An addition, subtraction or multiplication will be counted as one flop and we do not distinguish between real and complex data. As an example, if $\mathbf{a}, \mathbf{b} \in \mathbb{C}^I$, then the vector product $\mathbf{a}^H\mathbf{b}$ requires $2I-1$ flops (I multiplications and $I-1$ additions). Likewise, if $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{I\times J}$, then the matrix product $\mathbf{A}^H\mathbf{B}$ requires $J^2(2I-1)$ flops.

S.1.2.1. Computation of $\Phi_{r_1,r_2}^{(n)H}\Phi_{s_1,s_2}^{(n)}$. The entries of $\mathbf{P}^{(n)H}\mathbf{P}^{(n)}$ are given by

$$\Phi_{r_1,r_2}^{(n)H}\Phi_{s_1,s_2}^{(n)} = \sum_{i_1,i_2=1}^{I_n}\sum_{j_1,j_2=1}^{J_n}\left(u_{i_1j_1}^{(n,r_1)}u_{i_2j_2}^{(n,r_2)} + u_{i_1j_1}^{(n,r_2)}u_{i_2j_2}^{(n,r_1)} - u_{i_1j_2}^{(n,r_1)}u_{i_2j_1}^{(n,r_2)} - u_{i_1j_2}^{(n,r_2)}u_{i_2j_1}^{(n,r_1)}\right)^* \cdot \left(u_{i_1j_1}^{(n,s_1)}u_{i_2j_2}^{(n,s_2)} + u_{i_1j_1}^{(n,s_2)}u_{i_2j_2}^{(n,s_1)} - u_{i_1j_2}^{(n,s_1)}u_{i_2j_1}^{(n,s_2)} - u_{i_1j_2}^{(n,s_2)}u_{i_2j_1}^{(n,s_1)}\right). \quad \text{(S.1.9)}$$

Note that

$$\sum_{i_1,i_2=1}^{I_n}\sum_{j_1,j_2=1}^{J_n} u_{i_1j_1}^{(n,r_1)*}u_{i_2j_2}^{(n,r_2)*}\left(u_{i_1j_1}^{(n,s_1)}u_{i_2j_2}^{(n,s_2)}+u_{i_1j_1}^{(n,s_2)}u_{i_2j_2}^{(n,s_1)}-u_{i_1j_2}^{(n,s_1)}u_{i_2j_1}^{(n,s_2)}-u_{i_1j_2}^{(n,s_2)}u_{i_2j_1}^{(n,s_1)}\right)$$
$$\begin{aligned}
&=\mathrm{Vec}\left(\mathbf{U}^{(n,r_1)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_1)}\right)\cdot\mathrm{Vec}\left(\mathbf{U}^{(n,r_2)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_2)}\right)\\
&+\mathrm{Vec}\left(\mathbf{U}^{(n,r_1)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_2)}\right)\cdot\mathrm{Vec}\left(\mathbf{U}^{(n,r_2)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_1)}\right)\\
&-\mathrm{Vec}\left(\mathbf{U}^{(n,s_1)H}\mathbf{U}^{(n,r_1)}\right)^H\cdot\mathrm{Vec}\left(\mathbf{U}^{(n,r_2)H}\mathbf{U}^{(n,s_2)}\right)\\
&-\mathrm{Vec}\left(\mathbf{U}^{(n,s_2)H}\mathbf{U}^{(n,r_1)}\right)^H\cdot\mathrm{Vec}\left(\mathbf{U}^{(n,r_2)H}\mathbf{U}^{(n,s_1)}\right),
\end{aligned}\tag{S.1.10}$$

$$\sum_{i_1,i_2=1}^{I_n}\sum_{j_1,j_2=1}^{J_n} u_{i_1j_1}^{(n,r_2)*}u_{i_2j_2}^{(n,r_1)*}\left(u_{i_1j_1}^{(n,s_1)}u_{i_2j_2}^{(n,s_2)}+u_{i_1j_1}^{(n,s_2)}u_{i_2j_2}^{(n,s_1)}-u_{i_1j_2}^{(n,s_1)}u_{i_2j_1}^{(n,s_2)}-u_{i_1j_2}^{(n,s_2)}u_{i_2j_1}^{(n,s_1)}\right)$$
$$\begin{aligned}
&=\mathrm{Vec}\left(\mathbf{U}^{(n,r_2)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_1)}\right)\cdot\mathrm{Vec}\left(\mathbf{U}^{(n,r_1)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_2)}\right)\\
&+\mathrm{Vec}\left(\mathbf{U}^{(n,r_2)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_2)}\right)\cdot\mathrm{Vec}\left(\mathbf{U}^{(n,r_1)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_1)}\right)\\
&-\mathrm{Vec}\left(\mathbf{U}^{(n,s_1)H}\mathbf{U}^{(n,r_2)}\right)^H\cdot\mathrm{Vec}\left(\mathbf{U}^{(n,r_1)H}\mathbf{U}^{(n,s_2)}\right)\\
&-\mathrm{Vec}\left(\mathbf{U}^{(n,s_2)H}\mathbf{U}^{(n,r_2)}\right)^H\cdot\mathrm{Vec}\left(\mathbf{U}^{(n,r_1)H}\mathbf{U}^{(n,s_1)}\right),
\end{aligned}\tag{S.1.11}$$

$$\sum_{i_1,i_2=1}^{I_n}\sum_{j_1,j_2=1}^{J_n} -u_{i_1j_2}^{(n,r_1)*}u_{i_2j_1}^{(n,r_2)*}\left(u_{i_1j_1}^{(n,s_1)}u_{i_2j_2}^{(n,s_2)}+u_{i_1j_1}^{(n,s_2)}u_{i_2j_2}^{(n,s_1)}-u_{i_1j_2}^{(n,s_1)}u_{i_2j_1}^{(n,s_2)}-u_{i_1j_2}^{(n,s_2)}u_{i_2j_1}^{(n,s_1)}\right)$$
$$\begin{aligned}
&=\mathrm{Vec}\left(\mathbf{U}^{(n,r_1)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_1)}\right)\cdot\mathrm{Vec}\left(\mathbf{U}^{(n,r_2)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_2)}\right)\\
&+\mathrm{Vec}\left(\mathbf{U}^{(n,r_1)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_2)}\right)\cdot\mathrm{Vec}\left(\mathbf{U}^{(n,r_2)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_1)}\right)\\
&-\mathrm{Vec}\left(\mathbf{U}^{(n,s_1)H}\mathbf{U}^{(n,r_1)}\right)^H\cdot\mathrm{Vec}\left(\mathbf{U}^{(n,r_2)H}\mathbf{U}^{(n,s_2)}\right)\\
&-\mathrm{Vec}\left(\mathbf{U}^{(n,s_2)H}\mathbf{U}^{(n,r_1)}\right)^H\cdot\mathrm{Vec}\left(\mathbf{U}^{(n,r_2)H}\mathbf{U}^{(n,s_1)}\right),
\end{aligned}\tag{S.1.12}$$

$$\sum_{i_1,i_2=1}^{I_n}\sum_{j_1,j_2=1}^{J_n} -u_{i_1j_2}^{(n,r_2)*}u_{i_2j_1}^{(n,r_1)*}\left(u_{i_1j_1}^{(n,s_1)}u_{i_2j_2}^{(n,s_2)}+u_{i_1j_1}^{(n,s_2)}u_{i_2j_2}^{(n,s_1)}-u_{i_1j_2}^{(n,s_1)}u_{i_2j_1}^{(n,s_2)}-u_{i_1j_2}^{(n,s_2)}u_{i_2j_1}^{(n,s_1)}\right)$$
$$\begin{aligned}
&=\mathrm{Vec}\left(\mathbf{U}^{(n,r_2)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_1)}\right)\cdot\mathrm{Vec}\left(\mathbf{U}^{(n,r_1)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_2)}\right)\\
&+\mathrm{Vec}\left(\mathbf{U}^{(n,r_2)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_2)}\right)\cdot\mathrm{Vec}\left(\mathbf{U}^{(n,r_1)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_1)}\right)\\
&-\mathrm{Vec}\left(\mathbf{U}^{(n,s_1)H}\mathbf{U}^{(n,r_2)}\right)^H\cdot\mathrm{Vec}\left(\mathbf{U}^{(n,r_1)H}\mathbf{U}^{(n,s_2)}\right)\\
&-\mathrm{Vec}\left(\mathbf{U}^{(n,s_2)H}\mathbf{U}^{(n,r_2)}\right)^H\cdot\mathrm{Vec}\left(\mathbf{U}^{(n,r_1)H}\mathbf{U}^{(n,s_1)}\right).
\end{aligned}\tag{S.1.13}$$

Since $\text{Vec}\left(\mathbf{U}^{(n,s_1)H}\mathbf{U}^{(n,r_1)}\right)^H \cdot \text{Vec}\left(\mathbf{U}^{(n,r_2)H}\mathbf{U}^{(n,s_2)}\right) = \text{Vec}\left(\mathbf{U}^{(n,s_2)H}\mathbf{U}^{(n,r_2)}\right)^H \cdot \text{Vec}\left(\mathbf{U}^{(n,r_1)H}\mathbf{U}^{(n,s_1)}\right)$ the expressions (S.1.10)–(S.1.13) are identical.

S.1.2.2. Computation of $\mathbf{U}^{(n,s)H}\mathbf{U}^{(n,r)}$. The computation of $\mathbf{U}^{(n,r)H}\mathbf{U}^{(n,s)}$ with $\mathbf{U}^{(n,r)}, \mathbf{U}^{(n,s)} \in \mathbb{C}^{I_n \times J_n}$ requires $J_n^2(2I_n-1)$ flops. For each $n \in \{1,\ldots,N\}$ there are C_{R+1}^2 distinct matrix products of the form $\mathbf{U}^{(n,r)H}\mathbf{U}^{(n,s)}$. Hence, the computation of the matrices $\{\mathbf{U}^{(n,r)H}\mathbf{U}^{(n,s)}\}$ requires

$$C_{R+1}^2 \sum_{n=1}^{N} J_n^2(2I_n-1) \text{ flops.} \tag{S.1.14}$$

In the following subsections we assume that the matrix products $\{\mathbf{U}^{(n,s)H}\mathbf{U}^{(n,r)}\}$ have been computed.

S.1.2.3. Computation of $\text{Vec}\left(\mathbf{U}^{(n,r)}\right)^H \text{Vec}\left(\mathbf{U}^{(n,s)}\right)$. Assuming that the matrix product $\mathbf{U}^{(n,r)H}\mathbf{U}^{(n,s)}$ is known, then since

$$\mathbf{U}^{(n,r)H}\mathbf{U}^{(n,s)} = \begin{bmatrix} \mathbf{u}_1^{(n,r)H}\mathbf{u}_1^{(n,s)} & \mathbf{u}_1^{(n,r)H}\mathbf{u}_2^{(n,s)} & \cdots & \mathbf{u}_1^{(n,r)H}\mathbf{u}_{J_n}^{(n,s)} \\ \mathbf{u}_2^{(n,r)H}\mathbf{u}_1^{(n,s)} & \mathbf{u}_2^{(n,r)H}\mathbf{u}_2^{(n,s)} & \cdots & \mathbf{u}_2^{(n,r)H}\mathbf{u}_{J_n}^{(n,s)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_{J_n}^{(n,r)H}\mathbf{u}_1^{(n,s)} & \mathbf{u}_{J_n}^{(n,r)H}\mathbf{u}_2^{(n,s)} & \cdots & \mathbf{u}_{J_n}^{(n,r)H}\mathbf{u}_{J_n}^{(n,s)} \end{bmatrix},$$

$$\text{Vec}\left(\mathbf{U}^{(n,r)}\right) = \begin{bmatrix} \mathbf{u}_1^{(n,r)} & \mathbf{u}_2^{(n,r)} & \ldots & \mathbf{u}_{J_n}^{(n,r)} \end{bmatrix}^T,$$

$$\text{Vec}\left(\mathbf{U}^{(n,s)}\right) = \begin{bmatrix} \mathbf{u}_1^{(n,s)} & \mathbf{u}_2^{(n,s)} & \ldots & \mathbf{u}_{J_n}^{(n,s)} \end{bmatrix}^T,$$

it is clear that the vector product $\text{Vec}\left(\mathbf{U}^{(n,r)}\right)^H \text{Vec}\left(\mathbf{U}^{(n,s)}\right) = \sum_{j=1}^{J_n} \mathbf{u}_j^{(n,r)H}\mathbf{u}_j^{(n,s)} = \sum_{j=1}^{J_n}(\mathbf{U}^{(n,r)H}\mathbf{U}^{(n,s)})_{jj} = \text{Tr}\left(\mathbf{U}^{(n,r)H}\mathbf{U}^{(n,s)}\right)$ only requires $J_n - 1$ flops, where $\text{Tr}(\cdot)$ denotes the trace of a matrix.

S.1.2.4. Computation of $\mathbf{P}^{(1,n)H}\mathbf{P}^{(1,n)}$. From (S.1.10)–(S.1.13) we observe that the entries of $\mathbf{P}^{(1,n)H}\mathbf{P}^{(1,n)}$ are given by

$$\begin{aligned}\Phi_{r,r}^{(n)H}\Phi_{s,s}^{(n)} = 8\Big(& \left(\text{Vec}\left(\mathbf{U}^{(n,r)}\right)^H \text{Vec}\left(\mathbf{U}^{(n,s)}\right)\right)^2 \\ & - \text{Vec}\left(\mathbf{U}^{(n,s)H}\mathbf{U}^{(n,r)}\right)^H \text{Vec}\left(\mathbf{U}^{(n,r)H}\mathbf{U}^{(n,s)}\right)\Big).\end{aligned} \tag{S.1.15}$$

As explained in Subsection S.1.2.3 the computation of $\text{Vec}\left(\mathbf{U}^{(n,r)}\right)^H \text{Vec}\left(\mathbf{U}^{(n,s)}\right)$ requires $J_n - 1$ flops. Assuming that the matrices $\{\mathbf{U}^{(n,r)H}\mathbf{U}^{(n,s)}\}$ have already been computed, the computation of $\text{Vec}\left(\mathbf{U}^{(n,s)H}\mathbf{U}^{(n,r)}\right)^H \text{Vec}\left(\mathbf{U}^{(n,r)H}\mathbf{U}^{(n,s)}\right)$ additionally requires $2J_n^2 - 1$ flops. The subtraction of the terms in (S.1.15) costs one flop, the multiplications with 8 in (S.1.15) costs one flop and the squaring $(\cdot)^2$ in (S.1.15) costs one flop (i.e., 3 additional flops). Hence, the computation of (S.1.15) requires

$$(J_n - 1) + (2J_n^2 - 1) + 3 \text{ flops.}$$

Recall also that $\mathbf{P}^{(1,n)H}\mathbf{P}^{(1,n)}$ has C_{R+1}^2 distinct entries of the form (S.1.15). Hence, the computation of the N matrices $\{\mathbf{P}^{(1,n)H}\mathbf{P}^{(1,n)}\}$ requires

$$\begin{aligned} C_{R+1}^2 \quad &\sum_{n=1}^{N}((J_n-1)+(2J_n^2-1)+3) = C_{R+1}^2\sum_{n=1}^{N}(J_n(J_n+1)+J_n^2)+N\cdot C_{R+1}^2 \\ &= C_{R+1}^2\sum_{n=1}^{N}J_n(J_n+1)+C_{R+1}^2\sum_{n=1}^{N}J_n^2+N\cdot C_{R+1}^2 \ \text{ flops.} \end{aligned} \tag{S.1.16}$$

S.1.2.5. Computation of $\mathbf{P}^{(1,n)H}\mathbf{P}^{(2,n)}$. From (S.1.10)–(S.1.13) we observe that the entries of $\mathbf{P}^{(1,n)H}\mathbf{P}^{(2,n)}$ are given by

$$\begin{aligned} \Phi_{r,r}^{(n)H}\Phi_{s_1,s_2}^{(n)} = 8\Big(&\mathrm{Vec}\left(\mathbf{U}^{(n,r)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_1)}\right)\cdot\mathrm{Vec}\left(\mathbf{U}^{(n,r)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_2)}\right) \\ &-\mathrm{Vec}\left(\mathbf{U}^{(n,s_1)H}\mathbf{U}^{(n,r)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,r)H}\mathbf{U}^{(n,s_2)}\right)\Big). \end{aligned} \tag{S.1.17}$$

Comparing (S.1.15) with (S.1.17), the latter expression $\Phi_{r,r}^{(n)H}\Phi_{s_1,s_2}^{(n)}$ requires the additional vector-vector product $\mathrm{Vec}\left(\mathbf{U}^{(n,r)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_2)}\right)$ compared to the former expression $\Phi_{r,r}^{(n)H}\Phi_{s,s}^{(n)}$. Recall that the vector product $\mathrm{Vec}\left(\mathbf{U}^{(n,r)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_2)}\right)$ costs J_n-1 flops. Thus, the computation of (S.1.17) requires

$$2(J_n-1)+(2J_n^2-1)+3 \ \text{ flops.}$$

Recall also that the matrix $\mathbf{P}^{(1,n)H}\mathbf{P}^{(2,n)}$ has $R\cdot C_R^2$ distinct entries. Thus, the computation of the N matrices $\{\mathbf{P}^{(1,n)H}\mathbf{P}^{(2,n)}\}$ requires

$$R\cdot C_R^2\sum_{n=1}^{N}(2\cdot(J_n-1)+(2J_n^2-1)+3) = R\cdot C_R^2\sum_{n=1}^{N}2J_n(J_n+1) \text{ flops.} \tag{S.1.18}$$

S.1.2.6. Computation of $\mathbf{P}^{(2,n)H}\mathbf{P}^{(2,n)}$. From (S.1.10)–(S.1.13) we observe that the entries of $\mathbf{P}^{(2,n)H}\mathbf{P}^{(2,n)}$ are given by

$$\begin{aligned} \Phi_{r_1,r_2}^{(n)H}\Phi_{s_1,s_2}^{(n)} = 4\Big(&\mathrm{Vec}\left(\mathbf{U}^{(n,r_1)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_1)}\right)\cdot\mathrm{Vec}\left(\mathbf{U}^{(n,r_2)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_2)}\right) \\ &+\mathrm{Vec}\left(\mathbf{U}^{(n,r_2)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_1)}\right)\cdot\mathrm{Vec}\left(\mathbf{U}^{(n,r_1)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_2)}\right) \\ &-\mathrm{Vec}\left(\mathbf{U}^{(n,s_1)H}\mathbf{U}^{(n,r_1)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,r_2)H}\mathbf{U}^{(n,s_2)}\right) \\ &-\mathrm{Vec}\left(\mathbf{U}^{(n,s_1)H}\mathbf{U}^{(n,r_2)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,r_1)H}\mathbf{U}^{(n,s_2)}\right)\Big). \end{aligned} \tag{S.1.19}$$

Observe that (S.1.19) involves 4 vector products of the form

$$\mathrm{Vec}\left(\mathbf{U}^{(n,r_1)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,s_1)}\right)$$

each requiring (J_n-1) flops, 2 vector products of the form

$$\mathrm{Vec}\left(\mathbf{U}^{(n,s_1)H}\mathbf{U}^{(n,r_1)}\right)^H\mathrm{Vec}\left(\mathbf{U}^{(n,r_2)H}\mathbf{U}^{(n,s_2)}\right)$$

each requiring $(2J_n^2 - 1)$ flops, three scalar multiplications (3 flops) and three scalar additions/subtractions (3 flops). Overall, the computation of (S.1.19) requires

$$4(J_n - 1) + 2(2J_n^2 - 1) + 6 \text{ flops.}$$

Recall that $\mathbf{P}^{(2,n)H}\mathbf{P}^{(2,n)}$ has $C^2_{R(R-1)/2}$ distinct entries. We conclude that the computation of the N matrices $\{\mathbf{P}^{(2,n)H}\mathbf{P}^{(2,n)}\}$ requires

$$C^2_{R(R-1)/2}\sum_{n=1}^{N}(4(J_n-1)+2(2J_n^2-1)+6) = C^2_{R(R-1)/2}\sum_{n=1}^{N}4J_n(J_n+1) \text{ flops.} \quad \text{(S.1.20)}$$

S.1.2.7. Overall computation of $\mathbf{P}^H\mathbf{P}$. From (S.1.14), (S.1.16), (S.1.18) and (S.1.20) it is clear that the construction of the matrices $\{\mathbf{P}^{(1,n)H}\mathbf{P}^{(1,n)}\}$, $\{\mathbf{P}^{(1,n)H}\mathbf{P}^{(2,n)}\}$ and $\{\mathbf{P}^{(2,n)H}\mathbf{P}^{(2,n)}\}$ requires

$$C^2_{R+1}\sum_{n=1}^{N}J_n^2(2I_n - 1) + (C^2_{R+1} + 2R\cdot C^2_R + 4\cdot C^2_{R(R-1)/2})\sum_{n=1}^{N}J_n(J_n+1) +$$
$$C^2_{R+1}\sum_{n=1}^{N}J_n^2 + N\cdot C^2_{R+1} \text{ flops.} \quad \text{(S.1.21)}$$

Finally, in order to construct $\mathbf{P}^H\mathbf{P}$ we need to compute $\sum_{n=1}^{N}\mathbf{P}^{(1,n)H}\mathbf{P}^{(1,n)}$, $2\cdot\sum_{n=1}^{N}\mathbf{P}^{(1,n)H}\mathbf{P}^{(2,n)}$ and $4\cdot\sum_{n=1}^{N}\mathbf{P}^{(2,n)H}\mathbf{P}^{(2,n)}$. Taking the symmetries of $\mathbf{P}^{(1,n)H}\mathbf{P}^{(1,n)}$ and $\mathbf{P}^{(2,n)H}\mathbf{P}^{(2,n)}$ into account, this requires

$$(N-1)(C^2_{R+1} + RC^2_R + C^2_{R(R-1)/2}) + (RC^2_R + C^2_{R(R-1)/2}) \text{ flops,} \quad \text{(S.1.22)}$$

where the two terms in (S.1.22) correspond to the number of additions and multiplications, respectively. We conclude that the flops needed for the overall computation of $\mathbf{P}^H\mathbf{P}$ are equal to the sum of (S.1.21) and (S.1.22). For large dimensions $\{I_n, J_n\}$ the complexity of the computation of $\mathbf{P}^H\mathbf{P}$ is proportional to

$$\max\left(\left(\sum_{n=1}^{N}I_nJ_n^2\right), \left(\sum_{n=1}^{N}J_n^2\right)R^2\right)R^2. \quad \text{(S.1.23)}$$

The $\max(\cdot)$ operator in (S.1.23) says that if R is small compared to the dimensions $\{I_n, J_n\}$ such that $R^2 \leq (\sum_{n=1}^{N}I_nJ_n^2)/(\sum_{n=1}^{N}J_n^2)$, then the computational cost is dominated by the construction of matrices the $\{\mathbf{U}^{(n,r)H}\mathbf{U}^{(n,s)}\}$ with complexity (S.1.14). On the other hand, if R is large compared to the dimensions $\{I_n, J_n\}$, i.e., $R^2 > (\sum_{n=1}^{N}I_nJ_n^2)/(\sum_{n=1}^{N}J_n^2)$, then the computational cost is dominated by the computation of the scalars $\{\Phi^{(n)H}_{r_1,r_2}\Phi^{(n)}_{s_1,s_2}\}$ with complexity (S.1.20).

We also conclude that if $(\sum_{n=1}^{N}J_n^2)R^2 < \sum_{n=1}^{N}I_n^2J_n^2$, then this approach requires fewer flops than the SD procedure described in [4] that explicitly constructs $\mathbf{P}$ with a complexity proportional to $(\sum_{n=1}^{N}I_n^2J_n^2)R^2$.

S.2. Extension of the SD method to tensors of arbitrary order. In this section we explain how to extend the SD method for coupled CPD of third-order tensors to tensors of arbitrary order. More precisely, the goal of this section is to explain how to transform the coupled CPD problem of tensors of arbitrary order into a coupled CPD problem of a set of third-order tensors.

Consider the coupled PD of the second-order (matrix), third-order and higher-order tensor factorizations

$$\mathbb{C}^{H_m \times K} \ni \mathbf{Y}^{(m)} = \sum_{r=1}^{R} \mathbf{d}_r^{(m)} \circ \mathbf{c}_r \,, \; m \in \{1, \ldots, M\}, \tag{S.2.1}$$

$$\mathbb{C}^{I_{1,n} \times I_{2,n} \times K} \ni \mathcal{V}^{(n)} = \sum_{r=1}^{R} \mathbf{a}_r^{(n)} \circ \mathbf{b}_r^{(n)} \circ \mathbf{c}_r \,, \; n \in \{1, \ldots, N\}, \tag{S.2.2}$$

$$\mathbb{C}^{J_{1,p} \times \cdots \times J_{Q_p,p} \times K} \ni \mathcal{W}^{(p)} = \sum_{r=1}^{R} \mathbf{a}_r^{(1,p)} \circ \cdots \circ \mathbf{a}_r^{(Q_p,p)} \circ \mathbf{c}_r \,, \; p \in \{1, \ldots, P\}, \tag{S.2.3}$$

in which $Q_p > 3$, $\forall p \in \{1, \ldots, P\}$. The PD of the third-order tensors $\{\mathcal{V}^{(n)}\}$ and higher-order tensors $\{\mathcal{W}^{(p)}\}$ admit the following matrix representations

$$\mathbb{C}^{I_{1,n} I_{2,n} \times K} \ni \mathbf{V}^{(n)} = (\mathbf{A}^{(n)} \odot \mathbf{B}^{(n)}) \mathbf{C}^T, \; n \in \{1, \ldots, N\}, \tag{S.2.4}$$

$$\mathbb{C}^{J_{1,p} \times \cdots \times J_{Q_p,p} \times K} \ni \mathbf{W}^{(p)} = (\mathbf{A}^{(1,p)} \odot \cdots \odot \mathbf{A}^{(Q_p,p)}) \mathbf{C}^T, \; p \in \{1, \ldots, P\}. \tag{S.2.5}$$

Step 1: Coupled CPD via structured matrix decomposition. The first step is to formulate the coupled CPD problem as a low-rank constrained matrix decomposition problem. Similar to (S.1.2) this is achieved by collecting the matrices (S.2.1), (S.2.5) and (S.2.5) into the matrix

$$\mathbf{X} = \begin{bmatrix} \mathbf{Y} \\ \mathbf{V} \\ \mathbf{W} \end{bmatrix} = \mathbf{F}\mathbf{C}^T, \quad \mathbf{Y} = \begin{bmatrix} \mathbf{Y}^{(1)} \\ \vdots \\ \mathbf{Y}^{(M)} \end{bmatrix}, \; \mathbf{V} = \begin{bmatrix} \mathbf{V}^{(1)} \\ \vdots \\ \mathbf{V}^{(N)} \end{bmatrix}, \; \mathbf{W} = \begin{bmatrix} \mathbf{W}^{(1)} \\ \vdots \\ \mathbf{W}^{(P)} \end{bmatrix}, \tag{S.2.6}$$

where $\mathbf{F}$ in (S.1.3) now takes the form

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}^{(1)} \\ \mathbf{F}^{(2)} \\ \mathbf{F}^{(3)} \end{bmatrix},$$

in which

$$\mathbf{F}^{(1)} = \begin{bmatrix} \mathbf{D}^{(1)} \\ \vdots \\ \mathbf{D}^{(M)} \end{bmatrix}, \; \mathbf{F}^{(2)} = \begin{bmatrix} \mathbf{A}^{(1)} \odot \mathbf{B}^{(1)} \\ \vdots \\ \mathbf{A}^{(N)} \odot \mathbf{B}^{(N)} \end{bmatrix}, \; \mathbf{F}^{(3)} = \begin{bmatrix} \mathbf{A}^{(1,1)} \odot \cdots \odot \mathbf{A}^{(Q_1,1)} \\ \vdots \\ \mathbf{A}^{(1,P)} \odot \cdots \odot \mathbf{A}^{(Q_P,P)} \end{bmatrix}. \tag{S.2.7}$$

Step 2: Find a basis for range $(\mathbf{X})$ and apply dimensionality reduction. In the second step we first find a basis for range $(\mathbf{X})$. The matrix $\mathbf{Y}$ in (S.2.6) will only be used when finding a basis for range $(\mathbf{X})$. Let $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^H$ denote the compact SVD of $\mathbf{X}$, then the columns of $\mathbf{U}\Sigma$ will be used as a basis for range $(\mathbf{X})$. In order to reduce the complexity of the SD procedure we reduce the dimension of the "third-mode" matrix $\mathbf{C}$. Overall, the SVD of $\mathbf{X}$ yields

$$\mathbf{U}\Sigma = \mathbf{X}\mathbf{V} \text{ and } \widetilde{\mathbf{C}} = \mathbf{V}^T\mathbf{C} \in \mathbb{C}^{R \times R}. \tag{S.2.8}$$

Partition $\mathbf{U}\Sigma$ as follows

$$\mathbf{U}\Sigma = \begin{bmatrix} \mathbf{U}^{(1)} \\ \mathbf{U}^{(2)} \\ \mathbf{U}^{(3)} \end{bmatrix}, \tag{S.2.9}$$

where $\mathbf{U}^{(1)} \in \mathbb{C}^{(\sum_{m=1}^{M} H_m)\times R}$, $\mathbf{U}^{(2)} \in \mathbb{C}^{(\sum_{n=1}^{N} I_{1,n} I_{2,n})\times R}$ and $\mathbf{U}^{(3)} \in \mathbb{C}^{(\sum_{p=1}^{P} \prod_{q=1}^{Q_p} J_{q,p})\times R}$. Since the submatrix $\mathbf{Y}$ of $\mathbf{X}$ is only used to help find a basis for range$(\mathbf{X})$ = range$(\mathbf{U})$ and range$\left(\mathbf{C}^T\right)$ = range$\left(\mathbf{V}^T\right)$, the associated submatrix $\mathbf{U}^{(1)}$ of $\mathbf{U}$ will not be considered in the development of the following SD procedure. More precisely, the matrix $\mathbf{U}^{(1)}$ will not be used in the SD procedure since the columns of $\mathbf{U}^{(1)}$ are not subject to a low-rank constraint. We further partition $\mathbf{U}^{(2)}$ and $\mathbf{U}^{(3)}$ as follows

$$\mathbf{U}^{(2)} = \begin{bmatrix} \mathbf{U}^{(1,2)} \\ \vdots \\ \mathbf{U}^{(N,2)} \end{bmatrix}, \quad \mathbf{U}^{(n,2)} \in \mathbb{C}^{(I_{1,n} I_{2,n})\times R}, \tag{S.2.10}$$

$$\mathbf{U}^{(3)} = \begin{bmatrix} \mathbf{U}^{(1,3)} \\ \vdots \\ \mathbf{U}^{(P,3)} \end{bmatrix}, \quad \mathbf{U}^{(p,3)} \in \mathbb{C}^{(\prod_{q=1}^{Q_p} J_{q,p})\times R}, \tag{S.2.11}$$

with properties $\mathbf{U}^{(n,2)} = (\mathbf{A}^{(n)} \odot \mathbf{B}^{(n)})\widetilde{\mathbf{C}}^T$ and $\mathbf{U}^{(p,3)} = (\mathbf{A}^{(1,p)} \odot \cdots \odot \mathbf{A}^{(Q_p,p)})\widetilde{\mathbf{C}}^T$.

Note that the matrix decompositions $\mathbf{U}^{(n,2)} = (\mathbf{A}^{(n)} \odot \mathbf{B}^{(n)})\widetilde{\mathbf{C}}^T$, $n \in \{1, \ldots, N\}$, in (S.2.10) already constitute matrix representations of third-order CPDs. In the next step we will further transform the decompositions of $\{\mathbf{U}^{(p,3)}\}$ in (S.2.11) into a set of coupled *third-order* CPD problems. In other words, we reduce (S.2.6) to a standard coupled CPD problem that can be solved by via the SD procedure for coupled CPD of third-order tensors.

Step 3: From coupled CPD of tensors of arbitrary order to coupled CPD of third-order tensors. Let $\mathbf{P}^{(p,q,3)} \in \mathbb{C}^{(\prod_{q=1}^{Q_p} J_{q,p})\times(\prod_{q=1}^{Q_p} J_{q,p})}$ denote the row-permutation matrix with property $\mathbf{U}^{(q,p,3)} := \mathbf{P}^{(q,p,3)}\mathbf{U}^{(p,3)} = (\mathbf{A}^{(q,p)} \odot \mathbf{B}^{(q,p)})\widetilde{\mathbf{C}}^T$ in which $\mathbf{B}^{(q,p)} := \mathbf{A}^{(q+1,p)} \odot \cdots \odot \mathbf{A}^{(Q_p,p)} \odot \mathbf{A}^{(1,p)} \odot \cdots \odot \mathbf{A}^{(q-1,p)}$. From $\mathbf{U}^{(p,3)}$ we extract Q_p matrices with joint matrix factorization

$$\begin{bmatrix} \mathbf{U}^{(1,p,3)} \\ \vdots \\ \mathbf{U}^{(Q_p,p,3)} \end{bmatrix} = \begin{bmatrix} \mathbf{U}^{(1,p,3)} \\ \vdots \\ \mathbf{U}^{(Q_p,p,3)} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^{(1,p)} \odot \mathbf{B}^{(1,p)} \\ \vdots \\ \mathbf{A}^{(Q_p,p)} \odot \mathbf{B}^{(Q_p,p)} \end{bmatrix} \widetilde{\mathbf{C}}^T. \tag{S.2.12}$$

Overall, from (S.2.10) and (S.2.12) we obtain

$$\left[\begin{array}{c}\mathbf{U}^{(1,2)}\\ \vdots\\ \mathbf{U}^{(N,2)}\\ \hline \mathbf{U}^{(1,1,3)}\\ \vdots\\ \mathbf{U}^{(Q_1,1,3)}\\ \hline \vdots\\ \hline \mathbf{U}^{(1,P,3)}\\ \vdots\\ \mathbf{U}^{(Q_P,P,3)}\end{array}\right] = \left[\begin{array}{c}\mathbf{A}^{(1)}\odot\mathbf{B}^{(1)}\\ \vdots\\ \mathbf{A}^{(N)}\odot\mathbf{B}^{(N)}\\ \hline \mathbf{A}^{(1,1)}\odot\mathbf{B}^{(1,1)}\\ \vdots\\ \mathbf{A}^{(Q_1,1)}\odot\mathbf{B}^{(Q_1,1)}\\ \hline \vdots\\ \hline \mathbf{A}^{(1,P)}\odot\mathbf{B}^{(1,P)}\\ \vdots\\ \mathbf{A}^{(Q_P,P)}\odot\mathbf{B}^{(Q_P,P)}\end{array}\right]\widetilde{\mathbf{C}}^T. \tag{S.2.13}$$

From (S.2.13) it is now clear that the discussed SD procedures developed for third-order tensors can also be used to compute the coupled CPD of tensors of arbitrary order.

S.3. Supplementary material related to Section 5. In this section we prove the following lemma.

LEMMA S.3.1. *Assume that the matrix*

$$\boldsymbol{E} = \left[\begin{array}{c}\mathcal{C}_{L+1}\left(\boldsymbol{A}^{(1)}\right)\odot\mathcal{C}_{L+1}\left(\boldsymbol{B}^{(1)}\right)\\ \vdots\\ \mathcal{C}_{L+1}\left(\boldsymbol{A}^{(N)}\right)\odot\mathcal{C}_{L+1}\left(\boldsymbol{B}^{(N)}\right)\end{array}\right]\mathbf{P}\in\mathbb{C}^{\left(\sum_{n=1}^{N}C_{I_n}^{L+1}C_{J_n}^{L+1}\right)\times\left(C_{R+L}^{L+1}-R\right)},$$

has full column rank, where $\boldsymbol{A}^{(n)} = [\boldsymbol{A}^{(1,n)}\ \cdots\ \boldsymbol{A}^{(R,n)}] \in \mathbb{C}^{I_n\times RL}$ *and* $\boldsymbol{B}^{(n)} = [\boldsymbol{B}^{(1,n)}\ \cdots\ \boldsymbol{B}^{(R,n)}] \in \mathbb{C}^{J_n\times RL}$, *and* $\mathbf{P}$ *is defined by [4, eq. (5.16)]. Then*

(i) $\max\limits_{1\le n\le N} r(\boldsymbol{A}^{(r,n)}\boldsymbol{B}^{(r,n)T}) = L$ *for all* $r\in\{1,\ldots,R\}$;

(ii) the matrix $\mathbf{F}^{(red)}$ *given by [4, eq. (3.8)] has full column rank.*

Proof. (i) We use the following properties of compound matrices [2, p. 19–22]: if k is a positive integer and $\mathbf{X}$ and $\mathbf{Y}$ are matrices such that $\mathbf{XY}$, $\mathcal{C}_k(\mathbf{X})$, and $\mathcal{C}_k(\mathbf{Y})$ are defined, then

$$\mathcal{C}_k(\mathbf{XY}) = \mathcal{C}_k(\mathbf{X})\mathcal{C}_k(\mathbf{Y}), \tag{S.3.1}$$

$$r(\mathbf{XY}) \le k-1 \quad \text{if and only if} \quad \mathcal{C}_k(\mathbf{XY}) = \mathbf{O}. \tag{S.3.2}$$

The first property is known as "Binet-Cauchy formula" and the second property follows from the fact that the entries of $\mathcal{C}_k(\mathbf{XY})$ are all possible $k\times k$ minors of $\mathbf{XY}$.

We prove statement (i) for $r=1$. The general case can be proved in the same way. Assume to the contrary that (i) does not hold for $r=1$, that is

$$r(\mathbf{A}^{(1,n)}\mathbf{B}^{(1,n)T}) \le L-1 \quad \text{for all} \quad n\in\{1,\ldots,N\}. \tag{S.3.3}$$

We will arrive at a contradiction by showing that the first column of the matrix $\mathbf{E}$ is zero. By construction of $\mathbf{P}$, the first column of $\mathbf{P}$ is enumerated by the $(L+1)$-tuple

$(1,\dots,1,2)$ of Ω defined by [4, eq. (5.15)], and the entries of the first column (or the rows) of $\mathbf{P}$ can be enumerated by means of the elements of $\Sigma := \{(i_1,\dots,i_{L+1}) : 1 \le i_1,\dots,i_{L+1} \le LR\}$. Hence, by [4, eq. (5.16)], the nonzero entries of the first column of $\mathbf{P}$ are enumerated by means of the $(L+1)$-tuples $\{(1,2,\dots,L,k)\}_{k=L+1}^{2L}$. This in turn means that the first (or the $(1,\dots,1,2)$nd) column of the matrix $\mathbf{E}$ equals

$$
\begin{aligned}
\mathbf{E}_{(1,\dots,1,2)} &= \sum_{k=L+1}^{2L} \begin{bmatrix} \mathcal{C}_{L+1}\left(\mathbf{A}^{(1)}\right) \odot \mathcal{C}_{L+1}\left(\mathbf{B}^{(1)}\right) \\ \vdots \\ \mathcal{C}_{L+1}\left(\mathbf{A}^{(N)}\right) \odot \mathcal{C}_{L+1}\left(\mathbf{B}^{(N)}\right) \end{bmatrix}_{(1,2,\dots,L,k)} \\
&= \sum_{k=L+1}^{2L} \begin{bmatrix} \mathcal{C}_{L+1}\left(\mathbf{A}^{(1)}\right)_{(1,2,\dots,L,k)} \otimes \mathcal{C}_{L+1}\left(\mathbf{B}^{(1)}\right)_{(1,2,\dots,L,k)} \\ \vdots \\ \mathcal{C}_{L+1}\left(\mathbf{A}^{(N)}\right)_{(1,2,\dots,L,k)} \otimes \mathcal{C}_{L+1}\left(\mathbf{B}^{(N)}\right)_{(1,2,\dots,L,k)} \end{bmatrix} \\
&= \sum_{k=1}^{L} \begin{bmatrix} \mathcal{C}_{L+1}\left([\mathbf{A}^{(1,1)}\ \mathbf{a}_k^{(2,1)}]\right) \otimes \mathcal{C}_{L+1}\left([\mathbf{B}^{(1,1)}\ \mathbf{b}_k^{(2,1)}]\right) \\ \vdots \\ \mathcal{C}_{L+1}\left([\mathbf{A}^{(1,N)}\ \mathbf{a}_k^{(2,N)}]\right) \otimes \mathcal{C}_{L+1}\left([\mathbf{B}^{(1,N)}\ \mathbf{b}_k^{(2,N)}]\right) \end{bmatrix},
\end{aligned}
\tag{S.3.4}
$$

in which $[\cdot]_{(1,2,\dots,L,k)}$ denotes the $(1,2,\dots,L,k)$th column of matrix $[\cdot]$. From properties of Vec operation and (S.3.1) it follows that for all $n \in \{1,\dots,N\}$

$$
\begin{aligned}
&\mathcal{C}_{L+1}\left([\mathbf{A}^{(1,n)}\ \mathbf{a}_k^{(2,n)}]\right) \otimes \mathcal{C}_{L+1}\left([\mathbf{B}^{(1,n)}\ \mathbf{b}_k^{(2,n)}]\right) = \\
&\qquad \operatorname{Vec}\left(\mathcal{C}_{L+1}\left([\mathbf{B}^{(1,n)}\ \mathbf{b}_k^{(2,n)}]\right)^T \mathcal{C}_{L+1}\left([\mathbf{A}^{(1,n)}\ \mathbf{a}_k^{(2,n)}]\right)\right) = \\
&\qquad\qquad \operatorname{Vec}\left(\mathcal{C}_{L+1}\left([\mathbf{B}^{(1,n)}\ \mathbf{b}_k^{(2,n)}]^T[\mathbf{A}^{(1,n)}\ \mathbf{a}_k^{(2,n)}]\right)\right).
\end{aligned}
\tag{S.3.5}
$$

Since by assumption (S.3.3), $r\left([\mathbf{B}^{(1,n)}\ \mathbf{b}_k^{(2,n)}]^T[\mathbf{A}^{(1,n)}\ \mathbf{a}_k^{(2,n)}]\right) \le L$, from (S.3.2) and (S.3.5) it follows that $\mathcal{C}_{L+1}\left([\mathbf{A}^{(1,n)}\ \mathbf{a}_k^{(2,n)}]\right) \otimes \mathcal{C}_{L+1}\left([\mathbf{B}^{(1,n)}\ \mathbf{b}_k^{(2,n)}]\right) = \mathbf{0}$ for all $n \in \{1,\dots,N\}$. Hence, by (S.3.4), the first column of the matrix $\mathbf{E}$ is equal to zero.

(ii) Assume that $\mathbf{F}^{(\text{red})}\mathbf{f} = \mathbf{0}$ for some $\mathbf{f} \in \mathbb{C}^R$. Then, by [4, eq. (3.7)], the identity $\mathbf{0} = \mathbf{F}^{(\text{red})}\mathbf{f}$ can be considered as the matrix representation of a coupled BTD [4, eq. (3.2)] in which the tensors $\mathcal{X}^{(1)},\dots,\mathcal{X}^{(N)}$ are zero and $\mathbf{C}^{(\text{red})} = \mathbf{f}^T$ or, equivalently, $\mathbf{C} = [f_1,\dots,f_1,\dots,f_R,\dots,f_R]$ (each coordinate is repeated L times). Hence, by [4, eq. (5.17)],

$$
\mathbf{E} \cdot \mathcal{R}_{L+1}^{(\text{dis})}(\mathbf{C})^T = \begin{bmatrix} \mathbf{R}_{L+1}(\mathcal{X}^{(1)}) \\ \vdots \\ \mathbf{R}_{L+1}(\mathcal{X}^{(N)}) \end{bmatrix} = \mathbf{O}. \tag{S.3.6}
$$

By [4, eq. (5.14)], $\mathbf{g}^T := \mathcal{R}_{L+1}^{(\text{dis})}(\mathbf{C})$ is an 1-by-$(C_{R+L}^{L+1} - R)$ vector with coordinates $f_{j_1}\cdots f_{j_{L+1}}$, $(j_1,\dots,j_{L+1}) \in \Omega$. Since, by assumption, the matrix $\mathbf{E}$ has full column rank, it follows from (S.3.6) that $\mathbf{g}$ is equal to the zero vector. Hence, $f_{j_1}\cdots f_{j_{L+1}} = 0$ for all $(j_1,\dots,j_{L+1}) \in \Omega$. In particular, $f_i f_j^L = 0$ for all $1 \le i < j \le L+1$, implying that the vector $\mathbf{f}$ has at most one nonzero coordinate. If such a coordinate exists, then the equation $\mathbf{F}^{(\mathrm{red})}\mathbf{f} = \mathbf{0}$ implies that $\mathbf{F}^{(\mathrm{red})}$ has a zero column which will contradict statement (i). Hence, $\mathbf{f} = \mathbf{0}$. ∎

S.4. ALS method for coupled CPD. We consider coupled Polyadic Decompositions (PDs) of a given set of tensors $\mathcal{X}^{(n)} \in \mathbb{C}^{I_n \times J_n \times K}$, $n \in \{1, \ldots, N\}$, of the following form:

$$\mathcal{X}^{(n)} = \sum_{r=1}^{R} \mathbf{a}_r^{(n)} \circ \mathbf{b}_r^{(n)} \circ \mathbf{c}_r \,, \quad n \in \{1, \ldots, N\}.$$

In addition to $\mathbf{X}_{(1)}^{(n)} = \left[\mathbf{X}^{(1\cdot\cdot,n)T}, \ldots, \mathbf{X}^{(I_n\cdot\cdot,n)T}\right]^T = \left(\mathbf{A}^{(n)} \odot \mathbf{B}^{(n)}\right)\mathbf{C}^T$ and $\mathbf{X}_{(3)}^{(n)} = \left[\mathbf{X}^{(\cdot\cdot 1,n)T}, \ldots, \mathbf{X}^{(\cdot\cdot K,n)T}\right]^T = \left(\mathbf{C} \odot \mathbf{A}^{(n)}\right)\mathbf{B}^{(n)T}$ defined in [4], the ALS method will also make use of the following matrix representation. Let the matrices $\mathbf{X}^{(\cdot j\cdot,n)} \in \mathbb{C}^{K \times I_n}$ be such that $\left(\mathbf{X}^{(\cdot j\cdot,n)}\right)_{ki} = x_{ijk}^{(n)}$, then $\mathbf{X}^{(\cdot j\cdot,n)} = \mathbf{C}D_j\left(\mathbf{B}^{(n)}\right)\mathbf{A}^{(n)T}$ and

$$\mathbb{C}^{J_n K \times I_n} \ni \mathbf{X}_{(2)}^{(n)} := \left[\mathbf{X}^{(\cdot 1\cdot,n)T}, \ldots, \mathbf{X}^{(\cdot J_n\cdot,n)T}\right]^T = \left(\mathbf{B}^{(n)} \odot \mathbf{C}\right)\mathbf{A}^{(n)T}. \quad \text{(S.4.1)}$$

Recall that we have the following overall matrix representation of the coupled PD of $\{\mathcal{X}^{(n)}\}$:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_{(1)}^{(1)} \\ \vdots \\ \mathbf{X}_{(1)}^{(N)} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^{(1)} \odot \mathbf{B}^{(1)} \\ \vdots \\ \mathbf{A}^{(N)} \odot \mathbf{B}^{(N)} \end{bmatrix} \mathbf{C}^T = \mathbf{F}\mathbf{C}^T \in \mathbb{C}^{(\sum_{n=1}^N I_n J_n) \times K}, \quad \text{(S.4.2)}$$

where

$$\mathbf{F} = \begin{bmatrix} \mathbf{A}^{(1)} \odot \mathbf{B}^{(1)} \\ \vdots \\ \mathbf{A}^{(N)} \odot \mathbf{B}^{(N)} \end{bmatrix} \in \mathbb{C}^{(\sum_{n=1}^N I_n J_n) \times R}.$$

It is well-known that

$$\left(\mathbf{A}^{(n)} \odot \mathbf{B}^{(n)}\right)^\dagger = \left(\left(\mathbf{A}^{(n)H}\mathbf{A}^{(n)}\right) * \left(\mathbf{B}^{(n)H}\mathbf{B}^{(n)}\right)\right)^{-1}\left(\mathbf{A}^{(n)} \odot \mathbf{B}^{(n)}\right)^H.$$

Hence, the conditional least squares update of $\mathbf{C}$ while $\{\mathbf{A}^{(n)}\}$ and $\{\mathbf{B}^{(n)}\}$ are fixed is

$$\mathbf{C}^T = (\mathbf{F})^\dagger \mathbf{X} = \left(\sum_{n=1}^{N}\left(\mathbf{A}^{(n)H}\mathbf{A}^{(n)}\right) * \left(\mathbf{B}^{(n)H}\mathbf{B}^{(n)}\right)\right)^{-1}\mathbf{F}^H\mathbf{X}\,. \quad \text{(S.4.3)}$$

Using

$$\left(\mathbf{A}^{(n)} \odot \mathbf{B}^{(n)}\right)^H \mathbf{X}_{(1)}^{(n)}(:,k) = \operatorname{Vecd}\left(\mathbf{B}^{(n)H}\mathbf{X}^{(\cdot\cdot k,n)T}\mathbf{A}^{(n)*}\right) = \begin{bmatrix} \mathbf{b}_1^{(n)H}\mathbf{X}^{(\cdot\cdot k,n)T}\mathbf{a}_1^{(n)*} \\ \vdots \\ \mathbf{b}_R^{(n)H}\mathbf{X}^{(\cdot\cdot k,n)T}\mathbf{a}_R^{(n)*} \end{bmatrix}$$

relation (S.4.3) can be expressed in a way that avoids the explicit construction of the tall matrix $\mathbf{F}$, as follows:

$$\mathbf{C}^T = (\mathbf{F})^\dagger \mathbf{X} = \left(\sum_{n=1}^{N} \left(\mathbf{A}^{(n)H} \mathbf{A}^{(n)} \right) * \left(\mathbf{B}^{(n)H} \mathbf{B}^{(n)} \right) \right)^{-1}$$

$$\cdot \sum_{n=1}^{N} \begin{bmatrix} \mathbf{b}_1^{(n)H} \mathbf{X}^{(\cdot\cdot 1,n)T} \mathbf{a}_1^{(n)*} & \cdots & \mathbf{b}_1^{(n)H} \mathbf{X}^{(\cdot\cdot K,n)T} \mathbf{a}_1^{(n)*} \\ \vdots & \ddots & \vdots \\ \mathbf{b}_R^{(n)H} \mathbf{X}^{(\cdot\cdot 1,n)T} \mathbf{a}_R^{(n)*} & \cdots & \mathbf{b}_R^{(n)H} \mathbf{X}^{(\cdot\cdot K,n)T} \mathbf{a}_R^{(n)*} \end{bmatrix}.$$

The conditional least squares updates of $\mathbf{A}^{(n)}$ and $\mathbf{B}^{(n)}$ are the same as in the ALS method for ordinary CPD. We summarize the ALS method for computing a coupled CPD in Algorithm 1. Observe that $\mathbf{D} = \mathbf{C}^H \mathbf{C}$ appears in both $\mathbf{D}_{\mathbf{A}^{(n)}}$ and $\mathbf{D}_{\mathbf{B}^{(n)}}$. The normalization in steps 3 and 7 fixes the scaling ambiguity. The extension to coupled CPD of M_nth-order tensors in which $M_n \geq 4$ for one or more $n \in \{1, \ldots, N\}$ is straightforward.

The ALS method can also be used to compute coupled matrix-tensor factorizations by taking the following into account. Assuming $\mathbf{X}_{(1)}^{(n)} = \mathbf{A}^{(n)} \mathbf{C}^T$, the updates of $\mathbf{B}^{(n)}$ in steps 8 and 9 in Algorithm 1 can be omitted. More precisely, we fix $\mathbf{B}^{(n)} = \mathbf{1}_R^T$, and drop normalization step 7 so that $\mathbf{A}^{(n)}$ is updated as $\mathbf{A}^{(n)} = \mathbf{X}_{(1)}^{(n)} \left(\mathbf{C}^T \right)^\dagger$.

Algorithm 1 ALS method for coupled CPD.

Initialize: $\{\mathbf{A}^{(n)}\}$, $\{\mathbf{B}^{(n)}\}$ and $\mathbf{C}$

Repeat until convergence

1. $\mathbf{D}_{\mathbf{C}} = \sum_{n=1}^{N} \begin{bmatrix} \mathbf{b}_1^{(n)H} \mathbf{X}^{(\cdot\cdot 1,n)T} \mathbf{a}_1^{(n)*} & \cdots & \mathbf{b}_1^{(n)H} \mathbf{X}^{(\cdot\cdot K,n)T} \mathbf{a}_1^{(n)*} \\ \vdots & \ddots & \vdots \\ \mathbf{b}_R^{(n)H} \mathbf{X}^{(\cdot\cdot 1,n)T} \mathbf{a}_R^{(n)*} & \cdots & \mathbf{b}_R^{(n)H} \mathbf{X}^{(\cdot\cdot K,n)T} \mathbf{a}_R^{(n)*} \end{bmatrix}.$
2. $\mathbf{C}^T = \left(\sum_{n=1}^{N} \left(\mathbf{A}^{(n)H} \mathbf{A}^{(n)} \right) * \left(\mathbf{B}^{(n)H} \mathbf{B}^{(n)} \right) \right)^{-1} \mathbf{D}_{\mathbf{C}}\,.$
3. $\mathbf{c}_r \leftarrow \frac{\mathbf{c}_r}{\|\mathbf{c}_r\|_F}\,, \quad r \in \{1, \ldots, R\}\,.$
4. $\mathbf{D} = \mathbf{C}^H \mathbf{C}$.
5. $\mathbf{D}_{\mathbf{A}^{(n)}} = \begin{bmatrix} \mathbf{c}_1^H \mathbf{X}^{(1\cdot\cdot,n)T} \mathbf{b}_1^{(n)*} & \cdots & \mathbf{c}_1^H \mathbf{X}^{(I_n\cdot\cdot,n)T} \mathbf{b}_1^{(n)*} \\ \vdots & \ddots & \vdots \\ \mathbf{c}_R^H \mathbf{X}^{(1\cdot\cdot,n)T} \mathbf{b}_R^{(n)*} & \cdots & \mathbf{c}_R^H \mathbf{X}^{(I_n\cdot\cdot,n)T} \mathbf{b}_R^{(n)*} \end{bmatrix}, n \in \{1, \ldots, N\}.$
6. $\mathbf{A}^{(n)T} = \left(\left(\mathbf{B}^{(n)H} \mathbf{B}^{(n)} \right) * \mathbf{D} \right)^{-1} \mathbf{D}_{\mathbf{A}^{(n)}}\,, \quad n \in \{1, \ldots, N\}\,.$
7. $\mathbf{a}_r^{(n)} \leftarrow \frac{\mathbf{a}_r^{(n)}}{\|\mathbf{a}_r^{(n)}\|_F}\,, \quad r \in \{1, \ldots, R\}\,, n \in \{1, \ldots, N\}\,.$
8. $\mathbf{D}_{\mathbf{B}^{(n)}} = \begin{bmatrix} \mathbf{a}_1^{(n)H} \mathbf{X}^{(\cdot 1\cdot,n)T} \mathbf{c}_1^* & \cdots & \mathbf{a}_1^{(n)H} \mathbf{X}^{(\cdot J_n\cdot,n)T} \mathbf{c}_1^* \\ \vdots & \ddots & \vdots \\ \mathbf{a}_R^{(n)H} \mathbf{X}^{(\cdot 1\cdot,n)T} \mathbf{c}_R^* & \cdots & \mathbf{a}_R^{(n)H} \mathbf{X}^{(\cdot J_n\cdot,n)T} \mathbf{c}_R^* \end{bmatrix}, n \in \{1, \ldots, N\}.$
9. $\mathbf{B}^{(n)T} = \left(\mathbf{D} * \left(\mathbf{A}^{(n)H} \mathbf{A}^{(n)} \right) \right)^{-1} \mathbf{D}_{\mathbf{B}^{(n)}}\,, \quad n \in \{1, \ldots, N\}\,.$

S.5. ALS method for coupled BTD. Consider

$$\mathcal{X}^{(n)} = \sum_{r=1}^{R} \sum_{l=1}^{L_{r,n}} \mathbf{a}_l^{(r,n)} \circ \mathbf{b}_l^{(r,n)} \circ \mathbf{c}^{(r)} = \sum_{r=1}^{R} \left(\mathbf{A}^{(r,n)} \mathbf{B}^{(r,n)T} \right) \circ \mathbf{c}^{(r)}, \qquad \text{(S.5.1)}$$

where $\mathcal{X}^{(n)} \in \mathbb{C}^{I_n \times J_n \times K}$ and $n \in \{1, \ldots, N\}$.

In addition to $\mathbf{X}_{(1)}^{(n)} = \left[\mathbf{X}^{(1 \cdot \cdot, n)T}, \ldots, \mathbf{X}^{(I_n \cdot \cdot, n)T} \right]^T = \left(\mathbf{A}^{(n)} \odot \mathbf{B}^{(n)} \right) \mathbf{C}^{(n)T}$ and $\mathbf{X}_{(3)}^{(n)} = \left[\mathbf{X}^{(\cdot \cdot 1, n)T}, \ldots, \mathbf{X}^{(\cdot \cdot K, n)T} \right]^T = \left(\mathbf{C}^{(n)} \odot \mathbf{A}^{(n)} \right) \mathbf{B}^{(n)T}$ defined in [4] we have the following analogue of (S.4.1):

$$\mathbb{C}^{J_n K \times I_n} \ni \mathbf{X}_{(2)}^{(n)} = \left[\mathbf{X}^{(\cdot 1 \cdot, n)T}, \ldots, \mathbf{X}^{(\cdot J_n \cdot, n)T} \right]^T = \left(\mathbf{B}^{(n)} \odot \mathbf{C}^{(n)} \right) \mathbf{A}^{(n)T}.$$

Expression (S.4.2) can be extended as follows:

$$\mathbf{X} = \left[\mathbf{X}_{(1)}^{(1)T}, \ldots, \mathbf{X}_{(1)}^{(N)T} \right]^T = \mathbf{F}^{(\mathrm{red})} \mathbf{C}^{(\mathrm{red})T} \in \mathbb{C}^{(\sum_{n=1}^{N} I_n J_n) \times K}, \qquad \text{(S.5.2)}$$

where $\mathbf{F}^{(\mathrm{red})} \in \mathbb{C}^{(\sum_{n=1}^{N} I_n J_n) \times R}$ is given by

$$\mathbf{F}^{(\mathrm{red})} = \begin{bmatrix} \mathrm{Vec}\left(\mathbf{B}^{(1,1)} \mathbf{A}^{(1,1)T} \right) & \cdots & \mathrm{Vec}\left(\mathbf{B}^{(R,1)} \mathbf{A}^{(R,1)T} \right) \\ \vdots & \ddots & \vdots \\ \mathrm{Vec}\left(\mathbf{B}^{(1,N)} \mathbf{A}^{(1,N)T} \right) & \cdots & \mathrm{Vec}\left(\mathbf{B}^{(R,N)} \mathbf{A}^{(R,N)T} \right) \end{bmatrix}, \qquad \text{(S.5.3)}$$

$$\mathbf{C}^{(\mathrm{red})} = \left[\begin{array}{ccc} \mathbf{c}^{(1)}, & \ldots & , \mathbf{c}^{(R)} \end{array} \right] \in \mathbb{C}^{K \times R}, \qquad \text{(S.5.4)}$$

The conditional least squares update of $\mathbf{C}^{(\mathrm{red})}$ while $\{\mathbf{A}^{(n)}\}$ and $\{\mathbf{B}^{(n)}\}$ are fixed is

$$\mathbf{C}^{(\mathrm{red})T} = \left(\mathbf{F}^{(\mathrm{red})} \right)^{\dagger} \mathbf{X},$$

where $\mathbf{F}^{(\mathrm{red})}$ is given by (S.5.3) We normalize the column vectors of $\mathbf{C}^{(\mathrm{red})}$:

$$\mathbf{c}_r^{(\mathrm{red})} \leftarrow \frac{\mathbf{c}_r^{(\mathrm{red})}}{\|\mathbf{c}_r^{(\mathrm{red})}\|_F}, \quad \forall r \in \{1, \ldots, R\}.$$

Note that

$$\begin{aligned} \mathbf{D}_{\mathbf{A}^{(n)}} &= \left(\mathbf{B}^{(n)} \odot \mathbf{C}^{(n)} \right)^H \mathbf{X}_{(2)}^{(n)} \\ &= \left[\begin{array}{ccc} \mathbf{c}^{(1)H} \mathbf{X}^{(1 \cdot \cdot, n)T} \mathbf{b}_1^{(1,n)*} & \cdots & \mathbf{c}^{(1)H} \mathbf{X}^{(I_n \cdot \cdot, n)T} \mathbf{b}_1^{(r,n)*} \\ \vdots & \ddots & \vdots \\ \mathbf{c}^{(1)H} \mathbf{X}^{(1 \cdot \cdot, n)T} \mathbf{b}_{L_{1,n}}^{(1,n)*} & \cdots & \mathbf{c}^{(1)H} \mathbf{X}^{(I_n \cdot \cdot, n)T} \mathbf{b}_{L_{1,n}}^{(1,n)*} \\ \hline & \vdots & \\ \hline \mathbf{c}^{(R)H} \mathbf{X}^{(1 \cdot \cdot, n)T} \mathbf{b}_1^{(R,n)*} & \cdots & \mathbf{c}^{(R)H} \mathbf{X}^{(I_n \cdot \cdot, n)T} \mathbf{b}_1^{(R,n)*} \\ \vdots & \ddots & \vdots \\ \mathbf{c}^{(R)H} \mathbf{X}^{(1 \cdot \cdot, n)T} \mathbf{b}_{L_{R,n}}^{(R,n)*} & \cdots & \mathbf{c}^{(R)H} \mathbf{X}^{(I_n \cdot \cdot, n)T} \mathbf{b}_{L_{R,n}}^{(R,n)*} \end{array} \right] \\ &= \begin{bmatrix} \mathbf{B}^{(1,n)*} \cdot \mathbf{d}^{(1,1,n)} & \cdots & \mathbf{B}^{(1,n)*} \cdot \mathbf{d}^{(1,I_n,n)} \\ \vdots & \ddots & \vdots \\ \mathbf{B}^{(R,n)*} \cdot \mathbf{d}^{(R,1,n)} & \cdots & \mathbf{B}^{(R,n)*} \cdot \mathbf{d}^{(R,I_n,n)} \end{bmatrix}, \end{aligned} \qquad \text{(S.5.5)}$$

where $\mathbf{d}^{(r,i,n)} = \mathbf{X}^{(i\cdot\cdot,n)}\mathbf{c}^{(r)*}$. The conditional least squares updates of $\{\mathbf{A}^{(n)}\}$ while $\{\mathbf{C}^{(n)}\}$ and $\{\mathbf{B}^{(n)}\}$ are fixed are given by

$$\mathbf{A}^{(n)T} = \left(\left(\mathbf{B}^{(n)H}\mathbf{B}^{(n)}\right) * \mathbf{D}^{(n)}\right)^{-1} \mathbf{D}_{\mathbf{A}^{(n)}}, \quad n \in \{1, \ldots, N\},$$

where $\mathbf{D}^{(n)} = \mathbf{C}^{(n)H}\mathbf{C}^{(n)}$. We normalize the matrix blocks $\{\mathbf{A}^{(r,n)}\}$ of $\mathbf{A}^{(n)}$. Let $\mathbf{A}^{(r,n)} = \mathbf{Q}^{(r,n)}\mathbf{R}^{(r,n)}$ denote the QR-factorization of $\mathbf{A}^{(r,n)}$, where $\mathbf{Q}^{(r,n)} \in \mathbb{C}^{I_n \times L_{r,n}}$ is a columnwise orthonormal matrix and $\mathbf{R}^{(r,n)} \in \mathbb{C}^{L_{r,n} \times L_{r,n}}$ is an upper triangular matrix. We set

$$\mathbf{A}^{(r,n)} \leftarrow \mathbf{Q}^{(r,n)}, \quad \forall r \in \{1, \ldots, R\}, \forall n \in \{1, \ldots, N\}.$$

The conditional least squares updates of $\{\mathbf{B}^{(n)}\}$ while $\{\mathbf{C}^{(n)}\}$ and $\{\mathbf{A}^{(n)}\}$ are fixed are given by

$$\mathbf{B}^{(n)T} = \left(\mathbf{D}^{(n)} * \left(\mathbf{A}^{(n)H}\mathbf{A}^{(n)}\right)\right)^{-1} \mathbf{D}_{\mathbf{B}^{(n)}}, \quad n \in \{1, \ldots, N\},$$

where again $\mathbf{D}^{(n)} = \mathbf{C}^{(n)H}\mathbf{C}^{(n)}$ and

$$\begin{aligned}\mathbf{D}_{\mathbf{B}^{(n)}} &= \left(\mathbf{C}^{(n)} \odot \mathbf{A}^{(n)}\right)^H \mathbf{X}^{(n)}_{(3)} \\ &= \begin{bmatrix} \mathbf{A}^{(1,n)*} \cdot \mathbf{d}^{(1,1,n)} & \cdots & \mathbf{A}^{(1,n)*} \cdot \mathbf{d}^{(1,J_n,n)} \\ \vdots & \ddots & \vdots \\ \mathbf{A}^{(R,n)*} \cdot \mathbf{d}^{(R,1,n)} & \cdots & \mathbf{A}^{(R,n)*} \cdot \mathbf{d}^{(R,J_n,n)} \end{bmatrix},\end{aligned} \tag{S.5.6}$$

in which $\mathbf{d}^{(r,j,n)} = \mathbf{X}^{(\cdot j\cdot,n)T}\mathbf{c}^{(r)*}$. Observe that $\mathbf{D}^{(n)} = \mathbf{C}^{(n)H}\mathbf{C}^{(n)}$ appears in both $\mathbf{D}_{\mathbf{A}^{(n)}}$ and $\mathbf{D}_{\mathbf{B}^{(n)}}$. The ALS method is summarized as Algorithm 2. The normalization steps 3 and 7 fix the scaling and transformation ambiguities. The extension to coupled M_nth-order tensors in which $M_n \geq 4$ for one or more $n \in \{1, \ldots, N\}$ is straightforward.

Algorithm 2 ALS method for coupled BTD

Initialize: $\{\mathbf{A}^{(n)}\}$, $\{\mathbf{B}^{(n)}\}$ and $\mathbf{C}^{(\text{red})}$

Repeat until convergence

1. Build $\mathbf{F}^{(\text{red})}$ in (S.5.3).
2. $\mathbf{C}^{(\text{red})T} = \left(\mathbf{F}^{(\text{red})}\right)^{\dagger} \mathbf{X}$.
3. $\mathbf{c}_r^{(\text{red})} \leftarrow \frac{\mathbf{c}_r^{(\text{red})}}{\|\mathbf{c}_r^{(\text{red})}\|_F}, \quad r \in \{1, \ldots, R\}$.
4. $\mathbf{D}^{(n)} = \mathbf{C}^{(n)H}\mathbf{C}^{(n)}, \quad n \in \{1, \ldots, N\}$.
5. Build $\mathbf{D}_{\mathbf{A}^{(n)}}$ in (S.5.5) for every $n \in \{1, \ldots, N\}$.
6. $\mathbf{A}^{(r,n)} = \mathbf{Q}^{(r,n)}\mathbf{R}^{(r,n)}, \quad r \in \{1, \ldots, R\}, n \in \{1, \ldots, N\}$.
7. $\mathbf{A}^{(r,n)} \leftarrow \mathbf{Q}^{(r,n)}, \quad r \in \{1, \ldots, R\}, n \in \{1, \ldots, N\}$.
8. Build $\mathbf{D}_{\mathbf{B}^{(n)}}$ in (S.5.6) for every $n \in \{1, \ldots, N\}$.
9. $\mathbf{B}^{(n)T} = \left(\mathbf{D}^{(n)} * \left(\mathbf{A}^{(n)H}\mathbf{A}^{(n)}\right)\right)^{-1} \mathbf{D}_{\mathbf{B}^{(n)}}, \quad n \in \{1, \ldots, N\}$.

S.5.1. Proof of Proposition 5.2. Consider the PD

$$\mathbb{C}^{I\times J\times K} \ni \mathcal{X} = \sum_{r=1}^{R}\sum_{l=1}^{L_r} \mathbf{a}_l^{(r)} \circ \mathbf{b}_l^{(r)} \circ \mathbf{c}^{(r)} = \sum_{r=1}^{R}\left(\mathbf{A}^{(r)}\mathbf{B}^{(r)T}\right)\circ \mathbf{c}^{(r)} . \tag{S.5.7}$$

Assume that $\mathbf{C}^{(\mathrm{red})} = \left[\mathbf{c}^{(1)},\dots,\mathbf{c}^{(R)}\right] \in \mathbb{C}^{K\times R}$ is known. Let S denote a subset of $\{1,\dots,R\}$ and let $S^c = \{1,\dots,R\}\setminus S$ denote the complementary set. Stack the columns of $\mathbf{C}^{(\mathrm{red})}$ with index in S in $\mathbf{C}^{(S)} \in \mathbb{C}^{K\times \mathrm{card}(S)}$ and stack the columns of $\mathbf{C}^{(\mathrm{red})}$ with index in S^c in $\mathbf{C}^{(S^c)} \in \mathbb{C}^{K\times (R-\mathrm{card}(S))}$. Let the elements of S be indexed by $\sigma(1),\dots,\sigma(\mathrm{card}\,(S))$ and let the elements of S^c be indexed by $\mu(1),\dots,\mu(\mathrm{card}\,(S^c))$. The corresponding partitions of $\mathbf{A}^{(n)}$ and $\mathbf{B}^{(n)}$ are then given by

$$\begin{aligned}
\mathbf{A}^{(S)} &= \left[\mathbf{A}^{(\sigma(1))},\dots,\mathbf{A}^{(\sigma(\mathrm{card}(S)))}\right] \in \mathbb{C}^{I\times\left(\sum_{p\in S}L_p\right)},\\
\mathbf{A}^{(S^c)} &= \left[\mathbf{A}^{(\mu(1))},\dots,\mathbf{A}^{(\mu(\mathrm{card}(S^c)))}\right] \in \mathbb{C}^{I\times\left(\sum_{p\in S^c}L_p\right)},\\
\mathbf{B}^{(S)} &= \left[\mathbf{B}^{(\sigma(1))},\dots,\mathbf{B}^{(\sigma(\mathrm{card}(S)))}\right] \in \mathbb{C}^{J\times\left(\sum_{p\in S}L_p\right)},\\
\mathbf{B}^{(S^c)} &= \left[\mathbf{B}^{(\mu(1))},\dots,\mathbf{B}^{(\mu(\mathrm{card}(S^c)))}\right] \in \mathbb{C}^{J\times\left(\sum_{p\in S^c}L_p\right)}.
\end{aligned}$$

If there exists a subset $S\subseteq\{1,\dots,R\}$ with $0\le \mathrm{card}\,(S)\le r_{\mathbf{C}^{(\mathrm{red})}}$ such that[1]

$$\begin{cases}
\mathbf{C}^{(S)} \text{ has full column rank } \left(\text{i.e., } r_{\mathbf{C}^{(S)}} = S\right),\\
\mathbf{B}^{(S^c)} \text{ has full column rank } \left(\text{i.e., } r_{\mathbf{B}^{(S^c)}} = \sum_{p\in S^c}L_p\right),\\
r\left(\left[\left(\mathbf{P}_{\mathbf{C}^{(S)}}\widetilde{\mathbf{C}}^{(S^c)}\right)\odot\mathbf{A}^{(S^c)},\left(\mathbf{P}_{\mathbf{C}^{(S)}}\mathbf{c}_{\mu(r)}^{(S^c)}\right)\otimes\mathbf{I}_I\right]\right) = I+\sum_{p\in S^c}L_p - L_r,\ \forall r\in S^c,
\end{cases} \tag{S.5.8}$$

where $\widetilde{\mathbf{C}}^{(S^c)} = \left[\mathbf{1}_{L_{\mu(1)}}^T\otimes\mathbf{c}_{\mu(1)}^{(S^c)},\dots,\mathbf{1}_{L_{\mu(\mathrm{card}(S^c))}}^T\otimes\mathbf{c}_{\mu(\mathrm{card}(S^c))}^{(S^c)}\right]$,then the decomposition of $\mathcal{X}$ in (S.5.7) is unique.

Proof. The result is a variant of [3, Theorem 4.8] to the case where $\mathbf{C}$ contains collinear columns. W.l.o.g. we assume that $\mathbf{C}^{(\mathrm{red})}(1:\mathrm{card}\,(S),1:\mathrm{card}\,(S))$ is nonsingular, i.e., we set $\mathbf{C}^{(S)} = \mathbf{C}^{(\mathrm{red})}(:,1:\mathrm{card}\,(S))$. Observe that

$$\mathbf{P}_{\mathbf{C}^{(S)}}\mathbf{C}^{(\mathrm{red})} = \mathbf{P}_{\mathbf{C}^{(S)}}\left[\mathbf{C}^{(S)},\mathbf{C}^{(S^c)}\right] = \left[\mathbf{0}_{K,\mathrm{card}(S)},\mathbf{P}_{\mathbf{C}^{(S)}}\mathbf{C}^{(S^c)}\right].$$

We work in two steps. First we compute $\mathbf{A}^{(S^c)}$ and $\mathbf{B}^{(S^c)}$, later we also compute $\mathbf{A}^{(S)}$ and $\mathbf{B}^{(S)}$.

Step 1. Compute $\mathbf{Y}_{(1)} = \mathbf{X}_{(1)}\mathbf{P}_{\mathbf{C}^{(S)}}^T$, then

$$\mathbf{Y}_{(1)} = \left[\mathrm{Vec}\left(\mathbf{B}^{(\mu(1))}\mathbf{A}^{(\mu(1))T}\right),\dots,\mathrm{Vec}\left(\mathbf{B}^{(\mu(\mathrm{card}(S^c)))}\mathbf{A}^{(\mu(\mathrm{card}(S^c)))T}\right)\right]\mathbf{C}^{(S^c)T}\mathbf{P}_{\mathbf{C}^{(S)}}^T.$$

Denote $\widetilde{\mathbf{D}}^{(S^c)} = \mathbf{P}_{\mathbf{C}^{(S)}}\widetilde{\mathbf{C}}^{(S^c)}$. The tensor $\mathcal{Y}$ also has matrix representation

$$\mathbf{Y}_{(3)} = \left(\widetilde{\mathbf{D}}^{(S^c)}\odot\mathbf{A}^{(S^c)}\right)\mathbf{B}^{(S^c)T}.$$

[1]The last condition means that $\mathbf{M}_r = \left[\left(\mathbf{P}_{\mathbf{C}^{(S)}}\widetilde{\mathbf{C}}^{(S^c)}\right)\odot\mathbf{A}^{(S^c)},\left(\mathbf{P}_{\mathbf{C}^{(S)}}\mathbf{c}_{\mu(r)}^{(S^c)}\right)\otimes\mathbf{I}_I\right]$ has an L_r-dimensional kernel for every $r\in S^c$, which is minimal since for every $p\in\{1,\dots,L_r\}$ the vector $[\mathbf{n}_r^T,\mathbf{a}_p^{(\mu(r))T}]^T\in\ker(\mathbf{M}_r)$ for some $\mathbf{n}_r\in\mathbb{C}^{\mathrm{card}(S^c)}$.

By assumption, $r\left(\left[\widetilde{\mathbf{D}}^{(S^c)} \odot \mathbf{A}^{(S^c)}, \left(\mathbf{P}_{\mathbf{C}^{(S)}}\mathbf{c}^{(S^c)}_{\mu(r)}\right) \otimes \mathbf{I}_I\right]\right) = I + \sum_{p \in S^c} L_p - L_r$, $\forall r \in S^c$. Note that $\left(\mathbf{P}_{\mathbf{C}^{(S)}}\mathbf{c}^{(S^c)}_{\mu(r)}\right) \otimes \mathbf{a}^{(\mu(r))}_p \in \text{range}\left(\left(\mathbf{P}_{\mathbf{C}^{(S)}}\mathbf{c}^{(S^c)}_{\mu(r)}\right) \otimes \mathbf{I}_I\right)$, $\forall p \in \{1, \ldots, L_{\mu(r)}\}$. This implies that $r\left(\widetilde{\mathbf{D}}^{(S^c)} \odot \mathbf{A}^{(S^c)}\right) = \sum_{p \in S^c} L_p$, i.e., $\widetilde{\mathbf{D}}^{(S^c)} \odot \mathbf{A}^{(S^c)}$ has full column rank. By assumption, $\mathbf{B}^{(S^c)}$ also has full column rank. Let $\mathbf{Y}_{(3)} = \mathbf{U}\Sigma\mathbf{V}^H$ denote the compact SVD of $\mathbf{Y}_{(3)}$ in which $\mathbf{U} \in \mathbb{C}^{KI \times (\sum_{p \in S^c} L_p)}$, then there exists a nonsingular matrix $\mathbf{M} \in \mathbb{C}^{(\sum_{p \in S^c} L_p) \times (\sum_{p \in S^c} L_p)}$ such that

$$\mathbf{U}\mathbf{M} = \widetilde{\mathbf{D}}^{(S^c)} \odot \mathbf{A}^{(S^c)}. \tag{S.5.9}$$

Partition $\mathbf{M}$ as follows

$$\mathbf{M} = \left[\mathbf{M}^{(\mu(1))}, \ldots, \mathbf{M}^{(\mu(S^c))}\right], \quad \mathbf{M}^{(\mu(r))} \in \mathbb{C}^{(\sum_{p \in S^c} L_p) \times L_r}, \quad r \in S^c,$$

then (S.5.9) is equivalent to

$$\mathbf{G}^{(\mu(r), S^c)} \begin{bmatrix} \mathbf{M}^{(\mu(r))} \\ \mathbf{A}^{(\mu(r))} \end{bmatrix} = \mathbf{0}_{KI, L_r}, \quad r \in S^c, \tag{S.5.10}$$

in which $\mathbf{G}^{(\mu(r), S^c)} \in \mathbb{C}^{KI \times (I + \sum_{p \in S^c} L_p)}$ is given by

$$\mathbf{G}^{(\mu(r), S^c)} = \left[\mathbf{U}, -\left(\mathbf{P}_{\mathbf{C}^{(S)}}\mathbf{c}^{(S^c)}_{\mu(r)}\right) \otimes \mathbf{I}_I\right], \quad r \in S^c. \tag{S.5.11}$$

The assumption that $\mathbf{G}^{(\mu(r), S^c)}$ has rank $I + \sum_{p \in S^c} L_p - L_r$, $\forall r \in S^c$, implies that the matrices $\mathbf{M}^{(\mu(r))}$ and $\mathbf{A}^{(\mu(r))}$ follow from the kernel of (S.5.10), $\forall r \in S^c$. The solution is known up to right multiplication by a nonsingular $(L_r \times L_r)$ matrix, which is an intrinsic BTD indeterminacy. Next we find $\mathbf{B}^{(S^c)}$ from

$$\begin{aligned}
\mathbf{B}^{(S^c)T} &= \left(\widetilde{\mathbf{D}}^{(S^c)} \odot \mathbf{A}^{(S^c)}\right)^{\dagger} \mathbf{Y}_{(3)} \\
&= \left(\left(\widetilde{\mathbf{C}}^{(S^c)H}\widetilde{\mathbf{D}}^{(S^c)}\right) * \left(\mathbf{A}^{(S^c)H}\mathbf{A}^{(S^c)}\right)\right)^{-1} \left(\widetilde{\mathbf{D}}^{(S^c)} \odot \mathbf{A}^{(S^c)}\right)^{H} \mathbf{Y}_{(3)}.
\end{aligned}$$

Step 2. Now that $\mathbf{A}^{(S^c)}$ and $\mathbf{B}^{(S^c)}$ are known, we compute

$$\begin{aligned}
\mathbf{Q}_{(1)} &= \mathbf{Y}_{(1)} - \left[\text{Vec}\left(\mathbf{B}^{(\mu(1))}\mathbf{A}^{(\mu(1))T}\right), \ldots, \text{Vec}\left(\mathbf{B}^{(\mu(\text{card}(S^c)))}\mathbf{A}^{(\mu(\text{card}(S^c)))T}\right)\right] \mathbf{C}^{(S^c)T} \\
&= \left[\text{Vec}\left(\mathbf{B}^{(\sigma(1))}\mathbf{A}^{(\sigma(1))T}\right), \ldots, \text{Vec}\left(\mathbf{B}^{(\sigma(\text{card}(S)))}\mathbf{A}^{(\sigma(\text{card}(S)))T}\right)\right] \mathbf{C}^{(S)T}.
\end{aligned}$$

Recall that the matrix $\mathbf{C}^{(S)}$ is assumed to have full column rank. Hence, we can compute

$$\mathbf{H} = \mathbf{Q}_{(1)} \left(\mathbf{C}^{(S)T}\right)^{\dagger}.$$

The remaining unknowns $\mathbf{A}^{(S)}$ and $\mathbf{B}^{(S)}$ now follow from the rank-L_r approximation problems

$$\min_{\mathbf{A}^{(\sigma(r))}, \mathbf{B}^{(\sigma(r))}} \left\|\mathbf{h}_{\sigma(r)} - \text{Vec}\left(\mathbf{B}^{(\sigma(r))}\mathbf{A}^{(\sigma(r))T}\right)\right\|_F^2, \quad r \in S.$$

□

W.r.t. (5.3) in [4] we also note that, similarly to (S.5.6),

$$\begin{aligned}\mathbf{D}_{\mathbf{B}^{(S^c)}} &= \left(\widetilde{\mathbf{D}}^{(S^c)} \odot \mathbf{A}^{(S^c)}\right)^H \mathbf{Y}_{(3)} \\ &= \begin{bmatrix} \mathbf{A}^{(\mu(1))*} \cdot \mathbf{f}^{(1,1)} & \cdots & \mathbf{A}^{(\mu(1))*} \cdot \mathbf{f}^{(1,J)} \\ \vdots & \ddots & \vdots \\ \mathbf{A}^{(\mu(\mathrm{card}(S^c)))*} \cdot \mathbf{f}^{(\mathrm{card}(S^c),1)} & \cdots & \mathbf{A}^{(\mu(\mathrm{card}(S^c)))*} \cdot \mathbf{f}^{(\mathrm{card}(S^c),J)} \end{bmatrix},\end{aligned}$$

where $\mathbf{f}^{(r,j)} = \mathbf{Y}^{(\cdot j \cdot)T} \mathbf{P}_{\mathbf{C}^{(S)}} \mathbf{c}_{\mu(r)}^{(S^c)*}$.

REFERENCES

[1] L. De Lathauwer, *A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 642–666.

[2] R. A. Horn and C. Johnson, *Matrix analysis*, Cambridge University Press, Cambridge, 1990.

[3] M. Sørensen and L. De Lathauwer, *New uniqueness conditions for the canonical polyadic decomposition of third-order tensors*, Internal Report 13-05, ESAT-STADIUS, KU Leuven, Belgium.

[4] M. Sørensen, I. Domanov and L. De Lathauwer, *Coupled canonical polyadic decompositions and (Coupled) Decompositions in Multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ terms — Part II: Algorithms*, Internal Report 13-144, ESAT-STADIUS, KU Leuven (Leuven, Belgium), 2013.

[5] M. Sørensen and L. De Lathauwer, *Multidimensional ESPRIT: A coupled canonical polyadic decomposition approach*, Proc. of the 8th IEEE SAM Workshop, Jun. 22-25, A Coruna, Spain, 2014.