# Tensor Decompositions for Signal Processing: Theory, Advances, and Applications

Neriman Tokcan[01a], Shakir Showkat Sofi[1b,c], Van Tien Pham[1d], Clémence Prévost[1e], Sofiane Kharbech[1g], Baptiste Magnier[h,i], Thanh Phuong Nguyen[f], Yassine Zniyed[d], Lieven De Lathauwer[b,c]

[a]*Department of Mathematics, University of Massachusetts Boston, Boston, USA*
[b]*Dept. Electrical Engineering (ESAT), KU Leuven, Leuven, Belgium*
[c]*Group Science, Engineering and Technology, KU Leuven Kulak, Kortrijk, Belgium*
[d]*University of Toulon, Aix-Marseille University, CNRS, LIS UMR 7020, Toulon, France*
[e]*Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, , France*
[f]*University of Côte d'Azur, CNRS, I3S UMR 7271, Nice, France*
[g]*Mathematical and Algorithmic Sciences Lab, Huawei France R&D, Paris, France*
[h]*Euromov Digital Health in Motion, Montpellier, IMT Mines Ales, Ales, France*
[i]*Service de Médecine Nucléaire, Centre Hospitalier Universitaire de Nîmes, Université de Montpellier, Nîmes, France*

**Abstract**

In the era of big data, rapid advancements in technology and data collection methods have led to the generation and accessibility of vast amounts of multi-modal, high-dimensional data across a diverse range of disciplines. Tensor methods have emerged as essential tools in signal processing, providing powerful frameworks to model and analyze such complex data effectively. This survey offers a comprehensive overview of tensor factorization techniques and their applications in key areas. We explore their role in remote sensing, focusing on tensor-based methods for analyzing hyperspectral and multispectral images, tackling challenges such as recovering super-resolution images and addressing spectral unmixing. In wireless communication, we examine tensor methods used for signal modulation in unsourced massive random access communication, which achieve strong performance in multi-antenna channel and signal modeling. We also discuss tensor applications in network compression, where they reduce the computational demands of deep neural networks, making them more feasible for edge devices. Additionally, we highlight the use of tensor methods in high-dimensional missing data completion problems, showcasing their versatility across various domains. Furthermore, we explore applications in image analysis and computer vision, where tensors are effectively utilized for motion and object tracking, 3D modeling, analysis of satellite imaging, and medical imaging. By bridging theoretical advancements with practical applications, this survey aims to guide researchers in leveraging tensor methods to tackle emerging challenges in signal processing.

*Keywords:* tensor decomposition, low rank approximation, block-term decomposition, network compression, wireless communication, remote sensing, tensor completion, tensor-based modulation, massive access, nonorthogonal multiple access, multiple antenna techniques, subspace constraints

[1]These authors contributed equally to this work as co-first authors

[0]Corresponding author-email: neriman.tokcan@umb.edu

## 1. Introduction

In the era of big data, we generate and access massive amounts of multi-modal, multi-way data across various fields, driven by multiple factors. Wireless communication plays a crucial role, as the proliferation of connected devices—including smartphones, IoT devices, and sensors—continuously generates and transmits streams of information [1]. Electronic devices such as smartphones and social networking platforms create data that reflects human behaviors and preferences, while wearable health technologies monitor physiological parameters in real time [2]. Smart cities utilize extensive sensor networks to collect data on urban dynamics, enabling data-driven decision-making for improved infrastructure and services. Additionally, environmental monitoring employs sensors and satellites to capture comprehensive data on climate conditions, pollution levels, and natural resources, supporting applications in weather forecasting and resource management [3, 4]. These advancements underscore the growing amount of high-dimensional, multi-way data generated across diverse fields.

The rapid increase in data generation brings with it a substantial challenge: finding effective ways to analyze and interpret the vast and complex information embedded in these datasets. As multi-modal and high-dimensional data continue to grow, so does the demand for advanced analytical methods that can handle the intricate relationships within these data structures and uncover valuable insights. Over recent decades, tensor methods have advanced significantly, becoming pivotal in the analysis of complex, multi-dimensional datasets [5]. Matrix-based methods often transform multi-dimensional data into matrices through unfolding or slicing, which introduces several challenges. In unfolding, different modes or variations are merged into a single, long dimension, while slicing extracts submatrices along specific dimensions. While these reshaping techniques simplify data for certain machine learning methods, they may overlook important relationships between dimensions and fail to capture complex inter-dependencies or correlations across modalities. On the other hand, a tensor representation can preserve the multi-way information and provide a powerful framework for modeling intricate relationships within data, enabling sophisticated analyses that extend beyond the capabilities of traditional two-dimensional methods. Additionally, unlike matrix factorizations, tensor factorizations achieve uniqueness under milder conditions [5].

Besides offering a natural representation for multi-way phenomena, tensor-based methods offer a more powerful and versatile framework than matrix-based approaches across various applications. These areas include "blind" data analysis (enabled by milder conditions for uniqueness [6]), data compression (through $d$-fold compression) and denoising (via mirroring compression), data completion (where low-rank structure becomes relatively stronger as the order increases), modeling of nonlinear structures (such as bilinear and trilinear, which are polynomial-like), and "blind" pattern recognition [7], among others. Tensor decomposition factorizes a higher-order tensor into more manageable lower-order tensors, helping to overcome the curse of dimensionality (see, e.g., [8, 9, 10, 11, 12]). It is worth noting that the formidable compression pos-

sibilities have revolutionized the field of scientific computing, with compact tensor network representations of functions in many variables as key enabler. This revolution happened about at the same time in the study of quantum many-body systems, with "Matrix Product States" in a similar role as "Tensor Trains" (see, e.g., [13, 14, 15, 16]). Proper variants of the techniques are also valuable for signal processing, although, in our setting, the tensor order is typically less high. These advancements have made tensors essential tools in diverse fields, from image processing and remote sensing to wireless communications and genomics. In remote sensing, the high-dimensional data captured by satellite imaging systems can be organized as tensors, where dimensions might represent spatial coordinates, spectral bands, and time [17]. Tensor decompositions can then be applied to detect meaningful patterns, such as identifying land use changes or monitoring environmental impacts over time. In the realm of multi-omics research, where the goal is to comprehend sample-specific features across diverse omics platforms, data can be effectively represented using a 3-way tensor structured as samples $\times$ features $\times$ omics platforms [18]. In wireless communications and sensing, tensor-based methods allow efficient data handling through dimensionality reduction while preserving essential information [19]. By structuring data from various sources, such as users, locations, and time, tensors enable advanced analysis for resource allocation and interference management, optimizing network performance.

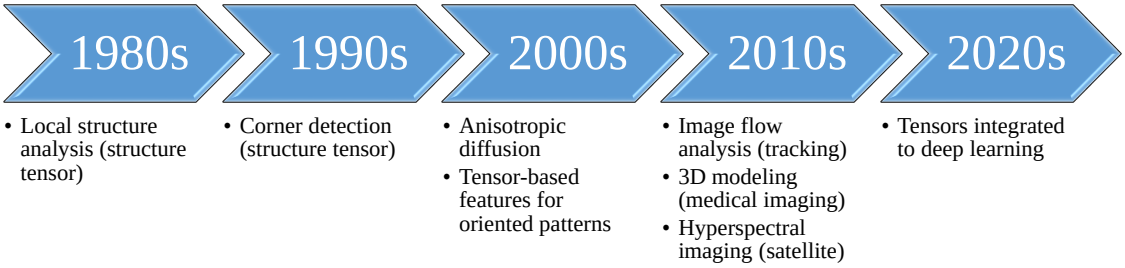



Figure 1: Timeline of tensor applications in image processing, illustrating key advancements and their applications from 1980 to today (2024) in 10-year intervals, highlighting significant milestones.

Tensors have also proven to be a powerful tool in image processing, capturing multi-way interactions and offering a robust framework for unifying the analysis of spatial, temporal, and spectral data, as illustrated by the timeline presented in Fig. 1. In this context, DiZenzo's pioneering work extended edge detection to multi-channel or color images by introducing a multi-spectral tensor, constructed from the combined first derivatives of each image channel [20] (an example is available in Fig. 2(e)). Thereafter, in the 1990s, the structure tensor was introduced as a way to capture the orientation and intensity of changes within an image region. This is achieved by computing the Cartesian product of the gradient vector with itself at each point of the image. Spatial averaging of this tensor –usually with a Gaussian filter– leads to the structure tensor. By analyzing its eigenvalues –and illustrated in Fig. 2(b)-(c), the structure tensor provides insights into the local image structure, enabling early applications that relied on gradient calculations to

extract key features such as corners and edges by detecting intensity variations [21, 22, 23, 24]. In the 2000s, tensor fields have been developed to describe the behavior of diffusion processes in the context of image denoising (diffusion tensors, see Fig. 2(f)), as in [25, 26, 27]. At the same time, tensor-based features –at single or multiple scales– have been used to extract meaningful information from images, emphasizing the orientation and structure of patterns and textures [28, 29]. By the 2010s, the use of second and third-order tensors significantly expanded their applications to increasingly complex datasets. This marked a major advancement in image analysis and computer vision, enabling tensors to be effectively utilized in areas such as motion/object tracking [30], 3D modeling [31, 32], satellite imaging [33], and medical imaging [34, 35].

While tensor methods have proven effective in a wide range of applications, as mentioned above, real-world datasets often present significant challenges. These datasets are frequently noisy and incomplete due to factors such as malfunctioning sensors, restricted access, and intentional partial sampling aimed at minimizing time or memory constraints [36, 37, 38]. These issues can complicate the analysis and lead to biased or incomplete insights if not properly addressed. In genomics, a high prevalence of zero values is primarily caused by the "dropout" effect, where certain genes expressed in a cell are not detected

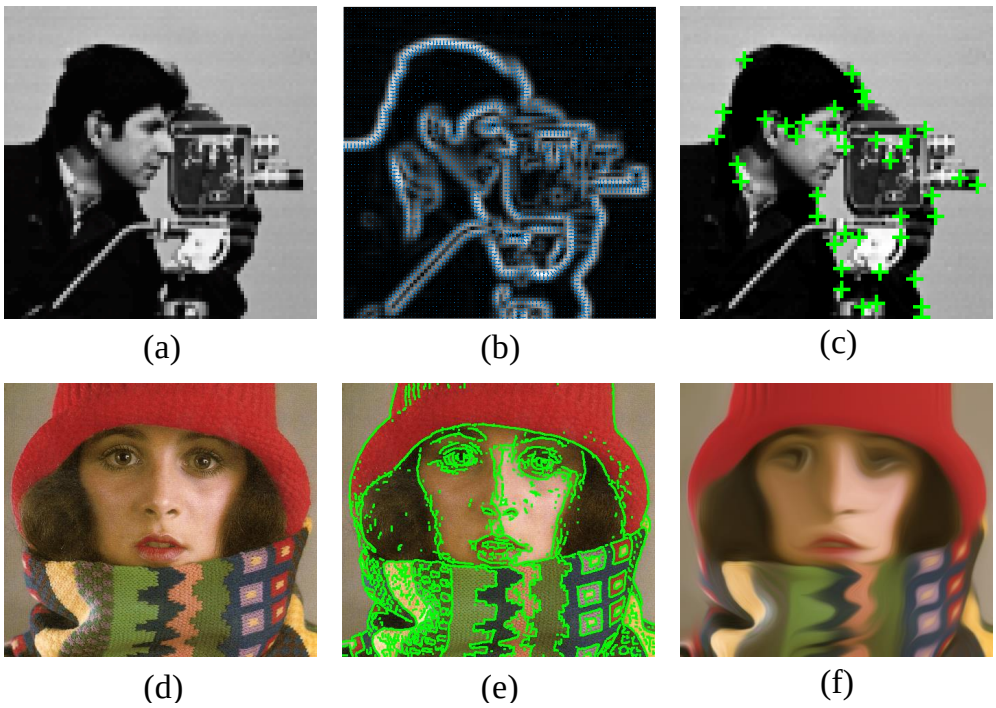



Figure 2: Comparative visualization of tensor-based techniques for image analysis. At the top, feature extraction using the structure tensor on a grayscale image is shown: (a) original 90×90 pixel image, (b) gradient norm and direction vector ponderated the gradient norm (blue arrows), (c) 40 detected corners [24] after non-maxima process (represented by the green crosses). At the bottom, results obtained with multi-spectral computation: (d) original color image (3 channels) of size 400×456 pixels, (e) detected contours [20] in green (dilated with a 3×3 mask for better visualization), and (f) anisotropic diffusion [27].

during sequencing [39]. Additionally, omics data can suffer from missing information due to incomplete measurements or absent experiments, such as missing drug-response data [40, 41]. Insights derived from incomplete data can be biased, leading to invalid conclusions [36]. Therefore, handling missing values appropriately is crucial.

This paper aims to demonstrate how tensor methods address the challenges of handling complex, high-dimensional, and often incomplete data, while exploring their diverse applications in signal processing and beyond. In Section 1.1, we first provide background information on tensors and tensor decompositions, establishing the foundational concepts necessary for understanding their applications in various domains. Building on this foundation, we examine their applications across a wide range of domains, highlighting their versatility and significance in addressing various challenges. Section 2 focuses on remote sensing, examining tensor-based methods for analyzing hyperspectral and multispectral images, tackling joint problems aimed at recovering super-resolution images from these sources, and addressing spectral unmixing. In Section 3 , we discuss tensor methods in wireless communication, including tensor-based signal modulation frameworks for unsourced massive random access communication. Section 4 provides an overview of deep neural network (DNN) compression, discussing both traditional methods and low-rank tensor decomposition-based approaches. Following this, Section 5 addresses missing data problems, imputation approaches, and tensor-based methods for imputing higher-order datasets. Finally, Section 6 explores applications in other areas and examines future challenges associated with tensor methods.

### *1.1. Background and notations*

Given that we will explore research from various fields, it is essential to provide a comprehensive background section. This section will first introduce key tensor notations, followed by an explanation of the tensor inner product and norms. We will also cover fundamental tensor operations, including vectorization and matricization techniques. Furthermore, we will discuss tensor-matrix multiplication and present several matrix operations that frequently appear in tensor-based algorithms, which may prove useful in the context of our discussion in Section 1.1.1. Then we will discuss the concept of rank in tensor and tensor factorization methods.

#### *1.1.1. Notations, tensor operations*

Tensors are multi-way arrays that extend matrices into multiple dimensions. The number of dimensions, also called modes or ways, defines the order of a tensor. An order-1 tensor is a vector, and an order-2 tensor corresponds to a matrix. Tensors with orders above 2 are categorized as higher-order tensors.

Throughout this paper, we denote scalars, vectors, matrices, and tensors by lower case (e.g., $\mathbf{x} \in \mathbb{R}^p$), upper case letters (e.g., $\mathbf{X} \in \mathbb{R}^{p \times q}$), and capital calligraphic letters (e.g., $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_d}$), respectively. We

denote the $(i_1, i_2, \dots i_d)$-th entry of an order-$d$ tensor $\mathcal{X}$ as $\mathcal{X}_{i_1,i_2,\dots i_d}$. The inner product of two same-sized tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_d}$ is defined as

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_d=1}^{I_d} \mathcal{X}_{i_1,i_2,\dots i_d} \mathcal{Y}_{i_1,i_2,\dots i_d}. \tag{1}$$

Using this inner product, the *Frobenius norm* of a tensor can be given as:

$$||\mathcal{X}||_F = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_d=1}^{I_d} \mathcal{X}^2_{i_1,i_2,\dots i_d}}. \tag{2}$$

In a matrix, data is organized along rows and columns, but in a tensor, the equivalent structures are called fibers. A mode-n fiber is obtained by fixing all indices of the tensor except for one, denoted as $\mathcal{X}_{i_1,\dots,i_{n-1}:i_{n+1},\dots i_d}$. In matrices, the row and column spaces have the same dimensionality, but this does not necessarily hold for tensor fibers. Many algorithms make use of matrix or vector forms of tensors. *Vectorization* transforms a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_d}$ into a vector, denoted as $\text{vec}(\mathcal{X}) \in \mathbb{R}^{(\prod_{j=1}^{d} I_j)}$, where the specific order of elements is arbitrary as long as it remains consistent throughout. Similarly, *matricization*—also called *unfolding* or *flattening*—reorganizes the tensor elements into a matrix form. For an order-$d$ tensor $\mathcal{X}$, matricization can be performed in $d$ different ways. The mode-$n$ unfolding of $\mathcal{X}$, written as $\mathcal{X}_{(n)}$, produces a matrix in $\mathbb{R}^{(I_n \times \prod_{j \neq n} I_j)}$, aligning mode-$n$ fibers as rows. There are also alternative unfolding methods, such as $\mathcal{X}_{[1,\dots,n;n+1,\dots,d]} \in \mathbb{R}^{I_1 \times \dots \times I_n \times I_{n+1} \times \dots \times I_d}$, where the semicolon in the subscript simply separates the indices, indicating how dimensions are grouped in the unfolding. Alternatively, fixing all but two of the tensor's indices provides a way to obtain matrix slices. A matrix slice along modes $m$ and $n$ can be written as $\mathcal{X}_{i_1,\dots,i_{m-1},:,i_{m+1},\dots,i_{n-1},:,i_{n+1},\dots i_d} \in \mathbb{R}^{I_m \times I_n}$. A tensor can be multiplied by a matrix along a designated mode. The *mode-n multiplication* of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_{n-1} \times I_n \times \dots \times I_d}$ with a matrix $\mathbf{A} \in \mathbb{R}^{J \times I_n}$ is defined as follows:

$$(\mathcal{X} \times_n \mathbf{A})_{i_1,\dots,i_{n-1},j,i_{n+1},\dots i_d} = \sum_{i_n=1}^{I_n} \mathcal{X}_{i_1,\dots,i_{n-1},i_n,i_{n+1},\dots i_d} \mathbf{A}_{j,i_n} \in \mathbb{R}^{I_1 \times \dots \times I_{n-1} \times J \times \dots \times I_d}. \tag{3}$$

This operation is especially useful when we need to expand a compressed tensor along a particular mode, such as in the case of expanding the core tensor during Tucker decomposition, as outlined in Section 1.1.2. Additionally, Eq. (3) can be reformulated using matrix multiplication by leveraging unfoldings, yielding the following equivalent expression:

$$\mathcal{X} \times_n \mathbf{A} = \mathcal{Y} \iff \mathbf{A}\mathcal{X}_{(n)} = \mathcal{Y}_{(n)}. \tag{4}$$

When multiplying a tensor by multiple matrices, the sequence of operations does not impact the outcome, provided each matrix is associated with the appropriate mode. Specifically, we have the following equalities:

$$\mathcal{X} \times_n \mathbf{A} \times_m \mathbf{B} = \mathcal{X} \times_m \mathbf{B} \times_n \mathbf{A} \quad (n \neq m), \quad \text{and} \quad \mathcal{X} \times_n \mathbf{A} \times_n \mathbf{B} = \mathcal{X} \times_n (\mathbf{B}\mathbf{A}). \tag{5}$$

Since tensor algorithms often make use of unfoldings, slices, and fibers, they typically include various vector, matrix operations. Below, we summarize some of the most commonly employed operations. The *outer product* of vectors $\mathbf{u} \in \mathbb{R}^p$ and $\mathbf{v} \in \mathbb{R}^q$ is given as $\mathbf{u} \otimes \mathbf{v} = \mathbf{u}\mathbf{v}^T \in \mathbb{R}^{p \times q}$. For matrices $\mathbf{A} \in \mathbb{R}^{I_1 \times J_1}$ and $\mathbf{B} \in \mathbb{R}^{I_2 \times J_2}$, the *Kronecker product* is defined as $\mathbf{A} \otimes \mathbf{B} = [\mathbf{a_1} \otimes \mathbf{b_1}, \mathbf{a_1} \otimes \mathbf{b_2}, \ldots, \mathbf{a}_{J_1} \otimes \mathbf{b}_{J_2}] \in \mathbb{R}^{(I_1 I_2) \times (J_1 J_2)}$. If $J_1 = J_2 = J$, we can perform a "matching column-wise" *Kronecker product*, known as the *Khatri-Rao product*, which is expressed as $\mathbf{A} \odot \mathbf{B} = [\mathbf{a_1} \otimes \mathbf{b_1}, \mathbf{a_2} \otimes \mathbf{b_2}, \ldots, \mathbf{a_J} \otimes \mathbf{b_J}] \in \mathbb{R}^{(I_1 I_2) \times J}$.

In the case where $I_1 = I_2 = I$ and $J_1 = J_2 = J$, we can perform the element-wise matrix product, referred to as the *Hadamard product*, denoted by $\mathbf{A} * \mathbf{B} \in \mathbb{R}^{I \times J}$, where $(\mathcal{A} * \mathcal{B})_{i,j} = \mathbf{A}_{i,j}\mathbf{B}_{i,j}$.

### *1.1.2. Tensor factorization, tensor norms*

The rank of a matrix is defined as the dimension of its column space or row space, and these dimensions are equal. However, fibers of a tensor can have different dimensions. The concept of rank does not generalize directly to higher-order tensors.

However, the notion of a rank-one matrix can be extended in a straightforward way: a matrix $\mathbf{X} \in \mathbb{R}^{p \times q}$ is rank one if it can be written as the outer product of two vectors, $\mathbf{X} = \mathbf{u} \otimes \mathbf{v}$. Similarly, a *rank-one tensor* $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_d}$ can be represented as the outer product of $d$ vectors, $\mathcal{X} = \mathbf{a^{(1)}} \otimes \mathbf{a^{(2)}} \otimes \cdots \otimes \mathbf{a^{(d)}}$, where each $\mathbf{a^{(i)}} \in \mathbb{R}^{I_i}$ and $\mathbf{a^{(i)}} \neq 0$ for $1 \leq i \leq d$. Tensors of this form are called *"simple" or "pure" tensors*. In practical applications, a pure tensor serves as a fundamental building block in expressing patterns within a latent space, clarifying relationships across different modalities. Tensor decompositions extend matrix decompositions to higher dimensions and come in several forms. The central idea is to express a tensor as a sum of rank-1 tensors. In this part, we introduce the most widely used methods.

*Canonical Polyadic Decomposition (CPD).* Canonical Polyadic factorization, also known as Candecomp or Parallel Factor Analysis (PARAFAC), is a widely used technique for tensor decomposition. It is most commonly referred to as CPD or simply CP. This method approximates a tensor as a sum of rank-1 tensors. For a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_d}$, the CP decomposition is expressed as:

$$\mathcal{X} = \sum_{r=1}^{R} \lambda_r \, \mathbf{a_r^{(1)}} \otimes \mathbf{a_r^{(2)}} \otimes \cdots \otimes \mathbf{a_r^{(d)}} \tag{6}$$

where $\mathbf{a_r^{(k)}} \in \mathbb{R}^{I_k}$ and $\lambda_\mathbf{r} \in \mathbb{R}, 1 \leq k \leq d,\ 1 \leq r \leq R$. For each mode $k$, the latent factor matrix is $\mathbf{A^{(k)}} = [\mathbf{a_1^{(k)}}, \ldots, \mathbf{a_R^{(k)}}] \in \mathbb{R}^{I_k \times R}$, capturing the relationships along the $k$-th mode. The representation in (6) can be written concisely as $\mathcal{X} = [[\mathbf{\Lambda}; \mathbf{A^{(1)}}, \mathbf{A^{(2)}}, \ldots, \mathbf{A^{(d)}}]]$, where $\mathbf{\Lambda} = [\lambda_\mathbf{1}, \ldots, \lambda_\mathbf{R}]$. The rank $R$ of the tensor can be thought of as the minimal number of rank-1 tensors that yield $\mathcal{X}$ in a linear combination. The number of parameters in this decomposition scales linearly with the order ( $\mathcal{O}(dIR)$ assuming $I_n = I$), so by reducing the number of parameters, CPD decomposition helps address the curse of dimensionality, allowing for efficient analysis of large datasets while retaining interpretability and effectively representing low-rank

structures. Moreover, low-rank CPDs are unique under mild conditions. Intuitively, the CPD being unique means that there is only one way to express the tensor as a sum of rank-1 terms (see, e.g., [6, 42, 43, 44, 45]). This uniqueness of low-rank CPD is notably different from the matrix settings, where the uniqueness of decomposition is only possible with additional constraints such as non-negativity or orthogonality of factor matrices. The fact that such constraints are not needed to ensure CPD uniqueness turns the decomposition into a tool with which we can split a tensor data set into its underlying components — in a "blind" manner, so to speak. At the same time, the absence of constraints such as orthogonality has some drawbacks, including the fact that computing the CP rank is an NP-hard problem [46]. The approximation (e.g., in a least-squares sense) with a fixed CP rank may run into numerical issues, as the optimal solution may not even exist, i.e., the approximation objective may only have an infimum and not a minimum [47]. Constraints (e.g., orthogonality, non-negativity) on (some) factor matrices can make the problem well-posed (see, e.g., [48, 49]). There are several variants of tensor decomposition methods, each defined by distinct constraints on their latent components. A prominent example is non-negative tensor factorization (NTF), which imposes non-negativity on the factor matrices to improve interpretability and facilitate the extraction of meaningful patterns from data [50].

*Tucker decomposition (TD).* Tucker decomposition, also known as low-multilinear rank approximation (LMLRA), represents a tensor as the multilinear product of a core tensor and latent factor matrices, with potentially different ranks along each mode. For a tensor $\mathcal{X}$, the Tucker decomposition is expressed as:

$$\mathcal{X} \approx \sum_{i_1=1}^{R_1} \sum_{i_2=1}^{R_2} \dots \sum_{i_d=1}^{R_d} \mathcal{G}_{i_1,i_2,\dots,i_d}\, \mathbf{a}_{\mathbf{i_1}}^{(\mathbf{1})} \otimes \mathbf{a}_{\mathbf{i_2}}^{(\mathbf{2})} \otimes \dots \otimes \mathbf{a}_{\mathbf{i_d}}^{(\mathbf{d})}, \tag{7}$$

where $\mathbf{A}^{(\mathbf{k})} = [\mathbf{a}_{\mathbf{1}}^{(\mathbf{k})}, \dots, \mathbf{a}_{\mathbf{R_k}}^{(\mathbf{k})}] \in \mathbb{R}^{I_k \times R_k}$ is the factor matrix for the $k$-th mode, with $R_k \leq I_k$. The core tensor $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \dots \times R_d}$ represents the multilinear interactions between different modes. The representation in (7) can be given as $\mathcal{X} = [[\mathcal{G}; \mathbf{A}^{(\mathbf{1})}, \mathbf{A}^{(\mathbf{2})}, \dots, \mathbf{A}^{(\mathbf{d})}]]$. The tuple of minimal integers $(R_1, R_2, \dots, R_d)$ for which equality in (7) holds is the multilinear (ML) rank of the tensor. Furthermore, when the factor matrices are column-wise orthonormal and the core tensor is all-orthogonal and ordered, this decomposition is referred to as *multilinear singular value decomposition* (MLSVD), also known as *higher-order singular value decomposition* (HOSVD) [51]. *All-orthogonality* of the core tensor means that the slices of order $(d-1)$[1] in each mode are mutually orthogonal, i.e. when stretched out into vectors, the inner product of any pair yields 0. Furthermore, *ordered* indicates that the Frobenius norms of these slices (or equivalently, the vectors obtained by stretching) decrease monotonically. The norms are "multilinear singular values", and an order-$d$ tensor has $d$ sets of them, consistent with the ML rank consisting of $d$ values [51]. The *all-orthogonality* property of the core can be seen as a generalization of the singular value matrix in the matrix SVD being

[1] A subtensor obtained by fixing one index

diagonal, as the rows and columns of a diagonal matrix are mutually orthogonal. Note also that the norms of the consecutive rows and columns of the diagonal matrix correspond to the singular values, and in the case of matrices, the row and column view happen to yield the same singular values, consistent with the fact that they also yield the same rank. Analogous to the matrix SVD, the multilinear singular values show the importance of basis vectors in the different modes. The MLSVD is stable with respect to perturbations and allows using standard linear algebra algorithms (in particular SVD) to compute it. However, Tucker decomposition remains suitable for low-order tensors, as the number of parameters required to store the core tensor $\mathcal{G}$ grows exponentially with the order $d$, while the factor matrices contribute only a linear term. This results in a complexity of $\mathcal{O}(dIR + R^d)$, assuming $I_n = I$ and $R_n = R$. Consequently, for higher-order tensors, the exponential growth of the core tensor size leads to increased computational and memory demands, making Tucker decomposition less scalable. Beyond scalability, Tucker and CP decompositions differ in flexibility and interpretability, making them suited for different data tasks. Tucker decomposition allows different ranks per mode, making it well-suited for capturing varying patterns and complex higher-order interactions. In contrast, CP decomposition enforces a fixed rank across all modes, resulting in a simpler and more interpretable factorization, which is particularly useful in applications like recommendation systems that benefit from low-rank approximations.

*ParaTuck decomposition (PTD).* Several designations have been introduced in the literature for this decomposition, including ParaTuck$-2$ in [52, 53] and ParaTuck$-(N_1, N)$ in [54]. For clarity, we adopt the term ParaTuck throughout this paper, as originally used in [52]. The name "ParaTuck" originates from its structural similarity to two well-known tensor decomposition models: PARAFAC, also called the CPD, and the Tucker decomposition. A third-order ParaTuck decomposition of an $I \times J \times K$ tensor $\mathcal{X}$ is defined by its mode-3 slices as:

$$\mathcal{X}_{:,:,k} = \mathbf{A}\mathbf{D}^k_{\mathbf{A}}\mathbf{H}\mathbf{D}^k_{\mathbf{B}}\mathbf{B}^T, \tag{8}$$

where $\mathcal{X}_{:,:,k}$ is an $I \times J$ matrix, while $\mathbf{A}$ and $\mathbf{B}$ are loading matrices of dimensions $I \times R$ and $J \times S$, respectively. The diagonal matrices $\mathbf{D}^k_{\mathbf{A}}$ and $\mathbf{D}^k_{\mathbf{B}}$ are of sizes $R$ and $S$. One can define a matrix $\mathbf{C}_A$ of size $K \times R$, whose $k$-th row corresponds to the diagonal elements of $\mathbf{D}^k_{\mathbf{A}}$. Similarly, a matrix $\mathbf{C}_B$ of size $K \times S$, associated with $\mathbf{D}^k_{\mathbf{B}}$, can be introduced. Lastly, $\mathbf{H}$ is a matrix of size $R \times S$ that encodes the interaction weights. The work in [55] establishes some uniqueness conditions for this decomposition and characterizes the ambiguities between two equivalent decompositions as follows $\mathbf{A} = \tilde{\mathbf{A}}(\mathbf{\Pi_A}\mathbf{\Lambda_A})$, $\mathbf{B} = \tilde{\mathbf{B}}(\mathbf{\Pi_B}\mathbf{\Lambda_B})$, $\mathbf{H} = (\bar{\mathbf{\Lambda}}_{\mathbf{A}}\mathbf{\Lambda}_{\mathbf{A}}^{-1}\mathbf{\Pi}_{\mathbf{A}}^T)\tilde{\mathbf{H}}(\mathbf{\Pi_B}\mathbf{\Lambda}_{\mathbf{B}}^{-1}\bar{\mathbf{\Lambda}}_{\mathbf{B}})$, $\mathbf{D}^k_{\mathbf{A}} = (z_k\mathbf{\Pi}_{\mathbf{A}}^T)\tilde{\mathbf{D}}^k_{\mathbf{A}}(\mathbf{\Pi_A}\bar{\mathbf{\Lambda}}_{\mathbf{A}}^{-1})$ and $\mathbf{D}^k_{\mathbf{B}} = (z_k^{-1}\mathbf{\Pi}_{\mathbf{B}}^T)\tilde{\mathbf{D}}^k_{\mathbf{B}}(\mathbf{\Pi_B}\bar{\mathbf{\Lambda}}_{\mathbf{B}}^{-1})$, where $\mathbf{\Lambda_A}$, $\mathbf{\Lambda_B}$, $\bar{\mathbf{\Lambda}}_{\mathbf{A}}$, and $\bar{\mathbf{\Lambda}}_{\mathbf{B}}$ are diagonal scaling matrices, $\mathbf{\Pi_A}$ and $\mathbf{\Pi_B}$ are permutation matrices, and $z_k$ are nonzero scalars. As with CPD, an alternating least squares (ALS)-type algorithm was introduced by Rasmus Bro for estimating PTD in [56].

This approach sequentially updates each factor matrix by minimizing a loss function while keeping the others fixed. Although this strategy is effective for CPD, it presents significant difficulties when applied to PTD. To overcome these challenges, a recently proposed stochastic algorithm in [57] incorporates stochastic updates within the ALS framework, helping mitigate common issues such as swamps and local minima encountered in the traditional ALS method. Despite its limited adoption in signal processing, PTD has been successfully applied in wireless communications [58, 59]. Moreover, recent advances in [60] highlight its suitability for modeling two-layer neural networks with adaptable activation functions. PTD combines aspects of CPD and Tucker, improving scalability over Tucker by avoiding a fully dense core tensor and leveraging structured factor matrices with diagonal scaling. While more efficient than Tucker, PTD's iterative optimization can limit scalability for large datasets, making it less scalable than CPD or Tensor Train (TT) decompositions for very high-order tensors. Similarly, in terms of interpretability, PTD lies between CPD and Tucker.

*Tensor Train (TT) decomposition.* Tensor Train decomposition efficiently factorizes a high-dimensional tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_d}$ into a sequence of 3-way tensors, or cores, expressed as:

$$\mathcal{X}_{i_1,i_2,\ldots,i_d} \approx \sum_{R_0,R_1,\ldots,R_d} \mathcal{G}^{(1)}_{R_0,i_1,R_1} \mathcal{G}^{(2)}_{R_1,i_2,R_2} \cdots \mathcal{G}^{(d)}_{R_{d-1},i_d,R_d}, \tag{9}$$

where each core $\mathcal{G}^{(k)} \in \mathbb{R}^{R_{k-1} \times I_k \times R_k}$ for $1 \leq k \leq d$. The tuple of minimal integers $(R_0, R_1, \ldots, R_d)$ for which equality in (9) holds is the TT rank of the tensor [13]. This decomposition achieves a substantial reduction in both computational complexity and memory storage. Note that the TT cores are only of order 3. The TT decomposition is stable like MLSVD and overcomes the curse of dimensionality like CPD since it has asymptotically the same number of parameters ( $\mathcal{O}(2IR + (d-2)IR^2)$, assuming $I_n = I, R_n = R$). Operations on the tensor, such as contractions and element-wise operations, become more manageable. TT decomposition is highly efficient for handling very high-order tensors due to its linear scaling in the number of parameters, making it memory-efficient [13]. It is particularly useful for datasets with sequential dependencies and is effective in compressing large tensors while retaining key structural information. However, for lower-order tensors (e.g., 3-way or 4-way), the advantages of TT in terms of efficiency and scalability become less pronounced, and its added complexity may not always be justified, especially when simplicity and interpretability are preferred. In contrast, CP decomposition offers a more intuitive and easily interpretable structure, making it a favorable option when interpretability is a priority.

*Scalability challenges and advances in tensor decompositions.* Scalability remains a central challenge in tensor-based methods, especially as data dimensions and sample sizes increase across various applications. CPD and Tucker decompositions, while widely used, face scalability issues due to their computational and memory demands. This challenge is particularly pronounced in Tucker decomposition, which requires

storing a dense core tensor in addition to factor matrices, leading to exponential memory growth with increasing tensor order. While rank truncation and hierarchical Tucker formats help mitigate this issue, scalability remains a concern [61]. CP decomposition, in contrast, avoids the need for a core tensor, making it more memory-efficient; however, its iterative algorithms often suffer from slow convergence, leading to high computational costs. To address these limitations, randomized algorithms have been introduced for CP and Tucker, reducing computational complexity through random projections that lower dimensionality before factorization [62, 63]. Tensor Train and Tensor Ring decompositions provide more scalable alternatives by factorizing tensors into a sequence of low-rank matrices, significantly reducing storage and computational costs [64]. Beyond decomposition strategies, memory-efficient techniques such as Memory-Efficient Tucker dynamically adjust computational strategies based on available memory resources, improving scalability [65]. In addition to algorithmic advancements, exploiting data characteristics such as sparsity and leveraging hardware-aware optimizations (e.g., cache-aware strategies, parallelism, and specialized data structures) further enhance scalability [66, 67]. Streaming tensor decomposition techniques improve scalability by processing tensors incrementally rather than loading them entirely into memory, making them well-suited for large-scale, dynamic datasets and real-time applications [68, 69].

*Nuclear, spectral norm of tensors.* Utilizing the tensor factorization described above, we define the tensor nuclear and spectral norms, which generalize the matrix norms. Key matrix norms include the Frobenius norm (also known as the Euclidean ( $\ell_2$-norm), the spectral norm (or operator norm), and the nuclear norm. If $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r \geq 0$ are the singular values of a matrix $\mathbf{A}$, the Frobenius norm is given by $\|\mathbf{A}\|_F = \sqrt{\lambda_1{}^2 + \lambda_2^2 + \cdots + \lambda_r{}^2}$, spectral norm is $\|\mathbf{A}\|_\sigma = \lambda_1$, and the nuclear norm is $\|\mathbf{A}\|_\star = \lambda_1 + \lambda_2 + \cdots + \lambda_r$. The nuclear norm can be considered as a convex relaxation of matrix rank and it has been used in matrix completion problems [70]. Similar to matrices, the nuclear norm of a tensor serves as a convex relaxation of its rank, and both the nuclear and spectral norms are crucial in tensor completion problems [71]. The spectral norm of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_d}$ is defined as:

$$\|\mathcal{X}\|_\sigma = \sup \left\{ \left\langle \mathcal{X}, a^{(1)} \otimes a^{(2)} \otimes \cdots \otimes a^{(d)} \right\rangle : a^{(j)} \in \mathbb{R}^{I_j}, \|a^{(j)}\| = 1,\ 1 \leq j \leq d \right\}. \tag{10}$$

The dual of the spectral norm is the *nuclear norm*, which can be expressed as:

$$\|\mathcal{X}\|_\star = \inf \left\{ \sum_{i=1}^{r} |\lambda_i| : \mathcal{X} = \sum_{i=1}^{r} \lambda_i a_i^{(1)} \otimes a_i^{(2)} \otimes \cdots \otimes a_i^{(d)},\ \lambda_i \in \mathbb{R}, \|a_i^{(j)}\| = 1 \right\}. \tag{11}$$

Various approaches to estimate and evaluate the spectral and nuclear norms, along with their upper and lower bounds, have been explored by multiple researchers, though these computations remain challenging due to the NP-hard nature of the associated decision problems [72, 73].

## 2. Tensor Methods for Remote Sensing

### *2.1. Introduction to Remote Sensing and Spectral Unmixing*

Remote sensing collects data about objects or phenomena from a distance, without direct physical interaction. It is essential for applications like environmental monitoring, resource exploration, urban planning, and disaster response. In remote sensing, satellite-mounted cameras capture portions of the Earth's surface, with spectral sensitivity ranging from 400nm to 2500nm, enabling detailed material interaction identification [74]. Each acquisition represents a scene in a narrow range of wavelengths (called a spectral band). Stacking these forms 3-way tensors of shape $I \times J \times K$, where $I$ and $J$ are spatial dimensions and $K$ is the spectral dimension. Hyperspectral images (HSIs) offer high spectral but low spatial resolution, while multispectral images (MSIs) provide high spatial resolution with few spectral bands. Super-resolution images (SRIs), which combine both high spatial and high spectral resolutions, are crucial for applications like environmental monitoring and mineral exploration.

*Blind spectral unmixing.* Traditional blind spectral unmixing involves identifying materials in a SRI with limited prior information by estimating the spectral signatures of endmembers and their corresponding abundance maps. The linear mixing model (LMM) assumes each pixel is a linear combination of the spectral signatures of the materials in the scene. This paper focuses on LMM methods, though nonlinear mixing and sparse models are used for more complex pixel interactions or when few materials contribute to each pixel. The traditional approach unfolds a SRI $\mathcal{Y} \in \mathbb{R}^{I\times J\times K}$ into the matrix $\mathcal{Y}_{(3)} \in \mathbb{R}^{K\times IJ}$, where each row represents a pixel and each column is a spectral band. The spectral unmixing problem involves estimating the spectral signatures matrix $\mathbf{E} \in \mathbb{R}^{K\times R}$ and the abundance matrix $\mathbf{S} \in \mathbb{R}^{IJ\times R}$, which represent the spectral signatures and their proportions in each pixel, respectively. The LMM is then formulated as:

$$\mathbf{Y_{(3)}} \approx \mathbf{E}\mathbf{S}^\top, \tag{12}$$

where the term $\approx$ stands for the presence of noise. The matrices $\mathbf{S}$ and $\mathbf{E}$ are non-negative with the constraint that the abundances for each pixel sum to 1, *i.e.*, $\sum_{j=1}^{R} \mathbf{S}_{i,j} = 1$. Determining the rank of the factorization in (12), which corresponds to the number of endmembers, is a challenging task.

Traditional matrix-based unmixing methods fall into four categories. Geometrical methods seek endmembers within the unit simplex of the observed image [75]. Statistical methods, e.g., [76], assume endmembers are statistically independent, but this may not hold in remote sensing, where spectral signatures can be correlated. Bayesian methods such as FUSE [77] and HySure [78] require detailed prior knowledge. Regularized methods such as SFIM [79] promote specific properties in the images, e.g., sparsity in the abundance maps. Non-negative matrix factorization (NMF), such as CNMF, enforces non-negativity constraints on the factors [80, 81].

Using matricized data instead of preserving the tensor format can lead to loss of structural information on intermodal interactions. While some methods promote spatial smoothness through total variation constraints, they introduce hard-to-determine tuning parameters. Relying solely on spatial smoothness fails to capture the interdependence between spatial and spectral dimensions that is naturally accessible from the tensor format [82]. Furthermore, matrix-based approaches can hinder identifiability, whereas tensor factorizations provide unique recovery under milder conditions [45, 44].

### *2.2. Hyperspectral super-resolution and joint problems*

*Hyperspectral super-resolution.* In remote sensing, a trade-off exists between spatial resolution, spectral resolution, and signal-to-noise ratio (SNR). The acquisition and storage of high-quality super-resolution images (SRI) can be challenging. Common practice is to resort to faster acquisition yielding images with lower resolutions, typically in the form of HSI and MSI images. Hyperspectral super-resolution (HSR) [79] involves recovering a SRI from an HSI and an MSI of the same scene (see Figure 3). Due to having more parameters to estimate than available observations, HSR is ill-posed, meaning the solution is not unique without additional diversities or regularizations.

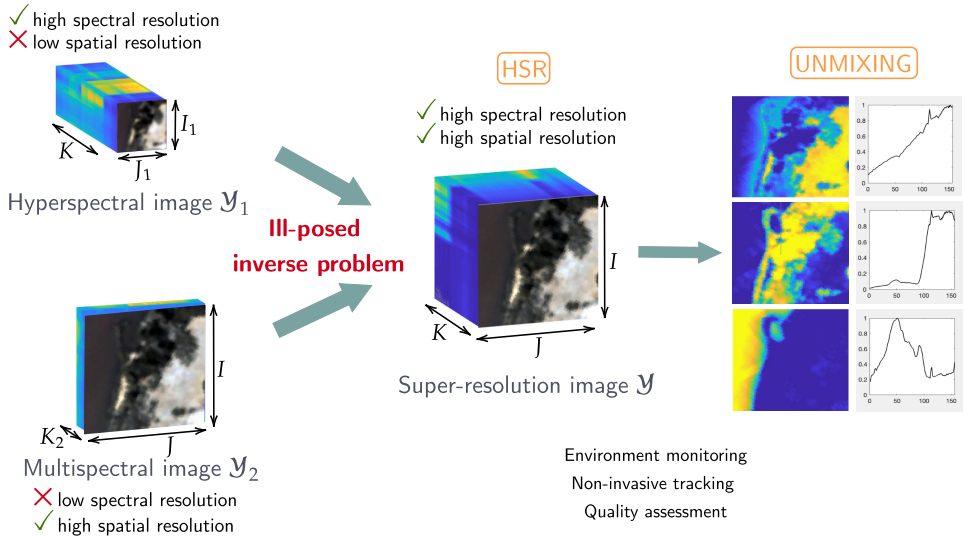



Figure 3: Recovering hyperspectral super-resolution (HSR) images from hyperspectral image and multispectral image resolution and unmixing in remote sensing and spectral unmixing of HSR images.

*Formulation of joint problems.* State-of-the-art unmixing algorithms aim at recovering high-resolution spectral signatures and abundance maps from mixed pixels in the SRI [83]. Here, since the SRI is unknown and only the HSI is observed with high spectral resolution, traditional unmixing algorithms would only able to recover spatially-degraded versions of the abundance maps from the HSI together with high-resolution spectral signatures [84]. Conversely, traditional unmixing performed on the MSI would yields high-resolution

abundance maps but degraded special signatures. Fusion of an HSI with an MSI enables recovery of a higher-resolution image, motivating the development of efficient methods to address both the HSR and unmixing problems in a unified framework. Early matrix approaches to the HSR problem [80, 77, 78, 85] performed coupled low-rank factorization of the matricized images. Some methods based on the linear mixing model addressed both HSR and unmixing [86, 80].

### *2.3. Tensor Decomposition Methods for HSR and Spectral Unmixing*

Tensor-based HSR can be performed by applying a low-rank tensor approximation to the target SRI $\mathcal{Y} \in \mathbb{R}^{I\times J\times K}$, see for instance:

$$\mathcal{Y} \approx \sum_{i=1}^{R_1}\sum_{j=1}^{R_2}\sum_{k=1}^{R_3} \mathbf{a_i} \otimes \mathbf{b_j} \otimes \mathbf{e_k} = [[\mathcal{I}; \mathbf{A}, \mathbf{B}, \mathbf{E}]], \tag{13}$$

where the matrices $\mathbf{A} \in \mathbb{R}^{I\times R_1}$ and $\mathbf{B} \in \mathbb{R}^{J\times R_2}$ represent latent factors for abundances across spatial dimensions, and $\mathbf{E} \in \mathbb{R}^{K\times R_3}$ contains spectral signatures. Tensor-based HSR methods built from (13), such as STEREO and Blind-STEREO in [82, 87], used coupled CPD but ignored non-negativity, which made the output factors unable to represent spectral signatures and abundance maps. A coupled Tucker decomposition named SCOTT was later used in [88], but its non-uniqueness limited interpretability and unmixing capabilities. Additional experiments in a semi-blind scenario, where spatial degradation matrices were unknown, showed that the spectra and abundance maps were correctly estimated without prior knowledge of the degradation matrix. The results of these experiments are available in [17, Section V.E].

*Observation model.* Let $\mathcal{Y}_1 \in \mathbb{R}^{I_1\times J_1\times K}$ and $\mathcal{Y}_2 \in \mathbb{R}^{I\times J\times K_2}$ represent the observed HSI and MSI, where $I_1 < I, J_1 < J$ and $K_2 < K$. Both images are downsampled versions of the SRI $\mathcal{Y}$ with linear downsampling operators $\mathbf{P}_1 \in \mathbb{R}^{I_1\times I}, \mathbf{P}_2 \in \mathbb{R}^{J_1\times J}$ and $\mathbf{P}_3 \in \mathbb{R}^{K_2\times K}$. The coupled observation model is given by

$$\mathcal{Y}_1 \approx \mathcal{Y} \times_1 \mathbf{P}_1 \times_2 \mathbf{P}_2 \text{ and } \mathcal{Y}_2 \approx \mathcal{Y} \times_3 \mathbf{P}_3, \tag{14}$$

where the HSI is degraded in the spatial dimensions and the MSI in the spectral dimension [82].

*Formulation as a coupled Block-Term Decomposition.* Building on tensor-based models, the Block-Term Decomposition (BTD) [44] offers more flexibility than standard low-rank models by decomposing the tensor into a sum of low-rank blocks, which allows for modeling more complex data structures. Furthermore, the capabilities of the BTD to provide a link with the LMM justify its prime interest for solving the joint HSR and unmixing problems. For HSR, we assume the SRI admits a BTD with ranks $(L_r, L_r, 1)$ as

$$\mathcal{Y} \approx \sum_{r=1}^{R} (\mathbf{A}_r \mathbf{B}_r^\top) \otimes \mathbf{e}_r, \tag{15}$$

where $\mathbf{A}_r \in \mathbb{R}^{I\times L_r}, \mathbf{B}_r \in \mathbb{R}^{J\times L_r}, \mathbf{e}_r \in \mathbb{R}^{K\times L_r}$ are the factor matrices (resp. vectors) corresponding to the spatial and spectral dimensions for each block, and the BTD factors are $\mathbf{A} = [\mathbf{A}_1 \dots \mathbf{A}_R] \in \mathbb{R}^{I\times\sum_r L_r}, \mathbf{B} = [\mathbf{B}_1 \dots \mathbf{B}_R] \in \mathbb{R}^{J\times\sum_r L_r}, \mathbf{E} = [\mathbf{e}_1 \dots \mathbf{e}_R] \in \mathbb{R}^{K\times R}$. The $\mathbf{A}_r\mathbf{B}_r^\top$ terms can be viewed as the abundance matrices for each of the R materials, while the $\mathbf{e_r}$ are associated with each spectral signature. As such, and under non-negativity constraints on the BTD factors, the low-rank model (15) can be viewed as a tensor version of the LMM (see [89] for more details). Hence the interest of the BTD model for modeling the SRI. Under this structure, the observation model (14) becomes a coupled BTD model of the form

$$\begin{cases} \mathcal{Y}_1 \approx & \sum_{r=1}^{R} (\mathbf{P}_1\mathbf{A}_r)(\mathbf{P}_2\mathbf{B}_r)^\top \otimes \mathbf{e}_r, \\ \mathcal{Y}_2 \approx & \sum_{r=1}^{R} (\mathbf{A}_r\mathbf{B}_r^\top) \otimes \mathbf{P}_3\mathbf{e}_r. \end{cases} \tag{16}$$

BTD captures intricate data variations, and has been widely used for SRI unmixing in [90] and HSR tasks [91, 84, 92, 93]. The BTD model (15) is the tensor extension of the linear mixing model (12), contributing to its popularity. Additionally, the algorithm CNN-BTD-Var [89] applied Coupled Non-Negative BTD for joint fusion and unmixing with spectral variability.

*Challenges in Tensor-Based HSR and Unmixing.* Addressing both HSR and unmixing poses several challenges. First, observation models and algorithms require non-negativity constraints. Second, observed images often contain similar or nearly collinear materials, leading to rank-deficient mixing factors—an issue for BTD, which is sensitive to initialization [89]. The most critical challenge is estimating the number of underlying materials, often unknown. Incorrect estimation can impair fusion and unmixing performance and compromise solution uniqueness. The number of materials can often be estimated from the HSI's spectral information using techniques like penalized optimization [94], Bayesian inference [95], or autoregressive models [96].

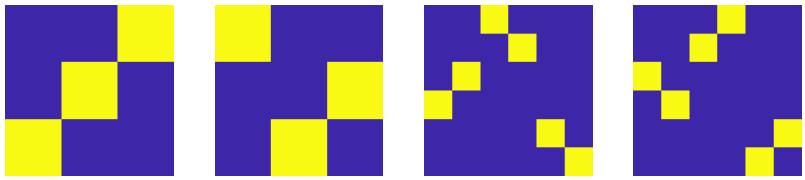
### *2.3.1. Case Study: Joint HSR and Unmixing via Block Term Decomposition*

This section highlights [17], which introduces coupled optimization problems using the BTD to address joint HSR and unmixing problems. Key insights from this paper are summarized below.

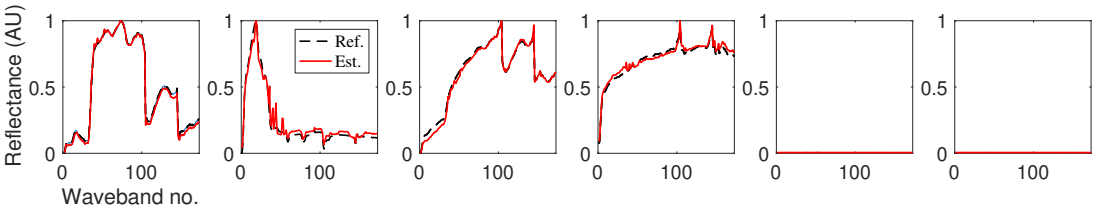
Minimizing the approximation errors in (16) reduces to minimizing the following cost function:

$$\Phi = D_\beta\left(\mathcal{Y}_1 \middle\| \sum_{r=1}^{R} \left(\mathbf{P}_1\mathbf{A}_r\left(\mathbf{P}_2\mathbf{B}_r\right)^\top\right) \otimes \mathbf{e}_r\right) + \lambda D_\beta\left(\mathcal{Y}_2 \middle\| \sum_{r=1}^{R} \left(\mathbf{A}_r\mathbf{B}_r^\top\right) \otimes \mathbf{P}_3\mathbf{e}_r\right) \tag{17}$$

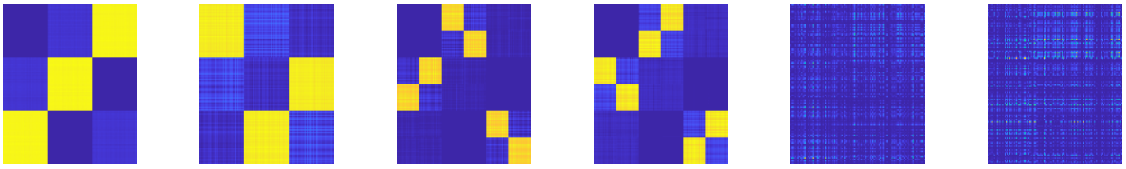
The tensor $\beta$-divergence, $D_\beta$, models various noise types: $\beta = 2$ corresponds to Euclidean distance for Gaussian noise, and $\beta = 1$ to KL-divergence for Poisson noise. In [97], the optimal value $\beta = 1.5$ for remote sensing captures a mix of Gaussian and Poisson noise. Smaller $\beta$ values emphasize low-magnitude entries, making the $\beta$-divergence effective for identifying spectral signatures of e.g., water-based materials. The



(a) Reference abundance maps ($R = 4$) for the synthetic dataset.




(b) Reference and estimated spectra ($\beta = 1$).



(c) Estimated abundance maps ($\beta = 1$).

Figure 4: Unmixing and rank estimation on the synthetic dataset.

optimization approach in [17] incorporates minimum-volume regularization which is given as

$$\underset{\mathbf{A_r}\mathbf{B_r^T},\mathbf{e_r},\mathbf{P_i}}{\text{minimize}} \ \Phi + \gamma \log\det(\mathbf{E}^\top\mathbf{E} + \delta\mathbf{I}), \tag{18}$$

s.t. $\Delta^K = \{\mathbf{x} \in \mathbb{R}_+^K \,|\, \sum_{k=1}^K \mathbf{x}_k = 1\}$ and $\mathbf{A_r}{\mathbf{B_r}}^\top \geq 0$, $\mathbf{P_i} \geq 0$, $\mathbf{e_r} \in \Delta^K$ for all $r, i$. The term $\log\det(\mathbf{E}^\top\mathbf{E}+\delta\mathbf{I})$ acts as measure of the volume spanned by the columns of $\mathbf{E}$, where $\delta$ is a small positive constant that prevents $\log\det(\mathbf{E}^\top\mathbf{E})$ from going to $-\infty$ when $\mathbf{E}$ becomes rank-deficient [98]. This regularization enhances BTD interpretability and addresses the challenges of rank deficiency and determining the number of factors. In [98], the minimum-volume regularization is applied to uncoupled matrix models. In remote sensing, HSI and MSI may be acquired at different times, requiring flexible models to estimate degradation operators linking observations to the target SRI. The proposed algorithm estimates these operators, extending the principle from coupled matrix models in [81] to a coupled tensor framework.

In [17], a volume-regularized algorithm is proposed using an iterative scheme with multiplicative updates (MU). The BTD factors and degradation operators are updated with MU, considering tensor unfoldings in the first two dimensions. For the spectral signature factor, the minimum-volume criterion (18) is incorporated. The computational complexity is similar to standard MU for $\beta$-NMF, with tensor unfoldings determining the complexity. Some steps can be parallelized on CUDA-enabled GPUs.

In the paper, several experiments were conducted.

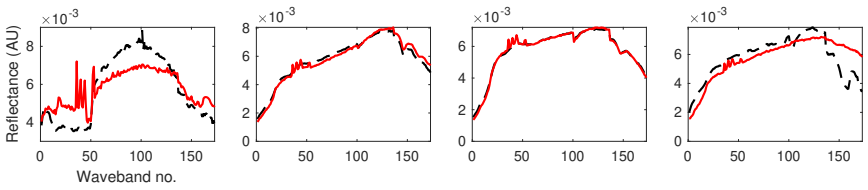
*Estimation of the number of endmembers.* In the first test, an SRI with an exact BTD was created using $R = 4$ spectral signatures from the Jasper Ridge dataset. The SRI $\mathcal{Y} \in \mathbb{R}^{120\times120\times173}$ was divided into 36 spatial blocks, with $(L_1, L_2, L_3, L_4) = (3, 3, 6, 6)$. Each abundance map $S_r$ $(1 \leq r \leq 4)$ is structured as a block matrix with $L_r$ blocks, where entries within each block are set to one and entries outside are zero, enforcing the "pure pixel assumption" to ensure each block represents a single material. The reference SRI was computed using (15). In the paper, $R$ is set to 6 in order to simulate rank overestimation. Figure 4 shows the unmixing results with the proposed approach. The proposed algorithm accurately recovers the four target spectral signatures and abundance maps, with the other two materials showing near-zero values. These results highlight the volume-constrained approach's ability to perform both model order selection and unmixing.

*Joint fusion and unmixing.* The authors also considered the Ivanpah Playa [99] reference SRI $\mathcal{Y} \in \mathbb{R}^{95\times95\times156}$. This dataset is composed of four, very similar materials: road, solar panels, light sand and darker sand, which is a challenging example for which unmixing may lead to rank-deficient solutions. Figure 5 shows the unmixing results provided by the proposed algorithm. Even though the materials seem very similar, they are correctly separated by the proposed approach. The proposed algorithm was compared to various tensor-based and matrix-based methods for hyperspectral unmixing, featuring different models, with and without minimum-volume regularization and using different divergence measures (see Table I in [17, Section V]). Despite a small tradeoff between reconstruction and unmixing performance, the proposed algorithm remained highly competitive for both tasks, even on a challenging dataset.

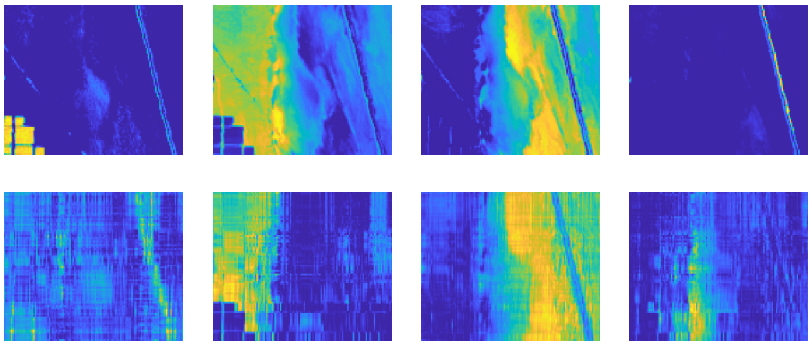
*Conclusions and perspectives.* This paper introduces a family of coupled tensor-based optimization problems using beta-divergence to account for various noise statistics. The proposed algorithms, leveraging BTD, address both reconstruction and unmixing tasks in remote sensing, even in challenging scenarios like semi-blind cases or near-collinear materials. A minimum-volume regularization is introduced, improving unmixing performance at the cost of a slight reconstruction tradeoff. The algorithm successfully estimates the number of materials while performing high-quality unmixing. The main challenge addressed is the correct estimation of material count, especially in the presence of spectral variability. Future work will explore the asymptotic behavior of estimates with minimum-volume constraints with the aim to perform such a task.

## 3. Tensor decomposition for wireless communication

We first define key concepts in wireless communication, such as multiple-input multiple-output systems, base station, user equipment, codebooks, and unsourced random access, as the terminology may not be familiar to all readers, ensuring clarity for the upcoming sections.




(a) Reference and estimated spectra.



(b) Reference (middle) and estimated (bottom) abundance maps.

Figure 5: Unmixing of the Ivanpah Playa dataset.

### *3.1. Preliminaries*

*MIMO system.* A multiple-input and multiple-output (MIMO) system is a communication system that uses multiple antennas at the transmitter and receiver to enhance performance. MIMO systems are widely used in wireless communications (e.g., Wi-Fi, LTE, 5G) to improve data rates, capacity, and reliability.

*Base Station (BS).* A central communication hub in wireless networks that connects mobile devices to the core network. It is responsible for transmitting and receiving radio signals to and from user equipment.

*User Equipment (UE).* Refers to any device that connects to a network to access services and communicate with other devices or systems. It encompasses a wide range of devices, including smartphones, Internet of Things (IoT) devices, and connected vehicles.

*Unsourced Random Access (URA).* A communication paradigm where multiple transmitters, randomly active, send messages (payloads) to a central receiver without explicitly identifying themselves, i.e., they cannot be identified before signals decoding.

*Codebook.* In communication systems, a codebook is a set of predefined codewords or symbols with which information can be encoded for transmission purposes, as well as for signal processing tasks. Every entry in the codebook represents a possible signal or message that can be transmitted or received.

*OFDM (Orthogonal Frequency Division Multiplexing).* A type of multicarrier modulation technique widely used in modern communication systems to transmit data efficiently. In OFDM, the data stream is split into several smaller sub-streams, which are transmitted simultaneously over different frequencies (subcarriers).

*Uniform Rectangular Array.* A type of planar antenna array where the antenna radiating elements are arranged in a rectangular grid with equal spacing between them along both the horizontal (azimuth) and vertical (elevation) directions.

*Antenna polarization.* Refers to the orientation of the electric field of the radio wave emitted or received by an antenna. It defines the direction in which the electric field oscillates as the electromagnetic wave propagates through space.

### *3.2. Introduction*

Tensor algebra has become an increasingly important tool at the heart of most modern communication, providing a very powerful framework for the representation, analysis, and optimization of multi-dimensional data structures. In wireless communication, signals are traditionally analyzed individually across dimensions such as time, frequency, space, and user indices, but this approach is computationally inefficient and struggles to scale with high-dimensional data. Tensor algebra models multi-way relationships more compactly, allowing for efficient representation and analysis of large datasets. This has been particularly useful in MIMO systems, cooperative communication, and cognitive radio, where the number of devices, antennas, and communication channels is notably growing [100]. Advances in tensor decomposition models, have enhanced key tasks like user separation and channel estimation [101], boosting system capacity and reliability. These techniques are also vital for developing multiuser communication schemes that enable massive access.

URA [102] enables an arbitrarily large number of UEs to share the same codebook, potentially reducing the minimum energy required for reliable communication, and it does not allow for user identification prior to signal decoding [1]. Since URA only concerns active users transmitting small payloads, activity detection is unnecessary, and decoding complexity scales with the number of active users, enhancing system scalability. URA operates on a grant-free paradigm, allowing UEs to transmit without prior scheduling, making it ideal for scenarios with many devices and sporadic activity. Tensor-based modulation (TBM) [103] is a promising scheme for massive URA, offering strong performance in multiple-antenna configurations. In TBM, assuming flat fading channels, the transmitted signals are arranged as rank-one tensors and efficient user separation is enabled by CPD without pilot sequences. Hence, the number of TBM-modulated users is, in fact, the rank of the tensor-structured signal at the receiver (assuming a noiseless channel), which is consequently bounded by the CPD bound.

*3.3. Tensor-based modulation digest: Signal modeling and related optimization problem*

Let $K_{\mathrm{a}}$ active users among a total of $K$ single-antenna UEs transmitting their signals $\mathbf{s}_k \in \mathbb{C}^T$ ($k = 1, \cdots, K_{\mathrm{a}}$) over the same $T$ resources (or channel uses). $\mathbf{s}_k$ is the signal mapping a binary message of size $B$. Since transmitted signals are rank-one tensor structured, they are expressed as $\mathbf{s}_k = \bigotimes_{i=1}^{d} \mathbf{x}_k^{(i)}$, where $\mathbf{x}_k^{(i)} \in \mathscr{C}_i \subseteq \mathbb{C}^{T_i}$. Thus, $\mathbf{s}_k$ is of order $d$ tensor of size $T_1 \times T_2 \times \cdots \times T_d$. $\mathscr{C}_i$ is the signal sub-constellation on the mode $i$, knowing that under the unsourced context, active users share the same sub-constellation codebook. Figure 6 illustrates how the transmitted signals $\mathbf{s}_k$ are structured. At an $N$-antenna receiver, the observed signal $\mathbf{y}$, the vectorized form of the $(d+1)$-dimensional tensor $\mathcal{Y} \in \mathbb{C}^{T_1 \times \cdots \times T_i \times \cdots \times T_d \times N}$, is formulated as

$$\mathbf{y} \triangleq \mathsf{vec}(\mathcal{Y}) = \sum_{k=1}^{K_{\mathrm{a}}} \mathbf{s}_k \otimes \mathbf{h}_k + \mathbf{w} \quad \in \mathbb{C}^{TN}, \tag{19}$$

where $\mathbf{h}_k \in \mathbb{C}^N$ is the block-fading channel, $\mathbf{w} \in \mathbb{C}^{TN}$ is the circularly symmetrical complex Gaussian noise at the receiver, and $T \triangleq \prod_{i=1}^{d} T_i$. Accordingly, the noiseless version of $\mathbf{y}$ (equivalently $\mathcal{Y}$) is a rank-$K_{\mathrm{a}}$ tensor. Then, the joint multi-user maximum likelihood (ML) receiver is expressed as

$$\left\{\hat{\mathbf{x}}_k^{(i)}\right\} = \underset{\substack{\mathbf{x}_k^{(i)} \in \mathscr{C}_i \\ 1 \le i \le d}}{\operatorname{argmin}} \left( \min_{\substack{\mathbf{h}_k \in \mathbb{C}^N \\ 1 \le k \le K_{\mathrm{a}}}} \left\| \mathbf{y} - \sum_{k=1}^{K_{\mathrm{a}}} \mathbf{x}_k^{(1)} \otimes \cdots \otimes \mathbf{x}_k^{(i)} \otimes \cdots \otimes \mathbf{x}_k^{(d)} \otimes \mathbf{h}_k \right\|^2 \right). \tag{20}$$

The optimization problem in TBM is a tensor rank computation problem, which is NP-hard and remains

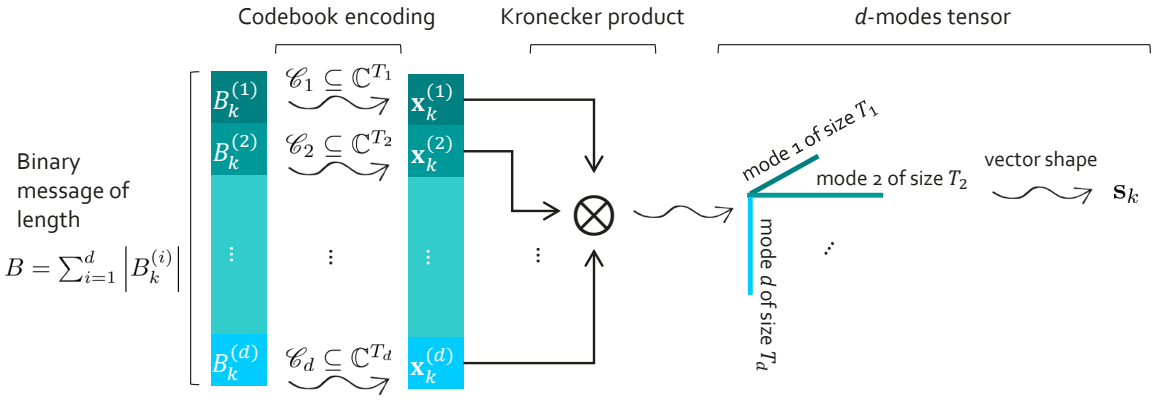



Figure 6: Description of signal modulation using the TBM scheme.

exponential in complexity even with discrete codebooks [104]. To handle user separation, tensor-rank decomposition variants like alternating least squares (ALS) for CPD are used, followed by refining steps for better estimates. The TBM decoder involves two main steps: multi-user separation (via CPD) and single-user demapping (likelihood tests using the codebook).

*3.4. Case study: Tensor-based modulation over frequency-selective channels*

Frequency-selective, or multipath channel modeling, is a crucial component of wireless communication due to the fact that it precisely emulates real-world signal propagation effects-where the signals reflect off various obstacles en route to a receiver with different delays and amplitudes. A number of undesirable consequences for the signals ensue from this frequency-dependent fading, including intersymbol interference. Figure 7 illustrates the considered transmission environment in [19]. In frequency-selective channels, the CPD structure is not maintained at the receiver due to the multipath effect. As a result, the performance of the state-of-the-art TBM decoder is largely degenerated. To address this issue, authors in [19] have developed a new tensor decomposition method that takes this effect into account, resulting in more effective user separation.

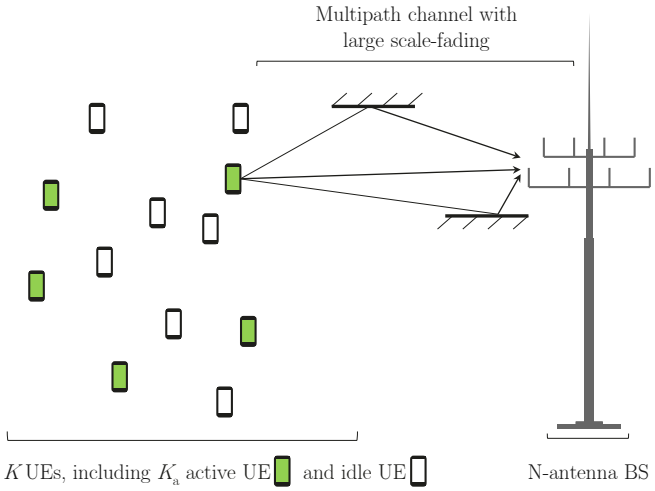



Figure 7: Illustration of massive random access communications over frequency-selective channels.

Under the assumption of time-invariant channel as in the state-of-the-art TBM, and OFDM, the channel realization in the frequency domain is expressed as $\mathbf{h}_{k,l}(f) = \mathbf{g}_{k,l} \exp\left(2\jmath\pi f \frac{\tau_{k,l}}{\Delta_{\mathrm{s}}}\right)$, $f$ is the subcarrier index, $\Delta_{\mathrm{s}}$ is the OFDM symbol duration, $\tau_{k,l}$ and $\mathbf{g}_{k,l} \in \mathbb{C}^N$ are the channel delay and the channel gain of the path $l$, respectively. Associating this channel modeling with the TBM signal structure, (19) is reformulated, considering the $L$ channel paths, as

$$\mathbf{y} \triangleq \mathsf{vec}(\mathcal{Y}) = \sum_{k=1}^{K_{\mathrm{a}}} \sum_{l=1}^{L} (\mathbf{s}_k * \mathcal{C}_{k,l}) \otimes \mathbf{g}_{k,l} + \mathbf{w}, \tag{21}$$

where $\mathcal{C}_{k,l}$ is the $T$-dimensional vector gathering the phase shifts over all the channel uses.

*Encoder side.* The main idea characterizing the proposed encoder is to divide the users into several orthogonal groups where every group will use fewer OFDM subcarriers for their signal transmission and, hence,

each will experience less severe channel frequency selectivity. Thus, the transmitted signals for users of the group $q$ are $\mathbf{s}_k = \mathscr{G}_q \left( \bigotimes_{i=1}^{d} \mathbf{x}_k^{(i)} \right)$, where $\mathscr{G}_q$ is the user-subcarriers mapping function for the group $q$.

*Decoder side.* The decoder is composed of two main steps: (i) user groups demapping, i.e., retrieving signals of each user group separately by applying $\mathscr{G}_q^{-1}$ to the received signal $\mathbf{y}$, and (ii) for each group, separate and decode the group-specific signals. It should be noted that signal separation is the main innovative point in the transceiving scheme designed in [19]. For each user group, at the user separation step, it is assumed that the receiver has an estimate of the main propagation parameters over the corresponding subcarriers (we may designate it as sub-channel), that is, the delay spread and a defined number of paths $L'$, which also refers to the number of possible delays. The delays over that sub-channel are set as uniformly distributed between zero and the estimated delay spread. Thus, at the receiver, signals are modeled with respect to phase shift vectors that derive from the estimated delay spread and the defined number of paths $L'$. Therefore, a matrix $\mathbf{C}_i \in \mathbb{C}^{T_i \times L'}$ is characterized with respect to each source signal factor $\mathbf{X}^{(i)} \triangleq \left[ \mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \cdots, \mathbf{x}_{K'_{\mathrm{a}}}^{(i)} \right] \in \mathbb{C}^{T_i \times K'_{\mathrm{a}}}$, $K'_{\mathrm{a}}$ is the number of active users in a user group. On the other hand, from (21), one can note that the tensor $\mathcal{Y}$ (the noiseless case) is no more of rank equal to the number of users, and each signal rank-one tensor is replicated to the number of propagation paths. To deal with that, the user separation approach introduced in [19] is based on a new CPD algorithm called Co-ALS, designed to perform tensor rank decomposition while handling correlated components. Recalling that the classical CPD ALS algorithm is based on the following formulation:

$$\widehat{\mathbf{X}}^{(i)} = \underset{\mathbf{X}^{(i)}}{\operatorname{argmin}} \left\| \mathcal{Y}_{(i)} - \mathbf{X}^{(i)} {\mathbf{K}^{(i)}}^{\top} \right\|_{\mathrm{F}}, \tag{22}$$

where $\mathbf{K}^{(i)}$ is the Khatri–Rao product, in descending order, of all factors except the mode-$i$ factor. Knowing that the matrix $\mathbf{C}^{(i)}$ models the correlation between the components of the factor $\mathbf{X}^{(i)}$, the core idea of the Co-ALS algorithm is based on the following formulation:

$$\widehat{\mathbf{X}}^{(i)} = \underset{\mathbf{X}^{(i)}}{\operatorname{argmin}} \left\| \mathcal{Y}_{(i)} - \left( {\mathbf{X}^{(i)}}^{\top} \odot {\mathbf{C}^{(i)}}^{\top} \right)^{\top} {\mathbf{K}^{(i)}}^{\top} \right\|_{\mathrm{F}}. \tag{23}$$

Here, it is worth mentioning that $\mathbf{K}^{(i)}$ is calculated based on the extended factors, i.e., $\left( {\mathbf{X}^{(i)}}^{\top} \odot {\mathbf{C}^{(i)}}^{\top} \right)^{\top}$ instead of $\mathbf{X}^{(i)}$. Afterward, (23) is transformed to be linear and then solved alternately for each mode. In this context, since there is modeling imperfection arising from the assumed $\mathbf{C}_i$ model at the receiver, the step succeeding user separation using Co-ALS is phase correction on the separated signals. Next, the decoding steps to decode the binary message of each user follow those of the state-of-the-art TBM decoder.

*Performance.* For highly frequency-selective channels, and with the same order of complexity as the state-of-the-art TBM, the TBM with the Co-ALS-based decoder (TBM—Co-ALS) has shown outstanding performances compared to TBM and FASURA [105], one of the best, recently proposed schemes in the context

of unsourced random access channels. In Figure 8, we see that TBM—Co-ALS achieve quite good performances, while performances of both TBM and FASURA diverge, since at $K_{\mathrm{a}} = 20$, they fail to achieve the targeted threshold of misdetection rate.

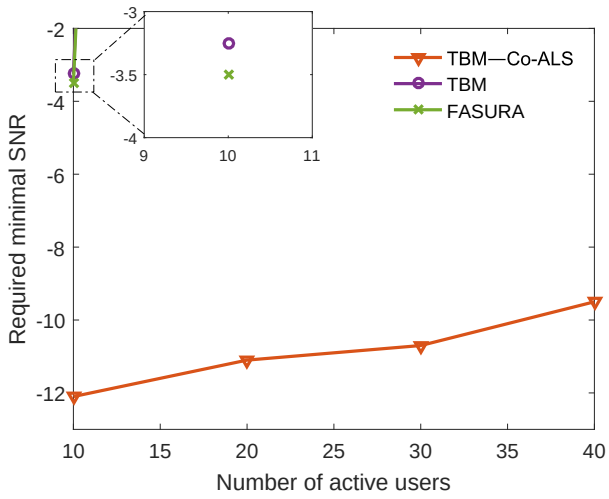



Figure 8: Minimal SNR in dB required to achieve a misdetection rate of 0.1 for the given number of active users. Simulated channel is TDLC-1000 [106] ($L = 24$ taps and delay spread equal to 1 $\mu$s). $T$=1008 time-frequency resources (with two modes signal tensor structure), $N = 8$ antennas, and $B = 88$ bits.

To sum up, the novel tensor-based transceiving scheme effectively handles the distortions brought about by the frequency-selective channels in unsourced random massive access. Structures of both the encoder and the decoder demonstrate significant performance in highly frequency-selective channels at linear/polynomial complexity. However, fine-tuning the hyperparameters of the transceiver could further strengthen its capability to decode an even larger number of active users.

### *3.5. Further tensor applications in wireless communications*

In the literature, tensor-based channel modeling and estimation is among the influential implications of tensor algebra in wireless communications, which should also be mentioned [101, 107, 108]. The MIMO wireless communication channel is basically represented in different domains, that is mainly time, frequency and space. At this point, one can easily deduce the possible tensorial representation or structure. Furthermore, for an advanced representation, especially with certain antenna panel configurations, some domains, essentially the spatial domain, can be further expanded. That expansion additionally involves tensorial structure. Assuming uniform rectangular array antenna structure at both transmitter and receiver, with $(M_{\mathrm{H}} \times M_{\mathrm{V}})$-elements transmitting antenna panel and $(N_{\mathrm{H}} \times N_{\mathrm{V}})$-elements receiving antenna panel, the communication channel, in tensor form, can be expressed as

$$\mathcal{H} = \mathcal{A} \times_1 \mathbf{R}_{\mathrm{H}} \times_2 \mathbf{R}_{\mathrm{V}} \times_3 \mathbf{T}_{\mathrm{H}} \times_4 \mathbf{T}_{\mathrm{V}} \times_5 \mathbf{D} \times_6 \mathbf{F}. \tag{24}$$

$\mathcal{A} \in \mathbb{C}^{L\times L\times L\times L\times L\times L}$ is the diagonal tensor representing the gains of the $L$ channel paths. $\mathbf{R}_{\mathrm{H}} \in \mathbb{C}^{N_{\mathrm{H}}\times L}$ and $\mathbf{R}_{\mathrm{V}} \in \mathbb{C}^{N_{\mathrm{V}}\times L}$ represent the receiver-side spatial domain in azimuth (i.e., the horizontal direction of the antenna panel, denoted with H subscript) and elevation (i.e., the vertical direction of the antenna panel, denoted with V subscript), respectively. The same for $\mathbf{T}_{\mathrm{H}} \in \mathbb{C}^{M_{\mathrm{H}}\times L}$ and $\mathbf{T}_{\mathrm{V}} \in \mathbb{C}^{M_{\mathrm{V}}\times L}$, representing the transmitter-side spatial domain. $\mathbf{D} \in \mathbb{C}^{I\times L}$ is the time domain mode defining $I$ time samples. $\mathbf{F} \in \mathbb{C}^{J\times L}$ is the frequency domain mode characterising $J$ subcarriers. Antenna polarization can also be considered in the channel modeling (24), thereby adding seventh mode to $\mathcal{H}$.

Consistent with the channel tensorial representation, and for a dual-polarized MIMO antenna system, authors in [107] have modeled the channel as a fifth-order tensor, taking advantage of the azimuth and elevation diversities at both transmitter and receiver, as well as the propagation channel's multipath characteristics. This tensor-based channel representation is expressed using the TT format to enable efficient, low-complexity channel estimation. Overall, the authors have proposed a TT-based framework for efficient MIMO channel estimation by addressing high-dimensionality challenges in fifth-order channel tensors, leveraging the JIRAFE (Joint dImensionality Reduction And Factor rEtrieval) method. [108] employs the JIRAFE algorithm with a rectification strategy to design a semi-blind receiver. The work considers a double-directional (channel modeling in both azimuth and elevation directions) dual-polarized MIMO system with tensor space-time coding. In the same vein, and with another example, authors in [101] have developed a tensor-based minimum mean square error (MMSE) channel estimation technique for massive MIMO-OFDM systems. The proposed tensor-MMSE estimator exploits the multilinear structure of the MIMO-OFDM channel, the multidimensional sparsity and structural separability, using a TD for a three-order tensor. In addition, in further works, [59] proposes a ParaTuck$-2$ tensor model for MIMO systems to optimize space-time spreading and multiplexing jointly while using two binary allocation matrices to control data stream assignments across transmit antennas and time slots. In [109], a generalized fourth-order ParaTuck$-2$ tensor model for MIMO communications with space-time-frequency spreading-multiplexing was introduced. In this model, the core of the fourth-order tensor primarily consisted of two third-order interaction tensors. A blind receiver based on the Levenberg-Marquardt algorithm was proposed for this generalized fourth-order ParaTuck$-2$ model.

### *3.6. Challanges and future directions*

With ubiquitous connectivity, tensor algebra application in wireless communication is a prominent and growing area, yet they face several key challenges, outlined below.

**Scalability and complexity.** As networks scale, like massive and extremely large-scale (XL) MIMO and dense networks, tensor methods must manage large-scale data without overwhelming computational resources or introducing latency, especially for real-time needs.

**Algorithmic development.** Despite their potential, tensor-based algorithms in wireless communications remain in the early stages. Most existing tensor models are based on simplified scenarios, and developing algorithms that efficiently handle tasks like signal detection, channel estimation, and beamforming is still an open challenge requiring advanced mathematical tools. Real-world wireless environments are increasingly complex and dynamic, involving interference and mobility, and adapting tensor models to realistically account for these aspects remains a significant challenge to their broader applicability.

**Hardware integration.** Many tensor-based algorithms require significant hardware resources, especially for higher-order tensors. Current hardware architectures may not be optimized for tensor-based operations, particularly in mobile and energy-constrained devices such as IoT devices and edge networks.

These challenges pave the way for promising future developments; below are several key areas to consider:

**Context-specific tensor decomposition techniques.** Considering the specificity of the application, like the structure of the signals constellation and/or time and frequency correlation and other properties, developing efficient tensor decomposition methods, such as PARAFAC and Tucker that are tailored for the given context and environment of wireless communications can improve system performance in areas like source separation and channel estimation.

**Distributed tensor computation.** As wireless networks become more distributed, tensor-based models need to be adapted for decentralized or edge computing environments. Distributed tensor algorithms could be developed to process large-scale data locally while minimizing the communication overhead between nodes.

**6G and beyond applications.** Holographic MIMO, a potential technology in 6G, involves holographic beam patterns for ultra-high precision communication. Tensors can represent the multi-dimensional structure of these beam patterns (e.g., spatial, temporal, frequency domains), making them a good candidate for designing and optimizing holographic MIMO systems. In addition, THz communication, a cornerstone of 6G, involves extremely high frequencies and large bandwidths, leading to highly complex propagation environments and high-dimensional channel state information (CSI) representation. Tensor modeling could help to capture the high-dimensional structure of THz channels, improving channel estimation and interference control.

## 4. Tensor methods for network compression

In recent years, deep neural networks (DNNs) have achieved remarkable progress across a range of fields, including computer vision [110], natural language processing [111], and signal processing [112], to mention a few. Despite their effectiveness, the large number of parameters in DNNs leads to considerable

processing time and memory demands [113]. These high computational costs present a significant challenge for deploying DNNs on edge devices with constrained resources, such as memory, computation, and energy [114]. To address this, researchers have developed various DNN compression techniques, such as pruning [115], quantization [114], knowledge distillation [116], neural architecture search (NAS) [117], and low-rank approximations [118].

Among them, low-rank approximations [110] emerge as an effective method for reducing both the number of parameters and the computational complexity of DNNs. The weights of DNNs, also called parameters, are inherently structured as matrices or higher-order tensors. The primary operations in DNNs—matrix multiplication and convolution—are applied to tensor data as it moves through the network, accounting for a significant portion of the computational load [119]. Low-rank approximations leverage the low-rank nature of the model's weights, factorizing them into smaller factors using either matrix or tensor decompositions. In the following, we review decomposition methods tailored to three popular architectures: Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and Transformer.

### *4.1. Convolutional neural networks*

A convolutional layer is parameterized by the weight tensor $\mathcal{W} \in \mathbb{R}^{O\times I\times K\times K}$, where $O$, $I$ and $K$ represent the number of output channels, input channels, and the kernel size, respectively. $\mathcal{W}$ maps an input tensor $\mathcal{I} \in \mathbb{R}^{H_{\text{in}}\times W_{\text{in}}\times I}$ into an output tensor $\mathcal{O} \in \mathbb{R}^{H_{\text{out}}\times W_{\text{out}}\times O}$, where $H_{\text{in}}$, $W_{\text{in}}$, $H_{\text{out}}$, and $W_{\text{out}}$ are the heights and widths of the input and output tensors, respectively. This mapping is expressed as

$$\mathcal{O}(x, y, o) = \sum_{m=0}^{K-1}\sum_{n=0}^{K-1}\sum_{i=0}^{I-1}\mathcal{I}(x+m, y+n, i)\mathcal{W}(o, i, m, n). \tag{25}$$

Existing low-rank decomposition methods [110, 120, 121, 122, 123, 124, 125] typically involve three main steps. First, the weight tensor $\mathcal{W}$ is factorized into smaller components, such as factor matrices [110, 120, 124, 125] or core tensors [121, 122, 123]. Next, the original convolution operation is replaced with new operators, for instance, lower-dimensional convolutions with reduced complexity [110, 120, 121, 124, 125] or tensor multiplications [122, 123], derived from the factorized representation. Finally, either the network is fine-tuned to recover accuracy [110, 120, 121, 125], or the compact model is trained from scratch [122, 123, 124]. We categorize existing works based on the factorization step, as it represents the key differentiating factor.

The most widely adopted and foundational approach to CNN compression using low-rank matrix approximations is based on SVD [126, 110, 127, 128, 129, 130, 131, 132, 133, 134, 135, 124]. Pioneering approaches [126, 110, 127] applied truncated SVD, where the weight tensor $\mathcal{W}$ was flattened into a second-order matrix $\mathbf{W} \in \mathbb{R}^{O\times IK^2}$. Given a truncated rank $R$, $\mathbf{W}$ was factorized into orthogonal and diagonal matrices, which were then used to replace the original layer with two new sequential convolutional layers of size $R\times I\times K\times K$ and $O \times R \times 1 \times 1$. Zhang *et al.* [110] used the product of the principal component analysis energy of all

layers to determine the truncated rank. Idelbayev *et al.* [128] formulated the rank selection as a combinatorial optimization problem and proposed to solve it with stochastic gradient descent (SGD) during the training process. In ALDS [132], Liebenwein *et al.* proposed an automatic layer-wise decomposition selector using the Eckart–Young–Mirsky theorem. BSR [133] employed a modified beam search that can perform a joint optimization of the rank allocations over all the layers. TRP [129] integrated low-rank approximations and regularization into the training process. SVD was combined with knowledge distillation in LRDKT [127], or with pruning in CC [131], leveraging the advantages of both approaches. PENNI [130] decomposed parameters into tiny sets of basis kernels and accompanying coefficient matrices. LRPET [124] trained low-rank compressed networks from scratch, alternately performing SGD training and projection of each weight matrix onto the corresponding low-rank manifold. Bi-JSVD [134] jointly decomposed similar layers, projecting weights with the same structures into the same subspace.

A natural progression from low-rank matrix approximations to higher-order tensor decomposition methods was seen with the introduction of the CPD in CNN compression [120]. Lebedev *et al.* [120] pioneered this approach by factorizing the 4-order weight tensor $\mathcal{W} \in \mathbb{R}^{O \times I \times K \times K}$ into a sum of rank-1 tensors, expressed as $\mathcal{W} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]\!]$, where $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$ were factor matrices of size $O \times R$, $I \times R$, $K \times R$ and $K \times R$, respectively, and $R$ was the CPD rank. These matrices were used to replace the original layer with four new sequential convolutional layers. Although this method preserved the multidimensional nature of the weight tensor, it processed the tensor at a coarse level, opening up the potential for a more fine-grained treatment. Subsequent CPD-based approaches [136, 137] proposed to reshape $\mathcal{W}$ to a 3-order tensor $\widetilde{\mathcal{W}} \in \mathbb{R}^{O \times I \times K^2}$ with the argument that the kernel size is small enough to ignore. In this case, the rank-$R$ CPD of $\widetilde{\mathcal{W}}$ was $\widetilde{\mathcal{W}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$, where $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ were factor matrices of size $O \times R$, $I \times R$, and $K^2 \times R$, respectively. This resulted in three new sequential convolutional layers of size $R \times I \times 1 \times 1$, $R \times 1 \times K \times K$, and $O \times R \times 1 \times 1$. Although this approach [136, 137] simplified the tensor structure, it compromised the multidimensionality of the weight tensor, potentially causing information loss—particularly in modern architectures that use larger kernel sizes [125]. Recently, Pham *et al.* [125] introduced NORTON, a filter decomposition method. This approach is built on the fundamental insight that stems from the convolution layer's nature, where the input undergoes convolution independently with each filter, and the resulting outputs are aggregated to generate the final feature map. In NORTON, the layer's 4-order weight was interpreted as a set of 3-order filters, and each was then factorized individually. In Fig. 9, we illustrate these three aforementioned CPD-based methods, namely layer decomposition [120], reshaped layer decomposition [136, 137], and filters decomposition [125], respectively. We provide a detailed explanation of the filters decomposition method [125] as a case study in subsection 4.4.

Around the same time that the CPD was introduced, the Tucker decomposition (TD) also made its way into CNN compression, first applied by Kim *et al.* [121]. In this latter, the authors decomposed the weight tensor $\mathcal{W}$ into a core tensor $\mathcal{C} \in \mathbb{R}^{O \times I \times K \times K}$ and two factor matrices $\mathbf{U}^{(1)} \in \mathbb{R}^{O \times R_1}$ and $\mathbf{U}^{(2)} \in \mathbb{R}^{I \times R_2}$,

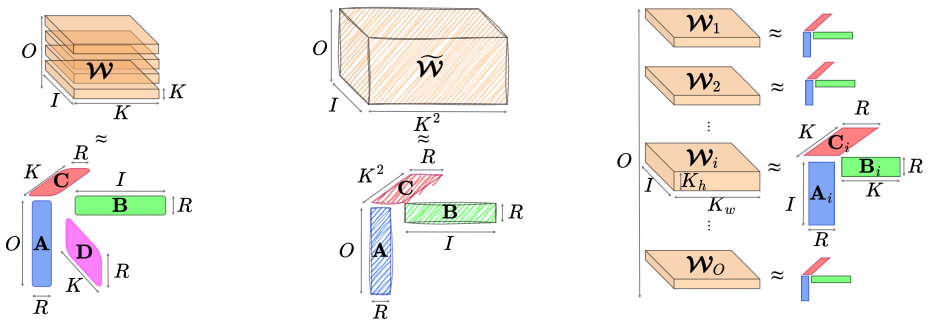


Figure 9: CPD-based methods: layer decomposition [120], reshaped layer decomposition [136], and filters decomposition [125].

given the multilinear rank-$(R_1, R_2)$. $\mathbf{U}^{(2)}$, $\mathcal{C}$, and $\mathbf{U}^{(1)}$ were sequentially used to replace the original layer with three new convolutional layers of size $R_2 \times I \times 1 \times 1$, $R_1 \times R_2 \times K \times K$, and $O \times R_2 \times 1 \times 1$, respectively. Other approaches in this line of work included [138, 139, 140, 141, 142, 118]. Zhong *et al.* [138] introduced ADA-Tucker, an adaptive dimension adjustment TD that compresses the model during training. Yin *et al.* [140] proposed BATUDE, a budget-aware TD-based approach that estimates the ranks during training. Dai *et al.* [118] presented a dual-model training strategy with an iterative and adaptive rank reduction. TD was jointly used with other compression methods such as NAS in HALOC [141] or pruning in [139]. Gabor *et al.* [142] pointed out that TD is often associated with a relatively large core tensor, leading to an increased number of parameters that need to be stored. To mitigate this, they proposed the use of hierarchical TD for CNN compression, where the 4-order core tensor is further decomposed into smaller 3-order core tensors.

As a generalized model that combines CPD and TD, BTD has gained attention for its application in CNN compression [143, 144]. Wang *et al.* [143] employed BTD by first reshaping $\mathcal{W}$ to a 3-order tensor $\widetilde{\mathcal{W}} \in \mathbb{R}^{O \times I \times K^2}$ and then factorizing it into a core tensor $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times R_3 \times K^2}$ and two factor matrices $\mathbf{A} \in \mathbb{R}^{R_1 R_2 \times O}$ and $\mathbf{B} \in \mathbb{R}^{R_1 R_3 \times I}$, where $(R_1, R_2, R_3)$ was the BTD rank. This decomposition was used to create three new sequential convolutional layers of size $R_3 R_1 \times I \times 1 \times 1$, $R_2 R_1 \times R_3 R_1 \times K \times K$, and $O \times R_2 R_1 \times 1 \times 1$. Recently, Phan *et al.* [144] proposed a more compact model that further compresses the BTD by enforcing identical core tensors. To address the numerical instability arising from this shared core structure, they proposed an alternating sensitivity correction method, ensuring the network remains robust while achieving better approximation.

The application of TTD to CNN compression was initially explored by Novikov *et al.* [145] for fully connected (FC) layers. However, directly applying this approach to convolutional layers produced poor experimental results [122]. To address this, Garipov *et al.* [122] proposed a novel reshaping scheme, reordering both the input tensor $\mathcal{I} \in \mathbb{R}^{H_{\text{in}} \times W_{\text{in}} \times I}$ into $\widetilde{\mathcal{I}} \in \mathbb{R}^{H_{\text{in}} \times W_{\text{in}} \times i_1 \cdots \times i_d}$ and the weight $\mathcal{W} \in \mathbb{R}^{O \times I \times K \times K}$ into $\widetilde{\mathcal{W}} \in \mathbb{R}^{(o_1 \times i_1) \times \cdots \times (o_d \times i_d) \times K \times K}$, where $O = \prod_{x=1}^{d} o_x$ and $I = \prod_{x=1}^{d} i_x$. The rank-$(R_0, R_1, \ldots, R_d)$ TTD then factorized $\widetilde{\mathcal{W}}$ into a set of core tensors, including $\mathcal{G}_0 \in \mathbb{R}^{K \times K}$ and $\mathcal{G}_x \in \mathbb{R}^{R_{x-1} \times o_x \times i_x \times R_x}$ for $1 \leq x \leq d$. Addi-

tional contributions in this domain included [146, 147, 148, 149]. For instance, Deng *et al.* [146] introduced TIE, an energy-efficient TTD-based inference engine for DNNs. Yin *et al.* [147] formulated TTD-based model compression as an optimization problem with constraints on tensor ranks, and then iteratively solved it with the Alternating Direction Method of Multipliers (ADMM). In HODEC [148], Yin *et al.* analyzed the underlying reason and mitigation strategy for the inefficiency of the classical TTD-format convolution. Recently, Sui *et al.* [149] introduced CEPD, co-exploring structured sparsification and TTD.

Inspired by the expressive capacity of TRD, Wang *et al.* [123] introduced Tensor Ring Nets (TRN), where the layers of a DNN were compressed through tensor ring factorization. In this method, 4-order weight tensor $\mathcal{W} \in \mathbb{R}^{O\times I\times K\times K}$ was decomposed by neglecting the dimensions associated with the kernel, resulting in three smaller factor tensors $\mathcal{U}^{(1)} \in \mathbb{R}^{R_1\times R_2\times K\times K}$, $\mathcal{U}^{(2)} \in \mathbb{R}^{R_3\times O\times R_2}$, and $\mathcal{U}^{(3)} \in \mathbb{R}^{R_1\times I\times R_3}$, where $(R_1, R_2, R_3)$ was the tensor ring rank. Other TRD-based approaches included [112, 150, 151, 152]. Cheng *et al.* [112] proposed an automatic rank selection scheme inspired by reinforcement learning. Li *et al.* [150] proposed a heuristic genetic algorithm to estimate optimal rank via progressive searching. In a related effort, Obukhov *et al.* [151] highlighted a limitation of the approach by Wang *et al.* [123], which requires estimating a separate TRD for each layer of the network. To address this, they proposed T-Basis, a method that learns a unified basis for the entire network, eliminating the need to factorize the diverse set of weight tensors across layers, each having a distinct tensor ring representation and independent parameters. More recently, Xie *et al.* [152] investigated the impact of the reshaping process and showed that different execution orders of the core tensors result in varying computational complexities.

### *4.2. Recurrent neural networks*

The fundamental operation in conventional components such as Long Short-Term Memory (LSTM) [153] cells and Gated Recurrent Units (GRU) [154] in RNNs is matrix multiplication, typically expressed as $\mathbf{y} = \mathbf{W}\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^N$ is the input vector, $\mathbf{W} \in \mathbb{R}^{M\times N}$ is the weight matrix, and $\mathbf{y} \in \mathbb{R}^M$ is the output vector. To leverage high-order tensor decompositions, existing studies [155, 156, 157, 158, 159] typically follow a three-stage process. First, vectors and matrix are rearranged into tensors $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_d}$, $\mathcal{W} \in \mathbb{R}^{m_1\times m_2\times\cdots m_d\times n_1\times n_2\times\cdots\times n_d}$, and $\mathcal{Y} \in \mathbb{R}^{m_1\times m_2\times\cdots\times m_d}$, with $M = \prod_{i=1}^{d} m_i$ and $N = \prod_{j=1}^{d} n_j$. Next, the weight $\mathcal{W}$ is decomposed into low-rank core tensors using specific tensor decomposition techniques, such as TTD [155], BTD [156], TRD [157], and HTD [158]. Finally, the original matrix multiplication operation is replaced with tensor computations involving the input and the factor tensors, followed by either fine-tuning the network or training from scratch to restore accuracy.

One of the earliest applications of tensor decomposition for RNN compression was the TT-RNN method introduced by Yang *et al.* [155]. To mitigate the challenge of an excessively large weight matrix in the input-to-hidden mapping of RNN models, they proposed factorizing the weight as follows $\mathcal{W}\big((i_1, j_1), \ldots, (i_d, j_d)\big) = \mathcal{G}_1(i_1, j_1)\cdots\mathcal{G}_d(i_d, j_d)$, where $\{\mathcal{G}_k\}_{k=1}^{d}$ was the set of the TTD-core tensors. This approach reduced the

original parameter count from $MN$ to $\sum_{k=1}^{d} m_k n_k r_{k-1} r_k$. Other TTD-based approaches included [160, 161, 162]. Adopting the TTD scheme from [145], Tjandra *et al.* [160] compressed the GRU, while Grachev *et al.* [161] compressed LSTM networks. Recently, Liu *et al.* [162] introduced TEDEPR, a dual-grained lightweight approach that combines TTD with unstructured pruning for LSTM compression.

Seeking a more efficient approximation of RNNs with fewer parameters than the TT-RNN method [155], Ye *et al.* [156, 163] introduced BT-LSTM, a BTD-based approach for tasks such as video action recognition, image captioning, and image generation. This decomposition was expressed as $\mathcal{W} = \sum_{n=1}^{N} \mathcal{G}_n \times_1 \mathcal{A}_n^{(1)} \times_2 \cdots \times_d \mathcal{A}_d^{(d)}$, where $\mathcal{G}_n \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_d}$ was the core tensor, $\mathcal{A}_n^{(d)} \in \mathbb{R}^{N_d \times M_d \times R_d}$ was the factor tensor, $N$ was the CP-rank, $(R_1, R_2, \ldots, R_d)$ was the Tucker-rank, and $d$ was the core-order. To facilitate hyper-parameters setting, the authors set $R_1 = R_2 = \ldots = R_d = R$. The method proposed in [155] had one drawback is that the restricted setting on the ranks and the restrained order of core tensors make the models sensitive to parameter selection [157]. To address this issue, Pan *et al.* [157] introduced TR-LSTM, which uses TRD to extract the low-rank structure of the input-to-hidden matrix in RNNs. Their empirical evaluations on two action recognition datasets, UCF11 and HMDB51, showed superior performance compared to both TT-RNN [155] and BT-LSTM [156].

Another direction in RNN compression is the HTD-based approach [158, 159]. In [158], Yin *et al.* introduced the fully decomposed hierarchical Tucker (FDHT) structure, which achieved greater storage reduction compared to earlier tensor decomposition methods [155, 156], while also improving accuracy for compact RNNs in video-based action recognition tasks. Building on this, Gong *et al.* [159] further developed a customized hardware architecture to efficiently execute the FDHT-LSTM model [158]. Through a carefully designed memory access scheme, the hardware efficiently handled the complex matrix transformations on the fly, eliminating access conflicts during execution [159].

### *4.3. Transformers*

A Transformer block [111] consists of two main components: an attention layer and a feed-forward network (FFN). In the attention layer, there are four weight matrices, which are used for linear projections: query $\mathbf{W}_\mathrm{Q}$, key $\mathbf{W}_\mathrm{K}$, value $\mathbf{W}_\mathrm{V}$, and output $\mathbf{W}_\mathrm{O}$, each of size $d \times d$, where $d$ represents the dimension of an input embedding vector (i.e., the number of features). For an input sequence of length $n$ (i.e., $n$ tokens), the input to the attention layer, $\mathbf{X}$, is an $n \times d$ matrix. Each of the query, key, and value matrices, $\mathbf{X}_\mathrm{Q}$, $\mathbf{X}_\mathrm{K}$, and $\mathbf{X}_\mathrm{V}$, respectively, is computed by multiplying the input matrix with their corresponding weight matrices: $\mathbf{X}_\mathrm{Q} = \mathbf{X}\mathbf{W}_\mathrm{Q}$, $\mathbf{X}_\mathrm{K} = \mathbf{X}\mathbf{W}_\mathrm{K}$, and $\mathbf{X}_\mathrm{V} = \mathbf{X}\mathbf{W}_\mathrm{V}$. After these projections, the attention score, $\mathbf{X}_\mathrm{S}$, is computed as softmax $\left(\mathbf{X}_\mathrm{Q}\mathbf{X}_\mathrm{K}^\mathsf{T}\right)$. Finally, the output of the attention layer, $\mathbf{X}_\mathrm{O}$, is computed by multiplying the attention value $\mathbf{X}_\mathrm{S}\mathbf{X}_\mathrm{V}$ with the output weight matrix $\mathbf{W}_\mathrm{O}$, and adding a residual connection to the input: $\mathbf{X}_\mathrm{O} = \mathbf{X}_\mathrm{S}\mathbf{X}_\mathrm{V}\mathbf{W}_\mathrm{O} + \mathbf{X}$. The FFN processes the output of the attention layer, $\mathbf{X}_\mathrm{O}$. It consists of two FC layers, with corresponding weight matrices $\mathbf{W}_\mathrm{FC1}$ and $\mathbf{W}_\mathrm{FC2}$, of dimensions $d \times h$ and $h \times d$, respectively.

The output of the FFN is computed as $\mathrm{ReLU}\left(\mathbf{X}_{\mathrm{O}}\mathbf{W}_{\mathrm{FC1}}\right)\mathbf{W}_{\mathrm{FC2}} + \mathbf{X}_{\mathrm{O}}$.

Since matrix multiplication is the primary operation in transformer blocks, SVD naturally emerges as a popular method to factorize and compress weight matrices [164, 165, 166, 167, 168, 169]. On the GLUE benchmark dataset, Noach *et al.* [164] applied this technique to compress the BERT-base model by first decomposing the weight matrices into smaller components and then using feature distillation to recover performance. Fisher-weighted SVD [165] enhanced this by leveraging Fisher information to prioritize the parameters critical for model predictions, thereby minimizing the performance drop caused by truncating the decomposed matrices. Hajimolahoseini *et al.* [166] compared two strategies—layer-by-layer and progressive decomposition—on pre-trained transformer-based models like ViT [170] on CIFAR and GPT-2 [171] on Wikitext-103. Guo *et al.* [167] introduced PELA, a method that enhances the parameter efficiency of pre-trained vision transformer models by integrating SVD with feature distillation and weight perturbation. More recently, Gao *et al.* [168] introduced an adaptive rank selection approach to compress BERT [172] and LLaMA-7B [173] on GLUE [174]. Furthermore, LASER [169] selectively factorized the weight matrices of the GPT-J model on the CounterFact dataset, demonstrating performance gains.

Recently, there has been a growing trend toward applying higher-order decomposition methods, such as BTD and TTD, by tensorizing weight matrices in transformer models into higher-order tensors for more effective compression [175, 176, 177, 178, 179, 180]. Ma *et al.* [175] introduced a novel self-attention model that combines BTD with parameter sharing, achieving strong performance on language modeling tasks. Hrinchuk *et al.* [176] used TTD to compress the embedding layers of transformers across four tasks: sentiment analysis, neural machine translation, language modeling, and click-through rate prediction. For vision tasks such as object classification and detection, Zhen *et al.* [177] reshaped the weight matrices of FC layers and then adopted TTD on them, compressing detection transformers [181] on ImageNet [182] and COCO [183]. In another effort, FacT [178] tensorized the weights of vision transformer models into a 3-order tensor, which was then decomposed into lightweight factors for parameter-efficient transfer learning. In T3SRS, Li *et al.* [179] introduced a tensor train layer that decomposes the original weight matrix into smaller tensors, effectively compressing sequential recommender systems. More recently, Yang *et al.* [180] introduced LoRETTA, a parameter-efficient framework that employs TTD to significantly reduce trainable parameters, demonstrating strong performance on BERT-family and LLaMA-2 models [173].

### *4.4. Case study: Hybrid network compression through tensor decompositions and pruning*

In [125], the authors introduced NORTON, for hybrid Network cOmpression thRough Tensor decompOsitions and pruNing. This approach falls under the category of structured pruning methods, and also incorporates low-rank approximations, specifically CPD-based techniques for CNN compression, mentioned in subsubsection 4.1. The rationale behind combining these two strategies stems from the observation that CNN weights exhibit both low-rank and sparse properties. Consequently, integrating tensor decomposition

and pruning aims to further enhance compression efficiency [125]. NORTON begins by decomposing the model into CPDBlocks, which are subsequently pruned using a filter pruning algorithm. The compact model is then fine-tuned. While this process is applied to all convolution layers, for clarity, we focus on a single layer, as shown in Fig. 10. In each layer, each filter is decomposed into three factor matrices. The pruning process then eliminates redundant filters by evaluating the similarities between the factor matrices.

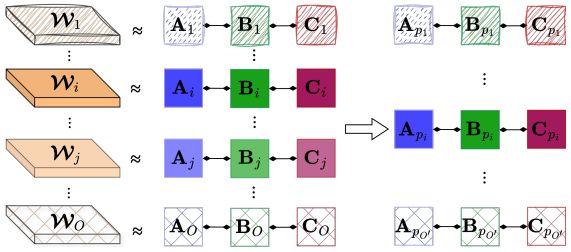


Figure 10: The decomposition then pruning process for one layer.

**Filters Decomposition Using the CPD.** NORTON apply the CPD to each individual filter $\mathcal{W}_k$:

$$\mathcal{W}_k(m,n,p) = \sum_{r=0}^{R-1} \mathbf{A}_k(m,r)\mathbf{B}_k(n,r)\mathbf{C}_k(p,r), \tag{26}$$

where $\mathbf{A}_k$, $\mathbf{B}_k$ and $\mathbf{C}_k$ are 3 factor matrices of size $I \times R$, $K \times R$ and $K \times R$, respectively. This approximation is graphically represented in the left half of Fig. 10. By substituting (26) into (25), we obtain a new CPD-based approach to compute the convolution. This approach involves a sequence of mappings using the factor matrices instead of high-order tensors, resulting in:

$$\mathcal{O}_k(x,y) = \sum_{m=0}^{I-1}\sum_{n=0}^{K-1}\sum_{p=0}^{K-1}\sum_{r=0}^{R-1} \mathcal{I}(x+n,y+p,m)\mathbf{A}_k(m,r)\mathbf{B}_k(n,r)\mathbf{C}_k(p,r). \tag{27}$$

Starting from (27), we observe that the CPD-based convolution involves element-wise multiplications between the input tensor $\mathcal{I}$ and the factor matrices $\mathbf{A}_k$, $\mathbf{B}_k$, and $\mathbf{C}_k$. We can perform this computation as a sequential block of convolutions with smaller kernels, followed by a summation:

$$\mathcal{O}_k^{\mathbf{A}}(x+n,y+p,r) = \sum_{m=0}^{I-1} \mathcal{I}(x+n,y+p,m)\mathbf{A}_k(m,r), \tag{28}$$

$$\mathcal{O}_k^{\mathbf{B}}(x+n,y,r) = \sum_{n=0}^{K-1} \mathcal{O}_k^{\mathbf{A}}(x+n,y+p,r)\mathbf{B}_k(n,r), \tag{29}$$

$$\mathcal{O}_k^{\mathbf{C}}(x,y,r) = \sum_{p=0}^{K-1} \mathcal{O}_k^{\mathbf{B}}(x+n,y,r)\mathbf{C}_k(p,r), \tag{30}$$

$$\mathcal{O}_k(x,y) = \sum_{r=0}^{R-1} \mathcal{O}_k^{\mathbf{C}}(x,y,r), \tag{31}$$

where $\mathcal{O}_k^{\mathbf{A}} \in \mathbb{R}^{H_{\text{in}} \times W_{\text{in}} \times R}$, $\mathcal{O}_k^{\mathbf{B}} \in \mathbb{R}^{H_{\text{in}} \times W_{\text{out}} \times R}$, and $\mathcal{O}_k^{\mathbf{C}} \in \mathbb{R}^{H_{\text{out}} \times W_{\text{out}} \times R}$. It is worth noting that with a suitable rank $R$, the sum of the computational complexities of the operations presented in equations (28), (29), (30), and (31) is smaller than the computational complexities of the original convolution outlined in equation (25). Additionally, equations (28), (29), and (30) can be seen as convolutions and can be implemented using common deep learning frameworks. Specifically, equation (28) can be computed using a classical 2D convolution operation, while equations (29) and (30) can be computed via group convolutions, as illustrated in Fig. 11.

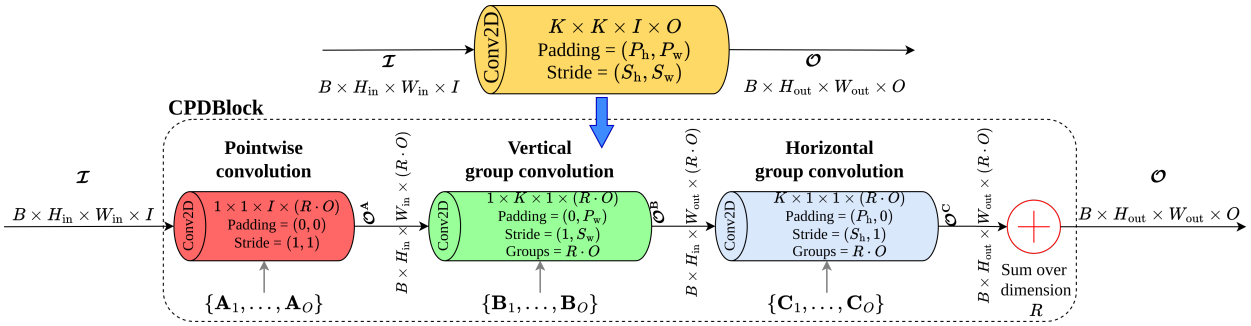



Figure 11: Visualization depicting the structure of CPDBlock [125].

**CPDBlock Pruning.** In equations (28)-(31), $\mathcal{O}_k$ is computed based on three factor matrices: $\mathbf{A}_k$, $\mathbf{B}_k$, and $\mathbf{C}_k$. Hence, when pruning the kernel associated with a specific output, the removal of all three matrices must be considered. This requires employing a pruning criterion that accounts for the interdependencies among these three matrices. Additionally, when the CPD is unique, it is unique up to scaling and permutation ambiguities. To adress this, NORTON opted for Principal Angles Between Subspaces (PABS) [184] as a metric to measure the distance between two CPDs. The use of PABS is justified in both unique and non-unique cases of CPD. The fundamental concept behind CPDBlock pruning is to create a distance matrix $\mathbf{D}$, where each element $\mathbf{D}_{ij}$ corresponds to the distance between the factor matrices $[\![\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i]\!]$ and $[\![\mathbf{A}_j, \mathbf{B}_j, \mathbf{C}_j]\!]$. The pruning process involves iteratively identifying the pair of decompositions, $i$ and $j$, with the minimum value of $\mathbf{D}_{ij}$ and removing one of them. In this strategy, the decompositions between $i$ and $j$ are iteratively assessed, and the one that most closely resembles the rest of the decompositions is removed. This ensures that the pruned one retains a representation most similar to the remaining ones. The distance matrix $\mathbf{D} \in \mathbb{R}^{O \times O}$ can be expressed as $\mathbf{D}_{ij} = \alpha \mathbf{D}_{ij}^{\mathbf{A}} + \beta \mathbf{D}_{ij}^{\mathbf{B}} + \gamma \mathbf{D}_{ij}^{\mathbf{C}}$, where $\mathbf{D}_{ij}^{A} = \phi(\mathbf{A}_i, \mathbf{A}_j)$ (similarly for $\mathbf{D}_{ij}^{B}$ and $\mathbf{D}_{ij}^{C}$), and $\alpha$, $\beta$, and $\gamma$ are weight parameters whose sum is equal to 1.

## 5. Tensor methods for data imputation

In today's digital age, vast data is generated through smartphones, wearables, social media, and ground and airborne sensors. Analyzing this data has become important across many fields, including healthcare,

agriculture, finance, e-commerce, and logistics. For instance, healthcare data can support personalized treatment, and agricultural insights can improve crop yield and manage risks [185, 4]. However, real-world datasets are often noisy and incomplete owing to various factors, such as broken sensors, limited access permissions and intentional partial sampling to meet time/memory requirements. To avoid biased insights and invalid conclusions drawn from incomplete data, *completion*—the process of estimating missing values with meaningful substitutes—is essential [36]. We will briefly discuss approaches for estimating missing entries in matrices and tensors.

### *5.1. Matrix Completion*

Matrix completion estimates missing entries in a partially observed data matrix by assuming properties such as minimum volume or low rank to reduce degrees of freedom and enable a unique solution. The principle of parsimony (Occam's razor) guides the selection of low-complexity models, making the *low-rank* constraint a natural choice for many applications. There are two main approaches to low-rank completion: *rank minimization* and *decomposition-based*. In rank minimization, rank information is used implicitly and treated as a numerical value to be minimized while satisfying the observation constraints. In contrast, decomposition-based methods compute a low-rank factorization for a specific, fixed rank, which is often meaningful in real-life applications where changing the rank may not be permissible. Once the decomposition is computed, its factors can be used to infer missing entries and interpret the data.

#### *5.1.1. Rank minimization approach*

The rank minimization approach aims to find a matrix $\hat{\mathbf{X}} \in \mathbb{R}^{I \times J}$ with the minimum rank that matches the observed entries of an incomplete matrix $\mathbf{X} \in \mathbb{R}^{I \times J}$ within a certain tolerance:

$$\min_{\hat{\mathbf{X}}} rank(\hat{\mathbf{X}}), \text{ s.t. } \|P_{\text{obs}}(\mathbf{X} - \hat{\mathbf{X}})\|_F^2 \leq \epsilon, \tag{32}$$

where the observation operator $P_{\text{obs}}$ for a matrix $\mathbf{X}$ is defined as $P_{\text{obs}}(\mathbf{X})_{i,j} = \mathbf{X}_{i,j}$ if the entry at index $(i, j)$ is observed, and 0 otherwise. The problem in (32) is non-convex and NP-hard [186, 187], but it can be relaxed using convex surrogates, with the nuclear norm being the tightest approximation of matrix rank [187]. Convex optimization techniques (e.g., interior point and semidefinite programming) solve this relaxed problem. If certain sampling conditions are met, a generic low-rank matrix can be perfectly recovered with high probability using a convex program [188, 70, 189]. The probabilities are typically derived under the assumption that $P_{\text{obs}}$ is a random sampler, though observation patterns can also be deterministic or fixed depending on the application. An efficient algorithm known as the singular value thresholding algorithm was proposed to solve nuclear norm minimization-based matrix completion [190], and alternative approaches have been explored as well [191, 192].

*5.1.2. Decomposition-based approach*

This approach explicitly imposes the low-rank matrix decomposition model on a partially observed matrix with a fixed rank and minimizes the error to fit the model parameters:

$$\min_{\mathbf{U},\mathbf{V}} \|P_{\text{obs}}(\mathbf{X} - \hat{\mathbf{X}})\|_F^2, \text{ s.t. } \hat{\mathbf{X}} = \mathbf{U}\mathbf{V}^\top \text{ and } \text{rank}(\hat{\mathbf{X}}) = R. \tag{33}$$

The factor matrices $\mathbf{U} \in \mathbb{R}^{I\times R}$ and $\mathbf{V} \in \mathbb{R}^{J\times R}$ are model parameters. The rank of the completed matrix is often known in applications like structure-from-motion and network localization. Despite requiring rank information, minimizing the Frobenius norm objective is advantageous due to its simplicity, differentiability, and compatibility with gradient-based methods (e.g., gradient descent and manifold optimization). Two main approaches exist for model fitting when rank information is available.

*5.1.3. Gradient-based optimization methods*

These methods begin by initializing the model parameters and then minimizing the loss function using gradients and/or Hessians to update the model parameters. Methods using (approximate) Hessian information, like Newton and quasi-Newton, converge near-quadratically but are computationally expensive due to the need for computation and inversion of Hessian [193, 194]. Gradient-based approaches are typically little restrictive in terms of sampling pattern, although some basic requirements need to be satisfied (see, e.g., [195, 196]). They also facilitate the incorporation of constraints (e.g., non-negativity, orthogonality, or autoregressive constraints) on decomposition parameters (see, e.g., [197]). Several techniques have been proposed for low-rank decomposition-based matrix completion, including gradient-based alternating minimization and manifold optimization methods [198, 199, 195, 200].

*5.1.4. Algebraic methods*

These methods exploit the low-rank structure to design algorithms that rely solely on standard numerical linear algebra (NLA) operations. These algorithms are fast and are guaranteed to work under reasonable deterministic conditions on the observation pattern. By way of example, consider a partially observed rank-$R$ matrix $\mathbf{X}$, which consists of observed submatrices $\mathbf{X}_1$ and $\mathbf{X}_2$ that share an overlap matrix $\mathbf{X}_\text{O}$ as depicted in Figure 12. It has been shown that the completion is necessarily unique iff $\text{rank}(\mathbf{X}_1) = \text{rank}(\mathbf{X}_2) = \text{rank}(\mathbf{X}_\text{O}) = R$ [201, 202, 203].

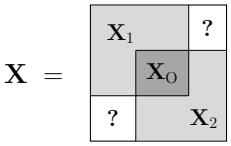



Figure 12: Partially observed matrix with overlapping observations.

A novel algebraic completion method for (such) a low-rank matrix was proposed, and convincing numerical experiments on structure-from-motion problems were discussed in [204, 205, 203]. This approach requires obtaining subspaces of small observed submatrices, which can be computed by SVD. These (local) subspaces are then combined to compute the full reconstruction, which can again be done by means of standard NLA. Therefore, it is fast and has the potential to be used in large-scale applications. The solutions of algebraic methods can be used directly (in noiseless cases) or as a good initialization for optimization methods. A more general approach to low-rank matrix completion, with extensive discussion on recovery guarantees, has also been studied (see, e.g., [206, 207, 203]).

### *5.2. Tensor completion*

Completion is crucial for tensors for several reasons. Higher-order datasets are larger, increasing the likelihood of missing or unreliable entries. Large-scale data often exhibit low-rank properties (intuitively, not every entry "matters", unlike in small matrices). As tensor order increases, computational complexity grows exponentially (curse of dimensionality). To mitigate this, an appropriate multilinear (low-parametric) representation is used, preserving multi-way information while making the problem more tractable. Multilinear algebra offers tools such as tensor decompositions that can uncover patterns in data undetected by other methods and tame the curse of dimensionality (see, e.g., the review papers [208, 209] for more details).

### *5.3. Rank minimization approach*

Researchers have been investigating tensor extensions of the rank minimization problem in (32). A difficulty is that, for tensors, the concept of rank can be defined in different, non-equivalent ways (CP rank, ML rank, TT rank). The nuclear norm relaxation of the CP rank is far from obvious, although some results have been obtained [71, 210, 211, 212]. As ML rank and TT rank can be expressed in terms of the basic matrix rank of certain unfoldings of a given tensor, researchers have primarily focused on convex relaxations of the latter. Nuclear norm extensions have been studied in [213, 214]. Applications to tensor completion have been discussed in [215, 216, 37, 38].

### *5.4. Decomposition-based approach*

Like the matrix case, one explicitly imposes a tensor decomposition model on a partially observed tensor with a fixed rank. Then, one minimizes the loss function, typically the Frobenius norm objective, to fit the model parameters. Problem (33) is extended as:

$$\begin{aligned} &\min_{\boldsymbol{\theta}} \|\mathcal{P}_{\text{obs}}(\mathcal{X} - \hat{\mathcal{X}})\|_F^2, \text{ with } \hat{\mathcal{X}} \text{ in CPD//Tucker //TT format,} \\ &\text{s.t. } \operatorname{rank}_{\text{CP}}(\hat{\mathcal{X}}) = R // \operatorname{rank}_{\text{ML}}(\hat{\mathcal{X}}) = (R_1, \ldots, R_d) // \\ &\qquad \operatorname{rank}_{\text{TT}}(\hat{\mathcal{X}}) = (R_0, \ldots, R_d), \end{aligned} \tag{34}$$

where $\boldsymbol{\theta}$ denotes the decomposition parameters (factor matrices and/or core tensors) and $\mathcal{P}_{\text{obs}}$ the observation operator. The choice of tensor decomposition depends on the goal and the data at hand. For instance, CPD can be used to identify the components in the data, while Tucker decomposition is often used for compression purposes, except in the case of higher-order tensors, where TT decomposition is a feasible option. The decomposition-based approach can be much more memory-efficient than the rank minimization approach because it only requires storing the model parameters. Furthermore, the model parameters may be used to interpret the data in a similar way as when they had been fully observed.

#### *5.4.1. Gradient-based optimization methods*

Analogous to the matrix case, various gradient-based algorithms have been studied, including the popular alternating least-square (ALS) technique, where model parameters are updated alternately (see, e.g., [217, 218, 219, 220, 221]). However, the per-iteration cost of the Gauss-Newton optimization technique, which has superior convergence properties, has been reduced to the order of the per-iteration cost of ALS, thanks to careful numerical exploitation of rank-1 structures in the tensor case. Moreover, this approach has been extended to allow constraints on and symmetries between factor matrices. See [222, 11] and references therein.

#### *5.4.2. Algebraic methods*

Tensor settings offer better compression and denoising capabilities than matrix settings and have recently shown promise in low-rank completion problems. Unlike matrices, where missing row(s)/column(s) lead to underdetermined problems, tensors can still be completed even with missing fibers along one mode. Even more, under reasonable conditions, CPD and MLSVD can still be computed using only standard NLA techniques [223, 224, 203]. This highlights the value of fiber-wise observations, where fibers along one mode are either completely observed or entirely missing, as opposed to the standard entry-wise way of observing. Let us assume that fibers are fully observed/completely missing along mode-$n$, then it is easy to estimate the mode-$n$ subspace of the tensor: we just need to estimate the common subspace for the observed fibers. It can further be shown that, after estimating the mode-$n$ subspace, the rest of the completion task essentially involves simple matrix completion problems (similar to the example discussed in Section 5.1.4) for matrix unfoldings of the incomplete tensor [223, 224, 203].

At second view, such fiber-wise observations (along one mode) are, in fact, of great interest. Indeed, in applications, it often happens that fibers can be acquired much more easily along one mode than along other modes. The "easy" mode may, for instance, correspond to the usual sampling in time, while "more difficult" modes may, for instance, require access to physical locations that are hard to reach. For example, consider weather time series data collected across spatial coordinates (latitude, longitude) [225]. Another example of a fiber-wise observation occurs when acquiring traffic speed data (road segment, day, time window). For

instance, this pattern arises when recording data for specific combinations of road segments and days [226]. Other examples include chemical kinetics data, where obtaining samples along a temporal mode is easier and cheaper than varying the concentrations (see, e.g., [203, 224, 227]).

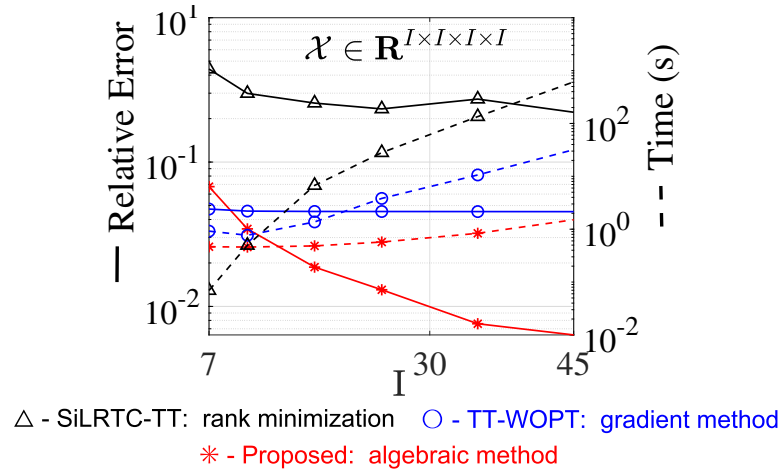



Figure 13: The relative error (solid lines) of the proposed algorithm is orders of magnitude lower than that of TT-WOPT and SiLRTC-TT. At the same time, the computational cost (dashed lines) is also orders of magnitude lower for larger tensors.

### *5.5. Case Study: Tensor completion of multi-way data observed along one mode*

Previous work showed that CPD and MLSVD of fiber-wise observed tensors could be computed using standard NLA techniques [223, 224, 203]. Given the advantages of TT decomposition over CPD and MLSVD, extending the fiber-wise completion in the TT format became consequential. This section discusses key results from the EUSIPCO 2024 paper [227], introducing an algebraic algorithm for computing the TT decomposition of an incomplete tensor observed fiber-wise along a single mode. Note that the approach in the paper differs from the popular TT cross-approximation technique, where a subset of fibers are observed across *all* modes [9]. The TT decomposition that is obtained by completion may, in a subsequent step, be used as a "proxy" to perform further computations efficiently (e.g., fitting a nonnegative CPD to TT approximation rather than the actual tensor) [228].

One essential step in computing the TT decomposition is finding the column spaces of matrix unfoldings. For fiber-wise observed tensors, the proposed algorithm (Algorithm 1 in [227]) computes the $R$-dimensional column space from a set of fully observed rank-$R$ submatrices of the respective matrix unfolding. Specifically, the orthogonal complements of the fully observed submatrices are zero-padded and concatenated along the ambient dimension, forming columns orthogonal to the desired subspace. Under certain overlapping conditions, similar to those discussed in Section 5.1.4, the kernel of the concatenated matrix is ensured to be of minimal dimension, i.e., $R$-dimensional. Therefore, the desired subspace is estimated as the space most orthogonal to its column space. After computing the column spaces of matrix unfoldings in this manner,

the proposed method utilizes them to compute the cores of the TT decomposition. For more details, please refer to [227].

The proposed algorithm uses only standard NLA tools and, therefore, is fast. For illustration, consider an example from Section IV of [227], where fourth-order tensors $\mathcal{X} \in \mathbb{R}^{I \times I \times I \times I}$ are generated with a fixed $\text{rank}_{\text{TT}}(\mathcal{X}) = (1, 3, 3, 4, 1)$, with entries drawn from a standard normal distribution. Gaussian noise was added to the generated tensor to achieve an SNR of 25 dB, and 35% of the mode-4 fibers were randomly removed. The proposed algorithm was run along with TT-WOPT (gradient method)[220] and SiLRTC-TT (rank minimization) [214] to compute the completion (see [227] and references therein for details). The computation time and relative error are compared in Figure 13.

### *5.6. Challenges, Future Directions*

Our completion algorithms for tensors observed along one mode [223, 224, 203, 227] could be extended to settings where data come in streams [229]. As mentioned, the algebraic methods exploit low-rank structures in a specific algebraic manner, and therefore, other types of observation patterns could be explored for which algebraic algorithms can be developed. For instance, banded, block Toeplitz, and overlapping block patterns could be explored [230, 201, 231].

## 6. Discussion

In this paper, we have highlighted the wide-ranging applications of tensor methods in fields such as computer vision, image processing, remote sensing, wireless communication, network compression, data imputation. We review state-of-the-art methods, highlighting their strengths and limitations, comparing them to traditional approaches, and outlining future challenges and research directions in the field. For a detailed discussion, please refer to the corresponding sections. In summary, Section 2 explores tensor methods for joint HSR and spectral unmixing, highlighting their advantages over traditional approaches. This field offers several promising future directions, including improving tensor-based hyperspectral image analysis for enhanced spectral-spatial feature extraction, enabling scalable tensor computations on cloud platforms for real-time remote sensing applications, and developing tensor methods that integrate multi-resolution and multi-temporal satellite data for more effective environmental monitoring. Section 3 defines key concepts in wireless communication and explores how tensor algebra models multi-way relationships and represents high-dimensional signals efficiently in the URA pattern with multi-antenna systems, where the number of devices, antennas, and communication channels is growing and highlights the rising importance of tensor decomposition for tasks such as user separation and channel estimation. In line with that, Tensor-Based Modulation is presented as a promising scheme for massive URA, demonstrating strong performance in multi-antenna configuration compared to traditional approaches. The section also touches on

the challenges tensor algebra faces in wireless communication, including scalability with growing networks and real-time processing demands, the need for more advanced algorithms to handle real-world conditions like interference and mobility, and hardware limitations for high-order tensor operations in mobile or energy-constrained devices. Future directions include context-specific tensor decomposition for improving system performances in source separation and channel estimation, distributed tensor computation for edge commuting, and 6G applications, where tensor models can optimize holographic MIMO and THz communication in high-dimensional environments. Section 4 examines the high computational and memory demands of DNNs, driven by their large parameter count, which limits deployment on resource-constrained edge devices. To address this challenge, the section reviews various compression techniques, with a focus on tensor representations and decompositions that exploit redundancy in high-dimensional structures, enabling structured compression while preserving model performance. It also discusses decomposition methods tailored for CNNs, RNNs, and Transformers. Future directions in this area include network compression through structured sparsity, where tensor decomposition techniques enhance model interpretability and efficiency by enforcing structured sparsity. Another key direction is hardware acceleration, which focuses on optimizing tensor-based compression for emerging hardware such as FPGAs and neuromorphic processors. Additionally, adaptive decomposition can be explored to develop tensor factorization strategies that dynamically adjust compression rates based on input characteristics. Section 5 explores the challenge of incomplete and noisy real-world datasets resulting from factors such as broken sensors, restricted access, and intentional partial sampling, making data completion essential for reliable insights. The section discusses both matrix- and tensor-based approaches, including numerical optimization and algebraic methods. It emphasizes that by exploiting low-rank structures in a specific algebraic way, we can develop better and faster algorithms with deterministic recovery guarantees. This enables the effective completion of high-dimensional, large-scale datasets. Future directions in data imputation include handling highly sparse data by enhancing tensor completion algorithms for scenarios with extreme sparsity, which is particularly relevant in healthcare and genomics. Another key direction is integrating uncertainty estimation by incorporating Bayesian or probabilistic tensor methods to quantify uncertainty in imputed values. Additionally, multi-source data fusion presents an important avenue, extending tensor-based imputation to heterogeneous datasets that combine structured and unstructured missing data.

However, it is important to note that tensor methods extend far beyond these areas. They are increasingly being applied in a variety of other domains, such as neuroscience, genomics, recommender systems, social network analysis, psychometrics, cheometrics, genomics, multi-omics data analysis among others [5, 232, 233, 18, 234, 235, 236]. The versatility and effectiveness of tensor-based approaches in handling multi-dimensional data make them invaluable across many scientific and engineering disciplines. As advancements in technology and data collection continue to evolve, it is expected that the scope of tensor applications will expand even further, offering solutions to increasingly complex challenges in diverse fields. Before concluding

this section, we briefly touch on two emerging areas—quantum programming and omics data analysis—that are not covered in detail in this paper.

Tensor methods are becoming increasingly popular in quantum computing. Executing quantum algorithms often requires building complex quantum circuits with numerous multi-qubit gates, making practical implementation challenging. Tensor methods have gained traction in quantum programming by enabling efficient representation, simulation, and optimization of quantum circuits. Tensor networks, such as Matrix Product States and Projected Entangled Pair States, help control the exponential growth of quantum state space, while tensor decompositions, including TT decomposition, compress high-dimensional states for scalable simulations [235]. Tensor methods enhance Variational Quantum Algorithms by improving the efficiency of classical optimization, reducing computational costs, and enabling scalable simulations of quantum circuits [237]. Tensor networks have proven effective for storing and processing quantum states while also being valuable in classical machine learning, making them an increasingly suitable tool for quantum machine learning [238]. While tensor methods in quantum computing and quantum machine learning are expected to become even more prominent in the future, their development is still in its early stages, requiring further research in many directions. A detailed discussion, however, is beyond the expertise of the authors and the main scope of this paper.

Tensor methods are also gaining significant traction in the analysis of multi-omics data. The rapid advancements in high-throughput sequencing have enabled the collection of vast amounts of biological data across different omics layers, spanning genomics, transcriptomics, proteomics, metabolomics, epigenomics [239, 240]. However, integrating and analyzing these inherently multi-dimensional datasets pose significant computational and statistical challenges [241]. Tensor-based approaches provide a powerful mathematical framework for capturing complex molecular interactions by modeling multi-way relationships among genes, samples, and omics types [242]. Unlike traditional pairwise correlation methods or matrix factorization techniques, tensor decompositions allow for a more structured representation of omics data, preserving multi-dimensional dependencies. For instance, non-negative CPD has been used in MONTI [243] to jointly analyze gene expression, DNA methylation, and miRNA expression across various cancer types, revealing biologically meaningful regulatory mechanisms such as the methylation-driven regulation of the gene EXOC6 in Basal-subtype breast cancer. Other notable applications include multi-sample multi-tissue genomics analysis [236], where tensor methods are used to uncover tissue-specific gene expression patterns, and cell-cell interaction analysis [244], which leverages tensor decomposition to infer ligand-receptor communication networks from transriptomics data. Future directions in this domain include integrating tensor methods with deep learning for improved feature extraction, developing scalable algorithms to handle ultra-high-dimensional omics data, and enhancing interpretability by incorporating domain-specific constraints into tensor factorization models. These advancements will further establish tensor approaches as essential tools in computational biology, enabling more precise and holistic insights into cellular processes and disease

mechanisms.

In conclusion, tensor methods are essential across various domains and play a key role in advancing interdisciplinary research. As datasets continue to expand in numerous fields, the need for advanced techniques to reveal underlying structures and patterns grows. The ability of tensors to handle complex, multi-dimensional data makes them a powerful tool in high-dimensional signal processing. However, fully realizing their potential requires the continued development of both methodologies and computational tools, along with ongoing collaboration across disciplines. Fostering this interdisciplinary collaboration is crucial to ensuring that tensor-based approaches evolve and remain effective in addressing the challenges posed by modern signal processing tasks.

## 7. Acknowledgments

This work is the result of a collaboration initiated at the 32nd European Signal Processing Conference (EUSIPCO 2024) in Lyon, France. Several case studies featured in this paper were first presented during the special session on "Advances in Tensor Theory and its Applications", which played a significant role in shaping its content.

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

## References

[1] X. Chen, D. W. K. Ng, W. Yu, E. G. Larsson, N. Al-Dhahir, R. Schober, Massive access for 5g and beyond, IEEE Journal on Selected Areas in Communications 39 (3) (2021) 615–637. doi:10.1109/JSAC.2020.3019724.

[2] F. Ali, E.-S. Shaker, R. I. S.M., A. Amjad, A. Muhammad, I. Muhammad, K. Kyung-Sup, An intelligent healthcare monitoring framework using wearable sensors and social networking data, Future Generation Computer Systems 114 (4) (2021) 23–43.

[3] J. Li, Y. Pei, S. Zhao, R. Xiao, X. Sang, C. Zhang, A review of remote sensing for environmental monitoring in china, Remote Sensing 12 (2020) 1130.

[4] S. Wolfert, L. Ge, C. Verdouw, M.-J. Bogaardt, Big data in smart farming–a review, Agric. Syst 153 (2017) 69–80.

[5] T. G. Kolda, B. W. Bader, Tensor decompositions and applications, SIAM Rev. 51 (3) (2009) 455–500. doi:10.1137/07070111X.

[6] I. Domanov, L. De Lathauwer, Canonical polyadic decomposition of third-order tensors: Relaxed uniqueness conditions and algebraic algorithm, Linear Algebra Appl. 513 (2017) 342–375.

[7] I. Domanov, L. De Lathauwer, From computation to comparison of tensor decompositions, SIAM J. Matrix Anal. Appl. 42 (2) (2021) 449–474.

[8] I. V. Oseledets, E. E. Tyrtyshnikov, Breaking the curse of dimensionality, or how to use svd in many dimensions, SIAM J. Sci. Comput. 31 (5) (2009) 3744–3759.

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

[10] B. N. Khoromskij, O (d log n)-quantics approximation of n-d tensors in high-dimensional numerical modeling, Constructive Approximation 34 (2011) 257–280.

[11] N. Vervliet, O. Debals, L. Sorber, L. De Lathauwer, Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis, IEEE Signal Process. Mag. 31 (5) (2014) 71–79.

[12] A. Cichocki, N. Lee, I. Oseledets, A.-H. Phan, Q. Zhao, D. P. Mandic, Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions, Found. Trends Mach. Learn. 9 (4–5) (2016) 249–429.

[13] I. V. Oseledets, Tensor-train decomposition, SIAM J. Sci. Comput. 33 (5) (2011) 2295–2317.

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

[15] J. I. Cirac, D. Perez-Garcia, N. Schuch, F. Verstraete, Matrix product states and projected entangled pair states: Concepts, symmetries, theorems, Rev. Mod. Phys. 93 (4) (2021) 045003.

[16] W. Hackbusch, Tensor spaces and numerical tensor calculus, Vol. 42, Springer, 2012.

[17] C. Prévost, V. Leplat, Minimum-volume non-negative block-term decomposition: Blind data fusion and unmixing with estimation of the number of endmembers, in: 2024 32nd European Signal Processing Conference (EUSIPCO), IEEE, 2024, pp. 1057–1061.

[18] I. Jung, M. Kim, S. Rhee, S. Lim, S. Kim, Monti: A multi-omics non-negative tensor decomposition framework for gene-level integrative analysis, Frontiers in Genetics 12 (2021) 682841. doi:h10.3389/fgene.2021.682841.

[19] S. Kharbech, A. Decurninge, Massive unsourced random access with tensor-based modulation and multi-antenna receiver over frequency-selective channels, in: EUSIPCO 2024 - 32nt European Signal Processing Conference, 2024, pp. 1077–1081.

[20] S. Di Zenzo, A note on the gradient of a multi-image, Computer vision, graphics, and image processing 33 (1) (1986) 116–125.

[21] W. Forstner, E. Gulch, A fast operator for detection and precise location of distinct points, corners and circular features, in: Fast Processing of Photogrammetric Data, 1987, pp. 281–305.

[22] J. A. Noble, Finding corners, Image Vis. Comput. 6 (2) (1988) 121–128.

[23] C. G. Harris, M. J. Stephens, A combined corner and edge detector, in: Alvey Vision Conference, 1988, pp. 147–151.

[24] K. Rohr, Localization properties of direct corner detectors, Mathematical Imaging and Vision 4 (1994) 139–150.

[25] G. Sapiro, D. L. Ringach, Anisotropic diffusion of multivalued images with applications to color filtering, IEEE transactions on image processing 5 (11) (1996) 1582–1586.

[26] J. Weickert, Coherence-enhancing diffusion filtering, International journal of computer vision 31 (1999) 111–127.

[27] D. Tschumperlé, R. Deriche, Vector-valued image regularization with pdes: A common framework for different applications, IEEE transactions on pattern analysis and machine intelligence 27 (4) (2005) 506–517.

[28] J. Bigün, Pattern recognition in images by symmetries and coordinate transformations, Computer vision and image understanding 68 (3) (1997) 290–307.

[29] J. Van de Weijer, T. Gevers, J.-M. Geusebroek, Edge and corner detection by photometric quasi-invariants, IEEE transactions on pattern analysis and machine intelligence 27 (4) (2005) 625–630.

[30] M. Donoser, S. Kluckner, H. Bischof, Object tracking by structure tensor analysis, in: 2010 20th International Conference on Pattern Recognition, IEEE, 2010, pp. 2600–2603.

[31] L. P. Lavorante, H. D. Ebert, Tensor3d: A computer graphics program to simulate 3d real-time deformation and visualization of geometric bodies, Computers & geosciences 34 (7) (2008) 738–753.

[32] W. Wang, X. Liu, L. Liu, Upright orientation of 3d shapes via tensor rank minimization, Journal of Mechanical Science and Technology 28 (2014) 2469–2477.

[33] Y. Ren, L. Liao, S. J. Maybank, Y. Zhang, X. Liu, Hyperspectral image spectral-spatial feature extraction via tensor

principal component analysis, IEEE Geoscience and Remote Sensing Letters 14 (9) (2017) 1431–1435.

[34] R. S. J. Estépar, Local structure tensor for multidimensional signal processing: applications to medical image analysis, Presses univ. de Louvain, 2007.

[35] D. Cherifi, A. Chellouche, A. A. Ouakli, Y. Benamara, R. Deriche, Tensor estimation and visualization using dmri, in: 2013 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA), IEEE, 2013, pp. 61–68.

[36] R. J. Little, D. B. Rubin, Statistical analysis with missing data, Vol. 793, John Wiley & Sons, 2019.

[37] J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2012) 208–220.

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

[39] P. Qiu, Embracing the dropouts in single-cell rna-seq analysis, Nature communications 11 (1) (2020) 1169. doi:10.1038/s41467-020-14976-9.

[40] R. Hodos, P. Zhang, H.-C. Lee, Q. Duan, Z. Wang, N. R. Clark, A. Ma'ayan, F. Wang, B. Kidd, J. Hu, et al., Cell-specific prediction and application of drug-induced gene expression profiles, in: PACIFIC SYMPOSIUM ON BIOCOMPUTING 2018: Proceedings of the Pacific Symposium, World Scientific, 2018, pp. 32–43.

[41] R. Hodos, P. Zhang, H.-C. Lee, Q. Duan, Z. Wang, N. R. Clark, A. Ma'ayan, F. Wang, B. Kidd, J. Hu, et al., Cell-specific prediction and application of drug-induced gene expression profiles, in: PACIFIC SYMPOSIUM ON BIOCOMPUTING 2018: Proceedings of the Pacific Symposium, World Scientific, 2018, pp. 32–43.

[42] J. B. Kruskal, Three-way arrays: Rank and uniqueness of trilinear decompositions, Lin. Alg. Appl. 18 (2) (1977) 95–138.

[43] J. B. Kruskal, Rank, decomposition, and uniqueness for 3-way and n-way arrays, Multiway Data Analysis, R. Coppi and S. Bolasco, eds., North-Holland, Amsterdam 18 (2) (1989) 7–18.

[44] L. De Lathauwer, Decompositions of a Higher-Order Tensor in Block Terms—Part II: Definitions and Uniqueness, SIAM J. Matrix Anal. Appl. 30 (3) (2008) 1033–1066.

[45] A. Bhaskara, M. Charikar, A. Vijayaraghavan, Uniqueness of tensor decompositions with applications to polynomial identifiability, in: Conference on Learning Theory, PMLR, 2014, pp. 742–778.

[46] J. Håstad, Tensor rank is np-complete, Journal of Algorithms 11 (4) (1990) 644–654.

[47] V. de Silva, L.-H. Lim, Tensor rank and the ill-posedness of the best low-rank approximation problem, SIAM J. Matrix Anal. Appl. 30 (3) (2008) 1084–1127.

[48] M. Sørensen, L. D. Lathauwer, P. Comon, S. Icart, L. Deneire, Canonical polyadic decomposition with a columnwise orthonormal factor matrix, SIAM J. Matrix Anal. Appl. 33 (4) (2012) 1190–1213.

[49] B. Alexandrov, D. F. DeSantis, G. Manzini, E. W. Skau, Nonnegative canonical tensor decomposition with linear constraints: nncandelinc, Numer. Linear Algebra Appl. 29 (6) (2022) e2443.

[50] A. Cichocki, R. Zdunek, S.-i. Amari, Nonnegative matrix and tensor factorization [lecture notes], IEEE signal processing magazine 25 (1) (2007) 142–145. doi:10.1109/MSP.2008.4408452.

[51] L. De Lathauwer, B. De Moor, J. Vandewalle, A multilinear singular value decomposition, SIAM J. Matrix Anal. Appl. 21 (4) (2000) 1253–1278.

[52] R. A. Harshman, M. E. Lundy, PARAFAC: Parallel factor analysis, Computational Statistics and Data Analysis 18 (1) (1994) 39–72. doi:https://doi.org/10.1016/0167-9473(94)90132-5.

[53] R. Bro, PARAFAC. tutorial and applications, Chemometrics and Intelligent Laboratory Systems 38 (2) (1997) 149–171. doi:https://doi.org/10.1016/S0169-7439(97)00032-4.

[54] G. Favier, A. de Almeida, Overview of constrained PARAFAC models, EURASIP Journal on Advances in Signal Processing 2014 (1) (2014) 142. doi:10.1186/1687-6180-2014-142.

[55] R. A. Harshman, M. E. Lundy, Uniqueness proof for a family of models sharing features of Tucker's three-mode factor analysis and parafac/candecomp, Psychometrika 61 (1) (1996) 133–154. doi:10.1007/BF02296963.

[56] R. Bro, Multi-way analysis in the food industry: models, algorithms, and applications, Ph.D. thesis, Universiteit van Amsterdam (1998).

[57] Y. Zniyed, A. L. de Almeida, A stochastic algorithm for the paratuck decomposition, Digital Signal Processing 156 (2025) 104767.

[58] A. L. F. de Almeida, G. Favier, J. P. C. L. da Costa, J. C. M. Mota, Overview of tensor decompositions with applications to communications, in: R. Coelho, V. Nascimento, R. de Queiroz, J. Romano, C. Cavalcante (Eds.), Signals and Images: Advances and Results in Speech, Estimation, Compression, Recognition, Filtering, and Processing, CRC-Press, 2016, pp. 325–356.

[59] A. L. de Almeida, G. Favier, J. C. Mota, Space–time spreading–multiplexing for MIMO wireless communication systems using the PARATUCK-2 tensor model, Signal Processing 89 (11) (2009) 2103–2116.

[60] K. Usevich, Y. Zniyed, M. Ishteva, P. Dreesen, A. L. F. de Almeida, Tensor-based two-layer decoupling of multivariate polynomial maps, in: 2023 31st European Signal Processing Conference (EUSIPCO), 2023, pp. 655–659. doi:10.23919/EUSIPCO58844.2023.10289900.

[61] W. Hackbusch, S. Kühn, A new scheme for the tensor representation, Journal of Fourier analysis and applications 15 (5) (2009) 706–722.

[62] N. B. Erichson, K. Manohar, S. L. Brunton, J. N. Kutz, Randomized cp tensor decomposition, Machine Learning: Science and Technology 1 (2) (2020) 025012.

[63] S. Ahmadi-Asl, S. Abukhovich, M. G. Asante-Mensah, A. Cichocki, A. H. Phan, T. Tanaka, I. Oseledets, Randomized algorithms for computation of tucker decomposition and higher order svd (hosvd), IEEE Access 9 (2021) 28684–28706.

[64] O. Mickelin, S. Karaman, On algorithms for and computing with the tensor ring decomposition, Numerical Linear Algebra with Applications 27 (3) (2020) e2289.

[65] T. G. Kolda, J. Sun, Scalable tensor decompositions for multi-aspect data mining, in: 2008 Eighth IEEE international conference on data mining, IEEE, 2008, pp. 363–372.

[66] S. Smith, N. Ravindran, N. D. Sidiropoulos, G. Karypis, Splatt: Efficient and parallel sparse tensor-matrix multiplication, in: 2015 IEEE International Parallel and Distributed Processing Symposium, IEEE, 2015, pp. 61–70.

[67] U. Kang, E. Papalexakis, A. Harpale, C. Faloutsos, Gigatensor: scaling tensor analysis up by 100 times-algorithms and discoveries, in: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, 2012, pp. 316–324.

[68] K. Abed-Meraim, N. L. Trung, A. Hafiane, et al., A contemporary and comprehensive survey on streaming tensor decomposition, IEEE Transactions on Knowledge and Data Engineering 35 (11) (2022) 10897–10921.

[69] Y. Du, Y. Zheng, K.-c. Lee, S. Zhe, Probabilistic streaming tensor decomposition, in: 2018 IEEE International Conference on Data Mining (ICDM), IEEE, 2018, pp. 99–108.

[70] E. J. Candès, T. Tao, The power of convex relaxation: Near-optimal matrix completion, IEEE transactions on information theory 56 (5) (2010) 2053–2080.

[71] H. Derksen, On the nuclear norm and the singular value decomposition of tensors, Foundations of Computational Mathematics 16 (3) (2016) 779–811.

[72] M. Yuan, C.-H. Zhang, On tensor completion via nuclear norm minimization, Foundations of Computational Mathematics 16 (4) (2016) 1031–1068.

[73] C. J. Hillar, L.-H. Lim, Most tensor problems are np-hard, Journal of the ACM (JACM) 60 (6) (2013) 1–39.

[74] G. A. Shaw, H. K. Burke, Spectral imaging for remote sensing, Lincoln laboratory journal 14 (1) (2003) 3–28.

[75] J. Plaza, E. M. Hendrix, I. García, G. Martín, A. Plaza, On endmember identification in hyperspectral images without

pure pixels: A comparison of algorithms, Journal of Mathematical Imaging and Vision 42 (2012) 163–175.

[76] A. Hyvärinen, E. Oja, Independent component analysis: algorithms and applications, Neural networks 13 (4-5) (2000) 411–430.

[77] Q. Wei, N. Dobigeon, J.-Y. Tourneret, Fast fusion of multi-band images based on solving a Sylvester equation, IEEE Trans. Image Process. 24 (11) (2015) 4109–4121.

[78] M. Simoes, J. M. Bioucas-Dias, L. B. Almeida, J. Chanussot, A convex formulation for hyperspectral image superresolution via subspace-based regularization, IEEE Trans. Geosci. Remote Sens. 53 (6) (2015) 3373–3388.

[79] N. Yokoya, C. Grohnfeldt, J. Chanussot, Hyperspectral and multispectral data fusion: A comparative review of the recent literature, IEEE Trans. Geosci. Remote Sens. 5 (2) (2017) 29–56.

[80] N. Yokoya, T. Yairi, A. Iwasaki, Coupled Nonnegative Matrix Factorization Unmixing for Hyperspectral and Multispectral Data Fusion, IEEE Trans. Geosci. Remote Sens. 50 (2) (2012) 528–537. doi:10.1109/TGRS.2011.2161320.

[81] V. Leplat, N. Gillis, C. Févotte, Multi-resolution beta-divergence nmf for blind spectral unmixing, Signal Processing (2021).

[82] C. I. Kanatsoulis, X. Fu, N. D. Sidiropoulos, W.-K. Ma, Hyperspectral Super-Resolution: A Coupled Tensor Factorization Approach, IEEE Trans. Signal Process. 66 (24) (2018) 6503–6517.

[83] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, J. Chanussot, Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches, IEEE journal of selected topics in applied earth observations and remote sensing 5 (2) (2012) 354–379.

[84] M. Ding, X. Fu, T.-Z. Huang, J. Wang, X.-L. Zhao, Hyperspectral super-resolution via interpretable block-term tensor modeling, arXiv e-prints (2020) arXiv:2006.10248arXiv:2006.10248.

[85] Q. Wei, J. M. Bioucas-Dias, N. Dobigeon, J.-Y. Tourneret, Multiband image fusion based on spectral unmixing, IEEE Trans. Geosci. Remote Sens. 54 (12) (2016) 7236–7249.

[86] Q. Li, W.-K. Ma, Q. Wu, Hyperspectral super-resolution: Exact recovery in polynomial time, in: 2018 IEEE SSP, IEEE, 2018, pp. 378–382.

[87] C. I. Kanatsoulis, X. Fu, N. D. Sidiropoulos, W.-K. Ma, Hyperspectral Super-Resolution: Combining Low Rank Tensor and Matrix Structure, in: 2018 IEEE ICIP, 2018, pp. 3318–3322.

[88] C. Prévost, K. Usevich, P. Comon, D. Brie, Hyperspectral Super-Resolution with Coupled Tucker Approximation: Identifiability and SVD-based algorithms, IEEE Trans. Signal Process. 68 (2020) 931–946.

[89] C. Prévost, R. A. Borsoi, K. Usevich, D. Brie, J. M. Bermudez, C. Richard, Hyperspectral super-resolution accounting for spectral variability: Coupled tensor ll1-based recovery and blind unmixing of the unknown super-resolution image, SIAM Journal on Imaging Sciences 15 (1) (2022) 110–138. doi:10.1137/21M1409354.

[90] Y. Qian, F. Xiong, S. Zeng, J. Zhou, Y. Tang, Matrix-vector nonnegative tensor factorization for blind unmixing of hyperspectral imagery, IEEE Trans. Geosci. Remote Sens. 55 (3) (2016) 1776–1792.

[91] G. Zhang, X. Fu, K. Huang, J. Wang, Hyperspectral super-resolution: A coupled nonnegative block-term tensor decomposition approach, in: 2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), IEEE, 2019, pp. 470–474.

[92] H. Guo, W. Bao, K. Qu, X. Ma, M. Cao, Multispectral and hyperspectral image fusion based on regularized coupled non-negative block-term tensor decomposition, Remote Sensing 14 (21) (2022) 5306.

[93] H. Liu, W. Jiang, Y. Zha, Z. Wei, Coupled tensor block term decomposition with superpixel-based graph laplacian regularization for hyperspectral super-resolution, Remote Sensing 14 (18) (2022) 4520.

[94] A. A. Rontogiannis, E. Kofidis, P. V. Giampouras, Online rank-revealing block-term tensor decomposition, Signal Processing (2023) 109126.

[95] P. V. Giampouras, A. A. Rontogiannis, E. Kofidis, Block-term tensor decomposition model selection and computation:

The bayesian way, IEEE Transactions on Signal Processing 70 (2022) 1704–1717.

[96] V. Zarzoso, Parameter estimation in block term decomposition for noninvasive atrial fibrillation analysis, in: 2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), IEEE, 2017, pp. 1–5.

[97] C. Févotte, N. Dobigeon, Nonlinear hyperspectral unmixing with robust nonnegative matrix factorization, IEEE Transactions on Image Processing 24 (12) (2015) 4810–4819.

[98] V. Leplat, N. Gillis, A. M. Ang, Blind audio source separation with minimum-volume beta-divergence nmf, IEEE Transactions on Signal Processing 68 (2020) 3400–3410.

[99] S. Mahalingam, P. Srinivas, P. K. Devi, D. Sita, S. K. Das, T. S. Leela, V. R. Venkataraman, Reflectance based vicarious calibration of hysis sensors and spectral stability study over pseudo-invariant sites, in: 2019 IEEE Recent Advances in Geoscience and Remote Sensing: Technologies, Standards and Applications (TENGARSS), IEEE, 2019, pp. 132–136.

[100] H. Chen, F. Ahmad, S. Vorobyov, F. Porikli, Tensor decompositions in wireless communications and mimo radar, IEEE Journal of Selected Topics in Signal Processing 15 (3) (2021) 438–453. doi:10.1109/JSTSP.2021.3061937.

[101] D. C. Araújo, A. L. F. de Almeida, J. P. C. L. Da Costa, R. T. de Sousa, Tensor-based channel estimation for massive mimo-ofdm systems, IEEE Access 7 (2019) 42133–42147. doi:10.1109/ACCESS.2019.2908207.

[102] Y. Polyanskiy, A perspective on massive random-access, in: 2017 IEEE International Symposium on Information Theory (ISIT), 2017, pp. 2523–2527. doi:10.1109/ISIT.2017.8006984.

[103] A. Decurninge, I. Land, M. Guillaud, Tensor-based modulation for unsourced massive random access, IEEE Wireless Communications Letters 10 (3) (2021) 552–556. doi:10.1109/LWC.2020.3037523.

[104] C. J. Hillar, L.-H. Lim, Most tensor problems are np-hard, Journal of the ACM (JACM) 60 (6) (2013) 1–39.

[105] M. Gkagkos, K. R. Narayanan, J.-F. Chamberland, C. N. Georghiades, Fasura: A scheme for quasi-static fading unsourced random access channels, IEEE Transactions on Communications 71 (11) (2023) 6391–6401. doi:10.1109/TCOMM.2023.3296593.

[106] Study on channel model for frequencies from 0.5 to 100 ghz, Tech. rep., 3GPP, 38.901 version 16.1.0 Release 16 (Jan. 2020).

[107] Y. Zniyed, R. Boyer, A. L. De Almeida, G. Favier, Tensor train representation of mimo channels using the jirafe method, Signal Processing 171 (2020) 107479.

[108] M. F. Couras, P. H. de Pinho, G. Favier, V. Zarzoso, A. L. de Almeida, J. P. da Costa, Semi-blind receivers based on a coupled nested tucker-parafac model for dual-polarized mimo systems using combined tst and msmkron codings, Digital Signal Processing 137 (2023) 104043.

[109] A. L. F. de Almeida, G. Favier, L. R. Ximenes, Space-time-frequency (stf) mimo communication systems with blind receiver based on a generalized paratuck2 model, IEEE Transactions on Signal Processing 61 (8) (2013) 1895–1909.

[110] X. Zhang, J. Zou, K. He, J. Sun, Accelerating very deep convolutional networks for classification and detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (10) (2016) 1943–1955.

[111] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 1–11.

[112] Z. Cheng, B. Li, Y. Fan, Y. Bao, A novel rank selection scheme in tensor ring decomposition based on reinforcement learning for deep neural networks, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 3292–3296.

[113] V. T. Pham, Y. Zniyed, T. P. Nguyen, Efficient tensor decomposition-based filter pruning, Neural Networks (2024) 106393.

[114] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, K. Keutzer, A survey of quantization methods for efficient neural network inference, in: Low-Power Computer Vision, Chapman and Hall/CRC, 2022, pp. 291–326.

[115] H. Cheng, M. Zhang, J. Q. Shi, A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations, IEEE Transactions on Pattern Analysis and Machine Intelligence (2024).

[116] J. Gou, B. Yu, S. J. Maybank, D. Tao, Knowledge distillation: A survey, International Journal of Computer Vision 129 (6) (2021) 1789–1819.

[117] T. Elsken, J. H. Metzen, F. Hutter, Neural architecture search: A survey, Journal of Machine Learning Research 20 (55) (2019) 1–21.

[118] W. Dai, J. Fan, Y. Miao, K. Hwang, Deep learning model compression with rank reduction in tensor decomposition, IEEE Transactions on Neural Networks and Learning Systems (2023) 1–14.

[119] V. T. Pham, Y. Zniyed, T. P. Nguyen, Enhanced network compression through tensor decompositions and pruning, IEEE Transactions on Neural Networks and Learning Systems (2024) 1–13.

[120] V. Lebedev, Y. Ganin, M. Rakhuba, I. V. Oseledets, V. S. Lempitsky, Speeding-up convolutional neural networks using fine-tuned cp-decomposition, in: International Conference on Learning Representations, 2015, pp. 1–11.

[121] Y.-D. Kim, E. Park, S. Yoo, T. Choi, L. Yang, D. Shin, Compression of deep convolutional neural networks for fast and low power mobile applications, in: International Conference on Learning Representations, 2016, pp. 1–11.

[122] T. Garipov, D. Podoprikhin, A. Novikov, D. P. Vetrov, Ultimate tensorization: compressing convolutional and fc layers alike, NIPS (2016).

[123] W. Wang, Y. Sun, B. Eriksson, W. Wang, V. Aggarwal, Wide compression: Tensor ring nets, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9329–9338.

[124] K. Guo, Z. Lin, C. Chen, X. Xing, F. Liu, X. Xu, Compact model training by low-rank projection with energy transfer, IEEE Transactions on Neural Networks and Learning Systems (2024).

[125] V. T. Pham, Y. Zniyed, T. P. Nguyen, Hybrid network compression through tensor decompositions and pruning, in: 32nd European Signal Processing Conference, EUSIPCO 2024, 2024, pp. 1–6.

[126] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, R. Fergus, Exploiting linear structure within convolutional networks for efficient evaluation, Advances in neural information processing systems 27 (2014).

[127] S. Lin, R. Ji, C. Chen, D. Tao, J. Luo, Holistic cnn compression via low-rank decomposition with knowledge transfer, IEEE Transactions on Pattern Analysis and Machine Intelligence 41 (12) (2019) 2889–2905.

[128] Y. Idelbayev, M. A. Carreira-Perpinán, Low-rank compression of neural nets: Learning the rank of each layer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8049–8059.

[129] Y. Xu, Y. Li, S. Zhang, W. Wen, B. Wang, Y. Qi, Y. Chen, W. Lin, H. Xiong, Trp: Trained rank pruning for efficient deep neural networks, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, 2020, pp. 977–983.

[130] S. Li, E. Hanson, H. Li, Y. Chen, Penni: Pruned kernel sharing for efficient cnn inference, in: International Conference on Machine Learning, PMLR, 2020, pp. 5863–5873.

[131] Y. Li, S. Lin, J. Liu, Q. Ye, M. Wang, F. Chao, F. Yang, J. Ma, Q. Tian, R. Ji, Towards compact cnns via collaborative compression, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 6434–6443.

[132] L. Liebenwein, A. Maalouf, D. Feldman, D. Rus, Compressing neural networks: Towards determining the optimal layer-wise decomposition, Advances in Neural Information Processing Systems 34 (2021) 5328–5344.

[133] M. Eo, S. Kang, W. Rhee, An effective low-rank compression with a joint rank selection followed by a compression-friendly training, Neural Networks 161 (2023) 165–177.

[134] S. Chen, J. Zhou, W. Sun, L. Huang, Joint matrix decomposition for deep convolutional neural networks compression, Neurocomputing 516 (2023) 11–26.

[135] V. T. Pham, Y. Zniyed, T. P. Nguyen, Élagage efficace des filtres basé sur les décompositions tensorielles, in: XXIXème

Colloque Francophone de Traitement du Signal et des Images, GRETSI - Groupe de Recherche en Traitement du Signal et des Images, 2023, pp. 937–940.

[136] M. Astrid, S.-I. Lee, Cp-decomposition with tensor power method for convolutional neural networks compression, in: 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), IEEE, 2017, pp. 115–118.

[137] A.-H. Phan, K. Sobolev, K. Sozykin, D. Ermilov, J. Gusak, P. Tichavskỳ, V. Glukhov, I. Oseledets, A. Cichocki, Stable low-rank tensor decomposition for compression of convolutional neural network, in: European Conference on Computer Vision, 2020, pp. 522–539.

[138] Z. Zhong, F. Wei, Z. Lin, C. Zhang, Ada-tucker: Compressing deep neural networks via adaptive dimension adjustment tucker decomposition, Neural Networks 110 (2019) 104–115.

[139] C. Dai, X. Liu, H. Cheng, L. T. Yang, M. J. Deen, Compressing deep model with pruning and tucker decomposition for smart embedded systems, IEEE Internet of Things Journal 9 (16) (2021) 14490–14500.

[140] M. Yin, H. Phan, X. Zang, S. Liao, B. Yuan, Batude: Budget-aware neural network compression based on tucker decomposition, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 8874–8882.

[141] J. Xiao, C. Zhang, Y. Gong, M. Yin, Y. Sui, L. Xiang, D. Tao, B. Yuan, Haloc: hardware-aware automatic low-rank compression for compact neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 10464–10472.

[142] M. Gabor, R. Zdunek, Compressing convolutional neural networks with hierarchical tucker-2 decomposition, Applied Soft Computing 132 (2023) 109856.

[143] P. Wang, J. Cheng, Accelerating convolutional neural networks for mobile applications, in: Proceedings of the 24th ACM international conference on Multimedia, 2016, pp. 541–545.

[144] A.-H. Phan, D. Ermilov, N. Kozyrskiy, I. Vorona, K. Sobolev, A. Cichocki, How to train your unstable looped tensor network, IEEE Journal of Selected Topics in Signal Processing (2024) 1–10.

[145] A. Novikov, D. Podoprikhin, A. Osokin, D. P. Vetrov, Tensorizing neural networks, Advances in neural information processing systems 28 (2015).

[146] C. Deng, F. Sun, X. Qian, J. Lin, Z. Wang, B. Yuan, Tie: Energy-efficient tensor train-based inference engine for deep neural network, in: Proceedings of the 46th International Symposium on Computer Architecture, 2019, pp. 264–278.

[147] M. Yin, Y. Sui, S. Liao, B. Yuan, Towards efficient tensor decomposition-based dnn model compression with optimization framework, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 10674–10683.

[148] M. Yin, Y. Sui, W. Yang, X. Zang, Y. Gong, B. Yuan, Hodec: Towards efficient high-order decomposed convolutional neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 12299–12308.

[149] Y. Sui, M. Yin, Y. Gong, B. Yuan, Co-exploring structured sparsification and low-rank tensor decomposition for compact dnns, IEEE Transactions on Neural Networks and Learning Systems (2024).

[150] N. Li, Y. Pan, Y. Chen, Z. Ding, D. Zhao, Z. Xu, Heuristic rank selection with progressively searching tensor ring network, Complex & Intelligent Systems (2021) 1–15.

[151] A. Obukhov, M. Rakhuba, S. Georgoulis, M. Kanakis, D. Dai, L. Van Gool, T-basis: a compact representation for neural networks, in: International Conference on Machine Learning, PMLR, 2020, pp. 7392–7404.

[152] K. Xie, C. Liu, X. Wang, X. Li, G. Xie, J. Wen, K. Li, Neural network compression based on tensor ring decomposition, IEEE Transactions on Neural Networks and Learning Systems (2024).

[153] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[154] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078 (2014).

[155] Y. Yang, D. Krompass, V. Tresp, Tensor-train recurrent neural networks for video classification, in: International Conference on Machine Learning, PMLR, 2017, pp. 3891–3900.

[156] J. Ye, L. Wang, G. Li, D. Chen, S. Zhe, X. Chu, Z. Xu, Learning compact recurrent neural networks with block-term tensor decomposition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 9378–9387.

[157] Y. Pan, J. Xu, M. Wang, J. Ye, F. Wang, K. Bai, Z. Xu, Compressing recurrent neural networks with tensor ring for action recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 4683–4690.

[158] M. Yin, S. Liao, X.-Y. Liu, X. Wang, B. Yuan, Towards extremely compact rnns for video recognition with fully decomposed hierarchical tucker structure, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 12085–12094.

[159] Y. Gong, M. Yin, L. Huang, C. Deng, B. Yuan, Algorithm and hardware co-design of energy-efficient lstm networks for video recognition with hierarchical tucker tensor decomposition, IEEE Transactions on Computers 71 (12) (2022) 3101–3114.

[160] A. Tjandra, S. Sakti, S. Nakamura, Compressing recurrent neural network with tensor train, in: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 4451–4458.

[161] A. M. Grachev, D. I. Ignatov, A. V. Savchenko, Compression of recurrent neural networks for efficient language modeling, Applied Soft Computing 79 (2019) 354–362.

[162] D. Liu, X. Bai, R. Zhao, X. Deng, L. T. Yang, Dual-grained lightweight strategy, IEEE Transactions on Pattern Analysis and Machine Intelligence (2024).

[163] J. Ye, G. Li, D. Chen, H. Yang, S. Zhe, Z. Xu, Block-term tensor neural networks, Neural Networks 130 (2020) 11–21.

[164] M. B. Noach, Y. Goldberg, Compressing pre-trained language models by matrix decomposition, in: Proceedings of the 10th International Joint Conference on Natural Language Processing, 2020, pp. 884–889.

[165] Y.-C. Hsu, T. Hua, S. Chang, Q. Lou, Y. Shen, H. Jin, Language model compression with weighted low-rank factorization, in: International Conference on Learning Representations, 2022, pp. 1–11.

[166] H. Hajimolahoseini, W. Ahmed, M. Rezagholizadeh, V. Partovinia, Y. Liu, Strategies for applying low rank decomposition to transformer-based models, in: 36th Conference on Neural Information Processing Systems (NeurIPS2022), 2022, pp. 1–6.

[167] Y. Guo, G. Wang, M. Kankanhalli, Pela: Learning parameter-efficient models with low-rank approximation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 15699–15709.

[168] S. Gao, T. Hua, Y.-C. Hsu, Y. Shen, H. Jin, Adaptive rank selections for low-rank approximation of language models, in: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 2024, pp. 227–241.

[169] P. Sharma, J. T. Ash, D. Misra, The truth is in there: Improving reasoning in language models with layer-selective rank reduction, in: The Twelfth International Conference on Learning Representations, 2024, pp. 1–12.

[170] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations, 2021, pp. 1–21.

[171] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI blog 1 (8) (2019) 9.

[172] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186.

[173] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971 (2023).

[174] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S. Bowman, GLUE: A multi-task benchmark and analysis platform for natural language understanding, in: T. Linzen, G. Chrupała, A. Alishahi (Eds.), Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 353–355.

[175] X. Ma, P. Zhang, S. Zhang, N. Duan, Y. Hou, M. Zhou, D. Song, A tensorized transformer for language modeling, Advances in neural information processing systems 32 (2019).

[176] O. Hrinchuk, V. Khrulkov, L. Mirvakhabova, E. Orlova, I. Oseledets, Tensorized embedding layers, in: Findings of the Association for Computational Linguistics: EMNLP 2020, 2019, pp. 4847–4860.

[177] P. Zhen, Z. Gao, T. Hou, Y. Cheng, H.-B. Chen, Deeply tensor compressed transformers for end-to-end object detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 4716–4724.

[178] S. Jie, Z.-H. Deng, Fact: Factor-tuning for lightweight adaptation on vision transformer, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 37, 2023, pp. 1060–1068.

[179] H. Li, J. Zhao, H. Huo, S. Fang, J. Chen, L. Yao, Y. Hua, T3srs: Tensor train transformer for compressing sequential recommender systems, Expert Systems with Applications 238 (2024) 122260.

[180] Y. Yang, J. Zhou, N. Wong, Z. Zhang, LoRETTA: Low-rank economic tensor-train adaptation for ultra-low-parameter fine-tuning of large language models, in: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 2024, pp. 3161–3176.

[181] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: European conference on computer vision, Springer, 2020, pp. 213–229.

[182] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.

[183] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, 2014, pp. 740–755.

[184] Åke Björck, G. H. Golub, Numerical methods for computing angles between linear subspaces, Mathematics of Computation 27 (123) (1973) 579–594.

[185] A. Alyass, M. Turcotte, D. Meyre, From big data analysis to personalized medicine for all: challenges and opportunities, BMC Medical Genom. 8 (2015) 1–12.

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

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

[188] E. J. Candès, B. Recht, Exact matrix completion via convex optimization, Found. Comput. Math. 9 (6) (2009) 717–772.

[189] B. Recht, A simpler approach to matrix completion, J. Mach. Learn. Res. 12 (104) (2011) 3413–3430.

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

[191] F. Nie, H. Huang, C. Ding, Low-rank matrix recovery via efficient schatten p-norm minimization, in: Proceedings of the 26th AAAI Conference on AI, Vol. 26, 2012, pp. 655–661.

[192] M.-J. Lai, Y. Xu, W. Yin, Improved iteratively reweighted least squares for unconstrained smoothed $l_q$ minimization, SIAM J. Numer. Anal. 51 (2) (2013) 927–957.

[193] J. E. Dennis, Jr, J. J. Moré, Quasi-newton methods, motivation and theory, SIAM Rev. 19 (1) (1977) 46–89.

[194] J. Nocedal, S. J. Wright, Numerical optimization, Springer, 1999.

[195] R. H. Keshavan, A. Montanari, S. Oh, Matrix completion from a few entries, IEEE Trans. Inf. Theory 56 (6) (2010) 2980–2998.

[196] R. Sun, Z.-Q. Luo, Guaranteed matrix completion via non-convex factorization, IEEE Trans. Inf. Theory 62 (11) (2016) 6535–6579.

[197] H.-F. Yu, N. Rao, I. S. Dhillon, Temporal regularized matrix factorization for high-dimensional time series prediction, Adv. Neural Inf. Process. Syst. 29 (2016).

[198] P. Jain, P. Netrapalli, S. Sanghavi, Low-rank matrix completion using alternating minimization, in: Proceedings of the 45th STOC, ACM, 2013, pp. 665–674.

[199] J. Tanner, K. Wei, Low rank matrix completion by alternating steepest descent methods, Appl. Comput. Harmon. Anal. 40 (2) (2016) 417–429.

[200] B. Vandereycken, Low-rank matrix completion by riemannian optimization, SIAM J. Optim. 23 (2) (2013) 1214–1236.

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

[202] V. Larsson, C. Olsson, E. Bylow, F. Kahl, Rank minimization with structured data patterns, in: Proceedings of the 13th ECCV, Springer, 2014, pp. 250–265.

[203] M. Sørensen, S. Hendrikx, L. De Lathauwer, Multilinear singular value decomposition based completion with fibers observed in a single mode, SIAM J. Matrix Anal. Appl.[Accepted for publication] (2025).

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

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

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

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

[208] T. G. Kolda, B. W. Bader, Tensor decompositions and applications, SIAM Rev. 51 (3) (2009) 455–500.

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

[210] S. Hu, Relations of the nuclear norm of a tensor and its matrix flattenings, Linear Algebra Appl. 478 (2015) 188–199.

[211] H. Derksen, Matrix completion and tensor rank, Linear Multilinear Algebra 64 (4) (2016) 680–685.

[212] S. Friedland, L.-H. Lim, Nuclear norm of higher-order tensors, Math. Comput. 87 (311) (2018) 1255–1281.

[213] M. Signoretto, L. De Lathauwer, J. A. Suykens, Nuclear norms for tensors and their use for convex multilinear estimation, Submitted to Linear Algebra Appl. 43 (2010).

[214] J. A. Bengua, H. N. Phien, H. D. Tuan, M. N. Do, Efficient tensor completion for color image and video recovery: Low-rank tensor train, IEEE Trans. Image Process. 26 (5) (2017) 2466–2479.

[215] M. Signoretto, R. Van de Plas, B. De Moor, J. A. Suykens, Tensor versus matrix completion: A comparison with application to spectral data, IEEE Signal Process. Lett. 18 (7) (2011) 403–406.

[216] S. Gandy, B. Recht, I. Yamada, Tensor completion and low-n-rank tensor recovery via convex optimization, Inverse Probl. 27 (2) (2011) 025010.

[217] E. Acar, D. M. Dunlavy, T. G. Kolda, M. Mørup, Scalable tensor factorizations for incomplete data, Chemometr. Intell. Lab Syst. 106 (1) (2011) 41–56.

[218] M. Filipović, A. Jukić, Tucker factorization with missing data with application to low-n-rank tensor completion, Multidimens. Syst. Signal Process. 26 (3) (2015) 677–692.

[219] L. Grasedyck, M. Kluge, S. Krämer, Variants of alternating least squares tensor completion in the tensor train format, SIAM J. Sci. Comput. 37 (5) (2015) A2424–A2450.

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

[221] A.-H. Phan, A. Cichocki, A. Uschmajew, P. Tichavský, G. Luta, D. P. Mandic, Tensor networks for latent variable analysis: Novel algorithms for tensor train approximation, IEEE Trans. Neural Netw. 31 (11) (2020) 4622–4636.

[222] N. Vervliet, L. De Lathauwer, Numerical optimization-based algorithms for data fusion, in: Data Handl. Sci. Technol., Vol. 31, Elsevier, 2019, pp. 81–128.

[223] M. Sørensen, L. De Lathauwer, Fiber sampling approach to canonical polyadic decomposition and application to tensor completion, SIAM J. Matrix Anal. Appl. 40 (3) (2019) 888–917.

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

[225] S. S. Sofi, I. Oseledets, A case study of spatiotemporal forecasting techniques for weather forecasting, GeoInformatica (2024).

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

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

[228] N. Vervliet, O. Debals, L. De Lathauwer, Exploiting efficient representations in large-scale tensor decompositions, SIAM J. Sci. Comput. 41 (2019) A789–A815.

[229] M. Vandecappelle, L. De Lathauwer, Updating the multilinear utv decomposition, IEEE Trans. Signal Process. 70 (2022) 3551–3565.

[230] H. J. Woerdeman, Toeplitz minimal rank completions, Linear Algebra Appl. 202 (1994) 267–278.

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

[232] A. Özdemir, M. A. Iwen, S. Aviyente, A multiscale approach for tensor denoising, in: 2016 IEEE Statistical Signal Processing Workshop (SSP), IEEE, 2016, pp. 1–5.

[233] Y. Hasin, M. Seldin, A. Lusis, Multi-omics approaches to disease, Genome biology 18 (1) (2017) 1–15. doi:10.1186/s13059-017-1215-1.

[234] E. Frolov, I. Oseledets, Tensor methods and recommender systems, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 7 (3) (2017) e1201.

[235] A. Termanova, A. Melnikov, E. Mamenchikov, N. Belokonev, S. Dolgov, A. Berezutskii, R. Ellerbrock, C. Mansell, M. Perelshtein, Tensor quantum programming, New Journal of Physics 26 (12) (2024) 123019.

[236] D. Chafamo, V. Shanmugam, N. Tokcan, C-ziptf: stable tensor factorization for zero-inflated multi-dimensional genomics data, BMC bioinformatics 25 (1) (2024) 323.

[237] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, et al., Variational quantum algorithms, Nature Reviews Physics 3 (9) (2021) 625–644.

[238] H.-M. Rieser, F. Köster, A. P. Raulf, Tensor networks for quantum machine learning, Proceedings of the Royal Society A 479 (2275) (2023) 20230218.

[239] J. A. Reuter, D. V. Spacek, M. P. Snyder, High-throughput sequencing technologies, Molecular cell 58 (4) (2015) 586–597.

[240] D. Lähnemann, J. Köster, E. Szczurek, D. J. McCarthy, S. C. Hicks, M. D. Robinson, C. A. Vallejos, K. R. Campbell, N. Beerenwinkel, A. Mahfouz, et al., Eleven grand challenges in single-cell data science, Genome biology 21 (2020) 1–35.

[241] D. Lähnemann, J. Köster, E. Szczurek, D. J. McCarthy, S. C. Hicks, M. D. Robinson, C. A. Vallejos, K. R. Campbell,

N. Beerenwinkel, A. Mahfouz, et al., Eleven grand challenges in single-cell data science, Genome biology 21 (1) (2020) 1–35. doi:10.1186/s13059-020-1926-6.

[242] T. Downing, N. Angelopoulos, A primer on correlation-based dimension reduction methods for multi-omics analysis, Journal of the Royal Society Interface 20 (207) (2023) 20230344.

[243] I. Jung, M. Kim, S. Rhee, S. Lim, S. Kim, Monti: a multi-omics non-negative tensor decomposition framework for gene-level integrative analysis, Frontiers in genetics 12 (2021) 682841.

[244] E. Armingol, H. M. Baghdassarian, C. Martino, A. Perez-Lopez, C. Aamodt, R. Knight, N. E. Lewis, Context-aware deconvolution of cell–cell communication with tensor-cell2cell, Nature communications 13 (1) (2022) 3665.