Deep-Learning-Aided Alternating Least Squares for Tensor CP Decomposition and Its Application to Massive MIMO Channel Estimation

Gong, Xiao; Chen, Wei; Ai, Bo; Leus, Geert

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Deep-Learning-Aided Alternating Least Squares for Tensor CP Decomposition and Its Application to Massive MIMO Channel Estimation

Xiao Gong, Wei Chen, *Senior Member, IEEE*, Bo Ai, *Fellow, IEEE*, and Geert Leus, *Fellow, IEEE*

Abstract— **CANDECOMP/PARAFAC (CP) decomposition is the mostly used model to formulate the received tensor signal in a massive MIMO system, as the receiver generally sums the components from different paths or users. To achieve accurate and low-latency channel estimation, good and fast CP decomposition (CPD) algorithms are desired. The CP alternating least squares (CPALS) is the workhorse algorithm for calculating the CPD. However, its performance depends on the initializations, and good starting values can lead to more efficient solutions. Existing initialization strategies are decoupled from the CPALS and are not necessarily favorable for solving the CPD. This paper proposes a deep-learning-aided CPALS (DL-CPALS) method that uses a deep neural network (DNN) to generate favorable initializations. The proposed DL-CPALS integrates the DNN and CPALS to a model-based deep learning paradigm, where it trains the DNN to generate an initialization that facilitates fast and accurate CPD. Moreover, benefiting from the CP low-rankness, the proposed method is trained using noisy data and does not require paired clean data. The proposed DL-CPALS is applied to millimeter wave MIMO-OFDM channel estimation. Experimental results demonstrate the significant improvements of the proposed method in terms of both speed and accuracy for CPD and channel estimation.**

Index Terms— **Massive MIMO, deep learning, channel estimation, tensor decomposition.**

I. INTRODUCTION

TO REDUCE the degrading effects in harmful propagation environments, modern wireless communication systems tend to add more degrees of freedom by transmitting signals covering multiple domains, e.g., space, frequency, polarization and/or code, which extends the traditional vector or matrix signals to multidimensional arrays, i.e., tensors. The multi-dimensional signal structures can be naturally characterized using tensor decomposition models. In the past decades, tensor models have found a wide range of applications

for wireless communication systems [1]. For example, the CANDECOMP/PARAFAC (CP) decomposition, also called canonical polyadic decomposition [2], [3], which factorizes a tensor into a sum of rank-one tensors, is widely applied to the received signal model [4], as the receiver generally sums the signal components of different paths or users.

As a remarkable 5G technology, massive multiple-input multiple-output (MIMO) antenna systems have been applied in practice for increasing the data rate and reliability [5]. The large-scale antenna arrays lead to high-dimensional channels, especially when combined with other transmitting domains, such as the frequency domain in orthogonal frequency division multiplexing (OFDM) systems. Channel estimation is crucial to embrace the potential gains of massive MIMO systems [6]. However, the acquisition of channel state information (CSI) becomes challenging and expensive in terms of accuracy, training overhead and computational complexity. Fortunately, when adopting the millimeter-wave (mmWave) band or antennas at a high altitude, the sparse scattering property enables an approximation of the channel using a low-rank model [7]. By exploiting this low-rankness of the channel tensor, one can decompose the high-dimensional channels into low-dimensional factors to reduce the storage and enhance the estimation of the channel or its latent parameters, e.g., angles of arrival and departure (AoAs/AoDs), delays, Doppler shifts and path gains [8], [9].

In addition to the pursuit of accurate channel estimation, computational efficient channel estimation methods for massive MIMO systems are also desired to reduce the latency and cope with potential channel variations. Some representative model-driven CP-based estimation methods for a MIMO channel or its latent parameters are listed in Table I. Following array signal processing, the spatial MIMO channel can be constructed by collecting the steering vectors of the different paths/users, thereby leading to a Vandermonde matrix [10]. Hence, for tensor MIMO channels, the CP factors could also be restricted by such a Vandermonde structure, which results in Vandermonde-constrained CP (VCP) decomposition [11]. Furthermore, by exploiting the Vandermonde structure of the factors, subspace-type algorithms based on the singular value decomposition (SVD) and estimation of signal parameters via rotational invariance techniques (ESPRIT) are developed in [12]. Various subspace-type estimation methods to calculate Vandermonde-constrained CP decomposition in different MIMO systems are proposed in [13], [14], [15], [16], and [17].

TABLE I
A LIST OF MODEL-DRIVEN CP-BASED ESTIMATION METHODS FOR A
MIMO CHANNEL OR ITS LATENT PARAMETERS ‡

MIMO Systems	Algorithms
MIMO radar [20]	ALS
MU MIMO [22]	ALS
DP MIMO [21]	ALS
CF MIMO [13]	subspace-type
TV MIMO [23], [24]	ALS
TV MU MIMO [14]	subspace-type
MIMO-OFDM [8]	ALS
DWB MIMO-OFDM [15]	subspace-type
TV MIMO-OFDM [9], [26]	ALS
IRS MIMO [16], [25], [31]	subspace-type/ALS
ISAC MIMO [17], [27], [32]	subspace-type/ALS
MA-enabled MIMO [33]	ALS

‡ MU: multi-user. DP: dual-polarized. CF: cell-free. TV: time-varying. DWB: dual-wide-band. IRS: intelligent-reflecting-surface-assisted. ISAC: integrated sensing and communication.

There are different methods for calculating the CP decomposition, such as gradient descent, quasi-Newton and nonlinear least squares [18], [19]. Nevertheless, the alternating least squares (ALS) is described as the "workhorse" for calculating the CP decomposition [3], and is widely used in massive MIMO channel estimation [8], [9], [20], [21], [22], [23], [24], [25], [26], [27]. The ALS is conceptually very simple, and only updates one factor at the time using least squares minimization. Thus, it can always reduce the objective function value monotonically. Many early works point out that the convergence speed and converged stationary points of the ALS depend on the initialization of the algorithm, and good starting points can help to speed up the ALS and find the minima [28], [29], [30]. While some competing methods may produce superior performance, the ALS can chase them once given better initializations [19]. However, the existing CPALS methods for channel estimation mostly use random initialization and SVD based initialization [3], resulting in thousands of iterations to achieve an expected estimation accuracy, which reduces their practicality. The reason for this problem is that existing initialization strategies are decoupled from the ALS iterations, and are not designed for converging quickly and well, which inspires us to formulate the initialization and iteration into an end-to-end framework.

Recently, deep neural networks (DNNs) show a powerful capability to capture the wireless channel characteristics from tons of data, and they have been widely applied for MIMO channel estimation [34], [35], [36]. By treating the channel estimation as a denoising task, the spatial-frequency convolutional neural network (SFCNN) has been proposed for multidimensional channel estimation, which uses a convolutional neural network (CNN) to exploit the multi-dimensional spatial and frequency structure of a MIMO-OFDM channel [37]. Unfortunately, DNNs are commonly utilized as a black box and data-driven deep learning does not yet offer the

interpretability and reliability of model-based methods [38]. As an alternate that benefits from the advantages of both model-driven and data-driven paradigms, model-based deep learning methods [39] have attracted the attention of the MIMO communication research, which generally incorporate an internal or external DNN into an iterative algorithm. Compared with traditional algorithms implemented for an individual sample, they benefit from learning the domain knowledge and have shown a performance improvement while keeping relative interpretability [40], [41], [42]. However, these methods are designed for matrix signals and are barely applied to exploit the multidimensional tensor signal structure. Focusing on the tensor network decomposition, a core tensor network (CTN) is proposed in [43] that integrates the gradient-descent-based tensor decomposition and transfer learning to learn a mapping of correlated tensors, which improves the initial condition for gradient-descent.

In this paper, we propose a deep-learning-aided CPALS (DL-CPALS) for CP decomposition and massive MIMO tensor channel estimation, which integrates the unrolled model-driven low-rank approximation algorithm with the deep-learning process to enhance the speed and accuracy. The contributions and advantages of this work are summarised as follows:

- A novel DL-aided tensor low-rank decomposition framework is introduced, which could be easily applied to other tensor decomposition models. As long as the algorithm is differentiable to ensure a gradient-based back-propagation, the CPALS of the framework can be replaced by alternative tensor decomposition algorithms.
- An easy-to-implement DNN is developed for the channel estimation application. Specifically, the DL-CPALS augments the workhorse algorithm CPALS by using a simple fully connected neural network to encode an input tensor into the initializations for the unrolled ALS using a small number of iterations. By integrating the DNN-based initialization and the CPALS algorithm into an end-to-end formulation, the DNN is trained to learn good initializations that are favorable for achieving a small low-rank approximation loss efficiently with CPALS.
- Different from existing DNNs for channel denoising, as a DL-aided model-driven method, the performance of the proposed method is guaranteed as it represents iterations of the CPALS. Moreover, benefiting from the denoising effect of a low-rank approximation of the CPALS itself, the proposed method is trained by noisy data and does not require the corresponding real clean data.

Compared with existing initialization strategies, experimental results demonstrate that the proposed method leads to a more accurate CP decomposition and channel estimate by using fewer iterative steps.

II. NOTATIONS

Tensors, matrices, vectors and scalars are represented by boldface calligraphic uppercase letters $\boldsymbol{\mathcal{X}}$, boldface capital letters $\mathbf{X}$, boldface lower case letters $\mathbf{x}$ and lowercase letters x, respectively. The symbol j is used to represent $\sqrt{-1}$. Superscripts $(\cdot)^T, (\cdot)^*, (\cdot)^H$ and $(\cdot)^\dagger$ denote the transpose,

complex conjugate, Hermitian transpose and pseudoinverse, respectively. Symbols $\circ$, $\odot$, $\diamond$ and $\otimes$ denote the outer product, Hadamard product, Khatri-Rao product and Kronecker product, respectively. The operators $\mathrm{rank}(\cdot)$ and $\mathrm{krank}(\cdot)$ denote the matrix rank and Kruskal rank, respectively. The operator $\mathrm{diag}(\cdot)$ converts a vector into a diagonal matrix. $\mathcal{U}(a,b)$ denotes the uniform distribution in the interval (a,b), and $\mathcal{CN}(m,\sigma^2)$ denotes the complex circularly-symmetric Gaussian distribution with mean value m and variance σ^2. $\mathbf{1}_m \in \mathbb{R}^m$ denotes the all-one vector. Mode-n matricization of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I_1 \times \ldots \times I_n \times \ldots \times I_N}$ is denoted as $\mathbf{X}_{(n)} \in \mathbb{C}^{I_n \times \prod_{m \neq n}^N I_m}$, where index n enumerates the rows and the rest enumerates the columns. The mode-n product of the tensor $\boldsymbol{\mathcal{X}}$ and a matrix $\mathbf{A}_n \in \mathbb{R}^{K \times I_n}$ is a tensor $\boldsymbol{\mathcal{X}} \times_n \mathbf{A}_n \in \mathbb{R}^{I_1 \times \ldots \times I_{n-1} \times K \times I_{n+1} \ldots \times I_N}$, whose elements are computed by $[\boldsymbol{\mathcal{X}} \times_n \mathbf{A}_n]_{i_1,\ldots,i_{n-1},k,i_{n+1},\ldots,i_N} = \sum_{i_n=1}^{I_n} x_{i_1,\ldots,i_n,\ldots,i_N} a_{k,i_n}$. Denote $\diamond_{l \neq n}^N \mathbf{A}_l = \mathbf{A}_1 \diamond \ldots \mathbf{A}_{n-1} \diamond \mathbf{A}_{n+1} \diamond \ldots \mathbf{A}_N$ as the Khatri-Rao product of all but one matrix. Similarly, denote $\bigodot_{l \neq n}^N \mathbf{A}_l = \mathbf{A}_1 \odot \ldots \mathbf{A}_{n-1} \odot \mathbf{A}_{n+1} \odot \ldots \mathbf{A}_N$ as the Hadamard product of all but one matrix. A matrix $\mathbf{A} \in \mathbb{C}^{I \times R}$ is said to be (exponential) Vandermonde if its elements are $[\mathbf{A}]_{i,r} = e^{-2\pi j(i-1)\omega_r}$. Denote $\mathbf{A} = \mathrm{Van}(\mathbf{z})$ as the generation of a Vandermonde matrix $\mathbf{A}$ using $\mathbf{z} = 2\pi[\omega_1,\ldots,\omega_R]^T$, where $\mathbf{z}$ is called the generating vector.

III. PRELIMINARIES

Here we introduce the CP decomposition and the ALS algorithm applied on tensor channel estimation. For a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I_1 \times \ldots \times I_N}$, its CP decomposition is formulated as

$$\boldsymbol{\mathcal{X}} = \sum_{r=1}^R \alpha_r \mathbf{a}_{1,r} \circ \ldots \circ \mathbf{a}_{N,r} = [\![\boldsymbol{\alpha}; \mathbf{A}_1,\ldots,\mathbf{A}_N]\!], \quad (1)$$

where α_r is the weight of the rth rank-one component, $[\![\cdot]\!]$ denotes the simplified CP representation and $\boldsymbol{\alpha} = [\alpha_1,\ldots,\alpha_R]^T$. $\mathbf{A}_n = [\mathbf{a}_{n,1},\ldots,\mathbf{a}_{n,R}]$ denotes the factor along the nth mode. The CP rank is defined as the smallest value of R satisfying equation (1). $\boldsymbol{\mathcal{X}}$ is said to be CP low-rank if R is relatively small. Then the high-dimensional $\boldsymbol{\mathcal{X}}$ can be modeled in a low-dimensional parameter space [44], i.e., $R \sum_{n=1}^N I_n \ll \prod_{n=1}^N I_n$. The mode-$n$ matricization of a CP tensor is expressed as $\mathbf{X}_{(n)} = \mathbf{A}_n \boldsymbol{\Lambda}(\diamond_{l \neq n}^N \mathbf{A}_l)^T$, where $\boldsymbol{\Lambda} = \mathrm{diag}(\boldsymbol{\alpha})$. The scaling and permutation ambiguities always exist in the CP model. Specifically, if there are some scaling diagonal matrices $\{\boldsymbol{\Delta}_n\}_{n=1}^N$ and a permutation matrix $\boldsymbol{\Upsilon}$ that map the factors $\{\mathbf{A}_n\}_{n=1}^N$ of (1) as

$$\hat{\mathbf{A}}_n = \mathbf{A}_n \boldsymbol{\Delta}_n \boldsymbol{\Upsilon}, \text{ for } n = 1,\ldots,N,$$
$$\prod_{n=1}^N \boldsymbol{\Delta}_n = \mathrm{diag}(\boldsymbol{\Upsilon}\boldsymbol{\alpha}), \quad (2)$$

then we have

$$\boldsymbol{\mathcal{X}} = [\![\boldsymbol{\alpha}; \mathbf{A}_1,\ldots,\mathbf{A}_N]\!] = [\![\mathbf{1}_R; \hat{\mathbf{A}}_1,\ldots,\hat{\mathbf{A}}_N]\!]. \quad (3)$$

Note that the uniqueness of the CP decomposition does not consider such ambiguity. The following theorem [45] gives a sufficient condition for uniqueness.

Theorem 1 (Uniqueness Condition [45]): Let an Nth-order tensor $\boldsymbol{\mathcal{X}}$ satisfy (1). Then, the CP decomposition is unique if

$$\sum_{n=1}^N \mathrm{krank}(\mathbf{A}_n) \geq 2R + N - 1. \quad (4)$$

When the CP model is applied to the received tensor signals of MIMO communication systems equipped with uniform linear arrays (ULAs) or uniform rectangular arrays (URAs), part of the factors may become Vandermonde matrices, which leads to a VCP model [11], i.e., $\boldsymbol{\mathcal{X}} = [\![\boldsymbol{\alpha}; \mathbf{A}_1,\ldots,\mathbf{A}_{N_1},\mathbf{A}_{N_1+1},\ldots,\mathbf{A}_N]\!]$, where the factors $\{\mathbf{A}_n\}_{n=N_1+1}^N$ are Vandermonde matrices whose elements can be formulated as $[\mathbf{A}_n]_{i_n,r} = e^{-2\pi j(i_n-1)\omega_{n,r}}$, and where the factors $\{\mathbf{A}_n\}_{n=1}^{N_1}$ are generalized matrices with no Vandermonde structure. The generating vectors of the Vandermonde factors can carry information about the AoAs, AoDs, delays and so on, which depends on the specific system as summarized in Table I.

Without accounting for the Vandermonde structure, the ALS algorithm can be used for calculating the VCP decomposition by solving the optimization problem as

$$\min_{\{\hat{\mathbf{A}}_n\}_{n=1}^N} \left\| \boldsymbol{\mathcal{Y}} - [\![\mathbf{1}_R; \hat{\mathbf{A}}_1,\ldots,\hat{\mathbf{A}}_N]\!] \right\|_F^2. \quad (5)$$

The ALS updates factor $\hat{\mathbf{A}}_n$ alternatingly by fixing the other factors. Thus, the optimization problem (5) in the mode-n matricization form becomes

$$\min_{\hat{\mathbf{A}}_n} \left\| \mathbf{Y}_{(n)} - \hat{\mathbf{A}}_n \left(\diamond_{l \neq n}^N \hat{\mathbf{A}}_l \right)^T \right\|_F^2, \quad (6)$$

and its LS solution is given as

$$\hat{\mathbf{A}}_n = \mathbf{Y}_{(n)} \left(\diamond_{l \neq n}^N \hat{\mathbf{A}}_l \right)^{T,\dagger}$$
$$= \mathbf{Y}_{(n)} \left(\diamond_{l \neq n}^N \hat{\mathbf{A}}_l^* \right) \left(\bigodot_{l \neq n}^N \hat{\mathbf{A}}_l^T \hat{\mathbf{A}}_l^* \right)^\dagger. \quad (7)$$

We summarize the CPALS in Algorithm 1. Note that the tolerance on the difference of the factors or reconstructed tensor between two adjacent iterations can also be used as the terminating condition. Nevertheless, we use the number of iterations K as the terminating condition to ease description. To simplify notations, we denote Φ as the operator to calculate CPALS in Algorithm 1, which is formulated as

$$\{\hat{\mathbf{A}}_n^K\}_{n=1}^N = \Phi_{R,K}(\boldsymbol{\mathcal{Y}}, \{\hat{\mathbf{A}}_n^0\}_{n=2}^N), \quad (8)$$

where $\boldsymbol{\mathcal{Y}}$ is the input tensor that needs to be decomposed and $\hat{\mathbf{A}}_n^0$ is the initial factor for $n = 2,\ldots,N$. The approximated CP rank R and the number of iterations K are treated as the predefined hyper-parameters of Φ.

After the ALS terminates, we can extract the rth elements of the generating vectors $\{\hat{\mathbf{z}}_n\}_{n=N_1+1}^N$ of the Vandermonde factors by solving the maximum correlation problem

$$\max_{\hat{z}_{n,r}} \frac{\left| \hat{\mathbf{a}}_{n,r}^H \mathrm{Van}(\hat{z}_{n,r}) \right|}{\|\hat{\mathbf{a}}_{n,r}\|_2}, \quad (9)$$

Algorithm 1 CPALS $\Phi_{R,K}(\boldsymbol{\mathcal{Y}}, \{\hat{\mathbf{A}}_n^0\}_{n=2}^N)$

Input: tensor $\boldsymbol{\mathcal{Y}}$, initializations $\{\hat{\mathbf{A}}_n^0\}_{n=2}^N$, CP rank R and the number of iterations K

Output: updated factors $\{\hat{\mathbf{A}}_n^K\}_{n=1}^N$

1: Update $\hat{\mathbf{A}}_1^0 \leftarrow \mathbf{Y}_{(1)} \left(\diamondsuit_{n=2}^N \hat{\mathbf{A}}_n^{0,*}\right) \left(\odot_{n=2}^N \hat{\mathbf{A}}_n^{0,T} \hat{\mathbf{A}}_n^{0,*}\right)^\dagger$.

2: **for** $k = 1 : K$ **do**

3:      **for** $n = 1 : N$ **do**

4:          Update $\hat{\mathbf{A}}_n^k$ by calculating (7).

5:      **end for**

6: **end for**

---

where $\hat{\mathbf{a}}_{n,r}$ is the rth column of $\hat{\mathbf{A}}_n$. The above problem can be solved by a simple one-dimensional search method.

IV. DEEP-LEARNING-AIDED CP DECOMPOSITION

In this section, we integrate the CP decomposition into the data-driven learning process and propose the deep-learning-aided CPALS, where we use a DNN to generate the initializations that are favorable for CP decomposition. Hence, the proposed method enhances the algorithm's accuracy and efficiency.

A. Motivations

Researchers have long found in experiments that unfavorable starting points can bring CPALS into swamps [28], [46], where CPALS has not converged to a minimum and using more iterations would not further reduce the objective value. Favorable initializations not only speed up CPALS in finding a mimimum, but also can ensure the global or local convergence theoretically [47], [48]. As the initialization plays an important role for CPALS, two questions naturally arise. *What are favorable initializations for CPALS*? And *how to generate favorable initializations*? Next, we address these two questions as a way to motivate our proposed approach.

What are favorable initializations for CPALS? To answer this question, we first illustrate the iterative behaviour of CPALS using different initializations. Consider a synthetic VCP tensor $\boldsymbol{\mathcal{Y}} \in \mathbb{C}^{4\times4\times4}$ with $R = 2$, whose factors are all Vandermonde matrices. Elements of the generating vectors are drawn from $\mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2})$ randomly, which are $\mathbf{z}_1 = [0.36, 0.18]^T$, $\mathbf{z}_2 = [1.10, -0.70]^T$ and $\mathbf{z}_3 = [-0.58, 0.89]^T$. Note that this tensor satisfies the uniqueness condition of Theorem 1. Here we only consider to decompose $\boldsymbol{\mathcal{Y}}$ with CPALS without calculating (9) for extracting parameters. We need to provide initializations of $\hat{\mathbf{z}}_2^0$ and $\hat{\mathbf{z}}_3^0$ to construct the initialization factors $\{\hat{\mathbf{A}}_n^0\}_{n=2}^3$, as shown in Algorithm 1. We use the randomly generated initialization $\hat{\mathbf{z}}_2^0 = [-0.46, -0.85]^T$, and evaluate the performance of CPALS with different starting points of $\hat{\mathbf{z}}_3^0 = [z_{3,1}, z_{3,2}]^T$. In Figure 1 (a) and 1 (b), we show the normalized square error (NSE) calculated by $\frac{\|\hat{\boldsymbol{\mathcal{Y}}} - \boldsymbol{\mathcal{Y}}\|_F^2}{\|\boldsymbol{\mathcal{Y}}\|_F^2}$, where $\hat{\boldsymbol{\mathcal{Y}}} = [\![\mathbf{1}_R; \hat{\mathbf{A}}_1^K, \hat{\mathbf{A}}_2^K, \hat{\mathbf{A}}_3^K]\!]$ is the reconstructed tensor by using CPALS. As shown in Figure 1 (a), four regions, which are surrounded by dashed circles, give favorable initializations that allow CPALS to quickly reduce the estimation
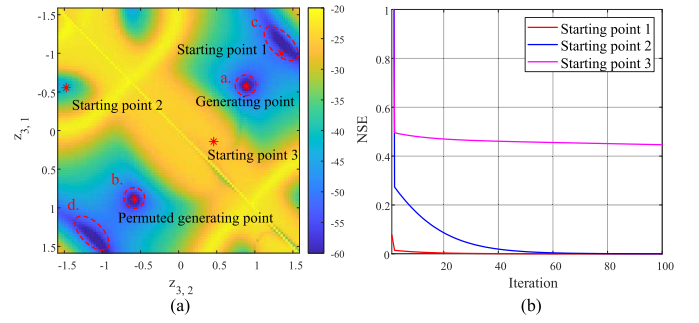

Fig. 1. Iterative behaviour of ALS using different starting points of $\hat{\mathbf{z}}_3^0 = [z_{3,1}, z_{3,2}]^T$ for computing a random tensor $\boldsymbol{\mathcal{Y}}$. (a) The reconstruction NSE of $\frac{\|\hat{\boldsymbol{\mathcal{Y}}} - \boldsymbol{\mathcal{Y}}\|_F^2}{\|\boldsymbol{\mathcal{Y}}\|_F^2}$ in dB with $K = 50$ using different starting points, where $\hat{\boldsymbol{\mathcal{Y}}}$ is the reconstructed tensor. (b) The reconstruction NSE as a function of iteration number.

error to 0. The first two regions, i.e., region a and region b, are near the true generating point $\mathbf{z}_3 = [0.89, -0.58]^T$ and its permuted point $[-0.58, 0.89]^T$ due to the permutation ambiguity. Interestingly, there are two more such regions, i.e., region c near the starting point 1 and the permuted region d. In Figure 1 (b), we plot the NSE as a function of the iteration number for CPALS by using the selected starting points. We can observe that the objective value using starting point 3 reduces slowly like in a swamp, but it reduces quickly using starting point 1. Hence, we could describe that favorable initializations would make the ALS reduce to a low objective value or decomposed error using just a few iterations. Note that when $\boldsymbol{\mathcal{Y}}$ is disturbed by the noise $\boldsymbol{\mathcal{N}}$, a low objective value of the approximation problem in (5), i.e., $\left\|\hat{\boldsymbol{\mathcal{Y}}} - \boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{N}}\right\|_F^2$, may not ensure a low NSE. Nevertheless, using a small objective value as the guidance is feasible if the uniqueness condition is satisfied and the noise is small.

How to generate favorable initializations? We denote $\Psi(\boldsymbol{\mathcal{Y}})$ as a functional generator that produces initializations for CPALS to decompose $\boldsymbol{\mathcal{Y}}$. In addition to the random initialization, the most commonly used strategy is based on the SVD, i.e.,

$$\Psi(\boldsymbol{\mathcal{Y}}): \hat{\mathbf{A}}_n^0 = R \text{ leading left singular vectors of } \mathbf{Y}_{(n)},$$
$$\text{for } n = 2, \ldots, N. \tag{10}$$

However, this initialization strategy fails to consider the updating steps of CPALS in the sequel and does not ensure to generate favorable initializations. Although one can also use other non-iterative decomposition methods like the generalized eigenvalue decomposition (GEVD) to pre-estimate a starting point close to the ground truth as initialization [49], they have higher computational complexity than using the SVD. We need a Ψ providing a guidance that prompts an accurate and fast CP decomposition by producing favorable initializations for the input tensor. Thus, Ψ needs outer domain knowledge learned from a large number of tensor decomposition processes, which inspires us to employ a DNN.

B. The Proposed Deep-Learning-Aided CPALS (DL-CPALS)

We consider to develop a DNN based generator Ψ_{θ} with parameters θ, which learns to produce favorable initial parameters to decompose new tensors fast and well. Note that the CPALS for a tensor is naturally treated as a model-driven learning task, which updates its model parameters, i.e., the CP factors, starting from the initializations. In this way, our goal is to pursue favorable initializations using a learned initialization generator Ψ_{θ}. Given a batch of CPALS tasks for decomposing tensors $\mathcal{Y}_1, \ldots, \mathcal{Y}_{L_{\mathrm{tr}}}$, we can integrate them into the data-driven learning process of Ψ_{θ}. If the unrolled CPALS is differentiable for updating the CP factors, gradients can be backpropagated to train Ψ_{θ} for generating favorable initializations. Specifically, the loss function of the learning process is formulated as

$$
\begin{aligned}
\min_{\theta} \quad & \frac{1}{L_{\mathrm{tr}}} \sum_{l=1}^{L_{\mathrm{tr}}} \left\| \mathcal{Y}_l - [\![\mathbf{1}_R; \hat{\mathbf{A}}_{l,1}^K, \ldots, \hat{\mathbf{A}}_{l,N}^K]\!] \right\|_F^2, \\
\text{s.t.} \quad & \{\hat{\mathbf{A}}_{l,n}^0\}_{n=2}^N = \Psi_{\theta}(\mathcal{Y}_l), \; \forall l, \\
& \{\hat{\mathbf{A}}_{l,n}^K\}_{n=1}^N = \Phi_{R,K}(\mathcal{Y}_l, \{\hat{\mathbf{A}}_{l,n}^0\}_{n=2}^N), \; \forall l.
\end{aligned} \tag{11}
$$

Note that the loss matches the objective function of the CP low-rank approximation problem in (5). When we set a small K, the proposed method aims to optimize a DNN for generating the initializations such that a small number of ALS iterations on a CP decomposition task will produce a maximally effective behavior to reduce the approximation error. Thus, the learned DNN would be a good initialization generator that can let the ALS converge to a low objective error quickly for decomposing an input tensor. Moreover, as a model-based optimization algorithm, CPALS is able to find a clean low-rank tensor from a noisy high-rank tensor by solving the low-rank approximation problem, which leads to a denoising effect. Hence, optimizing the loss function (11) and training the DNN Ψ_{θ} does not need the paired clean data. The training data $\mathcal{Y}_1, \ldots, \mathcal{Y}_{L_{\mathrm{tr}}}$ can be noisy, which improves the practicability of the proposed method, as clean data are not available in many cases, e.g., wireless channel responses.

We provide the training steps of DL-CPALS in Algorithm 2, where we use a batch-wise training strategy. A prerequisite to optimize the loss function (11) is that the CPALS $\Phi_{R,K}$ is differentiable in the updated factors for all tensors, which ensures the gradient back-propagation in line 6 of Algorithm 2. As shown in (7), the iterative updating steps of $\Phi_{R,K}$ employ a multiplication, Khatri-Rao product, Hadamard product and pseudoinverse of matrices, which are all differentiable [50]. The open source deep learning framework PyTorch [51] is able to compute gradients of these matrix operations automatically.[1] If $\bigodot_{l \neq n}^N \hat{\mathbf{A}}_l^T \hat{\mathbf{A}}_l^*$ has full rank, its pseudoinverse is equal to the inverse. Using the inverse operation can simplify the gradient calculation and make the gradient numerically stable. As we find that $\bigodot_{l \neq n}^N \hat{\mathbf{A}}_l^T \hat{\mathbf{A}}_l^*$ is usually full rank in experiments, we suggest using the inverse operation. Because the derivatives of $\Phi_{R,K}$ are complicated, here we only formulate the updating

---

[1] https://pytorch.org/docs/stable/generated/torch.linalg.pinv.html

---

Algorithm 2 Training of DL-CPALS

Input: the tensor dataset $\{\mathcal{Y}_l\}_{l=1}^{L_{\mathrm{tr}}}$ obeying $p(\mathcal{Y})$, the CP rank R, the number of iterations K, the number of epochs N_{epoch} and the tensor mini-batch size L

Output: DNN Ψ_{θ}

1: **for** $i = 1 : N_{\mathrm{epoch}}$ **do**
2:     **for** $j = 1 : L_{\mathrm{tr}}/L$ **do**
3:         Draw L tensors from dataset without replacement.
4:         **for** $l = 1 : L$ **do**
5:             Input $\mathcal{Y}_l$ into Ψ_{θ} and obtain the initialization factors $\{\hat{\mathbf{A}}_{l,n}^0\}_{n=2}^N = \Psi_{\theta}(\mathcal{Y}_l)$.
6:             Carry out CPALS and obtain updated factors $\{\hat{\mathbf{A}}_{l,n}^K\}_{n=1}^N = \Phi_{R,K}(\mathcal{Y}_l, \{\hat{\mathbf{A}}_{l,n}^0\}_{n=2}^N)$.
7:             Calculate the loss of low-rank CP approximation, i.e., $\mathcal{L}_l = \left\| \mathcal{Y}_l - [\![\mathbf{1}_R; \hat{\mathbf{A}}_{l,1}^K, \ldots, \hat{\mathbf{A}}_{l,N}^K]\!] \right\|_F^2$
8:         **end for**
9:         Calculate the mini-batch-wise loss $\frac{1}{L} \sum_{l=1}^L \mathcal{L}_l$, and update θ using optimizer.
10:     **end for**
11: **end for**

---

of $\hat{\mathbf{A}}_1^0$ with $N = 3$ for instance, which is processed as

$$
\begin{aligned}
\hat{\mathbf{A}}_1^0 &= \Phi_{R,0}(\mathcal{Y}, \{\hat{\mathbf{A}}_2^0, \hat{\mathbf{A}}_3^0\}) \\
&= \mathbf{Y}_{(1)} \left(\hat{\mathbf{A}}_2^{0,*} \diamond \hat{\mathbf{A}}_3^{0,*} \right) \left(\hat{\mathbf{A}}_2^{0,T} \hat{\mathbf{A}}_2^{0,*} \odot \hat{\mathbf{A}}_3^{0,T} \hat{\mathbf{A}}_3^{0,*} \right)^{-1}.
\end{aligned} \tag{12}
$$

Due to $\{\hat{\mathbf{A}}_2^0, \hat{\mathbf{A}}_3^0\} = \Psi_{\theta}(\mathcal{Y})$, both $\hat{\mathbf{A}}_2^0 \in \mathbb{C}^{I_2 \times R}$ and $\hat{\mathbf{A}}_3^0 \in \mathbb{C}^{I_3 \times R}$ are related to the DNN parameters θ. For ease of formulation, we only consider $\hat{\mathbf{A}}_2^0$ as a variable matrix and fix $\hat{\mathbf{A}}_3^0$ as a constant. Let

$$
\mathbf{g}(\mathrm{vec}(\hat{\mathbf{A}}_2^0)) = \mathrm{vec}\left(\left(\hat{\mathbf{A}}_2^{0,T} \hat{\mathbf{A}}_2^{0,*} \odot \hat{\mathbf{A}}_3^{0,T} \hat{\mathbf{A}}_3^{0,*} \right)^{-1} \right) \in \mathbb{C}^{R^2} \tag{13}
$$

represent a matrix function of $\hat{\mathbf{A}}_2^0$, where $\mathrm{vec}(\cdot)$ is the vectorization operator that stacks the column vectors of the argument matrix into a long column vector. The matrix function $\mathbf{g}$ maps an I_2R-dimensional variable to the R^2-dimensional space. We define the derivative as $\mathcal{D}_{\mathrm{vec}(\hat{\mathbf{A}}_2^0)}\mathbf{g} \in \mathbb{C}^{R^2 \times I_2 R}$. First, we have

$$
\mathrm{vec}\left(\hat{\mathbf{A}}_2^{0,T} \hat{\mathbf{A}}_2^{0,*} \right) = \left(\hat{\mathbf{A}}_2^{0,H} \otimes \mathbf{I}_R \right) \mathrm{vec}\left(\hat{\mathbf{A}}_2^{0,T} \right). \tag{14}
$$

Thus, we can obtain

$$
\mathcal{D}_{\mathrm{vec}(\hat{\mathbf{A}}_2^0)} \mathrm{vec}\left(\hat{\mathbf{A}}_2^{0,T} \hat{\mathbf{A}}_2^{0,*} \right) = \left(\hat{\mathbf{A}}_2^{0,H} \otimes \mathbf{I}_R \right) \mathbf{K}_{I_2,R}, \tag{15}
$$

where $\mathbf{K}_{k,l} \in \mathbb{R}^{kl \times kl}$ is called the commutation matrix satisfying $\mathbf{K}_{k,l}\mathrm{vec}(\mathbf{Z}) = \mathrm{vec}(\mathbf{Z}^T)$ for the matrix $\mathbf{Z} \in \mathbb{C}^{k \times l}$. According to [48], two important expressions are

$$
\begin{aligned}
\mathcal{D}_{\mathrm{vec}(\mathbf{X})}\mathrm{vec}\left(\mathbf{X}^{-1} \right) &= -\mathbf{X}^{-T} \otimes \mathbf{X}^{-1}, \\
\mathcal{D}_{\mathrm{vec}(\mathbf{X})}\mathrm{vec}\left(\mathbf{X} \odot \mathbf{Y} \right) &= \mathrm{diag}(\mathrm{vec}(\mathbf{Y})).
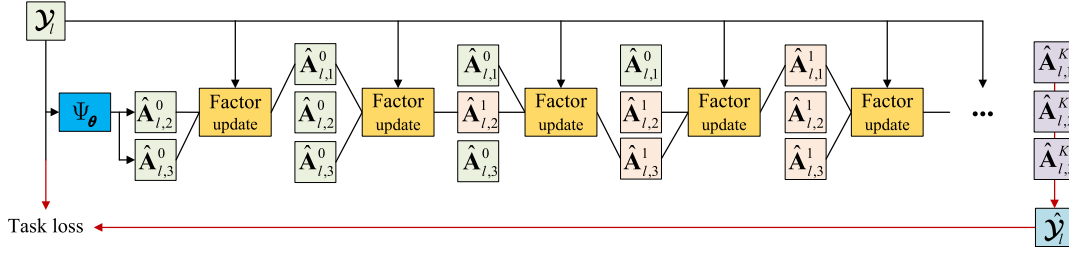\end{aligned} \tag{16}
$$

Fig. 2.   An illustration of the model forward propagation on a single tensor.

By using (13), (15) and (16), we can obtain

$$
\begin{aligned}
&\mathcal{D}_{\mathrm{vec}(\hat{\mathbf{A}}_2^0)}\mathbf{g} \\
&= -\left(\mathbf{g}^T \otimes \mathbf{g}\right)\mathrm{diag}(\mathrm{vec}(\hat{\mathbf{A}}_3^{0,T}\hat{\mathbf{A}}_3^{0,*}))\left(\hat{\mathbf{A}}_2^{0,H} \otimes \mathbf{I}_R\right)\mathbf{K}_{I_2,R}.
\end{aligned}
$$
(17)

Then, we can derive

$$
\mathcal{D}_{\mathrm{vec}(\hat{\mathbf{A}}_2^0)}\mathrm{vec}(\Phi_{R,0}) = \left(\mathbf{I}_R \otimes \mathbf{Y}_{(1)}\left(\hat{\mathbf{A}}_2^{0,*}\diamond\hat{\mathbf{A}}_3^{0,*}\right)\right)\mathcal{D}_{\mathrm{vec}(\hat{\mathbf{A}}_2^0)}\mathbf{g}.
$$
(18)

In this way, the differentiable iterative updating steps of CPALS eventually leads to the learnable Ψ_θ.

Moreover, we illustrate the forward propagation of a 3rd order tensor in Figure 2, which corresponds to line 5 and line 6 of Algorithm 2. It can be seen that CPALS is unrolled by multiple factor updating steps using (7). As long as the factor updating steps are differentiable, the DNN parameters θ can be optimized for learning by gradient back-propagation according to the updating loss in line 9 of Algorithm 2. After the learning terminates, one can obtain the favorable initializations by feeding the test tensor into Ψ_θ. Then, performing CPALS can produce an accurate decomposition solution with high efficiency.

C. DNN Designs

In this subsection, we describe the details about the network design of the generator Ψ_θ. Due to the CP low-rankness assumption that R is small in (11), the initializations $\{\hat{\mathbf{A}}_{l,n}^0\}_{n=2}^N$ live in a low-dimensional parameter space in comparison with the reconstructed tensor. The CPALS $\Phi_{R,K}$ uses a low-dimensional initialization to produce a low-rank approximation of the high-dimensional target tensor, which is like an explicit decoder. Thus, it is natural to design the generator Ψ_θ as an encoder. In addition, Ψ_θ should have a low computational complexity to compete with the SVD-based initialization. We use a simple fully connected network as generator, which outputs initial factors $\{\hat{\mathbf{A}}_{l,n}^0\}_{n=2}^N$. Note that the generator Ψ_θ directly outputs the factor matrices rather than generating vectors by utilizing the Vandermonde structure, which makes the proposed method suitable for a generalized CP decomposition.

Given an input tensor $\mathcal{Y}_l \in \mathbb{C}^{I_1 \times \cdots \times I_N}$, we reshape it into a vector, and concatenate its real part and imaginary part into a single vector, which has dimension $2\prod_{n=1}^N I_n$. Assume that each hidden layer has Q neurons. Then, the real weight

matrices employed on the input layer and the hidden layer are of sizes $2\prod_{n=1}^N I_n \times Q$ and $Q \times Q$, respectively. For the output layer, the weight matrix has size of $Q \times 2R\sum_{n=2}^N I_n$. In specific for the MIMO channel estimation application of Section V, the input dimensions $2I_1I_2M$ and output dimension $2R(I_1 + M)$ are determined by the numbers of BS antennas I_2, MS antennas I_1, frequency subcarriers M and paths R of MIMO systems and channels. We use a Relu activation for the hidden layers. As for the activation of the output layer, we suggest to use the hyperbolic tangent function (Tanh), which constrains the output values in the interval -1 to $+1$ so that the energies of the initial factors are not very large and result in imbalance.

If the Ψ_θ is stacked by D hidden layers, considering also the biases, its total number of parameters is $2Q\prod_{n=1}^N I_n + (D - 1)Q^2 + 2QR\sum_{n=2}^N I_n + DQ + 2R\sum_{n=2}^N I_n$. If the input is a real tensor, the total number of parameters is $Q\prod_{n=1}^N I_n + (D-1)Q^2 + QR\sum_{n=2}^N I_n + DQ + R\sum_{n=2}^N I_n$. In addition, in order to deal with the latent overfitting problem under limited training samples, a dropout layer is added after the activation function of each hidden layer to improve the generalization ability of the DNN model.

V. Application to Massive MIMO-OFDM Tensor Channel Estimation

Although the CPALS can be applied on a variety of massive MIMO systems that are not limited to those listed in Table I, we consider a representative application to MIMO-OFDM tensor channel estimation.

We first consider a downlink MIMO-OFDM system where both the mobile station (MS) and the base station (BS) are equipped with ULAs, which are of size I_1 and I_2, respectively. Suppose also that there are in total M_0 subcarriers with the sampling rate of f_s. As described in [8], [52], and [37], consider the mmWave channel is formulated based on the geometry-based channel model that has R resolvable paths without clustering. Denote ϑ_r and θ_r as the AoA and AoD of the rth path, respectively. To simplify notations, we define $\Theta_{1,r} = 2\pi d\sin\vartheta_r/\lambda$ and $\Theta_{2,r} = 2\pi d\sin\theta_r/\lambda$ as the spatial AoA and AoD. The MIMO-OFDM channel in the spatial-frequency domain can then be written in CP form as

$$
\begin{aligned}
\mathcal{H}_0 &= \sum_{r=1}^R \beta_r e^{-j2\pi\frac{\tau_r f_s}{M_0}}\mathrm{Van}(\Theta_{1,r}) \circ \mathrm{Van}(\Theta_{2,r}) \circ \mathrm{Van}(\frac{2\pi\tau_r f_s}{M_0}) \\
&= [\![\boldsymbol{\beta}; \mathbf{A}_1, \mathbf{A}_2, \bar{\mathbf{A}}_3]\!] \in \mathbb{C}^{I_1 \times I_2 \times M_0},
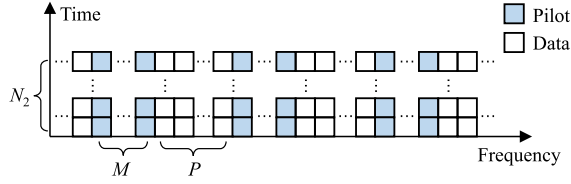\end{aligned}
$$
(19)

Fig. 3. An illustration of the pilot pattern.

where β_r and τ_r are the equivalent path gain and delay of the rth path, respectively. We define $\boldsymbol{\beta} = [\beta_1 e^{-j2\pi\frac{\tau_1 f_s}{M_0}}, \ldots, \beta_R e^{-j2\pi\frac{\tau_R f_s}{M_0}}]^T$. We further have $\mathbf{A}_1 = [\mathrm{Van}(\Theta_{1,1}), \ldots, \mathrm{Van}_1(\Theta_{1,R})] \in \mathbb{C}^{I_1 \times R}$, $\mathbf{A}_2 = [\mathrm{Van}_2(\Theta_{2,1}), \ldots, \mathrm{Van}_2(\Theta_{2,R})] \in \mathbb{C}^{I_2 \times R}$, and $\bar{\mathbf{A}}_3 = [\mathrm{Van}(\frac{2\pi\tau_1 f_s}{M_0}), \ldots, \mathrm{Van}(\frac{2\pi\tau_R f_s}{M_0})] \in \mathbb{C}^{M_0 \times R}$, which are stacked by the steering vectors corresponding to the receiving ULA, transmitting ULA and subcarriers, respectively.

To facilitate the comparison of our method with others in experiments, we follow the pilot transmission protocol described in [37]. As the frequency correlation is helpful to improve the channel estimation accuracy, suppose that a training block has M sequential subcarriers in the frequency domain to transmit N_2 training pilots in the time domain, which is illustrated in Figure 3. Two training blocks are separated by P subcarriers dedicated to data transmission. We only consider the estimation of the tensor channel of a training block that corresponds to the pilot positions. The channel is assumed to be block-fading, and it is constant during training. Based on the pilot channels, interpolation can be used to get the data channels. The training tensor channel of the tth block is given by

$$\boldsymbol{\mathcal{H}}_t = [\boldsymbol{\mathcal{H}}_{0,t}]_{:,:,m_s:m_s+M-1} = [\![\boldsymbol{\beta}_t; \mathbf{A}_{1,t}, \mathbf{A}_{2,t}, \mathbf{A}_{3,t}]\!], \quad (20)$$

where m_s denotes the starting index and $\mathbf{A}_{3,t}$ consists of a group of M rows of $\bar{\mathbf{A}}_{3,t}$. Note that the parameters $\{\beta_{r,t}\}_{r=1}^R, \{\tau_{r,t}\}_{r=1}^R, \{\Theta_{1,r,t}\}_{r=1}^R$ and $\{\Theta_{2,r,t}\}_{r=1}^R$ determine $\boldsymbol{\beta}_t, \mathbf{A}_{1,t}, \mathbf{A}_{2,t}$ and $\mathbf{A}_{3,t}$, respectively. For the tensor channel, the BS only activates one RF chain to transmit a pilot symbol x_{n_2} using a beamforming vector $\mathbf{f}_{2,n_2} \in \mathbb{C}^{I_2 \times 1}$. During the transmission of the pilot x_{n_2} at the BS, the MS employs N_1 combining vectors $\mathbf{f}_{1,1}, \ldots, \mathbf{f}_{1,N_1} \in \mathbb{C}^{I_1 \times 1}$ to process it. Note that the RF chains of the MS need to be reused when its total number is less than N_1. After transmitting N_2 pilots, the received baseband signal tensor can be written as

$$\boldsymbol{\mathcal{Y}}_t = \boldsymbol{\mathcal{H}}_t \times_1 \mathbf{F}_1^H \times_2 \mathrm{diag}(\mathbf{x})\mathbf{F}_2^T + \boldsymbol{\mathcal{N}}_t \times_1 \mathbf{F}_1^H \in \mathbb{C}^{N_1 \times N_2 \times M}, \quad (21)$$

where $\mathbf{x} = [x_1, \ldots, x_{N_2}]^T$, $\boldsymbol{\mathcal{N}}_t \in \mathbb{C}^{I_1 \times N_2 \times M}$ is the noise tensor. $\mathbf{F}_1 = [\mathbf{f}_{1,1}, \ldots, \mathbf{f}_{1,N_1}] \in \mathbb{C}^{I_1 \times N_1}$ and $\mathbf{F}_2 = [\mathbf{f}_{2,1}, \ldots, \mathbf{f}_{2,N_2}] \in \mathbb{C}^{I_2 \times N_2}$ are the known combiner and beamformer, respectively. Suppose that $\mathbf{x} = \mathbf{1}_{N_2}$ and the rows of $\mathbf{F}_1$ and $\mathbf{F}_2$ are orthogonal with $N_1 = I_1$ and $N_2 = I_2$. Then, according to (20) and (21), we can obtain a coarse estimation of $\boldsymbol{\mathcal{H}}$, which is processed as

$$\begin{aligned} \bar{\boldsymbol{\mathcal{H}}}_t &= \boldsymbol{\mathcal{Y}}_t \times_1 \mathbf{F}_1 \times_2 \mathbf{F}_2^* \\ &= \boldsymbol{\mathcal{H}}_t \times_1 \mathbf{F}_1 \mathbf{F}_1^H \times_2 \mathbf{F}_2^* \mathbf{F}_2^T + \boldsymbol{\mathcal{N}}_t \times_1 \mathbf{F}_1 \mathbf{F}_1^H \times_2 \mathbf{F}_2^* \end{aligned}$$

$$\begin{aligned} &= \boldsymbol{\mathcal{H}}_t + \boldsymbol{\mathcal{N}}_t \times_2 \mathbf{F}_2^* \\ &= [\![\boldsymbol{\beta}_t; \mathbf{A}_{1,t}, \mathbf{A}_{2,t}, \mathbf{A}_{3,t}]\!] + \boldsymbol{\mathcal{N}}_t \times_2 \mathbf{F}_2^*. \quad (22) \end{aligned}$$

In this way, one can collect plentiful samples $\{\bar{\boldsymbol{\mathcal{H}}}_t\}$ at the receiver. As the least-squares-based coarse estimation can not remove the noise, the obtained a dataset of the noisy channels that characterizes a distribution $p(\bar{\boldsymbol{\mathcal{H}}})$. Fortunately, the proposed DL-CPALS can be trained by only using the noisy tensor $\bar{\boldsymbol{\mathcal{H}}}$ according to Algorithm 2. Once the model is trained, in the inference phase, the pre-estimated channel is used as input to the model $\Psi_{\boldsymbol{\theta}}$ in order to obtain favorable initialization factors. Then, the ALS with a small number of iterations is calculated to compute the estimated (denoised) channel. In addition, one can also estimate the channel parameters for each sample $\bar{\boldsymbol{\mathcal{H}}}_t$, such as the spatial AoAs, spatial AoDs and delays from the decomposed factors $\mathbf{A}_{1,t}, \mathbf{A}_{2,t}$ and $\mathbf{A}_{3,t}$ by solving the optimization problem (9).

VI. SIMULATIONS

In this section, we evaluate the tensor CP decomposition and channel estimation performance of the proposed DL-CPALS in the considered mmWave MIMO-OFDM systems.

A. Results on Synthetic Data

We first compare the low-rank CP approximation performance of the proposed DL-CPALS with CPALS with random initializations (Random-CPALS), CPALS with SVD-based initializations (SVD-CPALS) and CPALS with GEVD-based initializations (GEVD-CPALS) [49].[2] Consider a noisy low-rank CP tensor distribution $p(\boldsymbol{\mathcal{X}}_{\mathrm{no}})$ with $\boldsymbol{\mathcal{X}}_{\mathrm{no},t} = \boldsymbol{\mathcal{X}}_t + \boldsymbol{\mathcal{N}}_t \in \mathbb{R}^{6\times6\times6}$, where the CP rank of $\boldsymbol{\mathcal{X}}_t$ is set as $R = 3$ and $\boldsymbol{\mathcal{N}}_t$ is Gaussian noise. Based on the CP model of $\boldsymbol{\mathcal{X}}_t$ in (1), the elements of the factors $\{\mathbf{A}_{n,t} \in \mathbb{R}^{6\times3}\}_{n=1}^3$ and the weight vector $\boldsymbol{\alpha}_t \in \mathbb{R}^{3\times1}$ are all drawn from the uniform distribution $\mathcal{U}(0,1)$. The factors $\{\mathbf{A}_{n,t} \in \mathbb{R}^{6\times3}\}_{n=1}^3$ have full column rank almost surely due to the randomness. According to Theorem 1, the constructed tensor is unique if $R < 3.5$. Hence $R = 3$ satisfies the sufficient condition of uniqueness. Then, by setting the signal-to-noise-ratio (SNR) to 15dB, we generate 100000 noisy tensors randomly of $\{\boldsymbol{\mathcal{X}}_{\mathrm{no},t}\}_{t=1}^{100000}$, where $L_{\mathrm{tr}} = 90000$ samples are used for training and $L_{\mathrm{te}} = 10000$ samples are used for testing. For the DNN model as described in Subsection IV-C, we set the number of hidden layers to $D = 4$, the number of neurons to $Q = 512$ and the dropout rate to 0.3. Thus, the total number of model parameters is 917540. For the model training as given in Algorithm 2, the DL-CPALS uses the Adam optimizer [53] with a learning rate 0.00002 and mini-batch size $L = 512$. The number of learning steps is decided by the number of epochs, which is set to $N_{\mathrm{epoch}} = 1000$. As for the setting of the iteration number K, we set $K = 2$ for training because using a larger K does not lead to significant gains in the experiments but bring more complexity for calculating the gradients. Unless otherwise specified, K is set to 50 for testing. The performance of the competing CPALS method is evaluated
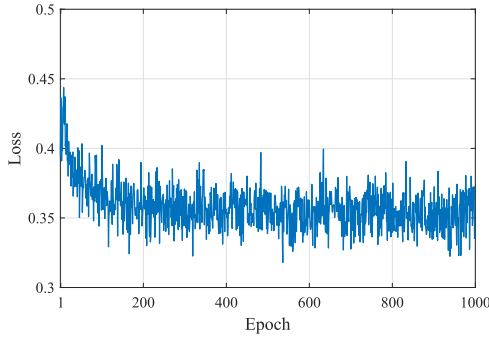
[2]https://www.tensorlab.net/
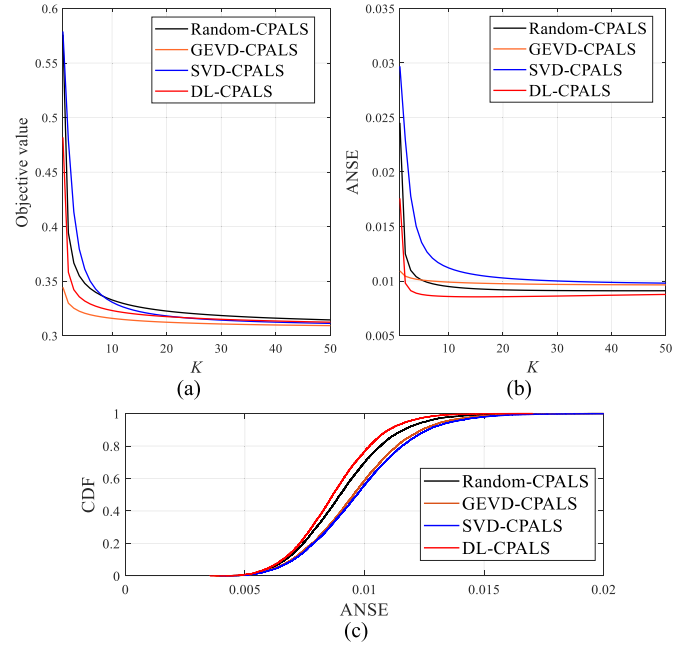
Fig. 4. The training loss of DL-CPALS.

Fig. 5. The performance comparison of CPALS using different initializations on synthetic data. (a) Iterative behavior of the objective value. (b) Iterative behavior of the ANSE. (c) Curves of the cumulative distribution function (CDF) of the ANSE with $K = 50$.

from two aspects: the iterative behavior of the objective value, i.e.,

$$\frac{1}{L_{\text{te}}} \sum_{t=1}^{L_{\text{te}}} \left\| \boldsymbol{\mathcal{X}}_{\text{no},t} - [\![\mathbf{1}_R; \hat{\mathbf{A}}_{1,t}^K, \hat{\mathbf{A}}_{2,t}^K, \hat{\mathbf{A}}_{3,t}^K]\!] \right\|_F^2, \qquad (23)$$

and the iterative behavior of the average NSE (ANSE),[3] i.e,

$$\frac{1}{L_{\text{te}}} \sum_{t=1}^{L_{\text{te}}} \frac{\left\| \boldsymbol{\mathcal{X}}_t - [\![\mathbf{1}_R; \hat{\mathbf{A}}_{1,t}^K, \hat{\mathbf{A}}_{2,t}^K, \hat{\mathbf{A}}_{3,t}^K]\!] \right\|_F^2}{\| \boldsymbol{\mathcal{X}}_t \|_F^2}. \qquad (24)$$

In Figure 4, we first show the training loss of the proposed DL-CPALS. It can be seen that the loss value generally decreases and tends to converge as the number of epochs increases, which demonstrates that the proposed DL-CPALS is learnable. In Figure 5, we plot the performance versus number of iterations of all test samples using different initializations. Note that the elements of the initial factors for the Random-CPALS obey the uniform distribution $\mathcal{U}(0, 1)$, which actually means non-negative priors are used. Figure 5(a) shows that the convergence of the proposed DL-CPALS is the steepest, because the trained model produces favorable initializations that reduce the objective value quickly. Although the GEVD makes a good initial guess, it fails to consider the updating steps of CPALS in the sequel. Moreover, the objective value with $K = 2$, i.e., the test loss after training is complete, is 0.36 in Figure 5(a), which is close to the final training loss in Figure 4. In Figure 5(b), it can be seen that the proposed method achieves the lowest reconstruction error by only using a small number of iterations (about $K = 5$), while the competing Random-CPALS, GEVD-CPALS and SVD-CPALS fail to reach the same accuracy as DL-CPALS even with $K = 50$. In order to observe the ANSE distribution of the competing methods on all testing samples, we plot the CDF curves in Figure 5(c). It can be seen that DL-CPALS has the highest probability of reaching a specific low error. Based on these results, the proposed DL-CPALS leads to an accurate and fast low-rank CP decomposition.

B. Results of MIMO-OFDM Channel Estimation

Consider the ULAs of the BS and the MS are of size $I_2 = 32$ and $I_1 = 8$, respectively. The inter-element spacing d of

---

[3]Note that each true sample $\boldsymbol{\mathcal{X}}_t$ has a single estimation trial $[\![\mathbf{1}_R; \hat{\mathbf{A}}_{1,t}^K, \hat{\mathbf{A}}_{2,t}^K, \hat{\mathbf{A}}_{3,t}^K]\!]$, so we call it ANSE for rigorousness.

the ULAs is set to half the carrier wavelength λ. The sampling rate is set to $f_{\text{b}} = 0.32\text{GHz}$. The total number of subcarriers is set to $M_0 = 128$, out of which a pilot block is selected randomly with $M = 4$ subcarriers and $m_{\text{s}} \in \{1, \ldots, 125\}$. Suppose that the elements of the noise tensor $\boldsymbol{\mathcal{N}}_t$ are i.i.d. $\mathcal{CN}(0, \sigma^2)$, where σ is determined by the SNR. The AoAs $\vartheta_{r,t}$, AoDs $\theta_{r,t}$, delays $\tau_{r,t}$ in nanoseconds and gains $\beta_{r,t}$ are i.i.d. with distribution $\mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2})$, $\mathcal{U}(-\frac{\pi}{2}, \frac{\pi}{2})$, $\mathcal{U}(0, 100)$ and $\mathcal{CN}(0, 1)$ for $r = 1, \ldots, R$, respectively. We generate 100.000 samples, where 10.000 samples are selected randomly for testing and the rest 90.000 samples are used for training. The number of channel paths is set to $R = 4$. As the generators are distinct, the Vandermonde factors have full Kruskal rank according to [54]. Based on Theorem 1, $R = 4$ satisfies the sufficient condition of uniqueness in this subsection, i.e., $R < 5$. Note that R is known for the proposed method as a priority. One can also determine it as an unknown value by using the information theoretic criteria of minimum description length (MDL) [55]. The fully connected neural network structure has the setting described in Subsection VI-A with 4 hidden layers and 512 neurons per layer, where the input layer dimension is $2I_2I_1M = 2048$, and the output layer dimension is $2R(I_1 + M) = 96$. Therefore, the total number of parameters is 1886304. In order to improve the practicality of the model in actual scenarios, note that the DNN is trained by using the channel samples with mixed SNRs which are selected randomly from $\{5, 10, 15, 20, 25\}$dB and tested for each SNR.

For the proposed DL-CPALS method, a two-stage training scheme is adopted. In the first stage, we set the number of iterations $K = 0$ (see Algorithm 1) and train the DNN for 1000 epochs with learning rate 0.0001, which sets the param-
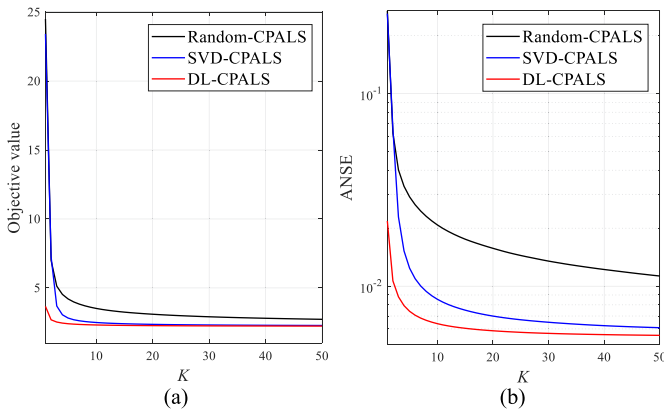
Fig. 6. The performance comparison of CPALS using different initializations on channel data. (a) Iterative behavior of the objective value. (b) Iterative behavior of the ANSE.
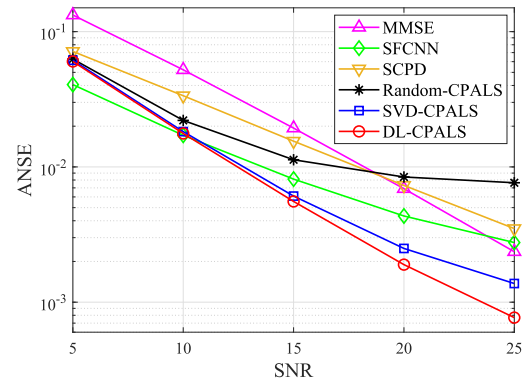

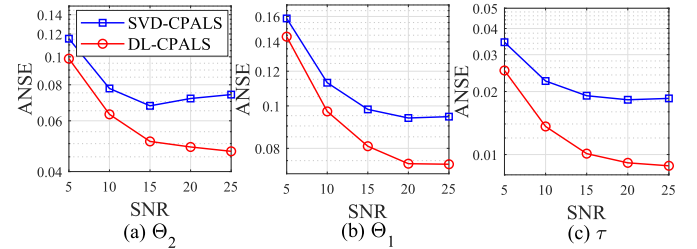Fig. 7. The ANSE of the estimated channels for different SNRs.

Fig. 8. ANSE versus SNR of estimated channel parameters including (a) spatial AoD Θ_2, (b) spatial AoA Θ_1 and (c) delay τ.

eters to a reasonable value range so that the reconstructed CP tensor is close to the input. Then, in the second stage, we set $K = 2$ and the learning rate to 0.00002 for training another 1000 epochs.

As for the competing methods, in addition to Random-CPALS and SVD-CPALS, the supervised deep-learning-based MIMO-OFDM channel estimation method SFCNN [37] is also considered as a baseline, which uses noisy channel samples and noise-free samples to train a convolutional neural network for denoising. The SFCNN is trained for 2000 epochs with learning rate 0.001. Other settings are the same as in the open-source codes of the paper.[4] As listed in Table I, subspace-type methods estimate the signal subspace and channel parameters by exploiting the Vandermonde structure of the factors, which can also reconstruct an estimated channel using the estimated channel parameters. When applied to the channel estimation of MIMO-OFDM systems, a subspace-type method SCPD (Structured CP decomposition) [15] can be obtained, which is also considered as a benchmark. In addition, we also compare the competing methods with the minimum mean-squared error (MMSE) estimator. According to (21), the signal model can be written as $\mathbf{y}_t = \mathbf{X}\mathbf{h}_t + \mathbf{n}_t$, where $\mathbf{X} = \mathbf{I}_M \otimes \text{diag}(\mathbf{x})\mathbf{F}_2^T \otimes \mathbf{F}_1^H$, and $\mathbf{y}_t$, $\mathbf{h}_t$ and $\mathbf{n}_t$ are the vectorization forms of $\mathcal{Y}_t$, $\mathcal{H}_t$ and $\mathcal{N}_t \times_1 \mathbf{F}_1^H$, respectively. Then, an MMSE estimation is written as $\hat{\mathbf{h}}_t = \hat{\mathbf{W}}\mathbf{y}_t$, where $\hat{\mathbf{W}}$ is given as

$$\hat{\mathbf{W}} = \arg\min_{\mathbf{W}} \mathbb{E}\{\|\mathbf{h}_t - \mathbf{W}\mathbf{y}_t\|_2^2\}$$
$$= \mathbb{E}\{\mathbf{h}_t\mathbf{h}_t^H\} \left(\mathbb{E}\{\mathbf{h}_t\mathbf{h}_t^H\} + \mathbb{E}\{\mathbf{n}_t\mathbf{n}_t^H\}\right)^{-1} \mathbf{X}^H. \quad (25)$$

Note that the covariance matrix of the channel is estimated from all training samples and the covariance matrix of the noise is set as true according to the SNR.

We first demonstrate the performance gain of the proposed DL-CPALS for CP decomposition on efficiency and accuracy in Figure 6, where SNR $= 15$dB. Figure 6(a) shows that the proposed DL-CPALS can generate favorable initializations for the CP decomposition that make the objective function decrease rapidly. Note that a better channel estimation per-

[4]https://github.com/phdong21/CNN4CE

formance always corresponds to a lower ANSE but may not to the lower objective value. Although as K increases, SVD-CPALS can achieve an objective function value similar to that of DL-CPALS, the channel estimation results of the proposed DL-CPALS are more accurate as shown in Figure 6(b).

Figure 7 shows the comparison of the channel estimation ANSE between the supervised SFCNN, the subspace-type SCPD and the CPALS algorithm with different initializations under different SNRs with $K = 50$. It can be seen that the performance of SFCNN at SNR $= 5$dB is better than that of the SCPD and the CPALS algorithm, because the DL-CPALS is trained only using noisy data and a low SNR may hurt CPALS for solving a CP low-rank approximation, whereas the SFCNN is trained specially for channel denoising by using the paired clean samples and noisy samples. Nevertheless, the performance of the proposed DL-CPALS becomes optimal as the SNR increases. In addition to the denoising effect, the proposed method can also decompose the high-dimensional channel tensor into low-dimensional factors, which can save the channel storage and further reduce the precoding complexity by using the decomposed factors [56]. Although MMSE describes a nearly optimal linear estimation performance, it cannot achieve the accuracy of the DL-CPALS that exploits the channel low-rankness. Moreover, we provide the channel parameter estimation results of SVD-CPALS and DL-CPALS in Figure 8, where the true parameters and estimated ones are all sorted in descending order due to the ambiguity. Note that the SFCNN cannot perform parameter estimation and the results of random-CPALS are poor, which is therefore not provided in the figure. It can be seen that the parameter estimation error of the proposed DL-CPALS is better than SVD-CPALS in all cases.

TABLE II
THE NUMBER OF ITERATIONS OF DIFFERENT CPALSS FOR ACHIEVING THE SAME ANSE OF CHANNEL ESTIMATION

ANSE	Random-CPALS	SVD-CPALS	DL-CPALS
0.01	36	5	2
0.004	259	15	5
0.002	Fail	158	36

TABLE III
COMPLEXITY COMPARISON OF DIFFERENT INITIALIZATION METHODS OF CPALS

Method	Time Complexity $\mathcal{O}$	Kilo Cflops	
		Init.	Per Iter.
Random	KI_1I_2MR $+K(I_1I_2 + I_1M + I_2M)R^2$	-	40
SVD	$I_1I_2M(I_1 + M) + KI_1I_2MR$ $+K(I_1I_2 + I_1M + I_2M)R^2$	~ 16	40
DL	$DI_1I_2M + QD^2 + KI_1I_2MR$ $+K(I_1I_2 + I_1M + I_2M)R^2$	314	40

C. Complexity Analysis

In this subsection, we discuss the computational complexity of the proposed DL-CPALS in the testing phase. The experiments are all implemented through Python and PyTorch programming on a single computer, which is equipped with an Intel(R) Core(TM) i7-9700K 3.60GHz CPU and 32G memory. During the testing phase, all methods are evaluated on the CPU. We consider SNR = 20dB and focus on the channel estimation experiments of Subsection VI-B. We set the maximum number of iterations to $K = 500$.

We report the number of iterations required for CPALS to achieve a certain channel estimation accuracy using different initialization approaches in Table II, where we can see that DL-CPALS greatly saves iterations to guarantee a certain accuracy. For example for the ANSE = 0.002, the proposed DL-CPALS only takes 36 iterations, while SVD-CPALS takes 158 iterations and Random-CPALS fails to achieve the prescribed accuracy even after 500 iterations.

Compared with Random-CPALS, both SVD-CPALS and DL-CPALS require some complexity for the initializations. In Table III, we first show the computational complexity of these methods. The initializations of SVD-CPALS and DL-CPALS lead to a complexity of $I_1I_2M(I_1 + M)$ and $DI_1I_2M + QD^2$, respectively, where I_1, I_2, M, Q and D are the numbers of receiving antennas, transmitting antennas, training subcarriers, nodes of each hidden layer and hidden layers, respectively. Although a simple fully connected network is employed to demonstrate the effectiveness of our method, the DNN structure could be more dedicated such as a convolutional network so that its complexity is comparable with the SVD. Moreover, the number of floating point operations (flops) is also provided in Table III. A flop serves as a basic unit of real number computation, which could denote one addition, subtraction, multiplication or division of real

floating point numbers. Denote that a complex flop (cflop) serves as a basic unit of complex number computation. Due to that a complex multiplication requires 6 real flops, a cflop is more expensive than a flop. Suppose that $\mathbf{A} \in \mathbb{C}^{m \times n}$, $\mathbf{B} \in \mathbb{C}^{k \times n}$, $\mathbf{C} \in \mathbb{C}^{n \times l}$ and positive definite $\mathbf{R} \in \mathbb{C}^{n \times n}$. According to [54], one can obtain that the numbers of cflops of $\mathbf{A}^H\mathbf{A}$, $\mathbf{A}\mathbf{C}$, $\mathbf{A}\diamond\mathbf{B}$, $\mathbf{A}\odot\mathbf{B}$ and $\mathbf{R}^{-1}$ are $mn^2 + mn - \frac{n^2}{2} - \frac{n}{2}$, $2mnl - ml$, mkn, mn and, $n^3 + n^2 + n$, respectively, where the matrix inverse is calculated based on the Cholesky decomposition. As for the SVD, the popular bidiagonalization algorithm produces $\sim \frac{8}{3}n^3$ cflops [57], where $\sim$ means that only the leading term is considered. Thus, we can obtain that each iteration of CPALS takes 40 kilo cflops. The SVD initialization takes 16 kilo cflops at least, which is roughly equivalent to one iteration. According to the flops counter THOP[5] of the DNN model for PyTorch, the DNN model of the proposed DL-CPALS requires 1884160 flops, which is about 314 kilo cflops. The storage of the DNN is 7.2 megabytes. Although the DL initialization consumes about 8 iterations of cflops, it saves more than 100 iterations of cflops at ANSE of 0.002 in Table II. In addition, the average initialization time of a single sample is about 0.6 milliseconds. Compared with the running time it took for the ALS iterations, e.g., about 22 milliseconds with $K = 50$, the time required for computing the initializations is negligible.

VII. CONCLUSION

This paper proposes a deep-learning-aided CP alternating least squares (DL-CPALS) method for efficient tensor CANDECOMP/PARAFAC (CP) decomposition. Specifically, by integrating the model-driven CP decomposition into the data-driven learning process, a neural network model is trained to generate favorable initializations for a fast and accurate CP decomposition. Due to the CP low-rankness constraint exploited in the model, the proposed DL-CPALS only requires noisy samples and does not require paired noise-free samples. The method is applied to the CP low-rank approximation using synthetic data and tensor channel estimation for millimeter wave multiple-input multiple-output orthogonal frequency division multiplexing (mmWave MIMO-OFDM) systems. Experimental results show that the proposed method achieves a fast and accurate CP decomposition compared with the baseline methods.

REFERENCES

[1] A. L. F. de Almeida, G. Favier, J. da Costa, and J. C. M. Mota, "Overview of tensor decompositions with applications to communications," in *Signals and Images: Advances and Results in Speech, Estimation, Compression, Recognition, Filtering, and Processing*, 2016, pp. 325–356.

[2] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *J. Math. Phys.*, vol. 6, nos. 1–4, pp. 164–189, Apr. 1927.

[3] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev. Soc. Ind. Appl. Math.*, vol. 51, no. 3, pp. 455–500, Aug. 2009.

[4] H. Chen, F. Ahmad, S. Vorobyov, and F. Porikli, "Tensor decompositions in wireless communications and MIMO radar," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 3, pp. 438–453, Apr. 2021.

[5] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[5] https://github.com/Lyken17/pytorch-OpCounter

[6] I. Barhumi, G. Leus, and M. Moonen, "Optimal training design for MIMO OFDM systems in mobile wireless channels," *IEEE Trans. Signal Process.*, vol. 51, no. 6, pp. 1615–1624, Jun. 2003.

[7] H. Xie, F. Gao, and S. Jin, "An overview of low-rank channel estimation for massive MIMO systems," *IEEE Access*, vol. 4, pp. 7313–7321, 2016.

[8] Z. Zhou, J. Fang, L. Yang, H. Li, Z. Chen, and R. S. Blum, "Low-rank tensor decomposition-aided channel estimation for millimeter wave MIMO-OFDM systems," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 7, pp. 1524–1538, Jul. 2017.

[9] R. Zhang, L. Cheng, S. Wang, Y. Lou, W. Wu, and D. W. K. Ng, "Tensor decomposition-based channel estimation for hybrid mmWave massive MIMO in high-mobility scenarios," *IEEE Trans. Commun.*, vol. 70, no. 9, pp. 6325–6340, Sep. 2022.

[10] H. L. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*. Hoboken, NJ, USA: Wiley, 2002.

[11] M. Sørensen and L. De Lathauwer, "Blind signal separation via tensor decomposition with Vandermonde factor: Canonical polyadic decomposition," *IEEE Trans. Signal Process.*, vol. 61, no. 22, pp. 5507–5519, Nov. 2013.

[12] R. Roy and T. Kailath, "Esprit-estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 7, pp. 984–995, Jul. 1989.

[13] J. Li, Z. Wu, Z. Wan, P. Zhu, D. Wang, and X. You, "Structured tensor CP decomposition-aided pilot decontamination for UAV communication in cell-free massive MIMO systems," *IEEE Commun. Lett.*, vol. 26, no. 9, pp. 2156–2160, Sep. 2022.

[14] J. Wang et al., "Time-varying channel estimation scheme for uplink MU-MIMO in 6G systems," *IEEE Trans. Veh. Technol.*, vol. 71, no. 11, pp. 11820–11831, Nov. 2022.

[15] Y. Lin, S. Jin, M. Matthaiou, and X. You, "Tensor-based channel estimation for millimeter wave MIMO-OFDM with dual-wideband effects," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4218–4232, Jul. 2020.

[16] Y. Lin, S. Jin, M. Matthaiou, and X. You, "Channel estimation and user localization for IRS-assisted MIMO-OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 21, no. 4, pp. 2320–2335, Apr. 2022.

[17] R. Zhang et al., "Integrated sensing and communication with massive MIMO: A unified tensor approach for channel and target parameter estimation," *IEEE Trans. Wireless Commun.*, vol. 23, no. 8, pp. 8571–8587, Aug. 2024.

[18] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3551–3582, Jul. 2017.

[19] L. Sorber, M. Van Barel, and L. De Lathauwer, "Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-$(L_r, L_r, 1)$ terms, and a new generalization," *SIAM J. Optim.*, vol. 23, no. 2, pp. 695–720, Jan. 2013.

[20] F. Wen, J. Shi, and Z. Zhang, "Joint 2D-DOD, 2D-DOA, and polarization angles estimation for bistatic EMVS-MIMO radar via PARAFAC analysis," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1626–1638, Feb. 2020.

[21] C. Qian, X. Fu, N. D. Sidiropoulos, and Y. Yang, "Tensor-based channel estimation for dual-polarized massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 66, no. 24, pp. 6390–6403, Dec. 2018.

[22] Z. Zhou, J. Fang, L. Yang, H. Li, Z. Chen, and S. Li, "Channel estimation for millimeter-wave multiuser MIMO systems via PARAFAC decomposition," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7501–7516, Nov. 2016.

[23] J. Du, M. Han, Y. Chen, L. Jin, and F. Gao, "Tensor-based joint channel estimation and symbol detection for time-varying mmWave massive MIMO systems," *IEEE Trans. Signal Process.*, vol. 69, pp. 6251–6266, 2021.

[24] L. Cheng, G. Yue, X. Xiong, Y. Liang, and S. Li, "Tensor decomposition-aided time-varying channel estimation for millimeter wave MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1216–1219, Aug. 2019.

[25] G. T. de Araújo, A. L. F. de Almeida, and R. Boyer, "Channel estimation for intelligent reflecting surface assisted MIMO systems: A tensor modeling approach," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 3, pp. 789–802, Apr. 2021.

[26] S. Park, A. Ali, N. González-Prelcic, and R. W. Heath, "Spatial channel covariance estimation for hybrid architectures based on tensor decompositions," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 1084–1097, Feb. 2020.

[27] J. Du, Y. Cheng, L. Jin, S. Li, and F. Gao, "Nested tensor-based integrated sensing and communication in RIS-assisted THz MIMO systems," *IEEE Trans. Signal Process.*, vol. 72, pp. 1141–1157, 2024.

[28] A. K. Smilde, P. Geladi, and R. Bro, *Multi-Way Analysis: Applications in the Chemical Sciences*. Hoboken, NJ, USA: Wiley, 2005.

[29] E. Sanchez and B. R. Kowalski, "Tensorial resolution: A direct trilinear decomposition," *J. Chemometrics*, vol. 4, no. 1, pp. 29–45, Jan. 1990.

[30] R. Bro, "PARAFAC. Tutorial and applications," *Chemometr. Intell. Lab. Syst.*, vol. 38, no. 2, pp. 149–171, 1997.

[31] P. R. B. Gomes, G. T. de Araújo, B. Sokal, A. L. F. de Almeida, B. Makki, and G. Fodor, "Channel estimation in RIS-assisted MIMO systems operating under imperfections," *IEEE Trans. Veh. Technol.*, vol. 72, no. 11, pp. 14200–14213, Nov. 2023.

[32] R. Zhang et al., "Channel training-aided target sensing for terahertz integrated sensing and massive MIMO communications," *IEEE Internet Things J.*, early access, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10643882

[33] R. Zhang et al., "Channel estimation for movable-antenna MIMO systems via tensor decomposition," *IEEE Wireless Commun. Lett.*, vol. 13, no. 11, pp. 3089–3093, Nov. 2024.

[34] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.

[35] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Commun. Lett.*, vol. 23, no. 4, pp. 652–655, Apr. 2019.

[36] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and DOA estimation based massive MIMO system," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8549–8560, Sep. 2018.

[37] P. Dong, H. Zhang, G. Y. Li, I. S. Gaspar, and N. NaderiAlizadeh, "Deep CNN-based channel estimation for mmWave massive MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 5, pp. 989–1000, Sep. 2019.

[38] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, Mar. 2021.

[39] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-based deep learning," *Proc. IEEE*, vol. 111, no. 5, pp. 465–499, May 2023.

[40] M. Borgerding, P. Schniter, and S. Rangan, "AMP-inspired deep networks for sparse linear inverse problems," *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4293–4308, Aug. 2017.

[41] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based channel estimation for beamspace mmWave massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 852–855, Oct. 2018.

[42] K. Pratik, B. D. Rao, and M. Welling, "RE-MIMO: Recurrent and permutation equivariant neural MIMO detection," *IEEE Trans. Signal Process.*, vol. 69, pp. 459–473, 2021.

[43] J. Zhang, Z. Tao, L. Zhang, and Q. Zhao, "Tensor decomposition via core tensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 2130–2134.

[44] D. Hong, T. G. Kolda, and J. A. Duersch, "Generalized canonical polyadic tensor decomposition," *SIAM Rev.*, vol. 62, no. 1, pp. 133–163, Jan. 2020.

[45] N. D. Sidiropoulos and R. Bro, "On the uniqueness of multilinear decomposition of N-way arrays," *J. Chemometr.*, vol. 14, no. 3, pp. 229–239, May 2000.

[46] B. C. Mitchell and D. S. Burdick, "Slowly converging PARAFAC sequences: Swamps and two-factor degeneracies," *J. Chemometrics*, vol. 8, no. 2, pp. 155–168, Mar. 1994.

[47] A. Uschmajew, "Local convergence of the alternating least squares algorithm for canonical tensor approximation," *SIAM J. Matrix Anal. Appl.*, vol. 33, no. 2, pp. 639–652, 2012.

[48] L. Wang and M. T. Chu, "On the global convergence of the alternating least squares method for rank-one approximation to generic tensors," *SIAM J. Matrix Anal. Appl.*, vol. 35, no. 3, pp. 1058–1072, Jan. 2014.

[49] N. Vervliet, O. Debals, and L. De Lathauwer, "Tensorlab 3.0—Numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization," in *Proc. 50th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2016, pp. 1733–1738.

[50] A. Hjorungnes and D. Gesbert, "Complex-valued matrix differentiation: Techniques and key results," *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 2740–2746, Jun. 2007.

[51] A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Dec. 2019.

[52] J. Rodríguez-Fernández, N. González-Prelcic, K. Venugopal, and R. W. Heath, "Frequency-domain compressive channel estimation for frequency-selective hybrid millimeter wave MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 2946–2960, May 2018.

[53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2015.

[54] T. Jiang, N. D. Sidiropoulos, and J. M. F. ten Berge, "Almost-sure identifiability of multidimensional harmonic retrieval," *IEEE Trans. Signal Process.*, vol. 49, no. 9, pp. 1849–1859, Sep. 2001.

[55] M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-33, no. 2, pp. 387–392, Apr. 1985.

[56] L. N. Ribeiro, S. Schwarz, A. L. F. de Almeida, and M. Haardt, "Low-complexity massive MIMO tensor precoding," in *Proc. 54th Asilomar Conf. Signals, Syst., Comput.*, Nov. 2020, pp. 348–355.

[57] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*. Philadelphia, PA, USA: SIAM, 2022.

Bo Ai (Fellow, IEEE) received the M.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 2002 and 2004, respectively. He was with Tsinghua University, Beijing, China, where he was an Excellent Post-Doctoral Research Fellow in 2007. He is currently a Full Professor with Beijing Jiaotong University, where he is also the Deputy Director of the State Key Laboratory of Advanced Rail Autonomous Operation and the Deputy Director of the International Joint Research Center. He has authored or co-authored six books and 270 scientific research articles and holds 26 invention patents in his research areas. His research interests include the research and applications of orthogonal frequency-division multiplexing techniques, high-power amplifier linearization techniques, radio propagation and channel modeling, global systems for mobile communications for railway systems, and long-term evolution for railway systems.

Xiao Gong received the B.S. and Ph.D. degrees from Beijing Jiaotong University, Beijing, China, in 2018 and 2023, respectively. His current research interests include tensor-based signal processing and machine learning.

Wei Chen (Senior Member, IEEE) received the B.Eng. and M.Eng. degrees from Beijing University of Posts and Telecommunications, China, in 2006 and 2009, respectively, and the Ph.D. degree in computer science from the University of Cambridge, U.K., in 2013. Later, he was a Research Associate with the Computer Laboratory, University of Cambridge, from 2013 to 2016. He is currently a Professor with Beijing Jiaotong University, Beijing, China. His current research interests include sparse representation, Bayesian inference, wireless communication systems, and image processing. He was a recipient of the 2013 IET Wireless Sensor Systems Premium Award and the 2017 International Conference on Computer Vision (ICCV) Young Researcher Award.

Geert Leus (Fellow, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering from KU Leuven, Leuven, Belgium, in 1996 and May 2000, respectively. He is currently a Full Professor with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands. He is a fellow of EURASIP. He was a Member-at-Large of the Board of Governors of the IEEE Signal Processing Society, the Chair of the IEEE Signal Processing for Communications and Networking Technical Committee, the Chair of the EURASIP Technical Area Committee on Signal Processing for Multisensor Systems, the Editor-in-Chief of the *EURASIP Journal on Advances in Signal Processing*, and the Editor-in-Chief of *EURASIP Journal on Advances in Signal Processing*. He was a recipient of the 2021 EURASIP Individual Technical Achievement Award, 2005 IEEE Signal Processing Society Best Paper Award, and 2002 IEEE Signal Processing Society Young Author Best Paper Award.