# Multilinear feature extraction with SVD-based tensor wheel decomposition

Rafał Zdunek[1]

***Abstract*— Multiway data analysis is a powerful technique for extracting meaningful and compact information from multi-dimensional data or signals, which frequently arise in various information technology applications. The tensor wheel (TW) decomposition model has a generalized and flexible tensor network structure that unifies many fundamental multilinear models, offering significant potential for extracting relevant features from multiway data. However, the standard alternating least squares (ALS)-based algorithm used for this model incurs high computational costs, especially when processing high-order tensors. To address this limitation, a new algorithm for estimating latent factors in the TW decomposition model is proposed in this study, specifically designed for classification tasks. Instead of relying on the conventional alternating optimization framework, our approach employs a recursive SVD-based update rule, which substantially reduces the computational cost of the original TW model. Numerical experiments, performed on several real-world datasets, demonstrate that the proposed method is not only significantly more efficient than the baseline TW algorithm but also yields more informative multilinear features for classification tasks compared to competing methods.**

***Index Terms*— Feature extraction, Tensor wheel decomposition, Classification, Recursive SVD**

## I. INTRODUCTION

In many areas of information technologies, the observed data or signals can be represented by multiway arrays, often called tensors. Examples of such data are RGB images, hyperspectral data cubes [1], [2], video sequences [3], [4], or a set of spectrograms in multichannel data processing systems. To extract the meaningful and low-rank features from such data, various tensor decompositions methods can be used. The CP and Tucker decompositions are probably the most popular techniques used for low-rank data representations in a variety of real-world applications, including signal/image processing [5], [6], computer vision [7], [8], data mining [9], biomedical data analysis, etc.

The Tucker decomposition model, whose topological diagram is illustrated in Fig. 1a, decomposes the $N$-th order multiway array into a set of $N$ factor matrices, each representing a given mode of the input tensor, and the $N$-th core tensor that contains cross-interactions between the modes. It has been considered as a generalized framework for multiway data decomposition for several decades. However, its disadvantage is an exponential grow in a storage complexity – the problem known as the curse of dimensionality, and a lack of interactions between the successive modes. These issues were alleviated in the tensor ring (TR) model [10] which well captures neighboring mode interactions and has the circular permutation invariance property. It also has a flexibly topology because the tensor train (TT) model [11] is its particular case. The diagram of its topology is depicted in Fig. 1b. However, it has a limited functionality in classification tasks because its ring tensor representing the mode of sample ordering is weakly affected by non-neighboring modes, i.e. there is no direct all modes interactions.

The tensor wheel (TW) decomposition model, whose topology is presented in Fig. 1c, has been recently proposed by Wu *et al.* [12]. It has a generalized and flexible structure and seems to combine the advantages of many tensor decomposition models, such as the CP, Tucker, TT, and TR. It has two levels of latent factors: the ring tensors that play a similar role as in the TR model, and the core tensor that has a similar functionality as in the basic Tucker model. In consequence, there are two sets of ranks: the outer ranks (denoted by set $\{R_1, \ldots, R_N\}$) which refer to the connections between the ring tensors, and the inner ranks that determine the links between the ring factors and the core tensor (given by set $\{L_1, \ldots, L_N\}$). More ranks give us higher flexibility in adaptation the model to data, however, it is more challenging to optimally determine all these parameters. It is easy to notice that if $R_1 = R_2 = \ldots = R_N = 1$, then the TW boils down to the standard Tucker model, as shown in Fig. 1a. When $L_1 = L_2 = \ldots = L_N = 1$, the TR model is obtained (see Fig. 1b). Additionally, if $L_1 = L_2 = \ldots = L_N = 1$ and $\exists n : R_n = 1$, then the TW model transforms to the TT model. Thus, the TW decomposition model has a generalized and flexibly structure, and can be easily adapted to a given data, which motivated us to study this model. However, due to the complex structure, its main disadvantage persists in the computational issue.

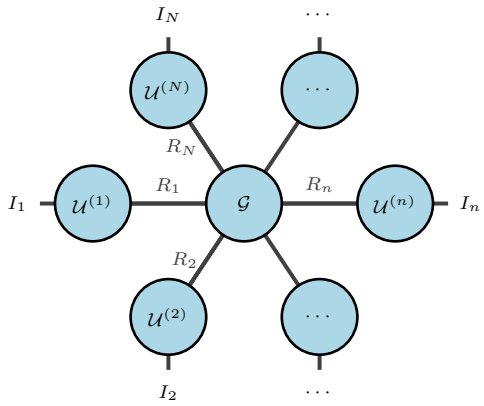
To estimate the ring tensors in the TW model, Wu *et al.* [12] used the generalized tensor $k$-unfolding and contraction operators to express this model in the form of the tensor subwheel equations, where $N$ equations of the $(N+2)$-th order are used for updating the $N$ ring tensors using the alternating least square (ALS) rule, and one $2N$-th order equation is used to estimate the core tensor. As a result, the computational complexity for updating all the latent factors in the TW model can be upper bounded by:

$$\mathcal{O}\left(k\sum_{l=2}^{N-1} I^l R^{l+3} + kNI^N R^3 + I^N R^{2N}\right), \tag{1}$$

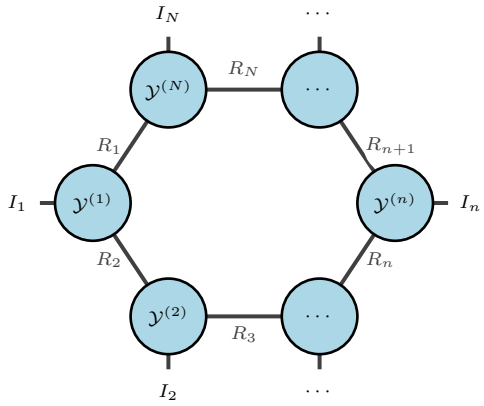
where $k$ is the number of iterative cycles.

The first two terms in (1) are related to the computation

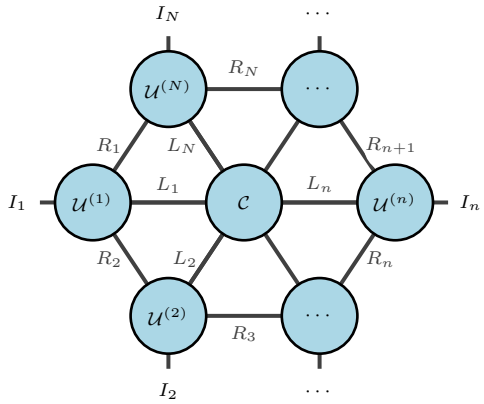
[1]Rafał Zdunek is with Faculty of Electronics, Photonics, and Microsystems, Wroclaw University of Science and Technology, Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland rafal.zdunek@pwr.edu.pl




(a) TD model




(b) TR model




(c) TW model

Fig. 1: Graphical representations of tensor decomposition models by using a tensor network diagram

of the ring tensors, while the last one corresponds to the core tensor. Assuming the typical scenario, i.e. $I >> R$, $R > 1$, and $N > 3$, we can conclude that the third term in (1) predominates over the others, which means that the computation of the core tensor is the most computationally challenging. Obviously, the computation of the ring tensors is also expensive, especially if many iterations need to be performed.

Thus, the overall computational cost is prohibitively large for decomposing high-order tensors. To decrease it considerably, a new computational algorithm is proposed in this study. The algorithm aims to provide orthogonal multilinear features, which are particularly desired in various classification problems, and has significant potential for processing much larger datasets than the conventional ALS-based approach. It is based on the recursive updating scheme with the singular value decomposition (SVD) for feature extraction, which is motivated by the recursive update rules for the standard and hierarchical Tucker decomposition [13]. The proposed algorithm uses the similar updating scheme as the TW-SVD method that has been recently proposed by Wang [14]. However, our approach is algorithmically different and addressed directly for extracting multilinear features for classification problems. The fundamental difference is that the core tensor and the ring tensor corresponding to the mode of ordering samples are represented by one $(N+2)$-th order extended core that contains training samples. The proposed algorithm was applied to classification problems with four datasets from different areas of computer science, including face images, object images with a wide variety of complex geometric and reflectance characteristics, textual documents, and spectrograms of musical instruments. All the tests showed the superior performance of the proposed method with respect to the other competitive methods.

The remainder of this paper is organized as follows: the preliminaries and the basic TW model are briefly introduced in Section II. Section III describes the proposed method. Experimental results are presented in Section IV. Finally, the conclusions are presented in Section V.

## II. PRELIMINARIES

### A. Notations

Some notations on multiway arrays and fundamental operations on tensors are taken from [15]. Calligraphic uppercase letters (e.g., $\mathcal{X}$), boldface uppercase letters (e.g., $\boldsymbol{X}$), lowercase boldface letters (e.g., $\boldsymbol{z}$) and unbolded letters (e.g., $x$) denote multiway arrays, matrices, vectors, and scalars, respectively. A multiway array will be called a tensor, and its order means the number of its modes or directions. The $(i_1, \ldots, i_N)$-th element of $N$-th order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ is denoted by $x_{i_1,\ldots,i_N}$. The symbol $\otimes$ stands for the Kronecker product.

**Mode-$n$ unfolding**: Let $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ be $N$-way array. Mode-$n$ unfolding of $\mathcal{X}$, also known as the mode-$n$ matricization, will be denoted by $\boldsymbol{X}_{(n)}$ and this operation reshapes $\mathcal{X}$ into a matrix $\boldsymbol{X}_{(n)} = [x_{i_n,j}] \in \mathbb{R}^{I_n \times \prod_{p \neq n} I_p}$ by mapping any tensor element $x_{i_1,\ldots,i_N}$ to matrix element $x_{i_n,j}$, where $j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^{N} (i_k - 1) j_k$ with $j_k = \prod_{\substack{m=1 \\ m \neq n}}^{k-1} I_m$.

**Generalized $k$-unfolding** reshapes tensor $\mathcal{X}$ into matrix $\boldsymbol{X}_{<n>} = [x_{i,j}] \in \mathbb{R}^{\prod_{p=1}^{n} I_p \times \prod_{r=n+1}^{N} I_r}$ by mapping tensor element $x_{i_1,\ldots,i_N}$ to matrix element $x_{i,j}$, where $i = 1 + \sum_{p=1}^{n} (i_p - 1) \prod_{m=1}^{p-1} I_m$ and $j = 1 + \sum_{r>n}^{N} (i_r - 1) \prod_{m=n+1}^{r-1} I_m$.

### B. Basic mathematical operations on tensors

**Tensor contraction** is a fundamental multiplication operation that links two tensors $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ and $\mathcal{Y} \in \mathbb{R}^{J_1 \times \ldots \times J_M}$ along their selected modes into one tensor $\mathcal{Z}$. Assuming the $n$-th mode of $\mathcal{X}$ is contracted with the $m$-th mode of $\mathcal{Y}$, where $I_n = J_m$, we have: $\mathcal{Z} = \mathcal{X} \times_n^m \mathcal{Y} \in \mathbb{R}^{I_1 \times \ldots \times I_{n-1} \times I_{n+1} \times \ldots \times I_N \times J_1 \times \ldots \times J_{m-1} \times J_{m+1} \times \ldots \times J_M}$.

**Generalized $k$-contraction** operator performs the tensor contraction simultaneously on $L$ modes, where $L \geq 1$, and $I_{n_l} = J_{m_l}$ for $l = 1, \ldots, L$. Assuming $n_{L+1} < \ldots < n_N$ and $m_{L+1} < \ldots < m_M$, the generalized $k$-contraction between $\mathcal{X}$ and $\mathcal{Y}$ results in the $(M+N-2L)$-th order tensor $\mathcal{Z}$, defined as: $\mathcal{Z} = \mathcal{X} \times_{n_1,\ldots,n_L}^{m_1,\ldots,m_L} \mathcal{Y} \in \mathbb{R}^{I_{n_{L+1}} \times \ldots I_{n_N} \times J_{m_{L+1}} \times \ldots \times J_{m_M}}$.

**Circular $k$-mode shift** is the operation that rearrange the modes of $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ by moving the first $k$ modes to the $k$ final modes. Thus $\overleftarrow{\mathcal{X}}_{[k]} = \texttt{circ}(\mathcal{X}, k) \in \mathbb{R}^{I_{k+1} \times \ldots \times I_N \times I_1 \times \ldots \times I_k}$.

### C. Model

The TW decomposition [12] of $\mathcal{X} = [x_{i_1,\ldots,i_N}] \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ can be defined as follows:

$$\begin{aligned} x_{i_1,\ldots,i_N} &= \sum_{r_1=1}^{R_1} \cdots \sum_{r_N=1}^{R_N} \sum_{l_1=1}^{L_1} \cdots \sum_{l_N=1}^{L_N} g^{(1)}_{r_1,i_1,l_1,r_2} \cdots \\ &\cdots \; g^{(N)}_{r_N,i_N,l_N,r_1} c_{i_1,\ldots,i_N}, \end{aligned} \tag{2}$$

where $R_1 = R_{N+1}$, $\forall k : \mathcal{G}^{(k)} = \left[ g^{(k)}_{r_k,i_k,l_k,r_{k+1}} \right] \in \mathbb{R}^{R_k \times I_k \times L_k \times R_{k+1}}$ is the 4-th order ring tensor that captures the multilinear features across the $k$-th mode of $\mathcal{X}$, and $\mathcal{C} = [c_{i_1,\ldots,i_N}] \in \mathbb{R}^{L_1 \times \ldots \times L_N}$ is the $N$-th order core tensor that contains the coefficients of cross-modal interactions. Equivalently, model (2) can be expressed in the following form:

$$\begin{aligned} \mathcal{X} &= \mathcal{G}^{(1)} \times_4^1 \mathcal{G}^{(2)} \times_6^1 \cdots \times_{2n}^1 \mathcal{G}^{(n)} \times_{2n+2}^1 \cdots \\ &\quad \times_{2N,1}^{1,4} \mathcal{G}^{(N)} \times_{2,4,\ldots,2N}^{1,2,\ldots,N} \mathcal{C}. \end{aligned} \tag{3}$$

## III. PROPOSED ALGORITHM

Let $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ be the $N$-th order tensor containing $I_N$ multiway training samples ordered along the $N$-th mode. Without loss of generality, let us assume the mode of ordering the training samples is the last one. If this assumption is not satisfied, a circular permutation should be first applied to permute the input tensor accordingly.

Note that model (3) can be rewritten as:

$$\mathcal{X} = \mathcal{G}^{(1)} \times_{2,3,4}^{1,2,3} \mathcal{W}^{(1)}, \tag{4}$$

where $\mathcal{G}^{(1)} \in \mathbb{R}^{R_1 \times I_1 \times L_1 \times R_2}$ is the ring tensor corresponding to the first mode of $\mathcal{X}$, and $\mathcal{W}^{(1)} \in \mathbb{R}^{R_1 \times L_1 \times R_2 \times I_2 \times \ldots \times I_N}$ is the $(N+2)$-th order tensor built from the core tensor and all the ring tensors except for $\mathcal{G}^{(1)}$, contracted accordingly. Applying the mode-1 unfolding to (4), we get:

$$\boldsymbol{X}_{(1)} = \boldsymbol{G}^{(1)}_{(2)} \boldsymbol{W}^{(1)}_{<3>}, \tag{5}$$

where $\boldsymbol{G}^{(1)}_{(2)} \in \mathbb{R}^{I_1 \times R_1 L_1 R_2}$ is $\mathcal{G}^{(1)}$ unfolded across the mode-2, and $\boldsymbol{W}^{(1)}_{<3>} \in \mathbb{R}^{R_1 L_1 R_2 \times I_2 \cdots I_N}$ is obtained from $\mathcal{W}^{(1)}$ by using the generalized $k$-unfolding with $k = 3$.

Model (5) can be regarded as a matrix factorization of $\boldsymbol{X}_{(1)}$ with rank $J_1 = R_1 L_1 R_2$. Thus, factors $\boldsymbol{G}^{(1)}_{(2)}$ and $\boldsymbol{W}^{(1)}_{<3>}$ can be obtained using various factorization techniques. In this study, truncated SVD is used, i.e. $\boldsymbol{X}_{(1)} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$ with the target rank $J_1$, assuming $\boldsymbol{G}^{(1)}_{(2)} = \boldsymbol{U}$ and $\boldsymbol{W}^{(1)}_{<3>} = \boldsymbol{S}\boldsymbol{V}^T$. Applying the inverse mode-2 unfolding operation to $\boldsymbol{G}^{(1)}_{(2)}$, we get $\mathcal{G}^{(1)}$. Similarly, $\mathcal{W}^{(1)}$ is obtained from $\boldsymbol{W}^{(1)}_{<3>}$ using the inverse generalized 3-unfolding.

Let $\mathcal{Q}^{(1)} = \texttt{circ}\left(\mathcal{W}^{(1)}, 2\right) \in \mathbb{R}^{R_2 \times I_2 \times \ldots \times I_N \times R_1 \times L_1}$. Note that

$$\mathcal{Q}^{(1)} = \mathcal{G}^{(2)} \times_{3,4}^{1,2} \mathcal{W}^{(2)}, \tag{6}$$

where $\mathcal{G}^{(2)} \in \mathbb{R}^{R_2 \times I_2 \times L_2 \times R_3}$ is the second ring tensor, and $\mathcal{W}^{(2)} \in \mathbb{R}^{L_2 \times R_3 \times I_3 \times \ldots \times I_N \times R_1 \times L_1}$ is the $(N+2)$-th order tensor created from all the latent factors except for $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$. Applying the generalized 2-unfolding to $\mathcal{Q}^{(1)}$, we obtain:

$$\boldsymbol{Q}^{(1)}_{<2>} = \boldsymbol{G}^{(2)}_{<2>} \boldsymbol{W}^{(2)}_{<2>}, \tag{7}$$

where $\boldsymbol{Q}^{(1)}_{<2>} \in \mathbb{R}^{R_2 I_2 \times I_3 \cdots I_N R_1 L_1}$, $\boldsymbol{G}^{(2)}_{<2>} \in \mathbb{R}^{R_2 I_2 \times L_2 R_3}$, and $\boldsymbol{W}^{(2)}_{<2>} \in \mathbb{R}^{L_2 R_3 \times I_3 \cdots I_N R_1 L_1}$. Similarly as above, factors $\boldsymbol{G}^{(2)}_{<2>}$ and $\boldsymbol{W}^{(2)}_{<2>}$ can also be obtained with truncated SVD applied to $\boldsymbol{Q}^{(1)}_{<2>}$ with target rank $J_2 = L_2 R_3$, and then the inverse unfolding operations can be used to get $\mathcal{G}^{(2)}$ and $\mathcal{W}^{(2)}$.

Then, let $\mathcal{Q}^{(2)} = \texttt{circ}\left(\mathcal{W}^{(2)}, 1\right) \in \mathbb{R}^{R_3 \times I_3 \times \ldots \times I_N \times R_1 \times L_1 \times L_2}$, and we have:

$$\mathcal{Q}^{(2)} = \mathcal{G}^{(3)} \times_{3,4}^{1,2} \mathcal{W}^{(3)}, \tag{8}$$

where $\mathcal{G}^{(3)} \in \mathbb{R}^{R_3 \times I_3 \times L_3 \times R_4}$ and $\mathcal{W}^{(3)} \in \mathbb{R}^{L_3 \times R_4 \times I_4 \times \ldots \times I_N \times R_1 \times L_1 \times L_2}$.

Following this nested decomposition scheme, the general recursive rule can be obtained:

$$\mathcal{Q}^{(k-1)} = \mathcal{G}^{(k)} \times_{3,4}^{1,2} \mathcal{W}^{(k)}, \tag{9}$$

where $\mathcal{Q}^{(k-1)} = \texttt{circ}\left(\mathcal{W}^{(k-1)}, 1\right) \in \mathbb{R}^{R_k \times I_k \times \ldots \times I_N \times R_1 \times L_1 \times \ldots \times L_{k-1}}$, $\mathcal{G}^{(k)} \in \mathbb{R}^{R_k \times I_k \times L_k \times R_{k+1}}$ and $\mathcal{W}^{(k)} \in \mathbb{R}^{L_k \times R_{k+1} \times I_{k+1} \times \ldots \times I_N \times R_1 \times L_1 \times \ldots \times L_{k-1}}$ for $k = 2, \ldots, N-1$. Applying the generalized 2-unfolding to (9), the generalized factorization rule is obtained:

$$\boldsymbol{Q}^{(k-1)}_{<2>} = \boldsymbol{G}^{(k)}_{<2>} \boldsymbol{W}^{(k)}_{<2>}, \tag{10}$$

where $\boldsymbol{Q}^{(k-1)}_{<2>} \in \mathbb{R}^{R_k I_k \times I_{k+1} \cdots I_N R_1 L_1 \cdots L_{k-1}}$, $\boldsymbol{G}^{(k)}_{<2>} \in \mathbb{R}^{R_k I_k \times L_k R_{k+1}}$, and $\boldsymbol{W}^{(k)}_{<2>} \in \mathbb{R}^{L_k R_{k+1} \times I_{k+1} \cdots I_N R_1 L_1 \cdots L_{k-1}}$. All the factors $\boldsymbol{G}^{(k)}_{<2>}$ and $\boldsymbol{W}^{(k)}_{<2>}$ can be obtained with truncated SVD applied to $\boldsymbol{Q}^{(k-1)}_{<2>}$ with target rank $J_k = L_k R_{k+1}$.

For $k = N-1$, there is $\mathcal{Q}^{(N-2)}_{<2>} = \mathcal{G}^{(N-1)} \times_{3,4}^{1,2} \mathcal{W}^{(N-1)}$ with $\mathcal{G}^{(N-1)} \in \mathbb{R}^{R_{N-1} \times I_{N-1} \times L_{N-1} \times R_N}$ and $\mathcal{W}^{(N-1)} \in \mathbb{R}^{L_{N-1} \times R_N \times I_N \times R_1 \times L_1 \times \cdots \times L_{N-2}}$. Factor $\mathcal{W}^{(N-1)}$ merges

two latent factors: last ring tensor $\mathcal{G}^{(N)}$ and core tensor $\mathcal{C}$, according to $\mathcal{W}^{(N-1)} = \texttt{circ}\left(\mathcal{G}^{(N)} \times_3^N \mathcal{C}, 1\right)$. Tensor $\mathcal{W}^{(N-1)}$ contains $N$ training samples across its third mode, expressed in the tensor latent space. After applying the circular mode shift to $\mathcal{W}^{(N-1)}$, and then unfolding the resulting tensor along its first mode, we get $\mathcal{Q} = \texttt{circ}\left(\mathcal{W}^{(N-1)}, 2\right) \in \mathbb{R}^{I_N \times R_1 \times L_1 \times \ldots \times L_{N-1} \times R_N}$ and $\boldsymbol{Q} = \boldsymbol{Q}_{(1)} \in \mathbb{R}^{I_N \times R_1 L_1 \cdots L_{N-1} R_N}$ that can be directly fed to train a classifier.

Finally, the above considerations lead to Algorithm 1, referred to as SVD-TW. It computes training latent factors, i.e. ring tensors $\{\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(N-1)}\}$ and core-ring tensor $\mathcal{Q}$ from multiway training samples ordered along the $N$-th mode of $\mathcal{X}$. Algorithm 2 is a part of Algorithm 1 and it performs the $k$-unfold operations.

**Algorithm 1 SVD-TW decomposition**

**Input :** $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ – multiway training samples ordered along the $N$-th mode,
$\boldsymbol{R} = [R_1, \ldots, R_N, L_1, \ldots, L_N]$ – TW ranks
**Output:** $\{\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(N-1)}\}$ – set of $N-1$ ring tensors,
$\mathcal{Q} \in \mathbb{R}^{I_N \times R_N \times R_1 \times L_1 \times \ldots \times L_N}$ – core-ring tensor

**for** $k = 1, \ldots, N$ **do**
  $[\boldsymbol{W}, J_k] = \texttt{k-unfolding}\left(\mathcal{X}, \mathcal{W}, \boldsymbol{R}, N, k\right)$,
  Compute $[\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}] = \texttt{tsvd}(\boldsymbol{W}, J_k)$,
  $\tilde{\boldsymbol{W}} = \boldsymbol{S}\boldsymbol{V}^T$,
  **if** $k = 1$ **then**
    $\tilde{\mathcal{G}}^{(1)} = \texttt{reshape}\left(\boldsymbol{U}, [I_1, R_1, L_1, R_2]\right)$,
    $\mathcal{G}^{(1)} = \texttt{permute}\left(\tilde{\mathcal{G}}^{(1)}, [2, 1, 3, 4]\right)$,
    $\tilde{\mathcal{W}} = \texttt{reshape}\left(\tilde{\boldsymbol{W}}, [R_1, L_1, R_2, I_2, \ldots, I_N]\right)$,
    $\mathcal{W} = \texttt{permute}\left(\tilde{\mathcal{W}}, [3, \ldots, N+2, 1, 2]\right)$,
  **else if** $1 < k < N$ **then**
    $\mathcal{G}^{(k)} = \texttt{reshape}\left(\boldsymbol{U}, [R_k, I_k, L_k, R_{k+1}]\right)$,
    $\tilde{\mathcal{W}} = \texttt{reshape}\left(\tilde{\boldsymbol{W}}, \ldots\right.$
    $\left.[L_k, R_{k+1}, I_{k+1}, \ldots, I_N, R_1, L_1, \ldots, L_{k-1}]\right)$,
    $\mathcal{W} = \texttt{permute}\left(\tilde{\mathcal{W}}, [2, \ldots, N+2, 1]\right)$,
  **else**
    $\mathcal{Q} = \texttt{permute}\left(\mathcal{W}, [2, 1, 3, \ldots, N+2]\right)$,
  **end**
**end**

**Algorithm 2** $k$**-unfolding**

**Input :** $\mathcal{X}$, $\mathcal{W}$, $\boldsymbol{R}$, $N$, $k$
**Output:** $\boldsymbol{W}$, $J_k$

**if** $k = 1$ **then**
  $\boldsymbol{W} = \boldsymbol{X}_{(1)} \in \mathbb{R}^{I_1 \times \prod_{p=2}^{N} I_p}, \quad J_1 = R_1 L_1 R_2,$
**else if** $1 < k < N$ **then**
  $\boldsymbol{W} = \boldsymbol{W}_{<2>}, \quad J_k = L_k R_{k+1},$
**else**
  $\boldsymbol{W} = \boldsymbol{W}_{<3>}, \quad J_k = L_N,$
**end**

A total number of hyperparameters in the proposed feature extraction method amounts to $2N$, i.e. the number of all ranks. Ranks $\{R_1, \ldots, R_N, L_1, \ldots, L_N\}$ are assumed to be fixed in Algorithm 1, which leads to fixed target ranks $\{J_1, \ldots, J_N\}$. However, this condition can be alleviated by using any rank adaptation criteria [16]. For example, given thresholds for singular values of the subsequent unfoldings, the target ranks can be estimated, and then approximated by integer division rules, e.g. such as in [17]. Thus, all the ranks can be adapted or data-driven estimated.

*Remark 1:* Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ with $m < n$. The computational complexity of the economy-size SVD of $\boldsymbol{A}$ amounts to $\mathcal{O}\left(2nm^2 + m^3\right)$. Under this assumption, the computational complexity of Algorithm 1 can be approximated by:

$$\mathcal{O}\left(2\sum_{l=2}^{N} I^l R^{N+4-l} + 2I^{N+1} + I^3 + (N-1)(RI)^3\right), \quad (11)$$

It is easy to note that under the assumption that $I >> R$ and $R \geq 1$, the first term in (1) for $k = 2$ goes faster with $I$ than the first term in (11), i.e. $2\sum_{l=2}^{N-1} I^l R^{N+4-l}$. For $k > 2$, this rate is even more favorable. After straightforward calculations, one can conclude that the complexity in (1) grows faster with $I$ than in (11), if the inequality $2NR^3 + R^{2N} > 2R^4 + 2I$ is satisfied, but this is usually the practical case when $R$ is not too small, i.e. when $R^{2N} > 2I$. Note that the computational complexity in (11) is calculated under the assumption that the economy-size SVD was used. However, the truncated SVD can be obtained with even a lower cost if only a few singular vectors need to be computed, which additionally works in favor of the proposed algorithm.

Let $\tilde{I}$ testing samples be ordered along the $N$-th mode of tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times \ldots \times \tilde{I}_N}$. Assuming that all the testing samples share the same tensor latent space as the training samples, ring tensors $\{\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(N-1)}\}$ obtained by Algorithm 1 from training tensor $\mathcal{X}$ are shared in the testing process. Having the set of the above ring tensors and testing tensor $\mathcal{Y}$, the respective core-ring tensor $\tilde{\mathcal{Q}} \in \mathbb{R}^{\tilde{I}_N \times R_1 \times L_1 \times \ldots \times L_{N-1} \times R_N}$ can be easily obtained by performing simple projections, summarized in Algorithm 3. Unfolding $\tilde{\mathcal{Q}}$ with respect to its first mode leads to matrix $\tilde{\boldsymbol{Q}} = \tilde{\boldsymbol{Q}}_{(1)} \in \mathbb{R}^{\tilde{I}_N \times R_1 L_1 \cdots L_{N-1} R_N}$ whose rows represent the testing samples in the reduced space. Both $\boldsymbol{Q}$ and $\tilde{\boldsymbol{Q}}$ are given to a classifier.

**Algorithm 3 Projection**

**Input :** $\tilde{\mathcal{X}} \in \mathbb{R}^{I_1 \times \ldots \times I_{N-1} \times \tilde{I}_N}$ – multiway testing samples ordered along the $N$-th mode,
$\{\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(N-1)}\}$ – ring tensors obtained from training
**Output:** $\tilde{\mathcal{Q}} \in \mathbb{R}^{\tilde{I}_N \times R_N \times R_1 \times L_1 \times \ldots \times L_N}$ – core-ring tensor for testing data

$\tilde{\mathcal{Q}} = \tilde{\mathcal{X}} \times_1^2 \mathcal{G}^{(1)}$,
**for** $j = 2, \ldots, N-1$ **do**
  $\tilde{\mathcal{Q}} \leftarrow \tilde{\mathcal{Q}} \times_{1,N+1}^{2,1} \mathcal{G}^{(j)}$
**end**
$\tilde{\mathcal{Q}} = \texttt{permute}\left(\tilde{\mathcal{Q}}, [1, N+2, 2, \ldots, N+1]\right)$

## IV. EXPERIMENTS

The numerical experiments were carried out using various datasets containing high-dimensional samples. The proposed method was tested on classification problems and compared with several competitive tensor decomposition methods.

### A. Setup

The following datasets were used in the tests:

- **MI** dataset was created from audio recordings obtained from the MIS database[1] of the University of Iowa. Six musical instruments, such as cello, soprano saxophone, violin, bassoon, flute, and piano, and 12 audio samples for each instrument were selected. The 4 s sound excerpts containing meaningful information in the frequency range from 86 Hz to 10.9 kHz were used in tests. The temporal waveforms were transformed into log-magnitude spectrograms with the resolution of 64 frequency points and 138 time intervals.
- **ORL** dataset[2] contains the facial images of 40 people. Each person has 10 facial images with differing lighting, facial expressions, and occlusion settings (such as glasses and no glasses). The resolution of each grayscale image is $112 \times 92$ pixels.
- **COIL-100** dataset[3] was obtained from the Columbia Object Image Library and contains 7200 RGB images of several rotated and rescaled objects illustrated on non-uniform background, and grouped into 100 classes. Each images has the resolution of $128 \times 128$ pixels.
- **Reuters** dataset (*Reuters21578*)[4] contains a collection of 8,293 textual documents using a vocabulary of 18,000 words. 8,203 documents, grouped into 40 non-uniform classes, were selected. The first class is represented by 3,713, while the last one has only 10 documents. Then, the document-word matrix was tensorized using the `reshape` operation into tensor $\mathcal{X} \in \mathbb{R}^{8203 \times 30 \times 30 \times 20}$, where each document vector ($18,000$ entries) was heuristically converted into tensor $30 \times 30 \times 20$.

Algorithm 3 was compared with the following algorithms: higher-order singular value decomposition (HO-SVD) [18], segmented convex-hull nonnegative tensor factorization based on higher-order cluster median model (cSCH-NTF(med)) [19], SVD-based tensor train (SVD-TT) decomposition [20], SVD-based tensor ring (SVD-TR) decomposition [21], and the baseline TW decomposition [12]. The target ranks and related parameters are listed in Table I. The other parameters are set as default. The algorithms were implemented in MATLAB 2023b and run on a workstation equipped with a 24-core Intel i9-13900 CPU, 128 GB RAM, an SSD drive, and Windows 11 Pro. The `tsvd` function in Algorithm 1 was computed with the in-built economy-size SVD routine, and the tensor contractions in Algorithm 3 were obtained with the in-built `tensorprod` function. The proposed algorithm can also be easily implemented in other programming environments, i.e. in Python, R, or C++.

The multilinear samples were divided into training and testing sets according to the 5-fold cross-validation (CV) rule, using the MATLAB `cvpartition` function. The Monte Carlo (MC) analysis was conducted with 10 runs for each algorithm. The extracted multilinear features were vectorized and given to the standard 1-nearest neighborhood (NN) classifier with the Euclidean distance, taken from *Statistics and Machine Learning Toolbox* in MATLAB 2023b. The algorithms were evaluated in terms of the misclassification rate (MCR), the Hinton diagrams of confusion matrices, and runtime of the training process.

TABLE I: Ranks and other parameters set in the tested methods on each dataset

| Algorithms | MI | ORL | COIL-100 | Reuters |
|---|---|---|---|---|
| HO-SVD | | | | |
| rank | 24 | 50 | 50 | 400 |
| cSCH-NTF(med) | | | | |
| target rank | 24 | 50 | 50 | 400 |
| SVD-TT | | | | |
| TT ranks | $[4, 6]$ | $[3, 15]$ | $[5, 15, 3]$ | $[20, 20, 20, 20]$ |
| SVD-TR | | | | |
| $R_0$ | 3 | 2 | 2 | 5 |
| $\tau$ | 0.08 | 0.2 | 0.28 | 0.65 |
| TW | | | | |
| outer ranks | $[2, 2, 3]$ | $[3, 3, 2]$ | $[2, 2, 2]$ | $[5, 2, 5, 5]$ |
| inner ranks | $[2, 2, 3]$ | $[3, 3, 2]$ | $[2, 2, 2]$ | $[2, 5, 2, 2]$ |
| SVD-TW | | | | |
| outer ranks | $[2, 3, 3]$ | $[3, 3, 2]$ | $[2, 2, 4]$ | $[5, 2, 5, 5]$ |
| inner ranks | $[2, 2, 3]$ | $[3, 3, 2]$ | $[2, 2, 7]$ | $[2, 5, 2, 2]$ |

### B. Results

The Hinton diagrams of the averaged confusion matrices obtained with the tested methods on the `MI`, `ORL`, `COIL-100`, and `Reuters` datasets are illustrated in Figs. 2, 3, 4, and 5, respectively. The MCR values averaged over MC runs are given in Table II. The runtime (in seconds) of training the tested algorithms on all the datasets is listed in Table III.

TABLE II: Averaged misclassification ratio (MCR) obtained with the tested algorithms on all datasets

| Algorithms | MI | ORL | COIL-100 | Reuters |
|---|---|---|---|---|
| HO-SVD | 17.21 | 6.5 | 0.32 | 17.41 |
| cSCH-NTF(med) | 4.56 | 4.88 | 0.97 | **9.33** |
| SVD-TT | 11.47 | 2.83 | 0.22 | 18.41 |
| SVD-TR | 7.21 | 7.58 | 3.18 | 16.51 |
| TW | 11.62 | 10.88 | 41.65 | 23.14 |
| SVD-TW | **3.38** | **2.25** | **0.13** | 12.99 |

### C. Discussion

The proposed algorithm is addressed for multilinear feature extraction from multiway arrays with a high number of modes. Due to the use of recursive SVD-based updates

[1] http://theremin.music.uiowa.edu

[2] https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[3] http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php

[4] https://paperswithcode.com/dataset/reuters-21578

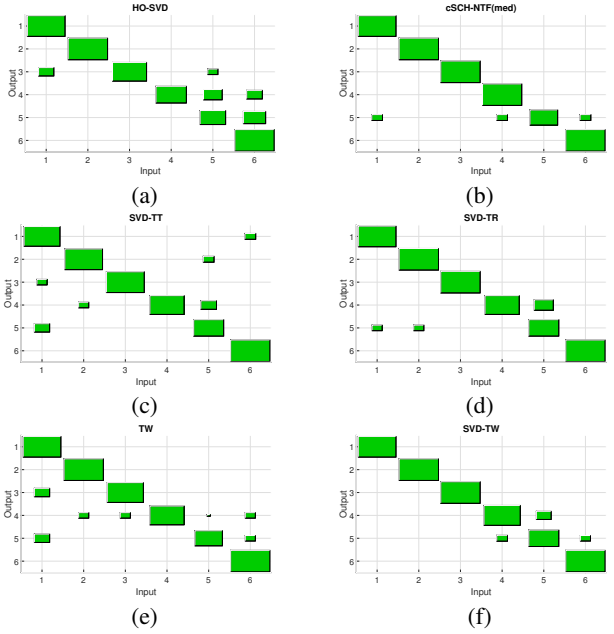



Fig. 2: Hinton diagrams of averaged confusion matrices obtained on the MI dataset with algorithms: (a) HO-SVD, (b) cSCH-NTF(med), (c) SVD-TT, (d) SVD-TR, (e) TW, (f) SVD-TR

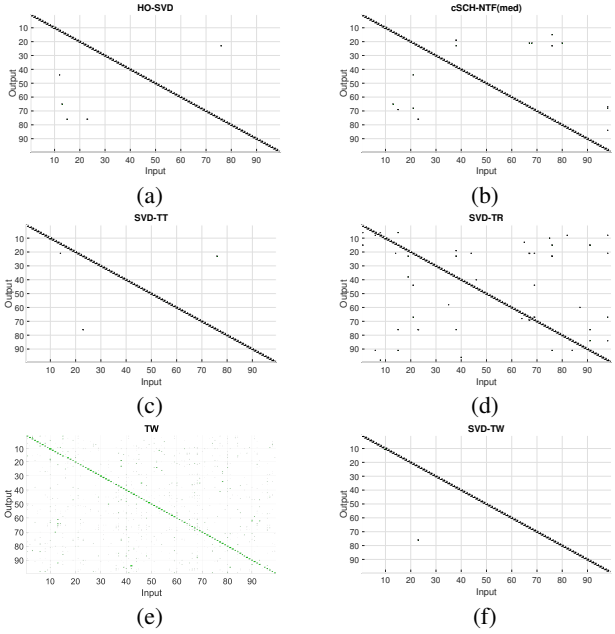



Fig. 4: Hinton diagrams of averaged confusion matrices obtained on the COIL-100 dataset with algorithms: (a) HO-SVD, (b) cSCH-NTF(med), (c) SVD-TT, (d) SVD-TR, (e) TW, (f) SVD-TR

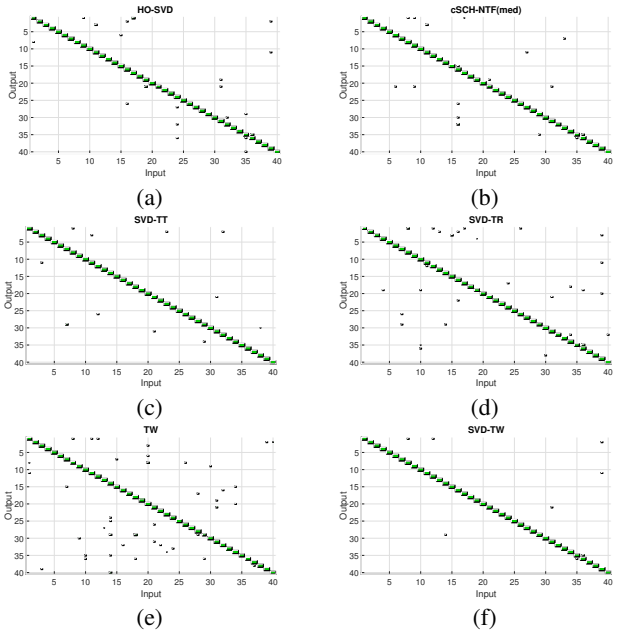



Fig. 3: Hinton diagrams of averaged confusion matrices obtained on the ORL dataset with algorithms: (a) HO-SVD, (b) cSCH-NTF(med), (c) SVD-TT, (d) SVD-TR, (e) TW, (f) SVD-TR

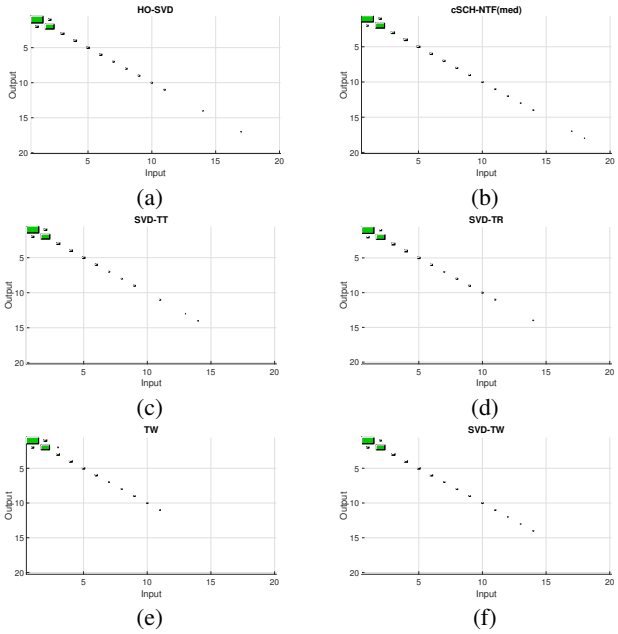



Fig. 5: Hinton diagrams of averaged confusion matrices obtained on the Reuters dataset with algorithms: (a) HO-SVD, (b) cSCH-NTF(med), (c) SVD-TT, (d) SVD-TR, (e) TW, (f) SVD-TR

the features are propagated from the starting mode until the last one that combines the core tensor capturing the cross-modal interactions with the ring tensor representing the samples. This architectural approach is used only in the proposed SVD-TW model. The numerical results confirm the efficiency of the proposed algorithm for classification tasks. According to Table II, SVD-TW outperforms all the tested algorithms on three datasets: MI, ORL, and COIL-100.

TABLE III: Runtime (in seconds) obtained with the tested algorithms on all datasets

| Algorithms | MI | ORL | COIL-100 | Reuters |
|---|---|---|---|---|
| HO-SVD | 0.04 | 0.13 | 32.54 | 36.86 |
| cSCH-NTF(med) | 0.27 | 2.73 | 261.4 | 132.7 |
| SVD-TT | **0.025** | **0.1** | **17.45** | **11.39** |
| SVD-TR | 0.04 | 0.35 | 35.52 | 49.66 |
| TW | 0.3 | 2.9 | 333.5 | 1029 |
| SVD-TW | 0.05 | 0.37 | 32.97 | 32.17 |

Figs. 2, 3, and 4 also confirm the efficiency of SVD-TW in the classification tasks. In each case, the false negative and false positive errors are lowest when SVD-TW is used. Only for the `Reuters` database, SVD-TW is the second best, conceding cSCH-NTF(med). Fig. 5 shows that cSCH-NTF(med) gives a little longer tail of the true positive entries than SVD-TW, which affects the MCR results. Maybe, it results from the specificity of such data that are very sparse and nonnegative. It is well-known that nonnegatively constrained decompositions are more suitable for such cases. It is also possible that this effect may be due to strong class imbalance, which takes place for the `Reuters` database. The 1-NN classifier with the Euclidean distance is also not the best choice for class imbalanced or noisy data.

Regarding the computational complexity, SVD-TW is not the fastest in all the tests, which obviously results from the complexity of the model. However, it is much faster than cSCH-NTF(med) and TW. For the `Reuters` dataset, it is even the second fastest algorithm.

## V. CONCLUSIONS

In this study, a new algorithm for multilinear feature extraction based on the TW decomposition model was proposed. Numerical experiments show that the proposed SVD-TW model outperforms several well-known tensor decomposition methods in classification tasks on various datasets, including spectrograms of selected musical instruments, facial images (ORL), and images of diverse objects (COIL-100). This makes it an effective solution for classification problems with limited training samples. For large-scale or high-dimensional datasets, other machine learning tools—such as convolutional neural networks—are generally more suitable.

The algorithm is also relatively fast—significantly faster than the baseline TW decomposition and the cSCH-NTF(med) algorithm which appears to be more effective for classifying textural documents. It is worth noting that the current version of the SVD-TW decomposition does not incorporate prior class label information during training, similar to other SVD-based feature extraction methods like PCA, as well as many existing tensor decomposition techniques. Naturally, classification accuracy could likely be improved by incorporating a penalty-based approach or post-processing step, such as tensor subspace analysis. However, this remains a challenging task and represents a promising direction for future research.

## REFERENCES


[1] M. Wang, D. Hong, Z. Han, J. Li, J. Yao, L. Gao, B. Zhang, and J. Chanussot, "Tensor decompositions for hyperspectral data processing in remote sensing: A comprehensive review," *IEEE Geoscience and Remote Sensing Magazine*, vol. 11, no. 1, pp. 26–72, 2023.

[2] T.-H. Zhang, J.-L. Zhao, S. Fang, Z. Li, and M.-G. Gong, "Full-mode-augmentation tensor-train rank minimization for hyperspectral image inpainting," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–13, 2024.

[3] S. Du, Y. Shi, W. Hu, W. Wang, and J. Lian, "Robust tensor factorization for color image and grayscale video recovery," *IEEE Access*, vol. 8, pp. 174 410–174 423, 2020.

[4] W. Dong, T. Huang, G. Shi, Y. Ma, and X. Li, "Robust tensor approximation with laplacian scale mixture modeling for multiframe image and video denoising," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 6, pp. 1435–1448, 2018.

[5] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. PHAN, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, 2015.

[6] 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 Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[7] Y. Panagakis, J. Kossaifi, G. G. Chrysos, J. Oldfield, M. A. Nicolaou, A. Anandkumar, and S. Zafeiriou, "Tensor methods in computer vision and deep learning," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 863–890, 2021.

[8] Y. Luo, X. Zhao, Z. Li, M. K. Ng, and D. Meng, "Low-rank tensor function representation for multi-dimensional data recovery," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 5, pp. 3351–3369, 2024.

[9] E. E. Papalexakis, C. Faloutsos, and N. Sidiropoulos, "Tensors for data mining and data fusion," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, pp. 1 – 44, 2016.

[10] Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki, "Tensor ring decomposition," *ArXiv*, vol. abs/1606.05535, 2016.

[11] I. V. Oseledets, "Tensor-train decomposition," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.

[12] Z.-C. Wu, T.-Z. Huang, L.-J. Deng, H.-X. Dou, and D. Meng, "Tensor wheel decomposition and its tensor completion application," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 27 008–27 020.

[13] K. Fonal and R. Zdunek, "Fast recursive nonnegative standard and hierarchical tucker decomposition," *IEEE Signal Process. Lett.*, vol. 26, no. 9, pp. 1265–1269, 2019.

[14] M. Wang, H. Cui, and H. Li, "SVD-based algorithms for tensor wheel decomposition," *Advances in Computational Mathematics*, vol. 50, no. 5, 2024.

[15] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[16] F. Sedighin, A. Cichocki, and A.-H. Phan, "Adaptive rank selection for tensor ring decomposition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 3, pp. 454–463, 2021.

[17] O. Mickelin and S. Karaman, "On algorithms for and computing with the tensor ring decomposition," *Numerical Linear Algebra with Applications*, vol. 27, no. 3, p. e2289, 2020.

[18] L. De Lathauwer, B. de Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal of Matrix Analysis and Applications*, vol. 21, pp. 1253–1278, 2000.

[19] R. Zdunek and T. Sadowski, "Segmented convex-hull algorithms for near-separable NMF and NTF," *Neurocomputing*, vol. 331, pp. 150–164, 2019.

[20] J. A. Bengua, H. N. Phien, and H. D. Tuan, "Optimal feature extraction and classification of tensors via matrix product state decomposition," in *2015 IEEE International Congress on Big Data*, June 2015, pp. 669–672.

[21] O. Mickelin and S. Karaman, "On algorithms for and computing with the tensor ring decomposition," *Numerical Linear Algebra with Applications*, vol. 27, 2020.