INDUSTRIAL AND SYSTEMS ENGINEERING

ISE

Deep Tensor Decomposition: A Survey

MENG ZHAO[1], JIUYUN HU[2], ZIYUE LI[3], ZEKAI WANG[4], AND LIJUN SUN[5]

[1]Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, 18015 USA

[2]Department of Industrial Engineering, Arizona State University, Tempe, 85281, AZ, USA

[3]Department of Operations and Technology, Technical University of Munich, Munich, 80333, Bavaria, Germany

[4]Department of Analytics, Fairfield University, Fairfield, 06824, CT, USA

[5]Department of Civil Engineering, McGill University, Montreal, H3A0G4, Quebec, Canada

ISE Technical Report 25T-022

LEHIGH UNIVERSITY

Neurocomputing 664 (2026) 132074

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Survey paper

Deep tensor decomposition: A survey

Meng Zhao [a,*], Jiuyun Hu [b], Ziyue Li [c], Zekai Wang [d], Lijun Sun [e]

[a] Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, 18015, PA, USA
[b] Department of Industrial Engineering, Arizona State University, Tempe, 85281, AZ, USA
[c] Department of Operations and Technology, Technical University of Munich, Munich, 80333, Bavaria, Germany
[d] Department of Analytics, Fairfield University, Fairfield, 06824, CT, USA
[e] Department of Civil Engineering, McGill University, Montreal, H3A0G4, Quebec, Canada

ARTICLE INFO

Communicated by X. Luo

Keywords:
Deep tensor decomposition
Neural tensor decomposition
Hierarchical representation
Multi-layer structure
Feature extraction

ABSTRACT

Tensor decomposition (TD) has been recognized as an effective technique for multilinear dimensionality reduction and feature extraction for decades. However, traditional TD approaches often struggle to capture complex hierarchical structures and nonlinear relationships in high-dimensional datasets. For instance, in biomedical settings, disease groups may naturally contain subgroups or exhibit hierarchical structures; mechanistic interactions among diseases, drugs and targets often demonstrate nonlinearity. To address these challenges, a new paradigm, deep tensor decomposition (deep TD) has recently emerged inspired by the success of deep learning. Deep TD techniques can be mainly divided into two categories: linear and nonlinear deep TD. Linear deep TD exploits the layered structure of deep neural networks (DNNs) to recursively factorize factor matrices obtained from the classic TD enabling feature extraction at multiple levels of granularity. Nonlinear deep TD leverages the expressive power of DNNs to capture nonlinear correlations within the data. Despite rapid progress, there remains no unified treatment of deep TD methods. In this survey, we provide a comprehensive review of deep TD models, together with the deep learning training schemes for TD, and applications of deep TD models. Finally, we discuss open challenges and outline promising directions for future research.

1. Introduction

With the increasing availability of high-dimensional data in tensor format across various domains, efficiently analyzing such data has become a critical challenge. A key goal is to extract features and identify latent structures that capture complex patterns in the data. Tensor decomposition (TD), has emerged as a powerful technique for this purpose, and has been successfully applied in diverse domains such as signal processing [1,2], healthcare analysis [3,4], and transportation systems [5,6]. Traditional TD approaches, such as CANDECOMP/PARAFAC (CP) [7] and Tucker decomposition [8], focus on factorizing a tensor into low-rank components to capture multilinear correlations among different modes. While these models perform well for relatively simple and structured data, they often face challenges when applied to complex datasets for representing hierarchical or nonlinear relationships. To overcome these limitations, researchers have proposed advanced hierarchical (multi-layer) and nonlinear TD approaches. Hierarchical TD models [9,10] extend traditional TD methods by recursively factorizing the components of initial decompositions. This enables the extraction of deeper, multi-level representations that better reflect the structure of real-world data. For example, Hierarchical Alternating Least Squares (HALS) [9] was the first to introduce layered TD structure, iteratively capturing local representations within each decomposition level. In parallel, nonlinear TD models aim to explore more flexible patterns by moving beyond the multilinear assumptions. For instance, [11] proposed the Bayesian nonlinear tensor factorization framework which incorporates Gaussian process (GP) to model a variety of nonlinear relationships in the tensor data.

In recent years, deep neural networks (DNNs) [12] have gained significant attention and achieved state-of-the-art performance in a wide range of tasks [13–15]. DNNs are composed of multiple layers between the input and output layers, where the term "deep" refers to the presence of multiple hidden layers enabling networks to learn complex representations from data that "shallow" architectures cannot easily handle. Building upon the remarkable success of DNNs, researchers have extended deep learning principles to tensor analysis, leading to the development of deep tensor decomposition (deep TD). These models

* Corresponding author.
Email address: m.zhao@lehigh.edu (M. Zhao).

https://doi.org/10.1016/j.neucom.2025.132074
Received 2 July 2025; Received in revised form 27 September 2025; Accepted 8 November 2025
Available online 11 November 2025

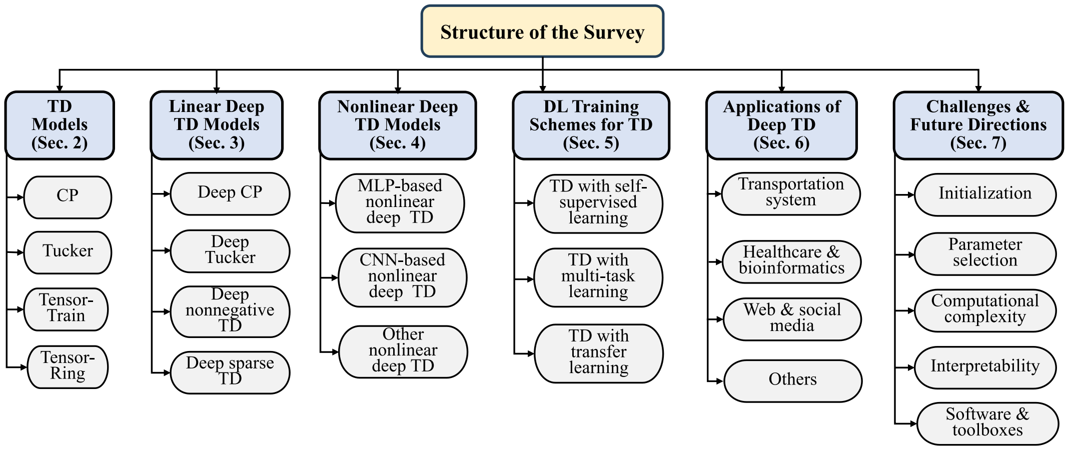


Fig. 1. Structure of the survey.

aim to integrate the representational power of deep learning with the structure-preserving benefits of traditional tensor decomposition. Broadly, existing deep TD methods mainly fall into two categories. The first involves hierarchical decomposition structures, in which each layer performs a basic multi-linear factorization of the previous layer's output [16–19]. These models are designed to extract features at varying levels of abstraction, progressing from global, coarse-grained patterns to more fine-grained, detailed structures. For example, HNCPD [17] and DNTF [19] extend CP decomposition and Tucker decomposition, respectively, through iterative decomposition to uncover hierarchical correlations in complex tensor data. The second category of deep TD incorporates layers of nonlinear operations through neural networks to model nonlinear interactions within tensor components [20–28]. A representative example is CoSTCo (Convolutional Sparse Tensor Completion) [20], the first method to embed convolutional neural network (CNN) structures into CP decomposition, enabling the model to learn complex nonlinear interactions among components. This integration leads to an improved performance on the sparse tensor completion task. Building on this line of work, [29] constructs a neural Tucker framework that employs an auto-encoding module to refine the core of Tucker decomposition. [30] introduced CoATR, a convolutional-based generalized autoregressive tensor-ring decomposition method designed for spatio-temporal data completion, extending the idea beyond CP and Tucker to more expressive tensor-ring representations.

Despite the growing interest in developing deep TD approaches for capturing hierarchical and nonlinear relationships within tensor data, this field remains relatively underexplored, and a systematic overview is still lacking. This paper aims to fill this gap by providing a thorough review of recent advancements in deep TD, covering fundamental models, key variants, training schemes, and representative applications, as summarized in Fig. 1. The rest of the paper is organized as follows. Section 3 presents linear deep TD models and their extensions, including the incorporation of constraints such as non-negativity and sparsity. Section 4 summarizes nonlinear deep TD models that utilize various DNN structures such as multi-layer perceptron (MLP) and CNN. In Section 5, we review the popular training schemes in deep learning being applied to deep TD such as self-supervised learning and transfer learning. Section 6 summarizes the main applications of deep TD models across various domains. Finally, Section 7 concludes this survey by discussing open challenges and presenting perspectives on future research directions for deep TD methods.

2. Tensor decomposition models

This section first introduces the notations used throughout the survey. It then provides definitions and formulations of classical tensor decomposition models, including CANDECOMP/PARAFAC (CP), Tucker, Tensor-Train (TT), and Tensor-Ring (TR) models.

Table 1
Description of the notations used in this survey.

Notations	Description
$x, \mathbf{x}, \mathbf{X}, \mathcal{X}$	scalar, vector, matrix, and tensor
$\mathcal{X}_{i_1,\ldots,i_N}$	the $(i_1,\ldots,i_N)$-element of the N-th order tensor $\mathcal{X}$
$\mathbf{X}^\top$	transpose of the matrix $\mathbf{X}$
$\mathbf{X}^{-1}$	inverse of the matrix $\mathbf{X}$
$\mathbf{X}_{(n)}$	the mode-n matricization of the tensor $\mathcal{X}$
$\circ, *, \odot, \otimes$	outer, Hadamard, Khatri-Rao, and Kronecker product
$\mathcal{X} \times_n \mathbf{X}$	the n-mode product of tensor $\mathcal{X}$ with matrix $\mathbf{X}$
$\mathrm{Tr}(\mathbf{X})$	trace of the matrix $\mathbf{X}$
$\|\cdot\|_F$	Frobenius norm
$\|\cdot\|_1$	ℓ_1 norm
$\|\cdot\|_2$	ℓ_2 norm

2.1. Notations

Table 1 summarizes the commonly used notations in this survey paper for ease of reference. More details can be found in this foundational work [31]. Matricization, also known as unfolding or flattening, is the process of converting a tensor into a matrix by rearranging its elements along specific modes.

2.2. Tensor decomposition models

Definition 1 (CP decomposition). Given a N-way tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$, the CP model factorizes it into the sum of R component rank-one tensors [31], formulated as follows,

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \circ \mathbf{a}_r^{(N)} \equiv [\![\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!], \tag{1}$$

where R is a positive integer denoting tensor canonical rank, $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n\times R}$ $(n \in \{1,\ldots,N\})$ is the n-th factor matrix and $\mathbf{a}_r^{(n)} \in \mathbb{R}^{I_n}$ $(r \in \{1,\ldots,R\})$ denotes the r-th column of the n-th factor matrix.

Definition 2 (Tucker decomposition). Tucker model factorizes the N-way tensor $\mathcal{X}$ as the mode products of a core tensor $\mathcal{G} \in \mathbb{R}^{J_1\times J_2\times\ldots\times J_N}$ and N factor matrices $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n\times J_n}$ [8],

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{A}^{(1)} \cdots \times_N \mathbf{A}^{(N)} \equiv [\![\mathcal{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \cdots, \mathbf{A}^{(N)}]\!], \tag{2}$$

where $J_n (1 \le n \le N)$ is the number of dimensionality in mode-n of the core tensor $\mathcal{G}$.

Definition 3 (Tensor-Train decomposition). Tensor-Train (TT) model [32] expresses the N-way tensor $\mathcal{X}$ as multilinear products of a chain of third-order tensors except the first and last are matrices. It is formulated as,

$$\mathcal{X}_{i_1,\ldots,i_N} \approx \mathcal{A}_1[1, i_1, :]\mathcal{A}_2[:, i_2, :] \cdots \mathcal{A}_N[:, i_N, 1], \tag{3}$$

where $\mathcal{A}_n \in \mathbb{R}^{r_{n-1}\times I_n \times r_n}$ is called the n-th TT-core and r_n is the TT-rank ($r_0 = r_N = 1$).

Definition 4 (Tensor-Ring decomposition). Tensor-Ring (TR) model [33] decomposes the N-way tensor $\mathcal{X}$ by a sequence of third-order latent tensors multiplied circularly where the first and last tensors are also connected. It is formulated as,

$$\begin{aligned}\mathcal{X}_{i_1,\ldots,i_N} &\approx \mathrm{Tr}\{\mathbf{A}_1(i_1)\mathbf{A}_2(i_2)\cdots \mathbf{A}_N(i_N)\} \\ &\approx \mathrm{Tr}\{\prod_{n=1}^{N} \mathbf{A}_n(i_n)\},\end{aligned} \tag{4}$$

where $\mathbf{A}_n(i_n) \in \mathbb{R}^{r_n \times r_{n+1}}$ is the i_n-th lateral slice matrix of the latent tensor $\mathcal{A}_n$, also called the n-th TR-core or node.

3. Track I: linear deep TD models

This section reviews the first category of deep TD models, namely linear deep TD, in which the factor matrices or core tensors derived from each layer are recursively decomposed into deeper levels, enabling the capture of hierarchical features within the tensor data. The main idea of linear deep TD leverages the multi-layer architecture of DNNs.

3.1. Basic linear deep TD models

In this subsection, we primarily introduce two types of linear deep TD models: deep CP decomposition and deep Tucker decomposition, which are built upon the foundations of the traditional CP and Tucker models, as illustrated in Fig. 2. The flowchart of linear deep TD built on CP model is demonstrated in Fig. 3.

3.1.1. Deep CP decomposition

Traditional one-layer CP decomposition defined in Eq. (1) may fail to extract hierarchical structures or multi-level features in complex datasets. To address this limitation, [17] proposed a deep CP decomposition model which extends the basic CP into a multi-layer structure by minimizing the total reconstruction loss resulting from using the factor matrices in each layer to reconstruct the tensor data. The first step of a deep CP model is to perform the regular CP decomposition defined in Eq. (1). For the layers that follow, a sequential and hierarchical TD is performed on each of the factor matrices. The loss function for deep CP is given as follows,

$$\min_{\mathbf{A},\mathbf{S}} \|\mathcal{X} - [\![\mathbf{A}_1^{(1)}\mathbf{A}_2^{(1)} \ldots \mathbf{A}_L^{(1)}\mathbf{S}_L^{(1)}, \ldots, \mathbf{A}_1^{(N)}\mathbf{A}_2^{(N)} \ldots \mathbf{A}_L^{(N)}\mathbf{S}_L^{(N)}]\!]\|_F^2, \tag{5}$$

where L is the number of decomposition layers, $\mathbf{A}_l^{(n)} \in \mathbb{R}^{r_{l-1}\times r_l}(l = 2,\ldots,L)$, $\mathbf{S}_L^{(n)} \in \mathbb{R}^{r_L \times R}$, and r_l is the tensor rank for layer l.

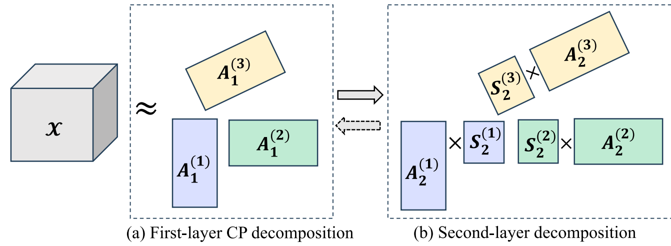


Fig. 3. Flowchart of linear deep TD model using CP decomposition as an example.

3.1.2. Deep tucker decomposition

Similarly, deep Tucker decomposition takes Tucker model as a building block to stack up a multi-layer structure, therefore, gradually extracting more abstract structures at higher layers. It begins with a basic Tucker decomposition defined in Eq. (2), followed by iterative decomposition of the core tensor from the previous step. Tensor $\mathcal{X}$ is reconstructed using the core tensor $\mathcal{G}_L$ in the highest layer and the factor matrices in each layer [19] by minimizing the following least square loss,

$$\min_{\mathcal{G}_L,\{\mathbf{A}_1^{(n)}\},\ldots,\{\mathbf{A}_L^{(n)}\}} \|\mathcal{X} - \mathcal{G}_L \times_1 \mathbf{A}_L^{(1)} \cdots \times_N \mathbf{A}_L^{(N)} \cdots \times_1 \mathbf{A}_1^{(1)} \cdots \times_N \mathbf{A}_1^{(N)}\|_F^2, \tag{6}$$

where L is the number of layers and in the l-th layer, the core tensor $\mathcal{G}_l \in \mathbb{R}^{J_{1,(l)}\times\cdots\times J_{N,(l)}}(l = 1,\ldots,L)$; $\mathbf{A}_1^n \in \mathbb{R}^{I_n \times J_{n,(1)}}$; $\mathbf{A}_l^n \in \mathbb{R}^{J_{n,(l-1)}\times J_{n,(l)}}$ $(l = 2,\ldots,L)$.

3.2. Variants of linear deep TD models

Beside basic linear deep TD models, recent studies have also introduced their variants by adding constraints on factor matrices, such as non-negativity, to enhance solution uniqueness and interpretability. For example, the non-negativity constraint has been imposed on the factor matrices $\mathbf{A}^{(n)}(n = 1,\ldots,N)$ in deep CP model and the core tensor $\mathcal{G}$ in deep Tucker model [17,19]. This subsection reviews the two most common variants of linear deep TD models: deep nonnegative TD and deep sparse TD. Using Tucker decomposition as an example, constrained linear deep TD can be generally formulated by incorporating regularization terms into the original objective function, as shown below,

$$\begin{aligned}\min_{\mathcal{G}_L,\{\mathbf{A}_1^{(n)}\},\ldots,\{\mathbf{A}_L^{(n)}\}} &\|\mathcal{X} - \mathcal{G}_L \times_1 \mathbf{A}_L^{(1)} \cdots \times_N \mathbf{A}_L^{(N)} \cdots \times_1 \mathbf{A}_1^{(1)} \cdots \times_N \mathbf{A}_1^{(N)}\|_F^2 \\ &+ \alpha c_{\mathcal{G}}(\mathcal{G}_L) + \sum_n \sum_l \beta_{n,l} c_{n,l}(\mathbf{A}_l^{(n)}),\end{aligned} \tag{7}$$

where $c_{\mathcal{G}}(\cdot)$ and $c_{n,l}(\cdot)$ are penalty functions for core tensor and factor matrices; α and $\beta_{n,l}$ are their corresponding penalty coefficients.

3.2.1. Deep nonnegative TD

In certain applications, domain knowledge imposes specific conditions and requirements for deep TD models. A notable example is topic

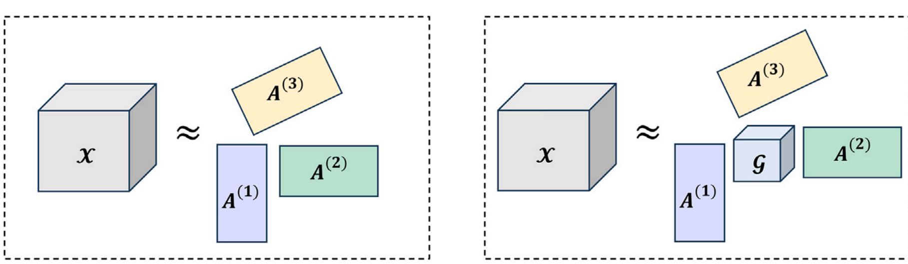


Fig. 2. Illustration of CP decomposition (left) and Tucker decomposition (right).

modeling, where non-negativity is essential to ensure the interpretability and semantic consistency of the extracted topics. To address this, [18] proposed a deep nonnegative CP decomposition based on Eqs. (5), where non-negativity constraints were applied to all factor matrices. The model began with a nonnegative CP decomposition in the first layer. From the second layer onward, each additional layer will right multiply all of the factor matrices by a nonnegative subtopic selection matrix $\mathbf{S}_l^{(n)}$, where l is the level of the layer and n is the n-th dimension. The model achieved superior reconstruction performance compared to the benchmarks and was capable of capturing latent hierarchical structures in multi-modal tensor data. Similarly, [19] proposed a deep non-negative Tucker factorization (DNTF) model, which enforces non-negativity on both the core tensor and the mode matrices to ensure a clearer clustering interpretation of the decomposed components.

3.2.2. Deep sparse TD

In specific fields such as signal processing, text mining, and gene expression analysis, obtaining sparse representations is often desirable to enhance interpretability and improve the uniqueness of decomposition. While sparsity constraints have been widely applied in single-layer TD models, their integration into deep TD frameworks remains an active area of research. To this end, [34] proposed a hierarchical sparse TD with a two-layer decomposition structure. In the first layer, the input tensor is factorized using a Tucker model augmented with an additive noise term, yielding a core tensor and three factor matrices. In the second layer, the core tensor is decomposed into a low-rank component and a sparse component, with a ℓ_1 norm and a nuclear norm imposed, respectively. Then the model was optimized using a block coordinate descent (BCD) and alternative direction method of multipliers (ADMM), which together enable efficient convergence and robust anomaly detection.

4. Track II: nonlinear deep TD models

This section presents a taxonomy of nonlinear deep TD models based on the underlying DNN architectures. We categorize these models into three main groups: (1) MLP-based nonlinear deep TD models, (2) CNN-based nonlinear deep TD models, and (3) other nonlinear deep TD models. Table 2 summarizes the recent advancements in this area. Notably, most existing approaches adopt CP decomposition as the foundational structure in conjunction with DNN, while Tucker, TT, and TR models remain underexplored. In addition, we also briefly mention emerging developments that integrate TD with neural networks [35] or advanced architectures such as Transformers [36,37] and diffusion models [38,39], primarily for parameter compression and model efficiency. While these approaches illustrate the versatility of TD in enhancing state-of-the-art deep learning models and presents a promising direction for further exploration, they fall outside the primary scope of this survey and are not reviewed in detail here. An illustrative flowchart of nonlinear deep TD models is presented in Fig. 4.

Table 2
Categories of nonlinear deep TD models based on the model structure.

Years	Models	Brief Description
2018	BNTD [40]	Bayesian NTN + MLP
	NeuralCP [41]	Bayesian CP + MLP
2019	CoSTCo [20]	CP + CNN
	NTF [21]	CP + MLP
2020	NeurTN [22]	CP + MLP
	POND [42]	GP + CNN
	NTM [43]	Generalized CP (GCP) + tensorized MLP
	NTC [23]	CP + CNN + MLP
	Avocado [44]	3D tensor + DNNs
2021	DAIN [45]	TD + MLP + CNN
	C-PIC [46]	Tensor-Train + CNN
	NePTuNe [47]	Tucker + CNN
2022	DeepTensor [48]	Low-rank MF/CP + 1D & 2D CNN
	JULIA [24]	CP + MLP
	PSC [49]	CP + hierarchical VAE + tiny MLP
	M^2DMTF [25]	Tucker + MLP
	HLRTF [26]	t-SVD + DNN
2023	LightNestle [50]	CP + MLP
	DATC [51]	Tensor completion model + Autoencoder
	GNTD [52]	CP + GNN
2024	ConvTR [53]	Tensor-Ring + CNN
	D-NORM[54]	CP + CNN
	NTRD[55]	Tensor-Ring + NNs
	NeAT[56]	CP + MLP
2025	MSNTucF [28]	Tucker + Multi-head self-attending NN
	Coatr [30]	Tensor-Ring + CNN
	NCPF [27]	CP + MLP
	ANLFT [57]	CP + Attention-mechanism-based NN
	ANTucF [29]	Tucker + Autoencoder

4.1. MLP-based nonlinear deep TD models

Multilayer perceptrons (MLPs) form the backbone of many deep learning architectures due to their ability to capture complex nonlinear relationships in hidden layers through activation functions such as ReLU, sigmoid, and tanh. When integrated with tensor decomposition (TD), MLPs extend traditional multilinear decomposition to better capture nonlinear structures in multi-dimensional data. This integration has demonstrated effectiveness across various domains such as healthcare and knowledge representation.

Two early approaches, NeuralCP [41] and the Bayesian neural tensor decomposition for knowledge base completion [40], both incorporate MLPs into a Variational Bayesian Inference framework but differ in applications and architectures. [41] replaces the standard multilinear product in CP decomposition with an MLP, allowing the model to learn complex interactions while incorporating Bayesian uncertainty modeling. [40] utilizes a Stochastic Gradient Variational Bayesian (SGVB) framework with a multivariate Bernoulli likelihood to model fact existence in knowledge graphs. MLPs are employed to enhance the interactions between latent entity and relation factors. Later, Chen and Li proposed neural tensor network (NeurTN) and neural tensor machine (NTM) [22,43], to model biomedical data. NeurTN combines tensor algebra and DNNs, which offers a more powerful way to capture the nonlinear relationships among drugs, targets, and diseases. Meanwhile, NTM extends multilinear decomposition by introducing a shallow Generalized CP (GCP) layer followed by a deep tensorized MLP. This hybrid design enables the model to learn rich nonlinear feature interactions from multi-aspect tensors.

Several other approaches that combine CP decomposition with MLPs architectures include neural tensor factorization (NTF) framework [21], Joint mUlti-linear and nonLinear IdentificAtion (JULIA) [24], and LightNestle [50]. Specifically, NTF generalizes the conventional CP model by incorporating a LSTM structure to learn temporal dynamics and a MLP structure to model the nonlinear relationships between latent factors; JULIA decomposes a tensor into both linear and nonlinear components, where the linear part is modeled by CP decomposition and the nonlinear part consists of a nonlinear function. The MLP with l_2 error is trained to fit the nonlinear relationships between factor matrices. LightNestle designs a three-component framework: (1) an expressive NN that transfers spatial knowledge from previous embeddings to current embeddings; (2) an attention-based module that encodes temporal patterns into current embeddings with linear complexity; (3) meta-learning-based algorithm that iteratively recovers missing data and updates transfer modules to catch up with learned knowledge.

In contrast to CP-based models, [25] proposed a Tucker-based multi-mode deep nonlinear TD where each factor matrix was parameterized by an MLP. Experiments on two real-world network traffic datasets showed

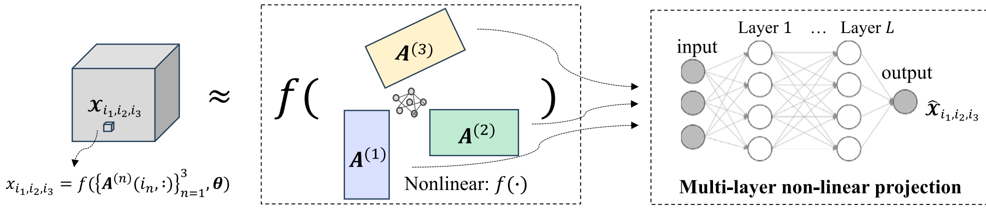


Fig. 4. Flowchart of nonlinear deep TD model using CP as an example.

that the proposed methods achieve both fast retraining and high recovery accuracy. Extending to Tensor Ring (TR) decomposition, [55] proposed a nonlinear TR model by fitting an MLP for each core tensor. These nonlinear mappings are set to be different for all the samples and dimensions, with Gaussian processes used to dynamically adjust the training step size.

4.2. CNN-based nonlinear deep TD models

The layer of CNN is a fundamental part in deep learning, particularly for processing spatially structured data such as images and videos. Combining CNN layer rather than fully connected MLP structure in nonlinear deep TD allows the model to efficiently capture spatial hierarchies and patterns and reduces the risk of overfitting. As the pioneering work in combination of CNN and TD, [20] proposed `CoSTCo` (Convolutional Sparse Tensor Completion), which leverages the expressive power of CNN to model the complex interactions inside tensors and its parameter-sharing scheme to preserve the desired low-rank structure. Based on this pioneering work, [58] provided `NTDMDA` method to obtain miRNA-gene-disease association prediction scores.

Other works in combination of CNN and CP decomposition include Neural Tensor Completion (`NTC`) [23], `DeepTensor` [48], and Distributed Neural tensOR coMpletion (`D-NORM`) [54]. `NTC` [23] is a scheme to infer the missing data in a large network with a representation of three-way tensor. This algorithm consists of a fully connected embedding layer to project the three one-hot encoding features from three dimensions into feature vectors; an interaction map layer to use outer product to map the feature vectors into a tensor; a feature extraction layer with CNN, and an inferring layer perception to complete the missing data. `DeepTensor` [48] decomposed a tensor into low-rank factors, where each factor is further modeled by 2D CNN layers. `D-NORM` [54] adopts two schemes to solve the data recovery problem. First, they design a parameter-efficient multi-layer architecture with CNN to learn nonlinear correlations among data. Second, they reformulate the initial model as an equivalent set function optimization problem under a matroid base constraint.

Additionally, several works have explored other tensor decomposition structures. [47] proposed Neural Powered Tucker Network (`NePTuNe`) to specifically resolve the overfitting in link prediction problem. In extension to Neural Tensor Network (`NTN`), `NePTuNe` utilized a shared core tensor to solve the potential overfitting problem. [42] proposed Probabilistic Neural-kernel tensor Decomposition (`POND`) that uses Gaussian processes (GPs) to model the hidden relationships to automatically detect their complexity in tensors, preventing both underfitting and overfitting, and then incorporates CNN to construct the GP kernel to greatly promote the capability of estimating highly nonlinear relationships. [53] proposed a novel tensor-ring (TR) decomposition method based on the convolutional computation (`ConvTR`), which can be regarded as a natural extension of deep learning models for the LRTC problem. Specifically, `ConvTR` employs the multi-layer CNN to model the complex interactions between TR factors. Each element in the index vector of the observation tensor can be embedded as a corresponding tensor slice in the factor tensor decomposed by the TR model. CoATR [30] couples a tensor-ring (TR) decomposition for global low-rank structure with a CNN-based nonlinear mapping and a learned autoregressive (AR) module for local temporal consistency, yielding a generalized TR model that replaces matrix products with convolutional operators. [45] presented `DAIN`, a general data augmentation framework that enhances the prediction accuracy of neural tensor completion methods. As a framework, `DAIN` is compatible with multiple tensor decomposition methods and multiple neural networks such as MLP, CNN, RNN, to recover and augment data.

4.3. Other nonlinear deep TD models

While most nonlinear deep TD models are built on MLP and CNN structures, a growing number of studies explore the integration of TD with alternative neural network architectures such as auto-encoder (AE), general adversarial network (GAN), graph neural network (GNN), etc. These models demonstrate the flexibility of deep TD and extend its applicability to more diverse and complex data modeling tasks. For example, [46] developed a modular deep TD, structured into four blocks for large-scale datasets. The model begins with an encoder block to reduce dimensionality, followed by tensor train decomposition combined with feature projection to extract compact latent representations. A final prediction block is then applied to perform forecasting tasks. Specifically, the use of cross-approximation within the tensor train decomposition enables efficient approximation of high-dimensional tensors. [49] introduced a novel 3D shape completion model by integrating tensor rank decomposition and the hierarchical variational autoencoder (VAE) to estimate the canonical factors. Their results showed that the developed approach outperforms previous methods in both fidelity and diversity metrics, when the input consists of the bottom half of the point cloud. [51] designed a novel deep Adversarial Tensor Completion (`DATC`) scheme based on DL techniques. `DATC` is the first scheme that exploits the data reconstruction ability of autoencoder and the power of adversarial training from Generative Adversarial Networks to infer the missing data. Furthermore, [52] introduced a graph-guided neural tensor decomposition (`GNTD`) model for reconstructing whole spatial transcriptomes in tissues. `GNTD` employs a hierarchical tensor structure and formulation to explicitly model the high-order spatial gene expression data with a hierarchical nonlinear decomposition in a three-layer neural network, enhanced by spatial relations among the capture spots and gene functional relations for accurate reconstruction from highly sparse spatial profiling data.

These models exemplify the potential of deep TD when integrated with advanced neural architectures beyond MLPs and CNNs. They also highlight the adaptability of TD frameworks in addressing challenging tasks such as shape reconstruction, clinical forecasting, adversarial imputation, and biological data recovery. Future research may further explore such hybrid approaches, combining TD with transformers, diffusion models, and other emerging architectures.

5. Track III: deep learning training schemes for TD

In this section, we further discuss the popular training schemes in deep learning being applied to TD to overcome innate limitations of original TD, including self-supervised learning, transfer learning, and multi-task learning (Fig. 5).

5.1. TD with self-supervised learning

Traditional TD is trained to minimize the reconstruction loss. As stated in [59], training via reconstruction loss ensures that the

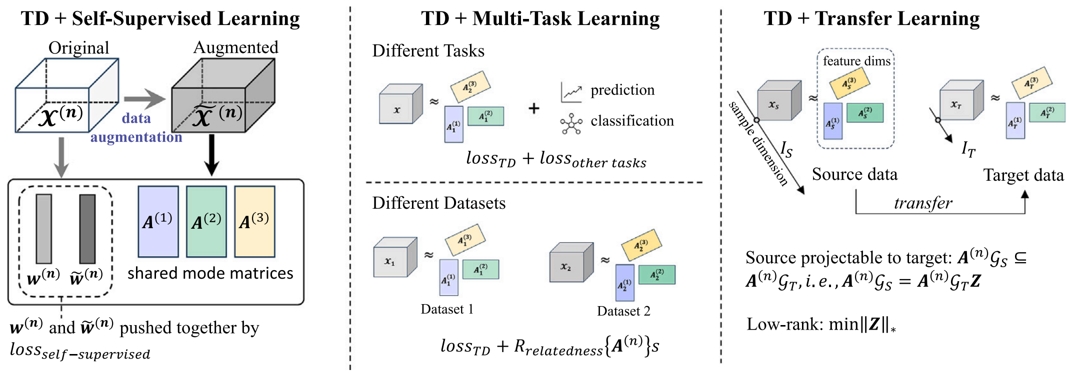


Fig. 5. Deep learning training schemes with tensor decomposition.

"reconstructed values" are close to the "ground truth values"; however, when the data have class structure, e.g., the four types of movement in human activity dataset [60], the decomposed components do not effectively represent and align the class features when the training is only supervised by the reconstruction loss.

Researchers [59] proposed using self-supervised learning, especially contrastive learning [61] to enhance the classification discrimination power of TD models. Contrastive learning is a training scheme originated in the computer vision community for more robust classification against noise and other perturbations [61]. It has three key steps: (1) data augmentation, which generates augmented images from the original one via various augmentation techniques; (2) positive and negative pairs: the child images (generated by various augmentation techniques) from the same parent image are usually positive pairs usually, otherwise, negative pairs; (3) contrastive loss, which puts positive pairs closer to each other, and pushes negative pairs away from each other. As such, the trained classification model will boost its robustness against perturbations. ATD [59] and PCL [62] adopted similar training scheme for TD: firstly, the original tensor $\mathcal{X}$ is augmented to a second tensor $\tilde{\mathcal{X}}$ based on adding noise, jittering, bandpass and so on. Then, both tensors go through the TD model: given the n-th sample $\mathcal{X}^{(n)}$ and its augmented counterpart $\tilde{\mathcal{X}}^{(n)}$, both are decomposed, with share factor matrices and their unique weight vectors, representing this n-th sample's original feature and the one after augmentation,

$$\mathcal{X}^{(n)} = [[\mathbf{w}^{(n)}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(K)}]],$$
$$\tilde{\mathcal{X}}^{(n)} = [[\tilde{\mathbf{w}}^{(n)}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(K)}]] \quad (8)$$

In ATD, the n-th sample's original feature $\mathbf{w}^{(n)}$ and the augmented one $\tilde{\mathbf{w}}^{(n)}$ are considered to be positive pairs. Apart from the regular reconstruction loss, a self-supervision loss is also added,

$$\ell_{self-supervised} = \frac{1}{N}\,\mathrm{Tr}(\mathbf{W}^{\mathrm{T}}\mathbf{\Lambda}(\mathbf{W})\mathbf{I}\mathbf{\Lambda}(\widetilde{\mathbf{W}})\widetilde{\mathbf{W}}), \quad (9)$$

where $\mathbf{\Lambda}(\mathbf{W}) = \mathrm{diag}\left(1/\|\mathbf{w}^{(1)}\|_2, \dots, 1/\|\mathbf{w}^{(N)}\|_2\right)$ is the row-wise scaling matrix, and $Tr(\cdot)$ is the trace of a matrix. Such a self-supervision loss is designed to maximize the similarity of the $\langle \frac{\mathbf{w}^{(n)}}{\|\mathbf{w}^{(n)}\|_2}, \frac{\tilde{\mathbf{w}}^{(n)}}{\|\tilde{\mathbf{w}}^{(n)}\|_2} \rangle$.

However, PCL further claimed that the ATD still only treats the same sample, e.g., the n-th sample's original feature and augmented feature as positive pair to each other, which basically only boosts the decomposition model's robustness to the perturbation, not actually helping the model to recognize the sample i and j are from the same class. Thus, PCL [62] uses the features $\{\mathbf{w}\}$s and $\{\tilde{\mathbf{w}}\}$s to construct a pseudo graph, and updates the self-supervised loss to achieve better class-awareness, simply by replacing the $\mathbf{I}$ to the graph Laplacian matrix $\mathbf{L}$,

$$\ell_{class-awareness} = \frac{1}{N}\,\mathrm{Tr}(\mathbf{W}^{\mathrm{T}}\mathbf{\Lambda}(\mathbf{W})\mathbf{L}\mathbf{\Lambda}(\widetilde{\mathbf{W}})\widetilde{\mathbf{W}}) \quad (10)$$

5.2. TD with multi-task learning

Multi-task learning is designed to learn a model with multiple tasks completed at the same time [63]: It aims to leverage useful information contained in multiple related tasks to help improve the generalization performance of all the tasks. In TD, multi-task learning is interpreted as two kinds: (1) decomposition, prediction, anomaly detection, and so on are considered as different tasks, or (2) different samples or datasets are considered as different tasks. For the first type, multi-task learning is simply achieved by combining multiple losses together [64],

$$\min loss_{TD} + loss_{othertasks} \quad (11)$$

For example, to learn both the spatiotemporal pattern and decomposition, [65] simultaneously performed decomposition (splitting spatiotemporal data into spatial and temporal matrices) and prediction (using these matrices). Consequently, a prediction loss was added as $\frac{1}{2}\sum_s \sum_t (\mathbf{x}_{s,t}^T(\mathbf{W}^T\mathbf{A}_s^{spatial} + \mathbf{V}^T\mathbf{A}t^{temporal}) - ys,t)$, where $\mathbf{W}$ and $\mathbf{V}$ are the spatial and temporal prediction coefficients. To achieve both subspace learning, anomaly detection, and decomposition, [64] combined the sparse coding loss, i.e., $\|\mathcal{X} - \mathcal{S} - \mathcal{G}\times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)}\|$, as well as the self-expression term in subspace learning, i.e., $\|\mathcal{G} - \mathcal{G}\times_1 \mathbf{Z}\| + \frac{1}{2}\|\mathbf{Z}\|_F^2$, where $\mathcal{S}$ is the anomaly, and $\mathbf{Z}$ is the self-expression matrix. To classify the EEG signals better, [66] also added the loss of minimizing the within-class variation and maximizing the between-class boundary together with the tensor decomposition loss.

For the second type, where different related samples or datasets are considered as different tasks, the intuition is to use a regularization term [67,68] to encourage such a shared relatedness across different samples or tasks, such as shared progressive patterns of different patients, shared temporal smoothness of traffic flow of different stations [5]. This can be generally formulated as,

$$\min loss_{TD} + R_{relatedness}(\{\mathbf{A}^{(n)}\}s) \quad (12)$$

where $R_{relatedness}(\{\mathbf{A}^{(n)}\}s)$ is the regularization added on the decomposed factor matrices to achieve relatedness across different samples. For example, one of the most popular regularizations for controlling the shared relatedness across different samples is the low-rank regularization, which can be formulated by L_1 norm [5], nuclear or trace norm [67,68], generalized trace norm [69] and so on. Usually, the regularization is designed specifically according to the application domain and purposes, and we recommend that readers who are interested refer to [63] for various design choices of regularizations.

5.3. TD with transfer learning

Transfer learning is designed to transfer the knowledge from the source data or tasks to the target data or tasks [70]. Tensor transfer learning, similarly, assumes that the source tensor and target tensor share transferable knowledge in some subspaces [71]. Taking four-dimensional tensors as example, the source tensor $\mathcal{X}_S \in \mathbb{R}^{(I_1 \times I_2 \times I_3) \times I_S}$ and the target tensor $\mathcal{X}_T \in \mathbb{R}^{(I_1 \times I_2 \times I_3) \times I_T}$ share the same feature dimensions $\{I_1, I_2, I_3\}$, but have their own sample dimensions I_T, I_S. When conducting the transfer of $\mathcal{X}_S \rightarrow \mathcal{X}_T$, it is usually assumed that,

- Both target and source tensor are projected in the same space on the feature dimension.
- The low-rank representation can be used to format using the target representing the source.

For example, [71] formulated such a tensor transfer learning based on Tucker decomposition: (1) Both target and source tensor are projected on the feature dimension, i.e., $\mathbf{A}^{(n)}\mathcal{G}_S \subseteq \mathbf{A}^{(n)}\mathcal{G}_T$; (2) Low-rank representation can be achieved by regularization. The formulation is given as,

$$\begin{aligned} &\min_{\mathbf{Z}} \|\mathbf{Z}\|_* \\ &s.t.\ \mathbf{A}^{(n)}\mathcal{G}_S = \mathbf{A}^{(n)}\mathcal{G}_T\mathbf{Z}, \\ &\mathbf{A}^{(n)T}\mathbf{A}^{(n)} = \mathbf{I}, n = 1, 2, 3, \end{aligned} \tag{13}$$

where $\mathbf{Z}$ is the projection matrix mapping from the target tensor to the source tensor.

Disclaimer: Methods that only use TD as a way to decompose the tensor of parameters and transfer learning is totally independently designed will not be covered [72–75].

6. Applications of deep TD models

Deep tensor decomposition (Deep TD) models have been applied across a wide range of domains due to their ability to model complex, nonlinear relationships within data. This section reviews their applications in five major areas: transportation, healthcare, computer vision, web and social media, and other emerging fields. Within these domains, deep TD methods support a variety of tasks including data recovery, feature extraction, classification, and prediction, as summarized in Table 3.

6.1. Domain-Specific Applications

Transportation. Deep TD models have been successfully applied to spatiotemporal data imputation and traffic forecasting. Approaches such as `LightNestle` [50] and `JULIA` [24] have demonstrated strong performance in recovering missing traffic flow and mobility data, leveraging multilayer perceptrons (MLPs) and attention mechanisms within the TD framework.

Healthcare. In healthcare systems, deep TD has been applied for reconstructing clinical measurements and modeling complex patient records. Studies such as [25,46], and [55] show that deep TD models can recover patient health records accurately. Beyond recovery, deep TD models have also been applied to diagnostic tasks; for example, works including [16,19,78], and [79], leverage deep TD for phenotype classification and disease prediction.

Computer vision. In visual data analysis, deep TD techniques have been used for data recovery (e.g., occluded image restoration) and object and scene classification. For example, [48] applies deep TD to capture spatial correlations in image data, improving recognition accuracy in complex scenes.

Web and social media. Deep TD has proven valuable in modeling social networks, temporal behavior prediction, and so on. Models such as [21,40] exploit deep TD to learn interactions among users, content, and time, enabling applications ranging from personalized recommendations to knowledge graph completion.

Other domains. Emerging areas such as bioinformatics and genomics have also benefited from deep TD models. For example, deep TD models have been used to analyze protein interactions and drug-target-disease relationships [22,43] by extracting nonlinear features from high-dimensional biological tensors.

6.2. Guidance for deep TD model selection in practice

Beyond cataloging application scenarios, it is important to consider how the characteristics of domain data can influence the choice of deep TD methods in practice. For example, in healthcare and recommendation systems, where data is often sparse and noisy, deep TD models that incorporate explicit regularization are better suited to handle missing values and uncertainty. In contrast, spatiotemporal domains such as transportation and sensor networks benefit from architectures that explicitly capture temporal dependencies, such as attention-augmented or recurrent deep TD models. For high-dimensional image and video data, CNN-based deep TD models are more effective in capturing local structures while maintaining scalability. Finally, when relational or graph structure is intrinsic, as in social networks or bioinformatics, GNN-based deep TD models may provide stronger performance. Although developing a comprehensive framework for model selection lies beyond the scope of this survey, these examples illustrate how practitioners can align data characteristics with appropriate deep TD designs in real-world scenarios.

6.3. Commonly used datasets

To support these diverse applications, a wide variety of datasets have been commonly used in the development and assessment of deep TD models in the literature. Here we provide an overview of the key datasets.

- **Text data**: WordNet,[1] Reddit,[2] NeurIPS Publication Data,[3] Enron Emails,[4] and Github Archive Data.[5]
- **Image data**: MNIST [6] Moving MNIST,[7] CIFAR-10,[8] Weizmann Human Action,[9] and Yale-B Dataset.[10]

Table 3
Applications of deep TD methods in different areas according to various tasks.

Task	Area				
	Transportation	Healthcare	Computer vision	Web & social media	Other domains
Data recovery	[50], [24] [54] [51], [20], [53], [27]	[46], [55], [25], [22] [24], [52], [48]	[49], [55], [48], [20], [53]	[24], [45], [54]	[24], [20], [76], [77]
Feature extraction	[51], [30]	[22]	[42]	[42]	[55], [43]
Classification		[16], [78], [19], [79]	[48]		
Prediction		[46], [41]		[21], [47]. [40]	[40], [20]

1 https://www.wordnet.princeton.edu/
2 https://www.reddit.com/r/datasets/
3 https://www.papers.nips.cc/
4 https://www.cs.cmu.edu/~enron/
5 https://www.gharchive.org/
6 MNIST http://www.yann.lecun.com/exdb/mnist/,
7 https://www.web.cs.toronto.edu/
8 https://www.cs.toronto.edu/
9 https://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html
10 https://www.vision.ucsd.edu/datasets/extended-yale-face-database-b-b

- **Other data**: domain-specific datasets in healthcare (e.g., DrugBank[11] and UniProt [12]) and recommendation systems (e.g. MovieLens,[13] Gowalla,[14] and Foursquare[15]).

In summary, the expanding diversity of applications demonstrates both the effectiveness and adaptability of deep tensor decomposition models. From sensor networks recovery and personalized recommendation to electronic health records modeling and genomic profile classification, deep TD methods have emerged as essential tools for tackling complex and multi-aspect data. Future research may further extend their impact to emerging domains such as natural language understanding, large-scale scientific simulations, and personalized education.

7. Challenges and future directions

Deep tensor decomposition (deep TD) is an emerging research topic situated at the intersection of low-rank tensor approximation and deep learning. It introduces either multi-layered decomposition schemes or integrates nonlinear transformations through deep learning architectures. These approaches significantly enhance the expressiveness and performance of traditional TD models. In this survey, we have provided a comprehensive review of deep TD, covering both linear and nonlinear deep TD models and their variants, as well as several training schemes that integrate deep learning techniques. We also summarized a broad range of applications of deep TD models spanning topic modeling, recommendation systems, image processing, and healthcare. By consolidating the most recent advances and practical implementations, this work offers a thorough understanding of the state-of-the-art in deep TD research. Despite substantial progress, certain limitations and challenges remain. This section outlines key challenges from multiple perspectives, including initialization, parameter selection, computational complexity, interpretability, and software support, thereby identifying potential directions for future research.

Initialization. The performance of deep TD models is highly sensitive to initialization. Poor initialization may result in suboptimal factorization, slow convergence, or convergence to local minima. However, this survey reveals that initialization techniques for deep TD models are not thoroughly discussed or developed. Future research should investigate more robust initialization schemes, potentially incorporating prior knowledge or data-driven heuristics.

Parameter selection. Currently, there are no established systematic strategies for determining the key parameters of deep TD models, such as the number of decomposition layers, inner ranks, as well as the regularization parameters. Thus, establishing proper guidelines for parameter selection is a crucial direction for future work.

Computational complexity and optimization strategies. Most deep TD models are currently solved by a two-stage process that includes a fine-tuning phase, resulting in high computational costs compared to traditional TD methods. Besides, as the dimensionality of tensor data increases, the computational complexity of deep TD rises significantly. This creates a clear trade-off for practitioners, i.e., when data exhibits strong nonlinear interactions that cannot be captured by shallow or linear TD, deep TD methods may justify their higher costs by delivering superior accuracy and representational power. In addition, systematic optimization strategies for deep TD remain underexplored. Therefore, developing resource-aware training algorithms, lightweight model designs, and hardware-optimized implementations for deep TD is essential to make the models more broadly applicable in practice.

Interpretability. One of the most persistent challenges in deep TD is the lack of "ground-truth" baselines. The hierarchical or nonlinear structure of deep TD models makes it difficult to trace or understand the role of individual components and their interactions across layers. Moreover, the lack of standardized benchmarks or ground-truth decompositions hinders fair evaluation. Advancing explainable deep TD methods through layer-wise attribution, disentanglement metrics, or interpretable training objectives remains a promising research frontier.

Software and toolboxes. At present, no standardized software library or toolbox exists for deep TD models. This absence restricts experimentation, reproducibility, and wider adoption by a broader research community. Developing an open-source toolbox (e.g., in Python or MATLAB) that integrates basic models, training routines, and visualization tools would significantly accelerate progress in this field.

The above challenges highlight promising opportunities for future research in deep TD techniques. By utilizing the hierarchical feature extraction capabilities of deep learning with the structural interpretability of low-rank tensor methods, deep TD offers a unique foundation for building more transparent, scalable, and generalizable machine learning models. Continued research in this area is essential to fully realize its potential across diverse scientific and applied domains.

Funding sources

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

CRediT authorship contribution statement

Meng Zhao: Writing – original draft, Investigation. **Jiuyun Hu:** Writing – original draft. **Ziyue Li:** Writing – original draft, Investigation. **Zekai Wang:** Writing – original draft. **Lijun Sun:** Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

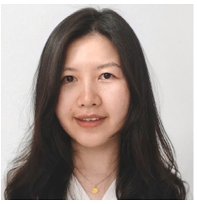
[1] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, H.A. Phan, Tensor decompositions for signal processing applications: from two-way to multiway component analysis, IEEE Signal Process. Mag. 32 (2015) 145–163.
[2] N.D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E.E. Papalexakis, C. Faloutsos, Tensor decomposition for signal processing and machine learning, IEEE Trans. Signal Process. 65 (2017) 3551–3582.
[3] M. Zhao, M.R. Gahrooei, N. Gaw, Robust coupled tensor decomposition and feature extraction for multimodal medical data, IISE Trans. Healthc. Syst. Eng. 13 (2023) 117–131.
[4] M. Zhao, M.R. Gahrooei, Fedpar: federated parafac2 tensor factorization for computational phenotyping, IISE Trans. Healthc. Syst. Eng. (2024) 1–12.
[5] Z. Li, N.D. Sergin, H. Yan, C. Zhang, F. Tsung, Tensor completion for weakly-dependent data on graph for metro passenger flow prediction, in: Proceedings of the AAAI conference on artificial intelligence, vol. 34, 2020, pp. 4804–4810.
[6] M. Zhao, M.R. Gahrooei, M. Ilbeigi, Change detection in partially observed large-scale traffic network data, IEEE Trans. Intell. Transp. Syst. (2024).
[7] R.A. Harshman, et al., Foundations of the parafac procedure: models and conditions for an "explanatory" multi-modal factor analysis, UCLA Work. Pap. Phon. 16 (1970) 84.
[8] L.R. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31 (1966) 279–311.
[9] A. Cichocki, R. Zdunek, S.-I. Amari, Hierarchical als algorithms for nonnegative matrix and 3d tensor factorization, in: International conference on independent component analysis and signal separation, Springer, 2007, pp. 169–176.
[10] W. Lu, F.-L. Chung, W. Jiang, M. Ester, W. Liu, A deep bayesian tensor-based system for video recommendation, ACM Trans. Inf. Syst. 37 (2018) 1–22.
[11] S. Zhe, K. Zhang, P. Wang, K.-C. Lee, Z. Xu, Y. Qi, Z. Ghahramani, Distributed flexible nonlinear tensor factorization, Adv. Neural Inf. Process. Syst. 29 (2016).
[12] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444.

[11] https://www.go.drugbank.com/releases/latest
[12] https://www.uniprot.org/
[13] https://www.grouplens.org/datasets/movielens/
[14] https://www.snap.stanford.edu/data/loc-gowalla.html
[15] https://www.sites.google.com/site/yangdingqi/home/foursquare-dataset

[13] L. Deng, J. Platt, Ensemble deep learning for speech recognition, in: Proc. interspeech, 2014.
[14] R. Chauhan, K.K. Ghanshala, R. Joshi, Convolutional neural network (cnn) for image detection and recognition, in: 2018 first international conference on secure cyber computing and communication (ICSCCC), IEEE, 2018, pp. 278–282.
[15] R. Miotto, F. Wang, S. Wang, X. Jiang, J.T. Dudley, Deep learning for healthcare: review, opportunities and challenges, Brief. Bioinform. 19 (2018) 1236–1246.
[16] I. Perros, R. Chen, R. Vuduc, J. Sun, Sparse hierarchical tucker factorization and its application to healthcare, in: 2015 IEEE International Conference on Data Mining, IEEE, 2015, pp. 943–948.
[17] J. Vendrow, J. Haddock, D. Needell, Neural nonnegative cp decomposition for hierarchical tensor analysis, in: 2021 55th Asilomar Conference on Signals, Systems, and Computers, IEEE, 2021, pp. 1340–1347.
[18] J. Vendrow, J. Haddock, D. Needell, A generalized hierarchical nonnegative tensor decomposition, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2022, pp. 4473–4477.
[19] Q. Tan, P. Yang, G. Wen, Deep non-negative tensor factorization with multi-way emg data, Neural Comput. Appl. (2022) 1–11.
[20] H. Liu, Y. Li, M. Tsang, Y. Liu, Costco: a neural tensor completion model for sparse tensors, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 324–334.
[21] X. Wu, B. Shi, Y. Dong, C. Huang, N.V. Chawla, Neural tensor factorization for temporal interaction learning, in: Proceedings of the Twelfth ACM international conference on web search and data mining, 2019, pp. 537–545.
[22] H. Chen, J. Li, Learning data-driven drug-target-disease interaction via neural tensor network, in: International joint conference on artificial intelligence (IJCAI), 2020.
[23] K. Xie, H. Lu, X. Wang, G. Xie, Y. Ding, D. Xie, J. Wen, D. Zhang, Neural tensor completion for accurate network monitoring, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 1688–1697.
[24] C. Qian, K. Huang, L. Glass, R.S. Srinivasa, J. Sun, Julia: joint multi-linear and nonlinear identification for tensor completion, arXiv preprint arXiv:2202.00071, (2022).
[25] J. Fan, Multi-mode deep matrix and tensor factorization, in: International conference on learning representations, 2021.
[26] Y. Luo, X.-L. Zhao, D. Meng, T.-X. Jiang, Hlrtf: hierarchical low-rank tensor factorization for inverse problems in multi-dimensional imaging, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 19303–19312.
[27] W. Luo, Y. Hou, P. Tang, Neural canonical polyadic factorization for traffic analysis, arXiv preprint arXiv:2506.15079, (2025).
[28] Y. Hou, P. Tang, Multi-head self-attending neural tucker factorization, arXiv preprint arXiv:2501.09776, (2025).
[29] P. Tang, X. Luo, J. Woodcock, Auto-encoding neural tucker factorization, IEEE Trans. Knowl. Data Eng. (2025).
[30] T. Liao, L. Zhang, J. Yang, C. Chen, Z. Zheng, Coatr: a convolutional autoregressive tensor-ring decomposition method for sparse spatio-temporal traffic data, Neurocomputing 617 (2025) 129006.
[31] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, SIAM Review 51 (2009) 455–500.
[32] I.V. Oseledets, Tensor-train decomposition, SIAM J. Sci. Comput. 33 (2011) 2295–2317.
[33] Q. Zhao, G. Zhou, S. Xie, L. Zhang, A. Cichocki, Tensor ring decomposition, arXiv preprint arXiv:1606.05535, (2016).
[34] T. Wu, B. Gao, W.L. Woo, Hierarchical low-rank and sparse tensor micro defects decomposition by electromagnetic thermography imaging system, Philos. Trans. R. Soc. A 378 (2020) 20190584.
[35] K. Xie, C. Liu, X. Wang, X. Li, G. Xie, J. Wen, K. Li, Neural network compression based on tensor ring decomposition, IEEE Trans. Neural Netw. Learn. Syst. 36 (2024) 5388–5402.
[36] X. Ma, P. Zhang, S. Zhang, N. Duan, Y. Hou, M. Zhou, D. Song, A tensorized transformer for language modeling, Adv. Neural Inf. Process. Syst. 32 (2019).
[37] P. Zhen, Z. Gao, T. Hou, Y. Cheng, H.-B. Chen, Deeply tensor compressed transformers for end-to-end object detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, 2022, pp. 4716–4724.
[38] M. Bai, D. Zhou, Q. Zhao, Tendiffpure: a convolutional tensor-train denoising diffusion model for purification, Front. Inf. Technol. Electron. Eng. 25 (2024) 160–169.
[39] Z. Wang, S. Fang, S. Li, S. Zhe, Dynamic tensor decomposition via neural diffusion-reaction processes, Adv. Neural Inf. Process. Syst. 36 (2023) 23453–23467.
[40] L. He, B. Liu, G. Li, Y. Sheng, Y. Wang, Z. Xu, Knowledge base completion by variational bayesian neural tensor decomposition, Cogn. Comput. 10 (2018) 1075–1084.
[41] B. Liu, L. He, Y. Li, S. Zhe, Z. Xu, Neuralcp: bayesian multiway data analysis with neural tensor decomposition, Cogn. Comput. 10 (2018) 1051–1061.
[42] C. Tillinghast, S. Fang, K. Zhang, S. Zhe, Probabilistic neural-kernel tensor decomposition, in: 2020 IEEE International Conference on Data Mining (ICDM), IEEE, 2020, pp. 531–540.
[43] H. Chen, J. Li, Neural tensor model for learning multi-aspect factors in recommender systems, in: International Joint Conference on Artificial Intelligence (IJCAI), vol. 2020, 2020.
[44] J. Schreiber, T. Durham, J. Bilmes, W.S. Noble, Avocado: a multi-scale deep tensor factorization method learns a latent representation of the human epigenome, Genome Biol. 21 (2020) 1–18.
[45] S. Oh, S. Kim, R.A. Rossi, S. Kumar, Influence-guided data augmentation for neural tensor completion, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 1386–1395.
[46] M. Usvyatsov, A. Makarova, R. Ballester-Ripoll, M. Rakhuba, A. Krause, K. Schindler, Cherry-picking gradients: learning low-rank embeddings of visual data via differentiable cross-approximation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 11426–11435.
[47] S. Sonkar, A. Katiyar, R. Baraniuk, Neptune: neural powered tucker networkfor knowledge graph completion, in: Proceedings of the 10th International Joint Conference on Knowledge Graphs, 2021, pp. 177–180.
[48] V. Saragadam, R. Balestriero, A. Veeraraghavan, R.G. Baraniuk, Deeptensor: low-rank tensor decomposition with deep network priors, arXiv preprint arXiv:2204.03145, (2022).
[49] W. Jiang, K. Daniilidis, Probabilistic shape completion by estimating canonical factors with hierarchical vae, arXiv preprint arXiv:2212.03370, (2022).
[50] Y. Li, W. Liang, K. Xie, D. Zhang, S. Xie, K. Li, Lightnestle: quick and accurate neural sequential tensor completion via meta learning, in: IEEE INFOCOM 2023-IEEE Conference on Computer Communications, IEEE, 2023, pp. 1–10.
[51] K. Xie, Y. Ouyang, X. Wang, G. Xie, K. Li, W. Liang, J. Cao, J. Wen, Deep adversarial tensor completion for accurate network traffic measurement, IEEE/ACM Trans. Netw. 31 (2023) 2101–2116.
[52] T. Song, C. Broadbent, R. Kuang, Gntd: reconstructing spatial transcriptomes with graph-guided neural tensor decomposition informed by spatial and functional relations, Nat. Commun. 14 (2023) 8276.
[53] T. Liao, J. Yang, C. Chen, Z. Zheng, A neural tensor decomposition model for high-order sparse data recovery, Inf. Sci. 658 (2024) 120024.
[54] C. Liu, K. Xie, T. Wu, C. Ma, T. Ma, Distributed neural tensor completion for network monitoring data recovery, Inf. Sci. 662 (2024) 120259.
[55] Z. Tao, T. Tanaka, Q. Zhao, Nonparametric tensor ring decomposition with scalable amortized inference, Neural Networks 169 (2024) 431–441.
[56] D. Ahn, U.S. Saini, E.E. Papalexakis, A. Payani, Neural additive tensor decomposition for sparse tensors, in: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, 2024, pp. 14–23.
[57] X. Xu, M. Lin, Z. Xu, X. Luo, Attention-mechanism-based neural latent-factorization-of-tensors model, ACM Trans. Knowl. Discov. Data 19 (2025) 1–27.
[58] Y. Liu, J. Luo, H. Wu, Mirna-disease associations prediction based on neural tensor decomposition, in: Intelligent Computing Theories and Application: 17th International Conference, ICIC 2021, Shenzhen, China, August 12–15, 2021, Proceedings, Part III 17, Springer, 2021, pp. 312–323.
[59] C. Yang, C. Qian, N. Singh, C.D. Xiao, M. Westover, E. Solomonik, J. Sun, Atd: augmenting cp tensor decomposition by self supervision, Adv. Neural Inf. Process. Syst. 35 (2022) 32039–32052.
[60] D. Anguita, A. Ghio, L. Oneto, X. Parra, J.L. Reyes-Ortiz, A public domain dataset for human activity recognition using smartphones, in: Esann, vol. 3, PMLR, Menlo Park, Calif, 2013, pp. 3.
[61] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: International conference on machine learning, PMLR, 2020, pp. 1597–1607.
[62] M. Li, Z. Li, L. Sun, F. Tsung, Enabling tensor decomposition for time-series classification via a simple pseudo-laplacian contrast, arXiv preprint arXiv:2409.15200, (2024).
[63] Y. Zhang, Q. Yang, A survey on multi-task learning, IEEE Trans. Knowl. Data Eng. 34 (2021) 5586–5609.
[64] N.D. Sergin, J. Hu, Z. Li, C. Zhang, F. Tsung, H. Yan, Low-rank robust subspace tensor clustering for metro passenger flow modeling, INFORMS J. Data Sci. (2024).
[65] J. Xu, J. Zhou, P.-N. Tan, X. Liu, L. Luo, Wisdom: weighted incremental spatio-temporal multi-task learning via tensor decomposition, in: 2016 IEEE International Conference on Big Data (Big Data), IEEE, 2016, pp. 522–531.
[66] Q. Zheng, Y. Wang, P.A. Heng, Multitask feature learning meets robust tensor decomposition for eeg classification, IEEE Trans. Cybern. 51 (2019) 2242–2252.
[67] K. Wimalawarne, M. Sugiyama, R. Tomioka, Multitask learning meets tensor factorization: task imputation via convex optimization, Adv. Neural Inf. Process. Syst. 27 (2014).
[68] X. Zhang, J. Wu, M.K. Ng, Multilinear multitask learning by transformed tensor singular value decomposition, Mach. Learn. Appl. 13 (2023) 100479.
[69] Y. Zhang, Y. Zhang, W. Wang, Multi-task learning via generalized tensor trace norm, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 2254–2262.
[70] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (2009) 1345–1359.
[71] C. Jia, Z. Ding, Y. Kong, Y. Fu, Semi-supervised cross-modality action recognition by latent tensor transfer learning, IEEE Trans. Circuits Syst. Video Technol. 30 (2019) 2801–2814.
[72] W. Mao, Z. Chen, Y. Zhang, X. Liang, Tensor-daad: when tensor meets online early fault detection with transfer learning, Measurement 208 (2023) 112478.
[73] V. Bishnoi, N. Goel, Tensor-rt-based transfer learning model for lung cancer classification, J. Digit. Imaging 36 (2023) 1364–1375.
[74] L. Karlsson, E. Fallenius, C. Bergeling, B. Bernhardsson, Tensor decomposition of eeg signals for transfer learning applications, Brain-Computer Interfaces 11 (2024) 178–192.
[75] C. Dai, X. Liu, Z. Li, M.-Y. Chen, A tucker decomposition based knowledge distillation for intelligent edge applications, Appl. Soft Comput. 101 (2021) 107051.
[76] X. Luo, H. Wu, Z. Li, Neulft: a novel approach to nonlinear canonical polyadic decomposition on high-dimensional incomplete tensors, IEEE Trans. Knowl. Data Eng. 35 (2022) 6148–6166.

[77] X. Liao, H. Wu, X. Luo, A novel tensor causal convolution network model for highly-accurate representation to spatio-temporal data, IEEE Trans. Autom. Sci. Eng. (2025).
[78] J. Luo, Z. Lai, C. Shen, P. Liu, H. Shi, Graph attention mechanism-based deep tensor factorization for predicting disease-associated mirna-mirna pairs, in: 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2021, pp. 189–196.
[79] S. You, Z. Lai, J. Luo, Multi-source data-based deep tensor factorization for predicting disease-associated mirna combinations, in: International Conference on Intelligent Computing, Springer, 2022, pp. 807–821.

Author biography



Meng Zhao is an Assistant Professor of the Department of Industrial and Systems Engineering at Lehigh University. She received the Ph.D. degree in the Department of Industrial and Systems Engineering, the University of Florida, Gainesville, Florida, USA. Her research interests focus on high-dimensional data modeling, tensor analysis, machine learning, and data mining techniques. She is a member of the Institute for Operations Research and the Management Science (INFORMS) and the Institute of Industrial and Systems Engineers (IISE).

Jiuyun Hu received his PhD degree at the School of Computing and Augmented Intelligence at Arizona State University. His research focuses on (1) High-dimensional tensor data analysis with specific interest in domain graph structure incorporation and personalization; (2) Statistical modeling and machine learning on signal monitoring and threat detection; (3) Large data mining and feature extraction. He is a member of the Institute for Operations Research and the Management Science (INFORMS) and the Institute of Industrial and Systems Engineers (IISE).

Ziyue Li is W2 professor in the Department of Operations and Technology and Heilbronn Data Science Center, Technical University of Munich. His research focuses on spatiotemporal machine learning models for smart cities. His works have been published as 40 + peer-reviewed papers in top-tier AI conferences (IJCAI, AAAI, KDD, ICLR, NeurIPS), 10 + papers in top journals (TKDE, TITS, TMM), awarded with 10 + best paper awards. His research focuses on spatiotemporal analysis, high-dimensional data modeling and tensor analytics.

Zekai Wang is an Assistant Professor in the Department of Analytics at the Charles F. Dolan School of Business, Fairfield University. Before joining Fairfield, he earned his Ph.D. in Industrial and Systems Engineering from the University of Tennessee, Knoxville. His research focuses on advanced data analytics, deep learning, and health informatics. He is a member of the Institute for Operations Research and the Management Science (INFORMS) and the Institute of Industrial and Systems Engineers (IISE).

Lijun Sun is an Associate Professor in the Department of Civil Engineering at McGill University. He received his PhD degree in Civil Engineering (Transportation) from National University of Singapore. His current research centers on the area of urban computing and smart transportation, developing innovative methodologies and applications to address efficiency, resilience, and sustainability issues in urban transportation systems. His work has been featured in popular media outlets, including Wired, Citylab, Scientific American, and MIT Technology Review.