Review

Not peer-reviewed version

Theoretical Foundations and Practical Applications in Signal Processing and Machine Learning

Petar Slavka and Aliona Tatyana *

Posted Date: 3 April 2025

doi: 10.20944/preprints202504.0276.v1

Keywords: tensor decomposition; multi-way data analysis; Canonical Polyadic (CP) decomposition; Tucker decomposition; Tensor Train (TT); Tensor Ring (TR); signal processing; machine learning; low-rank approximation; computational complexity; feature extraction; deep learning; robustness; noise handling; rank estimation; optimization algorithms

 doi:10.20944/preprints202504.0276.v1

Article

Theoretical Foundations and Practical Applications in Signal Processing and Machine Learning

Petar Slaveka and Aliona Tatyana *

Lomonosov Moscow State University, Russia
* Correspondence: aliona.tatyana@rector.msu.ru

Abstract: Tensor decomposition has emerged as a powerful mathematical framework for analyzing multi-dimensional data, extending classical matrix decomposition techniques to higher-order representations. As modern applications generate increasingly complex datasets with multi-way relationships, tensor methods provide a principled approach to uncovering latent structures, reducing dimensionality, and improving computational efficiency. This paper presents a comprehensive review of tensor decomposition techniques, their theoretical foundations, and their applications in signal processing and machine learning. We begin by introducing the fundamental concepts of tensor algebra, discussing key tensor operations, norms, and properties that form the basis of tensor factorization methods. The two most widely used decompositions—Canonical Polyadic (CP) and Tucker decomposition—are examined in detail, along with alternative factorization techniques such as Tensor Train (TT), Tensor Ring (TR), and Block Term Decomposition (BTD). We explore the computational complexity of these methods and discuss numerical optimization techniques, including Alternating Least Squares (ALS), gradient-based approaches, and probabilistic tensor models. The paper then delves into the applications of tensor decomposition in signal processing, where tensors have been successfully applied to source separation, multi-sensor data fusion, image processing, and compressed sensing. In machine learning, tensor-based models have enhanced feature extraction, deep learning efficiency, and representation learning. We highlight the role of tensor decomposition in reducing the parameter space of deep neural networks, improving generalization, and accelerating training through low-rank approximations. Despite its numerous advantages, tensor decomposition faces several challenges, including the difficulty of determining tensor rank, the computational cost of large-scale tensor factorization, and robustness to noise and missing data. We discuss recent theoretical advancements addressing uniqueness conditions, rank estimation strategies, and adaptive tensor factorization techniques that improve performance in real-world applications. Furthermore, we explore emerging trends in tensor methods, including their integration with quantum computing, neuroscience, personalized medicine, and geospatial analytics. Finally, we provide a detailed discussion of open research questions, such as the need for more scalable decomposition algorithms, automated rank selection mechanisms, and robust tensor models that can handle high-dimensional, noisy, and adversarial data. As data-driven applications continue to evolve, tensor decomposition is poised to become an indispensable tool for uncovering hidden patterns in complex datasets, advancing both theoretical research and practical implementations across multiple scientific domains.

Keywords: tensor decomposition; multi-way data analysis; Canonical Polyadic (CP) decomposition; Tucker decomposition; Tensor Train (TT); Tensor Ring (TR); signal processing; machine learning; low-rank approximation; computational complexity; feature extraction; deep learning; robustness; noise handling; rank estimation; optimization algorithms

---

1. Introduction

The analysis and processing of multidimensional data have become central to many fields, including signal processing, machine learning, neuroscience, and computational sciences. As data complexity increases, traditional vector- and matrix-based methods often fail to capture the underlying structure efficiently. Tensor decomposition, a mathematical framework for representing high-order data structures, has emerged as a powerful tool for uncovering latent patterns, reducing dimensionality, and improving computational efficiency in various applications [1]. Tensors, which generalize matrices to higher dimensions, naturally arise in a wide range of domains. For instance, in signal processing, multidimensional signals such as hyperspectral images, multi-antenna wireless signals, and electroencephalography (EEG) data can be more effectively represented using tensor models [2]. In machine learning, tensors are employed in deep learning, data fusion, and knowledge graph analysis, where multi-way interactions are essential for capturing meaningful relationships [3]. The ability of tensor decompositions to leverage multilinear algebra allows for richer representations of structured data, overcoming some of the limitations of conventional matrix factorization methods. Several prominent tensor decomposition techniques exist, each with its strengths and applications. The CANDECOMP/PARAFAC (CP) decomposition factorizes a tensor into a sum of rank-one components, facilitating interpretability and uniqueness under certain conditions. The Tucker decomposition provides a more flexible representation by decomposing a tensor into a core tensor and factor matrices, enabling dimensionality reduction and feature extraction [4]. More recent developments, such as Tensor Train (TT) and Tensor Ring (TR) decompositions, have introduced scalable methods for high-dimensional data compression while maintaining computational feasibility. Beyond theoretical advancements, tensor decomposition methods have found widespread use in real-world problems. In signal processing, they contribute to source separation, multi-channel filtering, and compressed sensing [5]. In machine learning, tensor factorization is leveraged for recommender systems, graph embedding, and deep neural network compression. Additionally, scientific applications, such as chemometrics, neuroscience, and quantum physics, have benefited from tensor-based models that enable efficient representation and analysis of complex data [6]. Despite their advantages, tensor decomposition techniques also present computational challenges. The curse of dimensionality, non-convex optimization, and scalability issues pose difficulties in practical implementations. Recent research has focused on developing efficient algorithms, low-rank approximations, and randomized methods to mitigate these limitations and make tensor decompositions applicable to large-scale datasets. This paper provides a comprehensive overview of tensor decomposition methods, exploring their mathematical foundations, computational strategies, and applications in signal processing and machine learning. We discuss key theoretical properties, algorithmic developments, and emerging trends that drive ongoing research in this area [7]. By highlighting both the strengths and challenges of tensor methods, we aim to provide insights into their potential for advancing data-driven disciplines and enabling more efficient analysis of high-dimensional data structures.

2. Fundamentals of Tensor Decomposition

2.1. Definition and Notation

A tensor is a multi-dimensional generalization of a matrix, often represented as a multi-way array. Formally, an N-th order (or N-way) tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ consists of elements indexed by N dimensions, also known as modes. When $N = 1$, the tensor reduces to a vector, and when $N = 2$, it corresponds to a matrix. For $N \geq 3$, the structure extends into higher dimensions [8]. The notation for tensors follows standard conventions:

- Scalars are denoted by lowercase letters (e.g., x) [9].
- Vectors are denoted by bold lowercase letters (e.g., $\mathbf{x}$) [10].
- Matrices are represented by bold uppercase letters (e.g., $\mathbf{X}$) [11].
- Tensors are denoted by calligraphic uppercase letters (e.g., $\mathcal{X}$).

A key operation in tensor analysis is the unfolding (or matricization) of a tensor, which reshapes it into a matrix along a specified mode [12]. The mode-n unfolding of a tensor $\mathcal{X}$ is denoted as $\mathbf{X}_{(n)}$ and rearranges the tensor fibers along mode n into a matrix [13].

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 April 2025 doi:10.20944/preprints202504.0276.v1

2.2. Basic Tensor Operations

Several fundamental operations are essential in tensor decomposition:

- **Outer Product:** Given vectors $\mathbf{a} \in \mathbb{R}^{I_1}$, $\mathbf{b} \in \mathbb{R}^{I_2}$, and $\mathbf{c} \in \mathbb{R}^{I_3}$, their outer product forms a rank-one tensor:
$$\mathcal{X} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}, \tag{1}$$
where each element is given by $x_{i_1,i_2,i_3} = a_{i_1} b_{i_2} c_{i_3}$.
- **Mode-n Product:** The multiplication of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ along mode n is defined as:
$$\mathcal{Y} = \mathcal{X} \times_n \mathbf{U}, \tag{2}$$
where the resulting tensor $\mathcal{Y}$ has dimensions $I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$ [14].
- **Frobenius Norm:** The norm of a tensor is given by [15]:
$$\|\mathcal{X}\|_F = \sqrt{\sum_{i_1,i_2,\ldots,i_N} x^2_{i_1,i_2,\ldots,i_N}}. \tag{3}$$

2.3. Low-Rank Representation and Importance in Applications

Many real-world tensor datasets exhibit low-rank structures, meaning they can be approximated using a small number of factors [16]. This property is crucial for efficient storage, computational feasibility, and interpretability. Low-rank tensor decompositions are widely used in applications such as:

- **Signal Processing:** Source separation, blind deconvolution, and multi-way filtering [17].
- **Machine Learning:** Dimensionality reduction, data fusion, and knowledge discovery [18].
- **Computer Vision:** Image compression, multi-view learning, and feature extraction [19].
- **Biomedical Engineering:** Brain imaging, genomic analysis, and medical signal processing.

These fundamental operations and properties set the stage for the tensor decomposition techniques discussed in the following sections [20].

3. Tensor Decomposition Methods

Tensor decomposition methods provide a framework for analyzing and factorizing high-dimensional data into meaningful low-rank components [21]. These decompositions extend classical matrix factorizations, such as singular value decomposition (SVD), to higher-order data structures, enabling efficient representation, compression, and interpretation of complex datasets. In this section, we present a comprehensive overview of major tensor decomposition techniques, including CANDECOMP/PARAFAC (CP) decomposition, Tucker decomposition, Tensor Train decomposition, and more recent approaches [22].

3.1. CANDECOMP/PARAFAC (CP) Decomposition

The CANDECOMP/PARAFAC (CP) decomposition [23,24] is one of the most fundamental tensor factorization techniques [25]. It decomposes an N-th order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ into a sum of rank-one tensors:
$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \circ \mathbf{a}_r^{(N)}, \tag{4}$$

where each $\mathbf{a}_r^{(n)} \in \mathbb{R}^{I_n}$ is a factor matrix along mode n, and R represents the rank of the decomposition [26].

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 April 2025 doi:10.20944/preprints202504.0276.v1

3.1.1. Properties and Uniqueness

One of the key advantages of CP decomposition is its uniqueness under mild conditions [27]. Unlike matrix factorizations, which often require additional constraints to ensure unique solutions, CP decomposition is unique (up to permutation and scaling) under the Kruskal condition:

$$k_1 + k_2 + \cdots + k_N \geq 2R + (N-1), \tag{5}$$

where k_n represents the Kruskal rank of the factor matrices [28].

3.1.2. Algorithms for CP Decomposition

Several algorithms have been developed to compute CP decomposition:

- **Alternating Least Squares (ALS):** The most commonly used method, which iteratively updates one factor matrix at a time while keeping others fixed.
- **Gradient-Based Methods:** These include stochastic gradient descent (SGD) and conjugate gradient techniques to improve convergence [29].
- **Randomized and Approximate Methods:** Tensor sketching and randomized SVD are used to accelerate CP decomposition for large-scale data [30].

3.1.3. Applications

CP decomposition has found extensive applications in:

- **Signal Processing:** Blind source separation, multi-sensor data fusion, and channel estimation [31].
- **Machine Learning:** Topic modeling, recommendation systems, and deep learning compression.
- **Neuroscience:** EEG and fMRI data analysis for identifying brain activity patterns.

3.2. Tucker Decomposition

The Tucker decomposition [32] is a more general factorization method that decomposes a tensor into a core tensor and factor matrices:

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)}, \tag{6}$$

where $\mathcal{G} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ is the core tensor and each $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times J_n}$ is a factor matrix.

3.2.1. Advantages Over CP Decomposition

Unlike CP decomposition, which enforces strict rank constraints, Tucker decomposition provides more flexibility by allowing a reduced-dimensional core tensor. This leads to:

- Better interpretability in applications like dimensionality reduction.
- Improved compression capabilities in image and video processing.
- Efficient representation of large-scale tensors with controlled rank [33].

3.2.2. Computation and Algorithms

Tucker decomposition is typically computed using:

- **Higher-Order SVD (HOSVD):** An extension of matrix SVD applied sequentially along each mode [34].
- **Higher-Order Orthogonal Iteration (HOOI):** An iterative refinement approach that improves the factorization accuracy.

3.2.3. Applications

Tucker decomposition is widely used in:

- **Data Compression:** Image and video compression through low-rank approximation [35].
- **Multimodal Learning:** Combining multiple data sources for improved feature extraction [36].

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 April 2025 doi:10.20944/preprints202504.0276.v1

- **Bioinformatics:** Protein structure analysis and gene expression modeling [37].

3.3. Tensor Train Decomposition

For very high-dimensional tensors, the Tensor Train (TT) decomposition [38] provides an efficient representation that overcomes the curse of dimensionality. TT decomposition expresses a tensor as a sequence of low-rank tensor contractions:

$$\mathcal{X}(i_1, i_2, \ldots, i_N) = \mathbf{G}^{(1)}(i_1)\mathbf{G}^{(2)}(i_2)\ldots\mathbf{G}^{(N)}(i_N), \tag{7}$$

where each $\mathbf{G}^{(n)}(i_n)$ is a small core tensor, drastically reducing storage complexity.

3.3.1. Advantages

- **Scalability:** TT decomposition scales logarithmically with tensor order, making it suitable for extremely high-dimensional data.
- **Efficient Computation:** Operations like tensor contraction and matrix-vector multiplication become computationally feasible.

3.3.2. Applications

Tensor Train decomposition is used in:

- **Quantum Physics:** Modeling high-dimensional quantum states [39].
- **Neural Networks:** Compression of large neural network weight matrices [40].
- **Large-Scale Optimization:** Solving high-dimensional PDEs and tensor regression problems [41].

3.4. Other Advanced Decomposition Methods

3.4.1. Tensor Ring Decomposition

Tensor Ring (TR) decomposition [42] generalizes Tensor Train by allowing circular dependencies among core tensors, offering even more flexible compression.

3.4.2. Sparse and Nonnegative Tensor Decompositions

In cases where tensors contain sparse or nonnegative values, specialized methods like:

- **Sparse CP/Tucker Decomposition:** Enforcing sparsity constraints for efficient representation [43].
- **Nonnegative Tensor Factorization (NTF):** Ensuring interpretability in applications like topic modeling [23].

3.5. Comparison of Tensor Decomposition Techniques

The choice of tensor decomposition method depends on the application requirements. Table 1 summarizes key properties.

Table 1. Comparison of tensor decomposition methods.

Method	Uniqueness	Scalability	Compression
CP	High	Moderate	Low
Tucker	Moderate	Moderate	High
TT	Low	High	Very High
TR	Low	High	Very High

This section has provided an extensive overview of tensor decomposition techniques, setting the stage for their applications in signal processing and machine learning.

4. Theoretical Advancements and Open Problems in Tensor Decomposition

While tensor decomposition has proven to be an invaluable tool in numerous applications, several theoretical challenges remain. The mathematical foundations of tensor factorization are still evolving, with ongoing research in areas such as uniqueness conditions, rank estimation, computational complexity, and robustness to noise [44]. This section explores key theoretical advancements in tensor decomposition and highlights open problems that warrant further investigation.

4.1. Uniqueness and Identifiability of Tensor Decomposition

One of the most fundamental questions in tensor analysis is the uniqueness of decomposition—whether a given tensor has a unique factorization. Unlike matrix decomposition, where Singular Value Decomposition (SVD) guarantees a unique representation up to orthogonal transformations, tensor decomposition does not always have a unique solution.

4.1.1. Uniqueness of CP Decomposition

The uniqueness of CP decomposition depends on the rank and structure of the tensor [45]. A key result in this domain is Kruskal's theorem, which states that CP decomposition is unique if the sum of the Kruskal ranks of the factor matrices satisfies:

$$k_A + k_B + k_C \geq 2R + 2. \tag{8}$$

However, determining the Kruskal rank of a matrix is itself a challenging problem. Recent research has explored alternative uniqueness conditions based on algebraic geometry, statistical independence, and sparsity constraints [46].

4.1.2. Uniqueness of Tucker Decomposition

Unlike CP decomposition, Tucker decomposition is not unique due to the rotational freedom of the core tensor. This non-uniqueness has implications for interpretability and compression applications. Researchers have proposed constraints such as orthogonality and sparsity to enforce uniqueness in Tucker factorization [47].

4.2. Tensor Rank Estimation and Low-Rank Approximation

Determining the rank of a tensor is a crucial yet difficult problem. Unlike matrices, where rank is well-defined and efficiently computable, the rank of a tensor is not straightforward to determine [48].

4.2.1. Challenges in Determining Tensor Rank

Tensor rank exhibits several counterintuitive properties:

- The rank of a tensor is not necessarily equal to the maximum of its matrix unfoldings [49].
- There exist tensors with arbitrarily high ranks despite having low-dimensional components.
- Tensor rank can be sensitive to small perturbations in the data [50].

Given these challenges, researchers have explored surrogate measures such as the nuclear norm and Schatten-p norms to approximate tensor rank efficiently.

4.2.2. Low-Rank Tensor Approximation

Low-rank approximation is essential in practical applications where full tensor decomposition is computationally infeasible [51]. Key approaches include:

- **Truncated CP and Tucker Decomposition:** Retains only the most significant components [52].
- **Randomized Tensor Factorization:** Uses random projections to estimate low-rank components [53].
- **Sparse and Structured Approximations:** Imposes sparsity constraints to enhance interpretability.

4.3. Computational Hardness and Approximation Guarantees

The computational complexity of tensor decomposition is significantly higher than that of matrix decomposition. Several fundamental hardness results have been established:

- Computing the rank of a given tensor is NP-hard for tensors of order three or higher.

- Finding the best rank-R CP decomposition is NP-hard in general [54].
- Tucker decomposition involves solving large-scale SVD problems, which can be computationally prohibitive.

To address these challenges, researchers have developed approximation algorithms that trade accuracy for efficiency. Probabilistic and heuristic methods, such as Alternating Least Squares (ALS) with regularization, have been widely adopted despite lacking theoretical guarantees on convergence speed and global optimality.

4.4. Robustness and Stability in Noisy Environments

Real-world tensor data often contains noise, missing values, and adversarial perturbations. Developing robust tensor decomposition methods is crucial for reliable data analysis.

4.4.1. Tensor Decomposition with Missing Data

Handling missing data in tensors is a fundamental challenge. Strategies for missing data imputation include:

- **Low-Rank Tensor Completion:** Uses nuclear norm minimization to estimate missing entries.
- **Bayesian Tensor Factorization:** Incorporates probabilistic priors to model uncertainty.
- **Graph-Based Completion:** Leverages relational structures in the data to infer missing values [55].

4.4.2. Robust Tensor Decomposition in Noisy Settings

Noisy tensor data arises in applications such as hyperspectral imaging and biomedical signal processing. To enhance robustness, researchers have explored:

- **Sparse and Robust CP Decomposition:** Incorporates ℓ_1 norm regularization to suppress outliers [19].
- **Total Variation (TV) Regularization:** Enforces smoothness in tensor factorization for denoising applications.
- **Bayesian Nonparametric Models:** Uses hierarchical priors to adaptively model noise distributions.

4.4.3. Adversarial Robustness in Machine Learning Applications

Recent studies have shown that tensor-based machine learning models can be vulnerable to adversarial attacks. Open research directions include:

- **Defending Against Tensor-Based Adversarial Attacks:** Developing regularization techniques to enhance model security.
- **Certifiable Robustness of Tensor Factorization:** Establishing theoretical guarantees on robustness under perturbations [56].
- **Adversarial Training for Tensor Networks:** Enhancing resilience of tensor models to adversarial manipulations.

4.5. Theoretical Connections Between Tensor Methods and Other Fields

Tensor decomposition shares deep mathematical connections with several areas of research. These connections provide new insights into tensor factorization and its applications [57].

4.5.1. Tensors and Algebraic Geometry

Algebraic geometry has been instrumental in understanding the structure of tensor decompositions [58]. Key areas of intersection include:

- **Tensor Rank Bounds:** Using algebraic varieties to establish rank constraints [59].
- **Secant Varieties and Decomposability:** Studying geometric conditions for unique decomposability.

 doi:10.20944/preprints202504.0276.v1

- **Homotopy Methods for Tensor Factorization:** Leveraging algebraic topology for efficient decomposition algorithms.

4.5.2. Tensors and Quantum Information Theory

Tensor networks play a crucial role in quantum computing and quantum information theory. Connections between tensor methods and quantum mechanics include:

- **Tensor Network Representations of Quantum States:** Efficiently encoding quantum many-body systems [60].
- **Entanglement Entropy and Tensor Ranks:** Analyzing the complexity of quantum entanglement using tensor factorizations [61].
- **Quantum Algorithms for Tensor Factorization:** Exploring quantum-inspired methods for high-dimensional tensor decomposition.

4.6. Conclusion

Despite significant progress in tensor decomposition, several open theoretical challenges remain. Understanding uniqueness conditions, developing efficient rank estimation techniques, and improving robustness in noisy settings are critical directions for future research. Furthermore, interdisciplinary connections between tensors, algebraic geometry, and quantum mechanics offer promising avenues for advancing both theoretical and applied tensor analysis [62]. Addressing these challenges will not only refine the mathematical foundations of tensor decomposition but also unlock new applications in artificial intelligence, scientific computing, and beyond.

5. Applications of Tensor Decomposition in Signal Processing and Machine Learning

Tensor decomposition techniques have found widespread applications across various domains, particularly in signal processing and machine learning. The ability of tensors to capture multi-way interactions makes them invaluable in representing and analyzing structured data. This section explores key application areas, highlighting how tensor methods contribute to performance improvements and computational efficiency.

5.1. Applications in Signal Processing

Signal processing often involves handling high-dimensional data, such as multi-sensor signals, speech recordings, biomedical signals, and image sequences [63]. Tensor decomposition plays a crucial role in several signal processing tasks, including source separation, array signal processing, and compressed sensing [64].

5.1.1. Blind Source Separation

Blind Source Separation (BSS) aims to recover underlying source signals from observed mixtures without prior knowledge of the mixing process. Traditional matrix-based methods, such as Independent Component Analysis (ICA), often assume statistical independence of sources [65]. However, tensor decomposition provides a more general framework by leveraging the multi-way structure of data.

- **CP Decomposition for BSS:** The CP decomposition is particularly useful in scenarios where the observed signal is modeled as a sum of rank-one components, each corresponding to an independent source [66].
- **Tucker Decomposition for Multimodal Data:** When multiple sources are recorded through different modalities (e.g., EEG and fMRI in neuroscience), Tucker decomposition enables joint analysis of multimodal data [67].

Applications of tensor-based BSS include:

- **EEG and fMRI Analysis:** Identifying independent brain activity sources.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 April 2025 doi:10.20944/preprints202504.0276.v1

- **Speech Processing:** Separating overlapping speech signals in audio recordings.
- **Wireless Communications:** Decoupling multiple transmitted signals in MIMO systems [68].

5.1.2. Array Signal Processing

Array signal processing involves analyzing signals collected by an array of sensors, such as radar, sonar, and antenna arrays in wireless communications [69]. Tensors naturally represent multi-dimensional array data, making decomposition techniques useful in:

- **Direction of Arrival (DOA) Estimation:** Tensor-based subspace methods improve the accuracy of DOA estimation in multi-antenna systems.
- **Beamforming:** Tensor decompositions help design optimal beamforming weights for interference suppression [70].
- **Channel Estimation:** In MIMO communication, tensor methods improve the estimation of channel state information (CSI) [71].

5.1.3. Compressed Sensing and Sparse Signal Recovery

Compressed sensing exploits the sparsity of signals to enable accurate reconstruction from fewer measurements. Tensor decomposition enhances compressed sensing by leveraging multi-way sparsity patterns, leading to more efficient signal recovery. Applications include:

- **Medical Imaging:** Accelerating MRI and CT image reconstruction [72].
- **Wireless Communications:** Efficient spectrum sensing in cognitive radio networks.
- **Astronomical Imaging:** Recovering high-resolution images from sparse telescope data [73].

5.2. Applications in Machine Learning

In machine learning, tensor decompositions serve as powerful tools for dimensionality reduction, feature extraction, deep learning compression, and graph-based learning. Below, we explore some major applications [74].

5.2.1. Dimensionality Reduction and Feature Extraction

Many machine learning tasks involve high-dimensional data, which can lead to issues such as overfitting and increased computational complexity. Tensor decomposition helps in reducing dimensionality while preserving critical information.

- **Tucker Decomposition for Feature Selection:** By extracting low-dimensional representations of data, Tucker decomposition improves classification and clustering performance.
- **CP Decomposition in Natural Language Processing (NLP):** Tensor factorization helps in word embedding models and topic modeling.

  Examples of applications include:

- **Face Recognition:** Tensor-based feature extraction improves accuracy in facial recognition systems.
- **Text Mining:** Tensor-based topic models reveal latent structures in large text corpora.
- **Recommender Systems:** Tensor decomposition enhances collaborative filtering by capturing higher-order user-item interactions.

5.2.2. Deep Learning Compression and Acceleration

Deep neural networks (DNNs) often suffer from high memory and computational costs [75]. Tensor decomposition provides a structured approach to compressing and accelerating deep learning models [76].

- **Tensor Train Decomposition for Model Compression:** TT decomposition reduces the number of parameters in fully connected layers, making deep learning models more efficient [77].

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 April 2025 doi:10.20944/preprints202504.0276.v1

- **CP and Tucker Decompositions for Convolutional Neural Networks (CNNs):** These methods help decompose convolutional filters, reducing computational complexity in CNNs [78].

Tensor-based deep learning acceleration is widely used in:

- **Edge and Mobile AI:** Deploying efficient neural networks on resource-constrained devices.
- **Autonomous Vehicles:** Optimizing deep learning models for real-time object detection.
- **Medical Diagnosis:** Improving efficiency in AI-driven medical image analysis [79].

5.2.3. Knowledge Graph Completion and Graph Learning

Knowledge graphs, which represent relationships between entities in large-scale databases, benefit significantly from tensor-based models. Tensor decomposition enables:

- **Link Prediction:** Identifying missing relationships in knowledge graphs.
- **Graph Embedding:** Representing nodes in a low-dimensional space for improved clustering and classification.

Applications include:

- **Biomedical Research:** Discovering new drug interactions and gene-disease associations.
- **Social Network Analysis:** Detecting hidden patterns in social media interactions.
- **Recommendation Systems:** Enhancing personalized recommendations based on multi-relational data.

5.3. Comparison of Tensor Decomposition Applications

Table 2 summarizes the key applications of different tensor decomposition techniques.

Table 2. Comparison of tensor decomposition applications.

Tensor Decomposition	Application	Domain
CP Decomposition	Blind Source Separation	Signal Processing
Tucker Decomposition	Feature Extraction	Machine Learning
Tensor Train	Neural Network Compression	Deep Learning
Tensor Ring	Large-Scale Data Compression	High-Performance Computing
Sparse Tensor Factorization	Recommender Systems	Machine Learning

5.4. Challenges and Future Directions

Despite their advantages, tensor decomposition methods face several challenges:

- **Scalability:** Handling large-scale tensors efficiently remains a challenge [80].
- **Computational Complexity:** Many tensor decomposition algorithms are iterative and require careful optimization.
- **Interpretability:** While tensor methods provide compact representations, interpreting the results in real-world applications can be complex.

Future research directions include:

- **Efficient Parallel and Distributed Algorithms:** Leveraging GPU and cloud computing for large-scale tensor computations [81].
- **Hybrid Models:** Integrating tensor decomposition with deep learning for enhanced performance [82].
- **Robustness and Generalization:** Developing tensor methods that are robust to noise and missing data [83].

This section has highlighted the broad spectrum of applications of tensor decomposition in signal processing and machine learning, showcasing its impact in real-world problems.

 doi:10.20944/preprints202504.0276.v1

6. Computational Challenges and Optimization Techniques for Tensor Decomposition

Although tensor decomposition provides powerful tools for analyzing multi-way data, its computational complexity and scalability present significant challenges [84,85]. As the size and order of tensors grow, traditional algorithms become impractical due to high memory requirements and increased computational costs [86]. This section discusses key computational challenges associated with tensor decomposition and explores optimization techniques, including efficient algorithms, parallelization strategies, and low-rank approximations.

6.1. Computational Complexity of Tensor Decomposition

Tensor decomposition algorithms often require solving large-scale optimization problems, making their computational complexity a critical concern. Below, we analyze the computational cost of common tensor decompositions [55].

6.1.1. CP Decomposition Complexity

The CANDECOMP/PARAFAC (CP) decomposition approximates an N-way tensor as a sum of rank-one components [87]. The Alternating Least Squares (ALS) algorithm, commonly used for CP decomposition, updates each factor matrix iteratively [88]. The computational complexity per iteration is:

$$O(R \sum_{n=1}^{N} I_n \prod_{m \neq n} I_m), \tag{9}$$

where R is the rank of decomposition, and I_n is the dimension along mode n. As the tensor order increases, the cost grows exponentially, leading to the well-known **curse of dimensionality** [89].

6.1.2. Tucker Decomposition Complexity

The Tucker decomposition requires computing a core tensor along with factor matrices. The standard approach uses the Higher-Order Singular Value Decomposition (HOSVD), which involves computing the SVD of mode-n unfolding matrices. The computational complexity is:

$$O(\sum_{n=1}^{N} I_n^3 \prod_{m \neq n} I_m). \tag{10}$$

Since each unfolding matrix can be large, computing Tucker decomposition for high-order tensors can be prohibitive.

6.1.3. Tensor Train and Tensor Ring Complexity

Tensor Train (TT) decomposition provides a scalable alternative for high-order tensors by factorizing them into a sequence of smaller core tensors [90]. The computational complexity of TT decomposition is:

$$O(NR^2 I), \tag{11}$$

where R is the TT-rank, and I is the average dimension size [91]. TT decomposition avoids the exponential growth in memory and computation but requires careful rank selection. Tensor Ring (TR) decomposition further generalizes TT by allowing circular dependencies among core tensors. While TR decomposition offers increased flexibility, it also introduces additional optimization complexity [92].

6.2. Memory Constraints and Storage Optimization

High-dimensional tensors require significant storage, making memory-efficient representations essential. Strategies for reducing memory footprint include:

- **Sparse Tensor Storage:** Instead of storing all elements, sparse representations store only nonzero values and their indices, significantly reducing memory usage [93].

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 April 2025 doi:10.20944/preprints202504.0276.v1

- **Compressed Formats:** Using quantization and low-bit representations to store tensor elements efficiently.
- **Distributed Storage:** Splitting tensors across multiple processing units in distributed computing environments.

6.3. Optimization Techniques for Tensor Decomposition

Given the high computational cost, various optimization techniques have been developed to accelerate tensor decomposition [94]. We discuss key strategies below.

6.3.1. Alternating Least Squares (ALS) and Variants

The Alternating Least Squares (ALS) method is a widely used optimization technique for CP and Tucker decomposition. ALS iteratively updates one factor matrix at a time while fixing the others. Despite its simplicity, ALS may suffer from slow convergence and local minima issues. Variants of ALS include:

- **Regularized ALS:** Adds L2 regularization to prevent overfitting and improve generalization.
- **Stochastic ALS:** Uses stochastic gradient updates to accelerate convergence.
- **Randomized ALS:** Incorporates randomization techniques to reduce computational cost [95].

6.3.2. Gradient-Based Optimization

Gradient-based optimization methods, such as Stochastic Gradient Descent (SGD) and Adam optimizer, have been successfully applied to tensor decomposition [96]. These methods are particularly useful when dealing with large-scale tensors.

- **SGD for CP and Tucker:** Updates factor matrices using mini-batches to improve scalability.
- **Second-Order Optimization:** Uses Newton's method and conjugate gradient techniques to accelerate convergence.
- **Momentum-Based Optimization:** Incorporates momentum to avoid oscillations in the optimization landscape [97].

6.3.3. Randomized and Approximate Methods

Randomized algorithms provide efficient approximations to tensor decompositions with lower computational cost. These methods include:

- **Randomized SVD for Tucker:** Computes low-rank approximations using random projections.
- **Sketching Methods:** Uses tensor sketching techniques to reduce dimensionality before applying decomposition [98].
- **Probabilistic Tensor Factorization:** Applies Bayesian inference to estimate tensor components in an approximate manner.

6.3.4. Parallel and Distributed Computing

As tensor dimensions increase, parallel and distributed implementations become essential for scalability. Approaches include:

- **GPU Acceleration:** Uses CUDA-based tensor operations to parallelize matrix multiplications.
- **Multi-Core Processing:** Splits tensor factorization tasks across multiple CPU cores.
- **Distributed Tensor Decomposition:** Implements factorization on large-scale clusters using frameworks like TensorFlow and Spark.

6.4. Tensor Decomposition in Large-Scale and Streaming Data

Modern applications often involve streaming data, where tensors evolve over time [99]. Traditional batch decomposition techniques may not be efficient in such scenarios. Instead, dynamic and online tensor decomposition methods are required [100].

 doi:10.20944/preprints202504.0276.v1

6.4.1. Incremental and Online Tensor Decomposition

Online tensor decomposition updates factor matrices incrementally as new data arrives. Methods include:

- **Online CP Decomposition:** Maintains a low-rank tensor approximation that evolves with streaming data [101].
- **Incremental Tucker Decomposition:** Updates core tensors and factor matrices without recomputing the entire decomposition.

6.4.2. Tensor Decomposition in High-Performance Computing

High-performance computing (HPC) techniques are essential for tensor decomposition in large-scale datasets. Techniques include:

- **Supercomputing Implementations:** Running tensor decomposition on high-performance clusters [102].
- **Hybrid CPU-GPU Models:** Combining CPU and GPU processing for improved efficiency [103].
- **Cloud-Based Tensor Factorization:** Leveraging cloud computing platforms for distributed tensor computations [104].

6.5. Future Directions in Tensor Computation Optimization

As tensor decomposition continues to evolve, several research challenges remain:

- **Scalable Algorithms:** Developing more efficient algorithms that can handle petabyte-scale tensors.
- **Adaptive Rank Selection:** Automating the selection of optimal tensor ranks for decomposition.
- **Robustness in Noisy Environments:** Enhancing tensor methods to handle missing and corrupted data.
- **Quantum Tensor Computation:** Exploring quantum algorithms for tensor decomposition to achieve exponential speedups [105].

This section has discussed the computational challenges of tensor decomposition and presented optimization techniques that enable large-scale and efficient implementations. The next section explores emerging applications and the future impact of tensor methods in artificial intelligence and data science.

7. Emerging Applications and Future Directions of Tensor Decomposition

As data-driven technologies continue to evolve, tensor decomposition is emerging as a powerful tool in cutting-edge applications across artificial intelligence, computational science, and engineering. The ability to analyze multi-dimensional data efficiently makes tensor methods invaluable in domains such as quantum computing, neuroscience, personalized medicine, and deep learning optimization [106]. This section explores emerging applications of tensor decomposition and discusses future research directions that will shape the next generation of multi-way data analysis.

7.1. Tensor Decomposition in Artificial Intelligence and Machine Learning

Tensor methods have gained significant attention in artificial intelligence (AI) due to their ability to model multi-relational data [107]. Beyond their traditional role in feature extraction and dimensionality reduction, tensors are now being used in areas such as explainable AI, federated learning, and continual learning [108].

7.1.1. Tensor-Based Deep Learning Architectures

Deep learning models, particularly convolutional neural networks (CNNs) and transformer-based architectures, suffer from high memory and computational costs. Tensor decomposition offers solutions for improving efficiency:

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 April 2025 doi:10.20944/preprints202504.0276.v1

- **Tensor Compression in Transformers:** Large language models (LLMs) such as GPT and BERT can be compressed using tensor train (TT) and tensor ring (TR) decomposition, reducing the number of parameters while preserving accuracy.
- **Factorized Convolutional Layers:** CP and Tucker decomposition can replace standard convolutional filters in CNNs, leading to faster inference and reduced model size.
- **Low-Rank Attention Mechanisms:** Tensor-based attention models improve efficiency in vision transformers (ViTs) and self-attention networks [109].

7.1.2. Federated Learning and Distributed AI

Federated learning enables machine learning models to be trained across multiple decentralized devices while preserving data privacy [110]. Tensor methods enhance federated learning in the following ways:

- **Efficient Model Aggregation:** Tensor factorization reduces the dimensionality of model updates, leading to faster communication in federated learning settings [111].
- **Privacy-Preserving AI:** Tensor-based representations enable secure and compressed data exchanges in privacy-sensitive applications such as healthcare and finance.

7.1.3. Explainable AI (XAI)

Tensor decomposition enhances interpretability in AI models by providing compact representations of high-dimensional decision spaces. Applications in XAI include:

- **Interpretable Neural Networks:** Tensor factorization helps visualize and analyze neural network activations to understand model decisions [112].
- **Bias Detection in AI:** Tensor analysis can uncover hidden biases in machine learning models by analyzing multi-modal data distributions.

7.2. Tensor Decomposition in Neuroscience and Biomedical Engineering

Neuroscience and biomedical applications generate complex, multi-modal data that require advanced analysis techniques [113]. Tensor decomposition is playing a transformative role in brain imaging, genomics, and personalized medicine [114].

7.2.1. Neuroimaging and Brain Signal Analysis

Tensor methods are widely used in analyzing functional MRI (fMRI), electroencephalography (EEG), and magnetoencephalography (MEG) data. Applications include:

- **Identifying Brain Networks:** CP and Tucker decomposition are used to extract latent brain connectivity patterns from fMRI data.
- **EEG Signal Classification:** Tensor-based models improve classification accuracy in brain-computer interfaces (BCIs).
- **Neurological Disease Diagnosis:** Tensor factorization aids in detecting early markers of neurodegenerative diseases such as Alzheimer's and Parkinson's.

7.2.2. Personalized Medicine and Genomic Data Analysis

Genomics and personalized medicine require high-dimensional data analysis techniques to uncover genetic patterns and optimize treatments. Tensor decomposition contributes to:

- **Drug Discovery:** Factorizing drug-response tensors helps in predicting personalized treatment outcomes [115].
- **Multi-Omics Integration:** Tensor methods combine genetic, transcriptomic, and proteomic data for a comprehensive understanding of diseases.
- **Cancer Biomarker Identification:** Decomposing patient gene expression tensors aids in identifying biomarkers for precision oncology.

 doi:10.20944/preprints202504.0276.v1

7.3. *Tensor Methods in Scientific Computing and Engineering*

Scientific computing applications often involve high-dimensional simulations and complex datasets that benefit from tensor-based representations [116].

7.3.1. Computational Chemistry and Quantum Physics

Tensor decomposition is increasingly used in quantum mechanics and computational chemistry:

- **Quantum State Representation:** Tensor network models provide efficient representations of quantum many-body states.
- **Density Matrix Factorization:** Low-rank tensor approximations help reduce computational complexity in quantum chemistry simulations.
- **Quantum Machine Learning:** Tensor-based learning methods optimize quantum circuit designs [117].

7.3.2. Climate Science and Geospatial Data Analysis

Tensor decomposition is helping climate scientists analyze multi-modal geospatial datasets:

- **Weather Prediction:** Tensor factorization improves climate models by identifying temporal-spatial patterns.
- **Remote Sensing:** Decomposing hyperspectral satellite images aids in land cover classification and environmental monitoring.
- **Disaster Forecasting:** Tensor-based anomaly detection identifies patterns in extreme weather events such as hurricanes and wildfires [118].

7.4. *Challenges and Future Directions in Tensor Decomposition Research*

Despite its growing adoption, tensor decomposition faces several challenges that need to be addressed to fully realize its potential.

7.4.1. Scalability and Computational Efficiency

- Developing faster, scalable algorithms that can handle massive tensors efficiently [119].
- Integrating tensor decomposition with distributed computing frameworks for cloud-based implementations [120].
- Exploring hardware acceleration (e.g., GPU, TPU, and quantum computing) to improve computational speed [121].

7.4.2. Automated and Adaptive Tensor Factorization

Selecting optimal tensor rank is a critical issue that affects accuracy and efficiency. Future research should focus on:

- Adaptive rank estimation techniques to dynamically adjust tensor decomposition models.
- Bayesian and probabilistic tensor methods for uncertainty quantification in factorization.
- Learning-based approaches that use neural networks to optimize tensor decomposition.

7.4.3. Robustness and Generalization in Real-World Data

- Enhancing robustness to noise and missing data in tensor decomposition applications.
- Developing generalizable tensor models that work across different domains and datasets.
- Investigating adversarial robustness of tensor-based models in security-sensitive applications.

7.4.4. Integration with Emerging AI Paradigms

Tensor decomposition is expected to integrate with emerging AI trends:

- **Hybrid AI Models:** Combining tensor factorization with graph neural networks (GNNs) and transformers [122].

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 April 2025 doi:10.20944/preprints202504.0276.v1

- **Neurosymbolic AI:** Leveraging tensor methods to improve symbolic reasoning in AI [123].
- **Edge AI and IoT:** Deploying tensor-optimized AI models on resource-constrained edge devices [55,124].

7.5. Conclusions

Tensor decomposition is rapidly transforming a wide range of scientific and engineering disciplines. From accelerating deep learning models to improving climate predictions, the ability to analyze multi-way data efficiently makes tensor methods indispensable in modern AI and computational science. As research progresses, further advancements in scalability, robustness, and real-world applicability will shape the future of tensor-based machine learning and data analysis [125].

8. Conclusions

Tensor decomposition has emerged as a fundamental mathematical tool with far-reaching implications across signal processing, machine learning, scientific computing, and artificial intelligence. By extending classical matrix factorizations to higher-order data structures, tensor methods enable efficient representation, compression, and analysis of complex multi-dimensional datasets. This paper has provided an in-depth discussion on the mathematical foundations of tensor decomposition, explored its diverse applications, examined recent theoretical advancements, and outlined open challenges that remain in the field.

8.1. Key Takeaways

The study of tensor decomposition has led to several key insights:

- **Mathematical Formulations:** Different tensor factorization techniques, including CP, Tucker, TT, and TR decompositions, provide versatile frameworks for analyzing multi-way data [126].
- **Computational Trade-offs:** While tensor methods offer powerful insights, computational complexity remains a major hurdle, necessitating efficient approximation algorithms.
- **Emerging Applications:** Tensors play a critical role in modern AI, deep learning, biomedical engineering, quantum computing, and geospatial data analysis [127].
- **Theoretical Advancements:** Ongoing research in uniqueness conditions, rank estimation, and robustness to noise continues to refine the mathematical understanding of tensors.

8.2. Challenges and Future Directions

Despite its widespread utility, tensor decomposition faces several unresolved challenges:

- **Scalability and Efficiency:** Handling large-scale tensors requires advanced parallel computing techniques and optimized hardware implementations.
- **Automated Model Selection:** Determining optimal tensor rank and structure remains an open problem, necessitating adaptive and probabilistic approaches.
- **Robustness and Interpretability:** Ensuring the reliability of tensor-based models in noisy and adversarial settings is crucial for real-world deployment.
- **Interdisciplinary Integration:** Bridging the gap between tensor methods and fields such as quantum computing, algebraic geometry, and neuroscience presents exciting research opportunities [128].

8.3. Final Remarks

Tensor decomposition continues to revolutionize data-driven research and technological advancements. As computing power and machine learning algorithms evolve, tensor-based methods will become increasingly integral to processing high-dimensional data efficiently. By addressing current limitations and exploring interdisciplinary collaborations, researchers can further enhance the capabilities of tensor decomposition, unlocking new insights across scientific and engineering domains.

 doi:10.20944/preprints202504.0276.v1

With the rapid growth of data-intensive applications, the future of tensor decomposition lies in the development of more scalable, interpretable, and automated frameworks. As theoretical and computational advancements continue to unfold, tensor methods will remain at the forefront of next-generation AI, machine learning, and scientific discovery.

References

1. Sedighin, F.; Cichocki, A. Image completion in embedded space using multistage tensor ring decomposition. *Frontiers in Artificial Intelligence* **2021**, *4*, 687176.
2. Davidson, I.; Gilpin, S.; Carmichael, O.; Walker, P. Network discovery via constrained tensor analysis of fmri data. In Proceedings of the ACM SIGKDD, 2013.
3. Phan, A.H.; Cichocki, A. Tensor decompositions for feature extraction and classification of high dimensional datasets. *Nonlinear Theory and its Applications, IEICE* **2010**, *1*, 37–68.
4. Lao, N.; Mitchell, T.; Cohen, W.W. Random walk inference and learning in a large scale knowledge base. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John Mcintyre Conference Centre, Edinburgh, Uk, A Meeting of Sigdat, A Special Interest Group of the ACL, 2012, pp. 529–539.
5. Huckle, T.; Waldherr, K.; Schulte-Herbrüggen, T. Computations in quantum tensor networks. *Linear Algebra and its Applications* **2013**, *438*, 750–781.
6. Bondarenko, D.; Feldmann, P. Quantum autoencoders to denoise quantum data. *Physical Review Letters* **2020**, *124*, 130502.
7. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
8. Suchanek, F.M.; Kasneci, G.; Weikum, G. Yago: A Core of Semantic Knowledge. In Proceedings of the Proc. WWW, 2007.
9. Kim, Y.; Park, E.; Yoo, S.; Choi, T.; Yang, L.; Shin, D. Compression of Deep Convolutional Neural Networks for Fast and Low Power Mobile Applications. In Proceedings of the ICLR; Bengio, Y.; LeCun, Y., Eds., 2016.
10. Yang, X.; Gao, M.; Pu, J.; Nayak, A.; Liu, Q.; Emberton Bell, S.; Setter, J.O.; Cao, K.; Ha, H.; Kozyrakis, C.; et al. DNN Dataflow Choice Is Overrated. *arXiv e-prints* **2018**, p. arXiv:1809.04070, [arXiv:cs.DC/1809.04070].
11. Weston, J.; Bordes, A.; Yakhnenko, O.; Usunier, N. Connecting Language and Knowledge Bases with Embedding Models for Relation Extraction. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2013, pp. 1366–1371.
12. Park, C.; Lu, Y.; Saha, S.; Xue, T.; Guo, J.; Mojumder, S.; Apley, D.W.; Wagner, G.J.; Liu, W.K. Convolution hierarchical deep-learning neural network (c-hidenn) with graphics processing unit (gpu) acceleration. *Computational Mechanics* **2023**, *72*, 383–409.
13. Bordes, A.; Usunier, N.; García-Durán, A.; Weston, J.; Yakhnenko, O. Translating Embeddings for Modeling Multi-relational Data. In Proceedings of the NeurIPS, 2013, pp. 2787–2795.
14. Vidal, G. Entanglement Renormalization: an introduction. In *Understanding Quantum Phase Transitions*; Carr, L.D., Ed.; Taylor & Francis, Boca Raton, 2010.
15. Beylkin, G.; Mohlenkamp, M.J. Numerical operator calculus in higher dimensions. *Proceedings of the National Academy of Sciences* **2002**, *99*, 10246–10251.
16. Vervliet, N.; Debals, O.; De Lathauwer, L. Tensorlab 3.0—Numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization. In Proceedings of the 2016 50th Asilomar Conference on Signals, Systems and Computers. IEEE, 2016, pp. 1733–1738.
17. Yang, K.; Wang, S.; Zhou, J.; Yoshimura, T. Energy-efficient scheduling method with cross-loop model for resource-limited CNN accelerator designs. In Proceedings of the Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), May 2017, pp. 1–4. https://doi.org/10.1109/ISCAS.2017.8050800.
18. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **1998**, *86*, 2278–2324.
19. Lebedev, V.; Ganin, Y.; Rakhuba, M.; Oseledets, I.; Lempitsky, V. Speeding-up convolutional neural networks using fine-tuned CP-decomposition. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015-Conference Track Proceedings, 2015.
20. Zhou, Y.; Lentz, E.; Michelson, H.; Kim, C.; Baylis, K. Machine learning for food security: Principles for transparency and usability. *Applied Economic Perspectives and Policy* **2022**, *44*, 893–910.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 3 April 2025 doi:10.20944/preprints202504.0276.v1

21. Tung, F.; Mori, G. Similarity-Preserving Knowledge Distillation. In Proceedings of the ICCV. IEEE, 2019, pp. 1365–1374.
22. Phan, A.H.; Sobolev, K.; Ermilov, D.; Vorona, I.; Kozyrskiy, N.; Tichavsky, P.; Cichocki, A. How to Train Unstable Looped Tensor Network. *arXiv preprint arXiv:2203.02617* **2022**.
23. Carroll, J.D.; Chang, J.J. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika* **1970**, *35*, 283–319.
24. HARSHMAN, R. Foundations of the parafac procedure: Models and conditions for an" explanatory" multimodal factor analysis. *UCLA Working Papers in Phonetics* **1970**, *16*, 1–84.
25. Denil, M.; Shakibi, B.; Dinh, L.; Ranzato, M.; De Freitas, N. Predicting parameters in deep learning. In Proceedings of the Advances in neural information processing systems, 2013, pp. 2148–2156.
26. Oseledets, I. DMRG approach to fast linear algebra in the TT-format. *Computational Methods in Applied Mathematics* **2011**, *11*, 382–393.
27. Mizutani, E.; Dreyfus, S.E.; Nishio, K. On derivation of MLP backpropagation from the Kelley-Bryson optimal-control gradient formula and its application. In Proceedings of the Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IEEE, 2000, Vol. 2, pp. 167–172.
28. LeCun, Y.; Chopra, S.; Hadsell, R.; Ranzato, M.; Huang, F. A tutorial on energy-based learning. *Predicting Structured Data* **2006**, *1*.
29. Cheng, S.; Wang, L.; Xiang, T.; Zhang, P. Tree tensor networks for generative modeling. *Physical Review B* **2019**, *99*, 155131.
30. Yunpeng, C.; Xiaojie, J.; Bingyi, K.; Jiashi, F.; Shuicheng, Y. Sharing Residual Units Through Collective Tensor Factorization in Deep Neural Networks. *arXiv preprint arXiv:1703.02180* **2017**.
31. Jose, C.; Cissé, M.; Fleuret, F. Kronecker Recurrent Units. In Proceedings of the ICML, 2018.
32. Tucker, L.R. Some mathematical notes on three-mode factor analysis. *Psychometrika* **1966**, *31*, 279–311.
33. Deoras, A.; Kombrink, S.; et al. Empirical evaluation and combination of advanced language modeling techniques **2011**.
34. Deng, J.; Berg, A.; Satheesh, S.; Su, H.; Khosla, A.; Fei-Fei, L. ImageNet large scale visual recognition competition 2012 (ilsvrc2012), 2012.
35. Pan, Y.; Xu, J.; Wang, M.; Ye, J.; Wang, F.; Bai, K.; Xu, Z. Compressing Recurrent Neural Networks with Tensor Ring for Action Recognition. In Proceedings of the AAAI, 2019.
36. Kiers, H.A. Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics: A Journal of the Chemometrics Society* **2000**, *14*, 105–122.
37. Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C.L.; Parikh, D. Vqa: Visual question answering. In Proceedings of the ICCV, 2015.
38. Oseledets, I.V. Tensor-Train decomposition. *SIAM Journal on Scientific Computing* **2011**, *33*, 2295–2317.
39. Baez, J.; Stay, M. Physics, topology, logic and computation: a Rosetta Stone. *Springer* **2010**, pp. New structures for physics, 95–172.
40. Moreau, T.; Chen, T.; Vega, L.; Roesch, J.; Yan, E.; Zheng, L.; Fromm, J.; Jiang, Z.; Ceze, L.; Guestrin, C.; et al. A Hardware–Software Blueprint for Flexible Deep Learning Specialization. *IEEE Micro* **2019**, *39*, 8–16.
41. Suzuki, M. Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems. *Communications in Mathematical Physics* **1976**, *51*, 183–190.
42. Zhao, Q.; Zhou, G.; Xie, S.; Zhang, L.; Cichocki, A. Tensor ring decomposition. *arXiv preprint arXiv:1606.05535* **2016**.
43. Liu, D.; Yang, L.T.; Wang, P.; Zhao, R.; Zhang, Q. TT-TSVD: A Multi-modal Tensor Train Decomposition with Its Application in Convolutional Neural Networks for Smart Healthcare. *TOMM* **2022**, *18*, 1–17.
44. Ma, X.; Zhang, P.; Zhang, S.; Duan, N.; Hou, Y.; Zhou, M.; Song, D. A tensorized transformer for language modeling. *Advances in neural information processing systems* **2019**, *32*.
45. De Lathauwer, L.; Castaing, J.; Cardoso, J.F. Fourth-order cumulant-based blind identification of underdetermined mixtures. *IEEE Transactions on Signal Processing* **2007**, *55*, 2965–2973.
46. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in neural information processing systems, 2012, pp. 1097–1105.
47. Lloyd, S.; Schuld, M.; Ijaz, A.; Izaac, J.; Killoran, N. Quantum embeddings for machine learning. *arXiv preprint arXiv:2001.03622* **2020**.

48. Hou, M.; Tang, J.; Zhang, J.; Kong, W.; Zhao, Q. Deep multimodal multilinear fusion with high-order polynomial pooling. *NeurIPS* **2019**.
49. Comon, P. Tensors: a brief introduction. *IEEE Signal Processing Magazine* **2014**, *31*, 44–53.
50. Zhu, J.; Jiang, J.; Chen, X.; Tsui, C. SparseNN: An energy-efficient neural network accelerator exploiting input and output sparsity. In Proceedings of the Proceedings of the Design, Automation Test in Europe Conference Exhibition (DATE), March 2018, pp. 241–244. https://doi.org/10.23919/DATE.2018.8342010.
51. Zhe, S.; Qi, Y.; Park, Y.; Xu, Z.; Molloy, I.; Chari, S. DinTucker: Scaling Up Gaussian Process Models on Large Multidimensional Arrays. In Proceedings of the Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA., 2016, pp. 2386–2392.
52. Wang, B.; Ren, Y.; Shang, L.; Jiang, X.; Liu, Q. Exploring Extreme Parameter Compression for Pre-Trained Language Models. In Proceedings of the ICLR, 2022.
53. Dahl, G.E.; Sainath, T.N.; Hinton, G.E. Improving deep neural networks for LVCSR using rectified linear units and dropout. In Proceedings of the ICASSP, 2013.
54. Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Ives, Z. DBpedia: A Nucleus for a Web of Open Data. In Proceedings of the Proc. ISWC, 2007, pp. 11–15.
55. Kolda, T.G.; Bader, B.W. Tensor decompositions and applications. *SIAM review* **2009**, *51*, 455–500.
56. Socher, R.; Chen, D.; Manning, C.D.; Ng, A.Y. Reasoning With Neural Tensor Networks for Knowledge Base Completion. In Proceedings of the NeurIPS, 2013, pp. 926–934.
57. Zhu, J. Max-margin nonparametric latent feature models for link prediction. In Proceedings of the Proceedings of the 29th International Conference on International Conference on Machine Learning. Omnipress, 2012, pp. 1179–1186.
58. Cover, T.M. Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. *IEEE Transactions on Electronic Computers* **1965**, *EC-14*, 326–334. https://doi.org/10.1109/PGEC.1965.264137.
59. Lao, N.; Cohen, W.W. Relational Retrieval Using a Combination of Path-constrained Random Walks. *Machine Learning* **2010**, *81*, 53–67.
60. Liu, Y.; Ng, M.K. Deep neural network compression by Tucker decomposition with nonlinear response. *Knowledge-Based Systems* **2022**.
61. Wang, Q.F.; Cambria, E.; Liu, C.L.; Hussain, A. Common sense knowledge for handwritten chinese text recognition. *Cognitive Computation* **2013**, *5*, 234–242.
62. Hrinchuk, O.; Khrulkov, V.; Mirvakhabova, L.; Orlova, E.; Oseledets, I. Tensorized embedding layers for efficient model compression. *arXiv preprint arXiv:1901.10787* **2019**.
63. Werbos, P.J. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE* **1990**, *78*, 1550–1560.
64. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the CVPR, 2017.
65. Cichocki, A. Tensor Networks for Dimensionality Reduction, Big Data and Deep Learning. In *Advances in Data Analysis with Computational Intelligence Methods*; Springer, 2018; pp. 3–49.
66. Novikov, A.; Podoprikhin, D.; Osokin, A.; Vetrov, D.P. Tensorizing neural networks. In Proceedings of the NeurIPS, 2015.
67. Chen, S.; Zhou, J.; Sun, W.; Huang, L. Joint Matrix Decomposition for Deep Convolutional Neural Networks Compression. *arXiv preprint arXiv:2107.04386* **2021**.
68. Sobolev, K.; Ermilov, D.; Phan, A.H.; Cichocki, A. PARS: Proxy-Based Automatic Rank Selection for Neural Network Compression via Low-Rank Weight Approximation. *Mathematics* **2022**, *10*, 3801.
69. Viebke, A.; Memeti, S.; Pllana, S.; Abraham, A. Chaos: a parallelization scheme for training convolutional neural networks on intel xeon phi. *The Journal of Supercomputing* **2017**, pp. 1–31.
70. Xu, Z.; Yan, F.; Qi, Y.A. Infinite Tucker Decomposition: Nonparametric Bayesian Models for Multiway Data Analysis. In Proceedings of the Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012, 2012.
71. Tu, F.; Yin, S.; Ouyang, P.; Tang, S.; Liu, L.; Wei, S. Deep Convolutional Neural Network Architecture With Reconfigurable Computation Patterns. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **2017**, *25*, 2220–2233. https://doi.org/10.1109/TVLSI.2017.2688340.
72. Bengio, Y.; Boulanger-Lewandowski, N.; Pascanu, R. Advances in optimizing recurrent networks. In Proceedings of the ICASSP. IEEE, 2013.

73. Lu, L.; Jin, P.; Pang, G.; Zhang, Z.; Karniadakis, G.E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature machine intelligence* **2021**, *3*, 218–229.
74. Schuch, N.; Wolf, M.M.; Verstraete, F.; Cirac, J.I. Computational complexity of projected entangled pair states. *Physical Review Letters* **2007**, *98*, 140506.
75. Chen, J.; Cheng, S.; Xie, H.; Wang, L.; Xiang, T. Equivalence of restricted Boltzmann machines and tensor network states. *Physical Review B* **2018**, *97*, 085104.
76. You, Y.; Buluç, A.; Demmel, J. Scaling deep learning on GPU and knights landing clusters. In Proceedings of the Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2017, p. 9.
77. Do, T.; Do, T.T.; Tran, H.; Tjiputra, E.; Tran, Q.D. Compact trilinear interaction for visual question answering. In Proceedings of the ICCV, 2019.
78. Huang, S.; Xu, Z.; Lv, J. Adaptive local structure learning for document co-clustering. *Knowl.-Based Syst.* **2018**, *148*, 74–84. https://doi.org/10.1016/j.knosys.2018.02.020.
79. Kossaifi, J.; Bulat, A.; Tzimiropoulos, G.; Pantic, M. T-Net: Parametrizing Fully Convolutional Nets With a Single High-Order Tensor. In Proceedings of the CVPR, 2019.
80. Li, S.; Pan, F.; Zhou, P.; Zhang, P. Boltzmann machines as two-dimensional tensor networks. *Physical Review B* **2021**, *104*, 075154.
81. Zhang, P.; Su, Z.; Zhang, L.; Wang, B.; Song, D. A quantum many-body wave function inspired language modeling approach. In Proceedings of the CIKM, 2018.
82. Cao, X.; Wei, X.; Han, Y.; Lin, D. Robust face clustering via tensor decomposition. *IEEE transactions on cybernetics* **2014**, *45*, 2546–2557.
83. OpenAI. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774* **2023**.
84. Zniyed, Y.; Nguyen, T.P.; et al. Efficient tensor decomposition-based filter pruning. *Neural Networks* **2024**, *178*, 106393.
85. Håstad, J. Tensor rank is NP-complete. *Journal of Algorithms* **1990**, *11*, 644–654.
86. Denton, E.L.; Zaremba, W.; Bruna, J.; LeCun, Y.; Fergus, R. Exploiting linear structure within convolutional networks for efficient evaluation. In Proceedings of the Advances in neural information processing systems, 2014, pp. 1269–1277.
87. Wang, D.; Wu, B.; Zhao, G.S.; Chen, H.; Deng, L.; Yan, T.; Li, G. Kronecker CP Decomposition with Fast Multiplication for Compressing RNNs. *IEEE Trans. NNLS* **2021**, *PP*.
88. Nakajima, S.; Tomioka, R.; Sugiyama, M.; Babacan, S.D. Perfect Dimensionality Recovery by Variational Bayesian PCA. In Proceedings of the NeurIPS, 2012.
89. Molchanov, P.; Mallya, A.; Tyree, S.; Frosio, I.; Kautz, J. Importance Estimation for Neural Network Pruning. In Proceedings of the CVPR, 2019.
90. Liu, J.; Luo, J.; Shah, M. Recognizing realistic actions from videos "in the wild". In Proceedings of the 2009 IEEE conference on computer vision and pattern recognition. IEEE, 2009, pp. 1996–2003.
91. Qi, G.; Sun, Y.; Gao, J.; Hu, Y.; Li, J. Matrix variate restricted Boltzmann machine. In Proceedings of the IJCNN, 2016.
92. Torlai, G.; Timar, B.; Van Nieuwenburg, E.P.; Levine, H.; Omran, A.; Keesling, A.; Bernien, H.; Greiner, M.; Vuletić, V.; Lukin, M.D.; et al. Integrating neural networks with a quantum simulator for state reconstruction. *Physical Review Letters* **2019**, *123*, 230504.
93. Rodrigues, C.F.; Riley, G.; Luján, M. Exploration of Task-based Scheduling for Convolutional Neural Networks Accelerators Under Memory Constraints. In Proceedings of the Proceedings of the ACM International Conference on Computing Frontiers, New York, NY, USA, 2019; CF '19, pp. 366–372. https://doi.org/10.1145/3310273.3323162.
94. ten Berge, J.M. The typical rank of tall three-way arrays. *Psychometrika* **2000**, *65*, 525–532.
95. Chen, S.; Lyu, M.R.; King, I.; Xu, Z. Exact and stable recovery of pairwise interaction tensors. In Proceedings of the NeurIPS, 2013, pp. 1691–1699.
96. Eisert, J. Entanglement and tensor network states. *Modelling and Simulation* **2013**, *3*.
97. Sedighin, F.; Cichocki, A.; Yokota, T.; Shi, Q. Matrix and tensor completion in multiway delay embedded space using tensor train, with application to signal reconstruction. *IEEE Signal Processing Letters* **2020**, *27*, 810–814.
98. Chen, Y.; Yang, T.; Emer, J.; Sze, V. Eyeriss v2: A Flexible Accelerator for Emerging Deep Neural Networks on Mobile Devices. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* **2019**, *9*, 292–308. https://doi.org/10.1109/JETCAS.2019.2910232.

99. Rendle, S. Factorization machines. In Proceedings of the ICDM, 2010.
100. Kak, S.C. Quantum neural computing. *Advances in Imaging and Electron Physics* **1995**, *94*, 259–313.
101. Meurice, Y.; Osborn, J.C.; Sakai, R.; Unmuth-Yockey, J.; Catterall, S.; Somma, R.D. Tensor networks for High Energy Physics: contribution to Snowmass 2021. *arXiv preprint arXiv:2203.04902* **2022**.
102. Yin, M.; Liao, S.; Liu, X.; Wang, X.; Yuan, B. Towards Extremely Compact RNNs for Video Recognition With Fully Decomposed Hierarchical Tucker Structure. In Proceedings of the CVPR, 2021.
103. Zhang, Z.; Allen, G.I.; Zhu, H.; Dunson, D. Tensor network factorizations: Relationships between brain structural connectomes and traits. *Neuroimage* **2019**, *197*, 330–343.
104. Jouppi, N.P.; Young, C.; Patil, N.; Patterson, D.; Agrawal, G.; Bajwa, R.; Bates, S.; Bhatia, S.; Boden, N.; Borchers, A.; et al. In-datacenter performance analysis of a tensor processing unit. In Proceedings of the Proceedings of the ACM/IEEE International Symposium on Computer Architecture (ISCA), June 2017, pp. 1–12. https://doi.org/10.1145/3079856.3080246.
105. Liu, X.; Su, J.; Huang, F. Tuformer: Data-Driven Design of Expressive Transformer by Tucker Tensor Representation. In Proceedings of the ICLR, 2022.
106. Chen, Y.; Emer, J.; Sze, V. Eyeriss: A Spatial Architecture for Energy-Efficient Dataflow for Convolutional Neural Networks. In Proceedings of the Proceedings of the ACM/IEEE International Symposium on Computer Architecture (ISCA), June 2016, pp. 367–379. https://doi.org/10.1109/ISCA.2016.40.
107. Han, S.; Liu, X.; Mao, H.; Pu, J.; Pedram, A.; Horowitz, M.A.; Dally, W.J. EIE: Efficient Inference Engine on Compressed Deep Neural Network. In Proceedings of the Proceedings of the ACM/IEEE International Symposium on Computer Architecture (ISCA), June 2016, pp. 243–254. https://doi.org/10.1109/ISCA.2016.30.
108. Parhi, K.K.; Unnikrishnan, N.K. Brain-inspired computing: Models and architectures. *IEEE Open Journal of Circuits and Systems* **2020**, *1*, 185–204.
109. Carrasquilla, J.; Torlai, G.; Melko, R.G.; Aolita, L. Reconstructing quantum states with generative models. *Nature Machine Intelligence* **2019**, *1*, 155–161.
110. Huggins, W.; Patil, P.; Mitchell, B.; Whaley, K.B.; Stoudenmire, E.M. Towards quantum machine learning with tensor networks. *Quantum Science and technology* **2019**, *4*, 024001.
111. Shashua, A.; Levin, A. Linear image coding for regression and classification using the tensor-rank principle. In Proceedings of the CVPR. IEEE, 2001, Vol. 1, pp. I–42.
112. Liu, H.; Singh, P. ConceptNet – A Practical Commonsense Reasoning Tool-Kit. *BT Technology Journal* **2004**, *22*, 211–226.
113. Li, Q.; Wang, B.; Melucci, M. CNM: An Interpretable Complex-valued Network for Matching. In Proceedings of the NAACL, 2019.
114. Valdez, F.; Melin, P. A review on quantum computing and deep learning algorithms and their applications. *Soft Computing* **2022**, pp. 1–20.
115. Han, S.; Mao, H.; Dally, W.J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149* **2015**.
116. Tjandra, A.; Sakti, S.; Nakamura, S. Recurrent Neural Network Compression Based on Low-Rank Tensor Representation. *IEICE Trans. Inf. Syst.* **2020**, *103-D*, 435–449.
117. Hubara, I.; Courbariaux, M.; Soudry, D.; El-Yaniv, R.; Bengio, Y. Binarized neural networks. In Proceedings of the NeurIPS, 2016, pp. 4107–4115.
118. Novikov, A.; Podoprikhin, D.; Osokin, A.; Vetrov, D.P. Tensorizing neural networks. *Advances in Neural Information Processing Systems* **2015**, *28*.
119. Chen, Y.; Luo, T.; Liu, S.; Zhang, S.; He, L.; Wang, J.; Li, L.; Chen, T.; Xu, Z.; Sun, N.; et al. DaDianNao: A Machine-Learning Supercomputer. In Proceedings of the Proceeding of the IEEE/ACM International Symposium on Microarchitecture (MICRO), Dec 2014, pp. 609–622. https://doi.org/10.1109/MICRO.2014.58.
120. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on t Classification. In Proceedings of the ICCV, 2015.
121. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.; Buell, D.A.; et al. Quantum supremacy using a programmable superconducting processor. *Nature* **2019**, *574*, 505–510.
122. Zhang, G.; Gheorghe, M.; Li, Y. A membrane algorithm with quantum-inspired subalgorithms and its application to image processing. *Natural Computing* **2012**, *11*, 701–717.

 doi:10.20944/preprints202504.0276.v1

123. Xin, M.; Wang, Y. Research on image classification model based on deep convolution neural network. *EURASIP J. Image Video Process.* **2019**, *2019*, 40.
124. Zniyed, Y.; Nguyen, T.P.; et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems* **2024**.
125. Xu, Z.; Yan, F.; Qi, Y. Infinite Tucker decomposition: nonparametric Bayesian models for multiway data analysis. In Proceedings of the ICML, 2012.
126. Kressner, D.; Tobler, C. htucker—A MATLAB toolbox for tensors in hierarchical Tucker format. *Mathicse, EPF Lausanne* **2012**.
127. Gao, H.; Sun, L.; Wang, J.X. PhyGeoNet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain. *Journal of Computational Physics* **2021**, *428*, 110079.
128. Fukui, A.; Park, D.H.; Yang, D.; Rohrbach, A.; Darrell, T.; Rohrbach, M. Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding. In Proceedings of the EMNLP, 2016.