

Enhancing Deep Learning Models through Tensorization: A Comprehensive Survey and Framework

Manal Helal^{1,*}

¹ School of Physics, Engineering & Computer Science, Hertfordshire University, HATFIELD, UK

ARTICLE INFO

Article history:

Received 30 January 2025

Received in revised form 25 March 2025

Accepted 25 July 2025

Available online 28 September 2025

Keywords:

Artificial intelligence, blind source separation, image denoising, CP Decomposition, neural network compression, tensor network (TN), tensor train (TT) decomposition, tensorization, Tucker decomposition, and singular value decomposition (SVD)

ABSTRACT

The rapid expansion of publicly available data and the growing complexity of deep learning models have highlighted the need for more effective data representation and analysis methods. Tensorization provides a revolutionary solution, aligning the multidimensional nature of data with compressed deep learning models to yield more interpretable results. This paper provides an in-depth, tutorial-style review of tensorization, multi-way analysis methods, and their integration with deep neural network models, illustrated through various case studies. A Blind Source Separation experiment compares the performance of 2-dimensional algorithms with multi-way algorithms. Experiments were conducted on multiple datasets under different noise and compression conditions. Results indicate that while traditional 2D methods achieve lower Root Mean Square Error, tensor-based methods preserve essential structural and frequency characteristics, making them valuable for applications when accurate signal reconstruction is required. Contrary to the expected difficulties of high dimensionality, utilising multidimensional datasets in their original form and applying multi-way analysis methods based on multilinear algebra can uncover complex relationships among dimensions while reducing model parameters and accelerating processing.

1. Introduction

In modern machine learning, data are often simplified to 2-dimensional matrices for ease of application in linear algebra-based algorithms, despite being inherently high-dimensional. However, applying multiway analysis through multilinear algebra to these multidimensional datasets provides more expressive models, reduces the number of parameters, and accelerates processing, thereby defying the expected dimensionality curse. This paper surveys the theoretical background necessary for understanding these methods, outlines the process of tensorizing matrix-form datasets, and reviews current methods and applications of multiway analysis in compressing deep learning models. It includes a framework for tensorization, an experiment on Blind Source Separation, and a comprehensive review of tensorized machine learning and deep learning applications, concluding with key insights and future research directions.

* Corresponding author.

E-mail address: mhelal@ieee.org

<https://doi.org/10.37934/ard.142.1.261276>

2. Fundamentals of Tensorization

Traditional machine learning (ML) models, including support vector machines (SVMs), regression models, decision trees, and various deep neural network (DNN) architectures like fully connected layers (FCNs), convolutional layers (CNNs), LSTMs, and transformers, are predominantly grounded in linear algebra. This typically necessitates representing data in a 2-dimensional matrix form, a simplified view of inherently high-dimensional data. For instance, image data are often described with two spatial dimensions (width and height), with additional colour channels forming a third dimension and video data introducing a fourth temporal dimension. While standard ML and DNN algorithms manage these dimensions within their design constraints, such as using 1D-CNNs for sequential data and 2D-CNNs for spatial data, they often limit the potential to capture higher-order interactions within the data due to their reliance on lower-dimensional representations.

Tensorization offers a more sophisticated approach by leveraging multilinear algebra to handle data in its full multidimensional form, thereby enabling more expressive models that can capture complex interactions among data dimensions. For instance, while traditional linear methods, such as PCA, project high-dimensional data onto a lower-dimensional space and SVD compress the least dominant factors, tensor methods maintain the intrinsic multi-way structure of the data. This is particularly useful in deep learning, where tensor decomposition techniques can be employed to optimise model performance, reduce the number of parameters, and enhance computational efficiency, as seen in applications like Blind Source Separation.

The mathematical foundation of tensorization is rooted in spaces that extend beyond basic Euclidean geometry, such as Riemannian and Hilbert spaces, which facilitate advanced geometric and algebraic operations on data.

Table 1 summarises the foundation of multilinear geometric spaces, ML applications, and metrics for analysis. These concepts are crucial for developing and applying manifold learning algorithms, such as t-SNE and UMAP, as well as for understanding the underlying structures of complex datasets. Moreover, tensor methods are pivotal in modern machine learning applications, including graph neural networks, variational autoencoders, and generative models like Wasserstein GANs. They enable more accurate and efficient data processing by preserving the inherent multidimensional relationships within the data.

Table 1
Multilinear Algebra & Tensors

Mathematical Space	Geometric Space Description	ML Applications	Similarity or Distance Measure
Euclidean Space	Standard Cartesian coordinates define flat space.	Feature space for various algorithms, including k-NN, SVM, and linear models.	Euclidean distance: $d(x, y) = \sqrt{\sum (x_i - y_i)^2}$
Manifolds	Generalised spaces that locally resemble Euclidean space but can have complex global structures.	Manifold learning, t-SNE, UMAP, LLE.	Geodesic distance (shortest path on the manifold).
Hilbert Space	Complete, infinite-dimensional inner product space.	Kernel methods in SVM, PCA in high-dimensional spaces, and quantum computing.	Inner product: $\langle x, y \rangle = \sum x_i y_i$.
Curves using Hyperbolic and Elliptic Geometry	Hyperbolic: negatively curved space. Elliptic: positively curved space.	Hyperbolic embeddings for hierarchical data, elliptic geometry for spherical data.	Hyperbolic 2d distance using the Poincaré disk model: $d(u, v) = 2 \operatorname{arsinh}(\frac{\ v-u\ }{2\sqrt{y_1 y_2}})$ where Euclidean

			distance $\ v - u\ = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ or for 3D $d(u, v) = 2 \operatorname{arsinh}(\frac{\ v-u\ }{2\sqrt{z_1 z_2}})$, or Elliptic distance on a sphere of radius R such as $d_E(u, v) = R \cdot \cos^{-1}(\frac{u \cdot v}{R^2})$.
Riemannian Geometry	Study of smooth manifolds with Riemannian metrics, describing how distances and angles are measured.	Riemannian manifold optimisation, GCNs.	Riemannian distance: involves integrating the metric tensor along a curve between points u, v on Manifold M with g Riemannian metric as the infimum of the lengths of all smooth curves connecting u and v along parameter t as $\gamma(0)=u$, $\gamma(1)=v$, and $\dot{\gamma}(t)$ is the tangent vector to the curve γ at t : $d_g(u, v) = \int_u^v \sqrt{g_{\gamma(t)}(\dot{\gamma}(t), \dot{\gamma}(t))} dt$
Differential Geometry on Manifolds	Study of curves, surfaces, and their higher-dimensional analogues using calculus.	Advanced manifold learning, optimisation on curved spaces.	Geodesic distance, curvature-based measures.

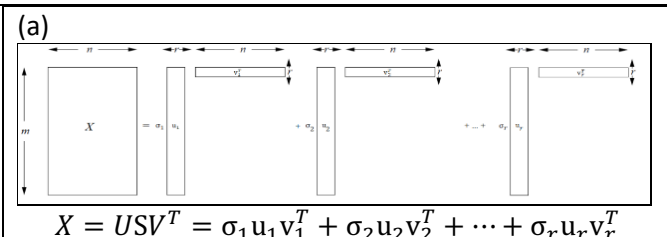
3. Survey of Existing Approaches

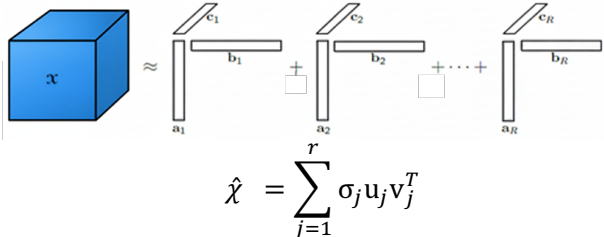
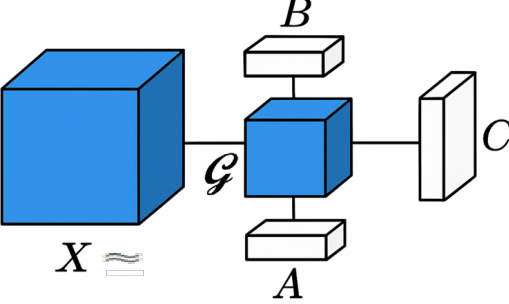
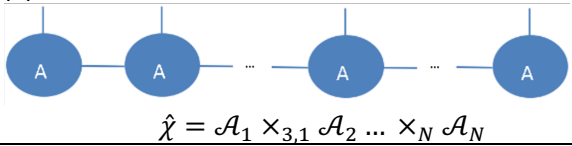
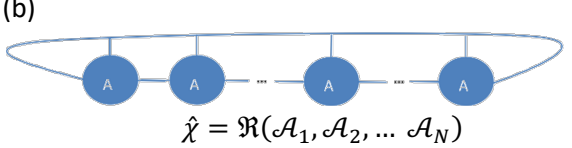
This section explores various methodologies for multi-way analysis and their applicability to tensorized datasets. The first subsection provides an overview of multiway analysis methods, drawing from foundational research and notable surveys, including [1] and chapter four in [2]. Following this, we explore tensorization techniques for both 2-dimensional and multiway datasets.

3.1 Multiway Analysis Methods

Multiway analysis methods encompass a variety of tensor-based algorithms, such as factorisation, regression, clustering, and completion, that analyse data across multiple dimensions (modes). These algorithms provide insights similar to those offered by conventional machine learning methods, such as PCA and SVD. Still, they can also serve as a pre-processing step for tensorized or non-tensorized machine learning (ML) and deep learning (DL) models.

Tensor decompositions are essential in reducing the dimensionality of tensors and identifying dominant factors within them. While methods such as SVD and PCA are widely used, they often fail to capture the nonlinear structure of the data. In contrast, techniques such as Multidimensional Scaling (MDS), Isomap, Locally Linear Embedding, and Spectral Clustering preserve or learn the nonlinear manifold of the dataset. The following tensor decomposition methods excel at capturing complex interactions within high-dimensional datasets:

Tensor decomposition method	Illustration
Candecomp/Parafac (CP) Decomposition: As a multiway extension of SVD, CP decomposition factorises a tensor into a sum of rank-one tensors, enabling the reconstruction of the original tensor from its dominant components. This method generalises the SVD approach to N-	 <p>(a)</p> $X = USV^T = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \dots + \sigma_r u_r v_r^T$

<p>dimensional tensors, capturing the essential structure of the data.</p>	<p>(b)</p>  <p>Fig. 1 : (a) 2-D SVD, (b) 3-D SVD / CP generalises to the higher dimension [1]</p>
<p>Tucker Decomposition: Known as a higher-order PCA, Tucker decomposition decomposes a tensor into a core tensor, which is then multiplied by factor matrices along each mode. Unlike CP decomposition, Tucker retains a richer structure within the core tensor, allowing for more nuanced data representation.</p>	 <p>Fig. 2: 3D Tucker Decomposition</p>
<p>Tensor Networks: Tensor networks hierarchically represent large-scale tensors using lower-rank core tensors. Common approaches include Tensor Train (TT), Tensor Ring (TR), and Matrix Product States (MPS), among others. These methods are particularly effective for handling large-scale data by reducing the computational complexity associated with high-dimensional tensors.</p>	<p>(a)</p>  <p>(b)</p>  <p>Fig. 3: (a) TT decomposition, (b) TR decomposition</p>

Tensor Completion: Tensor completion extends matrix completion techniques to multi-dimensional data, aiming to interpolate missing values within a tensor. Methods like Tensor Decomposition with Relational Constraints (TDRC) enhance traditional tensor decomposition by incorporating auxiliary data, such as similarity matrices, to improve prediction accuracy in applications like miRNA-disease association studies.

Tensor Regression: Tensor regression models generalise linear regression to Nth-order tensors as $\mathcal{Y} = f(\mathcal{X}) + \epsilon$, allowing for the mapping of high-dimensional predictors to target variables. Techniques like CP and Tucker regression reduce the number of parameters required, making it feasible to work with large, multidimensional datasets, such as MRI scans, while preserving the inherent structure of the data.

Tensor Clustering: Tensor clustering is an unsupervised learning approach that identifies clusters within tensor data X by factorising the data matrix into a canonical basis vector A , in which each row selects a row in B , which contains the clustering vectors, $X \approx AB^T$. Methods exist to estimate the two unknowns (A and B) from X , such as Independent Component Analysis (ICA) and dictionary learning algorithms adapted to handle tensor data. These algorithms facilitate the discovery of underlying patterns within high-dimensional datasets.

3.2 Multiway (Tensorized) Dataset Sources

This section explores sources of tensorized data, including traditional datasets that can be transformed into tensor form and naturally occurring multiway datasets.

Traditional Datasets: Public datasets from platforms such as Kaggle and UCI can be tensorized using data fusion techniques. Understanding the different modes within these datasets allows for the integration or segmentation of data to meet specific application requirements, such as combining outcomes from different hospital trials.

Graphs and Networks Datasets: High-dimensional datasets such as Wireless Sensor Networks (WSN) and Knowledge Graphs are naturally suited for tensor representation. For example, WSN data can be represented as a tensor capturing sensors, base stations, clusters, and messages. At the same time, Knowledge Graphs can be modelled as tensors with modes representing entities and relationships.

Image and Video Datasets: Image and video datasets, such as MNIST and video files, benefit significantly from a tensor representation. Tensorizing these datasets preserves spatial and temporal information, enabling more efficient processing in convolutional neural networks (CNNs) and other tensor-based models.

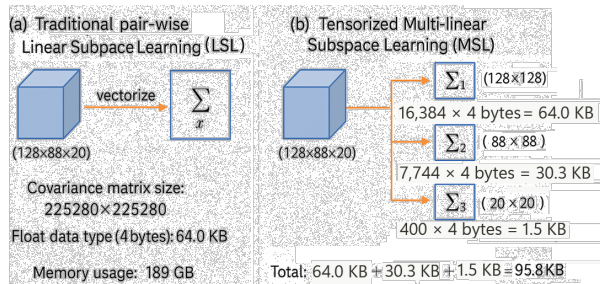


Fig. 4: (a) pair-wise approaches flatten datasets vs (b) tensorised data approaches compression example [3]

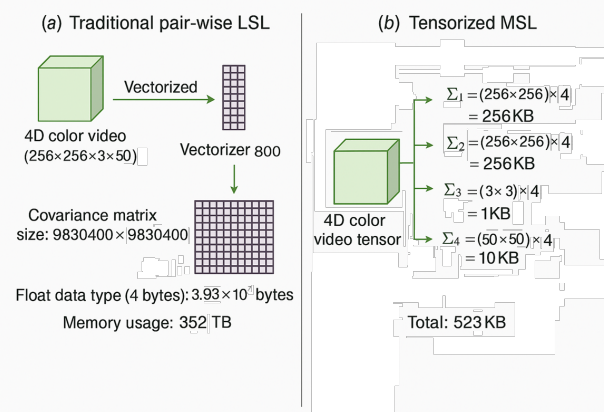


Fig. 5: (a) 4D pair-wise approaches flatten datasets vs (b) tensorised data approaches compression example

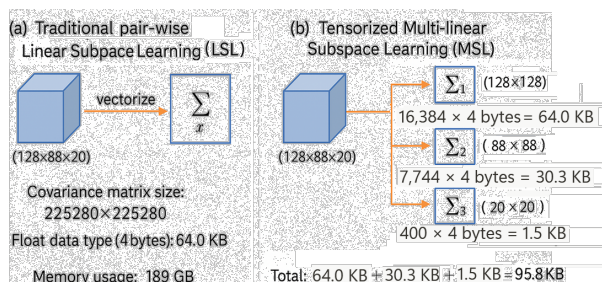


Fig. 4 presents an example of a 3D grey-shade video with an image size of 128 x 88 and 20 frames. Multiplying the shape vector for pair-wise Linear Subspace Learning (LSL) vectorisation in (a) produced a massive 189 GB large covariance matrix in memory and the required computational time. Applying a tensorial Multi-linear Subspace Learning (MSL) in (b) required summing three smaller covariance matrices, producing 95.8KB of memory and is both memory and computational-time efficient [3]. **Fig. 5** illustrates that as dimensionality increases in a higher-resolution 4D colour video tensor with a colour channel of shape 256×256×3×50, resulting in a 73-fold savings.

Health and Biomedical Datasets: Biomedical datasets, such as EEG signals and MRI scans, are inherently multidimensional. These datasets can be effectively analysed using multiway data analysis techniques, which capture the complex interactions between modes, such as time, subjects, and experimental conditions.

3.3 Tensorization Methods

Tensorization is the process of transforming traditional matrix-form datasets into multi-dimensional tensors. This section discusses various deterministic and stochastic methods for tensorizing data, depending on the analysis objectives and the nature of the original data.

Reshaping a dataset involves converting its structure, often from a matrix form to a tensor, to capture the underlying relationships among its variables more effectively. This process can include multiple indexing, pivot table transformations, or tensorization, which involves mapping data from lower to higher dimensions. For instance, student grades across various subjects, years, and exams can be transformed from a matrix into a 4th-order tensor, enabling more sophisticated analysis of trajectories across students, subjects, or time periods. The fusion of multiple data sources is another method for reshaping data. The process integrates data from multiple sources/modalities to create a comprehensive representation, leveraging the multi-way nature of tensors.

Sparse Representation: Not all data fills every possible combination of indices in a tensor. Sparse tensors, where most elements are zero, are crucial for efficiently handling high-dimensional data. Consider a 4th-order tensor where modes represent users, items, interaction types (e.g., view, click, purchase), and timestamps. In this tensor, most entries are likely to be zeros or nulls because most users interact with only a small subset of the available items, engage in a limited number of interaction types, and only at specific times. Compared to dense representations, sparse tensor structures can drastically reduce memory usage and computational requirements. Sparsity can be addressed by incorporating regularisation techniques like dropouts, using sparse tensor representations from scratch, and building ML packages that accept these representations. Various libraries, such as the 2-dimensional *scipy.sparse* Python package does not scale to sparse nd-arrays. A dictionary data structure can capture the n-dimensional indices tuple as key and values as a list of all aggregated features.

Quantisation: Quantisation reduces the precision of continuous variables, which can simplify tensor computations without significantly affecting accuracy. For instance, instead of using precise dates, temperature data might be aggregated by year or month to reduce the tensor size, making processing faster and less memory-intensive.

Parallelisation: When tensors become large, parallelisation techniques are employed to distribute the computational load across multiple processors. By partitioning tensors and distributing these partitions across a cluster of nodes, large datasets can be processed more efficiently. Each node works on a specific partition, enabling simultaneous computation, which is crucial in large-scale tensor operations. For example, parallelisation that is invariant of shape and dimension is applied for distributed processing on a cluster of computing nodes using a high-dimensional wavefront, which

was applied to the Multiple Sequence Alignment problem [4], [5], [6]. Tensors were dynamically created for various sequences in a linear memory array. Each computing node accesses its assigned dense partitions from a specific index up to the partition size, applying wave-front parallelism with the dependency-aware distribution.

Deterministic tensorisation refers to systematic methods that convert data into higher dimensions in a predictable manner, facilitating the application of reverse processes, like detensorisation, to reduce dimensions when needed. Techniques such as Hankelization and Löwnerization are examples, practical in fields like signal processing and telecommunication for harmonic retrieval and direction-of-arrival estimation [7].

Statistical tensorisation leverages statistical measures, such as covariance matrices, to structure data along specific modes. Higher-order statistics, such as cumulants and moments, offer a more comprehensive analysis of non-Gaussian datasets with independent variables. These methods are beneficial in applications such as Blind Source Separation (BSS), where they aid in identifying and separating latent variables within the data. Tensor-based methods for BSS, such as TenSOFO and TCBSS, offer a novel approach to solving BSS problems through tensor decomposition [8].

Domain-specific tensorisation techniques apply transformations tailored to particular types of data. For instance, a 3rd-order tensor might be used in signal processing to represent time-frequency data across multiple channels. Transformations like the Short-Time Fourier Transform (STFT) or wavelet transforms enable multi-scale, multi-orientation data representation, which is crucial for detailed signal analysis. Additionally, advanced methods, such as those involving Generalised Characteristic Functions (GCFs), can further enrich tensorisation by incorporating higher-order statistics, leading to more compact and expressive tensor representations [9], [10]. Using tensor network representations, it is possible to super-compress datasets with as many as 1050 entries down to 107 or even lower. These tensorisation techniques are vital for developing efficient machine learning algorithms and deep neural networks that handle complex, high-dimensional datasets with reduced computational and memory requirements.

3.4 Literature Review

Integrating tensorization techniques into machine learning and deep learning models has led to significant advancements across various applications. This section examines key case studies that demonstrate the benefits of tensorization in enhancing model performance, reducing complexity, and improving generalisation across multiple domains.

Tensorization in Artificial Neural Networks (ANNs) has been recognised to reduce the models' complexity as it increases with the addition of layers and neurons, often leading to over-parameterisation [11]. Tensorization offers a solution by compressing neural networks, thus reducing the number of parameters while maintaining or even improving model performance. For instance, tensorized activation functions, such as recursive neurons, can be used to compute weighted sums recursively within tree structures, effectively managing the hierarchical complexity of data [12]. Techniques such as CP decomposition or Tensor Train (TT) decomposition further refine this process by breaking down tensor aggregations, enabling a more efficient and structured approach to model learning. This compression is particularly evident when tensor decomposition algorithms, such as tensor networks, are applied to the weight tensors of an ANN, leading to shallower networks with fewer layers yet retaining high performance. Replacing specific layers in a model with tensor decomposition layers, such as TT layers, enables the model to capture latent variables effectively, optimising the learning process [13], [14], [15].

Tensorization in Machine Learning Models is outlined in a range of studies. In data warehousing and business intelligence, tensor decomposition methods have been utilised for processing large data cubes, significantly enhancing the efficiency of online analytical processing (OLAP) systems [16]. In signal processing, tensor methods have enhanced blind source separation (BSS) by improving the reconstruction of signals through techniques such as Hankelization [7] and Bayesian Tucker decomposition [17]. These tensor-based approaches outperform traditional methods, particularly in capturing the complex relationships inherent in multidimensional data. A bioinformatics application of tensorization advanced binary classification tasks, such as predicting miRNA-disease associations, by converting 2D datasets into multi-way tensors through the association with other available datasets and the application of tensor completion techniques [18]. Similarly, in social network analysis and semantic data mining, tensor models have been used to analyse the evolution of user interactions over time, allowing for identifying patterns and relationships that are not easily captured by conventional methods [19], [20]. For instance, tensor-based methods like Tucker decomposition and CP decomposition have been employed to analyse temporal knowledge graphs, predicting new links and proposing ontological terms more accurately than traditional approaches [21], [22].

Tensorization in Deep Learning Models, particularly in convolutional neural networks (CNNs) and recurrent neural networks (RNNs), has significantly advanced model compression and performance. For example, TensorNet, a CNN model utilising TT layers, substantially reduced the number of parameters, compressing the network size by a factor of 7 without sacrificing accuracy [14]. Similarly, hierarchical Tucker (HT) tensor formats have been used to compress CNNs, maintaining high accuracy while significantly reducing model size [23]. In RNNs and LSTM networks, tensorization techniques such as TT and HT have been demonstrated to compress models, although with varying impacts on accuracy. For instance, TT-LSTM models have shown better suitability for CNN compression, while HT formats offer higher compression ratios for RNNs, albeit with some loss in accuracy [24]. In natural language processing (NLP), tensor-based models have outperformed traditional deep learning models in tasks such as sentiment analysis and event prediction. Recursive Neural Tensor Networks (RNTNs) and tensor-based attention mechanisms have effectively modelled semantic relationships and context within language data [25]. Moreover, tensorization in transformers, mainly through block-term decomposition (BTD), has enhanced language modelling and neural translation tasks, achieving higher compression and performance than standard transformer models [26]. Multi-modal Visual Question Answering (VQA) used tensors to fuse visual and textual representations, outperforming the bilinear models based on the outer product and its massive parameters [27]. Also, for graph transformation, graph tensors learn embeddings of time-varying graphs based on a tensor framework [28]. A recent survey bridges the connections between tensor networks, neural networks, and quantum circuits [29]. Recent work includes a novel variational DMRG-inspired training algorithm for TNNs, a significant methodological advancement [30], Deep Tree Tensor Network (DTTN) leveraging parameter sharing for image recognition [31]. More case studies are summarised in Supplement A.

Tensorized Neural Networks (TNNs) are an underexplored territory that encompasses not only compressing neural networks but also enhancing their interpretability and exploring the role of "bond indices," which reveal new degrees of freedom within the tensorized neural network layer, thereby providing a novel latent space and a richer hyper-parameter space not found in conventional networks. The benefits of adopting this class of neural networks include the incorporation of inductive bias, benefiting from symmetry, as seen in convolutional neural networks that employ translation equivariance, stacking views to aid interpretation, and acceleration in both the forward and backwards passes. The challenges of adopting TNNs include the limited hardware and software

supporting them, the incomplete physics-aware packages that can guide the inductive bias intuitively, and the fine-tuning of the complex hyper-parameter space [32].

Python Packages for Tensorization have been developed to facilitate the implementation of tensor-based methods in machine learning and deep learning. These packages offer various functionalities, from tensor decomposition and regression to neural network layers optimised for tensor operations. Notable examples include Tensorly [33], which supports multiple tensor decomposition techniques such as CP and Tucker, and TensorNetwork, a package developed by Google for advanced tensor algebra operations. Other packages like scikit-tt [34], scikit-tensor [35], and TensorNet-TF [36] provide specialised tools for implementing tensor methods in different domains, enabling researchers and practitioners to leverage the power of tensorisation in their models. Supplement B lists more packages.

In conclusion, tensorization has proven to be a powerful tool in both machine learning and deep learning, offering significant benefits in terms of model compression, performance enhancement, and the ability to capture complex relationships within data. The successful application of tensorization across various case studies underscores its potential to transform traditional approaches, yielding more efficient and effective models in diverse applications.

3.5 Proposed Framework

To address the complexity of implementing tensorization in deep learning applications, we propose a comprehensive six-step framework that provides systematic guidance from data assessment to performance evaluation, as illustrated in **Fig. 6**. This framework integrates recent advances in tensor networks and provides clear decision criteria for method selection at each stage. The framework begins with Data Assessment, where the natural tensor structure of the dataset is analysed. For instance, MNIST images ($28 \times 28 \times 1$) exhibit clear spatial structure, colour videos ($256 \times 256 \times 3 \times 50$) contain spatial, spectral, and temporal dimensions, while EEG signals present multi-channel temporal patterns. The assessment phase establishes decision criteria that guide subsequent tensorization choices based on data characteristics such as sparsity, dimensionality, and inherent structure.

The Tensorization Strategy selection follows a decision-tree approach: sparse data structures benefit from sparse representation techniques, and temporal data leverages statistical tensorization methods using higher-order statistics. In contrast, spatial data typically employs reshaping strategies that preserve geometric relationships. This systematic approach ensures that the chosen tensorization method aligns with the underlying properties of the data. Decomposition Selection involves choosing among established tensor factorisation methods based on application requirements. CP decomposition ($\mathcal{X} \approx \sum_i \lambda_i \mathbf{a}_i \circ \mathbf{b}_i \circ \mathbf{c}_i$) provides interpretable factors suitable for applications requiring clear component separation. Tucker decomposition ($\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$) offers richer representations through its core tensor, while Tensor Train decomposition ($\mathcal{X} = \mathbf{G}_1 \times_1 \mathbf{G}_2 \times_2 \dots \times_n \mathbf{G}_n$) enables extreme compression for high-dimensional data. The selection depends on the trade-off between compression efficiency, computational complexity, and interpretability requirements. The Model Architecture phase integrates the selected tensor decomposition into neural network structures, whether through the replacement of tensor layers. These hybrid architectures combine tensor and conventional layers, or end-to-end tensorization. The Training Protocol encompasses optimisation algorithms, rank adaptation strategies, and convergence criteria designed explicitly for tensorized models. Finally, Performance Evaluation provides a comprehensive assessment across multiple dimensions, including compression ratio, accuracy, memory usage, and interpretability scores, with feedback loops enabling iterative refinement of the entire process. This framework

addresses the practical challenges of tensorization implementation while incorporating recent theoretical advances, providing researchers and practitioners with a systematic approach to leverage the benefits of tensor methods in deep learning applications.

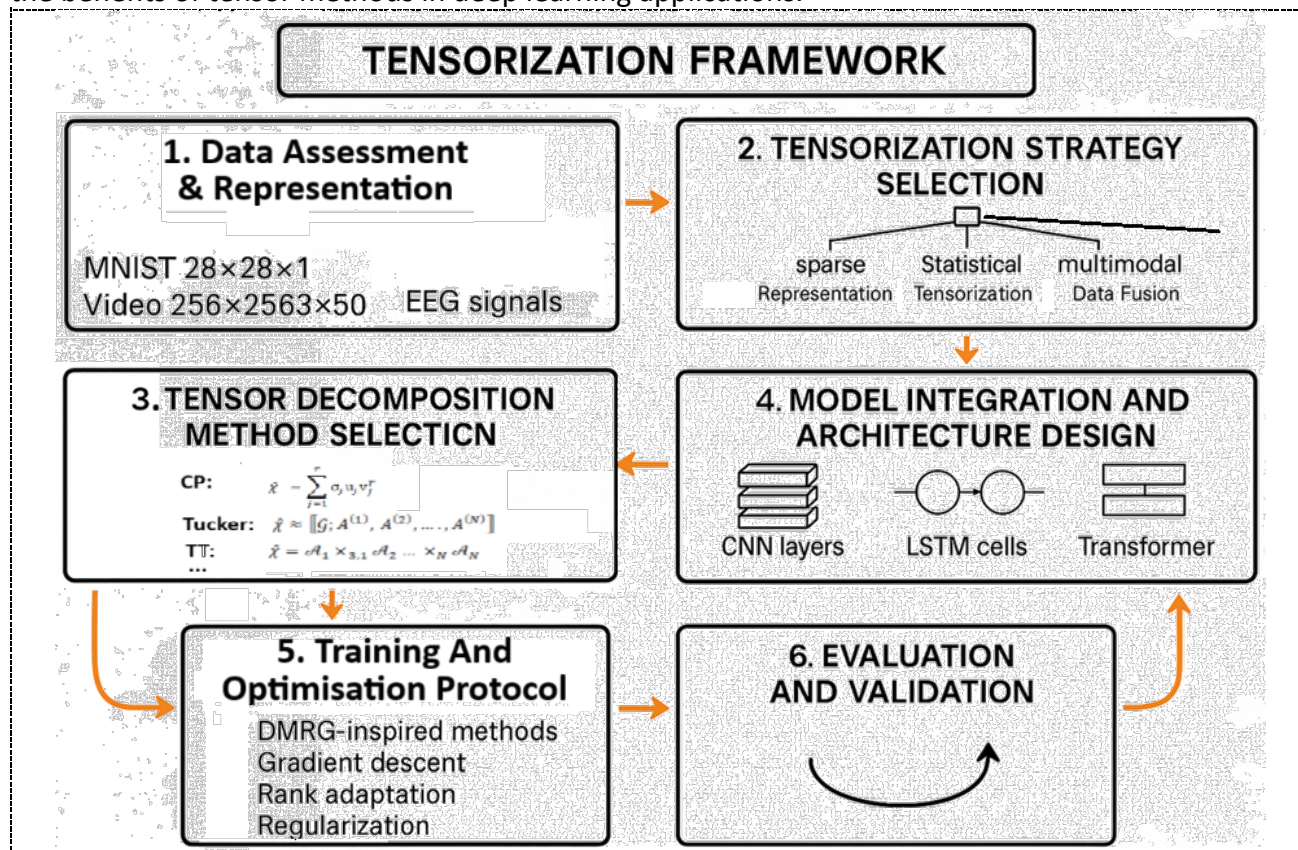


Fig. 6. Tensor Computing Framework

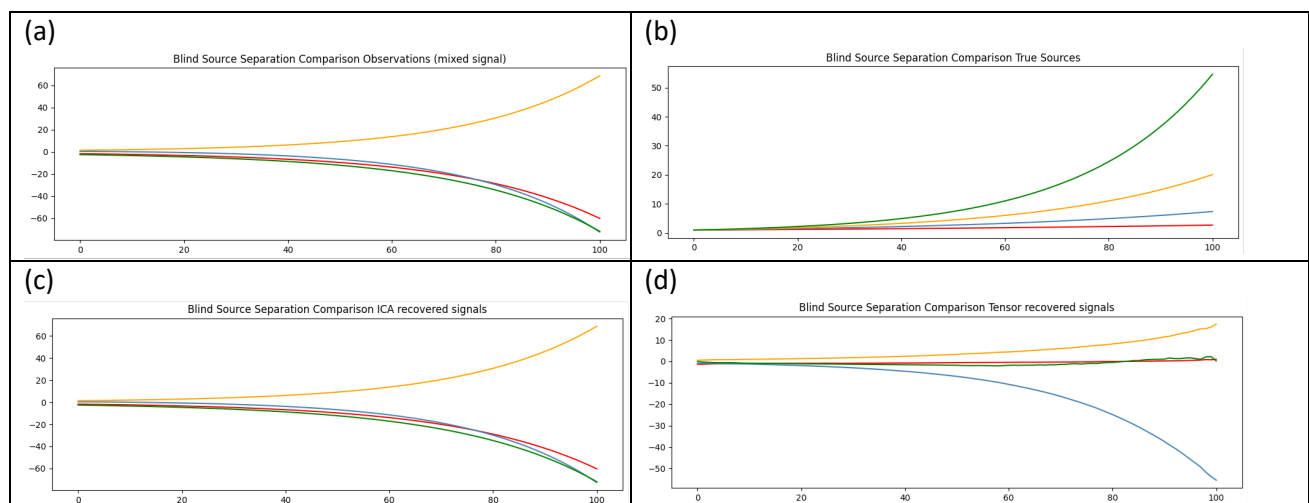
4. Experimental Setup and Results

To evaluate the proposed tensorization framework, various datasets from different domains were selected, including four synthetic signals of small lengths (101 time steps), four audio signals of varying lengths (117601), and MNIST images of 784 flat pixels, to ensure robust and comprehensive testing. The datasets underwent standardised preprocessing steps, including normalisation, denoising, and segmentation, to maintain consistency across experiments. The framework's effectiveness was tested under various conditions, including noise robustness, signal types, real-time processing, and parameter sensitivity. Controlled noise was introduced at varying levels and different types to assess the framework's robustness in signal reconstruction. Parameter sensitivity analysis included testing for other ranks. However, further fine-tuning of all involved parameters may enhance the results. Baseline comparisons with fundamental blind source separation approaches, including Independent Component Analysis (ICA), Principal Component Analysis (PCA), Discrete Wavelet Transform (DWT), and Non-negative Matrix Factorisation (NMF), provided a comprehensive evaluation. The evaluation of the Parafac multi-way decomposition methods employed the Hankelization of the signals. Given an exponential signal $f(k) = az^k$ Hankelization constructs a Hankel matrix H , where each descending diagonal is constant, leading to a matrix of rank one for simple exponentials. This framework generalises to exponential polynomials for applications such as harmonic retrieval and function approximation, analogous to Taylor series expansions. Then, after the reconstruction of the Parafac weights, dehankelisation was applied to retrieve the separated

sources in their original form. Experiments were carefully structured to include multiple repetitions, with the results averaged to account for variability. Tailored adjustments were made for specific problem requirements. In quantitative evaluation metrics, such as root mean square error (RMSE), lower values indicate better reconstruction accuracy; however, they do not directly measure the quality of the reconstruction. We evaluated the reconstruction using other metrics, such as the Structural Similarity Index (SSIM), correlation coefficients, and Frequency Domain Analysis, where higher values indicate a better structural quality of the reconstruction.

The results illustrated in

Fig. 7 shows the shape of the observed mixtures in (a), and the authentic sources are in (b). The ICA is structurally the most distant (c), the PCA is closer (e), and the tensor-based is the nearest (d) in reverse order of the reconstruction error. This may be due to the different scaling and signal permutations of each method. Further tests for three noise levels and reduced rank vs full rank revealed more insights. All methods suffered from performance degradation as noise levels increased, and increasing the rank did not consistently improve performance, particularly for PCA and NMF. ICA was the most effective technique, yielding the lowest Root Mean Square Error (RMSE) of 2.93 on synthetic data and 0.088 for audio files, while maintaining structural integrity in low-noise environments. NMF showed promise in retaining structural features, with an SSIM of 0.68 in synthetic data and -0.00029 for sound files. However, it struggled with accuracy in high-noise conditions, yielding higher RMSE values compared to the other methods. DWT demonstrated poor performance in high noise levels, particularly in terms of frequency similarity and SSIM, highlighting its limitations in noisy environments. Conversely, despite achieving the highest RMSE reconstruction errors, the Hankel and multiway methods attained the highest frequency similarity of 196168 in synthetic data and 6271.8 for sound files, leading to an overall positive correlation of 0.94 in synthetic data and -0.2 for audio files, suggesting that it preserves frequency characteristics well, which is crucial for specific applications like audio analysis. When structurally similar but with a negative correlation (e.g. -0.50), this suggests that while the method captures frequency features, it may not accurately reflect the overall trends of the original signal. This could indicate potential phase shifts or distortions in the reconstructed signal. All results metrics are available in Supplement C.



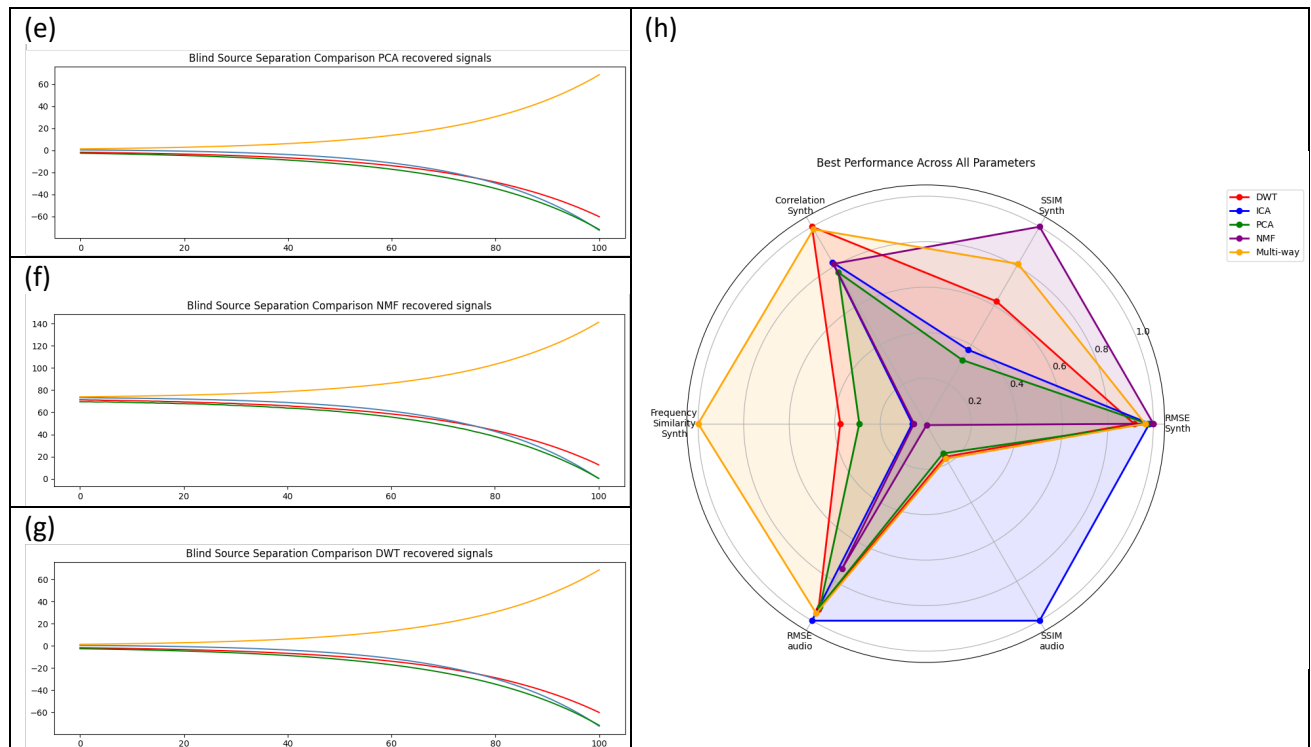


Fig. 7. (a) Observed Mixed Signals, (b) True Signals, (c) BSS reconstructed signals from ICA, (d) multiway compared, (e) PCA, (f) NMF, (g) DWT (h) RMSE, SSIM, Correlation and Frequency Similarity Metrics of all methods

Despite the high RMSE, the Hankel method achieved reasonable SSIM and correlation scores, particularly in audio data under low-noise conditions. This indicates its potential usefulness in specific scenarios, particularly where frequency retention is prioritised. Accurately capturing frequency content can be more important than perfectly reconstructing the time-domain signal if a specific task or analysis emphasises frequency preservation, such as in signal processing system identification in vibration analysis or fault detection. The multi-way methods could be precious despite their lower scores in other performance metrics.

5. Conclusion

This manuscript presents a comprehensive, tutorial-style survey that contrasts traditional Linear Subspace Learning (LSL) with Multi-linear Subspace Learning (MSL), tracing the journey from the mathematical foundations of multilinear algebra to their concrete applications in modern deep learning. We have elucidated how the inherent multi-dimensionality of data, often flattened into 2D matrices for conventional algorithms, can be more naturally and effectively processed using tensor decompositions (e.g., CP, Tucker, Tensor Train). This multi-way approach offers a fundamental shift in perspective, moving beyond the limitations of vector-space methods to capture the rich, complex interactions between data modes.

A central contribution of this work is the empirical perspective on model interpretability, demonstrated through a Blind Source Separation (BSS) case study. Our experiments revealed a critical and nuanced trade-off: while traditional 2D methods, such as ICA and PCA, achieved superior Root Mean Square Error (RMSE), tensor-based approaches, particularly those employing Hankelization and PARAFAC decomposition, excelled in preserving the essential structural (SSIM) and

frequency characteristics of the original signals. This finding underscores that RMSE alone is an insufficient metric for evaluating signal reconstruction quality. It highlights the value of tensor-based methods in applications where the integrity of the signal's structural and spectral properties is paramount, such as in audio analysis, vibrational system identification, and biomedical signal processing. The ability of multi-way analysis to maintain these characteristics, even under noisy conditions, points towards more interpretable and physically meaningful models.

5.1 Limitations of this Work

While this study establishes the foundational benefits of tensorization, it is not without limitations. The scope of the BSS experiment, although illustrative, was limited to a specific set of algorithms and datasets. A more comprehensive benchmark incorporating a broader range of tensor decompositions (e.g., Block-Term Decomposition) and contemporary deep learning baselines (e.g., transformer-based sequence models) would provide a more rigorous comparison. Furthermore, the computational complexity of tensor operations, especially for very high-order or large-scale tensors, remains a practical challenge that was not exhaustively analysed. Finally, the current implementation relied on stitching together various Python libraries, highlighting the lack of a standardised, end-to-end framework for tensorized deep learning, which can hinder reproducibility and adoption.

5.2 Future Work

Building on the insights and limitations of this work, several promising avenues for future research emerge:

Development Environments and Optimisation: Future work must focus on the seamless integration of tensor operations into mainstream deep learning frameworks (e.g., PyTorch, TensorFlow). This includes developing optimised Tensorised Layer types, tensorised activation functions, and efficient backpropagation algorithms, such as SGD with DMRG algorithms, AutoDiff [37] and DDSP (differentiable digital signal processing) [38] compatible with automatic differentiation. Leveraging and extending low-level libraries (e.g., tensor-aware BLAS operations) for variable-order tensors is crucial for achieving optimal performance on parallel hardware, such as GPUs and TPUs.

Advanced Hybrid Architectures: The integration of tensorized models with other advanced neural architectures presents a fertile ground for innovation. Exploring tensorized Graph Neural Networks (GNNs) for relational data, tensorized transformers for long-range dependencies, and Physics-Informed Tensor Networks (PITNs) for embedding domain knowledge directly into the model structure are compelling directions. These hybrids could unlock new levels of efficiency and interpretability in scientific machine learning [39].

Quantum-Tensor Synergy: As quantum computing advances, the synergy between tensor networks and quantum algorithms will become increasingly important. This is seen in recent studies, such as a variational quantum algorithm for singular value decomposition (VQSVD) and the generalisation of quantum ML algorithms that can be tensorized on quantum platforms [40]. Hybrid Tree Tensor Networks (HTTNs) offer a pathway to simulate quantum systems beyond the limits of current hardware [41]. Research into tensor network-inspired quantum machine learning models and quantum-enhanced tensor decomposition algorithms represents a frontier at the intersection of these two transformative fields. This field is considered so pivotal that a recent landmark review in *Nature Reviews Physics*, co-authored by researchers from NVIDIA, Google, NASA, and others, outlines a strategic roadmap. It positions tensor networks as the backbone for progress in key areas such as

quantum error correction, quantum circuit design, and enhancing quantum machine learning models by making them more efficient and interpretable [42].

Standardised Benchmarking: To accelerate progress, the community would benefit greatly from establishing standardised challenges and benchmarks with clear, multi-faceted metrics that go beyond RMSE to assess expressiveness, structural fidelity, computational efficiency, and parameter compression. This will enable meaningful and fair comparisons across different tensorization methodologies.

In conclusion, this survey has articulated the transformative potential of tensorization for machine learning. By bridging the gap between the theoretical elegance of multi-linear algebra and the practical demands of deep understanding, we pave the way for a new generation of machine learning systems that are not only more efficient and scalable but also more expressive and interpretable. This is particularly critical for deploying advanced models in resource-constrained environments (e.g., IoT devices) and in scientific domains where model trust and physical plausibility are as crucial as predictive accuracy. The journey from matrices to multi-way arrays is an essential step in mastering the complexity of modern data.

Acknowledgement

This research was not funded by any grant.

References

- [1] Kolda, T. G. and Bader, B. W., 'Tensor Decompositions and Applications'. *SIAM Review* 51, (3) p. 455–500 2009. <https://doi.org/10.1137/07070111X>
- [2] Helal, M., - *Introduction to Tensor Computing in Python, from first principles to Deep Learning*. (2023).
- [3] Lu, H., Plataniotis, K. N., and Venetsanopoulos, A. N., 'A survey of multilinear subspace learning for tensor data'. *Pattern Recognition* 44, (7) p. 1540–1551 2011. <https://doi.org/10.1016/j.patcog.2011.01.004>
- [4] Helal, M., El-Gindy, H., Gaeta, B., and Sinchenko, V., 'High Performance Multiple Sequence Alignment Algorithms for Comparison of Microbial Genomes'. In , Gold Coast, Australia (2008)
- [5] Helal, M., Mullin, L., Potter, J., and Sintchenko, V., 'Search Space Reduction Technique for Distributed Multiple Sequence Alignment'. In , *2009 Sixth IFIP International Conference on Network and Parallel Computing*, Gold Coast, Australia p., 219–226 (2009). <https://doi.org/10.1109/NPC.2009.43>
- [6] Helal, M., 'INDEXING AND PARTITIONING SCHEMES FOR DISTRIBUTED TENSOR COMPUTING WITH APPLICATION TO MULTIPLE SEQUENCE ALIGNMENT', PhD, University of New South Wales, Sydney, Australia, Aug. 2009. https://www.unsw.edu.au/permalink/f/a5fmj0/unsworks_8078
- [7] Debals, O. and De Lathauwer, L., 'Stochastic and Deterministic Tensorization for Blind Signal Separation'. In , *Latent Variable Analysis and Signal Separation* (E. Vincent, A. Yeredor, Z. Koldovský, and P. Tichavský, Eds) Cham: Springer International Publishing (2015) p.: 3–13. https://doi.org/10.1007/978-3-319-22482-4_1
- [8] Le, T. T., Abed-Meraim, K., Ravier, P., Buttelli, O., and Holobar, A., 'Tensor decomposition meets blind source separation'. *Signal Processing* 221 p. 109483 2024. <https://doi.org/10.1016/j.sigpro.2024.109483>
- [9] Cichocki, A., Lee, N., Oseledets, I., Phan, A.-H., Zhao, Q., and Mandic, D. P., 'Tensor Networks for Dimensionality Reduction and Large-scale Optimization: Part 1 Low-Rank Tensor Decompositions'. *Foundations and Trends® in Machine Learning* 9, (4–5) p. 249–429 2016. <https://doi.org/10.1561/22000000059>
- [10] Cichocki, A., Lee, N., Oseledets, I., Phan, A.-H., Zhao, Q., Sugiyama, M., and Mandic, D. P., 'Tensor Networks for Dimensionality Reduction and Large-scale Optimization: Part 2 Applications and Future Perspectives'. *Foundations and Trends® in Machine Learning* 9, (6) p. 249–429 2017. <https://doi.org/10.1561/22000000067>
- [11] Denil, M., Shakibi, B., Dinh, L., Ranzato, M., and de Freitas, N., 'Predicting Parameters in Deep Learning', Oct. 27, 2014, *arXiv*: arXiv:1306.0543. <http://arxiv.org/abs/1306.0543>
- [12] Bacciu, D. and Mandic, D. P., 'Tensor Decompositions in Deep Learning', Feb. 26, 2020, *arXiv*: arXiv:2002.11835. <http://arxiv.org/abs/2002.11835>
- [13] Yang, Y. and Hospedales, T., 'Deep Multi-task Representation Learning: A Tensor Factorisation Approach'. (2017). <http://arxiv.org/abs/1605.06391>
- [14] Novikov, A., Podoprikin, D., Osokin, A., and Vetrov, D., 'Tensorizing Neural Networks'. *arXiv:1509.06569 [cs]* 28 2015. <http://arxiv.org/abs/1509.06569>

- [15] Calvi, G. G., Moniri, A., Mahfouz, M., Zhao, Q., and Mandic, D. P., 'v'. *arXiv:1903.06133 [cs, eess]* 2020. <http://arxiv.org/abs/1903.06133>
- [16] Spelta, A., 'Financial market predictability with tensor decomposition and links forecast'. *Applied Network Science* 2, (1) p. 7 2017. <https://doi.org/10.1007/s41109-017-0028-1>
- [17] Böttcher, A., Brendel, W., Englitz, B., and Bethge, M., 'Trace your sources in large-scale data: one ring to find them all', Mar. 23, 2018, *arXiv: arXiv:1803.08882*. <http://arxiv.org/abs/1803.08882>
- [18] Huang, F., Yue, X., Xiong, Z., Yu, Z., Liu, S., and Zhang, W., 'Tensor decomposition with relational constraints for predicting multiple types of microRNA-disease associations'. *Briefings in Bioinformatics* 22, (3) p. bbaa140 2021. <https://doi.org/10.1093/bib/bbaa140>
- [19] Acar, E., Çamtepe, S. A., Krishnamoorthy, M. S., and Yener, B., 'Modeling and Multiway Analysis of Chatroom Tensors'. In , *Intelligence and Security Informatics* (P. Kantor, G. Muresan, F. Roberts, D. D. Zeng, F.-Y. Wang, H. Chen, and R. C. Merkle, Eds) Berlin, Heidelberg: Springer Berlin Heidelberg (2005) p.: 256–268. https://doi.org/10.1007/11427995_21
- [20] Acar, E., Çamtepe, S. A., and Yener, B., 'Collective Sampling and Analysis of High Order Tensors for Chatroom Communications'. In , *Intelligence and Security Informatics* (S. Mehrotra, D. D. Zeng, H. Chen, B. Thuraisingham, and F.-Y. Wang, Eds) Berlin, Heidelberg: Springer Berlin Heidelberg (2006) p.: 213–224. https://doi.org/10.1007/11760146_19
- [21] Lacroix, T., Obozinski, G., and Usunier, N., 'Tensor Decompositions for temporal knowledge base completion'. (2020). <http://arxiv.org/abs/2004.04926>
- [22] Nickel, M., Tresp, V., and Kriegel, H.-P., 'A three-way model for collective learning on multi-relational data'. In , *Proceedings of the 28th International Conference on International Conference on Machine Learning*, Madison, WI, USA p., 809–816 (2011)
- [23] Gabor, M. and Zdunek, R., 'Compressing convolutional neural networks with hierarchical Tucker-2 decomposition'. *Applied Soft Computing* 132 p. 109856 2023. <https://doi.org/10.1016/j.asoc.2022.109856>
- [24] Yang, Y., Krompass, D., and Tresp, V., 'Tensor-Train Recurrent Neural Networks for Video Classification'. In , *Proceedings of the 34 th International Conference on Machine Learning*, Sydney, Australia (2017). https://github.com/Tuyki/TT_RNN
- [25] Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C., 'Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank'. (2013)
- [26] Ma, X., Zhang, P., Zhang, S., Duan, N., Hou, Y., Song, D., and Zhou, M., 'A Tensorized Transformer for Language Modeling'. In , *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada (2019). <http://arxiv.org/abs/1906.09777>
- [27] Ben-younes, H., Cadene, R., Cord, M., and Thome, N., 'MUTAN: Multimodal Tucker Fusion for Visual Question Answering'. In , *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice p., 2631–2639 (2017). <https://doi.org/10.1109/ICCV.2017.285>
- [28] Malik, O. A., Ubaru, S., Horesh, L., Kilmer, M. E., and Avron, H., 'Tensor Graph Neural Networks for Learning on Time Varying Graphs'. In , Vancouver, Canada. (2019)
- [29] Wang, M., Pan, Y., Xu, Z., Li, G., Yang, X., Mandic, D., and Cichocki, A., 'Tensor Networks Meet Neural Networks: A Survey and Future Perspectives', Mar. 17, 2025, *arXiv: arXiv:2302.09019*. <https://doi.org/10.48550/arXiv.2302.09019>
- [30] Jahromi, S. S. and Orus, R., 'Variational Tensor Neural Networks for Deep Learning'. *Scientific Reports* 14, (1) p. 19017 2024. <https://doi.org/10.1038/s41598-024-69366-8>
- [31] Nie, C., Chen, J., and Chen, Y., 'Deep Tree Tensor Networks for Image Recognition', Feb. 14, 2025, *arXiv: arXiv:2502.09928*. <https://doi.org/10.48550/arXiv.2502.09928>
- [32] Hamreras, S., Singh, S., and Orús, R., 'Tensorization is a powerful but underexplored tool for compression and interpretability of neural networks', May 26, 2025, *arXiv: arXiv:2505.20132*. <https://doi.org/10.48550/arXiv.2505.20132>
- [33] Kossaifi, J., Panagakis, Y., Anandkumar, A., and Pantic, M., 'TensorLy: Tensor Learning in Python'. *Journal of Machine Learning Research*, (20) p. 1–6 2019. <https://dl.acm.org/doi/10.5555/3322706.3322732>
- [34] Gelß, P., 'Scikit-TT'. (2022). https://github.com/PGelss/scikit_tt
- [35] Nickel, M., 'scikit-tensor Library'. (Nov. 2013). <https://pypi.org/project/scikit-tensor/>
- [36] Garipov, T., Podoprikin, D., Novikov, A., and Vetrov, D., 'Ultimate tensorization: compressing convolutional and FC layers alike'. <https://github.com/timgaripov/TensorNet-TF>
- [37] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A., 'Automatic differentiation in PyTorch'. In , CA, USA p., 4 (2017). <https://openreview.net/forum?id=BJJsrmfCZ>
- [38] Engel, J., Hantrakul, L., Gu, C., and Roberts, A., 'DDSP: Differentiable Digital Signal Processing'. (2020). <http://arxiv.org/abs/2001.04643>

- [39] Ullah Babar, A., 'Physics Informed Neural Networks, A Proven PINNs Guide 2025', *Binary Verse AI*. <https://binaryverseai.com/physics-informed-neural-networks-pinns-explained/>
- [40] Zaman, K., Marchisio, A., Hanif, M. A., and Shafique, M., 'A Survey on Quantum Machine Learning: Current Trends, Challenges, Opportunities, and the Road Ahead', Oct. 16, 2023, *arXiv*: arXiv:2310.10315. <http://arxiv.org/abs/2310.10315>
- [41] Harada, H., Suzuki, Y., Yang, B., Tokunaga, Y., and Endo, S., 'Density matrix representation of hybrid tensor networks for noisy quantum devices'. *Quantum* 9 p. 1823 2025. <https://doi.org/10.22331/q-2025-08-07-1823>
- [42] Berezutskii, A., Liu, M., Acharya, A., Ellerbrock, R., Gray, J., Haghshenas, R., He, Z., Khan, A., Kuzmin, V., Lyakh, D., Lykov, D., Mandrà, S., Mansell, C., Melnikov, A., Melnikov, A., Mironov, V., Morozov, D., Neukart, F., Nocera, A., Perlin, M. A., Perelshtein, M., Steinberg, M., Shaydulin, R., Villalonga, B., Pflitsch, M., Pistoia, M., Vinokur, V., and Alexeev, Y., 'Tensor networks for quantum computing'. *Nature Reviews Physics* 2025. <https://doi.org/10.1038/s42254-025-00853-1>