Adaptive Tensor Networks Decomposition

Chang Nie
changnie@njust.edu.cn
Huan Wang
wanghuanphd@njust.edu.cn
Le Tian
119106021993@njust.edu.cn

School of Computer Science and Engineering, Nanjing University of Science & Technology, Nanjing, China

Abstract

Tensor Decomposition (TD) is a powerful technique in solving high-dimensional optimization problems and has been widely used in machine learning and data science. Many TD models aim to establish a trade-off between computational complexity and representation ability. However, they have the problem of tensor rank selection and latent factor arrangement, and the neglected internal correlation between different modes. In this paper, we propose a data-adaptive TD model established on a generalized tensor rank and name it adaptive tensor network (ATN) decomposition, which constructs an optimal topological structure for TD according to the intrinsic properties of the data. We exploit the generalized tensor rank to measure the correlation between two modes of the data and establish a multilinear connection between the corresponding latent factors with an adaptive rank. ATN possesses the merits of permutation invariance, strong robustness, and represents high-order data with fewer parameters. We verified ATN's effectiveness and superiority on three typical tasks: tensor completion, image denoising, and neural network compression. Experimental results on synthetic data and real datasets demonstrate that the overall performance of ATN surpasses the state-of-the-art TD methods.

1 Introduction

A tensor, also called a multi-dimensional array, is a favorable and powerful data format to represent high-dimension and multi-modal information. Many intrinsic properties about tensors have been revealed in recent years [14, 17]. However, there exists the issue of the curse of dimensionality, i.e. the storage and calculation of a tensor grow exponentially with the increase of its order. One effective way to deal with this issue is the tensor decomposition (TD) technique, which characterizes higher-dimension data as a multilinear operation of some low-order latent factors and has been extensively applied in machine learning [28, 33, 35], signal processing [6, 31], and neural network compression [2, 16, 19, 22].

Many TD models have been proposed in recent years, and the most classical ones are CANDECOMP/PARAFAC (CP) decomposition [9] and Tucker decomposition [34]. The main difference between them and other TD methods lies in the structure and connection of the employed core factors. Tensor network (TN) [4, 30], a new tool for tensor representation, can intuitively and concisely express the connection of cores factors in a graphical format and be easily extended for representing more complex topological structures. Of course, it also benefits for TD. For instance, chain-shaped Tensor Train (TT) [24, 27], ring-shaped Tensor Ring (TR) [40], and square lattice-shaped Projected Entangled Pair States (PEPS) [23], to name a few. TN recovers an original high-order data by contracting the shared index (standard edge connecting two core factors) between the connected factors. It has been received widespread attention recently and successfully applied in supervised learning [29, 32] and probability modeling [7].

Although TN shows its promising characteristics in TD, there still exist three limitations:

1) The tensor network edge rank determination is complicated and cumbersome. Setting each standard edge rank to be identical (e.g., TR [40]) is inappropriate for data with large unevenness in modes dimensions.

2) The exponential decay of correlation in TT and TR weakens the natural correlation between two nonadjacent modes, and determining the arrangement of factors is NP-hard [13].

3) There is no universal TD model that can be applied to arbitrarily complex data, and the model selection itself is difficult.

In this paper, we propose an *adaptive tensor network* (ATN) decomposition to overcome the limitations mentioned above, which can be regarded as a generalized form of existing TD models [23, 24, 40]. ATN transforms the global discrete optimization problem of topology search into a local connection problem of latent factors, measures the internal correlation between modes based on a generalized tensor rank, and establishes a connection for any two factors according to the adaptive rank. Moreover, ATN decomposition minimizes the structural loss function with less number of parameters and is not sensitive to the arrangement of factors. Experiments show that ATN can be used for large-scale tensor optimization problems and achieve excellent performance. The main contributions of this paper can be summarized as follows:

- We defined the generalized tensor rank to characterize the correlation between different modes of the target tensor, and theoretically establish its relationship with the multilinear tensor rank [17].

- We proposed a novel data-adaptive TD model and named it ATN, which characterizes the correlation between modes via the rank of standard edges connecting latent factors, and applies the degree of information retention to manipulate the model complexity.

- We apply ATN to several high-dimensional optimization tasks, including tensor completion [21, 39], image denoising [36], and neural network compression [16, 37]. Experimental results on synthetic data and real datasets demonstrate its effectiveness and superiority.

2 Related Work

The TD technology can efficiently express high-dimensional and multimodal data as a multilinear operation form of a few low-order latent factors, which can be seen as a trade-off between the data structure completeness and model complexity. Many TD models have been proposed in recent years by virtue of tensor networks, and we can divide them into two parts according to the topology, including manual design-based or automatic search-based.

The manual design-based TD models have been extensively studied and provided with well theoretical basis [8, 9, 24, 34, 40]. CP decomposition [9] represents the tensor as the

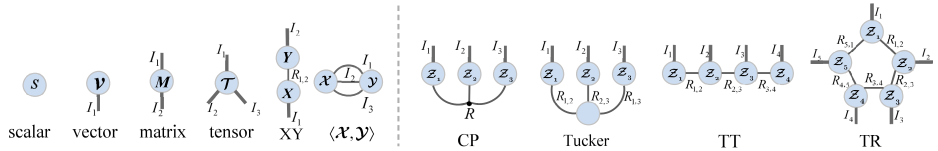


Figure 1: (Left) The graphical representation of scalar, vector, matrix, $3rd$-order tensor, matrix multiplication and tensor inner product. (Right) Tensor network representations of several TD models, including CP, Tucker, TT, and TR. The black dot in CP model denotes a hyperedge [10]. I and R represent the tensor index and standard edge rank, respectively.

sum of r rank-one tensors. Here r is called CP rank, which is the minimum value to make the structure loss vanish. Tucker decomposition [34] expresses the tensor into a core tensor with multiple factor matrices, and Hierarchical Tucker decomposition [8] decomposes the data in a hierarchical form similar to a binary tree. Many researchers have introduced TN to explore more complex TD models. TT [24] decomposes an Nth-order tensor as a chain-shaped contraction form composed of N-2 $3rd$-order tensors and two $2nd$-order tensors. TR [40] establishes the connection between the head and tail factors based on TT, and PEPS [23] is the two-dimensional expansion of TT. The fully-connected tensor network (FCTN) [41] decomposition establishes a connection between any two factors to characterize the correlation of two modes. However, the above TD models only study simple topological structures and have the defects of model selection, limited representation capability and flexibility. Some methods lack of considering rank determination of each edge and its computational complexity grows exponentially with the order of tenors, say FCTN.

Many works began to explore adaptive TD [3, 10, 11, 20], trying to obtain a more flexible topology that matches the properties of the data. The adaptive hierarchical tensor decomposition model was proposed in [3] based on agglomeration strategy and rank-adaptive cross approximation techniques. Li and Sun [20] represented the tensor network as an undirected graph and exploited the genetic algorithm to search the optimal topology iteratively. Meraj et al. [10] employed the simple greedy approach to gradually increase the standard edge rank to improve the model performance. Hayashi et al. [11] applied TD to neural network compression and explored the TN topological space by enumerating all possible decomposition models. The previous automatic search-based TD models increase the computational complexity significantly and are unsuitable for high-dimension data. In this paper, we proposed to transform the problem of topology search into that of correlation estimation of data modes via a generalized tensor rank.

3 Adaptive Tensor Networks Decomposition

3.1 Preliminaries

The order of tensors is the number of dimensions, also called ways or modes, and the operations between tensors are multilinear operations. In this paper, we use boldface calligraphic letters to denote tensors, e.g., $\boldsymbol{\mathcal{X}}$; matrices and vectors are denoted in boldface uppercase and lowercase letters, e.g., $\boldsymbol{X}$ and $\boldsymbol{x}$; scalars is denoted by lowercase letters, e.g., x. We use $[k]$ to represent the set of integers from 1 to k. For the Nth-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1\times\cdots\times I_N}$, we denote $(i_1,i_2,\ldots,i_N)$-th entry as $\boldsymbol{\mathcal{X}}(i_1,i_2,\ldots,i_N)$ or $\boldsymbol{\mathcal{X}}_{i_1,i_2,\ldots,i_N}$, where $i_n \in [I_n]$, $n \in [N]$. For the $3rd$-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1\times I_2\times I_3}$, the horizontal, lateral and frontal slices can be expressed as $\boldsymbol{\mathcal{X}}(i_1,:,:)$, $\boldsymbol{\mathcal{X}}(:,i_2,:)$, and $\boldsymbol{\mathcal{X}}(:,:,i_3)$. We denoted a mode-$k$ fibers as $\boldsymbol{\mathcal{X}}(i_1,\ldots,i_{k-1},:,i_{k+1},\ldots,i_N)$, where all indexes are fixed except index k. The rank of matrix $\boldsymbol{X}$ is defined as $rank(\boldsymbol{X})$. Spectral norm $||\boldsymbol{X}||$ and nuclear norm $||\boldsymbol{X}||_*$ is the largest and the sum of singular values, respectively. The inner product of two tensors $\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}$ is defined as $\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle = \sum_{i_1,\dots,i_N} \boldsymbol{\mathcal{X}}(i_1, i_2, \dots, i_N)\boldsymbol{\mathcal{Y}}(i_1, i_2, \dots, i_N)$, and the Frobenius norm of the tensor is defined as $||\boldsymbol{\mathcal{X}}||_F = \sqrt{\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}} \rangle}$. For more details of the tensor concept see literature [15, 17, 25].

TN can graphically and visually represent tensors, tensor operations, and TD models, as shown in Fig. 1. TN utilizes vertices to represent tensors; edges represent different modes; the number of dangling edges is equivalent to the tensor's order. The shared index dimension corresponds to the standard edges of connected vertices called rank, and the summation along these indexes can complete the tensor contraction.

3.2 ATN decomposition

Supposing $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is an Nth-order tensor, the ATN decomposes $\boldsymbol{\mathcal{X}}$ into a multilinear operation of N factors $\boldsymbol{\mathcal{Z}}^{(1)}, \boldsymbol{\mathcal{Z}}^{(2)}, \dots, \boldsymbol{\mathcal{Z}}^{(N)}$, denoted by $TN(\boldsymbol{\mathcal{Z}}^{(1)}, \boldsymbol{\mathcal{Z}}^{(2)}, \dots, \boldsymbol{\mathcal{Z}}^{(N)})$, where the size of $\boldsymbol{\mathcal{Z}}^{(k)}$ is $R_{1,k} \times \cdots \times R_{k-1,k} \times I_k \times \cdots \times R_{N,k}$ for any $k \in [N]$. Then the $(i_1, \dots, i_N)$-th entry of $TN(\boldsymbol{\mathcal{Z}}^{(1)}, \boldsymbol{\mathcal{Z}}^{(2)}, \dots, \boldsymbol{\mathcal{Z}}^{(N)})$ or $\boldsymbol{\mathcal{X}}$ can be written as:

$$\boldsymbol{\mathcal{X}}_{(i_1,\dots,i_N)} = \sum_{r_{1,2}}^{R_{1,2}} \cdots \sum_{r_{1,N}}^{R_{1,N}} \sum_{r_{2,3}}^{R_{2,3}} \cdots \sum_{r_{2,N}}^{R_{2,N}} \cdots \sum_{r_{N-1,N}}^{R_{N-1,N}} \boldsymbol{\mathcal{Z}}^{(1)}_{i_1, r_{1,2}, \dots, r_{1,N}} \boldsymbol{\mathcal{Z}}^{(2)}_{r_{2,1}, i_2, \dots, r_{2,N}} \cdots \boldsymbol{\mathcal{Z}}^{(N)}_{r_{N,1}, \dots, r_{N,N-1}, i_N}, \quad (1)$$

We call the vector $[R_{1,2}, \cdots, R_{1,N}, R_{2,3}, \cdots, R_{2,N}, \cdots, R_{N-1,N}]^T \in \mathbb{N}_+^{\frac{N(N-1)}{2}}$ as the rank of ATN. For any two factors $\boldsymbol{\mathcal{Z}}^{(i)}$ and $\boldsymbol{\mathcal{Z}}^{(j)}$, where $1 \leq i < j \leq N$, the correspondingly standard edge rank is defined as $R_{i,j}$ and satisfy $R_{i,j} = R_{j,i}$. One particular case is when the standard edge rank $R_{i,j}$ are all equal to 1, and then equation (1) is reduced to:

$$TN(\boldsymbol{\mathcal{Z}}^{(1)}, \boldsymbol{\mathcal{Z}}^{(2)}, \dots, \boldsymbol{\mathcal{Z}}^{(N)}) = \boldsymbol{\mathcal{Z}}^{(1)} \circ \boldsymbol{\mathcal{Z}}^{(2)} \cdots \circ \boldsymbol{\mathcal{Z}}^{(N)}. \quad (2)$$

Here the $\circ$ means outer product. Clearly, ATN is a complete graph composed of N vertices and $\frac{N(N-1)}{2}$ standard edges, and the number of parameters are $\sum_{k=1}^{N} I_k \prod_{i \neq k} R_{i,k}$. Such a topology structure (Fig. 2) of ATN enjoys favorable geometric properties (e.g., permutation invariance [41]) and solves the problem of correlation exponential decay [5] that appears in TT and TR, which severely limits the capability to model complex data.

Definition 1 *(Matricization [17]). Given an Nth-order tensor* $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$*, the mode-k matricization converts the mode-k fibers of* $\boldsymbol{\mathcal{X}}$ *into one column of the resulting matrix* $\boldsymbol{X}_{(k)} \in \mathbb{R}^{I_k \times \prod_{j \neq k} I_j}$*, with entries* $\boldsymbol{X}_{(k)}(i_k, \overline{i_1 \cdots i_{k-1} i_{k+1} \cdots i_N}) = \boldsymbol{\mathcal{X}}(i_1, \cdots, i_N)$*. Here we pay little attention to the arrangement order of the matrix columns for keeping the consistency in relevant calculations.*

Definition 2 *(Tensorization). Given an Nth-order tensor* $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$*, if we divide the set* $[N]$ *into r non-intersect subsets* $s_1, s_2, \dots, s_r$*, then merging the modes which belong to the same subset yields an rth-order tensor* $Ten(\boldsymbol{\mathcal{X}}, \{s_i\}_{i=1}^{r}) \in \mathbb{R}^{P_1 \times \cdots \times P_r}$*, where* $P_i = \prod_{j=1, j \in s_i}^{N} I_j$*. Tensorization decreases the modes of data and keeps the values and number of elements unchanged.*

As mentioned earlier, the rank of ATN determines its topology, and is crucial to the model's representation ability and computational complexity. However, one problem is the lack of a unified rank measurement strategy for TN and rank determination via search strategy [20] is impractical. Since there are various topological structures in different TD models, we propose a new definition for generalized tensor rank and nuclear norm, which can measure the intrinsic correlation between any two modes.

Definition 3 *(Generalized tensor rank and nuclear norm). Given an Nth-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1\times\cdots\times I_N}$, if one employs tensorization (Please see definition 2) to $\boldsymbol{\mathcal{X}}$ to obtain $\boldsymbol{\mathcal{X}}_{(n,m)} = Ten(\boldsymbol{\mathcal{X}}, \{\{n\},\{m\},\{i\}_{i=1,i\notin\{n,m\}}^{N}\})$, then the generalized tensor rank and nuclear norm between the n,m modes are $rank(\boldsymbol{\mathcal{X}}_{(n,m)})$ and $||\boldsymbol{\mathcal{X}}_{(n,m)}||_*$, which are written as:*

$$rank(\boldsymbol{\mathcal{X}}_{(n,m)}) = max\{rank(\boldsymbol{\mathcal{X}}_{(n,m)}(:,:,i))\}_{i=1}^{C}, \quad ||\boldsymbol{\mathcal{X}}_{(n,m)}||_* = \frac{1}{C}\sum_{i=1}^{C}||\boldsymbol{S}_i^{(n,m)}||_*, \tag{3}$$

where the $\boldsymbol{C} = \prod_{i=1,i\notin\{n,m\}}^{N} I_i$ and $\boldsymbol{S}_i^{(n,m)}$ is a diagonal matrix obtained by singular value decomposition [1] of the i-th frontal slice of $\boldsymbol{\mathcal{X}}_{(n,m)}$.

Our proposed generalized tensor rank has some connections with the multilinear tensor rank involved in Tucker decomposition [34] and their relationship is explained in theorem 1.

Theorem 1 *Let Nth-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1\times\cdots\times I_N}$ and $I_1 = \cdots = I_N = I$. The multilinear tensor rank is denoted by $(rank(\boldsymbol{X}_{(1)}),\cdots,rank(\boldsymbol{X}_{(N)}))$, where $\boldsymbol{X}_{(k)}$ is mode-k matricization (Please see definition 1) of $\boldsymbol{\mathcal{X}}$, then the following inequalities holds for $i = 1,\cdots,N$:*

$$\begin{gathered}\min\{N-1, rank(\boldsymbol{X}_{(i)})\} \leq \sum_{j=1,j\neq i}^{N} rank(\boldsymbol{\mathcal{X}}_{(i,j)}) \leq (N-1)rank(\boldsymbol{X}_{(i)}) \\ \max\{rank(\boldsymbol{\mathcal{X}}_{(i,j)})\}_{j=1,j\neq i}^{N} \leq rank(\boldsymbol{X}_{(i)}) \leq I(\frac{\sum_{j=1,j\neq i}^{N} rank(\boldsymbol{\mathcal{X}}_{(i,j)})}{N-1})^{N-2}.\end{gathered} \tag{4}$$

The mode correlation of the reconstructed tensor is established through the connection between the core factors, which allows us to determine the ATN's rank naturally by the definition 3.

Theorem 2 *For Nth-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1\times\cdots\times I_N}$ and Mth-order tensor $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_{N+1}\times\cdots\times I_{N+M}}$, let $\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{X}} \circ \boldsymbol{\mathcal{Y}}$, The generalized tensor rank of $\boldsymbol{\mathcal{Z}}_{(n,m)}$ is equal to 1 consistently for any $1 \leq n \leq N, N+1 \leq m \leq N+M$.*

According to Theorem 2, the generalized tensor rank between two irrelevant modes of $\boldsymbol{\mathcal{X}}$ is equal to 1. Assuming a TN can represent $\boldsymbol{\mathcal{X}}$ with N factors and can be divided into several sub-networks, then the edge rank between the factors belonging to each sub-network is equal to 1, and there is no correlation or entanglement between the sub-networks. The original tensor can be obtained through the outer product of sub-tensors that corresponds to sub-network contraction results. It shows that the generalized tensor ranks reasonably correspond to the edge ranks connecting the factors. Therefore, we exploit the generalized tensor rank and nuclear norm to adaptive decompose $\boldsymbol{\mathcal{X}}$. We use a hyperparameter κ to control the computational complexity of ATN, where $0 < \kappa \leq 1$. The edge rank between any two factors, $\boldsymbol{\mathcal{Z}}^{(i)}$ and $\boldsymbol{\mathcal{Z}}^{(j)}$, is defined as

$$R_{i,j} \overset{\text{def}}{=} \min_{x} \frac{diag(\sum_{k=1}^{C} \boldsymbol{S}_k^{(i,j)})_{1:x}}{||\sum_{k=1}^{C} \boldsymbol{S}_k^{(i,j)}||_*} \geq \kappa. \tag{5}$$

Here the *diag* represents the vectorizing operation of the diagonal elements of the matrix and $\boldsymbol{S}_k^{(i,j)}$ defined in (3). We can regard κ as the degree of information retention between modes. A larger κ is helpful to improve model performance, but increases the number of parameters and computational complexity exponentially.

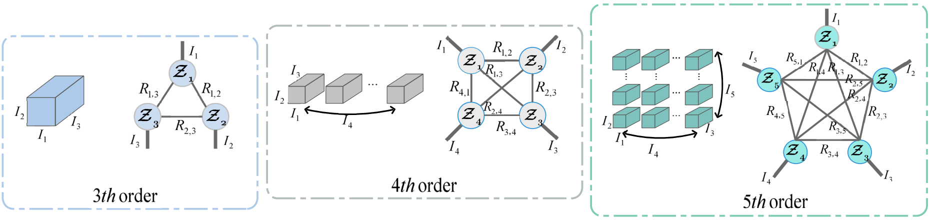


Figure 2: Graphical representation of ATN decomposition to $3rd$-order, $4th$-order, and $5th$-order tensors. The topology structure of ATN is a complete graph before removing all the rank-1 edges.

Note that only the hyperparameter κ is applied to determine the ATN's topology and rank. We further consider the problem of κ selection. Most TD models aim to establish a trade-off between constraint conditions and representational ability. For an Nth-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, supposing the upper limit of the number of parameters or computational complexity of ATN is given, we first need to calculate vector $\boldsymbol{V}_{i,j} = diag(\sum_{k=1}^{C} \boldsymbol{S}_k^{(i,j)})$ for any $1 \leq i < j \leq N$ and initialize the search interval of κ as $[\kappa_{min}, \kappa_{max}]$, where $\kappa_{min} = min\{\frac{V_{i,j}(1)}{sum(V_{i,j})}\}_{1 \leq i < j \leq N}$ and $\kappa_{max} = 1$. Then we utilize binary search strategy to get the optimal κ. We select κ as the midpoint of the current interval in each searching process and obtain the rank of ATN $R_{i,j}$ by (5). At this time, the number of parameter and computation complexity required for ATN is $\sum_{i=1}^{N} I_i \prod_{j \neq i} R_{i,j}$ and $\mathcal{O}(\sum_{j=2}^{N} (\prod_{i=1}^{j} I_i \prod_{k=j+1}^{N} R_{i,k})(\prod_{i=1}^{j-1} R_{i,j}))$. We update $\kappa_{min} = \kappa$ if constraint conditions is satisfied, else $\kappa_{max} = \kappa$. We accept $\kappa = \kappa_{min}$ as the ultimate result when the rank of ATN is unchangeable after a few consecutive searches. In short, the computational complexity of setting κ is mainly on the calculation of the singular value decomposition of all frontal slices of $\boldsymbol{\mathcal{X}}_{i,j}$ for any $1 \leq i < j \leq N$, which can be accelerated extremely by parallel computing. This greatly simplifies ATN's rank determination and topology search problems compared with other adaptive TD models [10].

3.3 Complexity Analysis and Properties

We further reveal the number of parameters' upper bound and computational complexity required for ATN contraction in Theorem 3.

Theorem 3 *Let Nth-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and $1 < I_1 \leq \cdots \leq I_N$. Under a certain κ, the number of parameter' upper bound and computational complexity of ATN decomposition is $\sum_{i=1}^{N} (\prod_{j=1}^{i} I_j) I_i^{N-i} \kappa^{N-1}$ and $\mathcal{O}(\sum_{j=2}^{N} (\kappa^{j(N-j)+j-1} \prod_{i=1}^{j} I_i^{N-j+1})/I_j)$.*

In fact, the information between modes is often concentrated on the first a few larger eigenvalues, which is similar to matrix decomposition. Therefore, the computational complexity of ATN is far less than the theoretical upper bound. Since any finite-dimension tensor can be represented by TN [30] and choosing a sufficiently large edge rank can make the structural loss of decomposition disappear. Supposing the ATN's rank $R_{i,j} = min\{I_i, I_j\}$, it is provable that existing N factors $\boldsymbol{\mathcal{Z}}^{(i)} \in \mathbb{R}^{R_{1,i} \times \cdots \times R_{i-1,i} \times I_i \times R_{i+1,i} \times \cdots \times R_{N,i}}, i \in [N]$ satisfy $||\boldsymbol{\mathcal{X}} - TN(\boldsymbol{\mathcal{Z}}^{(1)}, \ldots, \boldsymbol{\mathcal{Z}}^{(N)})||_F \leq \varepsilon$, for any given relative error $\varepsilon > 0$. This indicates the upper bound of the ATN's rank. In addition, ATN possesses the merit of permutation invariance, which makes it unnecessary to consider the arrangement of factors.

Lemma 1 *(Permutation Invariance). For Nth-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, its ATN decomposition is denoted as $\boldsymbol{\mathcal{X}} = TN(\boldsymbol{\mathcal{Z}}^{(1)}, \ldots, \boldsymbol{\mathcal{Z}}^{(N)})$. We introduce permutation operator $\mathfrak{P}$ and*

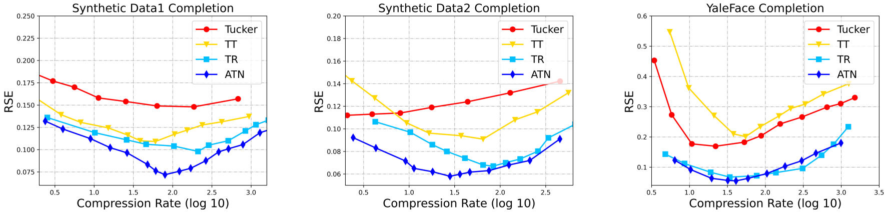


Figure 3: Tensor completion performance of four different TD models on synthetic data and YaleFace dataset. The missing ratio of the target tensor is 90%.

define:

$$\mathfrak{P}(\boldsymbol{\mathcal{X}},\, k) = TN(\boldsymbol{\mathcal{Z}}^{(k)}, \boldsymbol{\mathcal{Z}}^{(1)}, \ldots, \boldsymbol{\mathcal{Z}}^{(k-1)}, \boldsymbol{\mathcal{Z}}^{(k+1)}, \ldots, \boldsymbol{\mathcal{Z}}^{(N)}) \in \mathbb{R}^{I_k \times I_1 \times \cdots \times I_{k-1} \times I_{k+1} \cdots \times I_N}. \quad (6)$$

Here the $k \in [N]$. The permutation invariance of ATN is established by employing $\mathfrak{P}$ operations at most $N-1$ times.

The permutation invariance of ATN decomposition maintains the original tensor's inherent properties, which are not available in TT and TR [24, 40]. It is worth noting that ATN is a more general TD framework and other models like TT, TR, FCTN, and PEPS can be regarded as a special form of ATN. For instance, supposing the condition $R_{1,3} = \cdots = R_{1,N-1} = R_{2,4} = \cdots = R_{2,N} = \cdots = R_{N-2,N} = 1$ is satisfied, then ATN degenerates to TR.

Edge pruning refers to the process of removing rank-1 edges in ATN since they have no substantial contribution to the result. For the standard edges connecting the crucial factors, ATN is prone to select a larger rank to better adapt the data. We can control the behavior of ATN through the hyperparameter κ, and achieve better results even under the unbalanced modes dimensions.

Due to space restraints, the detailed implementation of the ATN-based models is available in the supplementary material.

4 Experiments

4.1 Settings

Environment Settings. We use the automatic differentiation tool PyTorch [26] to minimize the reconstruction error in tensor completion and image denoising task or cross-entropy classification loss in neural network compression task. All of our comparative experiments were conducted under the same configuration (GPU, Nvidia A10) for the sake of fairness.

Parameter Setting. As discussed earlier, the setting of κ is decisive to the topology and rank of ATN. The κ in all experiments is obtained via binary search under the given restraints (parameters compression rate r), as referred to Section 3. We analyze the influence of κ on the complete performance of ATN and compare it with other TD models. We select r as an independent variable for unity in comparison. As depicted in Fig. 3, ATN owns higher stability and performance since its corresponding curve fluctuates in lower values under a wide range of r. It should be remarked that a larger κ corresponds to a smaller r, and the ATN's edge rank and structure become large and complicate. In short, the existence of κ greatly simplifies the difficulty of ATN's rank determination and can be calculated directly by r.

4.2 Comparison results

Following [2, 10, 41], we utilize diverse test data to demonstrate the effectiveness of ATN. Over the test set, we compare ATN decomposition with other known TD models, including TT [24], TR [40], and Tucker decomposition [34]. The ranks of other TD models are adjusted carefully according to experimental results to achieve their best performance. As suggested in the literature [24, 40], the TT's rank is a vector where its items increase first and then decrease, and the rank of TR or Tucker is an integer.

We apply ATN to three typical high-dimensional optimization tasks: tensor completion, image denoising, and neural network compression. These optimization problems are mainly based on the low-rank assumption of the data. In other words, they aim to capture the internal structure of high-dimensional data and to eliminate redundancy through the low-rank representation.

Tensor Completion. TC aims to capture the global structure through partial observation entries. For the task of tensor completion, the synthetic data includes two $6th$-order tensors (Syn1 and Syn2) of size $12 \times 12 \times 12 \times 12 \times 12 \times 12$ and $3 \times 4 \times 8 \times 16 \times 32 \times 64$, both obtained by summing 64 rank-one tensors of the same size which are sampled from uniform distribution. The large unevenness in data mode dimension can verify the robustness of the TD models. The natural data includes YaleFace Dataset[10] (Contains 38×64 grayscale images) of size $48 \times 42 \times 64 \times 38$ and 3 RGB images reshaped into $16 \times 16 \times 16 \times 12 \times 4$. The missing ratio (MR) of data is set to $90\%, 80\%$ and 50%, respectively. The observed elements are obtained by random sampling.

The Relative Square Error (RSE) [38] is applied to evaluate model performance in tensor completion. We report the RSE values (average results of 20 independent experiments) in Table 1 of various TD models with six tensors. The smaller the RSE value, the better performance. We noticed that ATN achieved the best results under different data missing ratios, and Tucker decomposition was the worst. As the MR is larger, RSE values continue to increase and lose more details of the recovered image for all methods. For the Syn2 with large mode discrepancies and the large YaleFace dataset, ATN also achieves better results and shows its strong robustness. Fig. 4 presents the completed results of 3 real images (Img1∼Img3) and their TN topological structures obtained by ATN when the data MR reaches 90%. It can be clearly seen in the second image named "Lena" that ATN decomposition has obvious advantages in grasping local details and global structure.

Table 1: Comparison of RSE values of six completion results. Here Syn1 and Syn2 are synthetic data and Img1∼Img3 are real images.

	Tucker	TT	TR	ATN	Tucker	TT	TR	ATN	Tucker	TT	TR	ATN
Dataset	MR=50%				MR=80%				MR=90%			
Syn1	0.136	0.087	0.082	**0.058**	0.140	0.096	0.095	**0.064**	0.148	0.109	0.098	**0.072**
Syn2	0.093	0.075	0.042	**0.032**	0.107	0.088	0.059	**0.041**	0.112	0.091	0.066	**0.058**
Img1	0.185	0.134	0.141	**0.115**	0.203	0.139	0.143	**0.120**	0.207	0.165	0.159	**0.146**
Img2	0.174	0.131	0.123	**0.076**	0.190	0.145	0.134	**0.087**	0.213	0.150	0.137	**0.132**
Img3	0.228	0.182	0.175	**0.123**	0.240	0.187	0.182	**0.174**	0.251	0.211	0.199	**0.192**
YaleFace	0.133	0.087	0.043	**0.038**	0.143	0.131	0.049	**0.042**	0.170	0.201	0.067	**0.054**

Image Denoising. For the image denoising task, the test data includes 2 RGB color images Img4 and Img5 (both reshaped to the size of $8 \times 8 \times 8 \times 8 \times 3$), 2 videos Vid1 and Vid2, each composed of 32 frames (reshaped to $8 \times 4 \times 8 \times 8 \times 8 \times 8 \times 3$) and YaleFace Dataset. We add random noise sampled from normal distribution $N(0, 0.1)$ to the data. The RSE, Peak

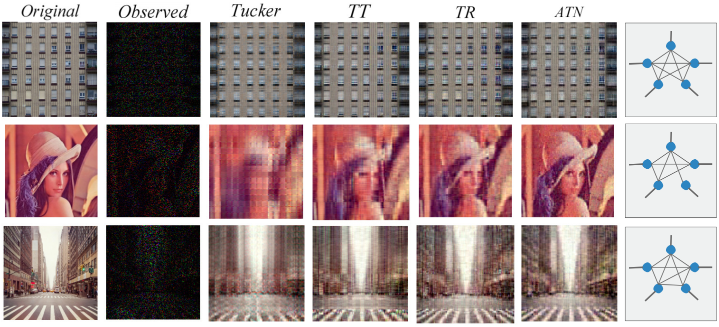


Figure 4: Completion results of 3 real images with a MR of 90%. The first two columns are the original and observed images, respectively. The last column illustrates the TN topological structures of images obtained by ATN.

Table 2: Comparison of RSE and PSNR values of five tensors after denoising, including two real images, two videos, and YaleFace dataset.

	Tucker			TT			TR			ATN		
Dataset	RSE	PSNR	Time (s)	RSE	PSNR	Time (s)	RSE	PSNR	Time (s)	RSE	PSNR	Time (s)
Img4	0.2135	21.29	301.1	0.1726	23.14	188.8	0.1547	24.08	172.2	**0.1339**	**25.33**	**166.5**
Img5	0.1559	22.84	315.4	0.1527	23.03	**227.6**	0.123	24.43	265.8	**0.1212**	**25.03**	282.6
Vid1	0.1541	21.56	968.3	0.1287	23.12	554.3	0.1187	23.82	617.7	**0.0947**	**25.78**	**541.0**
Vid2	0.2763	18.68	1023	0.2809	18.54	**575.2**	0.2762	18.67	611.5	**0.2101**	**21.06**	586.9
YaleFace	0.32333	21.84	1112	0.2167	25.32	644.0	0.1839	26.74	**619.1**	**0.0932**	**32.64**	688.4

Signal-to-Noise Ratio (PSNR) [38], and running time are applied to evaluate model performance and efficiency. Table 2 presents the average results of 20 independent experiments of different methods on the test datasets. Compared with other decomposition models, ATN obtains lower RSE and higher PSNR by a large margin in all cases. Besides, we noticed TR is better than TT, revealing the importance of factor connection. In terms of computational efficiency, we observe that the running time of ATN in five datasets has the same order of magnitude with TT and TR, while Tucker decomposition takes much more time. Although ATN has a denser connected topology and that theoretically should increase the computational complexity, our experimental results indicate that ATN has lower ranks and requires fewer iterations during the optimization phase.

Table 3: Performance comparison for compressing ResNet56 on CIFAR-10.

Model	Weights	FLOPs	Acc (%)
CP-ResNet56[19]	0.14M	23M	89.6
Tucker-ResNet56[16]	0.15M	22M	88.4
Tucker-VBMF-ResNet56[16]	0.16M	32M	88.7
TT-ResNet56[22]	0.16M	24M	87.1
TR-ResNet56[2]	0.21M	27M	89.7
ATN-ResNet56	0.14M	22M	**90.3**

Neural Network Compression. Neural network compression focuses on using TD to equivalently approximate the low-rank structure of the kernels in convolutional layers to obtain fewer number of network parameters and FLOPs. The dataset CIFAR10 [18] is used in the neural network compression task, which contains 50K training images and 10K verification

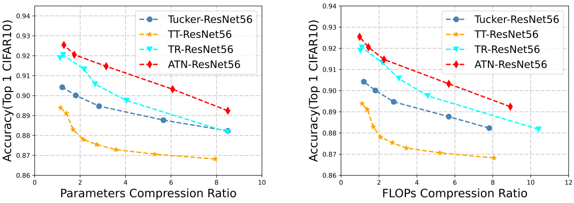
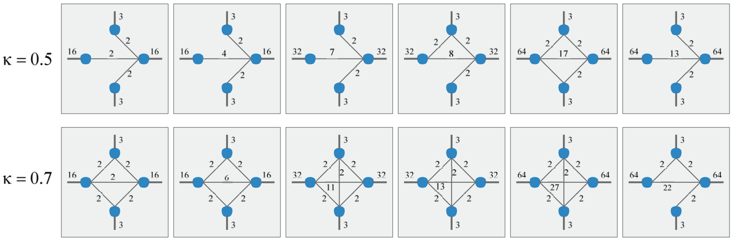


Figure 5: Tradeoff curves of accuracy vs. parameters and FLOPs compression ratio with TD-based ResNet56 models. The ATN-ResNet56 shows higher accuracy with the same space complexity and computational complexity.



(a) (b) (c) (d) (e) (f)

Figure 6: The TN topological structure diagrams of the kernels in (a) 8th, (b) 16th, (c) 24th, (d) 32th, (e) 40th, and (f) 48th convolutional layers of ResNet56 obtained by ATN under different κ. The ranks of ATN are marked around the edge. Note that the contraction results are unaffected to factors arrangement due to the permutation invariance of ATN.

images of the resolution 32×32. We apply various TD models to compress the convolutional layer with the same kernel size 3×3 in the residual network ResNet56 [12] and adjust their ranks to control the compression ratio of parametric and FLOPs. The accuracy on the verification set is used as evaluation criterion. As shown in Fig. 5, ATN-ResNet56 achieves higher accuracy with fewer parameters and FLOPs compared to the others, which attributes to its using pre-trained parameters to select an adaptive rank. From the performance comparison results in Table 3, we see that the accuracy of ATN-based ResNet56 is higher than Tucker-based ResNet56 with VBMF [16] rank initialization. That implies the improvement of model performance benefits from both topological structure and rank determination. In Fig. 6, we graphically show the topologies and ranks of kernels compressed by ATN. From this, we can know the correlation degree between different modes of the kernel, e.g., the correlation within channel modes is stronger than spatial modes.

5 Conclusion

We introduced a novel adaptive tensor network decomposition for high-dimensional optimization problems established on generalized tensor rank. ATN sufficiently utilizes the intrinsic correlation of data modes to determine the topology instead of time-consuming automatic search, yielding stronger robustness and representation ability for data with unevenness dimensions. ATN has the permutation invariance property and an upper bound of parameters. Moreover, we employ ATN on three typical learning tasks, and experiment results on synthetic and natural data show that ATN has significant advantages compared with other TD models. We will try to resolve the its limitations in the future and explore more inspiring and effective tensor network topologies.

Acknowledgement

This work was supported in part by the Postgraduate Research & Practice Innovation Program (KYCX21_0304) of Jiangsu Province, China, and in part by the National Natural Science Foundation of China (NSFC) under Grant 61703209.

References

[1] Hervé Abdi and Lynne J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

[2] Vaneet Aggarwal, Wenlin Wang, Brian Eriksson, Yifan Sun, and Wenqi Wang. Wide compression: Tensor ring nets. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9329–9338, 2018.

[3] Jonas Ballani and Lars Grasedyck. Tree adaptive approximation in the hierarchical tensor format. *SIAM Journal on Scientific Computing*, 36(4), 2014.

[4] Jacob C. Bridgeman and Christopher T. Chubb. Hand-waving and interpretive dance: An introductory course on tensor networks. *Journal of Physics A*, 50(22):223001, 2017.

[5] Song Cheng, Lei Wang, Tao Xiang, and Pan Zhang. Tree tensor networks for generative modeling. *Physical Review B*, 99(15):155131, 2019.

[6] Andrzej Cichocki, Danilo Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar Caiafa, and Huy Anh Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, 32(2):145–163, 2015.

[7] Ivan Glasser, Ryan Sweke, Nicola Pancotti, Jens Eisert, and J. Ignacio Cirac. Expressive power of tensor-network factorizations for probabilistic modeling. In *Thirty-third Conference on Neural Information Processing Systems (NeurIPS | 2019)*, volume 32, pages 1498–1510, 2019.

[8] Lars Grasedyck. Hierarchical singular value decomposition of tensors. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2029–2054, 2010.

[9] Richard A. Harshman. Foundations of the parafac procedure: Models and conditions for an “explanator“ multi-model factor analysis. 16:1–84, 1970.

[10] Meraj Hashemizadeh, Michelle Liu, Jacob Miller, and Guillaume Rabusseau. Adaptive tensor learning with tensor networks. *arXiv preprint arXiv:2008.05437*, 2020.

[11] Kohei Hayashi, Taiki Yamaguchi, Yohei Sugawara, and Shin ichi Maeda. Exploring unexplored tensor network decompositions for convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 32, pages 5552–5562, 2019.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[13] Christopher J. Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *Journal of the ACM*, 60(6):45, 2013.

[14] Yuwang Ji, Qiang Wang, Xuan Li, and Jie Liu. A survey on tensor techniques and applications in machine learning. *IEEE Access*, 7:162950–162990, 2019.

[15] Misha E. Kilmer and Carla D. Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641–658, 2011.

[16] Yong-Deok Kim, Eunhyeok Park, Sungjoo Yoo, Taelim Choi, Lu Yang, and Dongjun Shin. Compression of deep convolutional neural networks for fast and low power mobile applications. In *ICLR 2016 : International Conference on Learning Representations 2016*, 2016.

[17] Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *Siam Review*, 51(3):455–500, 2009.

[18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[19] Vadim Lebedev, Yaroslav Ganin, Maksim Rakhuba, Ivan Oseledets, and Victor Lempitsky. Speeding-up convolutional neural networks using fine-tuned cp-decomposition. In *ICLR 2015 : International Conference on Learning Representations 2015*, 2015.

[20] Chao Li and Zhun Sun. Evolutionary topology search for tensor network decomposition. In *ICML 2020: 37th International Conference on Machine Learning*, volume 1, pages 5947–5957, 2020.

[21] Chao Li, Mohammad Emtiyaz Khan, Zhun Sun, Gang Niu, Bo Han, Shengli Xie, and Qibin Zhao. Beyond unfolding: Exact recovery of latent convex tensor decomposition under reshuffling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4602–4609, 2020.

[22] Alexander Novikov, Dmitry Podoprikhin, Anton Osokin, and Dmitry Vetrov. Tensorizing neural networks. In *NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, volume 28, pages 442–450, 2015.

[23] Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117–158, 2014.

[24] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[25] Evangelos E. Papalexakis, Christos Faloutsos, and Nicholas D. Sidiropoulos. Tensors for data mining and data fusion: Models, applications, and scalable algorithms. *ACM Transactions on Intelligent Systems and Technology*, 8(2):16, 2016.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8026–8037, 2019.

[27] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac. Matrix product state representations. *Quantum Information & Computation*, 7(5):401–430, 2007.

[28] Ignacio Ramirez, Pablo Sprechmann, and Guillermo Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3501–3508, 2010.

[29] Shi-Ju Ran, Zheng-Zhi Sun, Shao-Ming Fei, Gang Su, and Maciej Lewenstein. Quantum compressed sensing with unsupervised tensor network machine learning. *arXiv preprint arXiv:1907.10290*, 2019.

[30] Shi-Ju Ran, Emanuele Tirrito, Cheng Peng, Xi Chen, Luca Tagliacozzo, Gang Su, and Maciej Lewenstein. *Tensor Network Contractions: Methods and Applications to Quantum Many-Body Systems*. 2020.

[31] Nicholas D. Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E. Papalexakis, and Christos Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582, 2017.

[32] Zheng-Zhi Sun, Cheng Peng, Ding Liu, Shi-Ju Ran, and Gang Su. Generative tensor network classification model for supervised machine learning. *Physical Review B*, 101 (7):75135, 2020.

[33] Dacheng Tao, Xuelong Li, Weiming Hu, S. Maybank, and Xindong Wu. Supervised tensor learning. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, volume 13, pages 450–457, 2005.

[34] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[35] Su-Jing Wang, Hui-Ling Chen, Wen-Jing Yan, Yu-Hsin Chen, and Xiaolan Fu. Face recognition and micro-expression recognition based on discriminant tensor subspace analysis plus extreme learning machine. *Neural Processing Letters*, 39(1):25–43, 2014.

[36] Yue Wu, Leyuan Fang, and Shutao Li. Weighted tensor rank-1 decomposition for nonlocal image denoising. *IEEE Transactions on Image Processing*, 28(6):2719–2730, 2019.

[37] Miao Yin, Yang Sui, Siyu Liao, and Bo Yuan. Towards efficient tensor decomposition-based dnn model compression with optimization framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10674–10683, 2021.

[38] Longhao Yuan, Chao Li, Danilo P. Mandic, Jianting Cao, and Qibin Zhao. Tensor ring decomposition with rank minimization on latent space: An efficient approach for tensor completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9151–9158, 2019.

[39] Lefei Zhang, Liangchen Song, Bo Du, and Yipeng Zhang. Nonlocal low-rank tensor completion for visual data. *IEEE Transactions on Systems, Man, and Cybernetics*, 51 (2):673–685, 2021.

[40] Qibin Zhao, Guoxu Zhou, Shengli Xie, Liqing Zhang, and Andrzej Cichocki. Tensor ring decomposition. *arXiv preprint arXiv:1606.05535*, 2016.

[41] Yu-Bang Zheng, Ting-Zhu Huang, Xi-Le Zhao, Qibin Zhao, and Tai-Xiang Jiang. Fully-connected tensor network decomposition and its application to higher-order tensor completion. In *the AAAI Conference*, 2021.