TRINET: A NOVEL AND MEMORY-EFFICIENT TENSOR NETWORK FOR HIGHER-ORDER TENSOR DECOMPOSITION

Nguyen Thi Ngoc Lan[⋆]*, Thanh Trung Le*[⋆]*, Nguyen Linh Trung*[⋆]*, Karim Abed-Meraim*[†]

[⋆] VNU University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam

[†] PRIMSE Laboratory, University of Orleans, France

ABSTRACT

In this paper, we propose a memory-efficient tensor network, called TriNet decomposition, designed for modeling higher-order dependencies in multivariate and high-dimensional data. By introducing "relay" factors, TriNet can connect core (loading) factors while restricting their order to at most three, yielding a compact tensor representation. To compute both tensor decomposition and completion, we develop an effective algorithm based on the alternating direction method of multipliers. Experimental results demonstrate that our method offers a favorable balance between space complexity and computational efficiency, while still achieving competitive estimation accuracy as compared to state-of-the-art tensor methods.

Index Terms— Tensor decomposition, TriNet decomposition, tensor networks, tensor completion, ADMM.

1. INTRODUCTION

Tensor decomposition (TD) has become a powerful processing tool for analyzing multidimensional and multivariate data in both batch and adaptive settings [1–3]. With its capability to factorize tensors (aka multiway arrays) into basic components, TD has successfully demonstrated its applications in various signal processing tasks [4].

With the growing need to model complex interactions across multiple dimensions in modern datasets, many TD techniques have been developed. Among these, CP/PARAFAC and Tucker/HOSVD are two of the most classical tensor models, often used to capture multiway relationships and provide low-rank structures in tensor data [1]. However, many problems related to CP/PARAFAC are NP-hard (e.g., determining the CP rank and computing the best low-rank approximation) [5]. While Tucker/HOSVD suffers from the so-called "curse of dimensionality", where the number of parameters required grows exponentially with the tensor order [6], see Table 1. To address these limitations, tensor networks (TN) have emerged as advanced formats that can efficiently represent higher-order tensors [7].

Among TNs, the tensor-train (TT) decomposition factorizes an Nth-order tensor $\boldsymbol{\mathcal{X}}$ into a sequence of $N-2$ third-order tensors in the middle and two matrices at the ends [8]. These components are linked together through contracted modes using multilinear operations. To enhance the flexibility of TT, tensor ring (TR) model replaces the two "end" matrices with third-order tensors, yielding a circularly interconnected structure of core tensors [9]. Despite their efficiency and compactness, both TT and TR primarily capture interactions between adjacent cores, which may limit their ability to model long-range dependence of tensor data.

To overcome this drawback, the fully connected tensor network (FCTN) model was introduced, which decomposes $\boldsymbol{\mathcal{X}}$ into N core tensors with full pairwise connections among all cores [10]. To achieve a better balance between flexibility and efficiency, the tensor wheel (TW) model was later proposed [11]. In TW, N core tensors are arranged around a central tensor, where each core is strongly connected to its neighbors while maintaining global coherence through the central tensor. Despite these advances, FCTN and TW still have inherent limitations. Their space complexity grows exponentially with the tensor order. Particularly, FCTN requires each core to be of order N, leading to substantial space complexity for higher-order tensors. TW requires each core to be fourth-order, but also depends on a central tensor of order N to coordinate correlations among all cores. Consequently, both models still suffer the "curse of dimensionality", similar to Tucker/HOSVD. Recently, the tensor star (TS) model was proposed, providing a more compact structure than TW with improved representation capability and reduced space complexity [12]. However, TS still incurs high computational and storage costs, since updating N fourth-order tensors is computationally expensive. In addition, direct correlations are limited to adjacent core tensors, requiring a large number of central tensors to bridge long-range dependencies.

Unlike classical CP, Tucker, and TN formats, tensor SVD (t-SVD) provides an interesting TD model for third-order tensors by operating in the Fourier domain [13]. However, its extensions to higher-order tensors typically require either all loading factors of order N [14] or a core tensor of order N [15], where N denotes the order of the original tensor. As a result, t-SVD still suffers from the "curse of dimensionalit" and may not be efficient for representing large-scale, higher-order tensors.

In this paper, we propose a compact tensor network model, called TriNet decomposition. The advantage of TriNet stems from its triangular connectivity among loading factors, while ensuring that each factor is of at most third order. This representation allows for an effective exploitation of correlations between adjacent core tensors compared to existing methods. Specifically, when $\mathrm{mod}(N,3) = 2$ (modulo operation), the N-th order tensor $\boldsymbol{\mathcal{X}}$ is decomposed into $N-1$ third-order core factors and one matrix, along with $2\lfloor N/3 \rfloor + 1$ relay factors of third-order. Otherwise, it is decomposed into N third-

Corresponding author: Thanh Trung Le (thanhletrung@vnu.edu.vn)

Table 1: Space complexity of TD models for decomposing an N-th order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times I \times \cdots \times I}$ ($N \geq 3$). Assume that all rank parameters in these TD models are equal to r.

Model	Space Complexity	Model	Space Complexity
CP	$\mathcal{O}(NIr)$	FCTN	$\mathcal{O}(NIr^{N-1})$
Tucker	$\mathcal{O}(NIr + r^N)$	TW	$\mathcal{O}(NIr^3 + r^N)$
TT	$\mathcal{O}((N-2)Ir^2 + 2Ir)$	TS	$\mathcal{O}(NIr^2 + Nr^4)$
TR	$\mathcal{O}(NIr^2)$	TriNet	$\mathcal{O}(NIr^2 + 2\lfloor \frac{N}{3} \rfloor r^3)$
tSVD	$\mathcal{O}(2I^{N-1}r + I^{N-2}r^2)$ or $\mathcal{O}(NI^2 + I^N)$		

order core factors and $2\lfloor N/3 \rfloor$ relay factors. Thanks to its lightweight structure, the proposed model is well suited for higher-order tensor data. We then develop an efficient and provable alternating direction method of multipliers (ADMM) for computing the TriNet decomposition and tensor completion. Experimental results on both synthetic and real-world datasets further demonstrate its effectiveness and superiority over state-of-the-art tensor decomposition models. See Table 1 for a comparison of the space complexity across state-of-the-art tensor models.

2. TENSOR NOTATIONS AND OPERATIONS

In this paper, we denote scalars, vectors, matrices, and tensors by x, $\mathbf{x}$, $\mathbf{X}$, and $\boldsymbol{\mathcal{X}}$, respectively. For an N-th order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, its $(i_1, i_2, ..., i_N)$-th element is denoted as $\boldsymbol{\mathcal{X}}(i_1, i_2, ..., i_N)$. The Frobenius norm of $\boldsymbol{\mathcal{X}}$ is defined as $\|\boldsymbol{\mathcal{X}}\|_F = \sqrt{\sum_{i_1, i_2, ..., i_N} \boldsymbol{\mathcal{X}}(i_1, i_2, ..., i_N)^2}$. We denote by $\boldsymbol{\mathcal{X}}_{n:m}$ the set $(\boldsymbol{\mathcal{X}}_n, \boldsymbol{\mathcal{X}}_{n+1}, \ldots, \boldsymbol{\mathcal{X}}_m)$ with $n \leq m$. The nuclear norm of a matrix is denoted by $\|\cdot\|_*$. The modulo $\mathrm{mod}(a, b)$ returns the remainder of a when divided by b. Given an ordering vector $\mathbf{n} = [n_1, n_2, \ldots, n_N]$, which is a permutation of the vector $[1, 2, \ldots, N]$, the tensor permutation is denoted as $\vec{\boldsymbol{\mathcal{X}}}^{\mathbf{n}} = \mathrm{permute}(\boldsymbol{\mathcal{X}}, \mathbf{n})$ with its inverse $\boldsymbol{\mathcal{X}} = \mathrm{ipermute}(\vec{\boldsymbol{\mathcal{X}}}^{\mathbf{n}}, \mathbf{n})$.

The mode-d unfolding matrix of $\boldsymbol{\mathcal{X}}$ is represented by $\mathbf{X}_{[d]}$ or $(\mathbf{X})_{[d]}$, whose entries are given by

$$\mathbf{X}_{[d]}(i_d, \overline{i_{d+1} \ldots i_N i_1 \ldots i_{d-1}}) = \boldsymbol{\mathcal{X}}(i_1, i_2, \ldots, i_N),$$

where $\overline{j_1 j_2 \ldots j_n}$ is a multi-index used in matricization for tensors, defined as in [7]. For easy of presentation, we denote $\mathbf{X}_{[d]} = \mathrm{Unfold}_{[d]}(\boldsymbol{\mathcal{X}})$ and its inverse $\boldsymbol{\mathcal{X}} = \mathrm{Fold}_{[d]}(\mathbf{X}_{[d]})$.

The generalized unfolding operator for a given mode d and the ordering vector $\mathbf{n}$ is defined as

$$\mathbf{X}_{[\mathbf{n};d]}(\overline{i_{n_1} \ldots i_{n_d}}, \overline{i_{n_{d+1}} \ldots i_{n_N}}) = \vec{\boldsymbol{\mathcal{X}}}^{\mathbf{n}}(i_{n_1}, i_{n_2}, \ldots, i_{n_N}).$$

We also denote $\mathbf{X}_{[\mathbf{n};d]} = \mathrm{Unfold}_{[n_{1:d}; n_{d+1:N}]}(\boldsymbol{\mathcal{X}})$ and its inverse $\boldsymbol{\mathcal{X}} = \mathrm{Fold}_{[n_{1:d}; n_{d+1:N}]}(\mathbf{X}_{[\mathbf{n};d]})$. In the case $d = 0$, the unfolding operator reduces to the vectorization operator.

The generalized tensor contraction is as follows. Given $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_M}$ with $I_{n_i} = J_{m_i}$ for $i = 1, 2, \ldots, d$, the contraction of $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ along $n_{1:d}$- and $m_{1:d}$-modes results in

$$\boldsymbol{\mathcal{X}} \times_{n_{1:d}}^{m_{1:d}} \boldsymbol{\mathcal{Y}} = \mathrm{Fold}_{[(1,2,\ldots,N+M-2d);\, N-d]} \left(\mathbf{X}_{[\mathbf{n};d]}^{\top} \mathbf{Y}_{[\mathbf{m};d]} \right).$$

3. TRINET TENSOR DECOMPOSITION

3.1. TriNet Model and Properties

As illustrated in Fig. 1, our TriNet decomposition factorizes an N-th order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ into third-order components, including a set of N *core* factors $\{\boldsymbol{\mathcal{G}}_n\}_{n=1}^N$ (dark cyan nodes) and a set of K *relay* factors $\{\boldsymbol{\mathcal{C}}_k\}_{k=1}^K$ (yellow nodes). For short, we denote the TriNet decomposition of $\boldsymbol{\mathcal{X}}$ as $\boldsymbol{\mathcal{X}} = \mathrm{TriNet}(\{\boldsymbol{\mathcal{G}}_n\}_{n=1}^N, \{\boldsymbol{\mathcal{C}}_k\}_{k=1}^K)$. Here, the n-th *core* factor $\boldsymbol{\mathcal{G}}_n$ is of size $r_{n,1} \times I_n \times r_{n,2}$ whose second mode corresponds to the n-th dimension of the original tensor $\boldsymbol{\mathcal{X}}$. Particularly when $\mathrm{mod}(N, 3) \neq 0$, $r_{N,2} = r_{1,1}$. We call the set of $\{r_{n,1}, r_{n,2}\}_{n=1}^N$ as the TriNet-rank. Thus, the factors $\{\boldsymbol{\mathcal{G}}_n\}_{n=1}^N$ encode mode-specific information from $\boldsymbol{\mathcal{X}}$. While the *relay* factors $\{\boldsymbol{\mathcal{C}}_k\}_{k=1}^K$ serve as connections among the *core* factors. The number of *relay* factors is determined as $K = 2\lfloor \frac{N}{3} \rfloor$ if $\mathrm{mod}(N, 3) = 0$ or $\mathrm{mod}(N, 3) = 1$, and $K = 2\lfloor \frac{N}{3} \rfloor + 1$ if $\mathrm{mod}(N, 3) = 2$. Accordingly, the size of $\boldsymbol{\mathcal{C}}_k$ is specified as follows

$$\boldsymbol{\mathcal{C}}_k \in \begin{cases} \mathbb{R}^{r_{\lfloor \frac{k}{3} \rfloor + k, 2} \times r_{\lfloor \frac{k}{3} \rfloor + k + 1, 2} \times r_{\lfloor \frac{k}{3} \rfloor + k + 2, 1}}, & \text{if } \mathrm{mod}(k, 2) = 0, \\ \mathbb{R}^{r_{\lfloor \frac{k}{3} \rfloor + k, 2} \times r_{\lfloor \frac{k}{3} \rfloor + k + 1, 1} \times r_{\lfloor \frac{k}{3} \rfloor + k + 2, 1}}, & \text{if } \mathrm{mod}(k, 2) \neq 0, \end{cases} \quad (1)$$

for $k = 1, 2, \ldots, K-1$, and the last *relay* factor $\boldsymbol{\mathcal{C}}_K$ is given by

$$\boldsymbol{\mathcal{C}}_K \in \begin{cases} \mathbb{R}^{r_{\lfloor \frac{K}{3} \rfloor + K, 2} \times r_{\lfloor \frac{K}{3} \rfloor + K + 2, 2} \times r_{1,1}}, & \text{if } \mathrm{mod}(N, 3) = 0, \\ \mathbb{R}^{r_{\lfloor \frac{K}{3} \rfloor + K, 2} \times r_{\lfloor \frac{K}{3} \rfloor + K + 1, 1}}, & \text{if } \mathrm{mod}(N, 3) \neq 0. \end{cases} \quad (2)$$

Formally, TriNet decomposition of $\boldsymbol{\mathcal{X}}$ is given in Definition 1.

Definition 1 (TriNet Decomposition)**.** *Let $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ be an N-th order non-zero tensor. The TriNet model aims to decompose $\boldsymbol{\mathcal{X}}$ into a set of third-order tensors $\{\boldsymbol{\mathcal{G}}_n\}_{n=1}^N$, and $\{\boldsymbol{\mathcal{C}}_k\}_{k=1}^K$. Mathematically, its TriNet decomposition $\boldsymbol{\mathcal{X}} = \mathrm{TriNet}(\{\boldsymbol{\mathcal{G}}_n\}_{n=1}^N, \{\boldsymbol{\mathcal{C}}_k\}_{k=1}^K)$ can be expressed as follows*

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{A}} \times_{3\lfloor \frac{N}{3} \rfloor + 2, 1}^{1,5} \boldsymbol{\mathcal{M}}_{\lfloor \frac{N}{3} \rfloor}, \quad \textit{if } \mathrm{mod}(N, 3) = 0, \quad (3a)$$

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{A}} \times_{3\lfloor \frac{N}{3} \rfloor + 2} \boldsymbol{\mathcal{M}}_{\lfloor \frac{N}{3} \rfloor} \times_{N+1, 1}^{1,3} \boldsymbol{\mathcal{G}}_N, \textit{if } \mathrm{mod}(N, 3) = 1, \quad (3b)$$

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{A}} \times_{3\lfloor \frac{N}{3} \rfloor + 2} \boldsymbol{\mathcal{M}}_{\lfloor \frac{N}{3} \rfloor} \times_{N+1} \boldsymbol{\mathcal{G}}_{N-1} \times_{N+1, 1}^{1,3} \boldsymbol{\mathcal{C}}_{3\lfloor \frac{N-1}{2} \rfloor + 1} \times_N \boldsymbol{\mathcal{G}}_N, \quad \textit{if } \mathrm{mod}(N, 3) = 2, \quad (3c)$$

where $\boldsymbol{\mathcal{M}}_\ell = \boldsymbol{\mathcal{G}}_{3\ell+1} \times_3 \boldsymbol{\mathcal{C}}_{2\ell+1} \times_3 \boldsymbol{\mathcal{G}}_{3\ell+2} \times_3 \boldsymbol{\mathcal{G}}_{3\ell+3} \times_{4,6}^{1,2} \boldsymbol{\mathcal{C}}_{2\ell+2}$ and $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{M}}_0 \prod_{\ell=1} \times_{3\ell+2} \boldsymbol{\mathcal{M}}_\ell$ with $0 \leq \ell \leq \lfloor \frac{N-2}{3} \rfloor$.

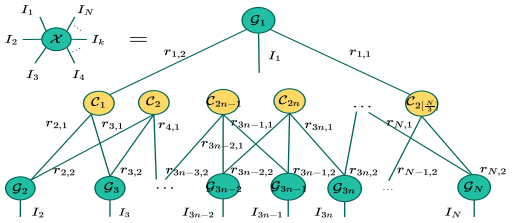


Fig. 1: TriNet decomposition of $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with the rank $\{r_{n,1}, r_{n,2}\}_{n=1}^N$, where $\mathrm{mod}(N, 3) = 0$.

In the following, we present useful properties that provide insights into TriNet's ability to represent higher-order tensors and support its optimization, as discussed in Section 3.2. Due to the space limit, we present key results and omit their proofs.

First, let $\mathbf{n} = [n_1, n_2, \ldots, n_N]$ be a circular permutation of the vector $[1, 2, \ldots, N]$, representing the reordered indices of the core factors $\{\boldsymbol{\mathcal{G}}_n\}_{n=1}^N$. Let $\mathbf{k} = [k_1, k_2, \ldots, k_K]$ denote the corresponding reordering of the relay factors $\{\boldsymbol{\mathcal{C}}_k\}_{k=1}^K$, aligned with the new ordering of $\{\boldsymbol{\mathcal{G}}_{n_i}\}_{i=1}^N$ and we denote by $\{\boldsymbol{\mathcal{C}}_{k_j}\}_{j=1}^K$.

Proposition 1 (Topology Invariance)**.** *Assume that $\boldsymbol{\mathcal{X}}$ is an N-th order tensor that admits the TriNet decomposition $\boldsymbol{\mathcal{X}} = \mathrm{TriNet}(\{\boldsymbol{\mathcal{G}}_n\}_{n=1}^N, \{\boldsymbol{\mathcal{C}}_k\}_{k=1}^K)$. If $\vec{\boldsymbol{\mathcal{X}}}^{\mathbf{n}} = \mathrm{permute}(\boldsymbol{\mathcal{X}}, \mathbf{n})$ for a given vector $\mathbf{n}$, then $\vec{\boldsymbol{\mathcal{X}}}^{\mathbf{n}} = \mathrm{TriNet}(\{\boldsymbol{\mathcal{G}}_{n_i}\}_{i=1}^N, \{\boldsymbol{\mathcal{C}}_{k_j}\}_{j=1}^K)$.*

Algorithm 1: TriNet-ADMM

Input: Observed tensor $\boldsymbol{\mathcal{F}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, missing set Ω, TriNet-rank r, $t_{\max} = 1000$, $\epsilon = 10^{-5}$, $1 < \alpha < 1.5$ and $\lambda = 10^4$.
Initialization: At $t = 0$, $\boldsymbol{\mathcal{X}}^{(0)} = \boldsymbol{\mathcal{F}}$, and TriNet factors $\{\boldsymbol{\mathcal{G}}_k\}_{k=1}^N$, and $\{\boldsymbol{\mathcal{C}}_m\}_{m=1}^K$ are generated at random, $\mathcal{V} = \mathbf{0}$, $\mathcal{M} = \mathbf{0}$, $\mathcal{U} = \mathbf{0}$, $\mathcal{N} = \mathbf{0}$, $\mu^{(0)} = 1$, $\mu_{max} = 100$, $\gamma^{(0)} = 1$, and $\gamma_{max} = 100$.
while *not converged* **and** $t < t_{\max}$ **do**
  for $n = 1, 2, \ldots, N$ **do**
    $\boldsymbol{\mathcal{G}}_n^{(t+1)} = \text{Fold}_{[2]}\Big(\big(\mu(\mathbf{M}_n)_{[2]}^{(t)} + (\mathbf{V}_n)_{[2]}^{(t)} + \lambda \mathbf{X}_{[n]}^{(t)} (\mathbf{Y}_{\neq n}^{(t)})_{[\mathbf{y_v};2]}^\top\big)\big(\mu \mathbf{I} + \lambda (\mathbf{Y}_{\neq n}^{(t)})_{[\mathbf{y_v};2]} (\mathbf{Y}_{\neq n}^{(t)})_{[\mathbf{y_v};2]}^\top\big)^{-1}\Big)$
    $\boldsymbol{\mathcal{M}}_n^{(t+1)} = \text{Fold}_{[2]}\Big(\text{SVT}_{1/\mu}\big((\mathbf{G}_n)_{[2]}^{(t)} - \frac{1}{\mu}(\mathbf{V}_n)_{[2]}^{(t)}\big)\Big)$
    $\boldsymbol{\mathcal{V}}_n^{(t+1)} = \boldsymbol{\mathcal{V}}_n^{(t)} + \mu\big(\boldsymbol{\mathcal{M}}_n^{(t)} - \boldsymbol{\mathcal{G}}_n^{(t)}\big)$
  end
  $\boldsymbol{\mathcal{X}}^{(t+1)} = P_\Omega(\boldsymbol{\mathcal{F}}) + P_{\Omega^c}\big(\text{TriNet}(\mathcal{G}^{(t)}, \mathcal{C}^{(t)})\big)$
  if $t \bmod 20 = 0$ **and** $t \le 100$ **then**
    for $k = 1, 2, \ldots, K$ **do**
      $\boldsymbol{\mathcal{U}}_k^{(t+1)} = \boldsymbol{\mathcal{U}}_k^{(t)} + \gamma\big(\boldsymbol{\mathcal{N}}_k^{(t)} - \boldsymbol{\mathcal{C}}_k^{(t)}\big)$
      $\boldsymbol{\mathcal{C}}_k^{(t+1)} = \text{reshape}\Big(\big(\gamma \mathbf{n}_k^{(t),\top} + \mathbf{u}_k^{(t),\top} + \lambda \mathbf{x}_{[n_1:n_N;0]}^{(t),\top} (\mathbf{Z}_{\neq k}^{(t)})_{[\mathbf{z_v};d]}^\top\big)\big(\gamma \mathbf{I} + \lambda (\mathbf{Z}_{\neq k}^{(t)})_{[\mathbf{z_v};d]} (\mathbf{Z}_{\neq k}^{(t)})_{[\mathbf{z_v};d]}^\top\big)^{-1}, [1:d]\Big)$
      $\boldsymbol{\mathcal{N}}_k^{(t+1)} = \text{Fold}_{[2]}\Big(\text{SVT}_{1/\gamma}\big((\mathbf{C}_k)_{[2]}^{(t)} - \frac{1}{\gamma}(\mathbf{U}_k)_{[2]}^{(t)}\big)\Big)$
    end
  end
  $\mu \leftarrow \min(\alpha\mu, \mu_{\max}), \gamma \leftarrow \min(\alpha\gamma, \gamma_{\max})$
  Check the stopping condition: $\|\boldsymbol{\mathcal{X}}^{(t+1)} - \boldsymbol{\mathcal{X}}^{(t)}\|_F / \|\boldsymbol{\mathcal{X}}^{(t)}\|_F < \epsilon$.
  $t \leftarrow t + 1$
end
Output: Recovered tensor $\boldsymbol{\mathcal{X}}$, and factors $\{\boldsymbol{\mathcal{G}}_n\}_{n=1}^N$ and $\{\boldsymbol{\mathcal{C}}_k\}_{k=1}^K$.

This result suggests that the reordering of the modes or dimensions of $\boldsymbol{\mathcal{X}}$ does not alter its TriNet core or relay factors. The following proposition indicates the relation between the n-th core factor $\boldsymbol{\mathcal{G}}_n$ and mode-n unfolding matrix of $\boldsymbol{\mathcal{X}}$.

Proposition 2 (Mode-n Unfolding)**.** *Assume that the tensor $\boldsymbol{\mathcal{Y}}_{\neq n}$ is determined by contracting all relay factors $\{\boldsymbol{\mathcal{C}}_k\}_{k=1}^K$ and core factors $\{\{\boldsymbol{\mathcal{G}}_i\}_{i=1}^N \backslash \boldsymbol{\mathcal{G}}_n\}$. The mode-$n$ unfolding matrix of $\boldsymbol{\mathcal{X}}$ can be expressed as follows*

$$\mathbf{X}_{[n]} = (\mathbf{G}_n)_{[2]} (\mathbf{Y}_{\neq n})_{[\mathbf{y_v};2]},$$

where the ordering vector $\mathbf{y_v}$ is defined as

$$\mathbf{y_v} = \begin{cases} [1, N+1, 2, ..., N] & \text{if } n = N \text{ and } \bmod(N,3) \neq 0 \\ & \text{or } \bmod(n,3) = 1, \\ [1, N, 2, ..., N-1, N+1] & \text{if } \bmod(n,3) = 0, \\ [2, N+1, 1, 3, 4, ..., N] & \text{if } \bmod(n,3) = 2. \end{cases}$$

Proposition 2 reveals an upper bound on $\text{rank}(\mathbf{X}_{[n]}) \le \min(I_n, r_{n,1} r_{n,2})$ where $r_{n,1}$ and $r_{n,2}$ are the first and third dimensions of the n-th core factor $\boldsymbol{\mathcal{G}}_n$. Proposition 3 exploits the direct connection between the relay factor $\boldsymbol{\mathcal{C}}_k$ and the original tensor. Together with Propositions 2, they play the central role in developing our method for computing the TriNet decomposition and completion.

Proposition 3. *Assume that the tensor $\boldsymbol{\mathcal{Z}}_{\neq k}$ is determined by contracting all core factors $\{\boldsymbol{\mathcal{G}}_{n_i}\}_{i=1}^N$ and relay factors $\{\{\boldsymbol{\mathcal{C}}_{k_j}\}_{j=1}^{k_K} \backslash \boldsymbol{\mathcal{C}}_k\}$, for any circular permutation vector $\mathbf{n}$ of $[1, 2, \ldots, n]$ if $k \neq K$ and $\bmod(N,3) = 0$, and $\mathbf{n} = [2, 3, ..., N-1, 1, N]$ otherwise. The relation between $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Z}}_{\neq k}$ is*

$$\mathbf{x}_{[n_1:n_N;0]}^\top = (\mathbf{c}_k)_{[1:d;0]}^\top (\mathbf{Z}_{\neq k})_{[\mathbf{z_v};d]}.$$

Here, the value of d is set to 2 if $k = K$ and $\bmod(N,3) = 2$, and 3 otherwise. The ordering vector $\mathbf{z_v}$ is defined as

$$\mathbf{z_v} = \begin{cases} [N-2, N, N+2, 1, ..., N-3, N-1, N+1, N+3] & \\ \qquad \text{if } k = K \text{ and } \bmod(N,3) = 1, & \\ [N-1, N+1, 1, ..., N-2, N, N+2] & \\ \qquad \text{if } k = K \text{ and } \bmod(N,3) = 2, & \\ [N-1, N+1, N+2, 1, ..., N-2, N, N+3] & \\ \qquad \text{if } \bmod(k,2) = 0, & \\ [N-1, N, N+2, 1, ..., N-2, N+1, N+3] \text{ otherwise.} & \end{cases}$$

3.2. TriNet-ADMM Method

In this subsection, we introduce an efficient alternating direction method of multipliers (ADMM)-based method for TriNet decomposition and completion, namely TriNet-ADMM.

Let $\boldsymbol{\mathcal{F}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ an incomplete observed data tensor and let $P_\Omega(\cdot)$ represent the projection operator onto the set of observed entries Ω. We define $\mathcal{G} = \{\boldsymbol{\mathcal{G}}_n\}_{n=1}^N$ and $\mathcal{C} = \{\boldsymbol{\mathcal{C}}_k\}_{k=1}^K$ as the sets of core and relay factors, respectively. The main objective function is formulated as follows

$$\begin{aligned} \underset{\boldsymbol{\mathcal{X}}, \mathcal{G}, \mathcal{C}, \mathcal{M}, \mathcal{N}}{\text{argmin}} \quad & \sum_{n=1}^N \sum_{k=1}^K \sum_{i=1}^3 \big\|(\mathbf{M}_n)_{[i]}\big\|_* + \big\|(\mathbf{N}_k)_{[i]}\big\|_* \\ & + \frac{\lambda}{2}\big\|\boldsymbol{\mathcal{X}} - \text{TriNet}(\{\boldsymbol{\mathcal{G}}_n\}_{n=1}^N, \{\boldsymbol{\mathcal{C}}_k\}_{k=1}^K)\big\|_F^2, \quad (4) \\ \text{s.t.} \quad & \boldsymbol{\mathcal{M}}_n = \boldsymbol{\mathcal{G}}_n, \boldsymbol{\mathcal{N}}_k = \boldsymbol{\mathcal{C}}_k, \text{ and } P_\Omega(\boldsymbol{\mathcal{X}}) = P_\Omega(\boldsymbol{\mathcal{F}}), \\ & \text{with } n = 1, 2, \ldots, N \text{ and } k = 1, 2, \ldots, K. \end{aligned}$$

where $\mathcal{M} = \{\boldsymbol{\mathcal{M}}_n\}_{n=1}^N$ and $\mathcal{N} = \{\boldsymbol{\mathcal{N}}_k\}_{k=1}^K$ represent ADMM auxiliary variables corresponding to $\mathcal{G}$ and $\mathcal{C}$, respectively. The corresponding augmented Lagrangian function is given by

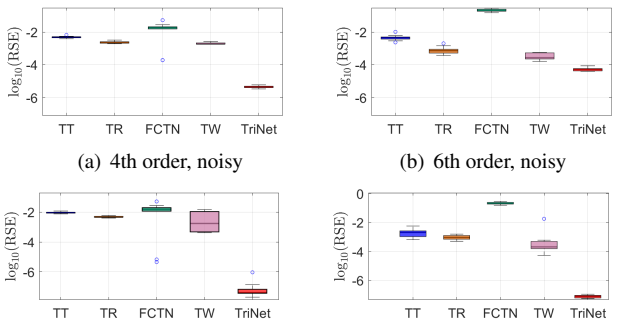
$$\begin{aligned} & \mathcal{L}(\boldsymbol{\mathcal{X}}, \mathcal{G}, \mathcal{C}, \mathcal{M}, \mathcal{N}, \mathcal{V}, \mathcal{U}) \\ & = \sum_{n=1}^N \sum_{k=1}^K \sum_{i=1}^3 \big\|(\mathbf{M}_n)_{[i]}\big\|_* + \big\|(\mathbf{N}_k)_{[i]}\big\|_* + \big\langle \boldsymbol{\mathcal{V}}_n, \boldsymbol{\mathcal{M}}_n - \boldsymbol{\mathcal{G}}_n \big\rangle \\ & \quad + \frac{\mu}{2}\big\|\boldsymbol{\mathcal{M}}_n - \boldsymbol{\mathcal{G}}_n\big\|_F^2 + \big\langle \boldsymbol{\mathcal{U}}_k, \boldsymbol{\mathcal{N}}_k - \boldsymbol{\mathcal{C}}_k \big\rangle + \frac{\gamma}{2}\|\boldsymbol{\mathcal{N}}_k - \boldsymbol{\mathcal{C}}_k\|_F^2 \\ & \quad + \frac{\lambda}{2}\big\|\boldsymbol{\mathcal{X}} - \text{TriNet}(\{\boldsymbol{\mathcal{G}}_n\}_{n=1}^N, \{\boldsymbol{\mathcal{C}}_k\}_{k=1}^K)\big\|_F^2, \quad (5) \end{aligned}$$

s.t. $P_\Omega(\boldsymbol{\mathcal{X}}) = P_\Omega(\boldsymbol{\mathcal{F}})$ where $\mathcal{V} = \{\boldsymbol{\mathcal{V}}_n\}_{n=1}^N$ and $\mathcal{U} = \{\boldsymbol{\mathcal{U}}_k\}_{k=1}^K$ are Lagrangian multipliers; $\mu, \gamma > 0$ are penalty parameters.

Our proposed ADMM-based algorithm for computing a stationary point of (5) is summarized in Algorithm 1. For notations, we denote $\text{vec}(\boldsymbol{\mathcal{N}}_n) = \mathbf{n}_n$, $\text{vec}(\boldsymbol{\mathcal{U}}_k) = \mathbf{u}_k$, and $\mathbf{I}$ the identify matrix. The operator $\text{SVT}_\beta(\cdot)$ refers to singular value thresholding, i.e., if $\mathbf{USV}^\top$ is the singular value decomposition (SVD) of $\mathbf{A}$, then $\text{SVT}_\beta(\mathbf{A}) = \mathbf{U} \max\{\mathbf{S} - \beta\mathbf{I}, 0\}\mathbf{V}^\top$. Next, Ω^c denotes the complement of the observed index set Ω.

In terms of computational complexity, the overall cost of Algorithm 1 is $\mathcal{O}\big(N \sum_{a=1}^{N-1} \sum_{b=3}^5 c_{a,b} I^a r^b + NI^N r^2 + Nr^6\big)$ flops, when we take $I_n = I$, $r_{n,1} = r_{n,2} = r$ for all n, and assume $I > r$. Here, $c_{a,b}$ is a constant bounded by $5\lfloor N/3 \rfloor$. Regarding theoretical performance, Lemma 1 indicates the convergence of our algorithm.

Lemma 1. *Denote $\mathcal{S}^{(t)} = \{\boldsymbol{\mathcal{X}}^{(t)}, \mathcal{G}^{(t)}, \mathcal{C}^{(t)}, \mathcal{M}^{(t)}, \mathcal{N}^{(t)}, \mathcal{V}^{(t)}, \mathcal{U}^{(t)}\}_{t \in \mathbb{N}}$ the solution generated by Algorithm 1 at each iteration t. When $t \to \infty$, the sequence $\{\mathcal{S}^{(t)}\}_{t=1}^\infty$ converges almost surely to a stationary point of* (5).



(a) 4th order, noisy (b) 6th order, noisy

(c) 4th order, noiseless (d) 6th order, noiseless

Fig. 2: Averaged performance of TT, TR, FCTN, TW, and TriNet on 20 synthetic tensors with missing ratio of 90%.

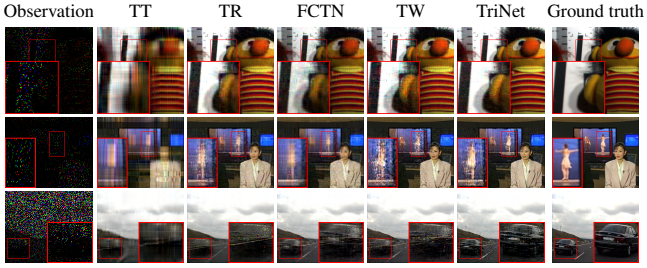


Fig. 3: Reconstructed results on *Toy* (95% missing) and *News* (95% missing) and *Highway* (80% missing).

Lemma 1 holds as the main objective function $f(\mathcal{S})$ in (4) is proper and continuous; the generated solution $\mathcal{S}^{(t)}$ is bounded and $\mathcal{L}(\mathcal{S}^{(t)})$ is lower bounded at each iteration t; and $\mathcal{L}(\mathcal{S}^{(t)})$ satisfies the sufficient descent property and the subgradient bound conditions [16]. Due to space limitations, the mathematical derivations of Algorithm 1 and the detailed proof of Lemma 1 are omitted here and will be presented in our forthcoming journal version.

4. NUMERICAL EXPERIMENTS

In this section, we demonstrate the performance of TriNet on both synthetic and real-world datasets, in comparison with TT [8], TR [17], FCTN [10], and TW [11]. All experiments were implemented in MATLAB R2019a on a machine with 32 GB RAM and an Intel Core i7-9750H @2.60 GHz.

4.1. Synthetic Data

Following the experimental setup in [11], we constructed a dataset consisting of 20 synthetic tensors, including 10 tensors of order 4 and 10 tensors of order 6. Each tensor was generated using Tucker decomposition with factors derived from $\mathcal{U}(0,1)$ and subsequently normalized to the range $[0,1]$. For 4th-order tensors, the dimensions were randomly selected from $\{20,24,28\}$ and for 6th-order tensors, from $\{7,8\}$. The Tucker ranks were fixed at $[5,5,5,5]$, and $[2,2,2,2,2,2]$, respectively. In this task, 90% of the data entries were supposed to be missing at random and we applied the tensor network methods for data imputation. Two levels of Gaussian noise interruption were considered, including noiseless and $\sigma_n = 10^{-4}$ (i.e., $\boldsymbol{\mathcal{X}}_{\text{noisy}} = \boldsymbol{\mathcal{X}}_{\text{true}} + \sigma_n \boldsymbol{\mathcal{N}}$ where entries of $\boldsymbol{\mathcal{N}}$ were obtained from $\mathcal{N}(0,1)$). The parameters of algorithms (e.g., rank and regularization parameters) were fine-tuned to achieve their best performance. The reconstruction performance of the algorithms was measured using the Residual Standard Error (RSE), $\|\boldsymbol{\mathcal{X}}_{\text{est}} - \boldsymbol{\mathcal{X}}_{\text{true}}\|_F / \|\boldsymbol{\mathcal{X}}_{\text{true}}\|_F$ where $\boldsymbol{\mathcal{X}}_{\text{est}}$ is the reconstructed tensor. As shown in Fig. 2, TriNet outperforms other state-of-the-art tensor networks in all cases.

Table 2: Mean PSNR values and run-times of state-of-the-art tensor completion methods on real-world datasets.

Dataset		TT	TR	FCTN	TW	TriNet
Toy	1%	18.548(13.55s)	20.304(122.97s)	16.923(21.33s)	20.645(255.67s)	**22.061**(78.20s)
	5%	23.116(41.61s)	30.430(256.19s)	30.020(77.45s)	30.223(133.44s)	**32.322**(85.60s)
	10%	24.468(97.72s)	35.500(363.43s)	34.322(127.32s)	37.113(322.66s)	**38.133**(152.21s)
	20%	26.492(96.22s)	40.152(371.69s)	40.504(164.68s)	43.265(404.35s)	**44.015**(194.31s)
	Time (s)	62.275	278.57	97.70	279.03	127.58
News	1%	16.678(32.30s)	17.028(405.36s)	13.641(102.54s)	18.152(340.09s)	**19.744**(176.23s)
	5%	20.389(88.33s)	25.848(427.87s)	25.411(225.64s)	27.089(372.40s)	**28.038**(188.85s)
	10%	21.537(127.59s)	30.843(564.64s)	30.461(232.05s)	31.421(312.08s)	**32.286**(295.99s)
	20%	22.580(164.14s)	33.184(560.79s)	35.811(362.58s)	35.842(348.13s)	**36.297**(203.89s)
	Time (s)	103.09	489.67	238.20	343.18	216.24
Highway	1%	22.234(23.89s)	23.787(296.57s)	24.981(258.88s)	23.850(416.46s)	**25.447**(269.86s)
	5%	25.385(58.63s)	27.496(257.74s)	28.191(350.25s)	28.304(266.63s)	**28.772**(151.15s)
	10%	26.307(82.83s)	28.809(333.09s)	30.008(162.52s)	29.826(319.39s)	**30.614**(169.22s)
	20%	27.044(136.99s)	29.741(494.33s)	31.829(270.07s)	31.751(508.58s)	**32.605**(210.86s)
	Time (s)	75.59	345.43	260.43	377.77	200.27

Bold: The best Underline: The second best

4.2. Real-World Data

Next, we evaluated the performance of the tensor network methods on two real-world datasets, including Multispectral Images (MSI: `https://cave.cs.columbia.edu/repository/Multispectral`) and YUV video sequences (`http://trace.eas.asu.edu/yuv/`). For the MSI dataset, we used the *Toy* image, represented as a tensor of size $200 \times 200 \times 31$ (corresponding to `height` × `width` × `spectral`). We considered two color video sequences from the YUV dataset, *News* and *Highway*, which were represented as a 4th-order tensors of size $144 \times 176 \times 3 \times 20$ and $144 \times 176 \times 3 \times 50$, respectively (corresponding to `height` × `width` × `color` × `time`). All tensors were normalized to the range $[0,1]$. Experiments were conducted under missing ratios of 99%, 95%, 90%, and 80%. As in the synthetic data case study above, the algorithm parameters were fine-tuned to achieve best performance.

Algorithm performance is evaluated using the mean peak signal-to-noise ratio (PSNR), where a higher PSNR indicates better reconstruction quality. The experimental results are illustrated graphically in Fig. 3 and summarized statistically in Tab. 2. As shown, TriNet achieves the best overall performance in terms of reconstructed image and video quality (i.e., highest PSNR value). In terms of runtime, it is moderate-slower than TT but considerably faster than TR and TW. For the 3rd-order *Toy* tensor, TriNet runs slower than FCTN; however, for higher-order tensors (*News* and *Highway*), it outperforms FCTN in speed.

5. CONCLUSIONS

In this paper, we introduced TriNet, a novel and efficient tensor network model that captures hidden interactions among core tensors while simultaneously reducing the orders of both core and central tensors, resulting in a more compact structure. Experiments on synthetic and real-world datasets demonstrate that TriNet achieves a balance between computational efficiency and accuracy, often outperforming state-of-the-art tensor network models. Future work will focus on developing adaptive and robust variants of TriNet to handle streaming data and address potential data corruptions.

6. REFERENCES

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

[2] L. T. Thanh, K. Abed-Meraim, N. L. Trung, and A. Hafiane, "A contemporary and comprehensive survey on streaming tensor decomposition," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 11, pp. 10 897–10 921, 2023.

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

[4] N. Tokcan, S. S. Sofi, C. Prévost, S. Kharbech, B. Magnier, T. P. Nguyen, A. Khoshnam, Y. Zniyed, and L. de Lathauwer, "Tensor decompositions for signal processing: Theory, advances, and applications," *Signal Process.*, vol. 238, p. 110191, 2026.

[5] C. J. Hillar and L.-H. Lim, "Most tensor problems are NP-hard," *J. ACM*, vol. 60, no. 6, p. 45, 2013.

[6] N. Vervliet, O. Debals, L. Sorber, and L. De Lathauwer, "Breaking the Curse of Dimensionality Using Decompositions of Incomplete Tensors: Tensor-based scientific computing in big data analysis," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 71–79, 2014.

[7] A. Cichocki, N. Lee, I. V. Oseledets, A.-H. Phan, Q. Zhao, and D. P. Mandic, "Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions," *Found. Trends Mach. Learn.*, vol. 9, no. 4-5, pp. 249–429, 2016.

[8] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2295–2317, 2011.

[9] Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki, "Tensor ring decomposition," *arXiv preprint arXiv:1606.05535*, 2016.

[10] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, Q. Zhao, and T.-X. Jiang, "Fully-connected tensor network decomposition and its application to higher-order tensor completion," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 12, 2021, pp. 11 071–11 078.

[11] Z.-C. Wu, T.-Z. Huang, L.-J. Deng, H.-X. Dou, and D. Meng, "Tensor wheel decomposition and its tensor completion application," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 27 008–27 020.

[12] W. Zhou, Y.-B. Zheng, Q. Zhao, and D. Mandic, "Tensor star decomposition," *arXiv preprint arXiv:2403.10481*, 2024.

[13] M. Kilmer and C. Martin, "Factorization strategies for third-order tensors," *Linear Algebra Appl.*, vol. 435, no. 3, pp. 641–658, 2011.

[14] S. Liu, X.-L. Zhao, J. Leng, B.-Z. Li, J.-H. Yang, and X. Chen, "Revisiting high-order tensor singular value decomposition from basic element perspective," *IEEE Trans. Signal Process.*, vol. 72, pp. 4589–4603, 2024.

[15] Y. Wang and Y. Yang, "Hot-SVD: higher order t-singular value decomposition for tensors based on tensor–tensor product," *Comput. Appl. Math.*, vol. 41, no. 8, p. 394, 2022.

[16] Y. Wang, W. Yin, and J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth optimization," *J. Sci. Comput.*, vol. 78, no. 1, pp. 29–63, 2019.

[17] L. Yuan, C. Li, D. Mandic, J. Cao, and Q. Zhao, "Tensor ring decomposition with rank minimization on latent space: An efficient approach for tensor completion," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 9151–9158.