Alternating Local Enumeration (TnALE): Solving Tensor Network Structure Search with Fewer Evaluations

Chao Li [1] **Junhua Zeng** [*2 1] **Chunmei Li** [*3 4] **Cesar Caiafa** [5 1] **Qibin Zhao** [1]

Abstract

Tensor network (TN) is a powerful framework in machine learning, but selecting a good TN model, known as TN structure search (TN-SS), is a challenging and computationally intensive task. The recent approach TNLS (Li et al., 2022) showed promising results for this task. However, its computational efficiency is still unaffordable, requiring too many evaluations of the objective function. We propose TnALE, a surprisingly simple algorithm that updates each structure-related variable alternately by local enumeration, *greatly* reducing the number of evaluations compared to TNLS. We theoretically investigate the descent steps for TNLS and TnALE, proving that both the algorithms can achieve linear convergence up to a constant if a sufficient reduction of the objective is *reached* in each neighborhood. We further compare the evaluation efficiency of TNLS and TnALE, revealing that $\Omega(2^K)$ evaluations are typically required in TNLS for *reaching* the objective reduction, while ideally $O(KR)$ evaluations are sufficient in TnALE, where K denotes the dimension of search space and R reflects the *"low-rankness"* of the neighborhood. Experimental results verify that TnALE can find practically good TN structures with vastly fewer evaluations than the state-of-the-art algorithms. Our code is available at `https://github.com/ChaoLiAtRIKEN/TnALE`.

[*]Equal contribution [1]RIKEN-AIP, Tokyo, Japan [2]School of Automation, Guangdong University of Technology, Guangzhou, China [3]College of Information and Communication Engineering, Harbin Engineering University, Harbin, China [4]Department of Computer Science and Communications Engineering, WASEDA University, Tokyo, Japan [5]Instituto Argentino de Radioastronomía, CONICET CCT La Plata/CIC-PBA/UNLP, V. Elisa, ARGENTINA . Correspondence to: Qibin Zhao <qibin.zhao@riken.jp>, Chao Li <chao.li@riken.jp>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

1. Introduction

Tensor network (TN) has been widely applied to solving high-dimensional problems in both machine learning (Anandkumar et al., 2014; Novikov et al., 2015; Zhe et al., 2015; Glasser et al., 2019; Kossaifi et al., 2020; Miller et al., 2021; Richter et al., 2021; Malik, 2022) and quantum physics (Orús, 2019; Felser et al., 2021). The success of *AlphaTensor* (Fawzi et al., 2022) once again confirmed the usefulness of tensors in various fields. TN practitioners, on the other hand, have to face in practice challenging problems associated with model selection, known as *TN structure search (TN-SS)*, for example: (1) how to determine the TN-ranks?; (2) should we prefer tensor-train (TT, Oseledets 2011), tensor-ring (TR, Zhao et al. 2016) or other TN topology?; (3) how to relate the tensor modes to core tensors of a TN (the permutation problem, Li et al. 2022), and so on. Unfortunately, some of these problems have been proven to be NP-hard (Hillar & Lim, 2013)[1], and most of them suffer from the *"combinatorial explosion"*[2] so that the brute force search is not a viable option in practice.

Several works have put effort into different aspects of TN-SS (see Section 1.1), but many of the methods are restricted to specific tasks or work poorly in practice, so a general and efficient TN-SS method is needed. Recently, Li et al. (2022) proposed an algorithm dubbed TNLS, which addressed the rank and permutation selection problem for TNs. However, its computational complexity is high since it requires evaluating the objective function on a large number of structure candidates.

To address this issue, we accelerate TNLS by equipping the algorithm with **A***lternating* **L***ocal* **E***numeration* — a surprisingly simple but efficient searching method in neighborhoods. The new algorithm, named *TnALE*, can improve the evaluation efficiency of TNLS *greatly*. To be specific, TnALE follows the "local-searching" scheme as TNLS but alternately updates each structure-related variable by enumerating its values within a neighborhood. The intuition

[1]For instance, it proves that determining the optimal ranks for Tucker decomposition (Tucker, 1966) is NP-hard.

[2]It means the rapid growth of TN structure searching space due to the combinatorics of ranks, topologies, permutations, etc.

is that the alternately updating avoids the combinatorial explosion and the enumeration in neighborhoods guarantees the non-increasing of values of the objective in the search. On the other hand, the original TNLS applies random sampling, causing the majority of samples would not provide helpful information for decreasing the value of the objective function.

The theoretical advantage of TnALE is also clear. We prove that, with new-defined *discrete* convexity-related assumptions of the objective function, both TNLS and TnALE can achieve a linear convergence up to a constant if a sufficient reduction of the objective function is reached in each neighborhood (Theorem 4.5). We also prove that the number of evaluations required in TNLS would grow *exponentially* with the dimension of search space (Prop. 4.8), with respect to the dimension of TN-ranks and the TN order, while TnALE shows a *linear* growth in the ideal case (Prop. 4.10). Our analysis reveals that such an improvement in the evaluation efficiency essentially comes from the *low-rankness* of the optimization landscape in neighborhoods, attributed to the close relationship between TnALE and cross-approximation methods for matrices and tensors (Tyrtyshnikov, 2000; Oseledets & Tyrtyshnikov, 2010; Sozykin et al., 2022). Numerically, extensive experiments on both synthetic and real-world data are implemented to assess the evaluation efficiency and the superior performance provided by TnALE.

Our main contributions can be summarized as follows:

- We propose TnALE, a novel algorithm that *greatly* reduces the computational cost for the task of TN structure search (TN-SS);
- We establish for the first time the convergence analysis for both TNLS (Li et al., 2022) and TnALE, and rigorously prove their evaluation efficiency.

1.1. Related Works

Tensor network structure search (TN-SS). TN-SS can be specified into three sub-problems: (1) TN-rank selection (TN-RS) (Rai et al., 2014; Zhao et al., 2015; Yokota et al., 2016; Cheng et al., 2020; Mickelin & Karaman, 2020; Cai & Li, 2021; Hawkins & Zhang, 2021; Long et al., 2021; Sedighin et al., 2021; Yin et al., 2022; Ghadiri et al., 2023); (2) TN-topology selection (TN-TS) (Hashemizadeh et al., 2020; Haberstich et al., 2021; Nie et al., 2021; Falcó et al., 2023; Hikihara et al., 2023; Kodryan et al., 2023; Liu et al., 2023); and (3) TN-permutation selection (TN-PS) (Acharya et al., 2022; Chen et al., 2022). Recently, some methods to solve the TN-SS problem were developed via discrete optimization (Hayashi et al., 2019; Hashemizadeh et al., 2020; Li & Sun, 2020; Li et al., 2021; 2022; Solgi et al., 2022). Although these methods typically achieve higher precision than their counterparts in practice, they suffer from the expensive computational cost and the lack of theoretical analysis. Our work follows the TNLS algorithm (Li et al., 2022) in this direction but develops a new approach to improve its computational efficiency and fill in the missing theoretical analysis for convergence and evaluation efficiency.

Finding the extreme entry value within a tensor. As discussed later in Section 4.2, the *alternating local enumeration* method is technically equivalent to finding the minimum entry within a multidimensional landscape. As such, our work is strongly related to the recently published method TTOpt (Sozykin et al., 2022), which finds the near-maximum entry of a tensor by cross-sampling (Tyrtyshnikov, 2000; Zhang, 2019) in TT topology (Oseledets & Tyrtyshnikov, 2010). Compared to TTOpt, the proposed TnALE recursively finds the extreme entry within a tensor associated with the neighborhood rather than the global search space, so that TnALE can handle the situation of *infinite* candidates (entries) in the optimization.

2. Preliminaries

In this section, we first summarize notations and review several central concepts related to the tensor network structure search (TN-SS). Then, we provide a quick review of TNLS (Li et al., 2022), a recently proposed algorithm for solving TN-SS, and point out that *TNLS suffers from the curse of dimensionality* in evaluation efficiency.

2.1. Notations

Throughout the paper, we typically use blackboard letters to denote sets of objects, *e.g.*, $\mathbb{G}, \mathbb{F}$. In particular, $\mathbb{R}$, $\mathbb{Z}_+$ and $\mathbb{Z}_{\geq 0}$ denote real numbers, positive integers and non-negative integers, respectively. We use boldface letters to denote vectors and matrices, *e.g.*, $\mathbf{x} \in \mathbb{Z}_+^K$ and $\mathbf{A} \in \mathbb{R}^{I \times J}$. For tensors of *arbitrary* order, we denote them using calligraphic letters, *e.g.*, $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. Given a vector, such as $\mathbf{x} \in \mathbb{Z}_+^K$, $\|\mathbf{x}\|$ and $\|\mathbf{x}\|_\infty$ denote the l_2-norm and l_∞-norm of $\mathbf{x}$, respectively. The norms are also directly applied to matrices and tensors by thinking of them as vectors. Following the notational conventions, $|x|$ denotes the absolute value of x if $x \in \mathbb{R}$ is a scalar, while $|\mathbb{A}|$ denotes the cardinality if $\mathbb{A}$ is a set. For the normed vector spaces armed with $\|\mathbf{x}\|_\infty$, we use $B_\infty(\mathbf{x}, r_\mathbf{x})$ to denote the neighborhoods centered at $\mathbf{x}$ with the radius $r_\mathbf{x} > 0$. For any two functions $f : \mathbb{B} \to \mathbb{C}$ and $g : \mathbb{A} \to \mathbb{B}$, the operation $f \circ g : \mathbb{A} \to \mathbb{C}$ denotes the function composition.

2.2. Tensor Network Structure Search (TN-SS)

We consider the *tensor network* (TN, Ye & Lim, 2019) as a set of real tensors of the dimension $I_1 \times I_2 \times \cdots \times I_N$, denoted $TNS(G, \mathbf{r})$, whose elements are in the form of

contractions of *core tensors* (Cichocki et al., 2017), associated to the TN structure modeled by the pair $(G, \mathbf{r})$, where $G = (V, E)$ denotes a simple *graph* of N vertices modeling the *TN-topology* (Li & Sun, 2020) and $\mathbf{r} \in \mathbb{Z}_+^{|E|}$ is a vector, whose entries are edge labels of G corresponding to the *TN-ranks* (Ye & Lim, 2019).

Tensor network structure search (TN-SS) aims generally to find the most compressed TN models for computational purposes while preserving the models' expressivity. Furthermore, the most compressed TN models also imply the potential advantage for the generalization capability in learning tasks (Khavari & Rabusseau, 2021). Suppose a dataset D and the task-specific loss function $\pi_D : \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \to \mathbb{R}_+$ involving D. TN-SS is to solve the following bi-level discrete optimization problem

$$\min_{(G,\mathbf{r}) \in \mathbb{G} \times \mathbb{F}_G} \left(\phi(G, \mathbf{r}) + \lambda \cdot \min_{\mathcal{Z} \in TNS(G,\mathbf{r})} \pi_D(\mathcal{Z}) \right), \quad (1)$$

where $G \in \mathbb{G}$ is a graph of N vertices and K edges, $\mathbf{r} \in \mathbb{F}_G \subseteq \mathbb{Z}_+^K$, $\phi : \mathbb{G} \times \mathbb{Z}_+^K \to \mathbb{R}_+$ represents the function measuring the model complexity of a TN whose structure is modeled by $(G, \mathbf{r})$, and $\lambda > 0$ is a tuning parameter. The intuition of (1) is that, the inner minimization is to evaluate the task-specific expressivity for a TN structure, while the outer minimization is to find the optimal structure for the task by balancing the complexity and the expressivity of a TN model.

We remark that the formulation (1) can be specified as different TN-SS sub-problems by restricting the feasible set $\mathbb{G} \times \mathbb{F}_G$ into different forms: for TN-PS, we specify $\mathbb{F}_G = \mathbb{Z}_+^K$ and $\mathbb{G}$ to be the set containing the isomorphic graphs to a "template" graph (Li et al., 2022); for TN-RS, it typically restricts G to be fixed, and only finds TN-ranks *i.e.*, $\mathbb{F}_G = \mathbb{Z}_+^K$; last for TN-TS, it relaxes $\mathbb{G}$ to be the set containing all possible simple graphs of N vertices and $\mathbf{r}$ is set to be fixed (Li & Sun, 2020) or searchable (Hashemizadeh et al., 2020). Note from Ye & Lim (2019); Li & Zhao (2021) that TN-TS (with rank selection) essentially coincides with TN-RS of a "fully-connected" TN (Zheng et al., 2021).

2.3. TNLS: a Discrete Optimization Approach to TN-PS

Recently, Li et al. (2022) proposed an algorithm called TNLS for solving (1) effectively by stochastic search. The core process of TNLS is reviewed in Alg. 1, from which we see that the candidate of the optimal TN structure is updated if the algorithm finds better structures within a neighborhood using *random sampling*. Although Li et al. (2022) illustrates the superiority of TNLS compared to its counterparts, the algorithm is still time-consuming as acknowledged in their work. To understand the reason, we theoretically observe that TNLS suffers from the curse of dimensionality, due to the random sampling. More specifically, we state that

Algorithm 1 *The core process* of TNLS (Li et al., 2022).

Initialize: Randomly choose a TN structure as the center of the neighborhood.
while not convergence **do**
1. Sampling $(G, \mathbf{r})$'s <u>*randomly*</u> in the neighborhood;
2. Evaluating the samples with the objective of (1);
3. Updating the center if better samples are obtained;

end while
Output: The center of the neighborhood.

under reasonable conditions, $\Omega(2^K/\epsilon)$ samples are required by random sampling (wrt. step 1 in Alg. 1) for achieving a constant probability $Pr \geq \epsilon$ for decreasing the objective of (1) *in a neighborhood, where $\epsilon > 0$ and K denotes the dimension of the search space.*

A formal statement is deferred to Prop. 4.8. It is known from Alg. 1 that each sampled structure should be evaluated by solving the inner minimization of (1), so the huge number of samples implies the prohibitive cost in computation. To address this problem, we introduce a more sampling-efficient approach by reforming the random sampling in Alg. 1, to accelerate the TN-SS procedure with fewer evaluations.

3. TnALE: Accelerating TNLS via Alternating Local Enumeration

We present now the new searching algorithm dubbed *TnALE* for solving the TN-SS problem. Figure 1 demonstrates the key idea for TnALE. In the rest of this section, we mainly focus on the technical details of *alternating local enumeration (ALE)*, which replaces the random sampling operation in Alg. 1 as the key factor for algorithm acceleration. A complete introduction of TnALE, including the pseudocode and other details, is given in Appendix A.

In TnALE, we find better structures within a neighborhood by updating each structure-related variable alternately. For instance, Figure 2 illustrates how ALE solves the TN-PS problem, searching for the optimal ranks and permutations for tensor ring (TR) decomposition of order-4. As shown in the initialization of panel (b), all structure-related variables, including $r_{\{1,2,3,4\}}$ and G, are initialized with the center of a given neighborhood. To start the search, TN structures are sampled by enumerating all values of r_1 *within* the neighborhood while fixing other variables. Next, the sampled TN structures with varying r_1 are evaluated individually by calculating the objective of (1), and r_1 is subsequently updated right off by choosing the one with the minimum objective in the samples. Following r_1, the same operation is applied to variables from r_2 to G sequentially (see panel (b)). After updating G, we turn the updating direction backward from r_4 to r_1, *i.e.*, in a "round-trip" manner (see panel (c)). Overall, the ALE will be stopped if all variables are not changed

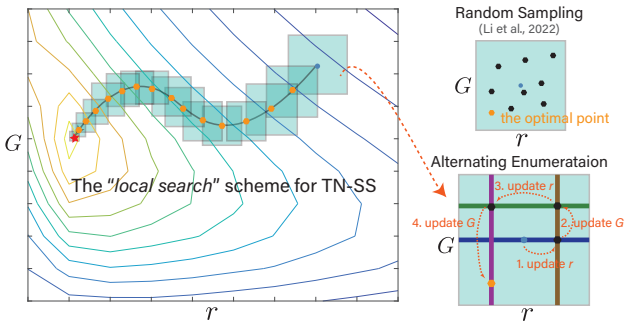


Figure 1. Schematic demonstration of *TnALE* and its discrepancy from TNLS (Li et al., 2022), where r, G denote two structure-related variables for example, and the squares represent the neighborhoods. The alternating (local) enumeration is further illustrated in detail in Figure 2.

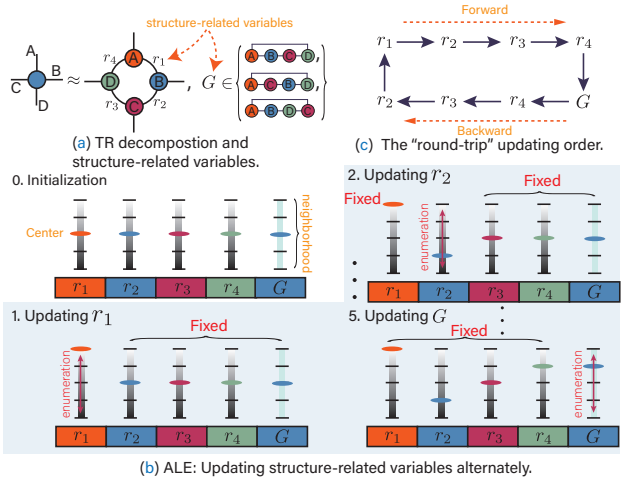


Figure 2. Illustration of *alternating local enumeration (ALE)* for TN-PS of TR decomposition. The search for TN-RS and TN-TS is similar, except that the ranges of the TN-ranks $r_{\{1,2,3,4\}}$ and the graph G need to be adjusted.

anymore. We empirically find that a *one-time* "round-trip" is sufficient to reach a good TN structure for the next iteration. Note that the operation of ALE for TN-RS and TN-TS is in the same fashion, except that the graph G will be fixed in TN-RS or enumerated in differently-defined neighborhoods in TN-TS for considering all TN-topologies. In TnALE, we follow Li et al. (2022) to construct the neighborhood of TN structures.

Remark 3.1 (**tricks: knowledge transfer**). An additional merit of implementing enumeration is the *"knowledge transfer"* capability (Hashemizadeh et al., 2020). We know that increasing the TN-ranks would decrease *monotonically* the value of the objective function in many learning tasks. Instead of evaluating each structure explicitly, it thus inspires us to accelerate the structure evaluation by reusing in enumeration the knowledge of the well-optimized core tensors and their corresponding objective. In particular, the acceleration by the knowledge transfer trick is two-fold: one is to use the well-optimized core tensors associated with the lower-rank structures to be the initialization for the ones with higher-rank structures, as in Hashemizadeh et al. (2020); the other is to employ the *objective estimation*, in which we apply linear interpolation to predicting the objective in the evaluation phase in place of the explicit calculation (see Appendix A.1). Although the objective estimation would be of no precision, it helps in the first $1 \sim 2$ iterations of TnALE (with a large radius) as initialization for quickly finding a reasonable structure candidate.

Remark 3.2 (**computational complexity**). TnALE is a *meta*-algorithm for TN-SS, in which the inner minimization of (1) can be solved by any practitioner-appointed algorithms. Therefore, the computational complexity of TnALE is mainly affected by the number of evaluations. Suppose the searching problem of order-N with the TN-ranks of dimension K. Furthermore, suppose each entry r_i, $i \in [K]$ of $\mathbf{r}$ is enumerated in I values in the neighborhood, *e.g.*, the interval $[r_i - I/2, r_i + I/2]$, and D times of the "round-trip" updates. In this setting, for the TN-RS problem, TnALE requires $O(DKI)$ evaluations in one neighborhood; for TN-PS, it requires $O(DKI + DN^2/2)$ evaluations since the neighborhood of G in TN-PS contains $(N-1)N/2$ elements (Li et al., 2022) in general; last for TN-TS, $O(DN^2I)$ evaluations are required. In practice, the value of I is typically set to 3 or 5 and $D = 1$.

In summary, the evaluation complexity of TnALE grows polynomially with the TN-order and the dimension of the TN-ranks. In the next section, we prove that such the number of evaluations is theoretically sufficient in TnALE for achieving quick convergence for TN-SS.

4. Theoretical Results

In this section, we first analyze the descent steps for both TNLS and TnALE, proving that using the "local search" (Li et al., 2022) scheme the algorithms would achieve a linear convergence rate up to a constant if the objective is sufficiently "convex" in the *discrete* domain. Following this, we analyze the evaluation efficiency for TNLS and TnALE, showing that the required number of evaluations in TNLS would grow exponentially with the dimension of the search space, while it can be ideally reduced to be a linear growth in TnALE if the neighborhood is *low-rank*. The proofs in this section are given in Appendix B.

4.1. Analysis of Descent Steps

We start the analysis by rewriting (1) into a more general form

$$\min_{\mathbf{x} \in \mathbb{Z}_+^K, p \in \mathbb{P}} f_p(\mathbf{x}) := f \circ p(\mathbf{x}), \tag{2}$$

where $f : \mathbb{Z}_{\geq 0}^L \to \mathbb{R}_+$ is a general form of the objective function of (1), $\mathbf{x} \in \mathbb{Z}_+^K$ corresponds to the TN-ranks $\mathbf{r}$ of (1), and the operator $p : \mathbb{Z}_+^K \to \mathbb{Z}_{\geq 0}^L$ and its feasible set $\mathbb{P}$ correspond to the graph variable G and its feasible set $\mathbb{G}$ of (1), respectively. The specific relationship of p and G is discussed in Appendix B.2.

The framework used in the proof follows from Golovin et al. (2019) for the zeroth-order convex optimization. However, due to the discrete essence of (2), we re-establish discrete analogues of the fundamental concepts such as the gradient, strong convexity and smoothness for the analysis, and all the proofs are re-derived non-trivially in the discrete domain.

In doing so, we begin with the definition of the finite gradient (Olver, 2014), as the alternative to the classic gradient in the continuous domain.

Definition 4.1 (finite gradient). For any function $f : \mathbb{Z}_{\geq 0}^L \to \mathbb{R}$, its finite gradient $\Delta f : \mathbb{Z}_{\geq 0}^L \to \mathbb{R}^L$ with respect to $\mathbf{x} \in \mathbb{Z}_{\geq 0}^L$ is defined as follows:

$$\Delta f(\mathbf{x}) = \left[f(\mathbf{x}+\mathbf{e}_1) - f(\mathbf{x}), \ldots, f(\mathbf{x}+\mathbf{e}_L) - f(\mathbf{x})\right]^\top, \tag{3}$$

where $\mathbf{e}_i$, $1 \leq i \leq L$ denote the unit vectors with the i-th entry being one and the others being zeros.

Next, we redefine the convexity and smoothness of the objective with finite gradients.

Definition 4.2 (α-strong convexity with finite gradient). We say f is α-strongly convex for $\alpha \geq 0$ if $f(\mathbf{y}) \geq f(\mathbf{x}) + \left\langle \Delta f(\mathbf{x}) - \frac{\alpha}{2}\mathbf{1}, \mathbf{y}-\mathbf{x}\right\rangle + \frac{\alpha}{2}\|\mathbf{y}-\mathbf{x}\|^2$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_{\geq 0}^L$, where $\mathbf{1} \in \mathbb{R}^L$ denotes the vector with all entries being one. We simply say f is convex if it is α-strongly convex and $\alpha = 0$.

Definition 4.3 ((β_1, β_2)-smoothness with finite gradient). We say f is (β_1, β_2)-smooth for $\beta_1, \beta_2 > 0$ if

1. $|f(\mathbf{x}) - f(\mathbf{y})| \leq \beta_1 \|\mathbf{x}-\mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_{\geq 0}^L$;
2. The function $l(\mathbf{x}) := \frac{\beta_2}{2}\|\mathbf{x}\|^2 - f(\mathbf{x})$ is convex.

We remark that Definition 4.2 and 4.3 are partially different from the ones used in Golovin et al. (2019) or other literature for convex analysis. Particularly in Definition 4.3, the smoothness is defined by additionally taking the Lipschitz continuity (corresponding to *Item 1*) into account, which controls the changing rate of f, while *Item 2* in Definition 4.3 controls the changing rate of the finite *gradient* of f. See Lemma B.5 in Appendix for the discussion. With the new definitions, we next give the main assumptions used in the results.

Assumption 4.4. Assume that $f : \mathbb{Z}_{\geq 0}^L \to \mathbb{R}_+$ of (2) is α-strongly convex, (β_1, β_2)-smooth, and its minimum, denoted $(p^*, \mathbf{x}^*) = \arg\min_{p,\mathbf{x}} f \circ p(\mathbf{x})$, satisfies $\|\Delta f_{p^*}(\mathbf{x}^*) - \frac{\beta_2}{2}\mathbf{1}\| \leq \gamma$ where $0 \leq \gamma < \alpha \leq \beta_1 \leq \beta_2 \leq 1$.

In Assumption 4.4, the inequality $\|\Delta f_{p^*}(\mathbf{x}^*) - \frac{\beta_2}{2}\mathbf{1}\| \leq \gamma$ implies that, up to a (small) bias $\frac{\beta_2}{2}\mathbf{1}$, the finite gradient at $(p^*, \mathbf{x}^*)$ should be sufficiently small, which can be analogous to the "gradient-equal-zero" (Boyd & Vandenberghe, 2004) property of the stationary points for convex functions in the continuous domain. The upper bound "1" is arbitrarily chosen just for simplifying the calculation.

Next, we reveal that the local-search scheme, used in both TNLS and TnALE, can achieve the linear convergence rate up to a constant. We first focus on TN-RS and TN-TS to simplify the problem, where p^* is assumed to be *known* beforehand. After that, we discuss TN-PS, showing that the searchable p would make the convergence more difficult.

Theorem 4.5 (convergence rate when p^* is known). *Suppose Assumption 4.4 is satisfied, the operator p of (2) is fixed to be p^*, and $0 \leq \theta \leq 1$. Then, for any $\mathbf{x}$ with $\|\mathbf{x}-\mathbf{x}^*\|_\infty \leq c$, we can find a neighborhood $B_\infty(\mathbf{x}, r_\mathbf{x})$ where $r_\mathbf{x} \geq \theta c + \frac{1}{2}$, such that there exists an element $\mathbf{y} \in B_\infty(\mathbf{x}, r_\mathbf{x})$ satisfying*

$$f_{p^*}(\mathbf{y}) - f_{p^*}(\mathbf{x}^*) \leq (1-\theta)(f_{p^*}(\mathbf{x}) - f_{p^*}(\mathbf{x}^*)) + \frac{7}{8}K. \tag{4}$$

Proving Theorem 4.5 requires the following lemma, which can be understood as the *discrete* version of the convex-combination inequality of convex functions.

Lemma 4.6 (convex combination in the discrete domain). *Suppose $\mathbf{q} = \theta\mathbf{x} + (1-\theta)\mathbf{y}$, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{Z}_{\geq 0}^L$, $\theta \in [0,1]$, and there is $\hat{\mathbf{q}} \in \mathbb{Z}_{\geq 0}^L$ following $\mathbf{\Lambda} = \mathbf{q} - \hat{\mathbf{q}}$. If f is α-strongly convex, then*

$$\theta f(\mathbf{x}) + (1-\theta) f(\mathbf{y}) \geq f(\hat{\mathbf{q}}) + \left\langle \Delta f(\hat{\mathbf{q}}) - \frac{\alpha}{2}\mathbf{1}, \mathbf{\Lambda}\right\rangle + \frac{\alpha}{2}\|\mathbf{\Lambda}\|^2. \tag{5}$$

Note that the inequality (5) would be the same as the crucial inequality $\theta f(\mathbf{x}) + (1-\theta) f(\mathbf{y}) \geq f(\mathbf{q})$ in convex analysis if $\mathbf{\Lambda} = 0$. However, due to $\mathbf{q} \notin \mathbb{Z}_{\geq 0}^L$ in general, the non-zero $\mathbf{\Lambda}$ is inevitable in the proof, yielding the essential difference from the convex analysis in the continuous domain. As a consequence, the inequality (4) shows that the convergence rate is formally close to being linear, but the constant $(7/8)K$ appears on the right-hand side *dampening* the search efficiency.

Suppose the search trajectory $\{f_{p^*}(\mathbf{x}_n)\}_{n=0}^\infty$, of which the starting point $\mathbf{x}_0 \in \mathbb{Z}_+^K$ is randomly chosen and $\mathbf{x}_n$ for $n > 0$ are determined by the vector $\mathbf{y}$ in Theorem 4.5. As an important corollary, it can be easily proved that $\{f_{p^*}(\mathbf{x}_n)\}_{n=0}^\infty$ converges to $f_{p^*}(\mathbf{x}^*)$ up to a constant if $\Omega(1/K) \leq \theta \leq 1$. A rigorous proof for the convergence guarantee can be found in Appendix B.12.

Remark 4.7 (**Finding p^* makes the convergence slower**.). As aforementioned, the non-zero $\|\mathbf{\Lambda}\|_\infty$ decreases the search efficiency due to the additional constant shown in (4). It is known from the proof of Theorem 4.5 that the constant is derived from the tight bound $\|\mathbf{\Lambda}\|_\infty \leq 1/2$, following the rounding operation. However, once the p in (2) is searchable as well, $\|\mathbf{\Lambda}\|_\infty$ would turn larger, dampening the convergence more seriously. To verify this, suppose $\mathbf{q} = \theta p^*(\mathbf{x}^*) + (1-\theta)p_{\mathbf{x}}(\mathbf{x})$ to be the convex combination between $(p^*, \mathbf{x}^*)$ and any point $(p_{\mathbf{x}}, \mathbf{x})$. It is known from the proof that, for decreasing the objective, $\hat{\mathbf{q}}$ should satisfy $\hat{\mathbf{q}} \in B_\infty(\mathbf{q}, \|\mathbf{\Lambda}\|_\infty) \cap \mathbb{B}(p_{\mathbf{x}}, \mathbf{x})$, where $\mathbb{B}(p_{\mathbf{x}}, \mathbf{x}) := \{\mathbf{z} = \bar{p}(\bar{\mathbf{x}}) : \bar{p} \in B(p_{\mathbf{x}}), \bar{\mathbf{x}} \in B_\infty(\mathbf{x}, r_{\mathbf{x}})\}$ for some $r_{\mathbf{x}} > 0$ and $B(p_{\mathbf{x}})$ denotes the neighborhood of $p_{\mathbf{x}}$ chosen in the algorithm. We thus see that, for the existence of $\hat{\mathbf{q}}$, the intersection $B_\infty(\mathbf{q}, \|\mathbf{\Lambda}\|_\infty) \cap \mathbb{B}(p_{\mathbf{x}}, \mathbf{x})$ should be non-empty. Following this, it satisfies $\|\mathbf{\Lambda}\|_\infty \geq \min_{\bar{\mathbf{q}} \in \mathbb{B}(p_{\mathbf{x}}, \mathbf{x})} \|\mathbf{q} - \bar{\mathbf{x}}\|_\infty \geq \min_{\hat{\mathbf{q}} \in \mathbb{Z}^L_{\geq 0}} \|\mathbf{q} - \hat{\mathbf{q}}\|_\infty = 1/2$ in the worst case. In this case, the larger value of $\|\mathbf{\Lambda}\|_\infty$ means a larger damping term appearing on the right-hand side of (4).

4.2. Evaluation Efficiency

Note that a premise for achieving the closely linear convergence rate by TNLS and TnALE is that the expected $\mathbf{y} \in B_\infty(\mathbf{x}, r_{\mathbf{x}})$ in Theorem 4.5 is reachable, meaning that the algorithms *should* find the $\mathbf{y}$ out in each neighborhood. In the rest of the section, we show that TNLS is required to cost $\Omega(2^K)$ samples in each neighborhood for stably reaching the $\mathbf{y}$, while $O(KIR)$ samples are ideally sufficient for TnALE. Here K denotes the dimension of the search space, $I \in \mathbb{Z}_+$ indicates an integer related to the radius of the neighborhood and $R \in \mathbb{Z}_+$ reflects the *low-rankness of the optimization landscape* in neighborhoods.

We first give the proposition for TNLS as follows.

Proposition 4.8 (**curse of dimensionality for TNLS**). *Let the assumptions in Theorem 4.5 be satisfied. Furthermore, assume that $\mathbf{x}^*$ is sufficiently smaller (or larger) than $\mathbf{x}$ entry-wisely, except for a constant number of entries. Then the probability of achieving a suitable $\mathbf{y}$ as mentioned in Theorem 4.5 by uniformly randomly sampling in $B_\infty(\mathbf{x}, r_{\mathbf{x}})$ with $r_{\mathbf{x}} \geq \theta c + \frac{1}{2}$ equals $O(2^{-K})$.*

Note that the additional assumption in Prop. 4.8 is commonly satisfied in practice when the searched TN-ranks are initialized uniformly with large (or small) values. It is also easily known from Prop. 4.8 that $\Omega(2^K/\epsilon)$ samples are required for achieving the probability $Pr \geq \epsilon$ for reaching the $\mathbf{y}$ in the neighborhood.

Remark 4.9. The intuition of Prop. 4.8 is as follows. Suppose $\mathbf{x}^*$ is entry-wisely smaller than $\mathbf{x}$ without loss of generality, then the objective would be decreased only if most of the entries of $\mathbf{x}$ are updated to be smaller values, *i.e.*, getting closer to $\mathbf{x}^*$. However, by random sampling, the probability of decreasing most entries of $\mathbf{x}$ would turn smaller exponentially with increasing the dimension K, yielding the curse of dimensionality for TNLS.

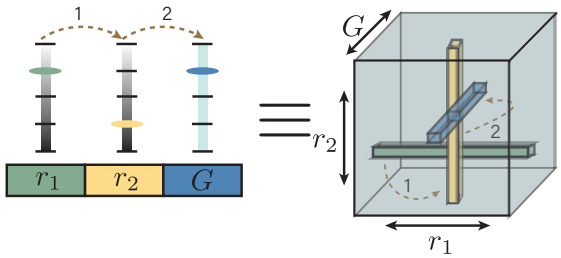


Figure 3. The relationship between alternating local enumeration (ALE) and TT-cross approximation (Oseledets & Tyrtyshnikov, 2010; Sozykin et al., 2022). As shown in the figure, enumerating structure-related variables alternately is equivalent to sampling fibers of a tensor along each mode. The yellow arrows indicate the alternation of variables from r_1 to r_2 and then to G, respectively.

In contrast to TNLS, TnALE essentially resolves the curse of dimensionality by leveraging the landscape's low-rank structure. To verify this, given $(p, \mathbf{x})$, we formulate first the neighborhood $B(p) \times B_\infty(\mathbf{x}, r_{\mathbf{x}})$ as a $(K+1)$-th order tensor $\mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{K+1}}$. Here $I_k = 2 \times \lceil r_{\mathbf{x}} \rceil + 1$ for $1 \leq k \leq K$ and $I_{K+1} = |B(p)|$. The $(i_1, i_2, \ldots, i_{K+1})$-th entry of $\mathcal{B}$, written $\mathcal{B}(\mathbf{i})$ with $\mathbf{i} = [i_1, i_2, \ldots, i_{K+1}]^\top$, satisfies

$$\mathcal{B}(\mathbf{i}) = 1/f_{p_{i_{K+1}}}(\mathbf{x} + \mathbf{i}(:K) - (\lceil r_{\mathbf{x}} \rceil + 1)), \qquad (6)$$

where $\mathbf{i}(:K)$ denotes the K-dimensional vector consisting of the first K entries of $\mathbf{i}$, and $p_{i_{K+1}}$ denotes the i_{K+1}-th element of $B(p)$ in any ordering fashion. We remark that the inverse $1/f(\mathbf{x})$ is always valid due to the assumption $f_p(\mathbf{x}) > 0$ in (2) for all $\mathbf{x}$ and p. We also remark that the equation (6) maps uniquely each element of $B(p) \times B_\infty(\mathbf{x}, r_{\mathbf{x}})$ onto the entries of $\mathcal{B}$.

By the tensor $\mathcal{B}$, we can find that the proposed *alternating local enumeration (ALE)* is strongly related to TT-cross (Oseledets & Tyrtyshnikov, 2010) and TTOpt (Sozykin et al., 2022). As demonstrated in Figure 3, the enumeration for each variable is equivalent to drawing a fiber of $\mathcal{B}$ as in TT-cross or TTOpt with the TT-ranks being *ones*. Such a relationship helps us reveal the evaluation advantage of TnALE. Specifically, let $\mathbb{B} := B(p) \times B_\infty(\mathbf{x}, r_{\mathbf{x}})$ and $f^*_{\mathbb{B}} := \min_{(p_y, \mathbf{y}) \in \mathbb{B}} f_{p_y}(\mathbf{y})$ for notational simplicity, then we have the following proposition.

Proposition 4.10 (**evaluation efficiency for TnALE**). *Let $\mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{K+1}}$ be the tensor of order-$(K+1)$ constructed as Eq. (6) with $I_1 = I_2 = \cdots = I_{K+1} = I$ for simplicity. Then, there exists its TT-cross approximation (Oseledets & Tyrtyshnikov, 2010) of rank-R[3], denoted*

[3]Here all elements of the TT-ranks equal R for simplicity.

$\hat{\mathcal{B}}$, such that $f^*_{\mathbb{B}} = f_{p_{j_{K+1}}}(\mathbf{x} + \mathbf{j}(:K) - (\lceil r_{\mathbf{x}} \rceil + 1))$ with $\mathbf{j} = \arg\max_{\mathbf{i}} \hat{\mathcal{B}}(\mathbf{i})$ holds, provided that

$$f^*_{\mathbb{B}} \leq f_{p_z}(\mathbf{z}) / \left(1 + 2\frac{(4R)^{\lceil \log_2 K \rceil} - 1}{4R - 1}(R+1)^2 \xi f_{p_z}(\mathbf{z})\right) \tag{7}$$

for all $(p_z, \mathbf{z}) \in \mathbb{B}$ and $f_{p_z}(\mathbf{z}) \neq f^*_{\mathbb{B}}$. Here, ξ denotes the error between $\mathcal{B}$ and its best approximation of TT-ranks R in terms of $\|\cdot\|_\infty$. Note that the inequality (7) holds trivially if $\mathcal{B}$ is exactly of the TT topology of rank-R, and Oseledets & Tyrtyshnikov (2010) shows that the $f^*_{\mathbb{B}}$ can be recovered from $O(KIR)$ entries from $\mathcal{B}$.

Prop. 4.10 is a natural corollary of Theorem 2 in Osinsky (2019). It implies that the desired $\mathbf{y}$ (corresponding to $f^*_{\mathbb{B}}$ in Prop. 4.10) in Theorem 4.5 is reachable by only $O(KIR)$ samples once $\mathcal{B}$ is of TT with rank-R. Even though $\mathcal{B}$ is not low-rank, the $\mathbf{y}$ can still be located if the inequality (7) is satisfied.

Prop. 4.10 shows the $O(KIR)$ evaluation advantage compared with TNLS that requires $\Omega(2^K/\epsilon)$ evaluations in the neighborhoods, but it remains open to prove the low-rankness of the optimization landscape in the TN-SS tasks. We empirically verify this with five synthetic tensors of order four. We calculate their complete optimization landscape associated with the l_2 loss, observing that the multidimensional landscape is indeed low-rank under all possible unfoldings (see Figure 6 in Appendix). We thus conjecture that in practice the low-rank structure of the landscape should be preserved, at least in neighborhoods. In the next section, the evaluation advantage by TnALE will be empirically verified with both synthetic and real-world data.

5. Experimental Results

In this section, we present numerical results to verify the superiority of TnALE in terms of evaluation cost. Due to the page limit, the experimental settings will be presented at the minimum level of clarity. Additional details are given in Appendix C.

5.1. Synthetic Data

First of all, we assess the superiority of TnALE by solving the TN-PS problem, in which both the optimal TN-ranks and permutations of synthetic tensors are searched for the tensor decomposition task.

In the experiment, we *re-use* from Li et al. (2022) the synthetic tensors, which are randomly generated in the topologies including TR (order-8), PEPS (order-6, Verstraete & Cirac, 2004), hierarchical Tucker (HT of order-6, Hackbusch & Kühn, 2009), and MERA (order-8, Cincio et al., 2008). Additionally, we also consider the *tensor wheel* model (TW of order-5, Wu et al., 2022). Since the mode dimension is typically irrelevant to the difficulty of TN-SS, we set them to equalling 3 in all tensors for simplicity. For each topology, four tensors (**A, B, C, D**) are generated, where the TN-ranks and permutations are randomly selected and remained unknown. The goal of this experiment is to compare different approaches to identifying the TN-ranks and permutations for each tensor, meaning that the conditions RSE$\leq 10^{-4}$ and the *Eff.*≥ 1 are satisfied[4]. Otherwise, we say the approach fails in the experiment.

Table 1. Number of evaluations, denoted "*#Eva.*", for the rank and permutation identification, where the symbol "-" in the table means the failure of the approach.

Topology	Methods	Data – *#Eva.* ↓			
		A	**B**	**C**	**D**
TR	TNGA	2850	2250	3900	1950
	TNLS	1020	960	1320	780
	Ours	**231**	**308**	**308**	**231**
PEPS	TNGA	1560	-	840	1080
	TNLS	781	781	421	661
	Ours	**407**	**465**	**233**	**175**
HT	TNGA	960	1320	840	1080
	TNLS	841	841	781	721
	Ours	**211**	**281**	**211**	**211**
MERA	TNGA	-	960	2800	3240
	TNLS	1561	841	1441	721
	Ours	**1450**	**484**	**323**	**323**
TW	TNGA+	1920	1440	600	720
	TNLS	661	601	601	481
	Ours	**285**	**143**	**285**	**214**

We implement three algorithms, TNGA (Li & Sun, 2020), TNLS and TnALE (ours). Note that in TNGA both the TN ranks rank permutations are encoded as chromosomes, as implemented in the work by Li et al. (2022). In the subsequent experiments, the the encoding scheme of TNGA would be also properly adjusted for handling different subproblems of TN-SS. For a fair comparison, the three approaches use the same objective and solver for the inner minimization of (1). Furthermore, TNLS and TnALE are initialized with the same TN structures. The rest of the experimental settings remain as Li et al. (2022).

The experimental results are shown in Table 1. We see that both TNLS and TnALE (ours) successfully identify the ranks and permutations for all tensors. Furthermore, TnALE requires *significantly* fewer evaluations than both TNGA and TNLS. Figure 4 further illustrates the change of the objective

[4]RSE means the relative squared error, and the *Eff.* index (Li & Sun, 2020) denotes the ratio of the parameter number of TNs between the searched structure and the one used in data generation. *Eff.*≥ 1 implies that the algorithm finds a TN structure identical or more compact than the one used in data generation.

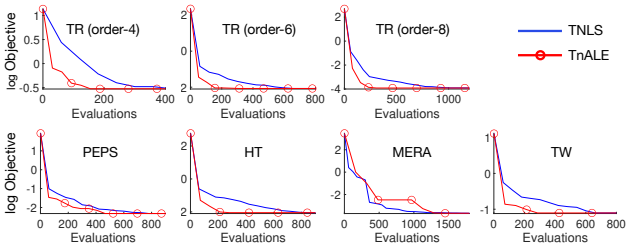


Figure 4. Averaged objective (in the log form) with varying the number of evaluations.

values (in log, averaged) versus the number of evaluations in TNLS and TnALE. The result confirms the consistency of the evaluation advantage of TnALE compared with TNLS[5].

Next, we evaluate the performance of TnALE for solving the classic rank selection problem, *i.e.*, TN-RS, within the TR decomposition task. To be specific, we randomly generate synthetic TR-tensors of order 8, and consider two configurations: "*lower-ranks*" and "*higher-ranks*". In the "lower-ranks" class, the TN-ranks are randomly chosen in the interval $[1, 4]$, while in the "higher-ranks" class the selection interval is lifted to $[5, 8]$, so that the ranks would be larger than the tensors' mode dimension (which equals 3 in this experiment). This situation commonly happens in practice for high-order TNs. For comparison, we implement various rank-adaptive TR decomposition methods, including TR-SVD and TR-BALS (Zhao et al., 2016), TR-LM (Mickelin & Karaman, 2020), and TRAR (Sedighin et al., 2021). In addition, the TTOpt algorithm (Sozykin et al., 2022) with ranks [6], denoted R, equaling $\{1, 2\}$ is also employed as a baseline.

The experimental results are shown in Table 2. We see that most of the methods can successfully identify the TN-ranks (implied by *Eff.* ≥ 1) in the "lower-ranks" class, but in the "higher-ranks" class only the methods at the bottom of the table manage to find the correct ranks. Furthermore, TnALE costs the fewest evaluations on average compared with TNGA, TNLS and TTOpt.

5.2. Real-World Data

We apply now the proposed method to real-world data to compress the learnable parameters of the tensorial Gaussian process (TGP, Izmailov et al. 2018) and to compress natural images. In TGP compression, we consider the regression task by TGPs for three datasets, including CCPP (Tüfekci, 2014), MG (Flake & Lawrence, 2002), and Protein (Dua & Graff, 2017), and compress the variational mean of the process with the TT/TR decomposition using the same settings as in Li et al. (2022). The goal of the experiment is to search for good TN-ranks and permutations, so that fewer parameters can be used to achieve the same mean square error (MSE) in regression. The experimental results are shown in Table 3. We can see that TnALE achieves the same compression ratio as TNGA and TNLS but costs *significantly* fewer evaluations than the counterparts in factor up to 14 $(3901/276)$.

Last, we consider the TN-PS and TN-TS tasks for compressing natural images. In TN-TS, we search for good TN-ranks and topologies for image compression. In the experiment, we randomly select four images (**A, B, C, D**, see Figure 5) from the dataset BSD500 (Arbelaez et al., 2010). Each image is resized by 256×256 and then reshaped into an order-8 tensor. For comparison, we also implement the "Greedy" method (Hashemizadeh et al., 2020) in the TN-TS task. Table 4 shows the results, including the compression ratio, RSE, and the number of evaluations in both tasks. We see that TnALE achieves the closest compression ratio and RSE to TNGA and TNLS, but it requires much fewer evaluations.

Table 2. Experimental results of TN-RS (rank-selection) in 8th-order TR topology. The columns of "lower-ranks" and "higher-ranks" indicate two experimental settings by which the TN-ranks are randomly selected. The *Eff.* and [*#Eva.*] values are averaged in five tensors.

Methods	**lower-ranks**	**higher-ranks**
	Eff.↑	*Eff.*↑
TR-SVD	0.65±0.46	0.13±0.20
TR-BALS	1.15±0.14	0.19±0.22
TR-LM	1.15±0.14	0.15±0.02
TRAR	0.55±0.10	0.63±0.06
	Eff.↑ *[#Eva.*↓*]*	*Eff.*↑ *[#Eva.*↓*]*
TNGA	1.08±0.06 [552]	1.00±0.00 [900]
TNLS	1.08±0.06 [492]	1.00±0.00 [588]
TTOpt ($R = 1$)	1.08±0.06 [104]	1.00±0.00 [178]
TTOpt ($R = 2$)	1.02±0.02 [314]	1.00±0.00 [752]
Ours	1.08±0.06 [**80**]	1.00±0.00 [**119**]

Table 3. Number of parameters ($\times 1000$, ↓) and MSE (in round brackets) for TGP model compression, where the values in [square brackets] show the number of evaluations required in each method.

	CCPP	**MG**	**Protein**
TGP	2.64 (0.06) [N/A]	3.36 (0.33) [N/A]	2.88 (0.74) [N/A]
TNGA	**2.24** (0.06) [1500]	**3.01** (0.33) [4900]	2.03 (0.74) [3900]
TNLS	**2.24** (0.06) [1051]	**3.01** (0.33) [3901]	**1.88** (0.74) [3601]
Ours	**2.24** (0.06) [**124**]	**3.01** (0.33) [**276**]	**1.88** (0.74) [**1053**]

[5]Note that the curves for MERA exhibit the opposite pattern compared to others due to the "local-convergence" issue. This phenomenon is further discussed in Appendix C.

[6]Here the ranks are the tuning parameters in the TTOpt algorithm, rather than the targeted TN structure.

Figure 5. Four images in the compression experiment.

Table 4. Results for natural image compression. The underlined values show the best compression ratio achieved in the same RSE.

Tasks	Methods	**Data** - compression ratio ($\log$, ↑) (RSE ↓) [*#Eva.* ↓]			
		A	**B**	**C**	**D**
TN-PS	**TNGA**	1.10 (0.15) [8400]	1.37 (0.17) [6300]	1.77 (0.08) [4800]	1.47 (0.10) [5100]
	TNLS	1.09 (0.16) [1351]	<u>1.41</u> (0.17) [1501]	1.71 (0.08) [2551]	1.47 (0.10) [2101]
	Ours	1.14 (0.16) [**647**]	1.39 (0.17) [**666**]	<u>1.80</u> (0.08) [**394**]	<u>1.49</u> (0.10) [**444**]
TN-TS	**Greedy**	0.81 (0.16)	0.97 (0.17)	1.44 (0.08)	0.68 (0.10)
	TNGA	1.16 (0.16) [2100]	<u>1.48</u> (0.17) [1800]	<u>1.81</u> (0.08) [1900]	1.48 (0.09) [1000]
	TNLS	1.15 (0.16) [1300]	1.40 (0.17) [1100]	1.80 (0.08) [1700]	1.50 (0.10) [1700]
	Ours	1.10 (0.15) [**177**]	1.46 (0.17) [**153**]	<u>1.81</u> (0.08) [**237**]	1.51 (0.10) [**246**]

6. Concluding Remarks

Extensive experimental results demonstrate that the proposed TnALE approach can *greatly* reduce the evaluation cost, up to $10\times$ fewer evaluations, compared with TNLS (Li et al., 2022) and other methods for the task of tensor network structure search (TN-SS). The theoretical results in this paper provide a rigorous analysis of the convergence rate and the evaluation efficiency for both TNLS and TnALE.

Limitation. The main limitation of TnALE is the local convergence issue. In particular, we empirically found in the TN-TS experiment (see Table 4) multiple local minima, which are poor in compression ratio, and TnALE can easily drop in them. Conversely, the methods TNGA (Li & Sun, 2020) and TNLS (Li et al., 2022) seem to better avoid local minima, owing to their stochastic essence. Solving this issue will be the direction of our future work. Furthermore, the identifiability of the proposed method for TN-SS in the presence of *noise* would be also investigated in the future.

Acknowledgements

This work was partially supported by JSPS KAKENHI (Grant No. 20H04249, 23H03419). Chunmei is supported by China Scholarship Council (CSC). Part of the computation was carried out at the Riken AIp Deep learning ENvironment (RAIDEN).

References

Acharya, A., Rudolph, M., Chen, J., Miller, J., and Perdomo-Ortiz, A. Qubit seriation: Improving data-model alignment using spectral ordering. *arXiv preprint arXiv:2211.15978*, 2022.

Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., and Telgarsky, M. Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research*, 15(1):2773–2832, 2014.

Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010.

Boyd, S. and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

Cai, Y. and Li, P. A blind block term decomposition of high order tensors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, number 8, pp. 6868–6876, 2021.

Chen, Z., Lu, J., and Zhang, A. R. One-dimensional tensor network recovery. *arXiv preprint arXiv:2207.10665*, 2022.

Cheng, Z., Li, B., Fan, Y., and Bao, Y. A novel rank selection scheme in tensor ring decomposition based on reinforcement learning for deep neural networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3292–3296. IEEE, 2020.

Cichocki, A., Phan, A.-H., Zhao, Q., Lee, N., Oseledets, I., Sugiyama, M., Mandic, D. P., et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 2 applications and future perspectives. *Foundations and Trends® in Machine Learning*, 9(6):431–673, 2017.

Cincio, L., Dziarmaga, J., and Rams, M. M. Multiscale entanglement renormalization ansatz in two dimensions: quantum Ising model. *Physical Review Letters*, 100(24): 240603, 2008.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Falcó, A., Hackbusch, W., and Nouy, A. Geometry of tree-based tensor formats in tensor banach spaces. *Annali di Matematica Pura ed Applicata (1923-)*, pp. 1–18, 2023.

Fawzi, A., Balog, M., Huang, A., Hubert, T., Romera-Paredes, B., Barekatain, M., Novikov, A., Ruiz, F. J. R., Schrittwieser, J., Swirszcz, G., Silver, D., Hassabis, D., and Kohli, P. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930): 47–53, 2022.

Felser, T., Trenti, M., Sestini, L., Gianelle, A., Zuliani, D., Lucchesi, D., and Montangero, S. Quantum-inspired machine learning on high-energy physics data. *npj Quantum Information*, 7(1):111, 2021.

Flake, G. W. and Lawrence, S. Efficient SVM regression training with SMO. *Machine Learning*, 46(1):271–290, 2002.

Ghadiri, M., Fahrbach, M., Fu, G., and Mirrokni, V. Approximately optimal core shapes for tensor decompositions. *arXiv preprint arXiv:2302.03886*, 2023.

Glasser, I., Sweke, R., Pancotti, N., Eisert, J., and Cirac, I. Expressive power of tensor-network factorizations for probabilistic modeling. *Advances in neural information processing systems*, 32, 2019.

Golovin, D., Karro, J., Kochanski, G., Lee, C., Song, X., and Zhang, Q. Gradientless descent: High-dimensional zeroth-order optimization. In *International Conference on Learning Representations*, 2019.

Haberstich, C., Nouy, A., and Perrin, G. Active learning of tree tensor networks using optimal least-squares. *arXiv preprint arXiv:2104.13436*, 2021.

Hackbusch, W. and Kühn, S. A new scheme for the tensor representation. *Journal of Fourier analysis and applications*, 15(5):706–722, 2009.

Hashemizadeh, M., Liu, M., Miller, J., and Rabusseau, G. Adaptive tensor learning with tensor networks. *arXiv preprint arXiv:2008.05437*, 2020.

Hawkins, C. and Zhang, Z. Bayesian tensorized neural networks with automatic rank selection. *Neurocomputing*, 453:172–180, 2021.

Hayashi, K., Yamaguchi, T., Sugawara, Y., and Maeda, S.-i. Exploring unexplored tensor network decompositions for convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 5553–5563, 2019.

Hikihara, T., Ueda, H., Okunishi, K., Harada, K., and Nishino, T. Automatic structural optimization of tree tensor networks. *Physical Review Research*, 5(1):013031, 2023.

Hillar, C. J. and Lim, L.-H. Most tensor problems are NP-hard. *Journal of the ACM (JACM)*, 60(6):45, 2013.

Izmailov, P., Novikov, A., and Kropotov, D. Scalable Gaussian processes with billions of inducing inputs via tensor train decomposition. In *International Conference on Artificial Intelligence and Statistics*, pp. 726–735. PMLR, 2018.

Khavari, B. and Rabusseau, G. Lower and upper bounds on the pseudo-dimension of tensor network models. *Advances in Neural Information Processing Systems*, 34, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kodryan, M., Kropotov, D., and Vetrov, D. Mars: Masked automatic ranks selection in tensor decompositions. In *International Conference on Artificial Intelligence and Statistics*, pp. 3718–3732. PMLR, 2023.

Kossaifi, J., Lipton, Z. C., Kolbeinsson, A., Khanna, A., Furlanello, T., and Anandkumar, A. Tensor regression networks. *Journal of Machine Learning Research*, 21: 1–21, 2020.

Li, C. and Sun, Z. Evolutionary topology search for tensor network decomposition. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

Li, C. and Zhao, Q. Is rank minimization of the essence to learn tensor network structure? In *Second Workshop on Quantum Tensor Networks in Machine Learning (QT-NML), Neurips*, 2021.

Li, C., Zeng, J., Tao, Z., and Zhao, Q. Permutation search of tensor network structures via local sampling. In *International Conference on Machine Learning*, pp. 13106–13124. PMLR, 2022.

Li, N., Pan, Y., Chen, Y., Ding, Z., Zhao, D., and Xu, Z. Heuristic rank selection with progressively searching tensor ring network. *Complex & Intelligent Systems*, pp. 1–15, 2021.

Liu, Y., Lu, Y., Ou, W., Long, Z., and Zhu, C. Adaptively topological tensor network for multi-view subspace clustering. *arXiv preprint arXiv:2305.00716*, 2023.

Long, Z., Zhu, C., Liu, J., and Liu, Y. Bayesian low rank tensor ring for image recovery. *IEEE Transactions on Image Processing*, 30:3568–3580, 2021.

Malik, O. A. More efficient sampling for tensor decomposition with worst-case guarantees. In *International Conference on Machine Learning*, pp. 14887–14917. PMLR, 2022.

Mickelin, O. and Karaman, S. On algorithms for and computing with the tensor ring decomposition. *Numerical Linear Algebra with Applications*, 27(3):e2289, 2020.

Miller, J., Rabusseau, G., and Terilla, J. Tensor networks for probabilistic sequence modeling. In *International Conference on Artificial Intelligence and Statistics*, pp. 3079–3087. PMLR, 2021.

Nie, C., Wang, H., and Tian, L. Adaptive tensor networks decomposition. In *BMVC*, 2021.

Novikov, A., Podoprikhin, D., Osokin, A., and Vetrov, D. P. Tensorizing neural networks. In *Advances in Neural Information Processing Systems*, pp. 442–450, 2015.

Olver, P. J. *Introduction to partial differential equations*. Springer, 2014.

Orús, R. Tensor networks for complex quantum systems. *Nature Reviews Physics*, 1(9):538–550, 2019.

Oseledets, I. and Tyrtyshnikov, E. TT-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*, 432(1):70–88, 2010.

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

Osinsky, A. Tensor trains approximation estimates in the chebyshev norm. *Computational Mathematics and Mathematical Physics*, 59(2):201–206, 2019.

Rai, P., Wang, Y., Guo, S., Chen, G., Dunson, D., and Carin, L. Scalable bayesian low-rank decomposition of incomplete multiway tensors. In *International Conference on Machine Learning*, pp. 1800–1808. PMLR, 2014.

Richter, L., Sallandt, L., and Nüsken, N. Solving high-dimensional parabolic PDEs using the tensor train format. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, pp. 8998–9009. PMLR, 2021.

Sedighin, F., Cichocki, A., and Phan, A.-H. Adaptive rank selection for tensor ring decomposition. *IEEE Journal of Selected Topics in Signal Processing*, 15(3):454–463, 2021.

Snyder, L. V. and Daskin, M. S. A random-key genetic algorithm for the generalized traveling salesman problem. *European journal of operational research*, 174(1):38–53, 2006.

Solgi, R., Loaiciga, H. A., and Zhang, Z. Evolutionary tensor train decomposition for hyper-spectral remote sensing images. In *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*, pp. 1145–1148. IEEE, 2022.

Sozykin, K., Chertkov, A., Schutski, R., Phan, A.-H., CICHOCKI, A. S., and Oseledets, I. Ttopt: A maximum volume quantized tensor train-based optimization and its application to reinforcement learning. *Advances in Neural Information Processing Systems*, 35:26052–26065, 2022.

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

Tüfekci, P. Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. *International Journal of Electrical Power & Energy Systems*, 60:126–140, 2014.

Tyrtyshnikov, E. Incomplete cross approximation in the mosaic-skeleton method. *Computing*, 64(4):367–380, 2000.

Verstraete, F. and Cirac, J. I. Renormalization algorithms for quantum-many body systems in two and higher dimensions. *arXiv preprint cond-mat/0407066*, 2004.

Wang, W., Aggarwal, V., and Aeron, S. Efficient low rank tensor ring completion. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5697–5705, 2017.

Wu, Z.-C., Huang, T.-Z., Deng, L.-J., Dou, H.-X., and Meng, D. Tensor wheel decomposition and its tensor completion application. In *Advances in Neural Information Processing Systems*, 2022.

Ye, K. and Lim, L.-H. Tensor network ranks. *arXiv preprint arXiv:1801.02662*, 2019.

Yin, M., Phan, H., Zang, X., Liao, S., and Yuan, B. Batude: Budget-aware neural network compression based on Tucker decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 1, 2022.

Yokota, T., Zhao, Q., and Cichocki, A. Smooth PARAFAC decomposition for tensor completion. *IEEE Transactions on Signal Processing*, 64(20):5423–5436, 2016.

Zhang, A. Cross: Efficient low-rank tensor completion. *The Annals of Statistics*, 47(2):936–964, 2019.

Zhao, Q., Zhang, L., and Cichocki, A. Bayesian CP factorization of incomplete tensors with automatic rank determination. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1751–1763, 2015.

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

Zhe, S., Xu, Z., Chu, X., Qi, Y., and Park, Y. Scalable nonparametric multiway data analysis. In *Artificial Intelligence and Statistics*, pp. 1125–1134. PMLR, 2015.

Zheng, Y.-B., Huang, T.-Z., Zhao, X.-L., Zhao, Q., and Jiang, T.-X. Fully-connected tensor network decomposition and its application to higher-order tensor completion. In *Proc. AAAI*, volume 35, pp. 11071–11078, 2021.

A. TnALE: Details for the algorithm

The pseudocode for ALE is given in Alg. 3. As discussed in the paper, each structure-related variable is updated alternately by enumeration in the neighborhood. Based on it, the entire algorithm of TnALE is shown in Alg. 2. Apart from the major iteration (lines 6-8), beforehand, there is an initial phase, where we apply a larger radius r_1 and the objective prediction trick to find the candidates in a broader range and a rough fashion.

A.1. The objective estimation trick.

As discussed in Remark 3.1, we employ the *objective estimation* by linear interpolation in place of the complete enumeration when updating the TN-ranks. Particularly in Alg. 2, we apply this trick to the initial phase, where we consider a broader range of the neighborhood for roughly finding the structure candidates.

Suppose a rank variable r is required to be enumerated in the range of $r_0 - b \leq r \leq r_0 + b$, where $r_0, b \in \mathbb{Z}_+$ present the center and radius of the searched neighborhood, respectively. The objective estimation trick aims to estimate the minimum of the inner optimization of (1) associated with each enumerated TN-structures with limited evaluations. In the trick, we first evaluate explicitly three TN structures, *i.e.*, $r = \{r_0 - b, r_0, r_0 + b\}$. Then, a simple linear regression model is applied to predicting the evaluations of TN-structures in the interval $(r_0, r_0 + b)$. The evaluations for the interval $(r_0 - b, r_0)$ will be predicted similarly using the data *w.r.t.* $\{r_0 - b, r_0\}$. We can see from the trick that the inner minimization of (1) can be quickly estimated with only three explicit evaluations, no matter how wide the searching range is. Although the simple linear regression can only give a rough estimation, it is sufficiently helpful for TnALE to find good TN-structures in the initial phase.

A.2. The neighborhood in the graph space

In the problem of TN-PS, we follow the idea of Li et al. (2022) to specify the neighborhood for a given graph G in (1). Similar to Alg. 1 in Li et al. (2022), we construct the neighborhood by swapping enumerately two vertices of G. Suppose the graph G is of N vertices, we consequently achieve the neighborhood of G of the size $N(N-1)/2$.

Algorithm 2 TnALE: the proposed solver of the optimization (1)

1: **Input:** A solver for the inner minimization of (1); the rank-related radius: $r_1 \geq r_2 > 0$; the number of Iterations in the initialization phase: L_0; the number of Iterations in the searching phase: L; the number of the round-trips for ALE: D;
2: **Initialize:** Uniformly choose a TN structure $(G, \mathbf{r})$ at random or choose $(G, \mathbf{r})$ with the specified value;
3: **for** $l = 1, \ldots L_0$ **do**
4: Update recursively $(G, \mathbf{r})$ using Alg. 3 with the center $(G, \mathbf{r})$, radius r_1, D round-trips and the objective estimation trick;
5: **end for**
6: **for** $l = 1, \ldots L$ **do**
7: Update recursively $(G, \mathbf{r})$ using Alg. 3 with the center $(G, \mathbf{r})$, radius r_2 and D round-trips ;
8: **end for**
9: **Output:** $(G, \mathbf{r})$.

Algorithm 3 ALE: alternating local enumeration

Input: The center of the neighborhood: $(G^{(0)}, \mathbf{r}^{(0)})$, where $\mathbf{r} = [r_1^{(0)}, r_2^{(0)}, \ldots, r_K^{(0)}]^\top \in \mathbb{Z}_+^K$; the rank-related radius: $r \in \mathbb{Z}_+$; the number of "round-trips": D;
Initialize: $(G, \mathbf{r}) = (G_0, \mathbf{r}_0)$, where $\mathbf{r} = [r_1, r_2, \ldots, r_K]^\top$
for $d = 1, \ldots, D$ **do**
  for $k = 1, \ldots, K$ **do**
    for $i = -r, \ldots, 0, \ldots, r$ **do**
      Copy $(G, \mathbf{r})$ into $(\bar{G}, \bar{\mathbf{r}})$
      Update $(\bar{G}, \bar{\mathbf{r}})$ by $\bar{r}_k = r_k + i$;
      Calculate the objective of (1) associated to $(\bar{G}, \bar{\mathbf{r}})$; # `Objective estimation is available.`
      Store the value of the objective as $h(i)$;
    end for
    Update $(G, \mathbf{r})$ by $r_k = \arg\min_i h(i)$;
  end for
  Take the neighborhood $B(G)$ according to section A.2;
  for all $G' \in B(G)$ **do**
    Update $(G, \mathbf{r})$ by $G = G'$;
    Calculate the objective of (1) associated to $(G, \mathbf{r})$;
    Store the value of the objective as $h(G')$;
  end for
  Update $(G, \mathbf{r})$ by $G = \arg\min_{G'} h(G')$;
  for $k = K, K-1, \ldots, 2$ **do**
    for $i = -r, \ldots, 0, \ldots, r$ **do**
      Copy $(G, \mathbf{r})$ into $(\bar{G}, \bar{\mathbf{r}})$
      Update $(\bar{G}, \bar{\mathbf{r}})$ by $\bar{r}_k = r_k + i$;
      Calculate the objective of (1) associated to $(\bar{G}, \bar{\mathbf{r}})$; # `Objective estimation is available.`
      Store the value of the objective as $h(i)$;
    end for
    Update $(G, \mathbf{r})$ by $r_k = \arg\min_i h(i)$;
  end for
end for
Output: $(G, \mathbf{r})$.

B. Theoretical analysis with proofs

In this section, we first give a rigorous convergence analysis for the algorithms using the local-sampling scheme. After that, we compare the evaluation efficiency for TNLS (Li et al., 2022) and the new algorithm TnALE.

B.1. A quick review of tensor network (TN) structure search

Suppose we have the dataset D and a task-specific loss function $\pi_D : \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \to \mathbb{R}_+$ associated to D. The tensor network structure search (TN-SS) problem is to solve the following bi-level optimization problem (Li et al., 2022)

$$\min_{(G,\mathbf{r}) \in \mathbb{G} \times \mathbb{F}_G} \left(\phi(G, \mathbf{r}) + \lambda \cdot \min_{\mathcal{Z} \in TNS(G,\mathbf{r})} \pi_D(\mathcal{Z}) \right), \tag{8}$$

where $G \in \mathbb{G}$ is a graph, which owns N vertices and K edges and corresponds to the TN-topology, $\mathbf{r} \in \mathbb{F}_N \subseteq \mathbb{Z}_+^K$ is a positive integer vector of K dimension corresponding to the TN-ranks, $\phi : \mathbb{G} \times \mathbb{Z}_+^K \to \mathbb{R}_+$ represents the function measuring the model complexity of a TN whose structure is modeled by $(G, \mathbf{r})$, and $\lambda > 0$ is a tuning parameter. As expected for TN-SS, solving the problem (8) is intuitively to search for a TN structure modeled by $(G, \mathbf{r})$, by which we can give the optimal balance between the complexity and the expressibility of a TN in the task.

We remark that TN-SS can be specified as three sub-problems: *permutation selection (TN-PS, Li et al. (2022)), rank selection (TN-RS) and topology selection (TN-TS, Li & Sun (2020))*, by specifying the feasible set $\mathbb{G} \times \mathbb{F}_G$ of (8) into different forms. Specifically, in TN-PS, we set $\mathbb{F}_G = \mathbb{Z}_+^K$ and $\mathbb{G}$ is defined as the isomorphic graphs to a "template" G_0, so that only the ranks and vertex permutations are searched while the TN-topology is preserved. In TN-RS, however, we typically restrict the graph G in (8), *i.e.*, $\mathbb{G} = \{G_0\}$ but consider searching for all possible ranks *i.e.*, $\mathbb{F}_G = \mathbb{Z}_+^K$. In TN-TS, we relax $\mathbb{G}$ to be a set containing all possible simple graphs of order N and the set $\mathbb{F}_N$ can be fixed (Li & Sun, 2020) or not (Hashemizadeh et al., 2020). It is known from (Ye & Lim, 2019; Hashemizadeh et al., 2020) that the TN-PS problem with the rank searching can be simplified as a TN-RS problem associated to a "fully-connected" TN (Zheng et al., 2021).

B.2. Analysis of descent steps

We start the analysis by rewriting (8) into a more general form:

$$\min_{\mathbf{x} \in \mathbb{Z}_+^K, p \in \mathbb{P}} f_p(\mathbf{x}) := f \circ p(\mathbf{x}), \tag{9}$$

where $\circ$ denotes the function composition, $f : \mathbb{Z}_{\geq 0}^L \to \mathbb{R}_+$ is a generalization of the objective function of (8), $\mathbf{x} \in \mathbb{Z}_+^K$ corresponds to the rank-related variable $\mathbf{r}$ of (8), and the operator $p : \mathbb{Z}_+^K \to \mathbb{Z}_{\geq 0}^L$ and its feasible set $\mathbb{P}$ correspond to the topology-related variable G and the set $\mathbb{G}$ of (8), respectively.

The relationship between $p \in \mathbb{P}$ of (9) and $G \in \mathbb{G}$ of (8) is demonstrated as follows. Since in (8) the entries of $\mathbf{r}$ can be regarded as labels on the edges of G, the pair $(G, \mathbf{r})$ can be described as a weighted adjacency matrix of $N \times N$. For example, a 4-th order tensor ring (TR, Zhao et al., 2016) of the ranks-$\{2, 3, 4, 5\}$ can be described as

$$(G_1, \mathbf{r}) = \left(\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} \right) \implies \mathbf{A}_1 = \begin{pmatrix} 0 & 0 & 2 & 3 \\ 0 & 0 & 4 & 5 \\ 2 & 3 & 0 & 0 \\ 4 & 5 & 0 & 0 \end{pmatrix}, \tag{10}$$

or

$$(G_2, \mathbf{r}) = \left(\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} \right) \implies \mathbf{A}_2 = \begin{pmatrix} 0 & 2 & 0 & 5 \\ 2 & 0 & 3 & 0 \\ 0 & 3 & 0 & 4 \\ 5 & 0 & 4 & 0 \end{pmatrix}. \tag{11}$$

Here G_1 and G_2 correspond to the TR topology with different permutations of vertices. In the settings of TN-PS (Li et al., 2022), we can prove that such the relationship is bijective. The operator p is thus to map the TN-ranks, denoted $\mathbf{x} \in \mathbb{Z}_+^K$ in (9), onto the vectorization of entries in the upper triangle part (except for the diagonal) of the adjacency matrix. For

example,

$$\mathbf{A}_1 = \begin{pmatrix} 0 & 0 & 2 & 3 \\ 0 & 0 & 4 & 5 \\ 2 & 3 & 0 & 0 \\ 4 & 5 & 0 & 0 \end{pmatrix} \iff p_1(\mathbf{x}) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ 3 \\ 4 \\ 5 \\ 0 \end{pmatrix}, \tag{12}$$

and

$$\mathbf{A}_2 = \begin{pmatrix} 0 & 2 & 0 & 5 \\ 2 & 0 & 3 & 0 \\ 0 & 3 & 0 & 4 \\ 5 & 0 & 4 & 0 \end{pmatrix} \iff p_2(\mathbf{x}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 5 \\ 3 \\ 0 \\ 4 \end{pmatrix}. \tag{13}$$

It is shown that p is essentially an operator produced by the permutation padding with several rows of zeros according to G.

The convergence analysis of this work is mainly inspired by Golovin et al. (2019), which establishes a convex framework for the gradient-less optimization algorithms in the real domain. The challenge is, the TN-SS problem is essentially discrete, so that many well-developed tools, such as convexity and smoothness, for convergence analysis turn invalid in the discrete scenario.

To bridge the graph from Golovin et al. (2019) to TN-SS, we first re-define several important concepts, by which the necessary tools for the analysis are re-derived. In doing so, we begin by introducing the finite gradient as the alternative to the classic one defined in the continuous domain.

Definition B.1 (finite gradient). For any function $f : \mathbb{Z}_{\geq 0}^L \to \mathbb{R}$, its finite gradient $\Delta f : \mathbb{Z}_{\geq 0}^L \to \mathbb{R}^L$ at the point $\mathbf{x}$ is defined as the vector

$$\Delta f(\mathbf{x}) = \left[f(\mathbf{x} + \mathbf{e}_1) - f(\mathbf{x}), \ldots, f(\mathbf{x} + \mathbf{e}_L) - f(\mathbf{x}) \right]^\top, \tag{14}$$

where $\mathbf{e}_i \, \forall i \in [L]$ denote the unit vectors with the i-th entry being one and other entries being zeros.

Applying the finite gradient defined in (14), we also re-define the strong convexity and smoothness for analysis in the discrete domain.

Definition B.2 (α-strong convexity with finite gradient). We say f is α-strongly convex for $\alpha \geq 0$ if $f(\mathbf{y}) \geq f(\mathbf{x}) + \left\langle \Delta f(\mathbf{x}) - \frac{\alpha}{2}\mathbf{1}, \mathbf{y} - \mathbf{x} \right\rangle + \frac{\alpha}{2}\|\mathbf{y} - \mathbf{x}\|^2$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_{\geq 0}^L$, where $\mathbf{1} \in \mathbb{R}^L$ denotes the vector with all entries being one. We simply say f is convex if it is α-strongly convex and $\alpha = 0$.

Compared to the definitions used in Golovin et al. (2019) and other literature for convex analysis, the additional term, $\frac{\alpha}{2}\mathbf{1}$, marked by the *blue* color, appears due to the discrepancy of the finite gradient and its counterpart in the continuous domain. Below, we prove several basic results using the α-strong convexity with finite gradient.

Lemma B.3. *If f is α-strongly convex in $\mathbb{Z}_{\geq 0}^L$, then the following inequalities are held:*

1. *$g(\mathbf{x}) = f(\mathbf{x}) - \frac{\alpha}{2}\|\mathbf{x}\|^2$ is convex in the discrete scenario for all $\mathbf{x} \in \mathbb{Z}_{\geq 0}^L$, and vice versa;*
2. *$\langle \Delta f(\mathbf{x}) - \Delta f(\mathbf{x}), \mathbf{x} - \mathbf{y} \rangle \geq \alpha \|\mathbf{x} - \mathbf{y}\|^2$ for any $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_{\geq 0}^L$;*
3. *$\|\Delta f(\mathbf{x}) - \Delta f(\mathbf{y})\| \geq \alpha \|\mathbf{x} - \mathbf{y}\|$ for any $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_{\geq 0}^L$;*

Here $\| \cdot \|$ denotes the l_2 norm for vectors.

Proof. $(1, \Rightarrow)$ According to Def. B.2, the first statement is equivalent to proving the inequality

$$g(\mathbf{y}) \geq g(\mathbf{x}) + \langle \Delta g(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle \tag{15}$$

holding for any $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_{\geq 0}^L$. Applying the α-strong convexity assumption, it follows that

$$\begin{aligned} g(\mathbf{y}) - g(\mathbf{x}) - \langle \Delta g(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle &= f(\mathbf{y}) - \frac{\alpha}{2}\|\mathbf{y}\|^2 - f(\mathbf{x}) + \frac{\alpha}{2}\|\mathbf{x}\|^2 - \langle \Delta g(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle \\ &= f(\mathbf{y}) - \frac{\alpha}{2}\|\mathbf{y}\|^2 - f(\mathbf{x}) + \frac{\alpha}{2}\|\mathbf{x}\|^2 - \left\langle \Delta f(\mathbf{x}) - \frac{\alpha}{2}(2\mathbf{x} + \mathbf{1}), \mathbf{y} - \mathbf{x} \right\rangle \\ &= f(\mathbf{y}) - f(\mathbf{x}) - \left\langle \Delta f(\mathbf{x}) - \frac{\alpha}{2}\mathbf{1}, \mathbf{y} - \mathbf{x} \right\rangle - \frac{\alpha}{2}\|\mathbf{y}\|^2 + \frac{\alpha}{2}\|\mathbf{x}\|^2 + \frac{\alpha}{2}\langle 2\mathbf{x}, \mathbf{y} - \mathbf{x} \rangle \\ &\geq \frac{\alpha}{2}\left(\|\mathbf{y} - \mathbf{x}\|^2 - \|\mathbf{y}\|^2 - \|\mathbf{x}\|^2 + 2\langle \mathbf{x}, \mathbf{y} \rangle\right) = 0. \end{aligned} \tag{16}$$

Here the first equality follows from the definition of $g(\mathbf{x})$, the second equality holds since the finite gradient $\Delta\|\mathbf{x}\|^2 = 2\mathbf{x} + \mathbf{1}$, and the inequality at the bottom line follows from the α-strong convexity assumption on f.

(1, $\Leftarrow$) By (15),

$$f(\mathbf{y}) - \frac{\alpha}{2}\|\mathbf{y}\|^2 \geq f(\mathbf{x}) - \frac{\alpha}{2}\|\mathbf{x}\|^2 + \left\langle \Delta f(\mathbf{x}) - \frac{\alpha}{2}(2\mathbf{x} + 1), \mathbf{y} - \mathbf{x} \right\rangle. \tag{17}$$

The α-strong convexity of f is thus proved by algebraically simplifying (17).

(2) To prove the second statement, we first know that the following inequality

$$\langle \Delta g(\mathbf{x}) - \Delta g(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \geq 0, \ \forall \mathbf{x}, \mathbf{y} \tag{18}$$

holds since the monotone gradient property of the convexity (it is true in both continuous and discrete scenarios). By the form of $\Delta g(\mathbf{x})$, it follows that

$$\left\langle \Delta f(\mathbf{x}) - \frac{\alpha}{2}(2\mathbf{x} + \mathbf{1}) - \Delta f(\mathbf{y}) + \frac{\alpha}{2}(2\mathbf{y} + \mathbf{1}), \mathbf{x} - \mathbf{y} \right\rangle \geq 0. \tag{19}$$

With algebraic simplification, we obtain

$$\langle \Delta f(\mathbf{x}) - \Delta f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \geq \alpha\|\mathbf{x} - \mathbf{y}\|^2 \tag{20}$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_{\geq 0}^L$. The second statement is thus proved.

(3) The third statement holds since the following relationship

$$\|\Delta f(\mathbf{x}) - \Delta f(\mathbf{y})\|\|\mathbf{x} - \mathbf{y}\| \geq \langle \Delta f(\mathbf{x}) - \Delta f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \geq \alpha\|\mathbf{x} - \mathbf{y}\|^2, \tag{21}$$

where the first inequality follows from the Cauchy–Schwarz inequality, and the second inequality follows from (20). The third statement is proved by dividing by $\|\mathbf{x} - \mathbf{y}\|$ on both sides. □

Apart from the convexity, the smoothness of the objective function is also required to be re-defined in the discrete scenario.

Definition B.4 (**(β_1, β_2)-smoothness with finite gradient**). We say f is (β_1, β_2)-smooth for $\beta_1, \beta_2 > 0$ if

1. $|f(\mathbf{x}) - f(\mathbf{y})| \leq \beta_1\|\mathbf{x} - \mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_{\geq 0}^L$;
2. The function $l(\mathbf{x}) := \frac{\beta_2}{2}\|\mathbf{x}\|^2 - f(\mathbf{x})$ is convex.

The first item of Def. B.4 restricts that f is β_1-Lipschitz, implying the "continuity" of the function, while the second item upper bounds the change of the finite gradient of f, implying a "continuity" over the finite gradient. In particular,

Lemma B.5. *If $l(\mathbf{x}) = \frac{\beta}{2}\|\mathbf{x}\|^2 - f(\mathbf{x})$ is convex, then for all $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_{\geq 0}^L$*

1. $f(\mathbf{y}) \leq f(\mathbf{x}) + \left\langle \Delta f(\mathbf{x}) - \frac{\beta}{2}\mathbf{1}, \mathbf{y} - \mathbf{x} \right\rangle + \frac{\beta}{2}\|\mathbf{y} - \mathbf{x}\|^2$ *and vise versa;*
2. $\langle \Delta f(\mathbf{x}) - \Delta f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \leq \beta\|\mathbf{x} - \mathbf{y}\|^2$.

Proof. $(1, \Rightarrow)$ By the form $l(\mathbf{x})$ and its convex property, we have the inequality

$$\frac{\beta}{2}\|\mathbf{y}\|^2 - f(\mathbf{y}) \geq \frac{\beta}{2}\|\mathbf{x}\|^2 - f(\mathbf{x}) + \left\langle \frac{\beta}{2}(2\mathbf{x}+\mathbf{1}) - \Delta f(\mathbf{x}), \mathbf{y}-\mathbf{x} \right\rangle. \tag{22}$$

The first statement is proved by algebraically simplifying (22). The $(\Leftarrow)$ direction can be proved similarly.

To prove the second item, by the convexity of $l(\mathbf{x})$,

$$l(\mathbf{y}) \geq l(\mathbf{x}) + \langle \Delta l(\mathbf{x}), \mathbf{y}-\mathbf{x} \rangle. \tag{23}$$

Similarly,

$$l(\mathbf{x}) \geq l(\mathbf{y}) + \langle \Delta l(\mathbf{y}), \mathbf{x}-\mathbf{y} \rangle. \tag{24}$$

Summing the two sides of (23) and (24) up, we have

$$\langle \Delta l(\mathbf{x}) - \Delta l(\mathbf{y}), \mathbf{x}-\mathbf{y} \rangle \geq 0. \tag{25}$$

Applying $l(\mathbf{x}) = \frac{\beta}{2}\|\mathbf{x}\|^2 - f(\mathbf{x})$,

$$\langle \beta(2\mathbf{x}+\mathbf{1}) - \Delta f(\mathbf{x}) - \beta(2\mathbf{y}+\mathbf{1}) + \Delta f(\mathbf{y}), \mathbf{x}-\mathbf{y} \rangle \geq 0. \tag{26}$$

By simplifying the inequality, we finally have

$$\langle \Delta f(\mathbf{x}) - \Delta f(\mathbf{y}), \mathbf{x}-\mathbf{y} \rangle \leq \beta\|\mathbf{x}-\mathbf{y}\|^2. \tag{27}$$

□

The first item of Def. B.4 gives the following crucial result, which is used in the main theorem of this paper.

Lemma B.6. *If $|f(\mathbf{x}) - f(\mathbf{y})| \leq \beta\|\mathbf{x}-\mathbf{y}\|$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_{\geq 0}^L$, then the norm of the finite gradient with respective to $\mathbf{x}$ is bounded, i.e., $\|\Delta f(\mathbf{x})\|_\infty \leq \beta$.*

Proof. Denote $\Delta f(\mathbf{x})_i$ the i-th entry of $\Delta f(\mathbf{x})$, then for all $1 \leq i \leq L$ the second item of the definition follows by

$$|\Delta f(\mathbf{x})_i| = |f(\mathbf{x}+\mathbf{e}_i) - f(\mathbf{x})| \leq \beta\|\mathbf{x}+\mathbf{e}_i-\mathbf{x}\| = \beta, \tag{28}$$

where $\mathbf{e}_i$ denotes the unit vector with i-th entry being one and others being zeros, and the first equality follows from the definition of the finite gradient. □

After the new definitions of convexity and smoothness with finite gradient, in the proof, we also use the concept of the sub-level set, which is widely used in optimization theory. For the self-consistency purpose, the specific definition is reviewed as follows:

Definition B.7 (sub-level set). The level set of f at point $\mathbf{x} \in \mathbb{Z}_{\geq 0}^L$ is $\mathbb{L}_{\mathbf{x}}(f) = \left\{\mathbf{y} \in \mathbb{Z}_{\geq 0}^L : f(\mathbf{y}) = f(\mathbf{x})\right\}$. The sub-level set of f at point $\mathbf{x} \in \mathbb{Z}_{\geq 0}^L$ is $\mathbb{L}_{\mathbf{x}}^{\downarrow}(f) = \left\{\mathbf{y} \in \mathbb{Z}_{\geq 0}^L : f(\mathbf{y}) \leq f(\mathbf{x})\right\}$.

The following lemma shows that, for any $\mathbf{x}$, there exists a cube, *i.e.*, a ball with infinity-norm, which is tangent at $\mathbf{x}$ and inside the sub-level set $\mathbb{L}_{\mathbf{x}}^{\downarrow}(f)$.

Lemma B.8 (the sub-level cube). *Assume that $f : \mathbb{Z}_{\geq 0}^L \to \mathbb{R}$ is α-strongly convex, (β_1, β_2)-smooth, and its minimum, denoted $f(\mathbf{x}^*)$, satisfies $\|\frac{\beta_2}{2}\mathbf{1} - \Delta f(\mathbf{x}^*)\| \leq \gamma$ where γ is a constant and $0 \leq \gamma < \alpha$. Then, for all $\mathbf{x} \in \mathbb{Z}_{\geq 0}^L$, there is a L-dimensional cube, which is of the edge length $\frac{2(\alpha-\gamma)}{\beta_2\sqrt{L}}\|\mathbf{x}-\mathbf{x}^*\|$, tangent at $\mathbf{x}$, and inside the sub-level set $\mathbb{L}_{\mathbf{x}}^{\downarrow}(f)$.*

Proof. Applying the smoothness assumption and Lemma B.5,

$$\begin{aligned} f\left(\mathbf{x} - \frac{1}{\beta_2}\Delta f(\mathbf{x}) + \frac{1}{2}\mathbf{1} + \mathbf{s}\right) &\leq f(\mathbf{x}) + \left\langle \Delta f(\mathbf{x}) - \frac{\beta_2}{2}\mathbf{1}, \mathbf{s} + \frac{1}{2}\mathbf{1} - \frac{1}{\beta_2}\Delta f(\mathbf{x}) \right\rangle + \frac{\beta_2}{2}\left\|\mathbf{s} + \frac{1}{2}\mathbf{1} - \frac{1}{\beta_2}\Delta f(\mathbf{x})\right\|^2 \\ &= f(\mathbf{x}) + \frac{\beta_2}{2}\left(\|\mathbf{s}\|^2 - \|\frac{1}{2}\mathbf{1} - \frac{1}{\beta_2}\Delta f(\mathbf{x})\|^2\right) \end{aligned} \tag{29}$$

for any $\mathbf{s}$. The inequality (29) implies that for any $\mathbf{y} \in \mathbb{Z}_{\geq 0}^L$ in the Euclidean ball $B\left(\mathbf{x} - \frac{1}{\beta_2}\Delta f(\mathbf{x}) + \frac{1}{2}\mathbf{1}, \|\frac{1}{2}\mathbf{1} - \frac{1}{\beta_2}\Delta f(\mathbf{x})\|\right)$ it yields $f(\mathbf{y}) \leq f(\mathbf{x})$, *i.e.*, $\mathbf{y} \in \mathbb{L}_{\mathbf{x}}^{\downarrow}(f)$. We also see that $\mathbf{x}$ is at the surface of this Euclidean ball, *i.e.*, the ball is tangent at $\mathbf{x}$. Furthermore, we also prove that the radius of the ball is lower bounded as follows:

$$\begin{aligned}\frac{1}{\beta_2}\|\frac{\beta_2}{2}\mathbf{1} - \Delta f(\mathbf{x})\| &= \frac{1}{\beta_2}\|\frac{\beta_2}{2}\mathbf{1} - \Delta f(\mathbf{x}^*) + \Delta f(\mathbf{x}^*) - \Delta f(\mathbf{x})\| \\ &\geq \frac{1}{\beta_2}\left(\|\Delta f(\mathbf{x}) - \Delta f(\mathbf{x}^*)\| - \|\frac{\beta}{2}\mathbf{1} - \Delta f(\mathbf{x}^*)\|\right) \\ &\geq \frac{(\alpha - \gamma)}{\beta_2}\|\mathbf{x} - \mathbf{x}^*\|,\end{aligned} \tag{30}$$

where the inequality at the bottom line follows from the third statement of Lemma B.3 and the assumption $\|\frac{\beta_2}{2}\mathbf{1} - \Delta f(\mathbf{x}^*)\| \leq \gamma$.

Next, we show that the ball $B\left(\mathbf{x} - \frac{1}{\beta_2}\Delta f(\mathbf{x}) + \frac{1}{2}\mathbf{1}, \|\frac{1}{2}\mathbf{1} - \frac{1}{\beta_2}\Delta f(\mathbf{x})\|\right)$ contains a cube of edge length $\frac{2(\alpha-\gamma)}{\beta_2\sqrt{L}}\|\mathbf{x} - \mathbf{x}^*\|$. First, we easily know in the ball there exists a cube, of which the volume is sufficiently small, and one vertex is at $\mathbf{x}$. Then, the cube gradually extends all edges until the adjacent vertices of $\mathbf{x}$ touch the surface of the ball. At this moment, it can be seen that the edges that touch the surface of the ball turn the chords of the ball. Furthermore, the line connecting the ball center to $\mathbf{x}$ has an equal angle to all edges connecting $\mathbf{x}$, due to the symmetry of the geometrical shapes. With basic geometry knowledge, we can thus calculate the chord length, *i.e.*, the edge length of the cube, with $2 \times R\cos(\theta)$, where R denotes the radius of the ball and $\theta = \arccos(1/\sqrt{L})$ is the angle between the chord and the "center-$\mathbf{x}$" line. Finally, using (30), we know the cube of the edge length $\frac{2(\alpha-\gamma)}{\beta_2\sqrt{L}}\|\mathbf{x} - \mathbf{x}^*\|$ is tangent at $\mathbf{x}$, and inside sub-level set $\mathbb{L}_{\mathbf{x}}^{\downarrow}(f)$. □

Lemma B.9 (**convex combination in the discrete domain**). *Suppose* $\mathbf{q} = \theta\mathbf{x} + (1-\theta)\mathbf{y}$, $\forall\theta \in [0,1]$, *and there is* $\hat{\mathbf{q}} \in \mathbb{Z}_{\geq 0}^L$ *where* $\mathbf{\Lambda} = \mathbf{q} - \hat{\mathbf{q}}$. *If* f *is* α*-strongly convex, then*

$$\theta f(\mathbf{x}) + (1-\theta)f(\mathbf{y}) \geq f(\hat{\mathbf{q}}) + \left\langle \Delta f(\hat{\mathbf{q}}) - \frac{\alpha}{2}\mathbf{1}, \mathbf{\Lambda}\right\rangle + \frac{\alpha}{2}\|\mathbf{\Lambda}\|^2. \tag{31}$$

Proof. By the definition of the α-strong convexity,

$$\begin{aligned} f(\mathbf{x}) &\geq f(\hat{\mathbf{q}}) + \left\langle \Delta f(\hat{\mathbf{q}}) - \frac{\alpha}{2}\mathbf{1}, \mathbf{x} - \hat{\mathbf{q}}\right\rangle + \frac{\alpha}{2}\|\mathbf{x} - \hat{\mathbf{q}}\|^2; \\ f(\mathbf{y}) &\geq f(\hat{\mathbf{q}}) + \left\langle \Delta f(\hat{\mathbf{q}}) - \frac{\alpha}{2}\mathbf{1}, \mathbf{y} - \hat{\mathbf{q}}\right\rangle + \frac{\alpha}{2}\|\mathbf{y} - \hat{\mathbf{q}}\|^2. \end{aligned} \tag{32}$$

Thus, we have their convex combination as

$$\begin{aligned} \theta f(\mathbf{x}) + (1-\theta)f(\mathbf{y}) &\geq f(\hat{\mathbf{q}}) + \left\langle \Delta f(\hat{\mathbf{q}}) - \frac{\alpha}{2}\mathbf{1}, \mathbf{\Lambda}\right\rangle + \frac{\alpha}{2}\left(\theta\|\mathbf{x}\|^2 + (1-\theta)\|\mathbf{y}\|^2 + \|\hat{\mathbf{q}}\|^2 - 2\left\langle \mathbf{q}, \hat{\mathbf{q}}\right\rangle\right) \\ &\geq f(\hat{\mathbf{q}}) + \left\langle \Delta f(\hat{\mathbf{q}}) - \frac{\alpha}{2}\mathbf{1}, \mathbf{\Lambda}\right\rangle + \frac{\alpha}{2}\left(\|\mathbf{q}\|^2 + \|\hat{\mathbf{q}}\|^2 - 2\left\langle \mathbf{q}, \hat{\mathbf{q}}\right\rangle\right) \\ &= f(\hat{\mathbf{q}}) + \left\langle \Delta f(\hat{\mathbf{q}}) - \frac{\alpha}{2}\mathbf{1}, \mathbf{\Lambda}\right\rangle + \frac{\alpha}{2}\|\mathbf{\Lambda}\|^2 \end{aligned}, \tag{33}$$

where the second inequality follows from the convexity of $\|\cdot\|^2$. The proof is completed. □

Assumption B.10. Assume that $f : \mathbb{Z}_{\geq 0}^L \to \mathbb{R}_+$ of (9) is α-strongly convex, (β_1, β_2)-smooth, and its minimum, denoted $(p^*, \mathbf{x}^*) = \arg\min_{p,\mathbf{x}} f \circ p(\mathbf{x})$, satisfies $\|\Delta f_{p^*}(\mathbf{x}^*) - \frac{\beta_2}{2}\mathbf{1}\| \leq \gamma$ where $0 \leq \gamma < \alpha \leq \beta_1 \leq \beta_2 \leq 1$.

Here the inequality $\|\Delta f_{p^*}(\mathbf{x}^*) - \frac{\beta_2}{2}\mathbf{1}\| \leq \gamma$ implies that, up to a (small) constant vector $\frac{\beta_2}{2}\mathbf{1}$, the finite gradient at $(p^*, \mathbf{x}^*)$ should be sufficiently small, which can be understood as the discrete version of the zero-gradient for the stationary points in the continuous domain. The upper bound "1" is arbitrarily chosen just for simplifying the calculation. Also note that β_2 must be larger than α due to the fact $\alpha\|\mathbf{x} - \mathbf{y}\|^2 \leq \langle \Delta f(\mathbf{x}) - \Delta f(\mathbf{y}), \mathbf{x} - \mathbf{y}\rangle \leq \beta_2\|\mathbf{x} - \mathbf{y}\|^2$ (see Lemma B.3 and Lemma B.5). With Assumption B.10, we next prove that the local-sampling-based searching algorithm achieves the linear convergence rate up to a constant, *if* p^* *is known beforehand*.

Theorem B.11 (convergence rate). *Suppose Assumption B.10 is satisfied, the operator p in (9) is fixed to be p^*, and $0 \leq \theta \leq 1$. Then, for any $\mathbf{x}$ with $\|\mathbf{x} - \mathbf{x}^*\|_\infty \leq c$, we can find a neighborhood $B_\infty(\mathbf{x}, r_\mathbf{x})$ where $r_\mathbf{x} \geq \theta c + \frac{1}{2}$, such that there exist a element $\mathbf{y} \in B_\infty(\mathbf{x}, r_\mathbf{x})$ satisfying*

$$f_{p^*}(\mathbf{y}) - f_{p^*}(\mathbf{x}^*) \leq (1-\theta)(f_{p^*}(\mathbf{x}) - f_{p^*}(\mathbf{x}^*)) + \frac{7}{8}K. \tag{34}$$

Proof. First of all, since the operator p is fixed to be p^*, the problem (9) can be equivalently simplified by removing the formulation of p out of (9), which is written as

$$\min_{\mathbf{x} \in \mathbb{Z}_+^K} f(\mathbf{x}), \tag{35}$$

where $f : \mathbb{Z}_{\geq 0}^K \to \mathbb{R}$ represents the objective function.[7] By Lemma B.9, we have the following inequality:

$$f(\hat{\mathbf{q}}) - f(\mathbf{x}^*) \leq (1-\theta)(f(\mathbf{x}) - f(\mathbf{x}^*)) + \left\langle \frac{\alpha}{2}\mathbf{1} - \Delta f(\hat{\mathbf{q}}), \mathbf{\Lambda} \right\rangle - \frac{\alpha}{2}\|\mathbf{\Lambda}\|^2. \tag{36}$$

Next, we prove in the neighborhood $B(\mathbf{x}, r_\mathbf{x})$ there exists an element $\mathbf{y}$, which belongs to as well the sub-level cube tangent at $\hat{\mathbf{q}}$ knowing by Lemma B.8, so that $f(\mathbf{y}) \leq f(\hat{\mathbf{q}})$ holds. To do so, we first know that the distance between $\hat{\mathbf{q}}$ and $p_\mathbf{x}(\mathbf{x})$ satisfying

$$\|\mathbf{x} - \hat{\mathbf{q}}\|_\infty = \|\mathbf{x} - \mathbf{q} + \mathbf{\Lambda}\|_\infty \leq \|\mathbf{x} - \mathbf{q}\|_\infty + \|\mathbf{\Lambda}\|_\infty = \theta\|\mathbf{x} - \mathbf{x}^*\|_\infty + \|\mathbf{\Lambda}\|_\infty \leq \theta c + \frac{1}{2}. \tag{37}$$

Here the last inequality follows from $\|\mathbf{\Lambda}\|_\infty \leq \frac{1}{2}$, which holds because $\hat{\mathbf{q}} \in \mathbb{Z}_{\geq 0}^K$ can be always found by rounding the entries of $\mathbf{q}$ into the closest integers. We thus know from the inequality that the intersection between the sub-level cube tangent at $\hat{\mathbf{q}}$ and $B(\mathbf{x}, r_\mathbf{x})$ is not empty if $r_\mathbf{x} \geq \theta c + \frac{1}{2}$, proving the existence of the $\mathbf{y}$. Last, we bound (36) as follows:

$$\begin{aligned}
f(\mathbf{y}) - f(\mathbf{x}^*) \leq f(\hat{\mathbf{q}}) - f(\mathbf{x}^*) &\leq (1-\theta)(f(\mathbf{x}) - f(\mathbf{x}^*)) + \left\langle \frac{\alpha}{2}\mathbf{1} - \Delta f(\hat{\mathbf{q}}), \mathbf{\Lambda} \right\rangle - \frac{\alpha}{2}\|\mathbf{\Lambda}\|^2 \\
&\leq (1-\theta)(f(\mathbf{x}) - f(\mathbf{x}^*)) + \left| \left\langle \frac{\alpha}{2}\mathbf{1}, \mathbf{\Lambda} \right\rangle \right| + |\langle \Delta f(\hat{\mathbf{q}}), \mathbf{\Lambda} \rangle| + \frac{\alpha}{2}\|\mathbf{\Lambda}\|^2 \\
&\leq (1-\theta)(f(\mathbf{x}) - f(\mathbf{x}^*)) + \frac{\alpha}{4}K + \|\Delta f(\hat{\mathbf{q}})\|_\infty \|\mathbf{\Lambda}\|_1 + \frac{\alpha}{2}\|\mathbf{\Lambda}\|^2 \\
&\leq (1-\theta)(f(\mathbf{x}) - f(\mathbf{x}^*)) + \frac{\alpha}{4}K + \frac{\beta_1}{2}K + \frac{\alpha}{8}K \\
&\leq (1-\theta)(f(\mathbf{x}) - f(\mathbf{x}^*)) + \frac{3\alpha + 4\beta_1}{8}K \\
&\leq (1-\theta)(f(\mathbf{x}) - f(\mathbf{x}^*)) + \frac{7}{8}K.
\end{aligned} \tag{38}$$

Here the inequality in the fourth line follows from Lemma B.6 and $\|\mathbf{\Lambda}\|_\infty \leq 1/2$, and the inequality at the bottom line follows from Assumption B.10 that $\alpha < \beta_1 \leq 1$. The proof is thus completed. □

It is known from the proof that the constant $(7/8)K$ appearing in (34) is due to the fact $\|\mathbf{\Lambda}\|_1 \leq K\|\mathbf{\Lambda}\|_\infty \leq K/2$ and $\|\mathbf{\Lambda}\|_2 \leq K\|\mathbf{\Lambda}\|_\infty \leq \sqrt{K}/2$. It means that with the rounding error $\|\mathbf{\Lambda}\|_\infty \leq 1/2$, the $l_{1,2}$ norm of $\mathbf{\Lambda}$ would become larger with increasing the dimension K, which is inevitable in the analysis. It only disappears if $\|\mathbf{\Lambda}\|_\infty = 0$, implying the conventional convex optimization in the continuous domain.

As an important corollary from Theorem B.11, we next prove the convergence guarantee for the local-sampling-based methods.

Corollary B.12 (convergence guarantee). *Suppose p^* is known and a series $\{\mathbf{x}_n\}_{n=0}^\infty$, where $\mathbf{x}_0$ is randomly chosen in $\mathbb{Z}_+^K$, and for each $n > 0$, $\mathbf{x}_n$ is equal to the $\mathbf{y}$ in Theorem B.11. Then we can achieve the following limit when $\Omega(1/K) \leq \theta \leq 1$,*

$$\lim_{n \to \infty} (f_{p^*}(\mathbf{x}_n) - f_{p^*}(\mathbf{x}^*)) = O(1) \tag{39}$$

[7] Here for brevity, we re-use the notation of f without ambiguity since the main properties of f are preserved up to the domain restricting from $\mathbb{Z}_{\geq 0}^L$ to $\mathbb{Z}_{\geq 0}^K$.

Proof. Let $C_K := (7/8)K$. By the updating rule,

$$\begin{aligned} f_{p^*}(\mathbf{x}_n) - f_{p^*}(\mathbf{x}^*) &\leq (1-\theta)(f_{p^*}(\mathbf{x}_{n-1}) - f_{p^*}(\mathbf{x}^*)) + C_K \\ &\leq (1-\theta)^2(f_{p^*}(\mathbf{x}_{n-2}) - f_{p^*}(\mathbf{x}^*)) + C_K + C_K(1-\theta) \\ &\leq (1-\theta)^3(f_{p^*}(\mathbf{x}_{n-3}) - f_{p^*}(\mathbf{x}^*)) + C_K + C_K(1-\theta) + C_K(1-\theta)^2 \\ &\leq \cdots \\ &\leq (1-\theta)^n(f_{p^*}(\mathbf{x}_0) - f_{p^*}(\mathbf{x}^*)) + C_K \sum_{m=1}^{n} (1-\theta)^{m-1}. \end{aligned} \tag{40}$$

Thus using the condition $\Omega(1/K) \leq \theta \leq 1$, we finally obtain that

$$\lim_{n\to\infty} (f_{p^*}(\mathbf{x}_n) - f_{p^*}(\mathbf{x}^*)) \leq 0 + C_K \frac{1}{\theta} = O(1). \tag{41}$$

□

B.3. Sampling efficiency

Proposition B.13 (**curse of dimensionality for TNLS**). *Let the assumptions in Theorem B.11 be satisfied. Furthermore, assume that $\mathbf{x}^*$ is sufficiently smaller (or larger) than $\mathbf{x}$ entry-wisely except for a constant number of entries. Then the probability of achieving a suitable $\mathbf{y}$ as mentioned in Theorem B.11 by uniformly randomly sampling in $B_\infty(\mathbf{x}, r_\mathbf{x})$ with $r_\mathbf{x} \geq \theta c + \frac{1}{2}$ equals $O(2^{-K})$.*

Proof. We only prove the case where $\mathbf{x}^*$ is sufficiently smaller than $\mathbf{x}$ in the entry-wise manner, except a constant number of entries. The "larger" case can be proved similarly. Recall Theorem B.11. By the construction of $\mathbf{y}$, we have $\mathbf{q} = \theta\mathbf{x} + (1-\theta)\mathbf{x}^*$ with $0 \leq \theta \leq 1$ and the approximation $\hat{\mathbf{q}} \in \mathbb{Z}_+^K$ with $\hat{\mathbf{q}} = \mathbf{q} + \mathbf{\Lambda}$ and $\|\mathbf{\Lambda}\|_\infty \leq 1/2$. According to the assumptions, we know $\mathbf{x} - \hat{\mathbf{q}}$ is entry-wisely larger than zero except C entries, where $C \geq 0$ is a constant. Since $r_\mathbf{x} \geq \theta c + \frac{1}{2}$, we further know from Theorem B.11 that the intersection between $B_\infty(x, r_\mathbf{x})$, denoted B in the rest of the proof for brevity, and the sub-level cube, denoted A, tangent at $\hat{\mathbf{q}}$ is not empty. In this case, we can easily bound the volume of the cube associated to the intersection of A and B as follows:

$$|A \cap B| \leq (r_\mathbf{x} - \delta_{\min})^{K-C} (r_\mathbf{x} + \delta_{\max})^C. \tag{42}$$

Here $|\cdot|$ denotes the volume of the cube. $\delta_{\min} = \min\{p_i : p_i = \mathbf{x}(i) - \hat{\mathbf{q}}(i) > 0, 1 \leq i \leq K\}$ and $\delta_{\max} = \max\{0, p_i : p_i = \hat{\mathbf{q}}(i) - \mathbf{x}(i) \leq 0, 1 \leq i \leq K\}$, where $\mathbf{x}(i), \hat{\mathbf{q}}(i)$ denote the i-th entry of $\mathbf{x}$ and $\hat{\mathbf{q}}$, respectively. Thus, the probability of uniformly drawing a sample $\mathbf{y}$ belonging to $A \cap B$ from $B_\infty(\mathbf{x}, r_\mathbf{x})$ is as follows:

$$\begin{aligned} Pr(y \in A \cap B) &\leq \frac{(r_\mathbf{x} - \delta_{\min})^{K-C} (r_\mathbf{x} + \delta_{\max})^C}{(2r_\mathbf{x})^K} \leq \left(\frac{r_\mathbf{x} + \delta_{\max}}{r_\mathbf{x} - \delta_{\min}}\right)^C 2^{-K} \\ &= O(2^{-K}). \end{aligned} \tag{43}$$

The proof is completed. □

Recall that let $\mathbb{B} := B(p) \times B_\infty(\mathbf{x}, r_\mathbf{x})$ and $f^*_\mathbb{B} := \min_{(p_y,\mathbf{y})\in\mathbb{B}} f_{p_y}(\mathbf{y})$ for notational simplicity, then

Proposition B.14 (**evaluation efficiency for TnALE**). *Let $\mathcal{B} \in \mathbb{R}^{I\times I\times\cdots\times I}$ be the tensor of order-$(K+1)$ constructed as Eq. (6) with $I_1 = I_2 = \cdots = I_{K+1} = I$. Then, there exists its TT-cross approximation (Oseledets & Tyrtyshnikov, 2010) of rank-R[8], denoted $\hat{\mathcal{B}}$, for which it satisfies $\mathbf{j} = \arg\max_\mathbf{i} \hat{\mathcal{B}}(\mathbf{i})$, such that the equality $f^*_\mathbb{B} = f_{p_{j_{K+1}}}(\mathbf{x} + \mathbf{j}(:K) - (\lceil r_\mathbf{x} \rceil + 1))$ holds, provided that*

$$f^*_\mathbb{B} \leq f_{p_z}(\mathbf{z}) / \left(1 + 2\frac{(4R)^{\lceil \log_2 K \rceil} - 1}{4R - 1}(R+1)^2 \xi f_{p_z}(\mathbf{z})\right) \tag{44}$$

for all $(p_z, \mathbf{z}) \in \mathbb{B}$ and $f_{p_z}(\mathbf{z}) \neq f^*_\mathbb{B}$. Here, ξ denotes the error between $\mathcal{B}$ and its best approximation of TT-ranks R in terms of $\|\cdot\|_\infty$. Note that the inequality (44) holds trivially if $\mathcal{B}$ is exactly of the TT topology of rank-R, and Oseledets & Tyrtyshnikov (2010) shows that the $f^*_\mathbb{B}$ can be recovered from $O(KIR)$ entries from $\mathcal{B}$.

[8] Here we assume that all elements of the TT-ranks are equal to R for brevity.

Proof. Since the "one-to-one" relation between the entries of the tensor $\mathcal{B}$ and all possible $f(\mathbf{z})$ for all $(p_z, \mathbf{z}) \in B(p) \times B_\infty(\mathbf{x}, r_\mathbf{x})$, it is easily to know the equality $f^*_\mathbb{B} = f_{p_{j,K+1}}(\mathbf{x} + \mathbf{j}(:K) - (\lceil r_\mathbf{x} \rceil + 1))$ holds if $\hat{\mathcal{B}}(\mathbf{i}^*) \geq \hat{\mathcal{B}}(\mathbf{k})$ for $\mathbf{i}^* = \arg\max_\mathbf{i} \mathcal{B}(\mathbf{i})$ and any index $\mathbf{k}$. To prove this condition true, we have the following inequalities for any $\mathbf{k}$:

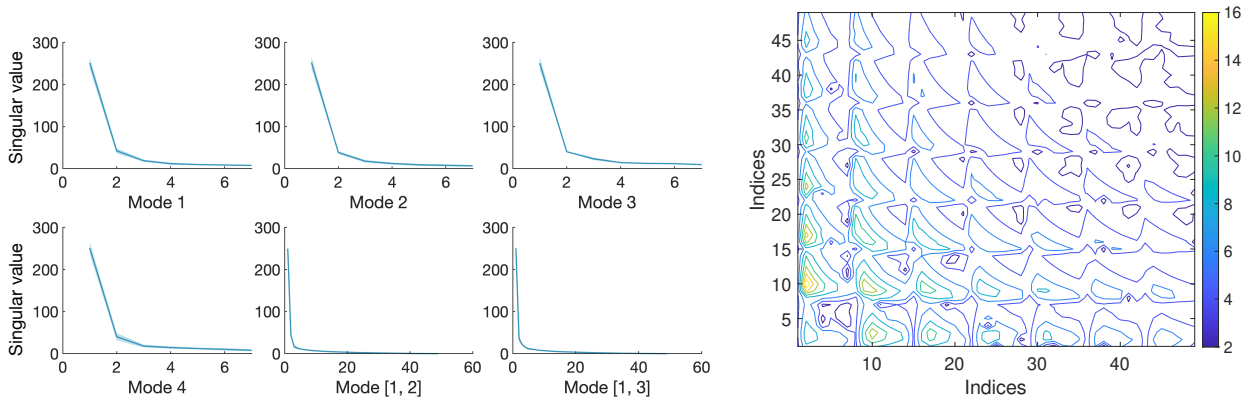
$$\begin{aligned}
\hat{\mathcal{B}}(\mathbf{i}^*) - \hat{\mathcal{B}}(\mathbf{k}) &\geq \mathcal{B}(\mathbf{i}^*) - \mathcal{B}(\mathbf{k}) - 2\frac{(4R)^{\lceil \log_2 K \rceil} - 1}{4R - 1}(R+1)^2\xi \\
&= 1/f^*_\mathbb{B} - 1/f_{p_{j_{K+1}}}(\mathbf{x} + \mathbf{k}(:K) - (\lceil r_\mathbf{x} \rceil + 1)) - 2\frac{(4R)^{\lceil \log_2 K \rceil} - 1}{4R - 1}(R+1)^2\xi, \\
&\geq 2\frac{(4R)^{\lceil \log_2 K \rceil} - 1}{4R - 1}(R+1)^2\xi - 2\frac{(4R)^{\lceil \log_2 K \rceil} - 1}{4R - 1}(R+1)^2\xi = 0
\end{aligned} \tag{45}$$

where the first inequality follows from Theorem 2 in (Osinsky, 2019), and the last inequality follows from the inequality (44). It can also be known if $\mathcal{B}$ is exactly of the TT topology of rank-R, $\hat{\mathcal{B}}$ is able to recover $\mathcal{B}$ exactly. In this case $\xi = 0$ and $f^*_\mathbb{B} \leq f(\mathbf{z})$ trivially for all $\mathbf{z}$. □

C. Experiment details

C.1. Low-rank structure of the optimization landscape

To verify the low-rank structure of the optimization landscape of (1), we empirically check the singular values of the landscape tensor using the synthetic data. To be specific, we re-use the fourth-order tensor in the experiment for TN-PS, *i.e.*, TR (order-4) in Table 5. Here we remove the influence of unknown permutations and calculate the objective for all possible combinations of values of the TN-ranks. As a result, for each data, we have a landscape tensor (a tensor whose entries are values of the objective function) of order-4, and the modes of the tensor corresponding to the four TN-ranks. Figure 6 (a) shows the singular values of the landscape tensor unfolded along different modes on average. We see that the landscape tensor provides a *significant* low-rank structure in the data. We also depict the complete landscape (contour line, unfolded along the first two modes) with respect to Data A in Figure 6 (b). We can see that the obviously repeated pattern shown in the figure is the main reason leading to the low-rank structure of the landscape.



(a) Averaged singular values for the 4th-order landscape tensor.

(b) Optimization landscapes (the inverse 1/f (x)) wrt. the tensor of order-4 and correct permutation.

Figure 6. Averaged singular values and Optimization landscapes for the tensor of order-4.

C.2. Details for the experiment of TN-PS (*w.r.t.*, Table 1).

Goal. In this experiment, our goal is to verify the superiority of TnALE in addressing the TN-PS problem.

Data generation. For the synthetic data with TR topology (order-4, order-6, and order-8), as well as PEPS(order-6), HT (order-6), and MERA (order-8), we *re-use* the data from Li et al. (2022). To generate data with TW (order-5) topology, we set the dimension of each tensor mode to 3. Additionally, we randomly select TN-ranks from the set $\{1, 2, 3\}$. Then we *i.i.d.* draw samples from Gaussian distribution $N(0, 1)$ as the values of core tensors. After contracting these core tensors based on the TW topology, we randomly and uniformly permute the tensor modes.

Settings. In the experiment, we implement TNGA and TNLS as comparison methods. We use the same objective function as described in Li & Sun (2020) for all the methods. Specifically, the objective function of (1) used in the experiment is as follows:

$$F(G, \mathbf{r}) = \underbrace{\frac{1}{\epsilon(G, \mathbf{r})}}_{\text{compression ratio (CR)}} + \lambda \cdot \underbrace{\min_{\mathcal{Z} \in TNS(G, \mathbf{r})} \|\mathcal{X} - \mathcal{Z}\|^2 / \|\mathcal{X}\|^2}_{\text{relative squared error (RSE)}}, \tag{46}$$

where $\mathcal{X}$ denotes the synthetic tensor, and $\epsilon(G, \mathbf{r})$ represents the compression ratio equalling to

$$\epsilon(G, \mathbf{r}) = \frac{\text{Dimension of } \mathcal{X}}{\text{Dimension sum of core tensors of the TN under } (G, \mathbf{r})}.$$

The trade-off parameter λ in (46) is set to 200. For the solver of the inner minimization, we utilize the Adam optimizer Kingma & Ba (2014) with a learning rate of 0.001. Additionally, the core tensors are initialized using Gaussian distribution

Table 5. Experimental results of the TN-PS task on TR topology. In the table, *Eff.* and the required evaluation numbers *#Eva.* are demonstrated. Specifically, *#Eva.* is shown in the square brackets.

Methods	Order 4					order 6					order 8				
	A	B	C	D	E	A	B	C	D	E	A	B	C	D	E
	Eff.↑ *[#Eva.*↓*]*														
TNGA	1.00 [450]	1.00 [450]	1.17 [450]	1.00 [300]	1.00 [450]	1.00 [1500]	1.00 [1350]	1.00 [1650]	1.16 [1650]	1.00 [1050]	1.00 [2850]	1.02 [2250]	1.11 [3950]	1.06 [1950]	0.88 [1500]
TNLS	1.00 [240]	1.00 [300]	1.17 [60]	1.00 [300]	1.00 [360]	1.00 [660]	1.00 [600]	1.00 [660]	1.16 [600]	1.00 [540]	1.00 [1020]	1.02 [960]	1.11 [1320]	1.06 [780]	1.17 [900]
TnALE(ours)	1.00 [**93**]	1.00 [**155**]	1.17 [**31**]	1.00 [**124**]	1.00 [**62**]	1.00 [**156**]	1.00 [**321**]	1.00 [**156**]	1.16 [**156**]	1.00 [**89**]	1.00 [**231**]	1.02 [**308**]	1.11 [**308**]	1.06 [**231**]	1.17 [**178**]

Table 6. Experimental results of the TN-PS task on PEPS, HT, MERA and TW topology. In the table, *Eff.* and the required evaluation numbers *#Eva.* are demonstrated. Specifically, *#Eva.* is shown in the square brackets. The symbol "-" in the table means the failure of the approach.

Methods	PEPS				HT				MERA				TW			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
	Eff.↑ *[#Eva.*↓*]*															
TNGA	1.14 [1560]	-	1.00 [840]	1.21 [1080]	1.45 [960]	1.21 [1320]	1.18 [840]	1.29 [1080]	-	1.32 [960]	2.30 [2800]	1.00 [3240]	1.24 [1920]	2.61 [1440]	1.23 [600]	1.30 [720]
TNLS	1.14 [781]	1.00 [781]	1.00 [421]	1.21 [661]	1.45 [841]	1.21 [841]	1.18 [781]	1.29 [721]	1.09 [1561]	1.88 [841]	2.88 [1441]	1.03 [721]	1.24 [661]	2.61 [601]	1.23 [601]	1.30 [481]
TnALE(ours)	1.14 [**407**]	1.00 [**465**]	1.00 [**233**]	1.21 [**175**]	1.45 [**211**]	1.21 [**281**]	1.18 [**211**]	1.29 [**211**]	1.09 [**1450**]	1.88 [**484**]	2.88 [**323**]	1.03 [**323**]	1.24 [**285**]	2.61 [**143**]	1.23 [**285**]	1.30 [**214**]

$N(0, 0.1)$. Furthermore, the search range for TN-ranks is set from 1 to 7, except for TW data, for which the search range is limited to 1 to 4. For TNGA, the maximum number of generations is set to 30. The population size in each generation is 120 for all the TN topologies except for TR, which is set as 150. During each generation, the elimination rate is 36% and the reproduction trick (Snyder & Daskin, 2006) is adopted and we set the reproduction number to be 2. Meanwhile, for the selection probability of the recombination operation, we set the hyper-parameters $\alpha = 20$ and $\beta = 1$. Moreover, there is a 24% chance for each gene to mutate after the recombination. For TNLS, we set the sample numbers in each local sampling stage to 60. The tuning parameter c_1 is fixed at 0.9 throughout the experiment. As for the tuning parameter c_2, it is adjusted based on the tensor order. Specifically, we set $c_2 = 0.9$ for order-4 TR, $c_2 = 0.94$ for order-6 TR, PEPS, TW and HT, and for MERA and order-8 TR, we set $c_2 = 0.98$. In our proposed method TnALE, we maintain consistent settings throughout the experiment. The rank-related radius is set as $r_1 = 2$ and $r_2 = 1$. During the initialization phase, we perform 2 iterations, and during the searching phase, we conduct 30 iterations. Additionally, we set the number of round-trips of ALE to 1. For performance evaluation, we use the *Eff.* index, and *Eff.*≥ 1 indicates an identical or more compact structure has been found. If the results do not satisfy the conditions of RSE $\leq 10^{-4}$ and *Eff.*≥ 1, we say the approach fails in the experiment.

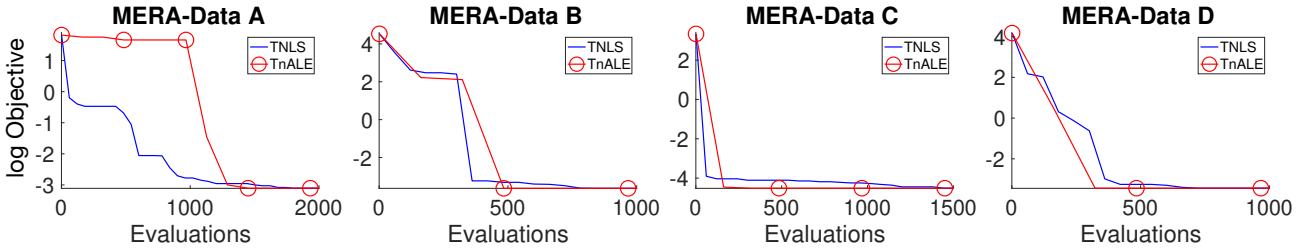


Figure 7. Objective (in the log form) with varying the number of evaluations: an observation of the local convergence of TnALE in MERA.

Results. The results for TR topology are presented in Table 5, and the results for PEPS, HT, MERA, and TW topology are shown in Table 6. Based on the results, we observe that both TNLS and TnALE can successfully identify the ranks and permutations of the data, as indicated by *Eff.*≥ 1. When comparing TNLS and TnALE, we find that TnALE achieves the same results with significantly fewer evaluation requirements. This highlights the superiority of TnALE in solving the TN-PS problem, demonstrating its efficiency and effectiveness. In Figure 4, the averaged log objective curves with varying evaluation numbers of TNLS and TnALE are displayed. It is apparent from the figures that TnALE demonstrates a faster descending trend and achieves lower objective values given the same number of evaluations compared to TNLS for most cases (except for MERA). These results indicate the practical advantage of the proposed method, particularly in scenarios where computational resources are limited, and only a certain number of evaluations can be performed. For the results of

MERA, we further draw the objective curves of each data in Figure 7. From the MERA-Data A curve, it is observed that TnALE descends at a slow pace until approximately 1000 evaluations, whereas TNLS continues to descend. The main reason for this behavior is that TnALE gets trapped in a local optimum and struggles to jump out by restarting the ALE algorithm with a new random center, while TNLS is more likely to overcome such local optima due to its stochastic essence. Moreover, in order to demonstrate the scalability of different TN-PS methods with respect to the tensor order, we draw the average number of evaluations with TR order in Figure 8. From the results, it is evident that the proposed method exhibits a slower increase in the number of evaluations with increasing tensor order compared to other methods. These results highlight the scalability of the proposed method, indicating its ability to handle higher-order tensors effectively.

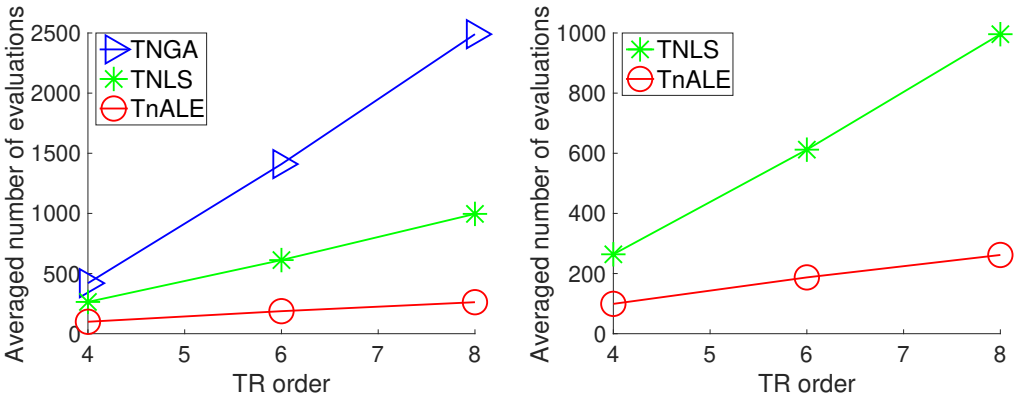


Figure 8. Number of evaluations with varying TR orders.

C.3. Details for the experiment of TN-RS (*w.r.t.* Table 2).

Goal. In this experiment, we consider the classic rank-selection problem, *i.e.*, TN-RS, for TR decomposition.

Data Generation. We generate synthetic tensors in TR topology with two configurations: "*lower-ranks*" and "*higher-ranks*". In both configurations, we generate five tensors by randomly selecting ranks and values of the vertices (core tensors). Each tensor has an order of 8, and the dimensions for each tensor mode are set to 3. We *i.i.d.* draw samples from Gaussian distribution N (0, 1) as the values of the vertices. In the "lower-ranks" group, we uniformly select the TN-ranks from the interval $[1, 4]$ randomly, while in the "higher-ranks" group, we increase the rank interval to $[5, 8]$. This ensures that the ranks would be larger than the dimensions of the tensor modes. This configuration aims to simulate the scenario of the over-determined ranks, which commonly occurs in practice for high-order TNs but has received limited attention in existing works.

Settings. In the experiment, we compare various rank-adaptive TR decomposition methods. These methods include TR-SVD, TR-rSVD, TR-ALSAR, TR-BALS and TR-BALS2 (Zhao et al., 2016), TR-LM (Alg. 2 and Alg. 3) (Mickelin & Karaman, 2020), TRAR (Sedighin et al., 2021). Additionally, the TTOpt algorithm (Sozykin et al., 2022) with ranks [9] equaling 1, 2 is also employed as a baseline. The purpose of including these methods is to assess the effectiveness of the "local-searching" scheme utilized in TnALE (our proposed method) and determine its superiority in comparison to existing approaches. In more detail, for TR-SVD, TR-rSVD, TR-ALSAR, TR-BALS, and TR-BALS2 (Zhao et al., 2016), the available codes have been used.[10] In order to achieve a larger *Eff.* value, we adjust the parameters *tol* and *MaxIter* to ensure the value of RSE is less than but close to 10^{-4}. For TR-LM (Alg. 2 and Alg. 3) (Mickelin & Karaman, 2020), we use the available codes [11] with default parameter settings. However, we adjust the value of *prec* to obtain a larger *Eff.* value. For TRAR (Sedighin et al., 2021), we replace the TR-ALS (Wang et al., 2017) in *Algorithm 1* of Mickelin & Karaman (2020) with the same decomposition method used in TTOpt. This modification is necessary because the initialization method of TR-ALS is not suitable for the case of higher ranks. Regarding TTOpt (Sozykin et al., 2022), we employ the same objective function as used in the TN-PS experiment, with the trade-off parameter $\lambda = 200$. For the *lower ranks* group, the rank searching range is set to $[1, 7]$, while for the *higher ranks* group, the range is extended to $[1, 10]$. During the initialization

[9] Here the ranks are tuning parameters in the TTOpt algorithm.
[10] https://qibinzhao.github.io/
[11] https://github.com/oscarmickelin/tensor-ring-decomposition

phase, we *i.i.d.* draw samples from Gaussian distribution $N(0, 1)$ to generate the values of core tensors. For the proposed method TnALE, we set the rank-related radius $r_1 = 3, r_2 = 2$ for the higher ranks group and $r_1 = 2, r_2 = 1$ for lower ranks group. The number of iterations in the initialization phase is set to 1, the number of iterations in the searching phase is set to 30, and the number of round-trips of ALE is set to 1 throughout the experiments. Other parameters of TnALE are set the same as TTOpt. For TNGA, we set the population in each generation to be 60. The searching ranges and the initialization scheme of core tensors are similar to TTOpt. The other parameters of TNGA are set the same as the TN-PS experiment. For TNLS, we set the sample numbers in each local sampling stage to be 60 and $c_1 = 0.9$, and the other parameters are set the same as in TTOpt. The success condition for all approaches in the experiment is set as RSE $\leq 10^{-4}$ and *Eff.* ≥ 1. If an approach fails to meet these criteria, it is considered a failure in rank selection.

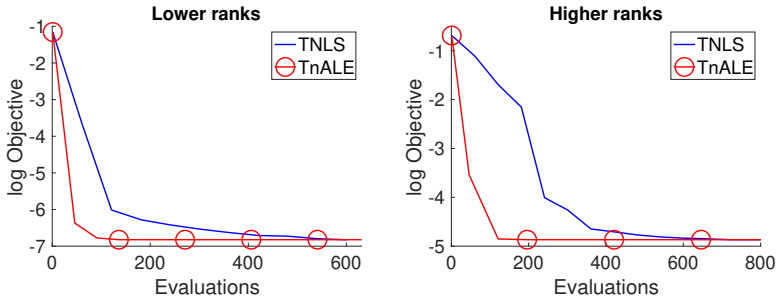


Figure 9. Objective (in the log form) with varying the number of evaluations.

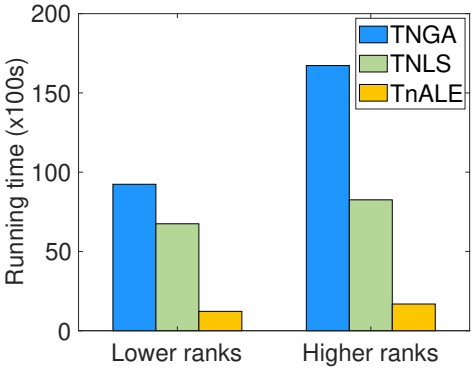


Figure 10. Running time in TN-RS experiment

Results. Based on the results presented in Table 7 and Table 8, it can be observed that in the lower rank regime, TR-BALS, TR-LM (Alg. 2), TTOpt, TNGA, TNLS, and TnALE (ours) are able to successfully select the optimal TR-ranks as indicated by *Eff.*≥ 1 and RSE$\leq 10^{-4}$. However, in the higher rank regime, only TTOpt, TNGA, TNLS, and TnALE (ours) are able to find the optimal ranks. In terms of the number of evaluations, TnALE (ours) outperforms TNGA, TNLS, and TTOpt, requiring the fewest evaluations while still achieving successful rank selection. This highlights the superiority of TnALE in solving the TN-RS problem efficiently. Furthermore, the running time comparison in Figure 10 demonstrates that TnALE saves a significant amount of time compared to TNGA and TNLS, primarily due to its lower number of evaluations. This further emphasizes the advantage of TnALE in scenarios where computational resources are limited. In Figure 9, the averaged log objective curves of TNLS and TnALE with varying evaluation numbers are illustrated. It can be observed that TnALE exhibits a faster descending trend and achieves lower objective values given the same number of evaluations compared to TNLS. This demonstrates the practical advantage of TnALE, particularly in scenarios with restricted computational resources.

C.4. Details for the experiment of knowledge transfer.

Goal. In this experiment, the goal is to investigate the acceleration effect of TnALE when employing the knowledge transfer trick.

Table 7. Experimental results of TN-RS (rank selection) in 8-th order TR topology under the "lower ranks" group. In the first column of the table, A, B, C, D, E (*Data*) and their corresponding vectors (*Rank_grt*) represent the five generated synthetic tensors and the TN-ranks of these five tensors. The item *Rank_est* indicates the specific value of the TN-ranks learned by the corresponding method under the constraint RSE $\leq 10^{-4}$, and *Time (s)* or *[#Eva.]* indicates the running time or the number of evaluations that the method required.

Methods	**TR-SVD**			**TR-rSVD**			**TR-ALSAR**		
Data[Rank_grt]	*Eff.* / RSE	*Rank_est*	*Time (s)*	*Eff.* / RSE	*Rank_est*	*Time (s)*	*Eff.* / RSE	*Rank_est*	*Time (s)*
A [3 4 2 3 1 3 4 2]	0.45 / 8.55E-13	[3 8 4 6 2 6 3 1]	0.0029	0.51 / 0.0013	[3 6 4 6 2 6 3 1]	0.0038	0.08 / 2.43E-05	[11 7 7 7 10 14 8 10]	0.5959
B [3 4 4 2 2 1 1 4]	0.23 / 4.45E-05	[3 9 11 6 6 3 3 1]	0.0081	0.37 / 0.0146	[3 6 6 6 6 3 3 1]	0.0043	0.79 / 4.02E-12	[4 4 4 2 2 2 3 3]	0.0324
C [2 4 2 3 3 1 4 4]	0.29 / 3.55E-05	[3 9 8 4 8 4 3 1]	0.002812	0.46 / 0.0210	[3 6 6 4 6 3 3 1]	0.0039	0.94 / 3.55E-05	[4 4 2 1 2 2 3 4]	0.0333
D [1 4 1 3 4 2 1 1]	**1.13 / 4.78E-05**	[1 2 1 3 4 2 1]	0.0084	**1.13 / 4.78E-05**	[1 2 1 3 4 2 1 1]	0.0055	0.64 / 2.29E-11	[1 3 3 4 4 2 2 1]	0.0233
E [4 1 4 2 3 2 1 1]	**1.17 / 2.04E-13**	[3 1 3 2 3 2 1 1]	0.0043	**1.17 / 3.13E-13**	[3 1 3 2 3 2 1 1]	0.0059	0.78 / 1.45E-11	[3 3 3 2 3 2 2 1]	0.0205
Methods	**TR-BALS**			**TR-BALS2**			**TRAR**		
Data	*Eff.* / RSE	*Rank_est*	*Time (s)*	*Eff.* / RSE	*Rank_est*	*Time (s)*	*Eff.* / RSE	*Rank_est*	*[#Eva.]*
A	**1.00 / 5.00E-13**	[3 4 2 3 1 3 4 2]	0.0146	0.45 / 6.90E-13	[3 8 4 6 2 6 3 1]	0.0188	0.48 / 3.79E-11	[4 4 3 4 5 4 4 3]	69
B	**1.07 / 6.07E-13**	[3 4 4 2 2 1 1 3]	0.0127	0.20 / 1.03E-05	[3 9 12 6 6 3 3 5]	0.0174	0.61 / 3.17E-08	[3 4 4 4 4 2 4 3]	37
C	**1.38 / 3.55E-05**	[2 4 2 1 2 1 3 4]	0.0189	0.26 / 3.57E-05	[3 9 8 4 8 4 3 6]	0.0151	0.65 / 3.55E-05	[3 4 3 5 3 3 3 4]	68
D	**1.13 / 4.78E-05**	[1 2 1 3 4 2 1 1]	0.021	**1.13 / 4.78E-05**	[1 2 1 3 4 2 1 1]	0.0487	0.41 / 5.55E-05	[4 5 3 3 3 3 3 2]	22
E	**1.17 / 6.34E-13**	[3 1 3 2 3 2 1 1]	0.0303	**1.17 / 9.10E-13**	[3 1 3 2 3 2 1 1]	0.0212	0.59 / 2.96E-11	[5 2 3 2 3 2 2 3]	36
Methods	**TR-LM (Alg. 3)**			**TR-LM (Alg. 2)**			**TTOpt** ($R = 1$)		
Data	*Eff.* / RSE	*Rank_est*	*Time (s)*	*Eff.* / RSE	*Rank_est*	*Time (s)*	*Eff.* / RSE	*Rank_est*	*[#Eva.]*
A	0.40 / 4.22E-13	[6 3 1 3 2 6 8 4]	0.0222	**1.00 / 2.86E-14**	[3 4 2 3 1 3 4 2]	0.2736	**1.00 / 9.41E-07**	[3 4 2 3 1 3 4 2]	98
B	0.46 / 2.05E-06	[6 7 3 1 3 2 2 6]	0.0204	**1.07 / 1.08E-14**	[3 4 4 2 2 1 1 3]	0.2402	**1.07 / 2.30E-06**	[3 4 4 2 2 1 1 3]	140
C	**1.38 / 3.55E-05**	[2 4 2 1 2 1 3 4]	0.0227	**1.38 / 3.55E-05**	[2 4 2 1 2 1 3 4]	0.2503	**1.11 / 3.55E-05**	[2 4 2 1 2 2 4 4]	56
D	**1.13 / 4.78E-05**	[1 2 1 3 4 3 1 1]	0.026	**1.13 / 4.78E-05**	[1 2 1 3 4 2 1 1]	0.2407	**1.06 / 4.82E-05**	[1 2 1 3 4 2 1 2]	91
E	**1.17 / 5.62E-14**	[3 1 3 2 3 2 1 1]	0.022	**1.17 / 5.62E-14**	[3 1 3 2 3 2 1 1]	0.2454	**1.17 / 6.61E-11**	[3 1 3 2 3 2 1 1]	133
Methods	**TTOpt** ($R = 2$)			**TTOpt** ($R = 3$)			**TNGA**		
Data	*Eff.* / RSE	*Rank_est*	*[#Eva.]*	*Eff.* / RSE	*Rank_est*	*[#Eva.]*	*Eff.* / RSE	*Rank_est*	*[#Eva.]*
A	**1.00 / 5.00E-06**	[3 4 2 3 1 3 4 2]	518	**1.00 / 8.93E-05**	[3 4 2 3 1 3 4 2]	1533	**1.00 / 9.98E-05**	[3 4 2 3 1 3 4 2]	480
B	**1.02 / 4.86E-07**	[3 4 4 2 2 2 1 3]	336	**1.02 / 3.72E-06**	[3 4 4 2 2 2 1 3]	735	**1.07 / 9.95E-05**	[3 4 4 2 2 1 1 3]	660
C	**1.02 / 3.56E-05**	[2 5 2 2 3 1 3 5]	154	**1.00 / 9.45E-05**	[2 5 2 2 3 2 3 4]	273	**1.11 / 9.95E-05**	[2 4 2 1 2 2 4 4]	600
D	**1.06 / 1.10E-08**	[1 3 1 3 4 2 1 1]	196	**1.00 / 4.87E-05**	[1 2 1 3 4 2 1 3]	483	**1.06 / 9.98E-05**	[1 2 1 3 4 2 1 2]	600
E	**1.03 / 7.94E-06**	[3 1 3 2 3 3 1 1]	364	**1.17 / 3.14E-11**	[3 1 3 2 3 2 1 1]	1071	**1.17 / 9.92E-05**	[3 1 3 2 3 2 1 1]	420
Methods	**TNLS**			**TnALE (ours)**					
Data	*Eff.* / RSE	*Rank_est*	*[#Eva.]*	*Eff.* / RSE	*Rank_est*	*[#Eva.]*			
A	**1.00 / 9.99E-05**	[3 4 2 3 1 3 4 2]	600	**1.00 / 9.98E-05**	[3 4 2 3 1 3 4 2]	66			
B	**1.07 / 9.97E-05**	[3 4 4 2 2 1 1 3]	420	**1.07 / 9.98E-05**	[3 4 4 2 2 1 1 3]	99			
C	**1.11 / 9.99E-05**	[2 4 2 1 2 2 4 4]	420	**1.11 / 9.95E-05**	[2 4 2 1 2 2 4 4]	99			
D	**1.06 / 9.97E-05**	[1 2 1 3 4 2 1 2]	480	**1.06 / 9.98E-05**	[1 2 1 3 4 2 1 2]	69			
E	**1.17 / 9.92E-05**	[3 1 3 2 3 2 1 1]	540	**1.17 / 9.93E-05**	[3 1 3 2 3 2 1 1]	63			

Data generation. We *re-use* the data from the lower ranks group of the TN-RS experiment.

Settings. In this experiment, we employ two variations of TnALE: one incorporates a knowledge transfer trick, while the other does not. Both methods share the same parameter settings, which are listed as follows: the rank searching range is set to $[1, 7]$, the trade-off parameter λ is set to 200, the rank-related radius $r_2 = 2$. Additionally, we set the number of iterations in the initialization phase to 0 and the number of iterations in the searching phase to 30. For the number of round-trips of ALE, we set it to 1. The Adam optimizer is utilized with a learning rate of 0.001, and the core tensors are initialized using Gaussian distribution $N(0, 1)$. Moreover, both methods are initialized with the same TN-ranks.

Results. Figure 11 displays the objective curves as a function of running time. From the figures, it is evident that both methods start with identical log objectives but exhibit significant differences in their descent patterns. In comparison to TnALE without the knowledge transfer trick, TnALE with the knowledge transfer trick showcases a rapid decline in objectives, achieving approximately twice or even nearly five times faster progress than its counterpart.

C.5. Details for the experiment of TGP (*w.r.t.* Table 3).

Goal. In this experiment, our goal is to utilize the proposed method TnALE to compress the learnable parameters of the TGP (Izmailov et al., 2018).

Data generation. In this task, we select three univariate regression datasets from the UCI and LIBSVM archives. The datasets chosen are as follows: The Combined Cycle Power Plant (CCPP)[12] dataset comprises 9569 data points collected from a power plant. It consists of 4 features and a single response. The MG[13] dataset contains 1385 data points with 6

[12] https://archive.ics.uci.edu/ml/datasets/Combined+Cycle+Power+Plant

[13] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/regression.html#mg

Table 8. Experimental results of TN-RS (rank selection) in 8-th order TR topology under the "higher ranks" group. In the first column of the table, A, B, C, D, E (*Data*) and their corresponding vectors (*Rank_grt*) represent the five generated synthetic tensors and the TN-ranks of these five tensors. The item *Rank_est* indicates the specific value of the TN-ranks learned by the corresponding method under the constraint RSE $\leq 10^{-4}$, and *Time (s)* or *[#Eva.]* indicates the running time or the number of evaluations that the method required.

Methods	**TR-SVD**			**TR-rSVD**			**TR-ALSAR**		
Data[Rank_grt]	*Eff.* / RSE	*Rank_est*	*Time (s)*	*Eff.* / RSE	*Rank_est*	*Time (s)*	*Eff.* / RSE	*Rank_est*	*Time (s)*
A [8 8 8 5 7 6 8 8]	0.16 / 9.40E-05	[3 9 27 38 27 9 3 1]	0.0174	0.16 / 9.40E-05	[3 9 27 38 27 9 3 1]	0.0238	1.11 / 0.0172	[7 7 7 6 6 6 8 8]	11.2686
B [6 5 7 7 6 5 6 5]	0.12 / 4.30E-05	[3 9 27 35 26 9 3 1]	0.0136	0.11 / 1.13E-28	[3 9 27 35 27 9 3 1]	0.0361	0.82 / 0.0133	[7 5 6 8 6 6 8 6]	10.536
C [8 7 7 8 7 5 8 7]	0.12 / 8.32E-05	[3 9 27 51 27 9 3 1]	0.0378	0.12 / 8.32E-05	[3 9 27 51 27 9 3 1]	0.0308	0.71 / 0.0094	[9 8 6 17 6 7 9 8]	16.2148
D [6 6 6 8 6 7 6 5]	0.13 / 9.64E-05	[3 9 26 38 26 9 3 1]	0.0092	0.12 / 5.32E-05	[3 9 27 38 27 9 3 1]	0.0308	0.63 / 9.62E-05	[9 8 8 8 7 9 7 7]	0.6589
E [6 6 6 6 5 6 6 6]	0.11 / 4.51E-05	[3 9 27 36 26 9 3 1]	0.0061	0.11 / 5.63E-05	[3 9 27 35 27 9 3 1]	0.0242	0.75 / 0.0298	[7 7 5 6 4 9 8 8]	11.5233
Methods	**TR-BALS**			**TR-BALS2**			**TRAR**		
Data	*Eff.* / RSE	*Rank_est*	*Time (s)*	*Eff.* / RSE	*Rank_est*	*Time (s)*	*Eff.* / RSE	*Rank_est*	*[#Eva.]*
A	0.03 / 6.04E-29	[28 20 19 26 44 89 51 39]	0.8749	0.16 / 8.39E-05	[3 9 27 39 27 9 3 1]	0.142	0.67 / 5.59E-14	[10 8 8 8 8 10 8 11]	76
B	0.57 / 9.25E-05	[6 6 10 9 9 7 9 6]	0.0941	0.12 / 9.89E-05	[3 9 27 35 26 9 3 1]	0.1608	0.71 / 1.82E-13	[8 5 7 7 7 7 6 9]	74
C	0.12 / 9.71E-05	[15 20 18 15 17 19 23 25]	0.2763	0.12 / 6.35E-05	[3 9 27 54 27 9 3 1]	0.1643	0.58 / 8.89E-14	[12 7 10 8 9 10 9 10]	143
D	0.19 / 8.18E-05	[9 15 17 22 10 16 15 10]	0.1773	0.12 / 9.75E-06	[3 9 27 40 26 9 3 1]	0.1704	0.57 / 3.75E-14	[11 6 7 8 8 9 7 10]	76
E	0.03 / 3.97E-05	[38 55 59 20 18 15 19 27]	0.446	0.11 / 6.40E-30	[3 9 27 36 27 9 3 1]	0.1749	0.62 / 2.05E-14	[10 6 7 7 7 8 6 9]	75
Methods	**TR-LM (Alg. 3)**			**TR-LM (Alg. 2)**			**TTOpt** (R = 1)		
Data	*Eff.* / RSE	*Rank_est*	*Time (s)*	*Eff.* / RSE	*Rank_est*	*Time (s)*	*Eff.* / RSE	*Rank_est*	*[#Eva.]*
A	0.16 / 9.40E-05	[3 9 27 38 27 9 3 1]	0.0257	0.16 / 2.87E-05	[3 9 27 39 27 9 3 1]	0.3318	**1.00 / 3.65E-07**	[8 8 8 5 7 6 8 8]	220
B	0.12 / 4.30E-05	[3 9 27 35 26 9 3 1]	0.001	0.15 / 8.36E-05	[3 1 3 9 26 25 25 9]	0.3336	**1.00 / 1.52E-07**	[6 5 7 7 6 5 6 5]	220
C	0.12 / 8.32E-05	[3 9 27 51 27 9 3 1]	0.0303	0.17 / 7.07E-05	[3 1 3 9 27 34 27 9]	0.3285	**1.00 / 1.01E-06**	[8 7 7 8 7 5 8 7]	150
D	0.13 / 9.64E-05	[3 9 26 38 26 9 3 1]	0.023	0.13 / 9.31E-05	[35 27 9 3 1 3 9 25]	0.3173	**1.00 / 1.83E-06**	[6 6 6 8 6 7 6 5]	150
E	0.11 / 4.51E-05	[3 9 27 36 26 9 3 1]	0.0299	0.13 / 9.38E-05	[20 26 9 3 1 3 9 26]	0.3845	**1.00 / 5.01E-07**	[6 6 6 6 5 6 6 6]	150
Methods	**TTOpt** (R = 2)			**TTOpt** (R = 3)			**TNGA**		
Data	*Eff.* / RSE	*Rank_est*	*[#Eva.]*	*Eff.* / RSE	*Rank_est*	*[#Eva.]*	*Eff.* / RSE	*Rank_est*	*[#Eva.]*
A	**1.00 / 6.49E-07**	[8 8 8 5 7 6 8 8]	540	**1.00 / 5.88E-07**	[8 8 8 5 7 6 8 8]	1710	**1.00 / 9.99E-05**	[8 8 8 5 7 6 8 8]	1020
B	**1.00 / 1.61E-07**	[6 5 7 7 6 5 6 5]	1060	**1.00 / 4.00E-07**	[6 5 7 7 6 5 6 5]	2670	**1.00 / 9.94E-05**	[6 5 7 7 6 5 6 5]	660
C	**1.00 / 1.32E-06**	[8 7 7 8 7 5 8 7]	780	**1.00 / 9.23E-07**	[8 7 7 8 7 5 8 7]	1740	**1.00 / 9.95E-05**	[8 7 7 8 7 5 8 7]	1380
D	**1.00 / 3.40E-07**	[6 6 6 8 6 7 6 5]	540	**1.00 / 1.67E-06**	[6 6 6 8 6 7 6 5]	1710	**1.00 / 9.93E-05**	[6 6 6 8 6 7 6 5]	1020
E	**1.00 / 8.02E-13**	[6 6 6 6 5 6 6 6]	840	**1.00 / 1.99E-07**	[6 6 6 6 5 6 6 6]	1740	**1.00 / 9.94E-05**	[6 6 6 6 5 6 6 6]	420
Methods	**TNLS**			**TnALE (ours)**					
Data	*Eff.* / RSE	*Rank_est*	*[#Eva.]*	*Eff.* / RSE	*Rank_est*	*[#Eva.]*			
A	**1.00 / 9.97E-05**	[8 8 8 5 7 6 8 8]	540	**1.00 / 9.96E-05**	[8 8 8 5 7 6 8 8]	115			
B	**1.00 / 9.92E-05**	[6 5 7 7 6 5 6 5]	720	**1.00 / 9.91E-05**	[6 5 7 7 6 5 6 5]	85			
C	**1.00 / 9.91E-05**	[8 7 7 8 7 5 8 7]	480	**1.00 / 1.00E-04**	[8 7 7 8 7 5 8 7]	150			
D	**1.00 / 9.93E-05**	[6 6 6 8 6 7 6 5]	600	**1.00 / 9.92E-05**	[6 6 6 8 6 7 6 5]	160			
E	**1.00 / 9.92E-05**	[6 6 6 6 5 6 6 6]	600	**1.00 / 9.93E-05**	[6 6 6 6 5 6 6 6]	85			

features and a single response. The Protein[14] dataset consists of 45730 instances with 9 attributes and a single response. For each of the datasets, we begin by randomly selecting 80% of the data for training purposes, while the remaining 20% is reserved for testing. Subsequently, we standardize the training and testing sets respectively by removing the mean and scaling them to have unit variance. In the case of the CCPP dataset, we opt to use 12 inducing points on each feature, resulting in an order-4 tensor with dimensions of $12 \times 12 \times 12 \times 12$. For the MG dataset, we choose 8 inducing points, which leads to an order-6 tensor with dimensions of $8 \times 8 \times 8 \times 8 \times 8 \times 8$. Lastly, for the Protein dataset, we choose 4 inducing points, generating an order-9 tensor with dimensions of $4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4$. Across all datasets, we set the TT-ranks for the TGP (Izmailov et al., 2018) algorithm to 10.

Settings. In the comparison of methods, we employ the same objective function as used in the TN-PS experiment. Additionally, we set specific values for certain parameters, $\lambda = 1 \times 10^5, 1 \times 10^7, 1 \times 10^3$ for CCPP, MG and Protein, respectively. Moreover, the following settings are common for all the methods being compared: the rank searching range, the learning rate of Adam, and the variance of the Gaussian distribution for core tensors initialization are set from 1 to 14, 0.001, and 0.01, respectively. For the TNGA method, we set the maximum number of generations to 30. The population in each generation is set to be 150, 190, and 300 for the TT variational mean of CCPP, MG, and Protein regression tasks. The elimination rate is set at 30% and the reproduction number is set to 1. Additionally, we assign $\alpha = 20$ and $\beta = 1$. The chance for each gene to mutate after the recombination is 30%. For TNLS, the maximum iteration is limited to 20, and the tuning parameters c_1 = 0.9, c_2 = 0.9. For the TT variational mean of CCPP, MG, and Protein regression tasks, we determined the number of samples in the local sampling stage to be 150, 300, and 300 respectively. For the proposed method TnALE, we consistently use the rank-related radius $r_1 = 3$ and $r_2 = 2$. In addition, we specifically designate the number of iterations in the initialization phase as 2 and the number of iterations in the searching phase as 30. Furthermore, we configure the number of round-trips in ALE to be 1.

[14] https://archive.ics.uci.edu/ml/datasets/Physicochemical+Properties+of+Protein+Tertiary+Structure

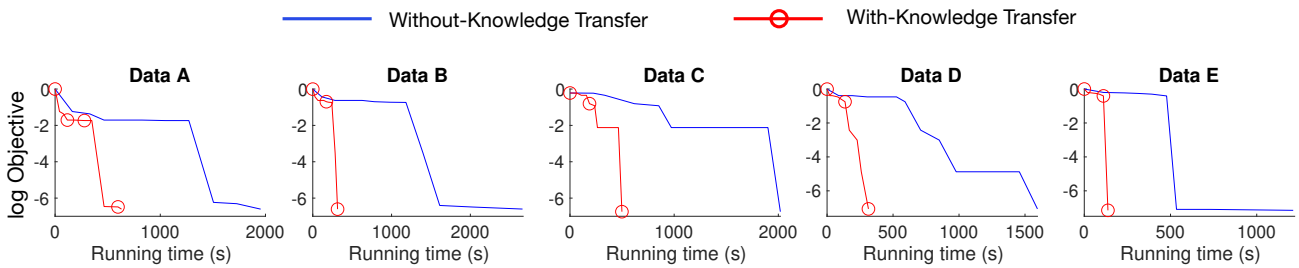


Figure 11. Objective (in the log form) curves with running time

In order to achieve more compact representations, we apply the TN-PS algorithms, which consist of TNGA, TNLS, and the proposed TnALE to TGP. The process involves training an initial TGP model with predefined TT-ranks and obtaining the TT representation of the variational mean. Subsequently, the TN-PS algorithms are employed to search for alternative structures that have a reduced number of parameters for the TT variational mean. Upon completion of the TN-PS algorithms, we reintegrate the reparameterized variational mean back into the original TGP model for inference. The performance is evaluated by measuring the mean squared error (MSE) of the regression tasks conducted on the test datasets.

C.6. Details for the experiment of natural images compression (*w.r.t.* Table 4).

Goal. In this experiment, we will investigate the effectiveness of the proposed TnALE method in tackling the TN-PS and TN-TS tasks associated with compressing natural images. Specifically, in TN-TS, our aim is to search for good TN-ranks and topologies for compressing images.

Data generation. For this experiment, we select 4 natural images from the BSD500 dataset (Arbelaez et al., 2010)[15] at random, as shown in Figure 5. The selected images are first converted to grayscaled images of size 256×256 using the "rgb2gray" and "resize" functions in Matlab, then scaled to the range of $[0, 1]$. Finally, we apply the Matlab function "reshape" directly to the preprocessed images to represent them as order-8 tensors of size $4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4$.

Settings. In the TN-PS task of the experiment, we use the same objective function as in the TN-PS experiment and set the tuning parameter $\lambda = 5$. The rank searching range, the learning rate of Adam, and the variance of the Gaussian distribution for core tensors initialization are set from 1 to 14, 0.01, and 0.1, respectively. For TNLS, we set the maximum number of iterations to 20, and tuning parameters c_1 = 0.95, c_2 = 0.9, and the number of samples in the local sampling stage to 150. In TNGA, the maximum number of generations is set to 30, with a population of 300 per generation. The elimination is set at 10% and the reproduction number is set to 1. We also set α = 25, β = 1, and establish a 30% chance for each gene to mutate following the recombination process. Regarding the proposed method TnALE, we set the rank-related radius as $r_1 = 3$ and $r_2 = 2$. We also set the number of iterations in the initialization phase to 1 and the number of iterations in the searching phase to 30. Finally, we set the number of round-trips of ALE to 1.

In the TN-TS task of the experiment, we use the same objective function, the learning rate of Adam, and the variance of the Gaussian distribution for core tensors initialization as in the TN-PS part, but set the rank searching range from 1 to 4. For TNLS, we set the maximum number of iterations to 20, tuning parameters c_1 = 0.99, and the number of samples in the local sampling stage to 100. For the parameter settings of TNGA, we only change the population number to 100 compared to the TN-PS part. For the proposed method TnALE, we set the rank-related radius r_2 to 1 and the number of iterations in the initialization phase to 0, while the number of iterations in the searching phase to 30. The number of round-trips of ALE is also set to 1. For Greedy, we set the RSE threshold to the same value as the result RSE of the proposed method TnALE.

[15] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/BSDS300/html/dataset/images.html