Efficient construction of tensor ring representations from sampling*

Yuehaw Khoo[†] Jianfeng Lu[‡] Lexing Ying[§]

November 6, 2017

Abstract

In this note we propose an efficient method to compress a high dimensional function into a tensor ring format, based on alternating least-squares (ALS). Since the function has size exponential in d where d is the number of dimensions, we propose efficient sampling scheme to obtain $O(d)$ important samples in order to learn the tensor ring. Furthermore, we devise an initialization method for ALS that allows fast convergence in practice. Numerical examples show that to approximate a function with similar accuracy, the tensor ring format provided by the proposed method has less parameters than tensor-train format and also better respects the structure of the original function.

1 Introduction

Consider a function $f : [n]^d \to \mathbb{R}$ which can be treated as a tensor of size n^d ($[n] := \{1, \dots, n\}$). We want to compress f into a *tensor ring* (TR), i.e., to find 3-tensors $H^1, \dots, H^d$ such that for $x := (x_1, \dots, x_d) \in [n]^d$

$$f(x_1, \dots, x_d) \approx \operatorname{Tr}\left(H^1(:, x_1, :)H^2(:, x_2, :) \cdots H^d(:, x_d, :)\right). \tag{1}$$

Here $H^k \in \mathbb{R}^{r_{k-1}\times n\times r_k}, r_k \le r$ and we often refer to $(r_1, \dots, r_d)$ as the TR rank. Such type of tensor format can be viewed as a generalization of the tensor train (TT) format proposed in [14], better known as the matrix product states (with open boundaries) proposed earlier in the physics literature, see e.g., [1, 15] and recent reviews [17, 12]. The difference between TR and TT is illustrated in Figure 1 using tensor network diagram introduced in Section 1.1. Due to the exponential number of entries, typically we do not have access to the entire tensor f. Therefore, TR format has to be found based on "interpolation" from $f(\Omega)$ where Ω is a subset of $[n]^d$. For simplicity, in the rest of the note, we assume $r_1 = r_2 = \ldots = r_d = r$.

1.1 Notations

We first summarize the notations used in this note and introduce tensor network diagrams for the ease of presentation. Depending on the context, f is often referred to as a d-tensor of size n^d (instead of a function). For a p-tensor T, given two disjoint subsets $\alpha, \beta \subset [p]$ where $\alpha \cup \beta = [p]$, we use

$$T_{\alpha;\beta} \tag{2}$$

to denote the reshaping of T into a matrix, where the dimensions corresponding to sets α and β give rows and columns respectively. Often we need to sample the values of f on a subset of $[n]^d$ grid points. Let α and β be two groups of dimensions where $\alpha \cup \beta = [d], \alpha \cap \beta = \emptyset$, and Ω_1 and Ω_2 be some subsampled grid points along the subsets of dimensions α and β respectively. We use

$$f(\Omega_1; \Omega_2) := f_{\alpha;\beta}(\Omega_1 \times \Omega_2) \tag{3}$$

*The work of Y.K. and L.Y. is supported in part by the National Science Foundation under award DMS-1521830, the U.S. Department of Energy's Advanced Scientific Computing Research program under award DE-FC02-13ER26134/DE-SC0009409, and KI-Net NSF grant RNMS-1107465. The work of J.L. is supported in part by the National Science Foundation under award DMS-1454939 and KI-Net NSF grant RNMS-1107465.

[†] Department of Mathematics, Stanford University, Stanford, CA 94305, USA (ykhoo@stanford.edu).

[‡] Department of Mathematics, Department of Chemistry and Department of Physics, Duke University, Durham, NC 27708, USA (jianfeng@math.duke.edu).

[§] Department of Mathematics and ICME, Stanford University, Stanford, CA 94305, USA (lexing@stanford.edu).

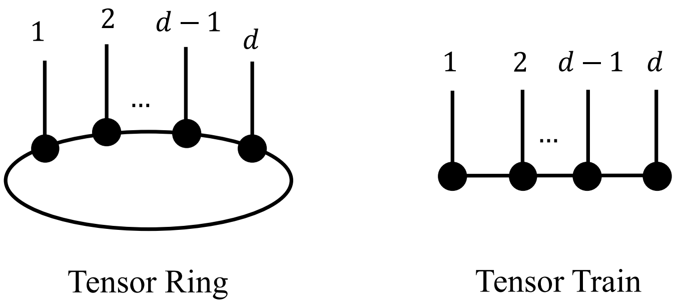


Figure 1. Comparison between a tensor ring and a tensor train.

to indicate the operation of reshaping f into a matrix, followed by rows and columns subsampling according to Ω_1, Ω_2. For any vector $x \in [n]^d$ and any integer i, we let

$$x_i := x_{[(i-1) \bmod d]+1}. \tag{4}$$

For a p-tensor T, we define its Frobenius norm as

$$\|T\|_F := \left(\sum_{i_1,\ldots,i_p} T(i_1,\ldots,i_p)^2 \right)^{1/2}. \tag{5}$$

The notation vec(A) is used to denote the vectorization of a matrix A, formed by stacking the columns of A into a vector. We also use $\alpha \setminus \beta$ to denote the relative complement of set β with respect to set α.

In this note, for the convenience of presentation, we use tensor network diagrams to represent tensors and contraction between them. A tensor is represented as a node, where the number of legs of a node indicates the dimensionality of the tensor. For example Figure 2a shows a 3-tensor A and a 4-tensor B. When joining edges between two tensors (for example in Figure 2b we join the third leg of A and first leg of B), we mean (with the implicit assumption that the dimensions represented by these legs have the same size)

$$\sum_k A_{i_1 i_2 k} B_{k j_2 j_3 j_4}. \tag{6}$$

See the review article [12] for a more complete introduction of tensor network diagrams.

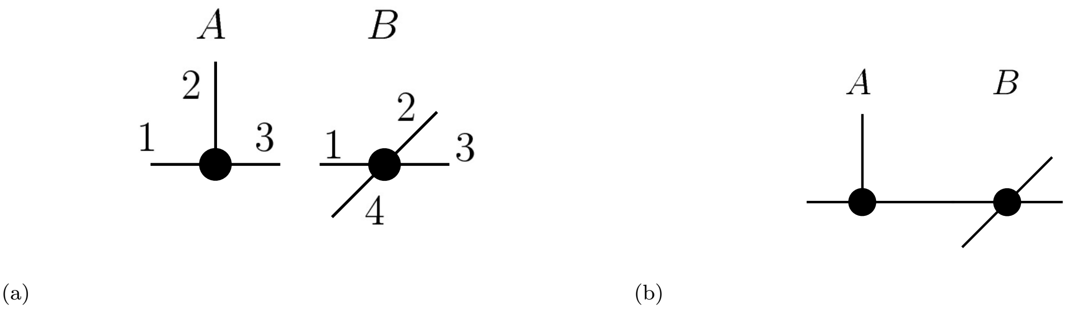


Figure 2. (a) Tensor diagram for a 3-tensor A and a 4-tensor B. (b) Contraction between tensors A and B.

1.2 Previous approaches

In [13], successive CUR (skeleton) decompositions [6] are applied to find a decomposition of tensor f in TT format. In [5], a similar scheme is applied to find a TR decomposition of the tensor. A crucial step in [5] is to "disentangle" one of the 3-tensors H^k's, say H^1, from the tensor ring. First, f is treated as a matrix where the first dimension of f gives rows, the 2-nd, 3-rd, $\ldots$, d-th dimensions of f give columns, i.e., reshaping f to $f_{1;[d]\backslash 1}$. Then CUR decomposition is applied such that

$$f_{1;[d]\backslash 1} = CUR \tag{7}$$

and the matrix $C \in \mathbb{R}^{n\times r^2}$ in the decomposition is regarded as $H^1_{2;3,1}$ (the R part in CUR decomposition is never formed due to its exponential size). As noted by the authors in [5], a shortcoming of the method lies in the reshaping of C into H^1. As in any factorization of low-rank matrix, there is an inherent ambiguity for CUR decomposition in that $CUR = CAA^{-1}UR$ for any invertible matrix A. Such ambiguity in determining H^1 may lead to large tensor-ring rank in the subsequent determination of $H^2, H^3 \ldots, H^d$. More recently, [20] proposes various ALS-based techniques to determine the TR decomposition of a tensor f. However, they only consider the situation where entries of f are fully observed, which limits the applicability of their algorithms to the case with rather small d.

1.3 Our contributions

In this note, assuming f admits a rank-r TR decomposition, we propose an ALS-based two-phase method to reconstruct the TR when only a few entries of f can be sampled. Here we summarize our contributions.

1. The optimization problem of finding the TR decomposition is non-convex hence requires good initialization in general. We devise method for initializing $H^1, \ldots, H^d$ that helps to resolve the aforementioned ambiguity issues via certain probabilistic assumption on the function f.

2. When updating each 3-tensors in the TR, it is infeasible to use all the entries of f. We devise a hierarchical strategy to choose the samples of f efficiently via interpolative decomposition.

While we focus in this note the problem of construction tensor ring format, the above proposed strategies can be applied to tensor networks in higher spatial configuration (like PEPS, see e.g., [12]), which will be considered in future works.

The paper is organized as followed. In Section 2 we detail the proposed algorithm. In Section 3, we demonstrate the effectiveness of our methods through numerical examples. Finally we conclude the paper in Section 4. In Appendix A, we further provide intuition and justification of the proposed initialization procedure, based on certain probabilistic assumption on f.

2 Proposed method

In order to find a tensor ring decomposition (1), our overall strategy is to solve the minimization problem

$$\min_{H^1,\ldots,H^d} \sum_{x\in[n]^d} \left(\mathrm{Tr}(H^1[x_1]\cdots H^d[x_d]) - f(x_1,\ldots,x_d)\right)^2 \tag{8}$$

where

$$H^k[x_k] := H^k(:, x_k, :) \in \mathbb{R}^{r\times r}$$

denotes the x_k-th slice of the 3-tensor H^k along the second dimension. It is computationally infeasible just to set up problem (8), as we need to evaluate f n^d times. Therefore, analogous to the matrix or CP-tensor [8, 4] completion problem [3, 19], a "tensor ring completion" problem

$$\min_{H^1,\ldots,H^d} \sum_{x\in\Omega} \left(\mathrm{Tr}(H^1[x_1]\cdots H^d[x_d]) - f(x_1,\ldots,x_d)\right)^2 \tag{9}$$

where Ω is a subset of $[n]^d$ should be solved instead. Since there are a total of dnr^2 parameters for the tensors $H^1, \ldots, H^d$, there is hope that by observing a small number of entries in f (at least $O(ndr^2)$), we can obtain the rank-r TR.

A standard approach for solving the minimization problem of the type (9) is via alternating least-squares (ALS). At every iteration of ALS, a particular H^k is treated as variable while $H^l, l \neq k$ are kept fixed. Then H^k is optimized w.r.t. the least-squares cost in (9). More precisely, to determine H^k, we solve

$$\min_{H^k} \sum_{x \in \Omega} \left(\mathrm{Tr}(H^k[x_k] C^{x,k}) - f(x) \right)^2, \tag{10}$$

where each coefficient matrix

$$C^{x,k} = H^{k+1}[x_{k+1}] \dots H^d[x_d] H^1[x_1] \dots H^{k-1}[x_{k-1}], \quad x \in \Omega. \tag{11}$$

As mentioned previously, $|\Omega|$ should be at least $O(ndr^2)$. This creates a large computational cost in each iteration of the ALS, as it takes $|\Omega|(d-1)$ matrix multiplications in general just to construct $C^{x,k}$ for all $x \in \Omega$ if Ω does not admit any exploitable pattern. When d is large, such quadratic scaling in d for setting up the least-squares problem in each iteration of the ALS is undesirable.

Such concern on computational cost motivates us to use different Ω_k's to determine different H^k's in the ALS steps instead of using a fixed set Ω. If Ω_k is constructed from densely sampling the dimensions near k (where neighborhood is defined according to ring geometry) while sparsely sampling the dimensions far away from k, computational savings can be achieved. The specific construction of Ω_k is made precise in Section 2.1. We further remark that if

$$\mathrm{Tr}(H^k[x_k] C^{x,k}) \approx f(x) \tag{12}$$

with a small approximation error for every $x \in [n]^d$, then using any $\Omega_k \in [n]^d$ in place of Ω in (10) should give similar solutions, as long as (10) is well-posed. This motivates us to solve

$$\min_{H^k} \sum_{x \in \Omega_k} \left(\mathrm{Tr}(H^k[x_k] C^{x,k}) - f(x) \right)^2 \tag{13}$$

instead of (10) where the index sets Ω_k's depend on k. We note that in practice, a regularization term $\lambda \sigma_k \|H^k(x_k)\|_F^2$ is added to the cost in (13) to reduce numerical instability resulting from potential high condition number of the least-squares problem (13). In all of our experiments, λ is set to 10^{-9} and σ_k is the top singular values of the Hessian of the least-squares problem (13). The quality of TR is rather insensitive to the choice of λ as long as the value is kept small.

At this point it is clear that there are two issues needed to be addressed. The first issue is concerning the selection of $\Omega_k, k \in [d]$. The second issue lies in the initialization of ALS. Since problem (9) is a non-convex optimization problem, it is important to have a good initialization. We solve the first issue using a hierarchical sampling strategy. As for the second issue, by making certain probabilistic assumption on f, we are able to obtain cheap and intuitive initialization. Before moving on, we summarize the full algorithm in Algorithm 1.

Algorithm 1 Alternating least squares

```
Require:
    Function $f : [n]^d \to \mathbb{R}$.
Ensure:
    Tensor ring $H^1, \dots, H^d \in \mathbb{R}^{r \times n \times r}$.
Identify the index sets $\Omega_k$'s and compute $f(\Omega_k)$ for each $k \in [d]$ (Section 2.1).
Initialize $H^1, \dots, H^d$ (Section 2.2).
Start ALS by solving (13) for each $k \in [d]$ (Section 2.3).
```

2.1 Constructing Ω_k

In this section, we detail the construction of Ω_k for each $k \in [d]$. We first construct an index set $\Omega_k^{\mathrm{envi}} \subset [n]^{d-3}$ with fixed size s. The elements in Ω_k^{envi} corresponds to different choices of indices for the $[d] \setminus \{k-1, k, k+1\}$-th dimensions of the function f. For each of the elements in Ω_k^{envi}, we sample all possible indices from the $(k-1)$-th, k-th, $(k+1)$-th dimensions of f to construct Ω_k, i.e., letting

$$\Omega_k = [n]^3 \times \Omega_k^{\mathrm{envi}}. \tag{14}$$

In this case, when determining $C^{x,k}, x \in \Omega_k$ in (13), $O(|\Omega_k^{\mathrm{envi}}|d + |\Omega_k^{\mathrm{envi}}|n^2)$ multiplications of $r \times r$ matrices are needed, giving a complexity that is linear in d.

It remains that Ω_k^{envi}'s need to be constructed. There are two criteria we use for constructing $\Omega_k^{\mathrm{envi}}, k \in [d]$. First, we require that the range of $f_{k;[d]\backslash k}(\Omega_k)$ to be the same as the range of $f_{k;[d]\backslash k}$. Since we want to obtain H^k's such that

$$H^k_{2;3,1}[\mathrm{vec}(C^{x,k})]_{x\in[n]^d} = f_{k;[d]\backslash k}, \tag{15}$$

and we expect

$$H^k_{2;3,1}[\mathrm{vec}(C^{x,k})]_{x\in\Omega_k} \approx f_{k;[d]\backslash k}(\Omega_k) \tag{16}$$

upon solving (13), it is necessary that the columns of $f_{k;[d]\backslash k}(\Omega_k)$ span the range of $f_{k;[d]\backslash k}$. Here we emphasize that it is possible to reshape $f(\Omega_k)$ into a matrix $f_{k;[d]\backslash k}(\Omega_k)$ due to the product structure of Ω_k in (14), where the indices along dimension k are fully sampled. The second criteria is that we require the cost in (13) to approximate the cost in (8).

To meet the first criteria, we propose a hierarchical strategy to determine Ω_k^{envi} such that $f_{k;[d]\backslash k}(\Omega_k)$ has large singular values. Assuming $d = 3 \cdot 2^L$ for some natural number L, we summarize such strategy in Algorithm 2 (the upward pass) and 3 (the downward pass). The dimensions are divided into groups of size $3 \cdot 2^{L-l}$ on each level l for $l = 1, \dots, L$. We emphasize that level $l = 1$ corresponds to the coarsest partitioning of the dimensions of the tensor f. The purpose of the upward pass is to hierarchically find skeletons $\Theta_k^{\mathrm{in},l}$ which represent the k-th group of indices, while the downward pass hierarchically constructs representative environment skeletons $\Theta_k^{\mathrm{envi},l}$. At each level, the skeletons are found by using rank revealing QR (RRQR) factorization [9].

Algorithm 2 Upward pass

Require:

Function $f : [n]^d \to \mathbb{R}$, number of skeletons s.

Ensure:

Skeleton sets $\Theta_k^{\mathrm{in},l}$'s

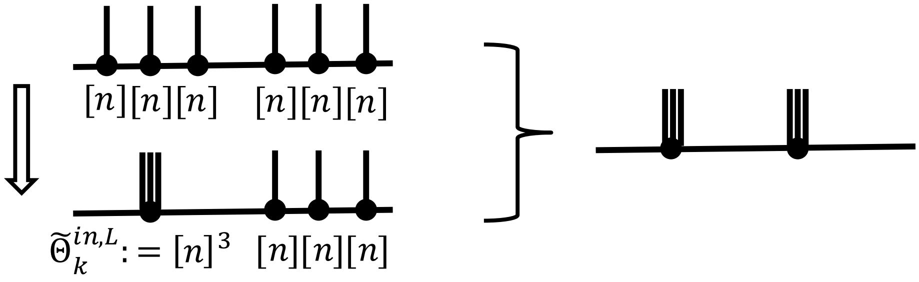
1: Decimate the number of dimensions by clustering every three dimensions. More precisely, for each $k \in [2^L]$, let

$$\tilde{\Theta}_k^{\mathrm{in},L} := \{(x_{3k-2}, x_{3k-1}, x_{3k}) \mid x_{3k-2}, x_{3k-1}, x_{3k} \in [n]\}.$$

There are 2^L index-sets after this step. For each $k \in [2^L]$, construct the set of environment *skeletons*

$$\Theta_k^{\mathrm{envi},l} \subset [n]^{d-3}, \tag{17}$$

with s elements either by selecting multi-indices from $[n]^{d-3}$ randomly, or by using the output of Algorithm 3 (when an iteration of upward and downward passes is employed). This step is illustrated in the following figure.



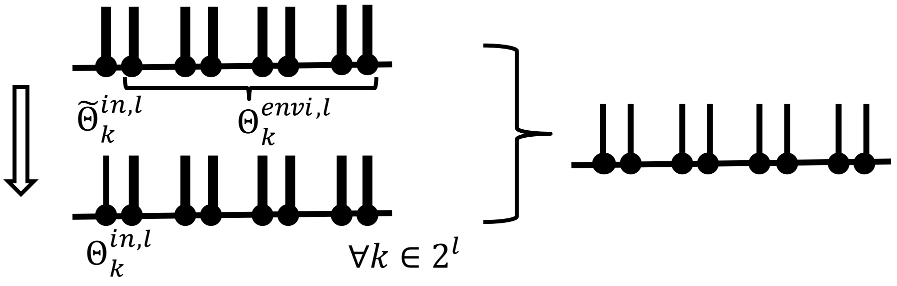
(continued on page 6.)

```
    (continued from Algorithm 2.)

for $l = L$ to $l = 1$
2:    Find the skeletons within each index-set $\tilde{\Theta}_k^{\text{in},l}$, $k \in [2^l]$ where the elements in each $\tilde{\Theta}_k^{\text{in},l}$ are multi-indices
      of length $3 \cdot 2^{L-l}$. Apply RRQR factorization to the matrix

          $f(\Theta_k^{\text{envi},l}; \tilde{\Theta}_k^{\text{in},l}) \in \mathbb{R}^{s \times |\tilde{\Theta}_k^{\text{in},l}|}$    (18)

      to select $s$ columns that best resembles the range of $f(\Theta_k^{\text{envi},l}; \tilde{\Theta}_k^{\text{in},l})$. The multi-indices for these $s$
      columns form the set $\Theta_k^{\text{in},l}$. Store $\Theta_k^{\text{in},l}$ for each $k \in [2^l]$. This step is illustrated in the following figure,
      where the thick lines are used to denote the index-sets with size larger than $s$.
```



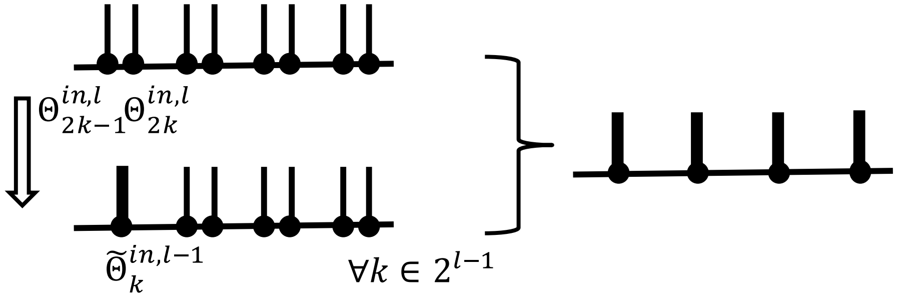
```
3:    If $l > 1$, for each $k \in [2^{l-1}]$, construct

          $\tilde{\Theta}_k^{\text{in},l-1} := \Theta_{2k-1}^{\text{in},l} \times \Theta_{2k}^{\text{in},l}.$    (19)

      Then, sample $s$ elements randomly from

          $\prod_{j \in [2^l] \setminus \{2k-1, 2k\}} \Theta_j^{\text{in},l}$    (20)

      to form $\Theta_k^{\text{envi},l-1}$, or by using the output of Algorithm 3 (when an iteration of upward and downward
      passes is employed). This step is depicted in the next figure, and again thick lines are used to denote
      the index-sets with size larger than $d$.
```


$\Theta_{2k-1}^{in,l}\Theta_{2k}^{in,l}$
$\tilde{\Theta}_k^{in,l-1}$
$\forall k \in 2^{l-1}$

```
end for
```

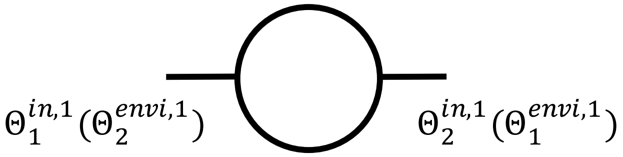
After a full upward-downward pass where RRQR are called $O(d \log d)$ times, $\Theta_k^{\text{envi},L}$ with $k \in [2^L]$ are obtained. Then another upward pass can be re-initiated. Instead of sampling new $\Theta_k^{\text{envi},l}$'s, the stored $\Theta_k^{\text{envi},l}$'s in the downward pass are used. Multiple upward-downward passes can be called to further improved these skeletons. Finally, we let

$$\Omega_{3k-1}^{\text{envi}} := \Theta_k^{\text{envi}}, \quad k \in [2^L]. \tag{21}$$

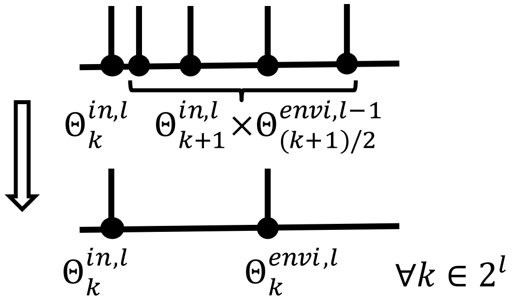
Observe that we have only obtained Ω_k^{envi} for $k = 2, 5, \ldots, d-1$. Therefore, we need to apply upward-downward pass to different groupings of tensor f's dimensions in step (1) of the upward pass. More precisely, we group the dimensions as $(2,3,4),(5,6,7),\ldots,(d-1,d,1)$ and $(d,1,2),(3,4,5),\ldots,(d-3,d-2,d-1)$ when initializing the upward pass to determine Ω_k^{envi} with $k=3,6,\ldots,d$ and $k=1,4,\ldots,d-2$ respectively.

Algorithm 3 Downward pass

```
Require:
    Function $f:[n]^d \to \mathbb{R}$, $\Theta_k^{\mathrm{in},l}$'s from the upward pass, number of skeletons $s$.
Ensure:
    Skeletons $\Theta_k^{\mathrm{envi},l}$'s
Let $\Theta_1^{\mathrm{envi},1} = \Theta_2^{\mathrm{in},1}$, $\Theta_2^{\mathrm{envi},1} = \Theta_1^{\mathrm{in},1}$.
```


$\Theta_1^{in,1}(\Theta_2^{envi,1})$
$\Theta_2^{in,1}(\Theta_1^{envi,1})$

```
for $l = 2$ to $l = L$
    For each $k \in [2^l]$, we obtain $\Theta_k^{\mathrm{envi},l}$ by applying RRQR factorization to
        $f(\Theta_k^{\mathrm{in},l}; \Theta_{k+1}^{\mathrm{in},l} \times \Theta_{(k+1)/2}^{\mathrm{envi},l-1})$    (22)
    or
        $f(\Theta_k^{\mathrm{in},l}; \Theta_{k-1}^{\mathrm{in},l} \times \Theta_{k/2}^{\mathrm{envi},l-1})$    (23)
    for odd or even $k$ respectively to obtain $s$ important columns. The multi-indices corresponding to
    these $s$ columns are used to update $\Theta_k^{\mathrm{envi},l}$. The selection of the environment skeletons when $k$ is odd
    is illustrated in the next figure.
```



```
end for
```

Finally, to meet the second criteria that the cost in (13) should approximate the cost in (8), to each Ω_k^{envi}, we add extra samples $x \in [n]^{d-3}$ by sampling x_i's uniformly and independently from $[n]$. We typically sample an extra $3s$ samples to each Ω_k^{envi}. This completes the construction for Ω_k^{envi} 's and their corresponding Ω_k's in Algorithm 1.

2.2 Initialization

Due to the nonlinearity of the optimization problem (9), it is possible for ALS to get stuck at local minima or saddle points. A good initialization is crucial for the success of ALS. One possibility is to use the "opening" procedure in [5] to obtain each 3-tensors. As mentioned previously, this may suffer from the gauge ambiguity issue, leading us to consider a different approach. The proposed initialization procedure consists of two steps. First we obtain H^k's up to gauges G^k's between them (Algorithm 4). Then we solve d least-squares problem to fix the gauges between the H^k's (Algorithm 5). More precisely, after Algorithm 4, we want to use $T^{k,C}$ as H^k. However, as in any factorization, SVD can only determine the factorization of $T^{k,C}$ up to gauge transformations, as shown in Figure 3. Therefore, between $T^{k,C}$ and $T^{k+1,C}$, some appropriate gauge G^k has to be inserted (Figure 3).

After gauge fixing, we complete the initialization step in Algorithm 1. Before moving on, we demonstrate the superiority of this initialization v.s. random initialization. In Figure 4 we plot the error between TR and the full function v.s. number of iterations in ALS, when using the proposed initialization and random initialization. By random initialization, we mean the H^k's are initialized by sampling their entries independently from the normal distribution. Then ALS is performed on the example detailed in Section 3.3 with $n = 3, d = 12$. We set the TR rank to be $r = 3$. As we can see, after one iteration of ALS, we already obtain 10^{-4} error using our proposed method, whereas with random initialization, the convergence of ALS is slower and the solution has a lower accuracy.

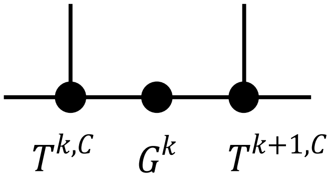


Figure 3. A gauge G^k needs to be inserted between $T^{k,C}$ and $T^{k+1,C}$

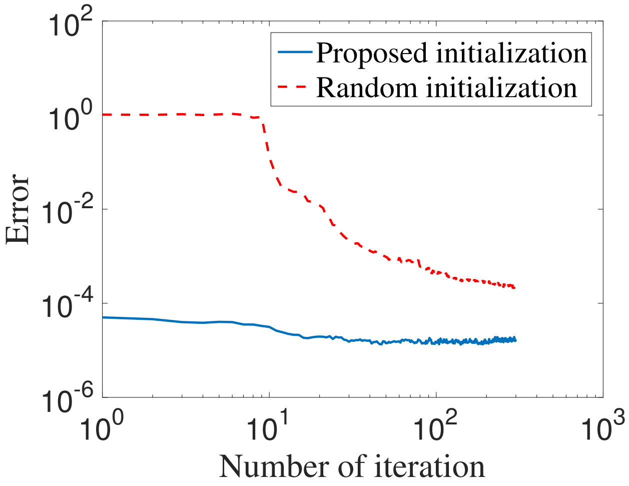
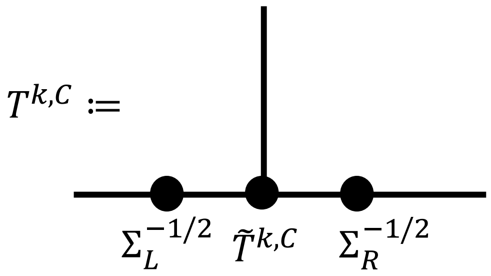


Figure 4. Plot of convergence of the ALS using both random and the proposed initializations for the numerical example given in Section 3.3 with $n = 3, d = 12$. The error measure is defined in (32).

Algorithm 4

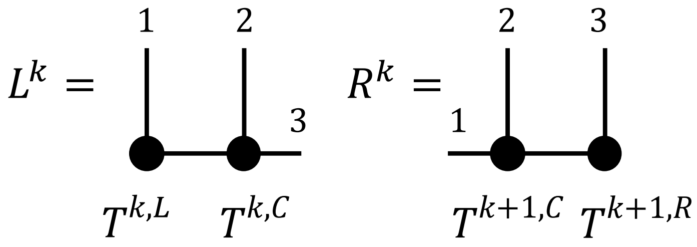
```
Require:
    Function $f : [n]^d \to \mathbb{R}$.
Ensure:
    $T^{k,L} \in \mathbb{R}^{n\times r}, T^{k,C} \in \mathbb{R}^{r\times n\times r}, T^{k,R} \in \mathbb{R}^{r\times n}$, $k \in [d]$.
    for $k = 1$ to $k = d$
        Pick an arbitrary $z \in [n]^{d-3}$ and let
            $$\Omega_k^{\text{ini}} := \big\{x \in [n]^d \mid x_{[d]\setminus\{k-1,k,k+1\}} = z, x_{k-1}, x_k, x_{k+1} \in [n]\big\}. \quad (24)$$
        Define
            $$T^k := f(\Omega_k^{\text{ini}}) \in \mathbb{R}^{n\times n\times n} \quad (25)$$
        where the first, second and third dimensions of $T^k$ correspond to the $(k-1), k, (k+1)$-th dimensions
        of $f$. Note that we only pick one $z$ in $\Omega_k^{\text{envi}}$, which is the key that we can use SVD procedure in the
        next step and avoid ambiguity in the initialization. The justification of such procedure can be found
        in Appendix A.
        Now we want to factorize the 3-tensor $T^k$ into a tensor train with three nodes using SVD. First treat
        $T^k$ as a matrix by treating the first leg as rows and the second and third legs as columns. Apply a
        rank-$r$ approximation to $T^k$ using SVD:
            $$T^k_{1;2,3} \approx U_L \Sigma_L V_L^T. \quad (26)$$
        Let $C^k \in \mathbb{R}^{r\times n\times n}$ be reshaped from $\Sigma_L V_L^T \in \mathbb{R}^{r\times n^2}$.
        Treat $C^k$ as a matrix by treating the first and second legs as rows and third leg as columns. Apply
        SVD to obtain a rank-$r$ approximation:
            $$C^k_{1,2;3} \approx U_R \Sigma_R V_R^T. \quad (27)$$
        Let $\tilde{T}^{k,C} \in \mathbb{R}^{r\times n\times r}$ be reshaped from $U_R\Sigma_R \in \mathbb{R}^{rn\times r}$.
        Let $T^{k,L} := U_L \Sigma_L^{1/2}$ and $T^{k,R} := \Sigma_R^{1/2} V_R^T$. Let $T^{k,C}$ be defined by
```



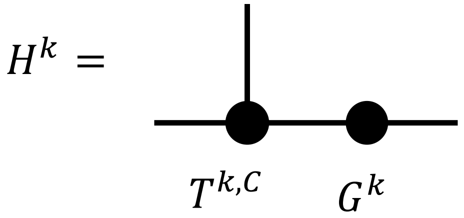
```
        3-tensor $T^k$ is thus approximated by a tensor train with three tensors $T^{k,L} \in \mathbb{R}^{n\times r}, T^{k,C} \in$
        $\mathbb{R}^{r\times n\times r}, T^{k,R} \in \mathbb{R}^{r\times n}$.
    end for
```

Algorithm 5

```
Require:
    Function $f : [n]^d \to \mathbb{R}$, $T^{k,L}, T^{k,C}, T^{k,R}$ for $k \in [d]$ from Algorithm 4.
Ensure:
    Initialization $H^k, k \in [d]$.
    for $k = 1$ to $k = d$
        Pick an arbitrary $z \in [n]^{d-4}$ and let
            $$\Omega_k^{\text{gauge}} := \left\{ x \in [n]^d \mid x_{[d] \setminus \{k-1,k,k+1,k+2\}} = z, \, \forall \, x_{k-1}, x_k, x_{k+1}, x_{k+2} \in [n] \right\} \quad (28)$$
        and sample
            $$S^k = f(\Omega_k^{\text{gauge}}) \in \mathbb{R}^{n \times n \times n \times n}. \quad (29)$$
        Solve the least-squares problem
            $$G^k = \underset{G}{\operatorname{argmin}} \, \| L^k_{1,2;3} G R^k_{1;2,3} - S^k_{1,2;3,4} \|_F^2 \quad (30)$$
        where $L^k$ and $R^k$ are defined as
```



```
        Obtain $H^k$:
```



```
    end for
```

2.3 Alternating least-squares

After constructing Ω_k and initializing H^k, $k \in [d]$, we start ALS by solving problem (13) at each iteration. This completes Algorithm 1.

When running ALS, sometimes we want to increase the TR-rank to obtain a higher accuracy approximation to the function f. In this case, we simply add a row and column of random entries to each H^k, i.e.

$$H^k(:, i, :) \leftarrow \begin{bmatrix} H^k(:, i, :) & \epsilon_1^{i,k} \\ \epsilon_2^{i,k} & 1 \end{bmatrix}, \quad i = 1, \ldots, n, \; k = 1, \ldots, d, \tag{31}$$

where each entry of $\epsilon_1^{i,k} \in \mathbb{R}^{r \times 1}, \epsilon_2^{i,k} \in \mathbb{R}^{1 \times r}$ is sampled from Gaussian distribution, and continue with the ALS procedure with the new H^k's until the error stops decreasing. The variance of each Gaussian random variable is typically set to 10^{-8}.

3 Numerical results

In this section, we present some results on the proposed method for tensor ring decomposition. We calculate the error between the obtained tensor ring decomposition and function f as:

$$E = \sqrt{\frac{\sum_{x\in\Omega} \big(\operatorname{Tr}(H^1[x_1]\cdots H^d[x_d]) - f(x_1,\ldots,x_d)\big)^2}{\sum_{x\in\Omega} f(x_1,\ldots,x_d)^2}}. \tag{32}$$

Whenever it is feasible, we let $\Omega = [n]^d$. If the dimensionality of f is large, we simply sample Ω from $[n]^d$ at random. For the proposed algorithm, we also measure the error on the entries sampled for learning TR as:

$$E_{\text{skeleton}} = \sqrt{\frac{\sum_{x\in\cup_k\Omega_k} \big(\operatorname{Tr}(H^1[x_1]\cdots H^d[x_d]) - f(x_1,\ldots,x_d)\big)^2}{\sum_{x\in\cup_k\Omega_k} f(x_1,\ldots,x_d)^2}}. \tag{33}$$

3.1 Example 1: A toy example

We first compress the function

$$f(x_1,\ldots,x_d) = \frac{1}{\sqrt{1+x_1^2+\ldots+x_d^2}}, \quad x_k \in [0,1] \tag{34}$$

considered in [5] into a tensor ring. In this example, we let $s = 14$ (recall that s is the size of Ω_k^{envi}). We compare our method with DMRG-Cross algorithm [16] (which gives a TT) and the SVD-based TR decomposition method proposed in [5]. As a method that is based on interpolative decomposition, DMRG-Cross is able to obtain high quality approximation if we allow a large TT-rank representation. Since we obtain the TR based on ALS optimization, the accuracy may not be comparable to DMRG-Cross. What we want to emphasize here is that if the given situation only requires moderate accuracy, our method could give a more economical representation than TT obtained from DMRG-Cross. To convey this message, we set the accuracy of DMRG-Cross so that it matches the accuracy of our proposed method, resulting TT-representations of lower rank. To compare with the algorithm in [5], we simply cite the results in [5] since the software is not publicly available. As expected, the TR-rank is lower than the TT-rank.

Setting	Format	Rank $(r_1,\ldots,r_d)$	E_{skeleton}	E	Run Time (s)
$d=6, n=10$	TR	(3,3,3,3,3,3)	2.0e-03	4.9e-04	4.8
	TT	(5,5,5,5,5,1)	-	1.2e-04	2.4
	TR[5]	(3,3,3,3,3,3)	-	2.3e-04	-
$d=6, n=20$	TR	(3,3,3,3,3,3)	5.1e-04	9.4e-05	24
	TT	(5,5,6,5,5,1)		6.8e-05	7.1
	TR[5]	(3,3,5,6,6,6)	-	1.8e-03	
$d=12, n=3$	TR	(3,3,3,3,3,3 3,3,3,3,3,3)	3.2e-04	2.8e-04	5.8
	TT	(5,5,6,6,5,5 5,5,5,3,3,1)	-	4.8e-04	4.5

Table 1. Results for Example 1. n corresponds to the number of uniform grid points on $[0,1]$ for each x_k. The tuple $(r_1,\ldots,r_d)$ indicates the rank of the learnt TR and TT. E_{skeleton} is computed on the samples used for learning the TR.

3.2 Example 2: Ising spin glass

We consider compressing the free energy of Ising spin glass with a ring geometry:

$$f(J_1,\ldots,J_d) = -\frac{1}{\beta}\log\left[\operatorname{Tr}\left(\prod_{i=1}^{d}\begin{bmatrix} e^{\beta J_i} & e^{-\beta J_i} \\ e^{-\beta J_i} & e^{\beta J_i}\end{bmatrix}\right)\right]. \tag{35}$$

We let $\beta = 10$, and $J_i \in \{-2.5, -1.5, 1, 2\}, i \in [d]$. This corresponds to Ising model with temperature of about 0.1K. In this example, we start with a tensor ring with $r = 4$ in ALS and increase the rank to $r = 5$ using the rank increasing heuristics detailed in Section 2.3 when the decrement in E_{skeleton} is small. The running time for this experiment is longer since after increasing the rank, ALS has to be used to further decrease the error. We let $s = 14$ when selecting the skeletons in the hierarchical sampling strategy. In this experiment, the accuracy of DMRG-Cross is set to 1e-03. When computing the error E for the case of $d = 24$, due to the size of f, we simply sub-sample 10^6 entries of f where J_i's are sampled independently and uniformly from $\{-2.5, -1.5, 1, 2\}$.

Setting	Format	Rank $(r_1, \ldots, r_d)$	E_{skeleton}	E	Run Time (s)
$d = 12, n = 4$	TR	(5,5,5,5,5,5 5,5,5,5,5,5)	3.1e-03	2.5e-03	16
	TT	(6,9,9,9,9,9 9,9,8,8,4,1)	-	1e-03	21
$d = 24, n = 4$	TR	(5,5,5,5,5,5 5,5,5,5,5,5 5,5,5,5,5,5 5,5,5,5,5,5)	1.7e-03	2.2e-03	56
	TT	(6,9,9,9,8,9 9,8,7,8,9,9 8,8,7,7,7,7 7,7,8,8,4,1)	-	1.7e-03	46

Table 2. Results for Example 2. Learning the free energy of Ising spin glass.

3.3 Example 3: Parametric elliptic partial differential equation (PDE)

In this section, we demonstrate the performance of our method in solving parametric PDE. We are interested in solving elliptic equation with random coefficients

$$\frac{\partial}{\partial x} a(x) \left(\frac{\partial}{\partial x} u(x) + 1 \right) = 0, \quad x \in [0, 1] \tag{36}$$

subject to periodic boundary condition, where $a(\cdot)$ is a random field. In particular, we want to parameterize the effective conductance function

$$A_{\text{eff}}(a(\cdot)) := \int_{[0,1]} a(x) \left(\frac{\partial}{\partial x} u(x) + 1 \right)^2 dx \tag{37}$$

as a TR. By discretizing the domain into d segments and assuming $a(x) = \sum_{i=1}^{d} a_i \chi_i(x)$, where each $a_i \in [1, 2, 3]$ and χ_i's being step functions on uniform intervals on $[0, 1]$, we determine $A_{\text{eff}}(a_1, \ldots, a_d)$ as a TR. In this case, the effective coefficients have an analytic solution

$$A_{\text{eff}}(a_1, \ldots, a_d) = \left(\frac{1}{d} \sum_{i=1}^{d} a_i \right)^{-1} \tag{38}$$

and we use this formula to generate samples to learn the TR. For this example, we pick $s = 14$. The results are reported in Table 3. When computing E with $d = 24$, again 10^6 entries of f are subsampled where a_i's are sampled independently and uniformly from $\{1, 2, 3\}$. We note that although in this situation, there is an analytic formula for the function we want to learn as a TR, we foresee further usages of our method when solving parametric PDE with periodic boundary condition, where there is no analytic formula for the physical quantity of interest (for example for the cases considered in [10]).

Setting	Format	Rank $(r_1,\ldots,r_d)$	E_{skeleton}	E	Run Time (s)
$d=12, n=3$	TR	(3,3,3,3,3,3 3,3,3,3,3,3)	2.0e-05	2.1e-05	6.1
	TT	(5,5,5,5,5,5 5,5,5,3,3,1)	-	2.5e-05	0.76
$d=24, n=3$	TR	(3,3,3,3,3,3 3,3,3,3,3,3 3,3,3,3,3,3 3,3,3,3,3,3)	1.8e-05	5.1e-05	12
	TT	(5,5,5,5,5,5 5,5,5,5,5,5 5,5,5,5,5,5 5,5,5,3,3,1)	-	1.7e-05	1.5

Table 3. Results for Example 3. Solving parametric elliptic PDE.

4 Conclusion

In this paper, we propose method for learning a TR representation based on ALS. Since the problem of determining a TR is a non-convex optimization problem, we propose an initialization strategy that helps the convergence of ALS. Furthermore, since using the entire tensor f in the ALS is infeasible, we propose an efficient hierarchical sampling method to identify the important samples. Our method provides a more economical representation of the tensor f than TT-format. As for future works, we plan to investigate the performance of the algorithms for quantum systems. One difficulty is that the Assumption 1 (Appendix A) for the proposed initialization procedure does not in general hold for quantum systems with short-range interactions. Instead, a natural assumption for a quantum state exhibiting a tensor-ring format representation is the exponential correlation decay [7, 2]. The design of efficient algorithms to determine the TR representation under such assumption is left for future works. Another natural direction is to extend the proposed method to tensor networks in higher spatial dimension, which we shall also explore in the future.

A Motivation of Algorithm 4

In this section, we motivate our initialization procedure. To this end, we place the following assumption on the TR f.

Assumption 1. *Let the TR f be partitioned into four disjoint regions (Fig 5): Regions a, b, c_1 and c_2 where $a, b, c_1, c_2 \subset [d]$. Regions a, b, c_1, c_2 contain $L_a, L_b, L_{c_1}, L_{c_2}$ number of dimensions respectively. If $L_a, L_b \geq L_{buffer}$, for any $z \in [n]^{L_a+L_b}$, the TR f satisfies*

$$f(x_{c_1}, x_{a\cup b}, x_{c_2})|_{x_{a\cup b}=z} \propto g(x_{c_1}, x_{a\cup b})|_{x_{a\cup b}=z} h(x_{a\cup b}, x_{c_2})|_{x_{a\cup b}=z} \tag{39}$$

for some functions g, h. Here "$\propto$" denotes the proportional up to a constant relationship.

We note that Assumption 1 holds if f is a non-negative function and admits a Markovian structure. Such functions can arise from Gibbs distribution with energy defined by short-range interactions [18], for example the Ising model.

Next we make certain non-degeneracy assumption on the TR f.

Assumption 2. *Any segment H of the TR f, as shown in Figure 6, satisfies*

$$\operatorname{rank}(H_{L+1,L+2;[L]}) = r^2 \tag{40}$$

if $L \geq L_0$ for some natural number L_0. In particular, if $L \geq L_0$, we assume the condition number of $H_{[L];L+1,L+2} \geq \kappa$ for some $\kappa = 1 + \delta\kappa$, where $\delta\kappa \geq 0$ is a small parameter.

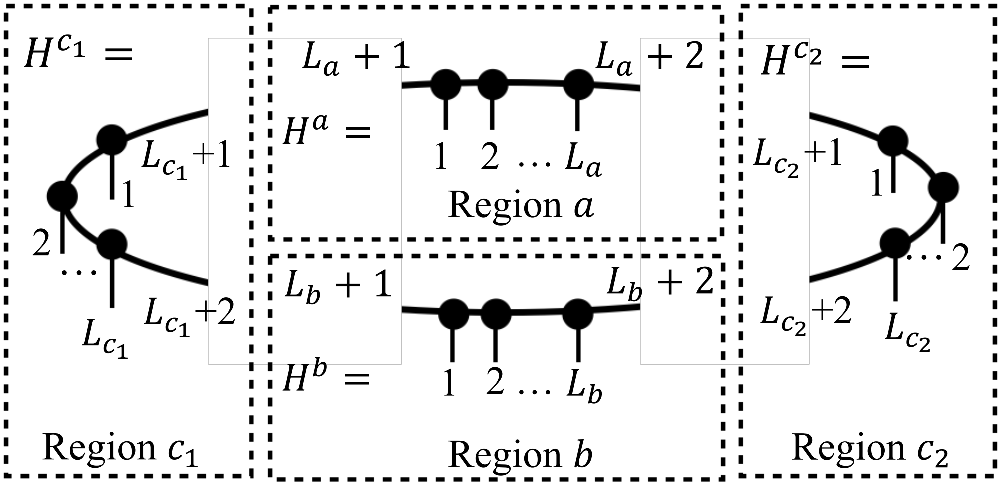


Figure 5. Figure of TR f partitioned into region a, b, c_1, c_2.

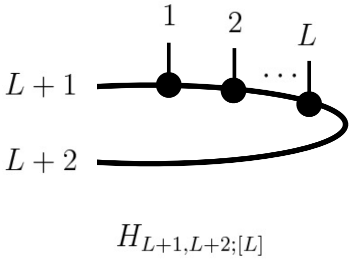


Figure 6. Figure of a segment of TR, denoted as H, with $L+2$ dimensions. The $1, \dots, L$-th dimensions have size n, corresponding to outgoing legs of the TR, and the $L+1, L+2$-th dimension are the latent dimensions with size r.

Since $H_{L+1,L+2;[L]} \in \mathbb{R}^{r^2 \times n^L}$, it is natural to expect when $n^L \geq r^2$, $H_{L+1,L+2;[L]}$ is rank r^2 generically [15].

We now state a proposition that leads us to the intuition behind designing the initialization procedure Algorithm 4.

Proposition 1. *Let*

$$s^1 = e_{i_1} \otimes e_{i_2} \otimes \cdots \otimes e_{i_{L_a}}, \quad s^2 = e_{j_1} \otimes e_{j_2} \otimes \cdots \otimes e_{j_{L_b}} \tag{41}$$

be any two arbitrary sampling vectors where $\{e_k\}_{k=1}^n$ is the canonical basis in $\mathbb{R}^n$. If $L_a, L_b, L_{c_1}, L_{c_2} \geq \max(L_0, L_{buffer})$, the two matrices $B^1, B^2 \in \mathbb{R}^{r \times r}$ defined in Figure 7 are rank-1.

Proof. Due to Assumption 2, $H^{c_1}_{L_{c_1}+1,L_{c_1}+2;[L_{c_1}]} \in \mathbb{R}^{r^2 \times n^{L_{c_1}}}$ and $H^{c_2}_{L_{c_2}+1,L_{c_2}+2;[L_{c_2}]} \in \mathbb{R}^{r^2 \times n^{L_{c_2}}}$ defined in Figure 7 are rank-r^2. Along with the implication of Assumption 1 that

$$\operatorname{rank}\big(\big(H^{c_1}_{L_{c_1}+1,L_{c_1}+2;[L_{c_1}]}\big)^T B^1 \otimes B^2 H^{c_2}_{L_{c_2}+1,L_{c_2}+2;[L_{c_2}]}\big) = 1, \tag{42}$$

we get

$$\operatorname{rank}(B^1 \otimes B^2) = 1. \tag{43}$$

Since $\operatorname{rank}(B^1)\operatorname{rank}(B^2) = \operatorname{rank}(B^1 \otimes B^2) = 1$, it follows that the rank of B^1, B^2 are 1. □

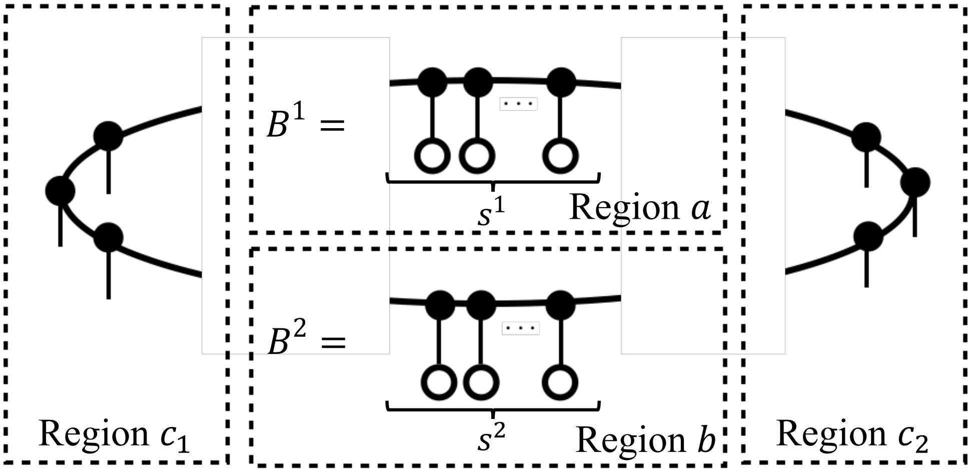


Figure 7. Definition of the matrices B^1, B^2 in Proposition 1.

The conclusion of Proposition 1 implies that to obtain the segment of TR in region a, one simply needs to apply some sampling vector s^2 in the canonical basis to region b to obtain the configuration in Figure 8 where the vectors $p^b, q^b \in \mathbb{R}^r$. Our goal is to extract the nodes in region a as H^k's. It is intuitively obvious that one can apply the TT-SVD technique in [13] to extract them. Such technique is indeed used in the proposed initialization procedure where we assume $L_{\text{buffer}} = 1, L_0 = 1, L_a = 1, L_b = d-3$. For completeness, in Proposition 2 we formalize the fact that one can use TT-SVD to learn each individual 3-tensor in the TR f up to some gauges. We further provide a perturbation analysis for the case when Markovian-type assumption holds only approximately in Proposition 2.

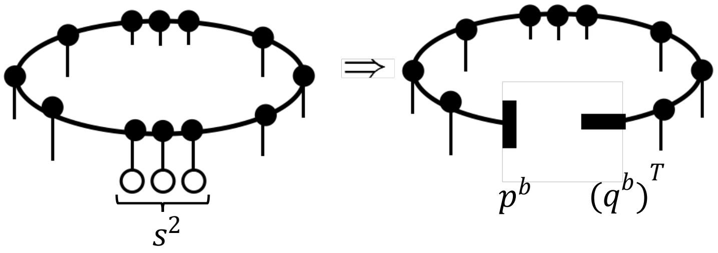


Figure 8. Applying a sampling vector s^2 in the canonical basis to region b gives the TT.

A.1 Stability of initialization

In this subsection, we analyze the stability of the proposed initialization procedure, where we relax Assumption 1 to approximate Markovianity.

Assumption 3. *Let*

$$\Omega_z := \left\{(x_{c_1}, x_{a\cup b}, x_{c_2}) \mid x_{c_1} \in [n]^{L_{c_1}}, x_{c_2} \in [n]^{L_{c_2}}, x_{a\cup b} = z\right\} \tag{44}$$

for some given $z \in [n]^{L_a+L_b}$. For any $z \in [n]^{L_a+L_b}$, we assume

$$\frac{\|f(\Omega_z)_{c_1;a\cup b\cup c_2}\|_2^2}{\|f(\Omega_z)_{c_1;a\cup b\cup c_2}\|_F^2} \geq \alpha. \tag{45}$$

for some $0 < \alpha \le 1$ *if* $L_a, L_b \ge L_{buffer}$.

This assumption is a relaxation of Assumption 1. Indeed, if (45) holds for $\alpha = 1$, it implies that $f(\Omega_z)_{c_1;a\cup b\cup c_2}$ is rank 1. Under the Assumption 3, we want to show that using Algorithm 4, one can extract H^k's approximately. The final result is stated in Proposition 2, obtained via the next few lemmas. In the first lemma, we show that B^1, B^2 defined in Figure 7 are approximately rank-1.

Lemma 1. *Let* $H^{c_1}, H^{c_2}, B^1, B^2$ *be defined according to Figure 5 and 7, where the sampling vectors* s^1, s^2 *are defined in Proposition 1. If* $L_{c_1}, L_{c_2}, L_a, L_b \ge \max(L_0, L_{buffer})$, *then*

$$\frac{\|B^1\|_2^2}{\|B^1\|_F^2}, \frac{\|B^2\|_2^2}{\|B^2\|_F^2} \ge \frac{\alpha}{\kappa^4}. \tag{46}$$

Proof. By Assumption 3,

$$\begin{aligned}\alpha &\le \frac{\big\|\big(H^{c_1}_{L_{c_1}+1,L_{c_1}+2;[L_{c_1}]}\big)^T B^1 \otimes B^2 H^{c_2}_{L_{c_2}+1,L_{c_2}+2;[L_{c_2}]}\big\|_2^2}{\big\|\big(H^{c_1}_{L_{c_1}+1,L_{c_1}+2;[L_{c_1}]}\big)^T B^1 \otimes B^2 H^{c_2}_{L_{c_2}+1,L_{c_2}+2;[L_{c_2}]}\big\|_F^2} \\ &\le \kappa_{c_1}^2 \kappa_{c_2}^2 \frac{\|B^1 \otimes B^2\|_2^2}{\|B^1 \otimes B^2\|_F^2} \\ &= \kappa_{c_1}^2 \kappa_{c_2}^2 \frac{\|B^1\|_2^2}{\|B^1\|_F^2} \frac{\|B^2\|_2^2}{\|B^2\|_F^2},\end{aligned} \tag{47}$$

where $\kappa_{c_1}, \kappa_{c_2} \le \kappa$ are condition numbers of $H^{c_1}_{L_{c_1}+1,L_{c_1}+2;[L_{c_1}]}$ and $H^{c_2}_{L_{c_2}+1,L_{c_2}+2;[L_{c_2}]}$ respectively. □

Let $p^b(q^b)^T$ be the best rank-1 approximation to B^2. Before registering the next corollary, we define $H^{[d]\backslash b}$ and $\tilde{H}^{[d]\backslash a}$ in Figure 9.

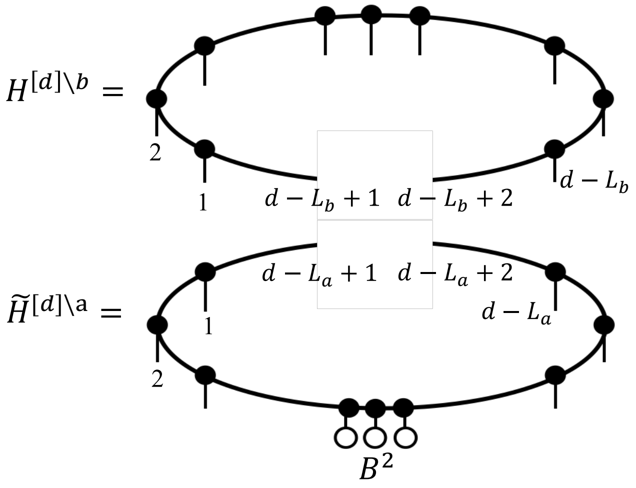


Figure 9. Definition of $H^{[d]\backslash b}$ and $\tilde{H}^{[d]\backslash a}$.

Corollary 1. *Under the assumptions of Lemma 1, for any sampling operator* s^2 *defined in Proposition 1,*

$$\frac{\|H^{[d]\backslash b}_{[d-L_b];d-L_b+1,d-L_b+2} vec(p^b(q^b)^T) - f_{[d]\backslash b;b} s^2\|_2^2}{\|f_{[d]\backslash b;b} s^2\|_F^2} \le \kappa^2(1 - \frac{\alpha}{\kappa^4}). \tag{48}$$

Proof. Lemma 1 implies

$$\frac{\|H^b_{L_b+1,L_b+2;[L_b]} s^2 - \mathrm{vec}(p^b(q^b)^T)\|_2^2}{\|H^b_{L_b+1,L_b+2;[L_b]} s^2\|_2^2}$$

$$= \frac{\|B^2 - p^b(q^b)^T\|_F^2}{\|B^2\|_F^2} = \frac{\|B^2\|_F^2 - \|p^b(q^b)^T\|_F^2}{\|B^2\|_F^2} \le 1 - \frac{\alpha}{\kappa^4}. \tag{49}$$

Then

$$\begin{aligned}
&\frac{\|H^{[d]\backslash b}_{[d-L_b];d-L_b+1,d-L_b+2}\mathrm{vec}(p^b(q^b)^T) - f_{[d]\backslash b;b}s^2\|_2^2}{\|f_{[d]\backslash b;b}s^2\|_2^2} \\
&\le \frac{\|H^{[d]\backslash b}_{[d-L_b];d-L_b+1,d-L_b+2}\|_2^2 \|H^b_{L_b+1,L_b+2;[L_b]}s^2 - \mathrm{vec}(p^b(q^b)^T)\|_2^2}{\|H^{[d]\backslash b}_{[d-L_b];d-L_b+1,d-L_b+2}H^b_{L_b+1,L_b+2;[L_b]}s^2\|_2^2} \\
&\le \kappa^2_{[d]\backslash b}\frac{\|H^b_{L_b+1,L_b+2;[L_b]}s^2 - \mathrm{vec}(p^b(q^b)^T)\|_2^2}{\|H^b_{L_b+1,L_b+2;[L_b]}s^2\|_2^2}
\end{aligned} \tag{50}$$

where $\kappa^2_{[d]\backslash b}$ is the condition number of $H^{[d]\backslash b}_{[d-L_b];d-L_b+1,d-L_b+2}$. Recall that H^b is defined in Figure 5. □

This corollary states that the situation in Figure 8 holds approximately. More precisely, let $T, \hat{T} \in \mathbb{R}^{n^{d-L_b}}$ be defined as

$$T := H^{[d]\backslash b}_{[d-L_b];d-L_b+1,d-L_b+2}\mathrm{vec}(p^b(q^b)^T), \ \hat{T} := f_{[d]\backslash b;b}s^2 \tag{51}$$

respectively, as demonstrated in Figure 10a, where p^b, q^b appear in Corollary 1. Corollary 1 implies

$$T = \hat{T} + E, \quad \frac{\|E\|_F^2}{\|\hat{T}\|_F^2} \le \kappa^2(1 - \frac{\alpha}{\kappa^4}). \tag{52}$$

In the following, we want to show that we can approximately extract the H^k's in region a. For this, we need to take the right-inverses of $\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]}$ and $\tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]}$, defined in Figure 10b. This requires a singular value lower bound, provided by the next lemma.

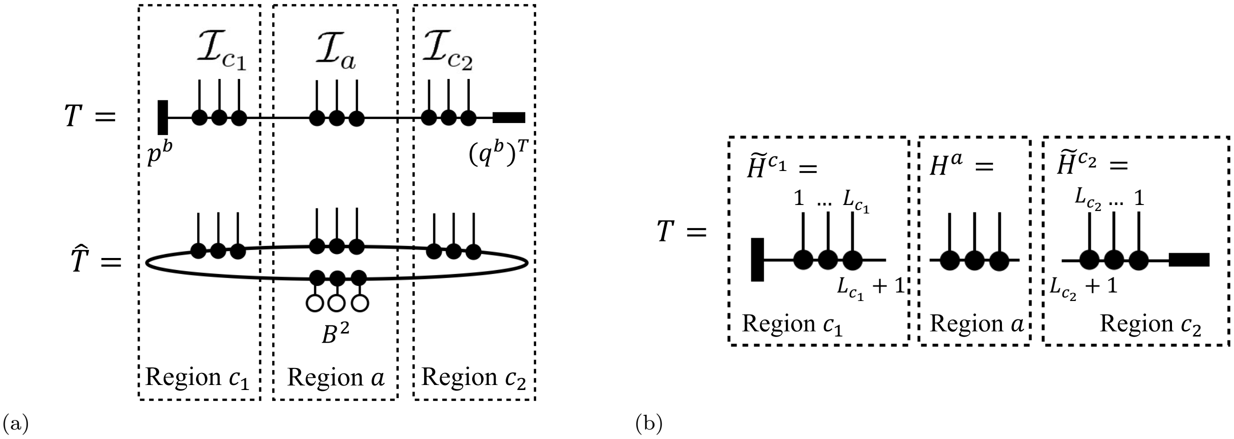


Figure 10. (a) Definition of T and $\hat{T}$. The dimensions in region a, c_1, c_2 are group into $\mathcal{I}_a, \mathcal{I}_{c_1}, \mathcal{I}_{c_2}$ respectively for the tensors T and $\hat{T}$. (b) Individual components of T.

Lemma 2. *Let $\sigma_k : \mathbb{R}^{m_1 \times m_2} \to \mathbb{R}$ be a function that extracts the $k-th$ singular value of a $m_1 \times m_2$ matrix. Then*

$$\frac{\sigma_r(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]})^2 \sigma_r(\tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^2}{\|\tilde{H}^{[d]\backslash a}_{d-L_a+1,d-L_a+2;[d-L_a]}\|_2^2} \ge \frac{1}{\kappa^6} - \frac{2\sqrt{r}}{\kappa^2}\sqrt{1 - \frac{\alpha}{\kappa^4}} \tag{53}$$

assuming

$$\frac{1}{\kappa^4} - 2\sqrt{r}\sqrt{1 - \frac{\alpha}{\kappa^4}} \ge 0. \tag{54}$$

Proof. Firstly,

$$\begin{aligned}
\frac{\sigma_{r^2}(T_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}})^2}{\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}\|_2^2} &\leq \frac{\|H^a_{[L_a];L_a+1,L_a+2}\|_2^2 \sigma_{r^2}(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]} \otimes \tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^2}{\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}\|_2^2} \\
&= \frac{\|H^a_{[L_a];L_a+1,L_a+2}\|_2^2 \sigma_r(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]})^2 \sigma_r(\tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^2}{\|H^a_{[L_a];L_a+1,L_a+2}\tilde{H}^{[d]\setminus a}_{d-L_a+1,d-L_a+2;[d-L_a]}\|_2^2} \\
&\leq \frac{\|H^a_{[L_a];L_a+1,L_a+2}\|_2^2 \sigma_r(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]})^2 \sigma_r(\tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^2}{\sigma_{r^2}(H^a_{[L_a];L_a+1,L_a+2})^2 \|\tilde{H}^{[d]\setminus a}_{d-L_a+1,d-L_a+2;[d-L_a]}\|_2^2} \\
&\leq \kappa^2 \frac{\sigma_r(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]})^2 \sigma_r(\tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^2}{\|\tilde{H}^{[d]\setminus a}_{d-L_a+1,d-L_a+2;[d-L_a]}\|_2^2} \, .
\end{aligned} \tag{55}$$

The equality follows from

$$\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}} = H^a_{[L_a];L_a+1,L_a+2}\tilde{H}^{[d]\setminus a}_{d-L_a+1,d-L_a+2;[d-L_a]},$$

which follows from (51), and the definition of $\tilde{H}^{[d]\setminus a}$ in Figure 9.

Observe that

$$\begin{aligned}
\frac{\sigma_{r^2}(T_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}})^2}{\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}\|_2^2} &\geq \frac{\sigma_{r^2}(\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}})^2 - 2\|E\|_F \sigma_{r^2}(\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}) + \|E\|_F^2}{\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}\|_2^2} \\
&\geq \frac{\sigma_{r^2}(\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}})^2}{\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}\|_2^2} - \frac{2\|E\|_F \sigma_{r^2}(\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}})}{\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}\|_2^2} \\
&\geq \frac{\sigma_{r^2}(H^a_{[L_a];L_a+1,L_a+2})^2 \sigma_{r^2}(\tilde{H}^{[d]\setminus a}_{d-L_a+1,d-L_a+2;[d-L_a]})^2}{\|H^a_{[L_a];L_a+1,L_a+2}\|_2^2 \|\tilde{H}^{[d]\setminus a}_{d-L_a+1,d-L_a+2;[d-L_a]}\|_2^2} \\
&\quad - \frac{2\|E\|_F \sigma_{r^2}(\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}})}{\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}\|_2^2} \\
&\geq \frac{1}{\kappa^4} - \frac{2\|E\|_F \sigma_{r^2}(\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}})}{\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}\|_2^2} \\
&\geq \frac{1}{\kappa^4} - \frac{2\sqrt{r}\sigma_{r^2}(\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}})\|E\|_F}{\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}\|_2 \|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}\|_F} \\
&\geq \frac{1}{\kappa^4} - 2\sqrt{r}\sqrt{1-\frac{\alpha}{\kappa^4}},
\end{aligned} \tag{56}$$

we established the claim. The first inequality regarding perturbation of singular values follows from theorem by Mirsky [11]:

$$\left|\sigma_{r^2}(T_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}) - \sigma_{r^2}(\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}})\right| \leq \|E\|_2 \leq \|E\|_F, \tag{57}$$

and assuming $\|E\|_F \leq \sigma_{r^2}(\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}})$. Such assumption holds when demanding the lower bound in (56) to be nonnegative, i.e.

$$\frac{\sigma_{r^2}(\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}})^2}{\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}\|_2^2} - \frac{2\|E\|_F \sigma_{r^2}(\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}})}{\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}\|_2^2} \geq \frac{1}{\kappa^4} - 2\sqrt{r}\sqrt{1-\frac{\alpha}{\kappa^4}} \geq 0 \tag{58}$$

The last inequality follows from Corollary 1. □

In the next lemma, we prove that when applying Algorithm 4 to $\hat{T}$, where $\hat{T}$ is treated as a 3-tensor formed from grouping the dimensions in each of set $\mathcal{I}_a, \mathcal{I}_{c_1}\mathcal{I}_{c_2}$, gives close approximation to $\hat{T}$.

Lemma 3. *Let*

$$\begin{aligned}
&\Pi_1 := \big\{Y \mid Y = XX^T, X \in \mathbb{R}^{n^{L_{c_1}}\times r},\ X^TX = I\big\},\\
&\Pi_2 := \big\{Y \mid Y = XX^T, X \in \mathbb{R}^{n^{L_{c_2}}\times r},\ X^TX = I\big\},
\end{aligned} \tag{59}$$

where I is the identity matrix. Let $P_1^* \in \Pi_1$ be the best rank-r projection for $\hat{T}_{\mathcal{I}_{c_2}\mathcal{I}_a;\mathcal{I}_{c_1}}$ such that $\hat{T}_{\mathcal{I}_{c_2}\mathcal{I}_a;\mathcal{I}_{c_1}}P_1^* \approx \hat{T}_{\mathcal{I}_{c_2}\mathcal{I}_a;\mathcal{I}_{c_1}}$ in Frobenius-norm, and

$$P_2^* = \min_{P_2\in\Pi_2} \|(\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I\otimes P_2) - \hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}})(P_1^*\otimes I)\|_F^2.$$

Then

$$\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I\otimes P_2^*)(P_1^*\otimes I) - \hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}\|_F^2 \le 2\|E\|_F^2. \tag{60}$$

Proof. To simplify the notations, let $\tilde{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}} := \hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I\otimes P_2)$. Then

$$\begin{aligned}
&\min_{P_2\in\Pi_2} \|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I\otimes P_2)(P_1^*\otimes I) - \hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}\|_F^2\\
&= \min_{P_2\in\Pi_2} \|(\tilde{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}} - \hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}} + \hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}})(P_1^*\otimes I) - \hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}\|_F^2\\
&= \min_{P_2\in\Pi_2} \|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I - P_1^*\otimes I)\|_F^2 + \|(\tilde{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}} - \hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}})(P_1^*\otimes I)\|_F^2\\
&\le \min_{P_2\in\Pi_2} \|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I - P_1^*\otimes I)\|_F^2 + \|\tilde{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}} - \hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}\|_F^2\\
&= \min_{P_2\in\Pi_2} \|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I - P_1^*\otimes I)\|_F^2 + \|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I - I\otimes P_2)\|_F^2.
\end{aligned} \tag{61}$$

The inequality comes from the fact that $P_1^*\otimes I$ is a projection matrix. Next,

$$\begin{aligned}
&\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I - P_1^*\otimes I)\|_F^2 + \min_{P_2\in\Pi_2} \|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I - I\otimes P_2)\|_F^2\\
&\qquad = \min_{P_1\in\Pi_1} \|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I - P_1\otimes I)\|_F^2 + \min_{P_2\in\Pi_2} \|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I - I\otimes P_2)\|_F^2\\
&\qquad \le \|E\|_F^2 + \|E\|_F^2 \le 2\|E\|_F^2,
\end{aligned} \tag{62}$$

and we can conclude the lemma. The equality comes from the definition of P_1^*, whereas the inequality is due to the facts that P_1, P_2 are rank-r projectors, and there exists T such that $\hat{T} = T - E$ where $\operatorname{rank}(T_{\mathcal{I}_{c_1}\mathcal{I}_a;\mathcal{I}_{c_2}})$, $\operatorname{rank}(T_{\mathcal{I}_{c_1};\mathcal{I}_a\mathcal{I}_{c_2}}) \le r$. □

We are ready to state the final proposition.

Proposition 2. *Let*

$$\hat{\hat{T}}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}} := \hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I\otimes P_2^*)(P_1^*\otimes I) \tag{63}$$

where P_1^*, P_2^* are defined in Lemma 3. Then

$$\frac{\|H^a_{[L_a];L_a+1,L_a+2} - \hat{\hat{T}}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]}\otimes \tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^\dagger\|_F^2}{\|H^a_{[L_a];L_a+1,L_a+2}\|_F^2} \le \frac{(1+\sqrt{2})^2\kappa^4(1-\frac{\alpha}{\kappa^4})}{\frac{1}{\kappa^4} - 2\sqrt{r}\sqrt{1-\frac{\alpha}{\kappa^4}}}, \tag{64}$$

where "†" is used to denote the pseudo-inverse of a matrix, if the upper bound is positive. When $\kappa = 1+\delta\kappa$ and $\alpha = 1-\delta\alpha$ where $\delta\kappa, \delta\alpha \ge 0$ are small parameters, we have

$$\frac{\|H^a_{[L_a];L_a+1,L_a+2} - \hat{\hat{T}}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]}\otimes \tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^\dagger\|_F^2}{\|H^a_{[L_a];L_a+1,L_a+2}\|_F^2} \le O(\delta\alpha + 4\delta\kappa). \tag{65}$$

Proof. From Lemma 3 and (52), we get

$$\begin{aligned}
&\|\hat{\hat{T}}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}} - T_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}\|_F \\
=&\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I \otimes P_2^*)(P_1^* \otimes I) - T_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}\|_F \\
\leq&\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I \otimes P_2^*)(P_1^* \otimes I) - \hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}\|_F + \|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}} - T_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}\|_F \\
\leq&(1+\sqrt{2})\|E\|_F.
\end{aligned} \tag{66}$$

Recall that

$$H^a_{[L_a];L_a+1,L_a+2} = T_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]} \otimes \tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^\dagger, \tag{67}$$

where the existence of a full-rank pseudo-inverse is guaranteed by the singular value lower bound in Lemma 2, we have

$$\begin{aligned}
&\frac{\|H^a_{[L_a];L_a+1,L_a+2} - \hat{\hat{T}}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]} \otimes \tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^\dagger\|_F^2}{\|H^a_{[L_a];L_a+1,L_a+2}\|_F^2} \\
\leq&\frac{(1+\sqrt{2})^2\|E\|_F^2\|(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]} \otimes \tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^\dagger\|_2^2}{\|H^a_{[L_a];L_a+1,L_a+2}\|_F^2} \\
\leq&\frac{(1+\sqrt{2})^2\|E\|_F^2}{\sigma_r(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]})^2\sigma_r(\tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^2\|H^a_{[L_a];L_a+1,L_a+2}\|_F^2} \\
=&\frac{(1+\sqrt{2})^2\|\hat{T}\|_F^2}{\sigma_r(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]})^2\sigma_r(\tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^2\|H^a_{[L_a];L_a+1,L_a+2}\|_F^2}\frac{\|E\|_F^2}{\|\hat{T}\|_F^2} \\
\leq&\frac{(1+\sqrt{2})^2\|\tilde{H}^{[d]\backslash a}_{d-L_a+1,d-L_a+2;[d-L_a]}\|_2^2}{\sigma_r(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]})^2\sigma_r(\tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^2}\frac{\|E\|_F^2}{\|\hat{T}\|_F^2} \\
\leq&\frac{(1+\sqrt{2})^2}{\frac{1}{\kappa^6} - \frac{2\sqrt{r}}{\kappa^2}\sqrt{1-\frac{\alpha}{\kappa^4}}}\kappa^2(1-\frac{\alpha}{\kappa^4}).
\end{aligned} \tag{68}$$

The first inequality follows from (66) and (67), and the last inequality follows from Corollary 1 and Lemma 2. □

When $L_a = L_{c_1} = L_{c_2} = 1$, applying Algorithm 4 to $\hat{T}$ results $\hat{\hat{T}}$ (represented by the tensors $T^{a,L}$,$T^{a,C}$ and $T^{a,R}$). Therefore, this proposition essentially implies $T^{a,C}$ approximates H^a up to gauge transformation.

References

[1] I. Affleck, T. Kennedy, E. H. Lieb, and H. Tasaki, *Valence bond ground states in isotropic quantum antiferromagnets*, Comm. Math. Phys. **115** (1988), 477–528.

[2] F. G. S. L. Brandao and M. Horodecki, *Exponential decay of correlations implies area law*, Commun. Math. Phys. **333** (2015), 761–798.

[3] Emmanuel J Candès and Benjamin Recht, *Exact matrix completion via convex optimization*, Foundations of Computational mathematics **9** (2009), no. 6, 717.

[4] Vin De Silva and Lek-Heng Lim, *Tensor rank and the ill-posedness of the best low-rank approximation problem*, SIAM Journal on Matrix Analysis and Applications **30** (2008), no. 3, 1084–1127.

[5] Mike Espig, Kishore Kumar Naraparaju, and Jan Schneider, *A note on tensor chain approximation*, Computing and Visualization in Science **15** (2012), no. 6, 331–344.

[6] Feliks Ruvimovich Gantmacher and Joel Lee Brenner, *Applications of the theory of matrices*, Courier Corporation, 2005.

[7] M. B. Hastings and T. Koma, *Spectral gap and exponential decay of correlations*, Commun. Math. Phys. **265** (2006), 781–804.

[8] Frank L Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, Studies in Applied Mathematics **6** (1927), no. 1-4, 164–189.

[9] Yoo Pyo Hong and C-T Pan, *Rank-revealing QR factorizations and the singular value decomposition*, Mathematics of Computation **58** (1992), no. 197, 213–232.

[10] Yuehaw Khoo, Jianfeng Lu, and Lexing Ying, *Solving parametric PDE problems with artificial neural networks*, 2017, preprint, arXiv:1707.03351.

[11] Leon Mirsky, *Symmetric gauge functions and unitarily invariant norms*, The quarterly journal of mathematics **11** (1960), no. 1, 50–59.

[12] R. Orus, *A practical introduction to tensor networks: Matrix product states and projected entangled pair states*, Ann. Phys. **349** (2013), 117–158.

[13] Ivan Oseledets and Eugene Tyrtyshnikov, *TT-Cross approximation for multidimensional arrays*, Linear Algebra and its Applications **432** (2010), no. 1, 70–88.

[14] Ivan V Oseledets, *Tensor-train decomposition*, SIAM Journal on Scientific Computing **33** (2011), no. 5, 2295–2317.

[15] David Perez-Garcia, Frank Verstraete, Michael M Wolf, and J Ignacio Cirac, *Matrix product state representations*, Quantum Inf. Comput. **7** (2007), 401.

[16] Dmitry Savostyanov and Ivan Oseledets, *Fast adaptive interpolation of multi-dimensional arrays in tensor train format*, Multidimensional (nD) Systems (nDs), 2011 7th International Workshop on, IEEE, 2011, pp. 1–8.

[17] Ulrich Schollwoeck, *The density-matrix renormalization group in the age of matrix product states*, Ann. Phys. **326** (2011), 96.

[18] M. M. Wolf, F. Verstraete, M. B. Hastings, and J. I. Cirac, *Area laws in quantum systems: Mutual information and correlations*, Phys. Rev. Lett. **100** (2008), 070502.

[19] Ming Yuan and Cun-Hui Zhang, *On tensor completion via nuclear norm minimization*, Foundations of Computational Mathematics **16** (2016), no. 4, 1031–1068.

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