MULTISCALE MODEL. SIMUL.
Vol. 19, No. 3, pp. 1261–1284

EFFICIENT CONSTRUCTION OF TENSOR RING REPRESENTATIONS FROM SAMPLING*

YUEHAW KHOO†, JIANFENG LU‡, AND LEXING YING§

Abstract. In this paper 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 an 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 fewer parameters than the tensor-train format and also better respects the structure of the original function.

Key words. tensor decompositions, tensor train, randomized algorithm, function approximation

AMS subject classifications. 65D15, 33F05, 15A69

DOI. 10.1137/17M1154382

1. Introduction.

Consider a function $f : [n]^d \to \mathbb{R}$ which can be treated as a tensor of size n^d ($[n] := \{1, \ldots, n\}$). In order to store and perform algebraic manipulation of the exponentially sized tensor, typically the tensor f has to be decomposed into various low complexity formats. Most current applications involve the CP [8] or Tucker decompositions [8, 17]. However, the CP decomposition for a general tensor is nonunique, whereas the components of a Tucker decomposition have exponential size in d. The tensor train (TT) [14], better known as the matrix product states (MPS) proposed earlier in the physics literature (see, e.g., [1, 19, 15]), emerges as an alternative that breaks the curse of dimensionality while avoiding the ill-posedness issue in tensor decomposition. For this format, function compression and evaluation can be done in $O(d)$ complexity. The situation is, however, unclear when generalizing a TT to a tensor network. Therefore, in this paper, we consider the compression of a black box function f into a *tensor ring* (TR), i.e., to find 3-tensors $H^1, \ldots, H^d$ such that for $x := (x_1, \ldots, x_d) \in [n]^d$

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

Here $H^k \in \mathbb{R}^{r_{k-1} \times n \times r_k}, r_k \leq r$ and we often refer to $(r_1, \ldots, r_d)$ as the TR rank. Such type of tensor format is a generalization of the TT format for which $H^1 \in \mathbb{R}^{1 \times n \times r_1}, H^d \in \mathbb{R}^{r_{d-1} \times n \times 1}$. The difference between TR and TT is illustrated in Figure 1 using tensor network diagrams introduced in section 1.1. Due to the exponential

*Received by the editors November 2, 2017; accepted for publication (in revised form) December 22, 2020; published electronically August 5, 2021.
https://doi.org/10.1137/17M1154382
Funding: The first and third authors were supported in part by the National Science Foundation under award DMS-1521830 and the U.S. Department of Energy's Advanced Scientific Computing Research program under award DE-FC02-13ER26134/DE-SC0009409. The second author is supported in part by the National Science Foundation under award DMS-1454939.
†Department of Statistics, The University of Chicago, Chicago, IL 60637 USA (ykhoo@uchicago.edu).
‡Departments of Physics, Chemistry, and Mathematics, Duke University, Durham, NC 27708 USA (jianfeng@math.duke.edu).
§Department of Mathematics, Stanford University, Stanford, CA 94305-2125 USA (lexing@stanford.edu).

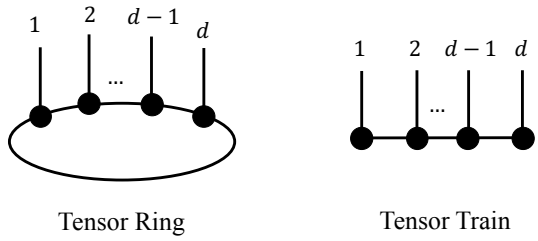


FIG. 1. *Comparison between a TR and a TT.*

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 $r_1 = r_2 = \cdots = r_d = r$.

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

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

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 $\mathrm{vec}(A)$ is used to denote the vectorization of a matrix A, formed by stacking the columns of A into a vector. For two sets α, β, we also use the notation

$$\alpha \setminus \beta := \{i \in \alpha \mid i \in \beta^c\} \tag{6}$$

to denote the set difference between α, β.

In this note, for the convenience of presentation, we use tensor network diagrams to represent tensors and contractions 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 2(a) shows a 3-tensor A and a 4-tensor B. When joining edges between two tensors (for example, in Figure 2(b) we join the third leg of A and first

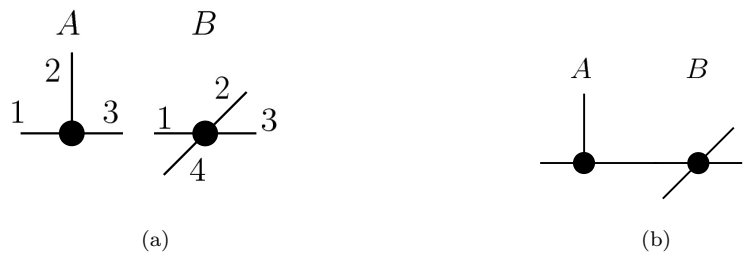


FIG. 2. (a) *Tensor diagram for a* 3*-tensor* A *and a* 4*-tensor* B. (b) *Contraction between tensors* A *and* B.

leg of B), we mean (with the implicit assumption that the dimensions represented by these legs have the same size)

$$\sum_k A_{i_1 i_2 k} B_{k j_2 j_3 j_4}. \tag{7}$$

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

1.2. Previous approaches. In this section, we survey previous approaches for compressing a blackbox function into TT or TR. In [13], successive CUR (skeleton) decompositions [6] are applied to find a decomposition of tensor f in TT format. In [4], a similar scheme is applied to find a TR decomposition of the tensor. A crucial step in [4] is to "disentangle" one of the 3-tensors H^k's, say H^1, from the TR. First, f is treated as a matrix where the first dimension of f gives rows, the second, third, $\ldots, d$th dimensions of f give columns, i.e., reshaping f to $f_{1;[d]\setminus 1}$. Then CUR decomposition is applied such that

$$f_{1;[d]\setminus 1} = CUR \tag{8}$$

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 [4], a shortcoming of the method lies in the reshaping of C into H^1. As in any factorization of a 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 TR rank in the subsequent determination of $H^2, H^3, \ldots, H^d$. More recently, [22] proposes various alternating least squares (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. Moreover, depending on the initialization, ALS can suffer from slow convergence. In [18], ALS is used to determine the TR in a more general setting where only partial observations of the function f are given. In this paper, we further assume the freedom to observe any $O(d)$ entries from the tensor f. As we shall see, leveraging such freedom, the complexity of the iterations can be reduced significantly compare to the ALS procedure in [18].

1.3. Our contributions. In this paper, 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 nonconvex hence requires good initialization in general. We devise a 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-tensor 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. Furthermore, the samples are chosen in a way that makes the per iteration complexity of the ALS linear in d.

While we focus in this note on the problem of construction of the TR 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 provide intuition and theoretical guarantess to motivate the proposed initialization procedure, based on certain probabilistic assumption on f. In section 4, we demonstrate the effectiveness of our methods through numerical examples. Finally we conclude the paper in section 5.

2. Proposed method. In order to find a TR decomposition (1), our overall strategy is to solve the minimization problem

$$\min_{H^1,\ldots,H^d} \sum_{x\in[n]^d} \big(\operatorname{Tr}(H^1[x_1]\cdots H^d[x_d]) - f(x_1,\ldots,x_d)\big)^2, \tag{9}$$

where

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

denotes the x_kth slice of the 3-tensor H^k along the second dimension. It is computationally infeasible just to set up problem (9), as we need to evaluate f n^d times. Therefore, analogously to the matrix or CP-tensor completion problem [3, 21], a "TR completion" problem [18]

$$\min_{H^1,\ldots,H^d} \sum_{x\in\Omega} \big(\operatorname{Tr}(H^1[x_1]\cdots H^d[x_d]) - f(x_1,\ldots,x_d)\big)^2, \tag{10}$$

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 (10) is via 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 (10). More precisely, to determine H^k, we solve

$$\min_{H^k} \sum_{x\in\Omega} \big(\operatorname{Tr}(H^k[x_k]C^{x\setminus x_k}) - f(x)\big)^2, \tag{11}$$

where each coefficient matrix

$$C^{x\setminus x_k} := H^{k+1}[x_{k+1}]\cdots H^d[x_d]H^1[x_1]\cdots H^{k-1}[x_{k-1}], \quad x\in\Omega. \tag{12}$$

By an abuse of notation, we use $x \setminus x_k$ to denote the exclusion of x_k from the d-tuple x. As mentioned previously, $|\Omega|$ should be at least $O(ndr^2)$ in order to determine the TR decomposition. This creates a large computational cost in each iteration of

the ALS, as it takes $|\Omega|(d-1)$ (which has $O(d^2)$ scaling as $|\Omega|$ has size $O(d)$) matrix multiplications just to construct $C^{x\setminus x_k}$ for all $x \in \Omega$. When d is large, such quadratic scaling in d for setting up the least-squares problem in each iteration of the ALS is undesirable.

The following simple but crucial observation allows us to gain a further speedup. Although $O(ndr^2)$ observations of f are required to determine all the components $H^1, \ldots, H^d$, when it comes to determining each individual H^k via solving the linear system (11), only $O(nr^2)$ equations are required for the well-posedness of the linear system. This motivates us to use different Ω_k's each having size $O(nr^2)$ (with $|\Omega_1| + \cdots + |\Omega_d| \sim O(ndr^2)$) to determine different H^k's in the ALS steps instead of using a fixed set Ω with size $O(ndr^2)$ for H^k's. If Ω_k is constructed from densely sampling the dimensions near k (where a 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

$$\mathrm{Tr}(H^k[x_k]C^{x\setminus x_k}) \approx f(x) \tag{13}$$

holds with small error for every $x \in [n]^d$, then using any $\Omega_k \in [n]^d$ in place of Ω in (11) should give similar solutions, as long as (11) is well-posed. Therefore, we solve

$$\min_{H^k} \sum_{x\in\Omega_k} \left(\mathrm{Tr}(H^k[x_k]C^{x\setminus x_k}) - f(x)\right)^2 \tag{14}$$

instead of (11) in each step of the ALS 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 (14) to reduce numerical instability resulting from a potential high condition number of the least-squares problem (14). In all of our experiments, λ is set to 10^{-9} and σ_k is the top singular value of the Hessian of the least-squares problem (14). From our experience, the quality of TR is rather insensitive to the choice of λ, which indicates the problem of determining H^k's is rather well-posed.

At this point it is clear that there are two issues needed to be addressed. The first issue is concerning the choice of $\Omega_k, k \in [d]$. Another issue is that the nonconvex nature of the TR completion problem 10 may cause difficulty in the convergence of ALS. We solve the first issue using a hierarchical sampling strategy. As for the second issue, by making certain probabilistic assumptions on f, we are able to obtain a cheap and intuitive initialization that allows fast convergence. Before moving on, we summarize the full algorithm in Algorithm 1. The steps of Algorithm 1 are further detailed in sections 2.1, 2.2, and 2.3.

Algorithm 1 Alternating least squares.

Require:
Function $f : [n]^d \to \mathbb{R}$.

Ensure:
TR $H^1, \ldots, H^d \in \mathbb{R}^{r\times n\times r}$.

1: Identify the index sets Ω_k's and compute $f(\Omega_k)$ for each $k \in [d]$ (section 2.1).
2: Initialize $H^1, \ldots, H^d$ (section 2.2).
3: Start ALS by solving (14) 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 $\Omega_k^{\text{envi}} \subset [n]^{d-3}$ with fixed size s. The elements in Ω_k^{envi} correspond to different choices of indices for the $[d]\backslash\{k-1, k, k+1\}$th dimensions of the function f. Then for each of the elements in Ω_k^{envi}, we sample all possible indices from the $(k-1)$th, kth, $(k+1)$th dimensions of f to construct Ω_k, i.e., letting

$$\Omega_k = [n]^3 \times \Omega_k^{\text{envi}}. \tag{15}$$

We let $|\Omega_k^{\text{envi}}| = s$ for all k where s is a constant that does not depend on the dimension d. In this case, when determining $C^{x\backslash x_k}, x \in \Omega_k$, in (14), only $O(|\Omega_k^{\text{envi}}|d)$ multiplications of $r \times r$ matrices are needed, giving a complexity that is linear in d when setting up the least-squares problem. We want to emphasize that although naively it seems that $O(n^3)$ samples are needed to construct Ω_k in (15), the n^3 samples corresponding to each sample in Ω_k^{envi} can be obtained via applying interpolative decomposition [5] to the $n \times n \times n$ tensor with $O(n)$ observations.

It remains that Ω_k^{envi}'s need to be constructed. There are two criteria we use for constructing $\Omega_k^{\text{envi}}, k \in [d]$. First, we want the range of $f_{k;[d]\backslash k}(\Omega_k)$ to be the same as the range of $f_{k;[d]\backslash k}$. This is a necessary condition of the least squares in (14) having a small residual. In this case, the following observation holds.

Observation 1. *If*

$$\sqrt{\sum_{x\in\Omega_k} \left(\text{Tr}(H^k[x_k]C^{x\backslash x_k}) - f(x)\right)^2} \le \epsilon, \tag{16}$$

then

$$\|H^k_{2;3,1} - f_{k;[d]\backslash k}(\Omega_k)[vec(C^{x\backslash x_k})]^{\dagger}_{x\in\Omega_k}\|_F \le \frac{\epsilon}{\sigma_{\min}([vec(C^{x\backslash x_k})]_{x\in\Omega_k})}, \tag{17}$$

where $\dagger$ denotes the pseudoinverse, and $\sigma_{\min}$ denotes the smallest singular value.

Therefore, $\text{Range}(H^k_{2;3,1})$ is similar to $\text{Range}(f_{k;[d]\backslash k}(\Omega_k))$. On the other hand, an optimal H^k should satisfy

$$H^k_{2;3,1}[\text{vec}(C^{x\backslash x_k})]_{x\in[n]^d} = f_{k;[d]\backslash k} \tag{18}$$

for all the entries of f, thus

$$\text{Range}(f_{k;[d]\backslash k}(\Omega_k)) \approx \text{Range}(f_{k;[d]\backslash k}). \tag{19}$$

Here we emphasize that it is possible to reshape $f(\Omega_k)$ into a matrix $f_{k;[d]\backslash k}(\Omega_k)$ as in (17) due to the product structure of Ω_k in (15), where the indices along dimension k are fully sampled. The second criterion is that we require the cost in (14) to approximate the cost in (9).

To meet the first criterion, 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 a 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 $l = 1, \ldots, L$. 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* $\Theta_k^{\text{in},l}$ which represent the kth group of indices, while the downward pass

hierarchically constructs representative environment skeletons $\Theta_k^{\text{envi},l}$. At each level, the skeletons are found by using rank revealing QR (RRQR) factorization [9].

After a full upward-downward pass where the 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 reinitiated. 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

$$\Omega_{3k-1}^{\text{envi}} := \Theta_k^{\text{envi}}, \quad k \in [2^L]. \tag{20}$$

Observe that we have only obtained Ω_k^{envi} for $k = 2, 5, \ldots, d-1$. Therefore, we need to apply the 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.

Finally, to meet the second criterion that the cost in (14) should approximate the cost in (9), 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 $5s$ samples to each Ω_k^{envi}. This completes the construction for Ω_k^{envi} 's and their corresponding Ω_k's in Algorithm 1.

Algorithm 2 Upward pass.

Require:

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

Ensure:

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

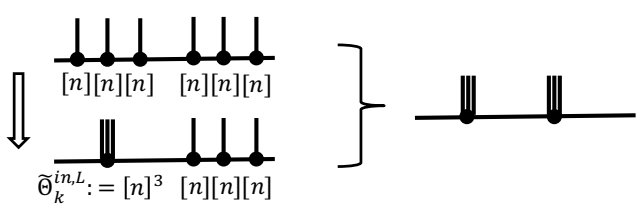
1: Decimate the number of dimensions by clustering every three dimensions. More precisely, for each $k \in [2^L]$, let

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

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

$$\Theta_k^{\text{envi},l} \subset [n]^{d-3} \tag{21}$$

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:

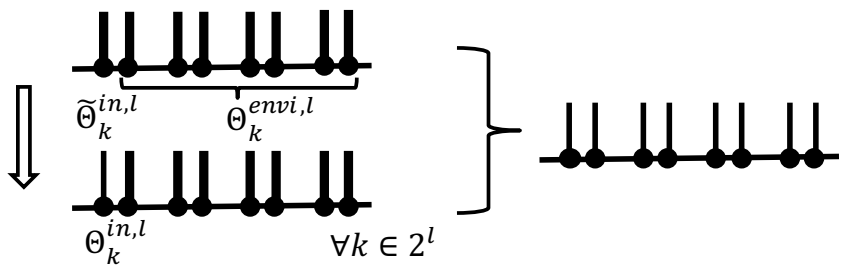


for $l = L$ to $l = 1$

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

$$f(\Theta_k^{\text{envi},l}; \tilde{\Theta}_k^{\text{in},l}) \in \mathbb{R}^{s \times |\tilde{\Theta}_k^{\text{in},l}|} \tag{22}$$

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.



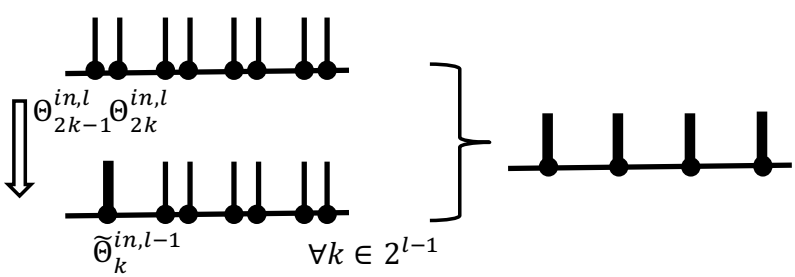
3: If $l > 1$, for each $k \in [2^{l-1}]$, construct

$$\tilde{\Theta}_k^{\text{in},l-1} := \Theta_{2k-1}^{\text{in},l} \times \Theta_{2k}^{\text{in},l}. \tag{23}$$

Then, sample s elements randomly from

$$\prod_{j \in [2^l] \setminus \{2k-1, 2k\}} \Theta_j^{\text{in},l} \tag{24}$$

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.



end for

Algorithm 3 Downward pass.

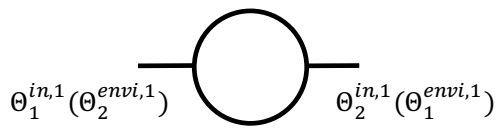
Require:

Function $f : [n]^d \to \mathbb{R}$, $\Theta_k^{\text{in},l}$'s from the upward pass, number of skeletons s.

Ensure:

Skeletons $\Theta_k^{\text{envi},l}$'s

Let $\Theta_1^{\text{envi},1} = \Theta^{\text{in},1}$, $\Theta^{\text{envi},1} = \Theta^{\text{in},1}$



for $l = 2$ to $l = L$

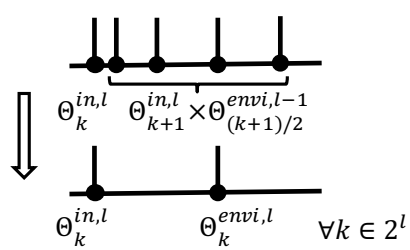
For each $k \in [2^l]$, we obtain $\Theta_k^{\text{envi},l}$ by applying RRQR factorization to

$$f(\Theta_k^{\text{in},l}; \Theta_{k+1}^{\text{in},l} \times \Theta_{(k+1)/2}^{\text{envi},l-1}) \tag{25}$$

or

$$f(\Theta_k^{\text{in},l}; \Theta_{k-1}^{\text{in},l} \times \Theta_{k/2}^{\text{envi},l-1}) \tag{26}$$

for odd or even k, respectively, to obtain s important columns. The multi-indices corresponding to these s columns are used to update $\Theta_k^{\text{envi},l}$. The selection of the environment skeletons when k is odd is illustrated in the next figure:



end for

2.2. Initialization. Due to the nonlinearity of the optimization problem (10), 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 [4] to obtain 3-tensors each. As mentioned previously, this may suffer an 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 problems 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 versus random initialization. In Figure 4 we plot the error between TR and the full function versus

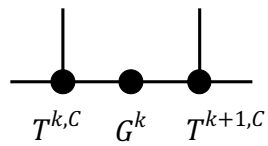


FIG. 3. *A gauge G^k needs to be inserted between $T^{k,C}$ and $T^{k+1,C}$.*

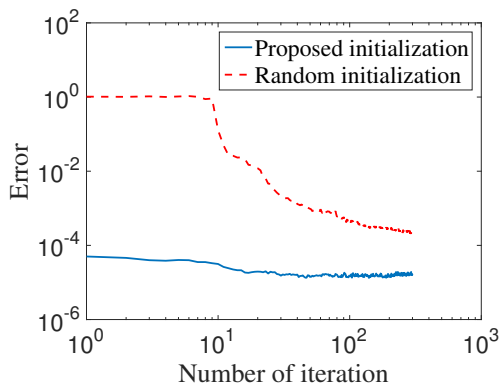


FIG. 4. *Plot of convergence of the ALS using both random and the proposed initializations for the numerical example given in section* 4.3 *with* $n = 3, d = 12$. *The error measure is defined in* (40).

the 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 4.3 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 a 10^{-4} error using our proposed method, whereas with random initialization, the convergence of ALS is slower and the solution has a lower accuracy.

2.3. Alternating least squares. After constructing Ω_k and initializing H^k, $k \in [d]$, we start ALS by solving problem (14) 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.,

$$H^k(:,i,:) \leftarrow \begin{bmatrix} H^k(:,i,:) & \epsilon_1^{i,k} \\ \epsilon_2^{i,k} & 1 \end{bmatrix}, \quad i = 1,\dots,n,\ k = 1,\dots,d, \tag{27}$$

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 a 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. Motivation of the initialization procedure. In this section, we motivate our initialization procedure in Algorithm 4. The main idea is by fixing a random index set, a portion of the ring can be singled out and extracted. To this end, we place the following assumption on the TR f.

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 $k = d$

1: Pick an arbitrary $z \in [n]^{d-3}$ and let

$$\Omega_k^{\text{ini}} := \left\{x \in [n]^d \mid x_{[d]\setminus\{k-1,k,k+1\}} = z, x_{k-1}, x_k, x_{k+1} \in [n]\right\}. \tag{28}$$

Define

$$T^k := f(\Omega_k^{\text{ini}}) \in \mathbb{R}^{n\times n\times n}, \tag{29}$$

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 Ω_k^{envi}, which is the key that we can use an SVD procedure in the next step and avoid ambiguity in the initialization. The justification of such a procedure can be found in Appendix 3.

2: Now we want to factorize the 3-tensor T^k into a TT 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:

$$T^k_{1;2,3} \approx U_L \Sigma_L V_L^T. \tag{30}$$

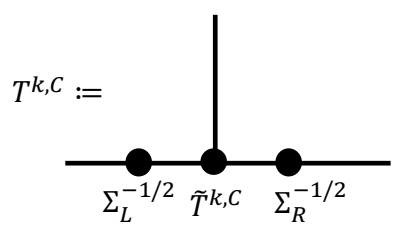
Let $C^k \in \mathbb{R}^{r\times n\times n}$ be reshaped from $\Sigma_L V_L^T \in \mathbb{R}^{r\times n^2}$.

3: Treat C^k as a matrix by treating the first and second legs as rows and the third leg as columns. Apply SVD to obtain a rank-r approximation:

$$C^k_{1,2;3} \approx U_R \Sigma_R V_R^T. \tag{31}$$

Let $\tilde{T}^{k,C} \in \mathbb{R}^{r\times n\times r}$ be reshaped from $U_R\Sigma_R \in \mathbb{R}^{rn\times r}$.

4: 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 TT 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

Assumption 1. Let the TR f be partitioned into four disjoint regions (Figure 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, where $L_a + L_b + L_{c_1} + L_{c_2} = d$. If $L_a, L_b \geq L_{\text{buffer}}$ for any $z \in [n]^{L_a+L_b}$, the TR f satisfies

$$f(x_{c_1}, x_{a\cup b}, x_{c_2})|_{x_{a\cup b}=z} \propto g(x_{c_1}, x_{a\cup b})|_{x_{a\cup b}=z} h(x_{a\cup b}, x_{c_2})|_{x_{a\cup b}=z} \tag{35}$$

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 $k = d$

1: Pick an arbitrary $z \in [n]^{d-4}$ and let

(32)
$$\Omega_k^{\text{gauge}} := \left\{ x \in [n]^d \mid x_{[d]\setminus\{k-1,k,k+1,k+2\}} = z, \forall\, x_{k-1}, x_k, x_{k+1}, x_{k+2} \in [n] \right\}$$

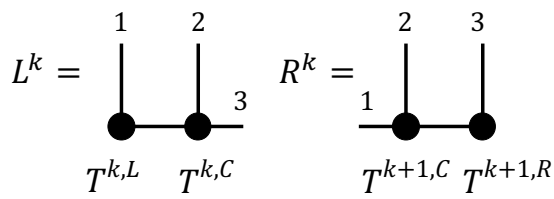
and sample

$$S^k = f(\Omega_k^{\text{gauge}}) \in \mathbb{R}^{n\times n\times n\times n}. \tag{33}$$

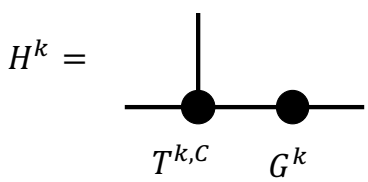
2: Solve the least-squares problem

$$G^k = \operatorname*{argmin}_G \|L^k_{1,2;3} G R^k_{1;2,3} - S^k_{1,2;3,4}\|_F^2 \tag{34}$$

where L^k and R^k are defined as



3: Obtain H^k:



end for

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 nonnegative function and admits a Markovian structure. Such functions can arise from a Gibbs distribution with energy defined by short-range interactions [20], 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 (for example $H^a, H^b, H^{c_1}, H^{c_2}$ shown in Figure 6), satisfies

$$\operatorname{rank}(H_{L+1,L+2;[L]}) = r^2 \tag{36}$$

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.

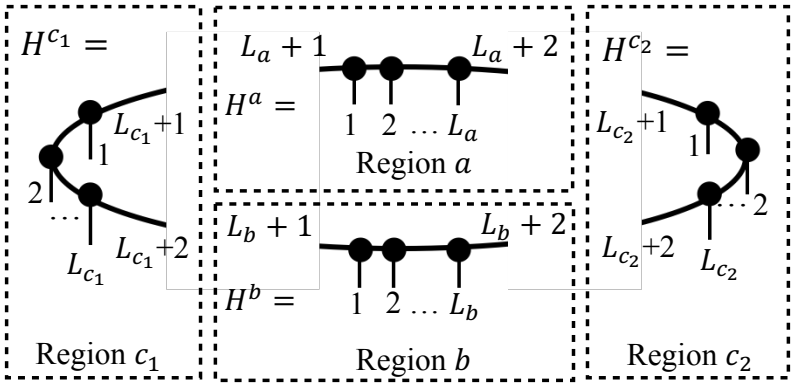


FIG. 5. *Figure of TR f partitioned into regions a, b, c_1, c_2.*

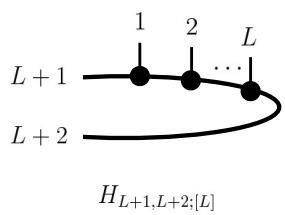


FIG. 6. *Figure of a segment of TR, denoted as H, with $L+2$ dimensions. The $1, \ldots, L$th dimensions have size n, corresponding to outgoing legs of the TR, and the $L+1, L+2$th dimensions are the latent dimensions with size r.*

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*

$$s^1 = e_{i_1} \otimes e_{i_2} \otimes \cdots \otimes e_{i_{L_a}}, \quad s^2 = e_{j_1} \otimes e_{j_2} \otimes \cdots \otimes e_{j_{L_b}} \tag{37}$$

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

$$\operatorname{rank}\big(\big(H^{c_1}_{L_{c_1}+1,L_{c_1}+2;[L_{c_1}]}\big)^T B^1 \otimes B^2 H^{c_2}_{L_{c_2}+1,L_{c_2}+2;[L_{c_2}]}\big) = 1, \tag{38}$$

we get

$$\operatorname{rank}(B^1 \otimes B^2) = 1. \tag{39}$$

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. ∎

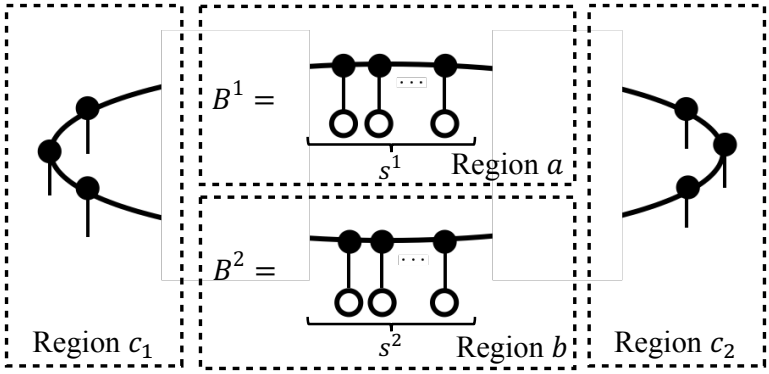


FIG. 7. *Definition of the matrices B^1, B^2 in Proposition* 1.

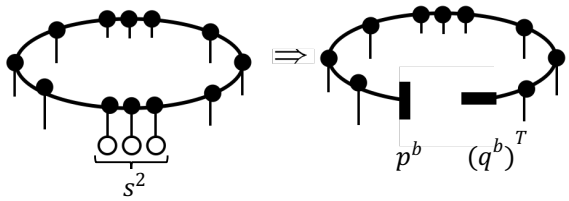


FIG. 8. *Applying a sampling vector s^2 in the canonical basis to region b gives the TT.*

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 a 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 in the appendix, 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 the Markovian-type assumption holds only approximately in Proposition 2.

4. Numerical results. In this section, we present numerical results on the proposed method for TR decomposition. We calculate the error between the obtained TR decomposition and function f as

$$E = \sqrt{\frac{\sum_{x\in\Omega}\left(\operatorname{Tr}(H^1[x_1]\cdots H^d[x_d]) - f(x_1,\ldots,x_d)\right)^2}{\sum_{x\in\Omega} f(x_1,\ldots,x_d)^2}}. \tag{40}$$

Whenever it is feasible, we let $\Omega = [n]^d$. Otherwise, we subsample Ω from $[n]^d$ at random: For every $x \in \Omega$, x_i is drawn from $[n]$ uniformly at random. 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

$$E_{\text{skeleton}} = \sqrt{\frac{\sum_{x\in\cup_k\Omega_k}\left(\operatorname{Tr}(H^1[x_1]\cdots H^d[x_d]) - f(x_1,\ldots,x_d)\right)^2}{\sum_{x\in\cup_k\Omega_k} f(x_1,\ldots,x_d)^2}}. \tag{41}$$

In the experiments, we compare our method, denoted as ITR-ALS ("I" stands for "initialized") with TR-ALS proposed in [18]. In [18], the cost in (9) is minimized using ALS where (11) is solved for each k in an alternating fashion. Although [18] proposed an SVD-based initialization approach similar to the recursive SVD algorithm for TT [13], this method has exponential complexity in d. Therefore the comparison with such an initialization is omitted and we use a randomized intialization for TR-ALS. As we shall see, ITR-ALS is generally an order of magnitude faster than TR-ALS, due to the special structure of the samples. For each experiment we run both TR-ALS and ITR-ALS five times and report the median accuracy. For TR-ALS, we often have to use fewer samples such that the running time is not excessively long (recall that TR-ALS has $O(d^2)$ complexity per iteration). To compare with the algorithm in [4], we simply cite the results in [4] since the software is not publicly available. We also compare ourselves with the density matrix renormalization group (DMRG)-cross algorithm [16] (which gives a TT). As a method that is based on interpolative decomposition, DMRG-cross is able to obtain a 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 TR-ALS.

4.1. Example 1: A toy example. We first compress the function

$$f(x_1,\ldots,x_d) = \frac{1}{\sqrt{1+x_1^2+\cdots+x_d^2}}, \quad x_k \in [0,1], \tag{42}$$

considered in [4] into a TR. The results are presented in Table 1. In this example, we let $s = 4$ (recall that s is the size of Ω_k^{envi}) in ITR-ALS. The number of samples we can afford to use for TR-ALS is less than ITR-ALS due to the excessively long running time since each iteration of TR-ALS has a complexity scaling of $O(d^2)$. In this example, although sometimes ITR-ALS has lower accuracy than TR-ALS, the running time of ITR-ALS is significantly shorter. In particular, for the case when $d = 12$, TR-ALS fails to converge using the same amount of samples as ITR-ALS. Both ITR-ALS and TR-ALS give TR with tensor components with smaller sizes than TT. The error E reported for the case of $d = 12$ is obtained from sampling 10^5 entires of the tensor f.

4.2. Example 2: Ising spin glass. In this example, we demosntrate the advantage of ITR-ALS in compressing a high-dimensional function arising from many-body physics, the traditional field where TT or MPS is extensively used [1, 19]. We consider compressing the free energy of Ising spin glass with a ring geometry:

$$f(J_1,\ldots,J_d) = -\frac{1}{\beta}\log\left[\operatorname{Tr}\left(\prod_{i=1}^{d}\begin{bmatrix} e^{\beta J_i} & e^{-\beta J_i} \\ e^{-\beta J_i} & e^{\beta J_i}\end{bmatrix}\right)\right]. \tag{43}$$

We let $\beta = 10$ and $J_i \in \{-2.5, -1.5, 1, 2\}, i \in [d]$. This corresponds to an Ising model with temperature of about 0.1K. The results are presented in Table 2. We let the number of environment samples $s = 5$. When computing the error E for the case

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 learned TR and TT. $E_{skeleton}$ is computed on the samples used for learning the TR.

Setting	Format	Rank $(r_1,\ldots,r_d)$	E_{skeleton}	E	$\frac{\text{Number of observations}}{n^d}$	Run time (s)
$d=6, n=10$	ITR-ALS	(3,3,3,3,3,3)	2.3e-03	6.3e-04	1.8e-01	4.7
	TR-ALS	(3,3,3,3,3,3)	4.3e-05	4.5e-05	2.8e-02	1360
	TT	(5,5,5,5,5,1)	-	1.2e-04	-	2.4
	TR[4]	(3,3,3,3,3,3)	-	2.3e-04	-	-
$d=6, n=20$	ITR-ALS	(3,3,3,3,3,3)	5.1e-04	9.4e-05	2.1e-02	24
	TR-ALS	(3,3,3,3,3,3)	5.0e-05	5.4e-05	8.2e-04	2757
	TT	(5,5,6,5,5,1)	-	6.8e-05	-	7.1
	TR[4]	(3,3,5,6,6,6)	-	1.8e-03	-	-
$d=12, n=5$	ITR-ALS	(3,3,3,3,3,3 3,3,3,3,3,3)	7.1e-04	5.9e-04	1.7e-04	28
	TR-ALS	(3,3,3,3,3,3 3,3,3,3,3,3)	0.97	0.97	1.7e-04	3132
	TT	(5,6,6,6,6,6 6,6,5,5,5,1)	-	2.2e-05	-	2.9

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

Setting	Format	Rank $(r_1,\ldots,r_d)$	E_{skeleton}	E	$\frac{\text{Number of observations}}{n^d}$	Run time (s)
$d=12, n=4$	ITR-ALS	(4,4,4,4,4,4 4,4,4,4,4,4)	3.9e-03	3.8e-03	1.6e-02	7
	TR-ALS	(4,4,4,4,4,4 4,4,4,4,4,4)	4.4e-02	5.2e-02	1.6e-02	994
	TT	(6,7,7,7,7,7 7,7,7,6,4,1)	-	4.2e-03	-	2.8
$d=24, n=4$	ITR-ALS	(3,3,3,3,3,3 3,3,3,3,3,3 3,3,3,3,3,3 3,3,3,3,3,3)	4.8e-03	2.7e-03	1.6e-10	19
	TR -ALS	-	-	-	1.6e-10	-
	TT	(6,8,8,8,6,6 6,6,6,6,7,6 5,6,6,6,6,7 7,6,6,6,4,1)	-	3.7e-03	-	9.3

of $d = 24$, due to the size of f, we simply subsample 10^5 entries of f, where J_i's are sampled independently and uniformly from $\{-2.5, -1.5, 1, 2\}$. For $d = 12$, the solution obtained by ITR-ALS is superior due to the initialization procedure. We see that in both $d = 12, 24$ cases, the running time of TR-ALS is much longer compare to ITR-ALS.

4.3. Example 3: Parametric elliptic partial differential equation (PDE). In this section, we demonstrate the performance of our method in solving a parametric PDE. We are interested in solving an elliptic equation with random coefficients

$$\frac{\partial}{\partial x} a(x)\left(\frac{\partial}{\partial x} u(x) + 1\right) = 0, \quad x \in [0,1], \tag{44}$$

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

$$A_{\text{eff}}(a(\cdot)) := \int_{[0,1]} a(x)\left(\frac{\partial}{\partial x} u(x) + 1\right)^2 dx \tag{45}$$

TABLE 3
Results for Example 3. Solving a parametric elliptic PDE.

Setting	Format	Rank $(r_1, \ldots, r_d)$	E_{skeleton}	E	$\frac{\text{Number of observations}}{n^d}$	Run time (s)
$d = 12, n = 3$	ITR-ALS	(3,3,3,3,3,3 3,3,3,3,3,3)	1.1e-05	1.1e-05	1.4e-02	22
	TR-ALS	(3,3,3,3,3,3 3,3,3,3,3,3)	5.7e-06	6.8e-06	1.4e-02	1414
	TT	(5,5,5,5,5,5 5,5,5,3,3,1)	-	2.5e-05	-	0.76
$d = 24, n = 3$	ITR-ALS	(3,3,3,3,3,3 3,3,3,3,3,3 3,3,3,3,3,3 3,3,3,3,3,3)	2.6e-05	2.8e-05	5.5e-06	47
	TR-ALS	-	-	-	5.5e-06	-
	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

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

$$A_{\text{eff}}(a_1, \ldots, a_d) = \left(\frac{1}{d} \sum_{i=1}^{d} a_i \right)^{-1} \tag{46}$$

and we use this formula to generate samples to learn the TR. For this example, we pick $s = 4$. The results are reported in Table 3. When computing E with $d = 24$, again 10^5 entries of f are subsampled, where the 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 usage of our method when solving parametric PDEs with periodic boundary conditions, where there is no analytic formula for the physical quantity of interest (for example for the cases considered in [10]).

5. Conclusion. In this paper, we propose a method for learning a TR representation based on ALS. Since the problem of determining a TR is a nonconvex 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 the 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 3) 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 TR format representation is the exponential correlation decay [7, 2]. The design of efficient algorithms to determine the TR representation under such an assumption is left for future works. Another natural direction is to extend the proposed method to tensor networks in higher spatial dimensions, which we shall also explore in the future.

Appendix A. Stability of initialization. In this section, we analyze the stability of the proposed initialization procedure, where we relax Assumption 1 to approximate Markovianity.

Assumption 3. Let

$$\Omega_z := \left\{(x_{c_1}, x_{a\cup b}, x_{c_2}) \mid x_{c_1} \in [n]^{L_{c_1}}, x_{c_2} \in [n]^{L_{c_2}}, x_{a\cup b} = z\right\} \tag{47}$$

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

$$\frac{\|f(\Omega_z)_{c_1;a\cup b\cup c_2}\|_2^2}{\|f(\Omega_z)_{c_1;a\cup b\cup c_2}\|_F^2} \geq \alpha \tag{48}$$

for some $0 < \alpha \leq 1$ if $L_a, L_b \geq L_{\text{buffer}}$.

This assumption is a relaxation of Assumption 1. Indeed, if (48) holds for $\alpha = 1$, it implies that $f(\Omega_z)_{c_1;a\cup b\cup c_2}$ is rank 1. Under 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 particular, we show that when the condition number κ of the TR components (defined in Lemma 1) satisfies $\kappa = 1$, as $\alpha \to 1$, the approximation error goes to 0. 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 Figures* 5 *and* 7*, where the sampling vectors s^1, s^2 are defined in Proposition* 1*. If $L_{c_1}, L_{c_2}, L_a, L_b \geq \max(L_0, L_{buffer})$, then*

$$\frac{\|B^1\|_2^2}{\|B^1\|_F^2}, \frac{\|B^2\|_2^2}{\|B^2\|_F^2} \geq \frac{\alpha}{\kappa^4}. \tag{49}$$

Proof. By Assumption 3,

$$\begin{aligned}
\alpha &\leq \frac{\left\|\left(H^{c_1}_{L_{c_1}+1,L_{c_1}+2;[L_{c_1}]}\right)^T B^1 \otimes B^2 H^{c_2}_{L_{c_2}+1,L_{c_2}+2;[L_{c_2}]}\right\|_2^2}{\left\|\left(H^{c_1}_{L_{c_1}+1,L_{c_1}+2;[L_{c_1}]}\right)^T B^1 \otimes B^2 H^{c_2}_{L_{c_2}+1,L_{c_2}+2;[L_{c_2}]}\right\|_F^2} \\
&\leq \kappa_{c_1}^2 \kappa_{c_2}^2 \frac{\|B^1 \otimes B^2\|_2^2}{\|B^1 \otimes B^2\|_F^2} \\
&= \kappa_{c_1}^2 \kappa_{c_2}^2 \frac{\|B^1\|_2^2}{\|B^1\|_F^2} \frac{\|B^2\|_2^2}{\|B^2\|_F^2},
\end{aligned} \tag{50}$$

where $\kappa_{c_1}, \kappa_{c_2} \leq \kappa$ 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 $H^{[d]\setminus b}$ and $\tilde{H}^{[d]\setminus a}$ in Figure 9.

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

$$\frac{\|H^{[d]\setminus b}_{[d-L_b];d-L_b+1,d-L_b+2} vec(p^b(q^b)^T) - f_{[d]\setminus b;b} s^2\|_2^2}{\|f_{[d]\setminus b;b} s^2\|_F^2} \leq \kappa^2 \left(1 - \frac{\alpha}{\kappa^4}\right). \tag{51}$$

Proof. Lemma 1 implies

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

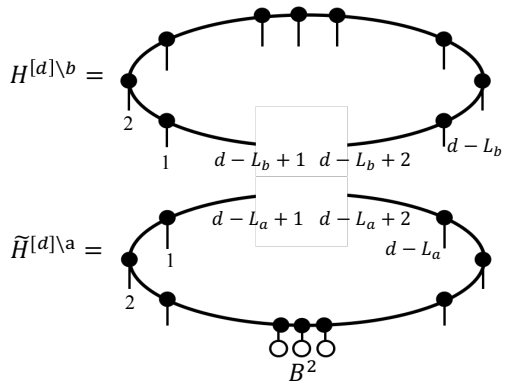


FIG. 9. *Definition of $H^{[d]\setminus b}$ and $\tilde{H}^{[d]\setminus a}$.*

$$= \frac{\|B^2 - p^b(q^b)^T\|_F^2}{\|B^2\|_F^2} = \frac{\|B^2\|_F^2 - \|p^b(q^b)^T\|_F^2}{\|B^2\|_F^2} \leq 1 - \frac{\alpha}{\kappa^4}. \tag{52}$$

Then

$$\begin{aligned}
&\frac{\|H^{[d]\setminus b}_{[d-L_b];d-L_b+1,d-L_b+2}\mathrm{vec}(p^b(q^b)^T) - f_{[d]\setminus b;b}s^2\|_2^2}{\|f_{[d]\setminus b;b}s^2\|_2^2} \\
&\quad\leq \frac{\|H^{[d]\setminus b}_{[d-L_b];d-L_b+1,d-L_b+2}\|_2^2\|H^b_{L_b+1,L_b+2;[L_b]}s^2 - \mathrm{vec}(p^b(q^b)^T)\|_2^2}{\|H^{[d]\setminus b}_{[d-L_b];d-L_b+1,d-L_b+2}H^b_{L_b+1,L_b+2;[L_b]}s^2\|_2^2} \\
&\quad\leq \kappa^2_{[d]\setminus b}\frac{\|H^b_{L_b+1,L_b+2;[L_b]}s^2 - \mathrm{vec}(p^b(q^b)^T)\|_2^2}{\|H^b_{L_b+1,L_b+2;[L_b]}s^2\|_2^2}, \qquad (53)
\end{aligned}$$

where $\kappa^2_{[d]\setminus b}$ is the condition number of $H^{[d]\setminus b}_{[d-L_b];d-L_b+1,d-L_b+2}$. 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

$$T := H^{[d]\setminus b}_{[d-L_b];d-L_b+1,d-L_b+2}\mathrm{vec}(p^b(q^b)^T),\ \hat{T} := f_{[d]\setminus b;b}s^2, \tag{54}$$

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

$$T = \hat{T} + E, \quad \frac{\|E\|_F^2}{\|\hat{T}\|_F^2} \leq \kappa^2\left(1 - \frac{\alpha}{\kappa^4}\right). \tag{55}$$

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 10(b). This requires a singular value lower bound, provided by the next lemma.

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

$$\frac{\sigma_r(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]})^2\sigma_r(\tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^2}{\|\tilde{H}^{[d]\setminus a}_{d-L_a+1,d-L_a+2;[d-L_a]}\|_2^2} \geq \frac{1}{\kappa^6} - \frac{2\sqrt{r}}{\kappa^2}\sqrt{1 - \frac{\alpha}{\kappa^4}} \tag{56}$$

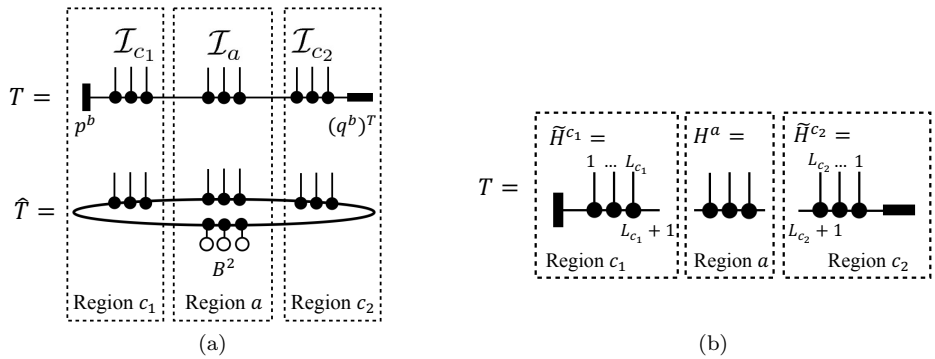


FIG. 10. (a) *Definition of T and $\hat{T}$. The dimensions in region a, c_1, c_2 are grouped into $\mathcal{I}_a, \mathcal{I}_{c_1}, \mathcal{I}_{c_2}$, respectively, for the tensors T and $\hat{T}$.* (b) *Individual components of T.*

assuming

$$\frac{1}{\kappa^4} - 2\sqrt{r}\sqrt{1 - \frac{\alpha}{\kappa^4}} \geq 0. \tag{57}$$

Proof. First,

$$\begin{aligned}
\frac{\sigma_{r^2}(T_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}})^2}{\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}\|_2^2} &\leq \frac{\|H^a_{[L_a];L_a+1,L_a+2}\|_2^2 \sigma_{r^2}(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]} \otimes \tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^2}{\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}\|_2^2} \\
&= \frac{\|H^a_{[L_a];L_a+1,L_a+2}\|_2^2 \sigma_r(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]})^2 \sigma_r(\tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^2}{\|H^a_{[L_a];L_a+1,L_a+2}\tilde{H}^{[d]\backslash a}_{d-L_a+1,d-L_a+2;[d-L_a]}\|_2^2} \\
&\leq \frac{\|H^a_{[L_a];L_a+1,L_a+2}\|_2^2 \sigma_r(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]})^2 \sigma_r(\tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^2}{\sigma_{r^2}(H^a_{[L_a];L_a+1,L_a+2})^2 \|\tilde{H}^{[d]\backslash a}_{d-L_a+1,d-L_a+2;[d-L_a]}\|_2^2} \\
&\leq \kappa^2 \frac{\sigma_r(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]})^2 \sigma_r(\tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^2}{\|\tilde{H}^{[d]\backslash a}_{d-L_a+1,d-L_a+2;[d-L_a]}\|_2^2}.
\end{aligned} \tag{58}$$

The equality follows from

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

which follows from (54), and the definition of $\tilde{H}^{[d]\backslash a}$ in Figure 9.

Observe that

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

$$
\begin{aligned}
&- \frac{2\|E\|_F \sigma_{r^2}(\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}})}{\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}\|_2^2} \\
&\geq \frac{1}{\kappa^4} - \frac{2\|E\|_F \sigma_{r^2}(\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}})}{\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}\|_2^2} \\
&\geq \frac{1}{\kappa^4} - \frac{2\sqrt{r}\sigma_{r^2}(\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}})\|E\|_F}{\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}\|_2 \|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}\|_F} \\
&\geq \frac{1}{\kappa^4} - 2\sqrt{r}\sqrt{1 - \frac{\alpha}{\kappa^4}}; \qquad (59)
\end{aligned}
$$

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

$$
\left|\sigma_{r^2}(T_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}) - \sigma_{r^2}(\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}})\right| \leq \|E\|_2 \leq \|E\|_F, \qquad (60)
$$

and assuming $\|E\|_F \leq \sigma_{r^2}(\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}})$. Such an assumption holds when demanding the lower bound in (59) to be nonnegative, i.e.,

$$
\frac{\sigma_{r^2}(\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}})^2}{\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}\|_2^2} - \frac{2\|E\|_F \sigma_{r^2}(\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}})}{\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}\|_2^2} \geq \frac{1}{\kappa^4} - 2\sqrt{r}\sqrt{1 - \frac{\alpha}{\kappa^4}} \geq 0. \qquad (61)
$$

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 $\mathcal{I}_a, \mathcal{I}_{c_1}\mathcal{I}_{c_2}$, it gives a close approximation to $\hat{T}$.

LEMMA 3. *Let*

$$
\begin{aligned}
\Pi_1 &:= \left\{Y \mid Y = XX^T, X \in \mathbb{R}^{n^{L_{c_1}} \times r},\ X^T X = I\right\}, \\
\Pi_2 &:= \left\{Y \mid Y = XX^T, X \in \mathbb{R}^{n^{L_{c_2}} \times r},\ X^T X = I\right\}, \qquad (62)
\end{aligned}
$$

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 the 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

$$
\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I \otimes P_2^*)(P_1^* \otimes I) - \hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}\|_F^2 \leq 2\|E\|_F^2. \qquad (63)
$$

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

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

$$= \min_{P_2 \in \Pi_2} \|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I - P_1^* \otimes I)\|_F^2 + \|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I - I \otimes P_2)\|_F^2. \tag{64}$$

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

$$\begin{aligned} &\|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I - P_1^* \otimes I)\|_F^2 + \min_{P_2 \in \Pi_2} \|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I - I \otimes P_2)\|_F^2 \\ &\quad = \min_{P_1 \in \Pi_1} \|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I - P_1 \otimes I)\|_F^2 + \min_{P_2 \in \Pi_2} \|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I - I \otimes P_2)\|_F^2 \\ &\qquad\qquad \le \|E\|_F^2 + \|E\|_F^2 \le 2\|E\|_F^2, \end{aligned} \tag{65}$$

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*

$$\hat{\hat{T}}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}} := \hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I \otimes P_2^*)(P_1^* \otimes I), \tag{66}$$

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

$$\frac{\|H^a_{[L_a];L_a+1,L_a+2} - \hat{\hat{T}}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]} \otimes \tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^\dagger\|_F^2}{\|H^a_{[L_a];L_a+1,L_a+2}\|_F^2} \le \frac{(1+\sqrt{2})^2\kappa^4(1-\frac{\alpha}{\kappa^4})}{\frac{1}{\kappa^4} - 2\sqrt{r}\sqrt{1-\frac{\alpha}{\kappa^4}}}, \tag{67}$$

where "†" is used to denote the pseudoinverse 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

$$\frac{\|H^a_{[L_a];L_a+1,L_a+2} - \hat{\hat{T}}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]} \otimes \tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^\dagger\|_F^2}{\|H^a_{[L_a];L_a+1,L_a+2}\|_F^2} \le O(\delta\alpha + 4\delta\kappa). \tag{68}$$

Proof. From Lemma 3 and (55), we get

$$\begin{aligned} &\|\hat{\hat{T}}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}} - T_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}\|_F \\ &= \|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I \otimes P_2^*)(P_1^* \otimes I) - T_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}\|_F \\ &\le \|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}(I \otimes P_2^*)(P_1^* \otimes I) - \hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}\|_F + \|\hat{T}_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}} - T_{\mathcal{I}_a;\mathcal{I}_{c_1}\mathcal{I}_{c_2}}\|_F \\ &\le (1+\sqrt{2})\|E\|_F. \end{aligned} \tag{69}$$

Recalling that

$$H^a_{[L_a];L_a+1,L_a+2} = T_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]} \otimes \tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^\dagger, \tag{70}$$

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

$$\begin{aligned} &\frac{\|H^a_{[L_a];L_a+1,L_a+2} - \hat{\hat{T}}_{\mathcal{I}_a;\mathcal{I}_{c_1},\mathcal{I}_{c_2}}(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]} \otimes \tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^\dagger\|_F^2}{\|H^a_{[L_a];L_a+1,L_a+2}\|_F^2} \\ &\le \frac{(1+\sqrt{2})^2\|E\|_F^2\|(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]} \otimes \tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^\dagger\|_2^2}{\|H^a_{[L_a];L_a+1,L_a+2}\|_F^2} \end{aligned}$$

$$
\begin{aligned}
&\leq \frac{(1+\sqrt{2})^2\|E\|_F^2}{\sigma_r(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]})^2\sigma_r(\tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^2\|H^a_{[L_a];L_a+1,L_a+2}\|_F^2} \\
&= \frac{(1+\sqrt{2})^2\|\hat{T}\|_F^2}{\sigma_r(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]})^2\sigma_r(\tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^2\|H^a_{[L_a];L_a+1,L_a+2}\|_F^2}\frac{\|E\|_F^2}{\|\hat{T}\|_F^2} \\
&\leq \frac{(1+\sqrt{2})^2\|\tilde{H}^{[d]\setminus a}_{d-L_a+1,d-L_a+2;[d-L_a]}\|_2^2}{\sigma_r(\tilde{H}^{c_1}_{L_{c_1}+1;[L_{c_1}]})^2\sigma_r(\tilde{H}^{c_2}_{L_{c_2}+1;[L_{c_2}]})^2}\frac{\|E\|_F^2}{\|\hat{T}\|_F^2} \\
&\leq \frac{(1+\sqrt{2})^2}{\frac{1}{\kappa^6}-\frac{2\sqrt{r}}{\kappa^2}\sqrt{1-\frac{\alpha}{\kappa^4}}}\kappa^2\left(1-\frac{\alpha}{\kappa^4}\right). \qquad (71)
\end{aligned}
$$

The first inequality follows from (69) and (70), 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 in $\hat{\hat{T}}$ (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 a 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), pp. 477–528.

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

[3] E. J. Candès and B. Recht, *Exact matrix completion via convex optimization*, Found. Comput. Math., 9 (2009), 717.

[4] M. Espig, K. K. Naraparaju, and J. Schneider, *A note on tensor chain approximation*, Comput. Vis. Sci., 15 (2012), pp. 331–344.

[5] S. Friedland, V. Mehrmann, A. Miedlar, and M. Nkengla, *Fast low rank approximations of matrices and tensors*, Electron. J. Linear Algebra, 22 (2011), 67.

[6] F. Ruvimovich Gantmacher and J. L. Brenner, *Applications of the Theory of Matrices*, Dover, Mineola, NY, 2005.

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

[8] F. L Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, J. Math. Phys., 6 (1927), pp. 164–189.

[9] Y. P. Hong and C.-T. Pan, *Rank-revealing QR factorizations and the singular value decomposition*, Math. Comp., 58 (1992), pp. 213–232.

[10] Y. Khoo, J. Lu, and L. Ying, *Solving parametric PDE problems with artificial neural networks*, European J. Appl. Math., 32 (2021), pp. 421–435.

[11] L. Mirsky, *Symmetric gauge functions and unitarily invariant norms*, Quart. J. Math., 11 (1960), pp. 50–59.

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

[13] I. Oseledets and E. Tyrtyshnikov, *TT-cross approximation for multidimensional arrays*, Linear Algebra Appl., 432 (2010), pp. 70–88.

[14] I. V. Oseledets, *Tensor-train decomposition*, SIAM J. Sci. Comput., 33 (2011), pp. 2295–2317.

[15] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac, *Matrix product state representations*, Quantum Inf. Comput., 7 (2007), pp. 401–430.

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

[17] L. R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.

[18] W. Wang, V. Aggarwal, and S. Aeron, *Efficient low rank tensor ring completion*, Proceedings of the IEEE International Conference on Computer Vision, IEEE Computer Society, Los Alamitos, CA, 2017, pp. 5697–5705.

[19] S. R. White, *Density matrix formulation for quantum renormalization groups*, Phys. Rev. Lett., 69 (1992), pp. 2863–2866.

[20] 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.

[21] M. Yuan and C.-H. Zhang, *On tensor completion via nuclear norm minimization*, Found. Comput. Math., 16 (2016), pp. 1031–1068.

[22] Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki, *Tensor Ring Decomposition*, preprint, arXiv:1606.05535, 2016.