Tensor-Ring Decomposition with Index-Splitting

Hyun-Yong Lee\textsuperscript{a,\ast}, Naoki Kawashima\textsuperscript{a,\ast}\ast

\textsuperscript{a}Institute for Solid State Physics, University of Tokyo, Kashiwa, Chiba 277-8581, Japan

Abstract

Tensor-ring decomposition of tensors plays a key role in various applications of tensor network representation in physics as well as in other fields. In most heuristic algorithms for the tensor-ring decomposition, one encounters the problem of local-minima trapping. Particularly, the minima related to the topological structure in the correlation are hard to escape. Therefore, identification of the correlation structure, somewhat analogous to finding matching ends of entangled strings, is the task of central importance. We show how this problem naturally arises in physical applications, and present a strategy for winning this “string-pull” game.

Keywords: Tensor network; Ring decomposition; Entanglement branching

1. Introduction

Whereas the method of the tensor network can be dated back to 1960s,\cite{baxter1969exact} when Baxter used it for numerical computation of the partition function of the dimer model on the square lattice, it is only quite recently that the tensor network has got the full attention it deserves in the physics community. Now the tensor networks are heavily used not only as a tool for achieving extremely accurate computation of model systems but also as an essential framework for conceptual developments. The use of the tensor network as the machine learning scheme has also been discussed recently.\cite{lee2018tensor} One prominent example for the use of the tensor network is the real space renormalization group. While the conventional Migdal-Kadanoff (MK) renormalization group makes

\textsuperscript{\ast}Corresponding author.
\textit{E-mail address:} hyunyong@issp.u-tokyo.ac.jp

\textsuperscript{\ast}\ast Correspoding author.
\textit{E-mail address:} kawashima@issp.u-tokyo.ac.jp

Preprint submitted to Elsevier  July 12, 2018
the essential idea clear, it is well-known that the simple implementation of
the idea does not yield the correct critical behavior. Moreover, it is hard to
generalize the MK scheme so that the accuracy is controllable and the method
can represent the exact fixed point in some limit. An alternative real-space
renormalization group method, tensor renormalization group (TRG)\textsuperscript{[3]} and
high-order tensor renormalization group (HOTRG)\textsuperscript{[4]} were proposed based
on the tensor network representation.

The tensor-network-based methods are not so unlike the MK scheme in
essence. In both the RG transformations, the model and the lattice is de-
formed by tracing out some local degrees of freedom to recover the same
lattice structure as before but with an enlarged lattice constant. The differ-
ence is that we lose accuracy in the MK scheme because of approximations
we make, the deformation in the new scheme is exact as long as a control
parameter, the bond dimension, is large enough. Here, the bond dimension
is the cut-off introduced in the truncation of the singular values in the sin-
gular value decomposition (SVD). Exact decomposition is indeed the case
with a few initial steps in TRG where the renormalized tensor can be split
into two sub-tensors by SVD with a relatively stringent cut-off. Of c ours,
as we proceed in the RG procedure, the renormalized tensor represents a
large-scale object that carries more mutual-information, and can no longer
be split without error. However, in a typical application, the tensor con-
verges to a so-called corner-double-line (CDL) tensor, in which the singular
values decays exponentially as a function of the index when ordered accord-
ing to its magnitude, so that the elimination of small singular values can be
an extremely good approximation. For this reason, we can often obtain the
estimates of critical exponents with many significant digits even when we use
the bond dimension of the order of 10.

Even so, with a typical computational environment, we cannot afford
large bond dimensions, e.g., 100 in the TRG of a two-dimensional lattice
model. A similar or even severer limitation applies to other schemes and
higher dimensions. Therefore, we have a good reason to be stingy about the
use of the bond dimension. One problem about this is a short-range corre-
lation. It was pointed out\textsuperscript{[5]} that the renormalized tensor in TRG does not
converge to a fixed-point tensor, the tensor that only carries information of
the fixed point. This is because the procedure in the TRG scheme does not
eliminate the correlation that lives in the scale smaller than the renormaliza-
tion scale. The mutual information arising from this short-range correlation
eats up the capacity of the tensor, and reduces the representing power for
the fixed point properties. This problem was solved\cite{5,6,7,8} by introducing an additional procedure in the RG step that explicitly eliminates local entanglement-loops that should not affect the properties in the larger scales. Another approach was proposed in Ref.\cite{9} that makes the flow of the local correlation branch out and form themselves into loops, which are traced out and become harmless.

In either way, the essence here is the identification of the local correlation/entanglement. The previous algorithm that solved the problem of local entanglement loops employed either filtering operators that eliminate the short-range correlation or branching operators that separate the short- and long-range correlations. They are computed iteratively with conditions that implicitly force the resulting operator to have the desired function. In the present article, we propose an alternative more direct method assuming that the target tensor approximately has the CDL structure. With this assumption on the tensor structure, the task is reduced to something analogous to the “string-pull” lottery, i.e., detecting which end is connected to which.

It is known that this structure is characteristic to the tensors that one encounters in the renormalization group methods based on the tensor network representation. In what follows, we show that the CDL structure emerges in the tensor renormalization group procedure and that it causes difficulty in the tensor-ring decomposition (TRD) when previously known heuristic algorithms are applied. For example, alternating least squares (ALS), which was originally proposed for the tensor train decomposition\cite{10,11}, is used for obtaining a tensor-ring decomposition\cite{12}. While the problem is not so prominent when a generic tensor is considered as the target tensor, its application to the CDL tensors reveals the drawback of previous iterative approaches. In this article, we show that it is indeed the case and that the new algorithm produces the optimal branching tensors within a much small number of iterations.

Figure 1: Tensor-ring decomposition of a tensor of the fourth-order.

In the present article, we focus on the TRD of tensors. A tensor ring
is a partially contracted set of tensors arrayed on a ring. For instance, if a tensor, $T_{ijkl}$, of the fourth-order is given, we may consider four tensors of the third-order that satisfy

$$T_{ijkl} \approx \sum_{\alpha\beta\gamma\delta} Z_1^{i\alpha\beta} Z_2^{j\beta\gamma} Z_3^{k\gamma\delta} Z_4^{l\delta\alpha}.$$  

The right-hand side is an example of the tensor ring. The TRD is schematically depicted in Fig. 1. It can be shown that we can exactly express any tensor in terms of a tensor ring if we allow each contracted index to have large enough dimension. In the case of the example shown above, where the original tensor have four indices ($i, j, k$ and $l$) of dimension $d$, we can have exact mapping if the representation can be exact if each one of the contracted indices, $\alpha, \beta, \gamma$, and $\delta$, has the dimension $\chi \geq d^2$. However, it is often desirable to obtain a tensor ring that is an accurate representation, though not exact, while the dimension of the inner bonds are small enough to meet some practical restrictions. Although it is not so widely appreciated so far, the TRD is a key procedure for various numerical calculations. In the RG calculation, for example, the TRD can be used as an alternative to the loop filtering in the loop tensor network renormalization proposed in Ref. [7].

2. CDL and HOSVD

In what follows, we mainly consider the case where the target tensor is of the third-order because it is the simplest non-trivial case and the problem with higher orders can be solved by repeating use of the method for the
third-order tensors. Also, we focus on the \textit{pseudo-CDL} tensors as defined below. A pseudo-CDL tensor is a tensor that can be expressed in terms of \textit{pseudo-unitary} matrices ($U, V$ and $W$) and positive weights $x_p, y_q, z_r \geq 0$ as

$$T_{ijk} = \sum_{pqr} U_{iqr} V_{jrp} W_{kpq} x_p y_q z_r. \quad (2)$$

Here, a pseudo-unitary matrix is an isometry that satisfies

$$\sum_{qr} U_{iqr} U_{i'r} = \delta_{i'i} \text{ (isometry)}, \quad (3)$$

and

$$\sum_{i} U_{iq'r} U_{iq'r'} = \delta_{qq'} \delta_{rr'} \Delta_{qr}, \quad (4)$$

with some function $\Delta_{qr} = 0$ or 1. Diagrammatically, $T_{ijk}$ is represented as Fig. 2. We call a tensor $T_{ijk}$ a pseudo-CDL tensor if it is expressed in the form of Eq. (2) with Eq. (3) and Eq. (4). The pseudo-CDL tensor $T_{ijk}$ is a full CDL tensor if $U, V$, and $W$ are unitary (i.e., $\Delta_{qr} = 1$ for $U$). In what follows, we refer a pseudo-CDL simply as a CDL. We impose an additional condition to remove redundancy in the representation that the indices are defined so that weight is sorted in the descending order $x_0 > x_1 > \cdots > 0$.

Note that we can express the isometries $U$ by using a mapping from subindex pairs to indices. To be more specific, let us consider the sets of the values that $q, r,$ and $i$ take, i.e., $\Omega_q = \{1, 2, \cdots, d_q\}$, $\Omega_r = \{1, 2, \cdots, d_r\}$, and $\Omega_i = \{1, 2, \cdots, D_i\}$ where $d_q, d_r$ and $D_i$ are dimensions of indices $q, r,$ and $i$, respectively. Then, we consider an “indexing function”, $I(q,r)$, from $\Omega_q \times \Omega_r$ to $\Omega_i \cup \{0\}$. We assume that $I$ is bijective when it is restricted to $I^{-1}(\Omega_i)$. Using this function we can define a third-order tensor

$$\Delta^U_{iqr} \equiv \delta_{i, I(q,r)}. \quad (5)$$

We can prove that if both the conditions Eqs. (3) and (4) hold, then such a (partially) bijective function $I(q,r)$, and therefore the tensor $\Delta^U_{iqr}$ as well, exist and the isometry $U$ can be expressed simply as a unitary rotation applied to the first index of $\Delta^U_{iqr}$, i.e.,

$$U_{iqr} = \sum_{i'=1}^{D_i} u_{ii'} \Delta^U_{iqr}, \quad (6)$$
where $u_{ii'}$ is a unitary matrix.

If we know $U,V,W,x,y$ and $z$, the TRD is given as

$$T_{ijk} = \sum_{pqr} Z_{iqr}^1 Z_{jrp}^2 Z_{kpq}^3$$  \hspace{1cm} (7)

where the latent tensors

$$Z_{iqr}^1 = U_{iqr} \sqrt{y_q z_r},$$
$$Z_{jrp}^2 = V_{jrp} \sqrt{z_r x_p},$$
$$Z_{kpq}^3 = W_{kpq} \sqrt{x_p y_q}.$$  \hspace{1cm} (8)

Therefore, our problem is to explicitly compute $U,V,W,x,y$ and $z$ for a given $T_{ijk}$.

To this end, we start from the high-order singular value decomposition (HOSVD). It is known [13] that an arbitrary tensor (a third-order tensor in the present case) can be expressed as

$$T_{ijk} = \sum_{i'j'k'} u_{ii'} v_{jj'} w_{kk'} t_{i'j'k'},$$  \hspace{1cm} (9)

where the matrices $u, v$ and $w$ are unitaries and $t_{i'j'k'}$ is a tensor whose matrix slices are mutually orthogonal, e.g.,

$$\sum_{jk} t_{ijk} t_{i'j'k'}^* = a_i \delta_{ii'}, \cdots.$$  \hspace{1cm} (10)

In fact, we can obtain the decomposition of Eq. (9) by applying the SVD repeatedly for each index, i.e.,

$$T_{ijk} = \sum_{i'j'k'} u_{ii'} v_{jj'} w_{kk'} t_{i'j'k'},$$

$$s_{i'j'k'}^1 = \sum_{i'} u_{ii'} s_{i'j'k'}^1,$$
$$s_{i'j'k'}^2 = \sum_{j'} v_{jj'} s_{i'j'k'}^2,$$
$$s_{i'j'k'}^3 = \sum_{k'} w_{kk'} s_{i'j'k'}^3.$$  \hspace{1cm} (11)

and

$$t_{ijk} = s_k^3 \bar{w}_{ijk}.$$  \hspace{1cm} (12)

We call $t_{ijk}$ the “core” tensor. Here, again, to remove redundancy, we require that the indices are defined so that $s_0^\lambda > s_1^\lambda > \cdots > 0$ ($\lambda = 1,2,3$). With
this restriction, the decomposition (9) is unique up to the gauge degrees of freedom \[14\], i.e.,
\[
t_{ijk} \rightarrow e^{(\alpha_i+\beta_j+\gamma_k)} t_{ijk}, \quad u_{ii'} \rightarrow e^{-i\alpha_i} u_{ii'},
\]
\[
v_{jj'} \rightarrow e^{-i\beta_j} v_{jj'}, \quad w_{kk'} \rightarrow e^{-i\gamma_k} w_{kk'}.
\]
(13)
Note that Eq. (2) can be cast into the form of Eq. (9), by expressing the \(U\), \(V\) and \(W\) as in Eq. (6). Namely,
\[
T_{ijk} = \sum_{i'j'k'} u_{ii'} v_{jj'} w_{kk'} t_{i'j'k'} (14)
\]
where
\[
t_{i'j'k'}' = \sum_{pqr} \Delta^U_{iqr} \Delta^V_{jrp} \Delta^W_{kpq} x_{pq} y_{pq} z_r (15)
\]
Here the “indexing” tensors \(\Delta^a (a = U, V \text{ and } W)\) are defined by using some indexing functions \(I(q, r), J(r, p)\) and \(K(p, q)\), respectively, as
\[
\Delta^U_{iqr} \equiv \delta_{i, I(q, r)}, \quad \Delta^V_{jrp} \equiv \delta_{j, J(r, p)}, \quad \Delta^W_{kpq} \equiv \delta_{k, K(p, q)}. (16)
\]
We can easily verify that, because of Eq. (4), \(t_{i'j'k'}'\)'s matrix slices are mutually orthogonal, i.e., \(t_{i'j'k'}'\) satisfies Eq. (10). This means that Eq. (14) is a HOSVD. The uniqueness of the HOSVD, then, demands that \(t'\) is identical to the core tensor of HOSVD, apart from the order of the indices and the gauge factors. This is an important observation in computing \(U\), \(V\), \(W\) and \(x, y, z\) in the following section.

One can also decompose the tensor \(T_{ijk}\) [Eq. (2)] into three latent tensors \(\{Z^1, Z^2, Z^3\}\) by applying previously proposed algorithms such as the sequential SVD or ALS and its applied versions in Ref. [12]:
\[
T_{ijk} \xrightarrow{\text{TRD}} \sum_{pqr} Z^1_{iqr} Z^2_{jrp} Z^3_{kpq}. (17)
\]
However, it is obvious that such decompositions do not remove the so-called short-range entanglement loop inside the ring network [gray line in Fig. 3(b)]. For example, in the sequential SVD method, we start with splitting an index, which connects the first and rest of latent tensors, into two subindices without concerning such indexing function. Then, it gives rise to an arbitrary entanglement by mixing the weight vectors. Also, in ALS algorithm,
we initialize the latent tensors as random tensors, which are generally known to be maximally entangled [15]. Optimization may discard many of such unexpected entanglements introduced at the beginning of both algorithms. However, some local entanglement may remain as a redundancy of the resulting tensor-ring representation. For example, consider the latent tensor of the form

\[ Z_{iqr}^{\mu} = A_{iq_1r_1}^{\mu} \delta_{q_2r_2} \]  

(18)

where \( q_1 \) and \( q_2 \) are subindices so that there is one-to-one correspondence between \( q \) and \((q_1, q_2)\). Then,

\[
\sum_{pqr} Z_{iqr}^{1} Z_{irp}^{2} Z_{ipq}^{3} = \sum_{p_1q_2r_1r_2} A_{iq_1r_1}^{1} A_{ir_1p_1}^{2} A_{ip_1q_1}^{3} \delta_{q_2r_2} \delta_{r_2p_2} \delta_{p_2q_2} 
\]

\[
\propto \sum_{p_1q_1r_1} A_{iq_1r_1}^{1} A_{ir_1p_1}^{2} A_{ip_1q_1}^{3}. 
\]

(19)

In this case, it is obvious that we can simply replace \( Z \) by a more compact tensor \( A \) without losing anything, while detecting such a redundancy is not trivial. An example is symbolically depicted in Fig. 3(a). Such redundancy is not only unnecessary for describing \( T_{ijk} \) but also leads to a higher computational complexity with less accuracy by requiring a larger bond dimension. We call this type of redundancy an entanglement loop. On the contrary, an ideal decomposition provided in Eq. (8) would be like Fig. 3(b) where the short-range entanglement is completely discarded.

3. Algorithm for Pure CDL

In this section, we propose a systematic way to find the ideal latent tensors [Eq. (8)] for TRD of CDL tensors. To this end, our main task is to find \( U, V, W, x, y \) and \( z \) for a given \( T_{ijk} \) as mentioned earlier. The whole procedure consists of four steps:
1. Finding the weight vectors $x_p, y_q$ and $z_r$
2. Finding the indexing tensors $\Delta^U_{iqr}, \Delta^V_{jrp}$ and $\Delta^W_{kpq}$
3. Fixing the gauge freedom in Eq. [13]
4. Constructing TRD using the obtained weight vectors, indexing tensors and gauge factors

Obtaining the indexing tensors and fixing the gauge, one can construct the isometries $U, V$ and $W$ up to the gauge. In the following, we present the details of each step.

3.1. Weight vectors

First, we decompose the tensor $T_{ijk}$ into unitaries ($u_{ii'}, v_{jj'}$ and $w_{kk'}$) and a core tensor ($t_{ijk}$) by applying HOSVD [Eq. [9]]:

$$t_{ijk} = \sum_{pqr} (\Phi^U_i)^* (\Phi^V_j)^* (\Phi^W_k)^* \Delta^U_{iqr} \Delta^V_{jrp} \Delta^W_{kpq} x_py_qz_r$$

To avoid confusion, we enclose the tensor network with a dotted loop throughout this paper if its internal structure is unknown, such as the left-hand side of the above equation. As pointed out in the previous section, the tensor $U$ is similar to $u$ up to the random phase difference and order of each column vector: $U_{iqr} = \sum_{i'} u_{ii'} \Phi^U_{i'} \Delta^U_{iqr}$ where $\Phi^U_{i'} = e^{i\alpha_i'}$ (similar for $V$ and $W$). Due to the gauge redundancy, the core tensor $t_{ijk}$ contains, in general, complex elements, and its internal structure of $t$ is the following:

$$t_{ijk} = \sum_{pqr} (\Phi^U_i)^* (\Phi^V_j)^* (\Phi^W_k)^* \Delta^U_{iqr} \Delta^V_{jrp} \Delta^W_{kpq} x_py_qz_r$$

or
where each blue (red) circle denotes $\Delta^a (\Phi^a)$ and $a = U, V$ and $W$. Since we are assuming those weights to be real and positive, eliminating the phases $\Phi^x$ by taking the absolute value of $t$-tensor does not affect them: $|t|_{ijk}$. Then, one can extract the $x_p$, $y_q$ and $z_r$ from $|t|_{ijk}$ as follows. Note that, because of the special internal structure of the CDL tensor, the $|t|$-tensor has the following property. When one of the three indices, say $i$, is fixed to be $i_0$, there is a one-to-one correspondence between non-zero elements of $|t|_{i_0jk}$ and the subindex weight $x_p$. Therefore, the list obtained by sorting the non-zero elements of $|t|_{i_0jk}$ in descending order must be identical to $x_p$ up to an overall constant ($c$):

$$\text{Sort}_{j,k:|t|_{i_0jk}\neq 0}(|t|_{i_0jk}, \text{\textquoteleft\textquoteleft descend\textquoteleft\textquoteleft}) = c \times (x_1, x_2, \cdots), \quad (23)$$

where the function $\text{Sort}$ returns the list of the values of the first argument sorted in descending order. Also, one can easily verify that the constant $c$ is always unity for $i_0 = 1$, resulting in $\bar{x} = x$. In a similar way, by fixing $j$- and $k$-indices to 1, one can find respectively $y$ and $z$, and then our first task is done.

3.2. Indexing tensors

Next, let us find the indexing tensors $\Delta^a_{qr}$ which is supposed to sort the singular values $s^a_{i}$ [in Eq. (11)] in descending order. Again we will consider the case where $a = \text{\textquoteleft\textquoteleft U\textquoteright\textquoteleft}$, and the extension to the other cases should be straightforward. From the internal structure of $T_{ijk}$ in Fig. 2 one can easily see that the $s^U_{i}$ is simply a vector obtained by Kronecker product of two weight vectors $y$ and $z$. Then, our index function $I(q,r)$ is such that we can find the element $y_q z_r$ in the $I(q,r)$-th place in the sorted list

$$y_{q_1} z_{r_1} > y_{q_2} z_{r_2} > y_{q_3} z_{r_3} > \cdots > y_{q_D} z_{r_D}. \quad (24)$$

(When the dimension of the index is smaller than the product of the dimensions of $q$ and $r$, we must truncate the list and assign $I(q,r) = 0$ to the elements beyond the dimension of the index $i$, i.e., $D_i$.) It is straightforward to find such a function once we obtain the vectors $y$ and $z$. Then, we can define the $\Delta^U$ tensor as

$$\Delta^U_{iqr} = \delta_{i,I(qr)}. \quad (25)$$
Algorithm 1 Finding $\Phi^a$

**Input:** A $n$th-order core tensor $t$ of size $(D_{i_1} \times \cdots \times D_{i_n})$.

**Output:** $\Phi^a$ with $a = 1, \ldots, n$ which satisfies Eq. (26).

1. Define and initialize an $n$th-order tensor $\tilde{t}$ with the input tensor $t$: $\tilde{t}_{i_1i_2\cdots i_n} = t_{i_1i_2\cdots i_n}$
2. for $l = 1$ to $n$ do
   3. for $i'_l = 1$ to $D_{i_l}$ do
      4. Find the maximum value ($c$) of the tensor $t$ with fixed $l$-th index to $i'_l$: $c := \max (t_{i_1\cdots i'_l\cdots i_n}, 'abs')$
      5. where 'abs' denotes the absolute maximum.
      6. Update $\Phi$: $\Phi^l_{i'_l} := |c|/c$
   7. end for
8. Update $\tilde{t}$ as follows: $\tilde{t}_{i_1i_2\cdots i_n} := (\Phi^l_{i'_l})^* \tilde{t}_{i_1i_2\cdots i_n}$
9. end for

3.3. Gauge factors

Now, let us turn to the phase factors $\Phi^a$ which should satisfy

$$(\Phi^U_i)^* (\Phi^V_j)^* (\Phi^W_k)^* t_{ijk} = |t|_{ijk}. \tag{26}$$

Due to the gauge freedom pointed out in Eq. (13), there is an infinite number of sets $\{\Phi^a\}$ respecting the above relation. However, any choice of the gauge does not affect the resulting tensor elements. Here, we only present an algorithm (Algorithm 1) that finds one of the solutions and leave the proof for its validity to Appendix.

3.4. Tensor ring decomposition

Using the obtained weight vectors, indexing tensors, gauge factors and unitaries from HOSVD, one can construct the ring tensor network exactly representing the CDL, $T_{ijk}$. As shown in Eq. (5), the latent tensors are given by
and similarly
\[ Z_{j'rp}^2 = \sum_{j'} v_{j'j} \Phi_{\nu}^{\nu} \Delta_{j'rp}^V \sqrt{z_{j'} x_{j} r p}, \]
\[ Z_{kpq}^3 = \sum_{k'} w_{k'k} \Phi_{\nu}^{\nu} \Delta_{kpq}^W \sqrt{x_{k'} y_{k} q}, \]
Finally, the TRD of the tensor \( T_{i,j,k} \) is accomplished with the latent tensors \( \{Z^1, Z^2, Z^3\} \):
\[ T_{ijk} = \sum_{pqr} Z_{iqr}^1 Z_{jrp}^2 Z_{kpq}^3. \]

The above decomposition is exact as long as the original tensor has the exact CDL structure [Fig. 3(a)], and the algorithm is readily generalized to the higher-order tensors.

3.5. Weak noise

In practical calculations such as the tensor renormalization group method, it is generally inevitable to encounter some noise in tensors. Unfortunately, the algorithm introduced in the previous subsection applies to only pure CDL tensors without any noise. In this subsection, we discuss how the TRD using IS, referred to as IS-TRD hereafter, can be modified to apply to such general cases.

Let us introduce random noises in the tensor as follows
\[ T_{ijk}^\gamma = T_{ijk} + \Gamma_{ijk}, \]
where \( \Gamma_{ijk} \)'s are independent uniform random numbers distributed in \([-\gamma, \gamma]\). Note that one can reasonably approximate the tensor as a pure CDL tensor as long as the noise amplitude \( \gamma \) is small enough, i.e., \( \gamma \ll 1 \) where the largest element of \( T_{ijk} \) is set to unity: \( T_{ijk}^{\gamma=0} \simeq T_{ijk}^{\gamma=1} \). Therefore, in such case,
the IS-TRD is still valid and expected to give the best ansatz for the TRD. However, because of the random noise, when we extract the weight vectors $x, y$ and $z$ in Eq. (23), the number of non-zero elements in $|t|_{1jk}$ is not $d_p$ but $D_jD_k$. We, therefore, should introduce a cut-off in Eq. (23):

$$x = \text{Sort}_{j,k: |t|_{1jk} \neq 0}(|t|_{1jk}, \text{`descend'}, d_p),$$

where the third argument $d_p$ in $\text{Sort}$ function denotes the size of the returned list. One can find the $y$ and $z$ vectors in a similar way, and then find the $\Delta^a$, $\Phi^*$ and $\{Z_1, Z_2, Z_3\}$ by following the same procedure as the pure CDL case. With a finite noise ($\gamma > 0$), the TRD

$$T^\gamma_{ijk} \simeq \sum_{pqr} Z^1_{iq}, Z^2_{jr}, Z^3_{kp},$$

with IS is not exact. However, it can be an excellent approximate decomposition for weak enough noise as demonstrated below. In addition, one can use the obtained $\{Z_1, Z_2, Z_3\}$ as a set of initial latent tensors and apply other iterative methods, e.g., ALS algorithm, to obtain better ansatz. In the present article, we combine the IS and ALS algorithms (IS-ALS for short) to obtain the best TR decomposition for the CDL with weak noise.

4. Benchmark

Figure 4: The dynamics of the approximation error $\delta \psi$ obtained by ALS and sALS for a rank-3 target tensor as a function of ALS step. The target tensor is generated by applying a random site-wise unitary transformation to a CDL tensor with inner-bonds of the rank 3 ($d_p = d_q = d_r = 3$). The truncation dimension of the inner indices, or TR-rank, is fixed to be equal to the inner-bond rank of the target tensor, i.e., $\chi = \chi_p = \chi_q = \cdots = 3$. 

13
Figure 5: The comparison of the approximation errors $\delta \psi$ obtained by the IS and sALS algorithms as a function of the inner-bond dimension $\chi$ of the ansatz. The target tensors are (a) a third-order tensor with the dimension of the inner bonds fixed to be 3, i.e., $d = d_p = d_q = \cdots = 3$, (b) a fourth-order tensor with $d = d_p = d_q = \cdots = 3$ and (c) a sixth-order tensor with $d = 2$. In (d), the dynamics of $\delta \psi$ obtained by sALS for a fourth-order tensor with $d = 3$ as a function of ALS step.

We have found that the ALS, which optimizes the initial random latent tensors iteratively, strongly depends on the initial condition and generally fails to converge to a stable ansatz for the pure or weakly disordered CDL tensors as shown in Fig. 4 (black circle). Here, the vertical axis $\delta \psi = \| T - t\text{Tr}\{Z_a\} \|_F / \| T \|_F$ stands for the approximation error by the TRD, where $\| \cdots \|_F$ and $t\text{Tr}\{Z_a\}$ denotes the Frobenius norm and the contraction of the latent tensors $\{Z_a\}$. The horizontal axis indicates the iteration step in the ALS algorithm. In order to make the ALS stable for the CDL tensors, we propose to set the latent tensors obtained by the sequential SVD algorithm[12] as the initial latent tensors of ALS (we call it sALS throughout this article). Even though sALS does not eliminate the unnecessary entanglement loop in the TRD, it is surprisingly stable and gives better ansatz than the ones obtained by ALS as shown in Fig. 4 (red triangle). Therefore, we examine the performance of the proposed IS and IS-ALS algorithms by
comparing them with the sALS decomposition.

First, we present the results of TRD errors by sALS and IS for third-order, fourth-order and sixth-order pure CDL tensors in Fig. 5 (a)-(c), respectively. The horizontal axis $\chi$ denotes the so-called TR-rank $[12]$, which is the truncation dimension of the inner bonds of the trial tensor. In other words, $\chi$ is the parameter that characterizes the capacity of the ansatz or the trial tensor. On the other hand, we denote the inner-bond dimension of the target tensor by $d$. For a target tensor, the CDL weight vectors $x_p, y_q, z_r, \cdots$ and unitary matrices $U, V, W, \cdots$ are randomly chosen. In doing so, the inner-bond dimension of the target tensor is fixed. We use $d = 3$ for the third-order and fourth-order test tensors [Fig. 5(a) and (b)] and $d = 2$ for the sixth-order test tensor [Fig. 5(c)], respectively. For IS, since the $\chi = d$ turned out to be sufficient for obtaining the optimal result, $\chi$ is set to be equal to $d$ in all calculations. As mentioned earlier, the IS decomposition is exact, i.e. $\delta \psi$ is zero up to the machine precision $O(10^{-16})$ [blue solid line in Fig. 5(a)-(c)], for pure CDL tensors ($\gamma = 0$). On the other hand, the sALS algorithm fails to produce the ideal ansatz even when $\chi$ is much larger than $d$, which suggests that the ring decomposition obtained by the sALS algorithm is redundant due to some inner entanglement loop, as anticipated. Fig. 5(d) presents the dynamics of $\delta \psi$ obtained by sALS for a third-order CDL tensor with $\chi = 3$ as a function of the iteration step. It indicates that the optimization gets stuck in a local minimum and thus cannot flow into the best solution for a given $\chi$. This also shows how efficient the IS algorithm is for the CDL tensor without any iteration.

Secondly, let us consider the disorder: $\gamma > 0$. To see the noise dependence of performances of the IS, IS-ALS and sALS algorithms, we present the dynamics of $\delta \psi$ as a function of iteration step for disorder strengths $\gamma = 10^{-5}, 10^{-4}$ and $10^{-3}$ in Fig. 6 (a), (b) and (c), respectively. Here, a fourth-order disordered CDL tensor $T$ with $d = 4$, i.e. $\text{dim}(T) = [16,16,16,16]$, is used, and the first data (at STEP=1) in the IS-ALS curves are obtained by the IS algorithm. The accuracy of IS is reduced with the finite random noise compared to the case of pure CDL tensor. However, the IS still gives a good TRD and is much more efficient than the sALS for a given bond dimension (see IS-ALS and sALS for $\chi = 4$). In addition, it is improved significantly by a single iteration of ALS, i.e. about one order of magnitude of $\delta \psi$ is reduced, as one can see from the second data in the IS-ALS curves. Note that the order of the approximation error obtained by IS-ALS is around the order of $\gamma$, which is good enough for the practical application. On the
Figure 6: Noise dependence of the performance of IS-ALS and sALS for a (randomly chosen) 4th-degree disordered CDL tensor with the disorder amplitude (a) $\gamma = 10^{-6}$, (b) $\gamma = 10^{-5}$ and (c) $\gamma = 10^{-4}$. The target tensors are obtained from randomly generated CDL with the 4-diensional inner indices, $d = 4$, and therefore the dimension of each one of the four outer indices of a target tensor $T$ is $D = 16$ or $\dim(T) = [16, 16, 16, 16]$.

contrary, the performance of sALS does not depend much on the disorder strength, since the special structure of CDL is not taken into account in the algorithm. Therefore, one expects that the IS does not produce any advantage over the sALS once the amplitude of the noise exceeds a certain threshold value. We found that, for tensors with the dimension $[16, 16, 16, 16]$, the sALS begins to produce a better TRD in the stronger disorder regime $\gamma \gtrsim 10^{-3}$, above which the CDL structure is almost non-detectable. Notice that, with the sALS, better accuracy is not guaranteed by a larger capacity of the ansatz, i.e., a larger $\chi$. For example, in Fig.6(b), the ansatz with $\chi = 5$ is better than the ones with $\chi = 6, 7$. Meanwhile, the IS-ALS ensures the better TRD by enlarging $\chi$ as shown in Fig.7, in which the same tensor as the one used in Fig.6(c) is decomposed with various $\chi$. It indicates that the initial latent tensors obtained by the IS algorithm are already close to the global minimum of the solution.

5. Application

In this section, we present an exemplary application of IS and IS-ALS algorithms for tensors obtained from a practical problem; two-dimensional classical Ising model in the square lattice, whose partition function reads

$$Z = \sum_{\sigma_1 = \pm 1} \cdots \sum_{\sigma_N = \pm 1} e^{\beta \sum_{\langle i,j \rangle} \sigma_i \sigma_j}.$$  

(33)
Here, \( N \) is the total number of Ising spin, \( \beta \) is the inverse temperature, \( \sigma_i \) denotes Ising spin at site \( i \) and \( \langle i,j \rangle \) stands for the nearest-neighbor pair in the whole lattice. Evaluation of above partition function can be carried out by contracting a square-lattice tensor network with a tensor \( T^{\text{Ising}}_{ijkl} (i,j,k,l = 0 \text{ or } 1) \) given by:

\[

T^{\text{Ising}}_{0101} = e^{-4\beta}, \quad T^{\text{Ising}}_{1010} = e^{-4\beta}, \quad T^{\text{Ising}}_{0000} = e^{4\beta}, \quad T^{\text{Ising}}_{1111} = e^{4\beta}, \quad \text{others} = 0.
\]

As mentioned in the introduction part, the TRG of the above tensor flows into the CDL-type tensor at all temperatures. To be more specific, above the critical temperature or \( \beta < \beta_c \), the fixed-point tensor has the exact CDL structure as depicted in Fig. 8(a). Therefore, one may expect that the fixed-point tensors can be decomposed very well by the IS and IS-ALS algorithms. Figure 9 shows the approximation error as a function of the temperature \( (T = 1/\beta) \) obtained by the IS and IS-ALS with \( \chi = 4 \) for the fixed-point tensor of dimension \([16,16,16,16] \). The pure IS decomposition becomes less accurate as the temperature approaches to the critical point. However, the following ALS algorithm suppresses the error significantly and produces the exact TR ansatz (red triangle in Fig. 9). In other words, one can obtain the exact TR decomposition at all temperatures above \( T_c \) with the IS-ALS. We believe that this result may open new possibilities for the index-splitting based algorithm eliminating the short-range entanglements in the tensor renormalization group.
Figure 8: Schematic figures for the fixed-point tensor of the two dimensional Ising model for (a) $\beta < \beta_c$ and (b) $\beta > \beta_c$ where $\beta (\beta_c)$ is the inverse (critical) temperature.

Figure 9: The approximation error $\delta \psi$ as a function of the temperature ($T$) by obtained by the IS and IS-ALS with $\chi = 4$ for the $[16, 16, 16, 16]$-sized fixed-point tensor of two dimensional Ising model. Here, $T_c$ denotes the critical temperature.

At $T \leq T_c$, even the IS-ALS cannot produce a good TRD. That is because the fixed-point tensors below $T_c$ are not precisely CDL but have a different structure in which an inner tensor (\textit{c}-tensor) is covered by the CDL structure as depicted in Fig.8(b). However, the index splitting of such generalized-CDL tensors are out of the current article. We leave a generalization of the current algorithm, which applies to the generalized-CDL tensors, for near-future work.

6. Conclusion

The CDL structure naturally arises in the TRG procedure. We have proposed a method, i.e., the index splitting, for efficiently compute the TRD of a tensor with approximate CDL structure, by decomposing each index into sub-indices so that the CDL structure is obvious. We have demonstrated that the CDL tensor causes serious convergence problem in previously proposed heuristic algorithms such as the ALS algorithm, whereas the present algorithm yields an optimal decomposition without convergence problem.
We have also presented that the IS combined with ALS shows great accuracy, efficiency and stability improvements in TRD for the weakly disordered CDL tensors. It implies that the index splitting procedure has not only great potential to be applicable to various problems, such as TRG, but also provides us a better understanding of the tensors we encounter in physics and mathematics.

Appendix A. Gauge Fixing

In this appendix, we discuss the problem of fixing the gauge. We can (and have to) assume that there are parameters \( p, q \) and \( r \) and functions \( I(q, r), J(r, p) \) and \( K(p, q) \) such that \( t_{ijk} \) is non-vanishing only when \( i = I(q, r), j = J(r, p) \) and \( k = K(p, q) \) for some combination of \( p, q \) and \( r \). We also assume that \( I, J \) and \( K \) are injective, e.g., if \( I(q, r) = I(q', r') \) then \( q = q' \) and \( r = r' \). Because of the injectivity, we can define the “inverse” function \( q_I(i) \) and \( r_I(i) \) so that \( i = I(q_I(i), r_I(i)) \). However, we do not assume that we know these functions. Let us define the symbol \( \epsilon \) as

\[
\epsilon_{ijk} \equiv \frac{t_{ijk}}{|t_{ijk}|} \quad (t_{ijk} \neq 0),
\]

and \( \epsilon_{ijk} = 0 \) if \( t_{ijk} = 0 \). Our problem is to find the factors \( \alpha_i, \beta_j \) and \( \gamma_k \), such that

\[
\epsilon_{ijk} = \alpha_i \beta_j \gamma_k
\]

whenever the left-hand side is not zero. Then, it is convenient to introduce abbreviation \( \alpha_{qr} \equiv \alpha_{I(qr)}, \beta_{rp} \equiv \beta_{J(rp)}, \) and \( \gamma_{pq} \equiv \gamma_{K(pq)} \). In addition, we define \( j_p \) as the “projection” of \( j \) onto the \( p = 1 \) plane, i.e., \( j_p \equiv J(r, j(1)) \). Symbols \( j_r, k_p, k_q, i_q, i_r \) are defined similarly. Assuming the existence of the solution to (A.2), we have

\[
\epsilon_{I(qr)J(rp)K(pq)} = \alpha_{qr} \beta_{rp} \gamma_{pq}.
\]

With these definitions, let us consider the following functions

\[
\epsilon'_{ijk} \equiv \epsilon_{ij, k_p} \epsilon_{ijk},
\]

\[
\epsilon''_{ijk} \equiv \epsilon'_{i, j, k} \epsilon_{ijk},
\]

\[
\epsilon'''_{ijk} \equiv \epsilon''_{i, j, k} \epsilon''_{ijk}.
\]
where $\bar{\epsilon} \equiv \epsilon^{-1}$. We can show that $\epsilon''_{ijk} = 1$ for all $ijk$ that makes $\epsilon_{ijk}$ non-zero. We can see this as follows

\[
\epsilon'_{ijk} = (\bar{\alpha}_{qr}\bar{\beta}_{r1}\bar{\gamma}_{1q})(\alpha_{qr}\beta_{rp}\gamma_{pq}) = \bar{\beta}_{r1}\bar{\gamma}_{1q}\bar{\beta}_{rp}\gamma_{pq}
\]
\[
\epsilon''_{ijk} = (\beta_{r1}\gamma_{11}\bar{\beta}_{rp}\gamma_{pq})(\bar{\beta}_{r1}\gamma_{1q}\beta_{rp}\gamma_{pq}) = \gamma_{11}\gamma_{1q}\beta_{rp}\gamma_{pq}
\]
\[
\epsilon'''_{ijk} = (\bar{\gamma}_{11}\gamma_{1q}\beta_{rp}\gamma_{pq})(\gamma_{11}\gamma_{1q}\beta_{rp}\gamma_{pq}) = 1
\]

Therefore, using (A.4, A.5, A.6),

\[
\epsilon_{ijk} = \epsilon''_{i\alpha_{j,k}}\epsilon_{ijk}^{p}\epsilon_{ijk}^{p}\epsilon_{ijk}^{p} \quad (A.7)
\]

Since $\epsilon_{ijp}k_{p}$, $\epsilon'_{i\alpha_{j,k}}$, and $\epsilon''_{i\alpha_{j,k}}$ depend only on $i$, $j$, and $k$, respectively, we can re-express them as

\[
\alpha'_{i} \equiv \epsilon_{ijp}k_{p}, \quad \beta'_{j} \equiv \epsilon'_{i\alpha_{j,k}}, \quad \text{and} \quad \gamma'_{k} \equiv \epsilon''_{i\alpha_{j,k}} \quad (A.8)
\]

which allows us to write

\[
\epsilon_{ijk} = \alpha'_{i}\beta'_{j}\gamma'_{k}. \quad (A.9)
\]

Though this may not be the same as (A.2), it is a solution to our problem nonetheless. (Note that the solution is not unique because of the degree of freedom of multiplying $\alpha'_{i}$ and $\beta'_{j}$ by the same arbitrary factor $\eta_{r}$, and similar degrees of freedom concerning $\beta'_{j}$ and $\gamma'_{k}$, and $\gamma'_{k}$ and $\alpha'_{i}$.)

Now, we have to ask if we can compute $\alpha'$, $\beta'$, $\gamma'$ even if we do not know the explicit form of the functions such as $J(r,p)$ and $r_{i}(j)$ that we have used, through $j_{p}$ and $k_{p}$, in the definition of $\epsilon'$ and $\epsilon''$. To answer this question, notice that the element with $p = 1$ is largest among the ones with the same $q$ and $r$. Sharing the same $q$ and $r$ means sharing the same $i$. Therefore,

\[
(j_{p},k_{p}) = \arg \max_{(j',k')} |t_{I(j,k)}| (A.10)
\]

where $I(j,k)$ is the value of $i$ that makes $t_{ijk}$ non-zero. Such $i$ is unique when the tensor is exactly a CDL. For tensors that are only approximately CDL, the uniqueness is not guaranteed. Therefore, we had better define the function $I(j,k)$ by the largest element in order to make the procedure applicable to such cases:

\[
I(j,k) \equiv \arg \max_{i} |t_{ijk}|. \quad (A.11)
\]
With this definition and (A.10) together with (A.4), (A.5), and (A.6) we can compute $\alpha'$, $\beta'$ and $\gamma'$ without knowing the explicit form of subindex decomposition.

Acknowledgements

The present work was motivated and inspired by discussions with S. Morita, T. Okubo, T. Suzuki and K. Harada. The computation in the present work is executed on computers at the Supercomputer Center, ISSP, University of Tokyo, and also on K-computer (project-ID: hp170262). The present work is financially supported by the MEXT project “Exploratory Challenge on Post-K computer” (Frontiers of Basic Science: Challenging the Limits), and also by ImPACT Program of Council for Science, Technology and Innovation (Cabinet Office, Government of Japan).

References

[1] R. J. Baxter, Dimers on a rectangular lattice, Journal of Mathematical Physics 9 (4) (1968) 650–654.

[2] E. Stoudenmire, D. J. Schwab, Supervised learning with tensor networks, in: D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems 29, Curran Associates, Inc., 2016, pp. 4799–4807. URL http://papers.nips.cc/paper/6211-supervised-learning-with-tensor-networks.pdf

[3] M. Levin, C. P. Nave, Tensor renormalization group approach to two-dimensional classical lattice models, Phys. Rev. Lett. 99 (2007) 120601. doi:10.1103/PhysRevLett.99.120601 URL https://link.aps.org/doi/10.1103/PhysRevLett.99.120601

[4] Z. Y. Xie, J. Chen, M. P. Qin, J. W. Zhu, L. P. Yang, T. Xiang, Coarse-graining renormalization by higher-order singular value decomposition, Phys. Rev. B 86 (2012) 045139. doi:10.1103/PhysRevB.86.045139 URL https://link.aps.org/doi/10.1103/PhysRevB.86.045139

[5] Z.-C. Gu, X.-G. Wen, Tensor-entanglement-filtering renormalization approach and symmetry-protection, Phys. Rev. B 80 (2009) 155131. doi:10.1103/PhysRevB.80.155131 URL https://link.aps.org/doi/10.1103/PhysRevB.80.155131
[6] G. Evenbly, G. Vidal, Tensor network renormalization, Phys. Rev. Lett. 115 (2015) 180405. doi:10.1103/PhysRevLett.115.180405
URL https://link.aps.org/doi/10.1103/PhysRevLett.115.180405

[7] S. Yang, Z.-C. Gu, X.-G. Wen, Loop optimization for tensor network renormalization, Phys. Rev. Lett. 118 (2017) 110504. doi:10.1103/PhysRevLett.118.110504
URL https://link.aps.org/doi/10.1103/PhysRevLett.118.110504

[8] M. Hauru, C. Delcamp, S. Mizera, Renormalization of tensor networks using graph-independent local truncations, Phys. Rev. B 97 (2018) 045111. doi:10.1103/PhysRevB.97.045111
URL https://link.aps.org/doi/10.1103/PhysRevB.97.045111

[9] K. Harada, Entanglement branching operator, Phys. Rev. B 97 (2018) 045124. doi:10.1103/PhysRevB.97.045124
URL https://link.aps.org/doi/10.1103/PhysRevB.97.045124

[10] S. Holtz, T. Rohwedder, R. Schneider, The alternating linear scheme for tensor optimization in the tensor train format, SIAM Journal on Scientific Computing 34 (2) (2012) A683–A713. arXiv:https://doi.org/10.1137/100818893
URL https://doi.org/10.1137/100818893

[11] T. Rohwedder, A. Uschmajew, On local convergence of alternating schemes for optimization of tensors, SIAM Journal on Numerical Analysis 51 (2) (2013) 1134–1162. arXiv:https://doi.org/10.1137/110857520
URL https://doi.org/10.1137/110857520

[12] Q. Zhao, G. Zhou, S. Xie, L. Zhang, A. Cichocki, Tensor ring decomposition, arXiv preprint arXiv:1606.05535.

[13] L. R. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31 (3) (1966) 279–311. doi:10.1007/BF02289464
URL https://doi.org/10.1007/BF02289464

[14] L. De Lathauwer, B. De Moor, J. Vandewalle, A multilinear singular value decomposition, SIAM Journal
on Matrix Analysis and Applications 21 (4) (2000) 1253–1278. arXiv:https://doi.org/10.1137/S0895479896305696, doi:10.1137/S0895479896305696.
URL https://doi.org/10.1137/S0895479896305696

[15] P. Hayden, D. Leung, P. W. Shor, A. Winter, Randomizing quantum states: Constructions and applications. Communications in Mathematical Physics 250 (2) (2004) 371–391. doi:10.1007/s00220-004-1087-6
URL https://doi.org/10.1007/s00220-004-1087-6