Sci. Rep. Kanazawa Univ.
Vol. 65, pp. 1–9, 2021

ARTICLES

Cost reduction of the bottleneck contraction in the higher-order tensor renormalization group

Hideaki OBA*

Graduate School of Natural Science and Technology, Kanazawa University
Kakuma, Kanazawa 920-1192, Japan

(Received January 23, 2021 and accepted in revised form March 2, 2021)

Abstract We reduce the computational cost of the bottleneck contraction in the higher-order tensor renormalization group (HOTRG) by using the singular value decomposition. To confirm the effectiveness of the proposed technique, we examine the HOTRG with or without the proposed technique by using the square lattice Ising model at criticality. Then, we find that the proposed method provides the consistent results with the original HOTRG while the computational cost and the elapsed time are reduced. Therefore, we conclude that the performance of the proposed method is better than that of the original one.

Keywords. Higher-order tensor renormalization group, tensor contraction, singular value decomposition

1 Introduction

Tensor network (TN) methods are very useful tools to study quantum and classical many-body systems [1,2]. By using the methods, one can calculate a wave function and a partition function of a lattice system. There are remarkable properties of TN methods. For example, they can handle a large volume system easily, and avoid the sign problem which occurs in some lattice models when using the Monte Carlo methods. Therefore, we expect that we can explore the models which have the sign problem by using the TN methods.

Tensor renormalization group (TRG) [3] is the one of the TN methods by which one can calculate the partition function of a two-dimensional classical lattice model. The concept of TRG is that the tensor network of the partition function is transformed into one which has less numbers of tensors by using the singular value decomposition (SVD). By extending TRG, various TN methods have been proposed since it was introduced [4–18], and there are some algorithms which allow us to deal with arbitrary-dimensional systems. Higher-order tensor renormalization group (HOTRG) [6] is one of them and its important feature is to use the higher-order SVD. However, the computational cost of the HOTRG turns out to be worse when the dimension of the lattice model becomes large. The computational cost of the HOTRG is $O(\chi^{4d-1})$, where χ is the bond dimension and d is the dimension of the system. The most time-consuming calculation in the HOTRG is the contraction for making a coarse-grained tensor since one has to deal with a lot of tensors'

*Email: oba.h534@gmail.com

bonds in this calculation. To solve this problem, anisotropic tensor renormalization group [16, 17] and triad tensor renormalization group [18] have been proposed and they can significantly reduce the computational cost for the high-dimensional system compared to the HOTRG, though the accuracies of the results obtained by using the algorithms are worse than that of the HOTRG with the same bond dimension.

In this paper, we focus on the HOTRG and reduce the computational cost of the algorithm by applying the SVD to the tensor in the bottleneck contraction. We compare the original HOTRG to the HOTRG with the proposed technique by using the square lattice Ising model at criticality. Then, we find that the proposed method provides the consistent results with the original HOTRG while the computational cost and the elapsed time are reduced. We conclude that the performance of the proposed method is better than that of the original one especially when χ becomes larger.

The rest of this paper is organized as follows. In Sec. 2, we review the original HOTRG algorithm for a square lattice system. In Sec. 3, the proposed contraction technique is explained. In Sec. 4, we present numerical results of the square lattice Ising model for a comparison between the original HOTRG and the HOTRG with the proposed technique, and discuss their performances. A summary and outlooks are given in Sec. 5.

2 Higher-order tensor renormalization group

In this section, we review the HOTRG [6] for a square lattice model. First of all, we rewrite the partition function of the system in terms of a tensor network as

$$Z = \mathrm{Tr} \prod_i T^{(\mathrm{init})}_{i;x_0 x_1 y_0 y_1}, \tag{2.1}$$

where $T^{(\mathrm{init})}_i$ is an initial tensor, i runs all lattice sites, and Tr means summing up all the bond indices of the initial tensors. Then, we execute the coarse graining of the tensor network to reduce the degrees of freedom by using the HOTRG [6]. The whole flow of the HOTRG is shown in Figure 1. There are two steps in the HOTRG, making the isometries Us and inserting them in the tensor network [Fig. 1(a)], and contracting two Ts and two Us [Fig. 1(b)]. First, to obtain the isometry U, two Ts are contracted in the following ways,

$$K_{x_1 y_0 \tilde{x}_1 \tilde{y}_0} = \sum_{x_0, y_1 = 1}^{\chi} T_{x_0 x_1 y_0 y_1} T_{x_0 \tilde{x}_1 \tilde{y}_0 y_1}, \tag{2.2}$$

$$L_{x_1 y_2 \tilde{x}_1 \tilde{y}_2} = \sum_{x_2, y_3 = 1}^{\chi} T_{x_1 x_2 y_2 y_3} T_{\tilde{x}_1 x_2 \tilde{y}_2 y_3}, \tag{2.3}$$

$$K'_{x_1 y_1 \tilde{x}_1 \tilde{y}_1} = \sum_{x_0, y_0 = 1}^{\chi} T_{x_0 x_1 y_0 y_1} T_{x_0 \tilde{x}_1 y_0 \tilde{y}_1}, \tag{2.4}$$

$$L'_{x_1 y_3 \tilde{x}_1 \tilde{y}_3} = \sum_{x_2, y_2 = 1}^{\chi} T_{x_1 x_2 y_2 y_3} T_{\tilde{x}_1 x_2 y_2 \tilde{y}_3}. \tag{2.5}$$

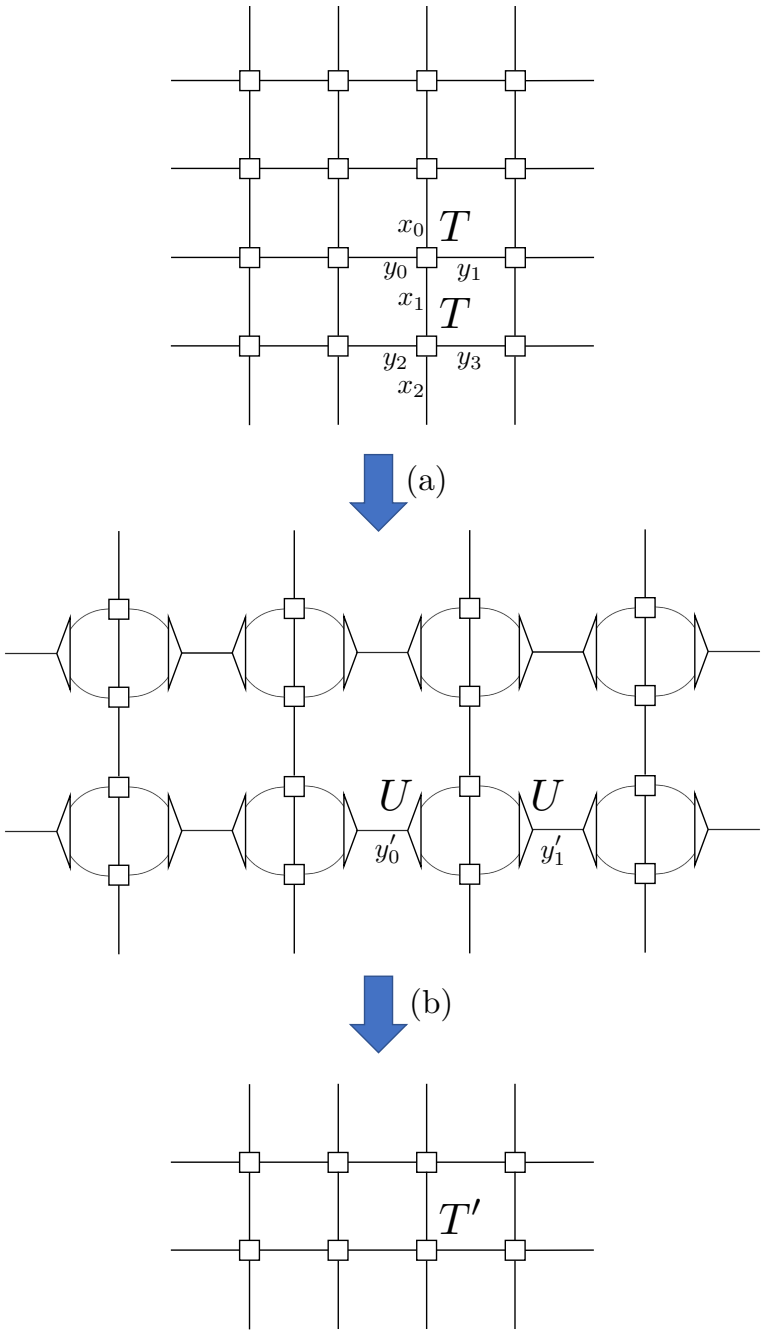

Figure 1: The flow of the HOTRG. (a) Making the isometries Us and inserting them in the tensor network. (b) Contracting two Ts and two Us to obtain T'.

where T is the $\chi \times \chi \times \chi \times \chi$ tensor [1]. Next, K and L are contracted to M, and then K' and L' are contracted to N as

$$M_{y_0 y_2 \tilde{y}_0 \tilde{y}_2} = \sum_{x_1, \tilde{x}_1 = 1}^{\chi} K_{x_1 y_0 \tilde{x}_1 \tilde{y}_0} L_{x_1 y_2 \tilde{x}_1 \tilde{y}_2}, \tag{2.6}$$

$$N_{y_1 y_3 \tilde{y}_1 \tilde{y}_3} = \sum_{x_1, \tilde{x}_1 = 1}^{\chi} K'_{x_1 y_1 \tilde{x}_1 \tilde{y}_1} L'_{x_1 y_3 \tilde{x}_1 \tilde{y}_3}. \tag{2.7}$$

[1] Even if the bond dimensions of the tensor T are smaller than χ, one can implement the contractions and decompositions of the tensors in HOTRG similarly.

Then, one implements the SVD or the eigenvalue decomposition to M and N as,

$$M_{y_0y_2\tilde{y}_0\tilde{y}_2} = \sum_{i=1}^{\chi^2} U_{\{\mathrm{L}\}y_0y_2i}\Lambda_{\{\mathrm{L}\}ii}U_{\{\mathrm{L}\}\tilde{y}_0\tilde{y}_2i}, \tag{2.8}$$

$$N_{y_1y_3\tilde{y}_1\tilde{y}_3} = \sum_{j=1}^{\chi^2} U_{\{\mathrm{R}\}y_1y_3j}\Lambda_{\{\mathrm{R}\}jj}U_{\{\mathrm{R}\}\tilde{y}_1\tilde{y}_3j}, \tag{2.9}$$

where the singular values, $\Lambda_{\{\cdot\}}$s, are ordered such that $\Lambda_{\{\cdot\}kk} \geq \Lambda_{\{\cdot\}ll} \geq 0$ $(k < l)$, and evaluates $\varepsilon_1, \varepsilon_2$ defined by

$$\varepsilon_1 = \sum_{i>\chi} \Lambda_{\{\mathrm{L}\}ii}, \tag{2.10}$$

$$\varepsilon_2 = \sum_{j>\chi} \Lambda_{\{\mathrm{R}\}jj}. \tag{2.11}$$

If $\varepsilon_1 < \varepsilon_2$, then we set the isometry $U = U_{\{\mathrm{L}\}}$, which is truncated by keeping the index i corresponding to the largest χ singular values. Otherwise $U = U_{\{\mathrm{R}\}}$ truncated in the same way as $U_{\{\mathrm{L}\}}$. Then, the isometries Us are inserted in the tensor network [see Fig. 1(a)].

As second step, T and U are contracted in the following two ways to obtain A and B as,

$$A_{x_0x_1y_0y_3y_1'} = \sum_{y_1=1}^{\chi} T_{x_0x_1y_0y_1}U_{y_1y_3y_1'}, \tag{2.12}$$

$$B_{x_1x_2y_0y_3y_0'} = \sum_{y_2=1}^{\chi} T_{x_1x_2y_2y_3}U_{y_0y_2y_0'}. \tag{2.13}$$

Then A and B are contracted to make a coarse-grained tensor, T',

$$T'_{x_0x_2y_0'y_1'} = \sum_{x_1,y_0,y_3=1}^{\chi} A_{x_0x_1y_0y_3y_1'}B_{x_1x_2y_0y_3y_0'}. \tag{2.14}$$

The computational cost of making U is $O(\chi^6)$ and that of the contraction of four tensors is $O(\chi^7)$. Therefore, the latter part is the bottleneck in the HOTRG.

3 Cost reduction of the bottleneck contraction in the HOTRG

In this section, we explain how to apply the SVD to the tensor T and reduce the computational cost of the bottleneck contraction part in the HOTRG, Eq. (2.14). The arrow named (a) of Figure 2 represents executing Eq. (2.14) and the flow of the proposed technique can be seen from (b) to (c) of Figure 2. First, we decompose T by implementing the truncated SVD with the bond dimension χ_{p} $(\chi \leq \chi_{\mathrm{p}} \leq \chi^2)$ as,

$$\begin{aligned} T_{x_0x_1y_0y_1} &\approx \sum_{n=1}^{\chi_{\mathrm{p}}} X_{x_1y_0n}S_{nn}Y_{x_0y_1n} \\ &= \sum_{n=1}^{\chi_{\mathrm{p}}} C_{x_1y_0n}D_{x_0y_1n}, \end{aligned} \tag{3.1}$$

where

$$C_{x_1y_0n} = \sqrt{S_{nn}}X_{x_1y_0n}, \tag{3.2}$$

$$D_{x_0y_1n} = \sqrt{S_{nn}}Y_{x_0y_1n}. \tag{3.3}$$

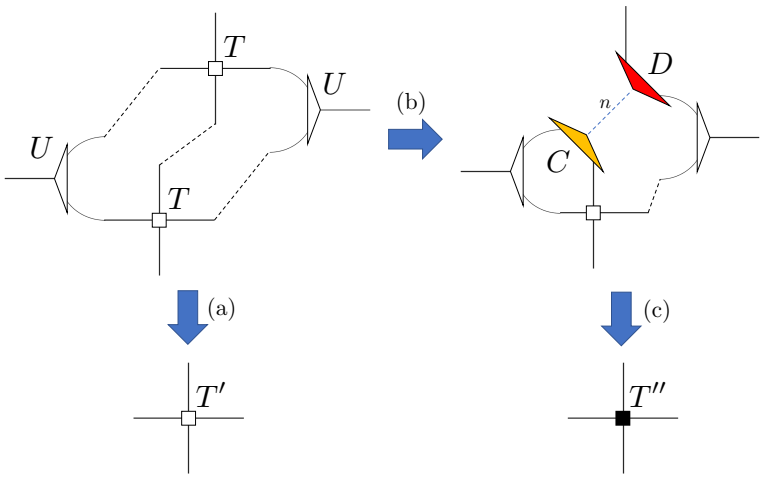

Figure 2: The way to implement the SVD to the tensor T in the bottleneck contraction of the HOTRG. (a) The contraction of two Ts and two Us in the HOTRG. (b) T is decomposed to C and D. (c) The contractions shown in Eq. (3.4)–(3.6) are executed and T'' is obtained.

Then, we replace Eq. (2.12)–(2.14) with the following calculations,

$$A'_{x_0 y_3 y'_1 n} = \sum_{y_1=1}^{\chi} D_{x_0 y_1 n} U_{y_1 y_3 y'_1}, \tag{3.4}$$

$$B'_{x_2 y_3 y'_0 n} = \sum_{x_1, y_0, y_2=1}^{\chi} T_{x_1 x_2 y_2 y_3} \left(U_{y_0 y_2 y'_0} C_{x_1 y_0 n} \right), \tag{3.5}$$

$$T''_{x_0 x_2 y'_0 y'_1} = \sum_{y_3=1}^{\chi} \sum_{n=1}^{\chi_{\rm p}} A'_{x_0 y_3 y'_1 n} B'_{x_2 y_3 y'_0 n} \approx T'_{x_0 x_2 y'_0 y'_1}. \tag{3.6}$$

The computational cost of this alternative method to obtain T'', which is the approximation of T' in Eq. (2.14), is $O(\chi_{\rm p}\chi^5)$. If $\chi_{\rm p} < \chi^2$, then the computational cost of this part becomes less than $O(\chi^7)$.

4 Numerical results

In this section, we show the numerical results of the $2^{15} \times 2^{15}$ Ising model at criticality with the periodic boundary condition in order to confirm the effectiveness of the proposed technique [2]. The initial tensor of the Ising model on the square lattice at the temperature $T = 1/\beta$ is given as

$$T^{(\rm init)}_{x_0 x_1 y_0 y_1} = \sum_{p,q,r,s=1}^{2} \delta_{pq} \delta_{qr} \delta_{rs} \delta_{sp} W_{p x_0} W_{q x_1} W_{r y_0} W_{s y_1}, \tag{4.1}$$

where

$$W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix} = \begin{bmatrix} \sqrt{\cosh\beta} & \sqrt{\sinh\beta} \\ \sqrt{\cosh\beta} & -\sqrt{\sinh\beta} \end{bmatrix}. \tag{4.2}$$

[2] The spectrum of the tensor T decays the most slowly and χ_p should be the largest at criticality. Therefore, it is enough to investigate the performance of the proposed algorithm only at the critical point since the computational cost is the maximum.

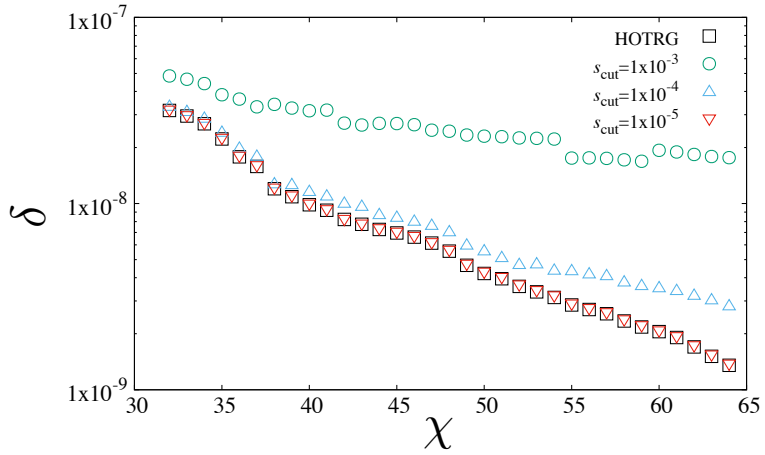

Figure 3: The relation between χ and δ of the original HOTRG and the proposed method when s_{cut} changes. We use the $2^{15} \times 2^{15}$ Ising model at criticality with the periodic boundary condition.

To see the accuracy of the free energy, the relative error of it is defined by

$$\delta = \left| \frac{f - f_{\text{exact}}}{f_{\text{exact}}} \right|, \tag{4.3}$$

where f is the free energy given by the HOTRG with or without the proposed technique, and f_{exact} is the exact value obtained by the Onsager's solution [19]. The free energy of $2^{15} \times 2^{15}$ Ising model can be obtained by implementing the 30 HOTRG steps. The result of δ is shown in Figure 3. In this figure, the bond dimension χ is set to be the integers in $32 \leq \chi \leq 64$. s_{cut} is a parameter which represents the truncation magnitude of the singular values in Eq. (3.1), and defined by 1.0×10^{-m}, where m is set to be some integers. For given s_{cut}, χ_{p} in Eq. (3.1) is defined by the number of singular values, S_{nn}, which satisfy $S_{nn} \geq S_{11} \times s_{\text{cut}}$. From Fig. 3, it is enough to set $s_{\text{cut}} \leq 1 \times 10^{-5}$ in $32 \leq \chi \leq 64$ since the results of the proposed method ($s_{\text{cut}} = 1 \times 10^{-5}$) are consistent with those of the original HOTRG.

Next, let us see the difference between the free energies obtained by the proposed method and the original HOTRG in detail. For this purpose, we define another relative error of the free energy, δ_f, as

$$\delta_f = \left| \frac{f_{\text{proposal}} - f_{\text{hotrg}}}{f_{\text{hotrg}}} \right|, \tag{4.4}$$

where f_{proposal} means the free energy given by the proposed method and f_{hotrg} is the free energy obtained by the original HOTRG. The relation between χ and δ_f is shown in Figure 4. From this figure, we find that the values of δ_f depend on s_{cut} rather than χ, and $\delta_f = 0$ with double precision for $s_{\text{cut}} = 10^{-8}$. Therefore, we conclude that f_{proposal} is equal to f_{hotrg} when $s_{\text{cut}} = 10^{-8}$.

Then, to see the performance of the proposed method, we introduce $\chi_{\text{p}}^{\text{M}}$, which is the largest χ_{p} after the 7th coarse-graining of the proposed method. Figure 5 shows the relation between χ and $\chi^2/\chi_{\text{p}}^{\text{M}}$. From this figure, we can see $\chi^2/\chi_{\text{p}}^{\text{M}}$ is increasing when χ becomes larger. Therefore, the power of χ in $\chi_{\text{p}}^{\text{M}}$ is less than 2 as $\chi_{\text{p}}^{\text{M}}$ is the functions of χ. Moreover, we can see the relation between χ and $\chi_{\text{p}}^{\text{M}}$ in Figure 6. To investigate the power of χ in $\chi_{\text{p}}^{\text{M}}$, we fit the data with the following function:

$$\chi_{\text{p}}^{\text{M}} = a\chi^c + b, \tag{4.5}$$

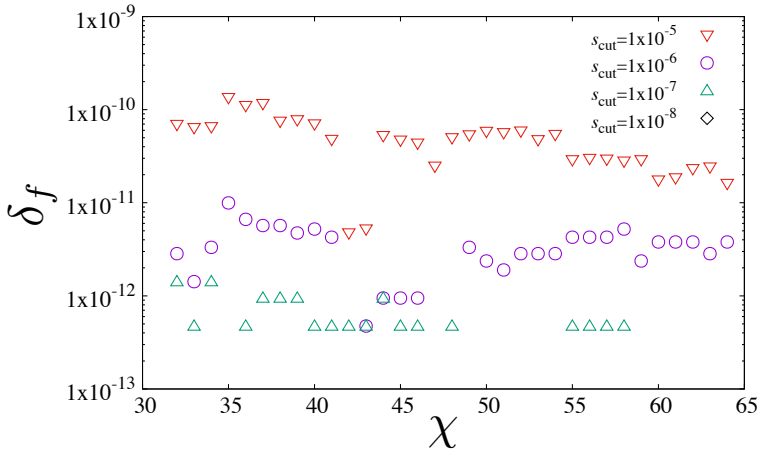

Figure 4: The relation between χ and δ_f when $s_{\rm cut}$ changes. We cannot see the results for $s_{\rm cut} = 1\times10^{-8}$ since $\delta_f = 0$ with double precision.

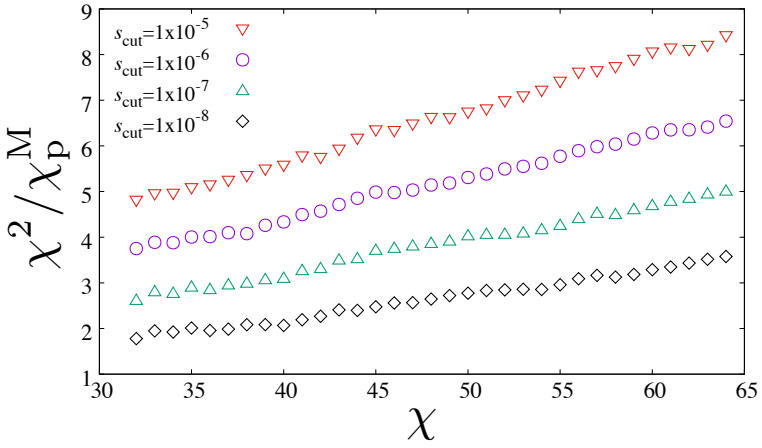

Figure 5: The relation between χ and $\chi^{\rm M}_{\rm p}/\chi^2$ of the proposed method when $s_{\rm cut}$ changes.

Table 1: The fitting results of the data in Figure 6 with Eq. (4.5).

$s_{\rm cut}$	a	b	c
10^{-5}	$1.01(44)\times10^{1}$	$-6.85(38)\times10^{1}$	$9.63(90)\times10^{-1}$
10^{-6}	$8.7(39)$	$-4.85(38)\times10^{1}$	$1.045(92)$
10^{-7}	$1.05(76)\times10^{1}$	$-1.0(86)\times10^{1}$	$1.05(15)$
10^{-8}	$1.0(10)\times10^{2}$	$-4.4(37)\times10^{2}$	$6.6(19)\times10^{-1}$

and show the fitting results in Table 1. The fitting range is set to be $32 \le \chi \le 64$. From this table, we find

$$c \approx 1 \tag{4.6}$$

in Eq. (4.5) at the all $s_{\rm cut}$. Therefore, the proposed method can reduce the computational cost from $O(\chi^7)$ to $O(\chi^6)$.

Finally, we investigate the total elapsed times of the two algorithms and a performance ratio which is defined as the division of the elapsed time of the original HOTRG by that of the proposed method with the same bond dimension. The results of the elapsed times and the performance ratio

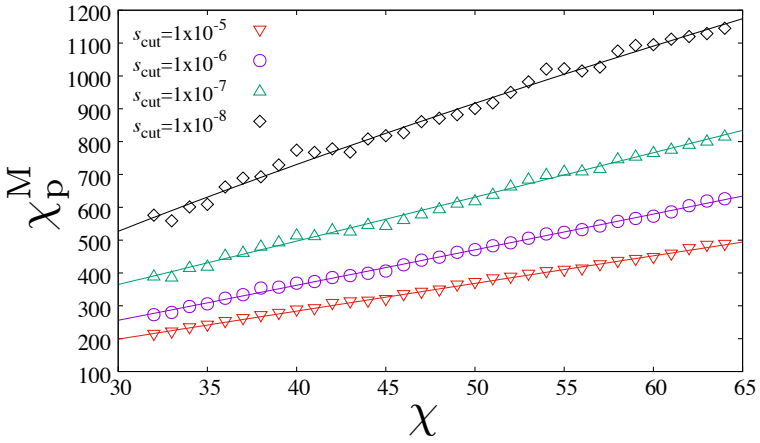

Figure 6: The relation between χ and χ_p^M of the proposed method when s_{cut} changes. The solid lines are the fitting lines of the data with the function shown in Eq. (4.5).

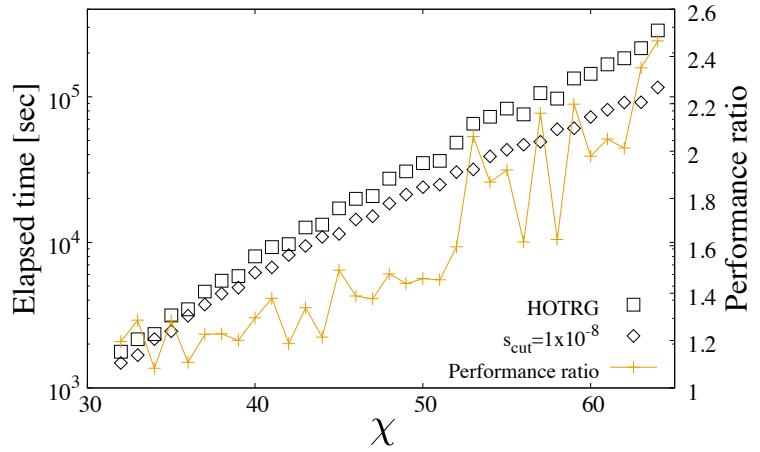

Figure 7: The elapsed times of the original HOTRG and the proposed method ($s_{cut} = 10^{-8}$), and the performance ratio between the two methods as the functions of χ.

are shown in Figure 7. From this figure, we find that the elapsed time of the proposed method is smaller than that of the original HOTRG in $32 \leq \chi \leq 64$ and the performance ratio becomes larger as the bond dimension grows. Therefore, we conclude that the performance of the proposed method is better than that of the original HOTRG. For measuring the elapsed times, we used the machine which has 24GB for the memory and Xeon X5670 (2.93 Ghz 6 core) for the two CPUs. The programs of the algorithms are written with python 2.7.6rc1 and we used numpy.tensordot in numpy 1.8.0 for the tensor contractions and scipy.linalg.svd in scipy 0.14.0 for the SVD of the tensors.

5 Summary and outlooks

We reduce the computational cost of the bottleneck contraction in the HOTRG by using the SVD. The computational cost of the bottleneck part in the HOTRG becomes $O(\chi_p\chi^5)$ from $O(\chi^7)$. To confirm the effectiveness of the proposed technique, we use the square lattice Ising model and find that the free energy of the proposed method are consistent with the original HOTRG in $32 \leq \chi \leq 64$ if we set $s_{cut} = 10^{-8}$. We also find that the proposed method can reduce the computational cost

from $O(\chi^7)$ to $O(\chi^6)$, and actually the elapsed time of the proposed method is smaller than that of the original HOTRG. Therefore, we conclude that the proposed method shows better performance than the original HOTRG.

For the future works, we consider extending this technique to the HOTRG which can be applied to the high dimensional systems. However, the spectra of the tensors in the bottleneck calculation seem to decay much more slowly. To solve this situation, we may take account of applying the environment of the tensor [4] and disentanglers [7] for the new algorithm.

References

[1] J Ignacio Cirac and Frank Verstraete. Renormalization and tensor product states in spin chains and lattices. *Journal of Physics A: Mathematical and Theoretical*, 42(50):504004, dec 2009.

[2] Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117 – 158, 2014.

[3] Michael Levin and Cody P. Nave. Tensor renormalization group approach to two-dimensional classical lattice models. *Phys. Rev. Lett.*, 99:120601, Sep 2007.

[4] Z. Y. Xie, H. C. Jiang, Q. N. Chen, Z. Y. Weng, and T. Xiang. Second renormalization of tensor-network states. *Phys. Rev. Lett.*, 103:160601, Oct 2009.

[5] Zheng-Cheng Gu and Xiao-Gang Wen. Tensor-entanglement-filtering renormalization approach and symmetry-protected topological order. *Phys. Rev. B*, 80:155131, Oct 2009.

[6] Z. Y. Xie, J. Chen, M. P. Qin, J. W. Zhu, L. P. Yang, and T. Xiang. Coarse-graining renormalization by higher-order singular value decomposition. *Phys. Rev. B*, 86:045139, Jul 2012.

[7] G. Evenbly and G. Vidal. Tensor network renormalization. *Phys. Rev. Lett.*, 115:180405, Oct 2015.

[8] Shuo Yang, Zheng-Cheng Gu, and Xiao-Gang Wen. Loop optimization for tensor network renormalization. *Phys. Rev. Lett.*, 118:110504, Mar 2017.

[9] M. Bal, M. Mariën, J. Haegeman, and F. Verstraete. Renormalization group flows of hamiltonians using tensor networks. *Phys. Rev. Lett.*, 118:250602, Jun 2017.

[10] Satoshi Morita, Ryo Igarashi, Hui-Hai Zhao, and Naoki Kawashima. Tensor renormalization group with randomized singular value decomposition. *Phys. Rev. E*, 97:033310, Mar 2018.

[11] Markus Hauru, Clement Delcamp, and Sebastian Mizera. Renormalization of tensor networks using graph-independent local truncations. *Phys. Rev. B*, 97:045111, Jan 2018.

[12] Kenji Harada. Entanglement branching operator. *Phys. Rev. B*, 97:045124, Jan 2018.

[13] Glen Evenbly. Gauge fixing, canonical forms, and optimal truncations in tensor networks with closed loops. *Phys. Rev. B*, 98:085155, Aug 2018.

[14] Yoshifumi Nakamura, Hideaki Oba, and Shinji Takeda. Tensor renormalization group algorithms with a projective truncation method. *Phys. Rev. B*, 99:155101, Apr 2019.

[15] Shumpei Iino, Satoshi Morita, and Naoki Kawashima. Boundary tensor renormalization group. *Phys. Rev. B*, 100:035449, Jul 2019.

[16] Daiki Adachi, Tsuyoshi Okubo, and Synge Todo. Anisotropic tensor renormalization group. *Phys. Rev. B*, 102:054432, Aug 2020.

[17] Hideaki Oba. Cost reduction of the bond-swapping part in an anisotropic tensor renormalization group. *Progress of Theoretical and Experimental Physics*, 2020(1), 01 2020. 013B02.

[18] Daisuke Kadoh and Katsumasa Nakayama. Renormalization group on a triad network, 2019.

[19] Lars Onsager. Crystal statistics. i. a two-dimensional model with an order-disorder transition. *Phys. Rev.*, 65:117–149, Feb 1944.