


Anisotropic tensor renormalization groupDaiki Adachi,¹ Tsuyoshi Okubo^{1,2}, and Syngge Todo^{1,3}¹*Department of Physics, University of Tokyo, Tokyo 113-0033, Japan*²*JST, PRESTO, Tokyo 113-0033, Japan*³*Institute for Solid State Physics, University of Tokyo, Kashiwa 277-8581, Japan* (Received 8 July 2019; revised 13 February 2020; accepted 6 August 2020; published 24 August 2020)

We propose a different tensor renormalization group algorithm, anisotropic tensor renormalization group (ATRG), for lattice models in arbitrary dimensions. The proposed method shares the same versatility with the higher-order tensor renormalization group (HOTRG) algorithm, i.e., it preserves the lattice topology after the renormalization. In comparison with HOTRG, both the computation cost and the memory footprint of our method are drastically reduced, especially in higher dimensions, by renormalizing tensors in an anisotropic way after the singular value decomposition. We demonstrate the ability of ATRG for the square lattice and the simple cubic lattice Ising models. Although the accuracy of the present method degrades when compared with HOTRG of the same bond dimension, the accuracy with fixed computation time is improved greatly due to the drastic reduction of the computation cost.

DOI: [10.1103/PhysRevB.102.054432](https://doi.org/10.1103/PhysRevB.102.054432)**I. INTRODUCTION**

Understanding critical phenomena observed universally in many-body systems is one of the central topics in statistical physics. Due to the difficulties in solving the many-body systems analytically, however, we often have to rely on numerical methods, such as the Monte Carlo method, in the investigation of such complex systems. In recent years, on the other hand, different alternatives, the real-space renormalization group methods, have become popular more and more, and many studies have been conducted using these approaches. The *tensor renormalization group* (TRG), proposed by Levin and Nave in 2007, calculates approximately the contraction of a tensor network by using the singular value decomposition (SVD) [1]. By representing the partition function of the Ising models by a tensor network, the free energy for the honeycomb or square lattice Ising models is calculated for huge system sizes, which can be regarded virtually as in the thermodynamic limit.

Despite the success of TRG for two-dimensional systems, it is challenging to extend TRG to higher dimensions, because the framework of TRG is tightly related to the two-dimensional lattice topology. In 2012, Xie *et al.* proposed the *higher-order tensor renormalization group* (HOTRG) method, which performs the real-space renormalization of tensor networks based on the higher-order singular value decomposition [2]. Unlike TRG, HOTRG can be applied to arbitrary dimensions. Although the computation cost of HOTRG is higher than TRG, its accuracy is higher when we compare these two methods with the same bond dimension χ , which is the maximum number of the tensor indices. A lot of methods inspired by TRG and HOTRG have been proposed [3–15], and they have been used for examining classical and quantum systems not only in the condensed matter physics but also in the particle physics [16–31].

Although HOTRG can be applied to higher-dimensional systems in principle, its computation cost in the d -dimensional lattice model increases quite rapidly as $O(\chi^{4d-1})$, where d is the dimension of the lattice, as a function of bond dimension χ . Thus, it is impractical to apply HOTRG for large bond dimensions and/or high spatial dimensions. New algorithms that can perform the real-space renormalization in general dimensions with a reasonable cost are strongly demanded.

In this paper, we propose a different TRG method, referred to as the *anisotropic tensor renormalization group* (ATRG), in which the computational complexity for systems on hypercubic lattices is drastically reduced by renormalizing tensors in an anisotropic way after the tensor decomposition. The computation cost of ATRG scales as $O(\chi^{2d+1})$, which is much lower than that of the conventional HOTRG, $O(\chi^{4d-1})$, and the memory footprint of ATRG scales as $O(\chi^{d+1})$, which is also much smaller than that of HOTRG, $O(\chi^{2d})$.

The present paper is organized as follows. We describe the two- and three-dimensional ATRG algorithms in Sec. II. ATRG can be applied to dimensions higher than three in a straightforward way. In Sec. III, we calculate the free energies of the square lattice and simple cubic lattice Ising models at their critical points as demonstrations of ATRG. Finally, we discuss the advantages of our method over TRG and HOTRG in Sec. IV.

II. ALGORITHMS

Let us begin by describing the ATRG algorithm for the two-dimensional square lattice network. Hereafter, $U_{\{X\}}$, $S_{\{X\}}$, and $V_{\{X\}}$ represent, respectively, the left isometry, the singular value matrix, the right isometry of SVD of matrix X , i.e., $X = U_{\{X\}}S_{\{X\}}V_{\{X\}}^t$ or $X_{ij} = \sum_{\alpha} S_{\{X\}i\alpha}U_{\{X\}i\alpha}V_{\{X\}j\alpha}$. In ATRG, we renormalize two neighboring tensors in the horizontal (x)

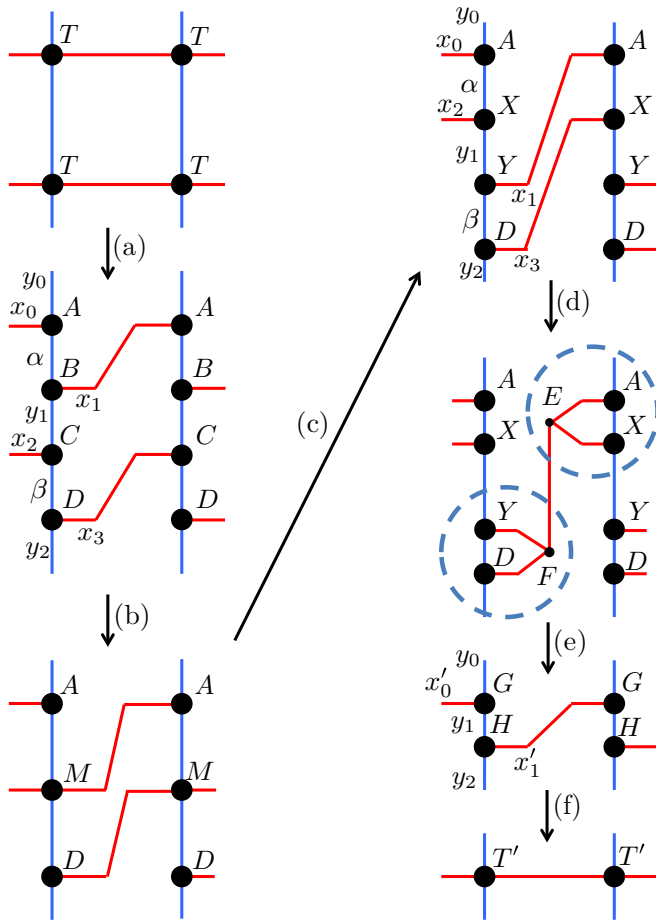


FIG. 1. Renormalization step of ATRG in y direction for the two-dimensional square lattice model.

and the vertical (y) directions alternately as in the same way as HOTRG. The outline of one ATRG renormalization step along y direction is shown in Fig. 1.

First, four-leg tensor T with bond dimension χ is approximated by a rank- χ matrix by using the partial SVD as

$$T_{y_0 y_1 x_0 x_1} \approx \sum_{\alpha=1}^{\chi} S_{\{T\}\alpha\alpha} U_{\{T\}y_0 x_0 \alpha} V_{\{T\}y_1 x_1 \alpha}. \quad (1)$$

The partial SVD (or the truncated SVD) is a method to obtain largest singular values and corresponding isometries by using the Arnoldi method or the low-rank approximation based on a randomized technique [32,33]. By using the partial SVD, the computation cost for calculating K largest singular values of an M by N matrix ($K \ll M, N$) can be reduced to $O(KMN)$, which is significantly smaller than $O(\min(M, N)MN)$, the computation cost of the standard full SVD. It has been shown that by using this technique, one can perform the original TRG with $O(\chi^5)$ computation cost [11]. In the present calculation, the decomposition of T [Eq. (1)] can also be done with $O(\chi^5)$ instead of $O(\chi^6)$. Another advantage of the partial SVD is that it does not require the full matrix, but only the multiplication of the matrix to a vector. This significant property enables us to reduce the total memory footprint of the algorithm in addition to the reduction of the computation cost as shown in the Appendix.

Then, we define four tensors, A , B , C , and D , as

$$A_{y_0 x_0 \alpha} = U_{\{T\}y_0 x_0 \alpha}, \quad (2)$$

$$B_{y_1 x_1 \alpha} = S_{\{T\}\alpha\alpha} V_{\{T\}y_1 x_1 \alpha}, \quad (3)$$

$$C_{y_1 x_2 \beta} = S_{\{T\}\beta\beta} U_{\{T\}y_1 x_2 \beta}, \quad (4)$$

$$D_{y_2 x_3 \beta} = V_{\{T\}y_2 x_3 \beta}, \quad (5)$$

[step (a) in Fig. 1]. Note that the accuracy of the final free energy may depend on how to separate the singular value matrix in the above step. Indeed, we observe that by including the singular matrix S in B and C , the error of the final free energy is minimized. Such a construction gives us better free energy than the equal weight decomposition, \sqrt{S} , of the singular matrix into A and B (or C and D). If we do not introduce truncations in step (a), our splitting of singular matrix S , together with step (b) and (c) below, gives an identical singular value spectrum to the direct partial SVD of $TT = ABCD$. In this sense, the present splitting gives the best local approximation, and we think it remains optimal even if there exist truncations in step (a).

Next, by using partial SVD, we swap the bond of B and C [step (b) and (c) in Fig. 1]. In order to swap the x_1 bond of B and x_2 bond of C , we define tensor M as

$$M_{\alpha\beta x_1 x_2} = \sum_{y_1} B_{y_1 x_1 \alpha} C_{y_1 x_2 \beta}, \quad (6)$$

and, by partial SVD of M and truncating the singular values to χ , we define new X and Y as

$$M_{\alpha\beta x_1 x_2} \approx \sum_{y_1} S_{\{M\}y_1 y_1} U_{\{M\}\alpha x_2 y_1} V_{\{M\}\beta x_1 y_1}, \quad (7)$$

$$X_{\alpha x_2 y_1} = \sqrt{S_{\{M\}y_1 y_1}} U_{\{M\}\alpha x_2 y_1}, \quad (8)$$

$$Y_{\beta x_1 y_1} = \sqrt{S_{\{M\}y_1 y_1}} V_{\{M\}\beta x_1 y_1}. \quad (9)$$

Then, we renormalize the horizontal two bonds into one by using squeezers E and F [step (d) and (e) in Fig. 1]. We call them squeezers since they are not necessarily isometries, unlike conventional HOTRG. By applying squeezer E (F) to A and X (Y and D), we obtain new tensor G (H) as

$$G_{y_0 y_1 x'_0} = \sum_{\alpha, x_0, x_2} A_{y_0 x_0 \alpha} X_{\alpha x_2 y_1} E_{x_0 x_2 x'_0}, \quad (10)$$

$$H_{y_1 y_2 x'_1} = \sum_{\beta, x_1, x_3} D_{y_2 x_3 \beta} Y_{\beta x_1 y_1} F_{x_1 x_3 x'_1}. \quad (11)$$

Finally, a new renormalized tensor, T' , is made from the product of G and H as

$$T'_{y_0 y_2 x'_0 x'_1} = \sum_{y_1} G_{y_0 y_1 x'_0} H_{y_1 y_2 x'_1} \quad (12)$$

[step (f) in Fig. 1], which is used as an input to the next renormalization step in x direction.

It should be noted that the explicit form of the squeezers is not needed for calculating the free energy. One can obtain G

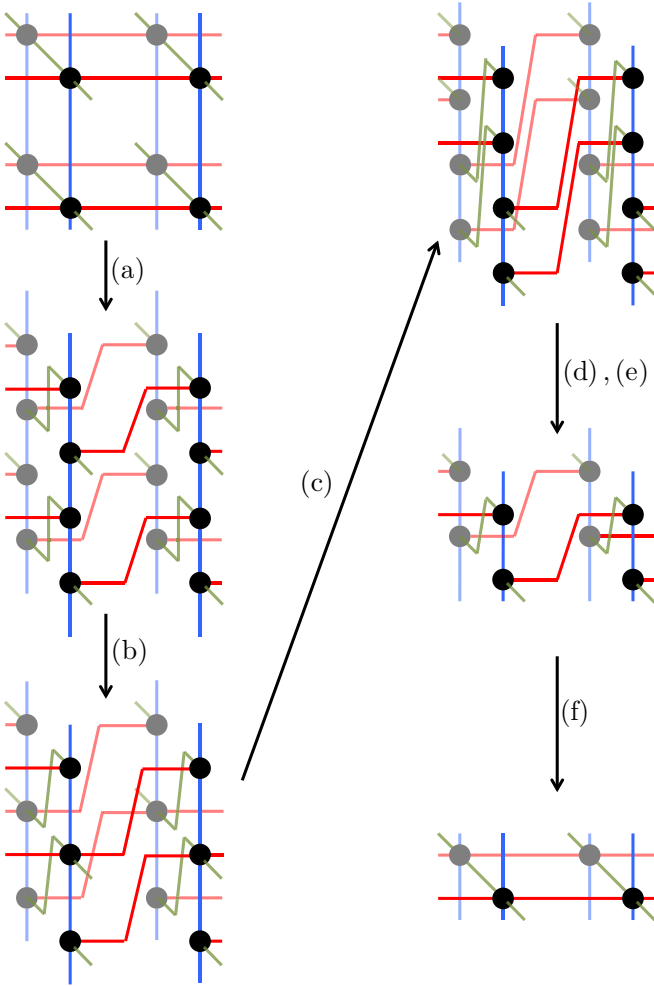


FIG. 2. Renormalization step of ATRG in z direction for the three-dimensional simple cubic lattice model.

and H directly by partial SVD as

$$\begin{aligned} Q_{y_0 y_1 y_2 y_3} &= \sum_{x_1 x_3 \alpha \beta} Y_{\beta y_1 x_1} D_{y_2 x_3 \beta} A_{y_0 x_1 \alpha} X_{\alpha x_3 y_3} \\ &\approx \sum_{x_1}^{\chi} S_{\{Q\}x_1' x_1'} U_{\{Q\}y_1 y_2 x_1'} V_{\{Q\}y_0 y_3 x_1'}, \end{aligned} \quad (13)$$

$$H_{y_1 y_2 x_1'} = \sqrt{S_{\{Q\}x_1' x_1'} U_{\{Q\}y_1 y_2 x_1'}}, \quad (14)$$

$$G_{y_0 y_3 x_1'} = \sqrt{S_{\{Q\}x_1' x_1'} V_{\{Q\}y_0 y_3 x_1'}}. \quad (15)$$

The explicit form of the squeezers is needed for calculating other physical quantities, e.g., the energy and the magnetization. We discuss the computation cost for calculating squeezers in Appendix.

One of the most significant differences between ATRG and HOTRG is that in ATRG, before applying the squeezers E and F , we construct X and Y from a low-rank approximation of M [step (b) and (c) in Fig. 1], where the singular value matrix is truncated to χ . As a result, the contraction of high-rank tensors, which dominates the computation cost in HOTRG, can be avoided. As a price, ATRG involves an additional approximation, by which the accuracy of ATRG becomes

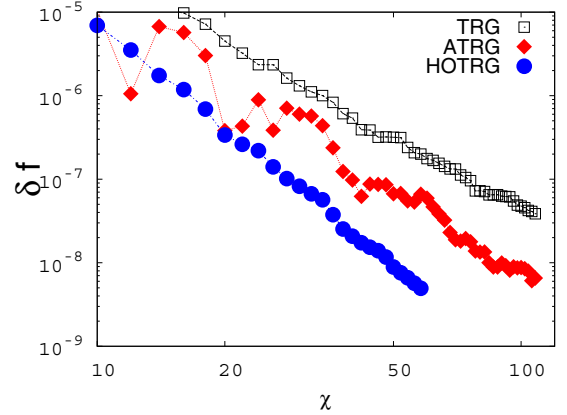


FIG. 3. Absolute error of the free energy density of the two-dimensional Ising model at $T = T_c$ as a function of bond dimension χ calculated by TRG (black squares), HOTRG (blue circles), and ATRG (red diamonds). Even though ATRG has the same computation cost as TRG, it produces more accurate results than TRG.

worse than HOTRG with a fixed bond dimension, as seen below. Also, unlike TRG and HOTRG, ATRG does not sustain the axisymmetry of the networks anymore. It is interesting to see that despite these differences, ATRG in two dimensions still has the corner double line tensor as a fixed-point tensor as TRG and HOTRG [3].

The computation cost of the whole ATRG renormalization procedure is dominated by SVD. As already mentioned above, the cost of SVD can be reduced to $O(\chi^5)$ by adopting the partial SVD. The overall computation cost of ATRG is thus the same as TRG with the partial SVD or projective truncation method [11,34] in two dimensions. The memory footprint is $O(\chi^4)$, which is needed to store the intermediate tensor M . As we see in the Appendix, however, the memory footprint can be reduced to $O(\chi^3)$.

One can easily generalize ATRG to systems in dimensions higher than two. In Fig. 2, we illustrate the renormalization step of ATRG in three dimensions. For d -dimensional hypercubic lattices, ATRG can renormalize tensors with computation cost of $O(\chi^{2d+1})$ and memory footprint of $O(\chi^{d+1})$.

III. BENCHMARKS

To demonstrate the efficiency of ATRG, we apply ATRG to the two- and three-dimensional Ising models. In both cases, each axis (x and y , or x , y , z) is renormalized 15 times alternately, i.e., the system contains $(2^{15})^d$ spins.

First, we discuss the two-dimensional Ising model on a square lattice. The free energy density calculated at the critical temperature, $T = T_c = 2/\log(1 + \sqrt{2})$ [35], is shown in Figs. 3 and 4. In Fig. 3, we plot the absolute error of the free energy density as a function of bond dimension χ . We calculate up to $\chi = 108$ ($\chi = 58$) for TRG and ATRG (HOTRG). For all χ , the result of ATRG is between those of TRG and HOTRG. Note that ATRG has the same computation cost as TRG, while the cost of HOTRG is higher than them. In order to investigate the performance of ATRG more precisely, we compare the χ dependence of the error with fixed computation time. For that purpose, we introduce the leading-order

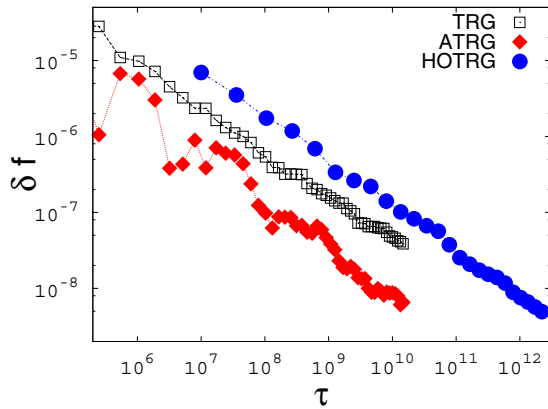


FIG. 4. Absolute error of the free energy density of the two-dimensional Ising model at $T = T_c$ as a function of leading-order computation time τ [Eq. (16)] calculated by TRG (black squares), HOTRG (blue circles), and ATRG (red diamonds). ATRG achieves the most accurate results among the three methods with fixed computation time.

computation time τ (dimensionless quantity) defined as

$$\tau = \begin{cases} \chi^5 & \text{for TRG and ATRG} \\ \chi^7 & \text{for HOTRG} \end{cases} \quad (16)$$

based on their leading computation cost. In Fig. 4, we plot the absolute error of the free energy density as a function of τ for the three methods. With fixed τ , ATRG has the smallest error among the three methods. In ATRG, partial SVD is the most expensive operation, while it is the contraction in HOTRG. In practice, the partial SVD takes a much longer time than the contraction, even when their computation costs are in the same order. Thus, the actual performance difference between ATRG and HOTRG is smaller than Fig. 4, though ATRG becomes more and more advantageous for larger χ due to the difference in the order in computation cost [Eq. (16)].

It should be mentioned that the present method suffers from larger and nonmonotonic fluctuations in the convergence of the error. This observed behavior is probably related to the two independent truncations in the ATRG renormalization procedure. Because ATRG optimizes only the local tensors in each truncation, increasing χ does not necessarily improve the accuracy of the free energy, which is determined by the global tensor network. The nonmonotonic convergence in ATRG should be subjected to further investigation.

Next, we move to the three-dimensional Ising model on a simple cubic lattice. In the three-dimensional case, we compare ATRG with HOTRG. In Fig. 5, we show the free energy density as a function of τ at $T = T_c = 4.5115$ [2,36–38]. Here, we again define the leading-order computation time τ for three dimensions as

$$\tau = \begin{cases} \chi^7 & \text{for ATRG} \\ \chi^{11} & \text{for HOTRG.} \end{cases} \quad (17)$$

In the present case, we calculate up to $\chi = 56$ ($\chi = 27$) for ATRG (HOTRG). In Fig. 5, it is demonstrated again that the ATRG gives the better (lower) free energy density than that of HOTRG for the same leading-order computation time τ . We

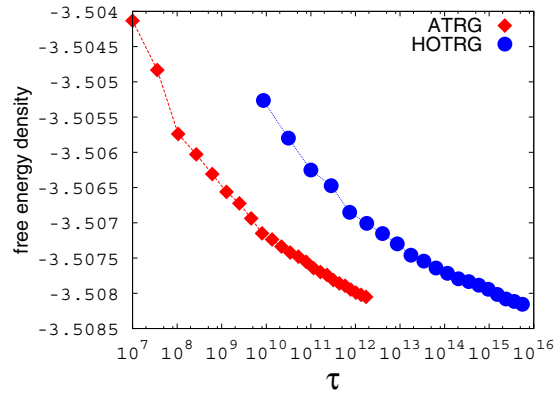


FIG. 5. Free energy density of the three-dimensional Ising model at $T = T_c$ as a function of leading-order computation time τ [Eq. (17)] calculated by HOTRG (blue circles) and ATRG (red diamonds). ATRG achieves much lower free energy than HOTRG with fixed computation time.

expect that the advantage of ATRG over HOTRG should be more pronounced in higher dimensions.

IV. SUMMARY

In this paper, we proposed the ATRG method that can perform tensor renormalization operations with computation cost of $O(\chi^{2d+1})$ and memory footprint of $O(\chi^{d+1})$ for d -dimensional hyper cubic lattices. The computation cost and the memory footprint of the proposed method are much lower than that of the conventional HOTRG, $O(\chi^{4d-1})$ and $O(\chi^{2d})$, respectively, which enables us to apply the tensor renormalization method in higher dimensions. Unlike HOTRG, our algorithm involves the truncation of the bond dimension by using SVD when we swap the bonds of two tensors. Due to this additional approximation, the accuracy in the final result degrades compared with HOTRG. However, this disadvantage is compensated by the drastic reduction of the computation cost from $O(\chi^{4d-1})$ to $O(\chi^{2d+1})$. We confirmed that for two- and three-dimensional Ising models, our method achieves higher accuracy than HOTRG with fixed leading-order computational time. Since ATRG is a real space renormalization method similar to HOTRG and preserves the lattice topology after the renormalization, it can be applied to various lattice systems in arbitrary dimensions.

Finally, as we pointed out already, the partial SVD is the most expensive operation in ATRG. The performance of the partial SVD is thus essential in ATRG, and the development of more efficient and stable partial SVD algorithms is desired for future application of ATRG to large-scale complex lattice systems.

ACKNOWLEDGMENTS

The authors thank T. Yamamoto and H. Nakano for the careful reading of the manuscript and comments. D.A. is supported by the Japan Society for the Promotion of Science through the Program for Leading Graduate Schools (MERIT). This work is partially supported by MEXT as “Exploratory Challenge on Post-K computer” (Frontiers of Basic Science:

Challenging the Limits), by JSPS KAKENHI Grants No. 15K17701, No. 17K05564, and No. 19K03740, and by JST, PRESTO Grant No. JPMJPR1912, Japan.

APPENDIX: FURTHER TECHNIQUES

In this Appendix, we discuss further detailed techniques to achieve the optimal cost in ATRG.

1. Choice of bond geometry in step (a)

In step (a) of Fig. 1, there remains freedom in choosing the bond geometry of A , B , C , and D . By setting the bond geometry of A' and B' (or C' and D') in the following renormalization step so as to match the geometry of G and H in the previous step, respectively, we can avoid SVD of T' , which introduces additional truncation errors, and continue the renormalization procedure by using SVDs for G and H , which have no truncation error. The modified step is illustrated in Fig. 6. Note that an explicit partial SVD of T' requires the computation cost of $O(\chi^5)$, while in the modified procedure the cost is reduced to $O(\chi^4)$ that includes SVDs of G and H , and that of the intermediate two-bond tensor, K , defined as $K = S_{\{G\}} V_{\{G\}}^\dagger U_{\{H\}} S_{\{H\}}$.

2. Calculation of squeezers in step (d)

Next, we explain how to obtain squeezers E and F in step (d) in Fig. 1. In this step, as we explained, we want to prepare squeezers E and F that satisfy Eqs. (10)–(15). In general, such squeezers should minimize $\|LR - LFER\|^2$, where $L = YD$, $R = AX$, and $\|\cdot\|$ denotes the Frobenius norm, and thus they can be obtained from the partial SVD of LR as

$$E = S_{\{LR\}}^{-\frac{1}{2}} U_{\{LR\}}^\dagger L, \quad (\text{A1})$$

$$F = R V_{\{LR\}} S_{\{LR\}}^{-\frac{1}{2}}, \quad (\text{A2})$$

(Fig. 7) [39]. The partial SVD of LR can be performed by $O(\chi)$ -times matrix-vector multiplications. Therefore, the computation cost of this process is $O(\chi^5)$ in two dimensions.

In higher dimensions, the computation cost of the above procedure can be reduced from $O(\chi^{2d+1})$ to $O(\chi^{\max(d+3,7)})$, as shown in Fig. 8, where A , X , Y , and D in Fig. 1 are represented as three-bond tensors with bond dimensions χ^{d-1} , χ , χ . In Fig. 8, before making squeezers, SVD or the QR decomposition is performed for each tensor. Furthermore,

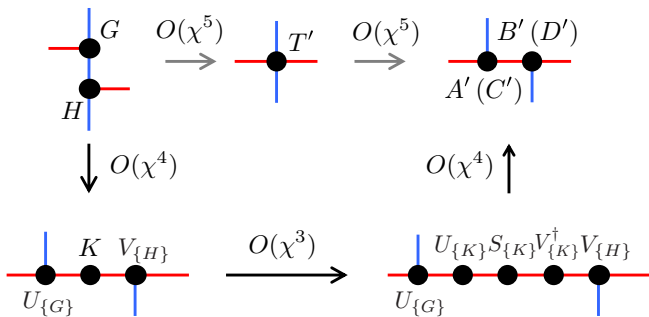


FIG. 6. Reduction of SVD computation cost for the two-dimensional square lattice model. Here, $K = S_{\{G\}} V_{\{G\}}^\dagger U_{\{H\}} S_{\{H\}}$.

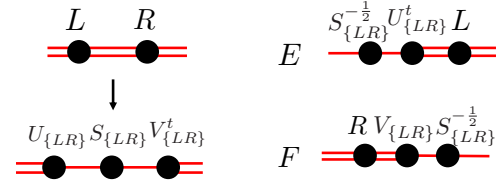


FIG. 7. Preparation of squeezers E and F in step (d) in Fig. 1. These squeezers E and F are chosen so as to minimize $\|LR - LFER\|^2$.

the matrix-vector multiplications in the partial SVD can be performed, as in Fig. 9, without making a full matrix or $\chi^4 \times \chi^4$. By introducing these preprocesses, the cost of the partial SVD for making squeezers in steps (d) and (e) in Fig. 2 is reduced to $O(\chi^{\max(d+3,7)})$, which means the cost of SVDs for making squeezers becomes subleading for $d > 3$. Although the cost for the contraction of tensors at the final stage in Fig. 8 remains $O(\chi^{2d-1})$, reduction of the cost of SVD is essential as the prefactor in the cost of the partial SVD is typically much larger than that for the contraction.

3. Swapping bonds in step (c)

Finally, we discuss the computation cost for swapping bonds [step (c)]. The cost of this procedure can be reduced to $O(\chi^{d+3})$. In addition, the memory footprint can also be reduced from $O(\chi^{2d})$ to $O(\chi^{d+1})$. In Fig. 10, we show the procedure to reduce memory in step (b) and (c) in Fig. 1 in d dimensions. The upper sequence in Fig. 10 is the original procedure shown in Fig. 1, where the memory footprint of the intermediate tensor is $O(\chi^{2d})$. In the partial SVD based on the Arnoldi method, the most time-consuming part is the matrix-vector multiplication. The matrix-vector multiplication can be factorized into two successive tensor-vector multiplications, as shown in the lower row of Fig. 1. In this way, generating a large intermediate tensor can be avoided. This

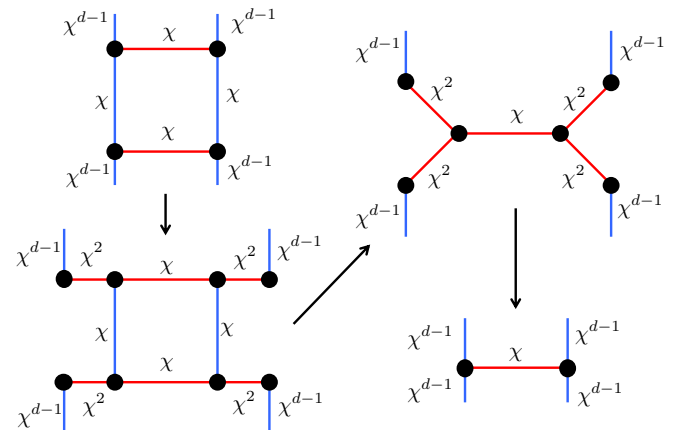


FIG. 8. Reduction of SVD cost in step (d) and (e) in Fig. 2 by using SVD or the QR decomposition for $d > 3$. The partial SVD from lower left to upper right can be performed in $O(\chi^7)$ without making a full matrix of $\chi^4 \times \chi^4$ by using the factorization technique shown in Fig. 9.

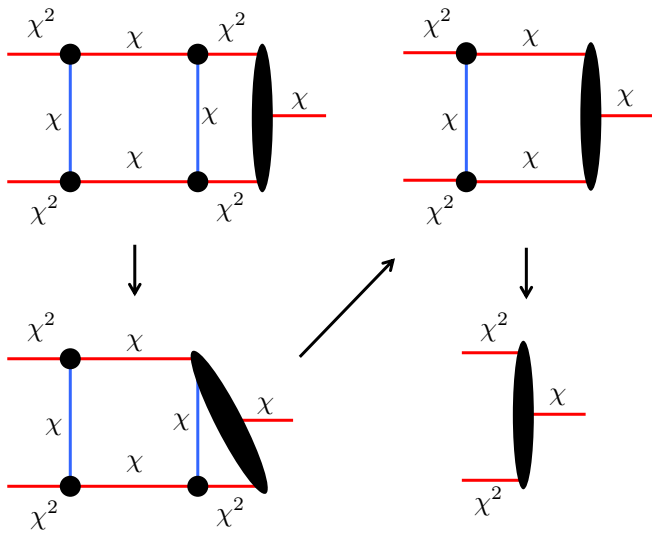


FIG. 9. Factorization of matrix-vector multiplication in the partial SVD to make squeezers in Fig. 8. To obtain χ singular values and corresponding isometries, we need χ -time matrix-vector multiplications, which is manifested by an additional bond with bond dimension χ to the rightmost vector.

procedure requires only the memory footprint of $O(\chi^{d+1})$ (to store χ vectors of length χ^d) and the computation cost of $O(\chi^{d+3})$.

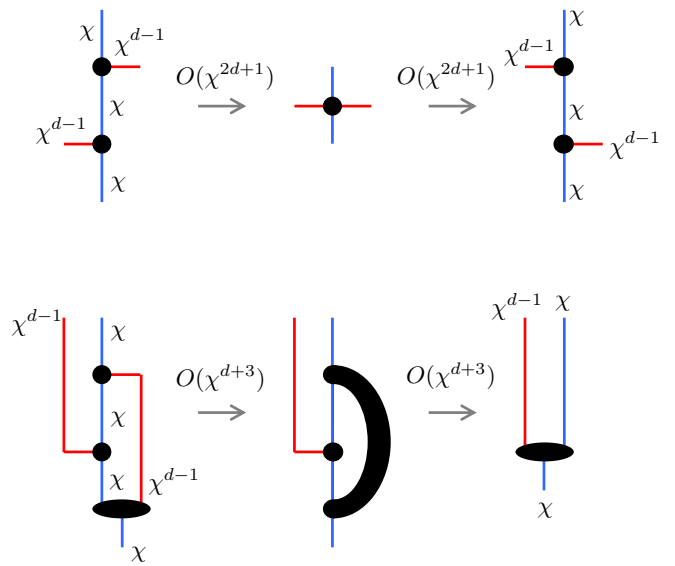


FIG. 10. Factorization of matrix-vector multiplication in SVD. The upper row is the original procedure [step (b) and (c)] presented in Fig. 1. The memory footprint of the intermediate tensor is $O(\chi^{2d})$. In the lower row, the matrix-vector multiplication in the partial SVD is factorized in two successive tensor-vector multiplications. This procedure reduces the memory footprint from $O(\chi^{2d})$ to $O(\chi^{d+1})$ and also reduces the computation cost from $O(\chi^{2d+1})$ to $O(\chi^{d+3})$. Similar to Fig. 9, the vector at the bottom have the bond with bond dimension χ .

- [1] M. Levin and C. P. Nave, *Phys. Rev. Lett.* **99**, 120601 (2007).
- [2] Z. Y. Xie, J. Chen, M. P. Qin, J. W. Zhu, L. P. Yang, and T. Xiang, *Phys. Rev. B* **86**, 045139 (2012).
- [3] Z.-C. Gu and X.-G. Wen, *Phys. Rev. B* **80**, 155131 (2009).
- [4] Z. Y. Xie, H. C. Jiang, Q. N. Chen, Z. Y. Weng, and T. Xiang, *Phys. Rev. Lett.* **103**, 160601 (2009).
- [5] G. Evenbly and G. Vidal, *Phys. Rev. Lett.* **115**, 180405 (2015).
- [6] G. Evenbly, *Phys. Rev. B* **95**, 045117 (2017).
- [7] M. Bal, M. Mariën, J. Haegeman, and F. Verstraete, *Phys. Rev. Lett.* **118**, 250602 (2017).
- [8] S. Yang, Z.-C. Gu, and X.-G. Wen, *Phys. Rev. Lett.* **118**, 110504 (2017).
- [9] M. Hauru, C. Delcamp, and S. Mizera, *Phys. Rev. B* **97**, 045111 (2018).
- [10] K. Harada, *Phys. Rev. B* **97**, 045124 (2018).
- [11] S. Morita, R. Igarashi, H.-H. Zhao, and N. Kawashima, *Phys. Rev. E* **97**, 033310 (2018).
- [12] C. Wang, S.-M. Qin, and H.-J. Zhou, *Phys. Rev. B* **90**, 174201 (2014).
- [13] H.-H. Zhao, Z.-Y. Xie, T. Xiang, and M. Imada, *Phys. Rev. B* **93**, 125115 (2016).
- [14] G. Evenbly, *Phys. Rev. B* **98**, 085155 (2018).
- [15] A. J. Ferris, *Phys. Rev. B* **87**, 125139 (2013).
- [16] W. Li, S.-S. Gong, Y. Zhao, S.-J. Ran, S. Gao, and G. Su, *Phys. Rev. B* **82**, 134434 (2010).
- [17] H. H. Zhao, Z. Y. Xie, Q. N. Chen, Z. C. Wei, J. W. Cai, and T. Xiang, *Phys. Rev. B* **81**, 174411 (2010).
- [18] Q. N. Chen, M. P. Qin, J. Chen, Z. C. Wei, H. H. Zhao, B. Normand, and T. Xiang, *Phys. Rev. Lett.* **107**, 165701 (2011).
- [19] B. Dittrich and F. C. Eckert, *J. Phys.: Conf. Ser.* **360**, 012004 (2012).
- [20] H. C. Jiang, Z. Y. Weng, and T. Xiang, *Phys. Rev. Lett.* **101**, 090603 (2008).
- [21] G. Evenbly and G. Vidal, *Phys. Rev. Lett.* **116**, 040401 (2016).
- [22] Y. Meurice, *Phys. Rev. B* **87**, 064422 (2013).
- [23] S. Wang, Z.-Y. Xie, J. Chen, N. Bruce, and T. Xiang, *Chin. Phys. Lett.* **31**, 070503 (2014).
- [24] J. F. Yu, Z. Y. Xie, Y. Meurice, Y. Liu, A. Denbleyker, H. Zou, M. P. Qin, J. Chen, and T. Xiang, *Phys. Rev. E* **89**, 013308 (2014).
- [25] H. Ueda, K. Okunishi, and T. Nishino, *Phys. Rev. B* **89**, 075116 (2014).
- [26] J. Genzor, A. Gendiar, and T. Nishino, *Phys. Rev. E* **93**, 012141 (2016).
- [27] L.-P. Yang, Y. Liu, H. Zou, Z. Y. Xie, and Y. Meurice, *Phys. Rev. E* **93**, 012138 (2016).
- [28] H. Kawauchi and S. Takeda, *Phys. Rev. D* **93**, 114503 (2016).
- [29] R. Sakai, S. Takeda, and Y. Yoshimura, *Prog. Theor. Exp. Phys.* **2017**, 063B07 (2017).
- [30] Y. Yoshimura, Y. Kuramashi, Y. Nakamura, S. Takeda, and R. Sakai, *Phys. Rev. D* **97**, 054511 (2018).
- [31] S. Akiyama, Y. Kuramashi, T. Yamashita, and Y. Yoshimura, *Phys. Rev. D* **100**, 054510 (2019).

- [32] W. E. Arnoldi, *Q. Appl. Math.* **9**, 17 (1951).
- [33] N. Halko, P. G. Martinsson, and T. P. A., *SIAM Rev.* **53**, 217 (2011).
- [34] Y. Nakamura, H. Oba, and S. Takeda, *Phys. Rev. B* **99**, 155101 (2019).
- [35] L. Onsager, *Phys. Rev.* **65**, 117 (1944).
- [36] Y. Deng and H. W. J. Blöte, *Phys. Rev. E* **68**, 036125 (2003).
- [37] M. Hasenbusch, *Phys. Rev. B* **82**, 174433 (2010).
- [38] J. Kaupužs, R. V. N. Melnik, and J. Rimšāns, *Int. J. Mod. Phys. C* **28**, 1750044 (2017).
- [39] P. Corboz, T. M. Rice, and M. Troyer, *Phys. Rev. Lett.* **113**, 046402 (2014).