Corner transfer matrix renormalization group analysis of the two-dimensional dodecahedron model

Hiroshi Ueda,[1,2] Kouichi Okunishi,[3] Seiji Yunoki,[1,4,5] and Tomotoshi Nishino[6]
[1]*Computational Materials Science Research Team, RIKEN Center for Computational Science (R-CCS), Kobe 650-0047, Japan*
[2]*JST, PRESTO, Kawaguchi, 332-0012, Japan*
[3]*Department of Physics, Niigata University, Niigata 950-2181, Japan*
[4]*Computational Condensed Matter Physics Laboratory, RIKEN Cluster for Pioneering Research (CPR), Wako, 351-0198, Japan*
[5]*Computational Quantum Matter Research Team, RIKEN Center for Emergent Matter Science (CEMS), Wako, 351-0198, Japan*
[6]*Department of Physics, Graduate School of Science, Kobe University, Kobe 657-8501, Japan*

(Received 20 April 2020; accepted 3 September 2020; published 18 September 2020)

We investigate the phase transition of the dodecahedron model on the square lattice. The model is a discrete analog of the classical Heisenberg model, which has continuous $O(3)$ symmetry. In order to treat the large on-site degree of freedom $q = 20$, we develop a massively parallelized numerical algorithm for the corner transfer matrix renormalization group method, incorporating EigenExa, the high-performance parallelized eigensolver. The scaling analysis with respect to the cutoff dimension reveals that there is a second-order phase transition at $T_c = 0.4398(8)$ with the critical exponents $\nu = 2.88(8)$ and $\beta = 0.21(1)$. The central charge of the system is estimated as $c = 1.99(6)$.

DOI: 10.1103/PhysRevE.102.032130

I. INTRODUCTION

Clarification of the role of local symmetry in phase transition is important for the fundamental understanding of critical phenomena. Two-dimensional (2D) polyhedron models have been attracting theoretical interest, in particular in their variety of phase transitions. The models are discrete analogs of the classical Heisenberg model, which has continuous $O(3)$ symmetry. The polyhedron models are described by the pairwise ferromagnetic interaction $h_{s\sigma} = -\boldsymbol{v}^{(s)} \cdot \boldsymbol{v}^{(\sigma)}$ between neighboring sites, where $\boldsymbol{v}^{(s)}$ with $1 \leqslant s \leqslant q$ represents the unit-vector spin directing one of the q vertices of the polyhedron. Figure 1 shows the pictorial representation of the dodecahedron model, where $q = 20$.

The regular polyhedron models on the square lattice have been intensively studied, and it has been revealed that each of them has a characteristic phase transition. The tetrahedron model ($q = 4$) can be mapped to a four-state Potts model [1], and it exhibits a second-order transition with logarithmic correction [2,3]. The octahedron model ($q = 6$) exhibits a weak first-order phase transition [4,5], whose latent heat is close to that of the five-state Potts model [6]. The cube model ($q = 8$) can be trivially mapped to three sets of Ising models, in the same manner as the square model corresponds to two sets [7]. Recent numerical studies of the icosahedron model ($q = 12$) clarified that the model exhibits a continuous phase transition [4,8,9], whose universality class may not be explained by the minimal unitary models in the conformal field theories (CFTs). Curiously, for the dodecahedron model ($q = 20$), the possibility of an intermediate phase was suggested by Monte Carlo simulations in Refs. [10] and [11], whereas a single second-order transition was suggested by other Monte Carlo simulations in Ref. [8]. In this article, we investigate the dodecahedron model to resolve the unclear situation. This is a small step to answer the question how these discrete symmetry models can approximate the classical Heisenberg model, which has no order in finite temperature [12].

An efficient numerical method for the investigation of 2D statistical models is the corner transfer matrix renormalization group (CTMRG) method [13–15], which is a typical tensor network method based on Baxter's corner-transfer matrix (CTM) formalism [16–18]. In the CTMRG, the area of CTMs and the half of row-to-row (column-to-column) transfer matrices are iteratively extended in combination with their low-rank approximation to maintain the matrix size within a certain cutoff dimension m. The numerical accuracy of the method is good even for small m, while its computational cost is proportional to $O(m^3 q^3)$ [19]. Thus, the CTMRG method enables us to obtain precise numerical data with the use of a realistic computational resource, even for the polyhedron models with large on-site degrees of freedom. However, we also noted that the computational cost required for the dodecahedron model ($q = 20$) is about 20 times larger than that of the icosahedron model ($q = 12$). We therefore develop a massively parallelized algorithm for the CTMRG method by means of the message-passing interface (MPI) [20], combined with the numerical diagonalization package EigenExa [21,22], which is also MPI parallelized.

In the previous study of the icosahedron model ($q = 12$) [9], the calculations were performed up to $m = 500$. Critical exponents associated with magnetization M and correlation length ξ are estimated by means of the finite m-scaling analysis [9,23–27]. The central charge c is also extracted from the finite-m scaling applied to the entanglement entropy S_{E}. It was suggested that the model exhibits the second-order transition with a nontrivial central charge $c = 1.90(2)$. Thus, a focus in

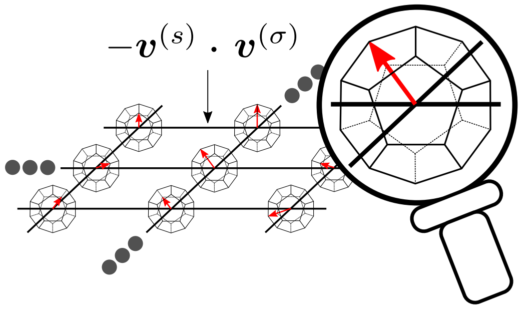

FIG. 1. Dodecahedron model on the square lattice. Each unit vector spin points to one of the 20 vertices of the dodecahedron.

the study of the dodecahedron model ($q = 20$) is the nature of the phase transition. If it is second-order, what is the value of c? In this article we perform finite m-scaling analysis for the dodecahedron model up to $m = 800$.

This article is organized as follows. In the next section, we briefly explain the outline of the CTMRG method applied to polyhedron models. In Sec. III we explain a parallelization technique implemented for the CTMRG method when it is combined with EigenExa. We benchmark the numerical program on the K computer, which was operated at RIKEN R-CCS, through the test application on the icosahedron model. In Sec. IV we show temperature dependencies of the spontaneous magnetization and the entanglement entropy. We perform the finite-m scaling analysis in association with the effective correlation length induced by the finite cutoff effect. The conclusions are summarized in Sec. V, and the role of dodecahedral symmetry is discussed.

II. CORNER TRANSFER MATRIX FORMALISM

We represent the regular polyhedron model on the square lattice in terms of the 2D tensor network, which is written as the contraction among four-leg "vertex" tensors. Let us consider q-state vector spins $\boldsymbol{v}^{(s)}$, $\boldsymbol{v}^{(\sigma)}$, $\boldsymbol{v}^{(s')}$, and $\boldsymbol{v}^{(\sigma')}$ of unit length, which are located at each corner of a unit square on the lattice. The local energy associated with these vector spins is written as the sum of pairwise ferromagnetic interactions

$$E_{s\sigma s'\sigma'} = h_{s\sigma} + h_{\sigma s'} + h_{s'\sigma'} + h_{\sigma' s}, \tag{1}$$

where $h_{s\sigma}$ denotes $-\boldsymbol{v}^{(s)} \cdot \boldsymbol{v}^{(\sigma)}$ as we introduced in the previous section. We have chosen the interaction parameter as unity. The corresponding Boltzmann weight

$$W_{s\sigma s'\sigma'} = \exp\left[-\frac{E_{s\sigma s'\sigma'}}{k_{\rm B}T}\right] \tag{2}$$

can be regarded as the local four-leg vertex tensor [28], where $k_{\rm B}$ is the Boltzmann constant, and T is the thermodynamic temperature. Throughout this article we choose the temperature scale where $k_{\rm B} = 1$. It should be noted that the vertex tensors are defined on every other unit square on the lattice. The product over all the vertex tensors contained in the system represents the Boltzmann weight for the entire system under a specific spin configuration. Taking the configuration sum for this weight, we obtain the partition function Z.

In the CTM formalism [16–18], a finite-size system with square geometry is considered. The partition function Z is then represented as

$$Z \equiv {\rm Tr}\, C^4, \tag{3}$$

where C denotes the CTM corresponding to each quadrant of the finite-size system. We have used the fact that C is real symmetric, since $W_{s\sigma s'\sigma'}$ defined in Eq. (2) is invariant under rotation and spacial inversions of indices. In this article, we assume the ferromagnetic boundary condition in order to choose one of the q types of the ordered state, where all the vector spins at the system boundary point to the specified direction $s = 1$.

In the CTMRG method [13–15], we recursively update C and the half row-to-row or half column-to-column transfer matrices P toward their bulk fixed point. Thus the fixed boundary condition can be imposed just fixing the boundary spins in the initial transfer matrices. In order to prevent the exponential blow-up of the matrix dimension, these matrices are successively compressed by means of the truncated orthogonal transformations, which are obtained from the diagonalization of C. In this renormalization group (RG) process, the number of "kept" eigenvalues m plays the role of the cutoff dimension [29,30].

After a sufficient number of iterations in the CTMRG calculation, we obtain the fixed point matrices $\tilde{C}$ and $\tilde{P}$, which are dependent on both T and m. It is convenient to create the normalized density matrix

$$\tilde{\rho} \equiv \frac{\tilde{C}^4}{{\rm Tr}\, \tilde{C}^4} \tag{4}$$

for the evaluation of one-point functions. Spontaneous magnetization in the thermodynamic limit can be approximately obtained as

$$M(T, m) = {\rm Tr}\left[\boldsymbol{v}^{(1)} \cdot \boldsymbol{v}^{(s)}\, \tilde{\rho}\right], \tag{5}$$

where $\boldsymbol{v}^{(s)}$ is the vector spin located at the center of the system. The entanglement entropy

$$S_{\rm E}(T, m) = -{\rm Tr}\, \tilde{\rho} \ln \tilde{\rho} \tag{6}$$

is essential for the determination of the central charge c. In addition to these one-point functions, we can calculate the effective correlation length $\xi_{\rm e}(T, m)$ by diagonalizing the renormalized row-to-row transfer matrices reconstructed from $\tilde{P}$. These physical functions are dependent on m, and therefore we have to take the extrapolation $m \rightarrow \infty$ by any means, which we consider in Sec. IV.

III. PARALLEL COMPUTATION

In this section we explain the massively parallelized numerical algorithm, which is implemented to the CTMRG method. The incorporation of the parallelized diagonalization routine "EigenExa" [21] is essential in this computational programming. To readers who do not care about numerics, we recommend skipping this part and proceeding to the next section.

Using MPI [20], we distribute all the elements of large-scale matrices to n processes along "the 1 × 1 2D block-cyclic

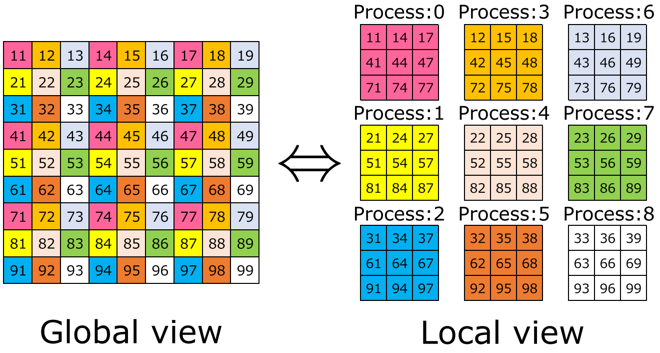

FIG. 2. The 1 × 1 2D block-cyclic distribution for the 9 × 9 matrix with 3 × 3 processes.

distribution" shown in Fig. 2, where n is the number of processes in MPI. We can then employ the PDGEMM routine contained in "the Basic Linear Algebra Communication Subprograms" (BLACS) package [31] for the matrix-matrix multiplication, and we can also employ the EigenExa package for the diagonalization of CTMs. Both of these linear numerical procedures support the block-cyclic distribution.

To achieve high performance in matrix-matrix multiplications, we often encounter the situation where reordering of tensor indices is necessary. Suppose that we have a four-leg tensor A_{pqrs}, and that we have to store the elements to another one $B_{prqs} := A_{pqrs}$, where ":=" denotes substitution from the right to the left. This reordering can be quickly done even under the block-cyclic distribution, as it is abbreviated in the numerical pseudocode Algorithm 1. For the legs $p, q, r,$ and s, respectively, we denote their leg dimension by $a, b, c,$ and d. In the algorithm, the four-leg tensor A_{pqrs} is represented as a matrix $A_{i,j}$ with the use of combined indices $i := p + a(q-1)$ and $j := r + c(s-1)$. Such an "addressing" is often used in the tensor-network frameworks. Note that the symbol MPI_Alltoallv in line 2 denotes the address management—to arrange which tensor elements should be stored in which array address under which process—in MPI. This management enables the substitution of tensor elements consistent with the block-cyclic distribution. In addition to the substitution $B_{prqs} := A_{pqrs}$, another type of reordering $B_{prq} := A_{pqr}$ between three-leg tensors is often necessary. This process is represented by the pseudocode Algorithm 2.

ALGORITHM 1. Permutation of middle two-leg indices for a four-leg tensor in the matrix representation.

```
Require:
    Input: positive integer $a, b, c, d$; real $\mathbf{A} = \{A_{ij}\}_{1 \leqslant j \leqslant cd}^{1 \leqslant i \leqslant ab}$
    $1 \leqslant p \leqslant a$; $1 \leqslant q \leqslant b$; $1 \leqslant r \leqslant c$; $1 \leqslant s \leqslant d$
Ensure:
    Output: real $\mathbf{B} = \{B_{\gamma\zeta}\}_{1 \leqslant \zeta \leqslant bd}^{1 \leqslant \gamma \leqslant ac}$
    ▷ Matrices A and B are distributed to $n$ processes using the
    $1 \times 1$ 2D block-cyclic distribution.
1: function P-INDEX (A,$a$,$b$,$c$,$d$)
2:     $\{B_{p+a(r-1),q+b(s-1)}\} := \{A_{p+a(q-1),r+c(s-1)}\}$
                                                   ▷ Using MPI_Alltoallv
3:     return B
4: end function
```

ALGORITHM 2. Permutation of the last two indices for a three-leg tensor in the matrix representation.

```
Require:
    Input: positive integer $a, b, c$; real $\mathbf{A} = \{A_{ij}\}_{1 \leqslant j \leqslant c}^{1 \leqslant i \leqslant ab}$
    $1 \leqslant p \leqslant a$; $1 \leqslant q \leqslant b$; $1 \leqslant r \leqslant c$
Ensure:
    Output: real $\mathbf{B} = \{B_{\gamma\zeta}\}_{1 \leqslant \zeta \leqslant cb}^{1 \leqslant \gamma \leqslant a}$
    ▷ Matrices A and B are distributed to $n$ processes using the
    $1 \times 1$ 2D block-cyclic distribution.
1: function P-INDEX2(A,$a$,$b$,$c$)
2:     $\{B_{p,r+c(q-1)}\} := \{A_{p+a(q-1),r}\}$          ▷ Using MPI_Alltoallv
3:     return B
4: end function
```

Generally speaking, the number of processes n and the dimensions of tensor legs can vary during numerical calculations; therefore in principle the allocation managements should be performed dynamically. In the case of the CTMRG calculation, however, the maximum dimensions of all the matrices are always qm. Thus, we can make lists for the address management in advance to reduce communication complexity in MPI.

Combining Algorithms 1 and 2 and EigenExa, we can construct the CTMRG algorithm that is MPI parallelized. In Algorithm 3 we present the resulting pseudocode for a lattice model that is invariant under 90° rotation. The main loop contains four MPI_Alltoallv communications with the cost $O(m^2q^2)$, five matrix-matrix multiplications labeled by PDGEMM with the cost $O(m^3q^2 + m^2q^4)$, and EigenExa with the cost $O(m^3q^3)$. Thus, in this algorithm, EigenExa could be the numerical bottle neck. Note that Algorithm 3 is executable on any standard computer if MPI is implemented and if EigenExa is replaced by a matrix diagonalization package such as PDSYEVD in ScaLAPACK [32].

We check the performance of Algorithm 3 by means of a benchmark computation applied to the icosahedron model ($q = 12$) at the critical temperature [9]. Figure 3(a) shows the elapsed time t for single iteration in the CTMRG method with respect to n, the number of nodes used, up to $n = 16\,380$ ($= 130 \times 126$). All calculations were performed on the K computer (CPU: eight-core SPARC64 VIIIfx) installed at RIKEN R-CCS. If the maximum matrix dimension $N = mq$ is much larger than n, the elapsed time decreases with respect to n, implying that the parallelization properly works. For $n \gtrsim N/10$, however, the parallelization efficiency saturates, where the MPI communication time among the nodes becomes non-negligible.

We examine a scaling hypothesis given by

$$t = N^3 n^{-1} F(nN^{-K}) \tag{7}$$

in order to capture the relation among t, N, and n. The scaling function $F(y)$ has the asymptotic forms $F(y) \sim y^{3/K}$ for $y \gg 1$, namely, $t \sim n^{-1+3/K}$, and $F(y) \sim$ const for $y \to 0$. Under the ideal MPI parallelization, the exponent K could be three, but it is empirically less than that in practical computations.

ALGORITHM 3. Main part of the CTMRG calculation for a vertex model with the fixed boundary condition

```
Require:
    Input: positive integer $L$, $q$, and $m$; real $T$ and $\epsilon$
    $2 \leqslant q \leqslant m$; $0 < \epsilon \ll 1$
    $1 \leqslant i \leqslant q^2$; $1 \leqslant j \leqslant q^2$
    $1 \leqslant \alpha \leqslant m$; $1 \leqslant \beta \leqslant mq$
    $1 \leqslant s \leqslant q$; $1 \leqslant \sigma \leqslant q$
Ensure:
    Output: real $S_E$
    $0 \leqslant S_E \leqslant \ln m$
function SYMMETRIC-CTMRG($L,q,m,T,\epsilon$)
    $k := 1$; $S := 0$                                        ▷ initialization
    $\mathbf{P} = \{p_{\alpha\beta}\}$; $p_{\alpha\beta} := \begin{cases} 1 & \alpha = \beta = 1 \\ 0 & \text{otherwise} \end{cases}$        ▷ initialization.
    $\mathbf{\Omega} = \{\omega_\beta\}$; $\omega_\beta := \begin{cases} 1 & \beta = 1 \\ 0 & \text{otherwise} \end{cases}$        ▷ initialization.
    $\mathbf{U} = \{u_{\beta\beta}\} := 0$                        ▷ initialization.
    $\mathbf{W} = \{w_{ij}\}$; $w_{s+q(\sigma-1),s'+q(\sigma'-1)} := W_{s\sigma s'\sigma'}$
                                   ▷ local Boltzmann weight in Eq. (2)
    while $k \leqslant L \wedge c \geqslant \epsilon$ do ▷ CTMRG iteration
        $\mathbf{P}' := \{\omega_\alpha p_{\alpha\beta}\}$
        $\mathbf{C} :=$ SUB-CTMRG($\mathbf{P}$, $\mathbf{P}'$, $\mathbf{W}$)
        $(\{u_{\gamma\gamma'}\}, \{w_\gamma\}) :=$ EigenExa($\{c_{\gamma\gamma'}\}$)
            with $1 \leqslant \gamma \leqslant \min(q^k, mq)$ ▷ diagonalization
        $\mathbf{U}' :=$ P-INDEX2($\{u_{\beta\alpha}\}, m, q, m$)
        $\mathbf{X} :=$ SUB-CTMRG($\mathbf{P}$, $\mathbf{U}'$, $\mathbf{W}$)
        $\mathbf{P} := \{u_{\beta\alpha}\}^t\mathbf{X}$                ▷ Using PDGEMM
        $\mathbf{P} := \mathbf{P}/\max_{\alpha\beta}|p_{\alpha\beta}|$            ▷ normalization
        $\mathbf{\Omega} := \mathbf{\Omega}/\sqrt[4]{\sum_\beta \omega_\beta^4}$              ▷ normalization
        $S_E := -\sum_\beta \omega_\beta^4 \ln \omega_\beta^4$          ▷ equivalent to Eq. (6)
        $c := |1 - S_E/S|$
        $S := S_E$
        $k := k + 1$
    end while
    return $S_E$
end function

function SUB-CTMRG(P,P',W)
    $\mathbf{X}_1 := \mathbf{P}^t\mathbf{P}'$                        ▷ Using PDGEMM
    $\mathbf{X}_2 :=$ P-INDEX($\mathbf{X}_1, m, q, m, q$)
    $\mathbf{X}_3 := \mathbf{X}_2\mathbf{W}$                         ▷ Using PDGEMM
    $\mathbf{X}_1 :=$ P-INDEX($\mathbf{X}_3, m, m, q, q$)
    return $\mathbf{X}_1$
end function
                    ▷ All matrices are distributed to $n$ processes using
    the $1 \times 1$ 2D block-cyclic distribution. Matrices $\mathbf{P}'$, $\mathbf{U}'$, $\mathbf{X}$, $\mathbf{X}_2$,
    and $\mathbf{X}_3$ are working arrays. The 2D arrays $\mathbf{C}$ and $\mathbf{X}$ can share the
    common physical memory in this algorithm.
```

For the estimation of K, we invoke the benchmark data in EigenExa with $N = 1 \times 10^4$, 5×10^4, and 1.3×10^5, which are available on the EigenExa web page [22]. Performing the polynomial fitting to the scaling form in Eq. (7), we obtain $K = 1.66$. Assuming that the data shown in Fig. 3(a) share the same exponent, we show the scaling plot for all the benchmark data in Fig. 3(b). The plotted points almost collapse on a certain scaling curve, and the result supports the fact that the diagonalization of CTMs with EigenExa is certainly the numerical bottleneck.

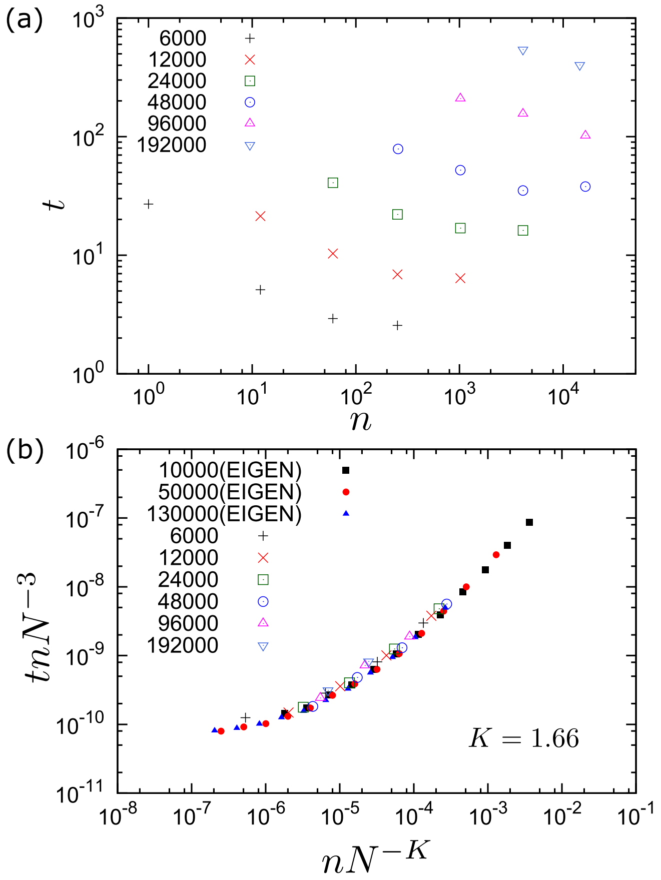

FIG. 3. (a) Elapsed time (sec) per single iteration in the parallelized CTMRG algorithm performed by means of the K computer, when the method is applied to the icosahedron model [9]. The holizontal axis denotes the number of nodes n. The maximum matrix dimension $N = mq$ is shown by numbers beside the legends. (b) Scaling plot for computational times required for EigenExa and for the time shown in Fig. 3(a).

IV. SCALING ANALYSIS

We performed the CTMRG calculation for the dodecahedron model, assuming the ferromagnetic boundary conditions. We choose the cutoff dimensions up to $m = 800$ [33] for all the numerical data analyses shown in this section. Figure 4 shows the temperature dependence of the spontaneous magnetization $M(T, m)$. The overall behavior of the magnetization, which exhibits a shoulder-like structure in the region $0.45 \lesssim T \lesssim 0.5$, is very similar to $M(T, m)$ observed in the icosahedron model [9].

We perform the finite-m scaling analysis [9,23–27], in order to check whether the transition is second-order or not. At the fixed point—the large system size limit—of the CTMRG method, the presence of finite cutoff dimension m modifies the intrinsic correlation length $\xi(T)$ to an effective one $\xi_e(T, m)$. At the critical temperature $T = T_c$ the behavior $\xi_e(T_c, m) \sim m^\kappa$ is expected, where κ is a particular exponent [23–25]. Meanwhile, the intrinsic correlation length $\xi(T)$ away from the critical point obeys $\xi(T) \sim |T - T_c|^\nu$, where ν is the exponent characterizing the divergence of the correlation length. Taking account of these relations, we can assume the finite-m

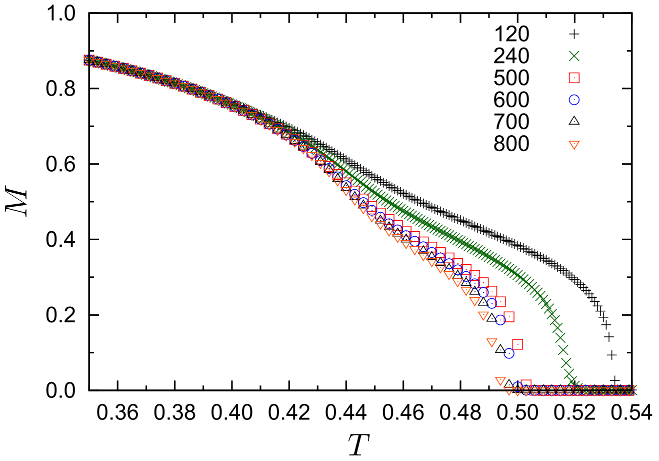

FIG. 4. Temperature dependence of spontaneous magnetization $M(T, m)$.

scaling form

$$\xi_{\mathrm{e}}(T, m) \sim m^{\kappa} f[(T - T_{\mathrm{c}})m^{\kappa/\nu}], \tag{8}$$

where the scaling function behaves as $f(y) \sim |y|^{-\nu}$ for $y \gg 1$ and $f(y) \sim$ const for $y \to 0$. We can also assume the finite-m scaling form

$$M(T, m) \sim m^{-\kappa\beta/\nu} g[(T - T_{\mathrm{c}})m^{\kappa/\nu}] \tag{9}$$

for the spontaneous magnetization, where β denotes the critical exponent for the magnetization, and g is a scaling function. It should be noted that Eqs. (8) and (9) are basically equivalent to the conventional finite-size scalings if we substitute the system size ℓ to $\xi_{\mathrm{e}}(T_{\mathrm{c}}, m) \sim m^{\kappa}$. For the bipartite entanglement entropy, the finite-size scaling form $S_{\mathrm{E}}(T_{\mathrm{c}}, \ell) \sim \frac{c}{6}\log \ell + \mathrm{const}$ suggests that the effective scaling dimension for $e^{S_{\mathrm{E}}(T_{\mathrm{c}}, m)}$ can be expressed as $c/6$ [34,35]. Thus, we can assume the finite-m scaling form

$$e^{S_{\mathrm{E}}(T,m)} \sim m^{c\kappa/6} h[(T - T_{\mathrm{c}})m^{\kappa/\nu}] \tag{10}$$

for the entanglement entropy, where the scaling function behaves as $h(y) \sim |y|^{-\nu}$ for $y \gg 1$ and $h(y) \sim$ const for $y \to 0$.

In order to estimate scaling parameters, we employ the Bayesian scaling analysis proposed in Refs. [36,37], which is based on the Gaussian process regression for a smooth scaling function. We perform the Bayesian fitting of the scaling parameters by varying a range of T and m in input data to determine estimation errors. Moreover, we check the stability of the resulting parameters against corrections to scaling in the Appendix. In the following, we basically present the final results of the scaling parameters in Eqs. (8), (9), and (10).

We empirically find that the analysis on $\xi_{\mathrm{e}}(T, m)$ is more stable than that for $M(T, m)$ and $e^{S_{\mathrm{E}}(T,m)}$. From the calculated $\xi_{\mathrm{e}}(T, m)$ in the temperature range $0.35 \leqslant T \leqslant 0.56$, the values $T_{\mathrm{c}} = 0.4398(8)$, $\nu = 2.88(8)$, and $\kappa = 0.845(4)$ are extracted. Figure 5(a) shows the corresponding scaling plot for $\xi_{\mathrm{e}}(T, m)$, where the data collapse well to a scaling function, which exhibit an intermediate plateau, as was observed in the icosahedron model [9].

Using the obtained T_{c}, ν, and κ, we can further estimate $\beta = 0.21(1)$ by means of the Bayesian analysis applied to $M(T, m)$ shown in Fig. 4. The resulting scaling plot is presented in Fig. 5(b), where the scaling function exhibits the shoulder structure. We finally perform the Bayesian analysis for $e^{S_{\mathrm{E}}(T,m)}$ and estimate the value of central charge $c = 1.99(6)$. The scaling plot in Fig. 5(c) clearly shows that the calculated $e^{S_{\mathrm{E}}(T,m)}$ also collapsed on a scaling function, which exhibits a nontrivial intermediate structure.

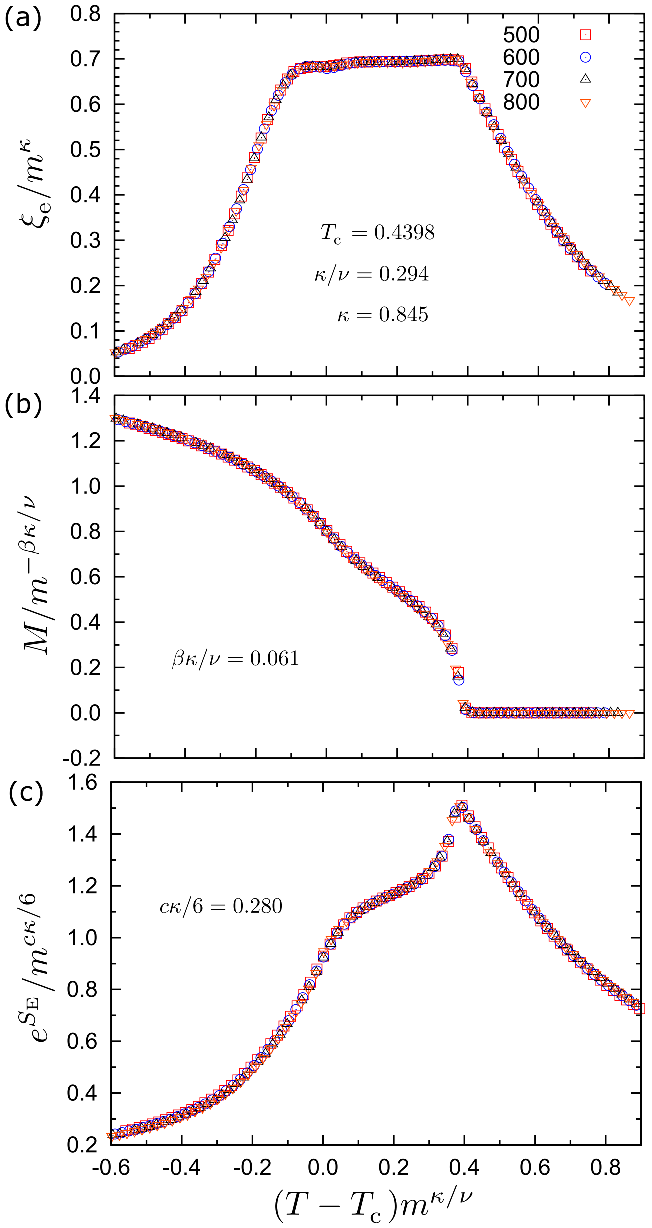

FIG. 5. Scaling plots for (a) effective correlation length $\xi_{\mathrm{e}}(T, m)$ [Eq. (8)], (b) magnetization $M(T, m)$ [Eq. (9)], and (c) the exponential of the entanglement entropy $e^{S_{\mathrm{E}}(T,m)}$ [Eq. (10)]. Note that correction terms to scaling in the Appendix are not included in these scaling plots.

It should be noted that for a 2D classical system at criticality, the central charge c and κ can be related to each other through the nontrivial relation,

$$c\kappa/6 = (1 + \sqrt{12/c})^{-1}, \tag{11}$$

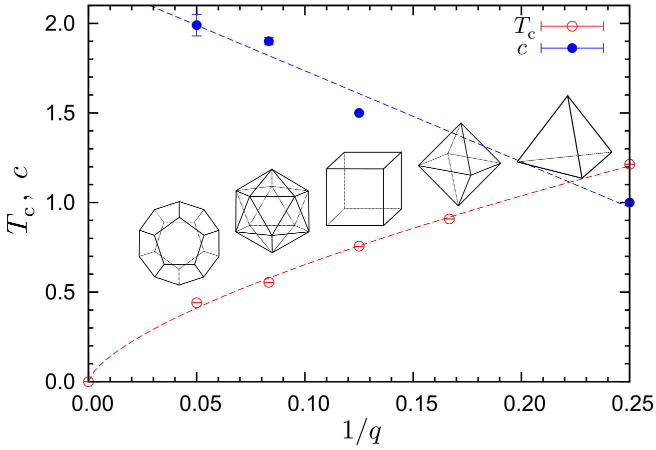

FIG. 6. Critical temperatures and central charges in the regular polyhedron models. Broken lines are guides for the eye.

which was originally derived from the matrix-product-state description of 1D critical quantum systems [26]. The relation is satisfied within the error bars if the above estimations of $c = 1.99$ and $\kappa = 0.845$ are substituted. This fact provides a complemental check to the present finite-m scaling analysis performed on the numerically calculated results. Since the estimated values of the exponents in the dodecahedron model are different from $\nu = 1.62(2)$ and $\beta = 0.12(1)$ in the icosahedron model [9], phase transitions of these two models belong to different universality classes.

V. SUMMARY AND DISCUSSION

We have investigated the phase transition and critical properties of the dodecahedron model on the square lattice, where the vector spin has 20 degrees of freedom ($q = 20$). In order to deal with the large on-site degree of freedom, we developed the massively parallelized CTMRG algorithm cooperating with EigenExa [21,22]. Spontaneous magnetization $M(T, m)$, effective correlation length $\xi_e(T, m)$, and entanglement entropy $S_E(T, m)$ are calculated for the cut-off dimensions m up to 800. The finite-m scaling analyses [9,23–26] around the transition temperature revealed that the model undergoes a single second-order phase transition at $T_c = 0.4398(8)$, which is consistent with the Monte Carlo simulations in Ref. [8]. We also estimated the scaling exponents $\nu = 2.88(8)$, $\beta = 0.21(1)$, and the central charge $c = 1.99(6)$.

Let us summarize the critical temperatures and central charges for the series of regular polyhedron models in Fig. 6. The transition temperature monotonically decreases with respect to the number of on-site degrees of freedom q. The behavior in T_c is consistent with the fact that it converges to zero in the large-q limit, which is the classical Heisenberg model [12]. Note that the octahedron model ($q = 6$) is known to exhibit a weak first-order phase transition. Meanwhile, the central charge monotonically increases with q. The exact value $c = 1$ is known for the tetrahedron model ($q = 4$), which corresponds to the four-state Potts model. Also for the cubic model ($q = 8$), which is nothing but three sets of Ising models, the value $c = 3/2$ is known.

So far, we have no theoretical explanation for the central charges $c = 1.90(2)$ and $c = 1.99(6)$, respectively, for the icosahedron model and dodecahedron model. How can we explain the universality classes of the phase transitions and interpret the intermediate shoulder structures in the scaling functions? In these two models, there are several ways of introducing anisotropy to the vector spins, according to the subgroup structure of the polyhedral symmetry [38]. A preliminary numerical calculation suggests that introduction of XY anisotropy to these models induces KT transitions. A more promising deformation is the introduction of the cubic anisotropy. If the phase transition splits into two different ones subject to different subgroup symmetries, the value of the central charge in each transition would explain the value of c obtained in this study. Complementary, an effective field theoretical treatment within the regular polyhedron symmetry is also a nontrivial future problem.

If we consider polyhedron models in general, in addition to the regular ones, semiregular (or truncated) polyhedron models would be important candidates for the future study of attacking the large-q limit. The pioneering work by Krčmár *et al.* shows that the truncated tetrahedron model exhibits two phase transitions [5]. If we introduce the truncation scheme to the current study, we have to treat the truncated icosahedron, which has 60 on-site degrees of freedom. In a couple of years realistic computation will be possible for this system. At present, the rhombic icosahedron model ($q = 24$) can be the next target of the analysis in the near future.

ACKNOWLEDGMENTS

H.U. thanks Y. Hirota and T. Imamura for helpful comments on EigenExa and S. Morita for discussions about the MPI parallelization. The work was partially supported by KAKENHI No. 26400387, No. 17H02926, No. 17H02931, and No. 17K14359, by JST PRESTO No. JPMJPR1911, and by MEXT as "Challenging Research on Post-K Computer" (Challenge of Basic Science: Exploring the Extremes through Multi-Physics Multi-Scale Simulations). This research used computational resources of the K computer provided by the RIKEN R-CCS through the HPCI System Research project (Project ID: hp160262) and of the HOKUSAI-Great Wave supercomputing system at RIKEN.

APPENDIX: CORRECTIONS TO SCALINGS AND THEIR *m* DEPENDENCES

We present details of the finite-m scaling for the CTMRG results of the dodecahedron model. As mentioned in the main text, a CFT describing the universality class of the dodecahedron model is not specified yet. Thus, it is difficult to directly estimate how the fitting for the leading scaling functions of Eqs. (8), (9), and (10) is stable against correction terms associated with less relevant scaling dimensions. Thus, replacing the system size L with m^κ in the standard finite-size scaling with corrections, we phenomenologically introduce the finite-m scaling functions with correction terms as

TABLE I. Transition temperatures and caling exponents estimated by Eqs. (8)–(10) and Eqs. (A1)–(A3) from data sets A: $m \in \{120, 240, 500, 800\}$ and B: $m \in \{500, 600, 700, 800\}$.

Set	Scaling eqs.	T_c	κ	ν	β	c
A	(8)–(10)	0.4406(2)	0.858(1)	2.92(2)	0.22(1)	1.90(1)
B	(8)–(10)	0.4404(2)	0.842(1)	2.92(1)	0.21(1)	1.96(2)
A	(A1)–(A3)	0.4408(4)	0.844(3)	2.64(5)	0.21(1)	1.99(3)
B	(A1)–(A3)	0.4397(7)	0.845(4)	2.86(6)	0.21(1)	2.00(4)

Set	Scaling eqs.	ω_1	ω_2	ω_3
A	(A1)–(A3)	0.8(1)	1.7(1)	0.4(1)
B	(A1)–(A3)	0.34(2)	1.4(2)	0.7(1)

follows:

$$\xi_e(T,m) \sim m^{\kappa}\{f[(T-T_c)m^{\kappa/\nu}] + m^{-\kappa\omega_1} f_1[(T-T_c)m^{\kappa/\nu}]\}, \tag{A1}$$

$$M(T,m) \sim m^{-\kappa\beta/\nu}\{g[(T-T_c)m^{\kappa/\nu}] + m^{-\kappa\omega_2} g_1[(T-T_c)m^{\kappa/\nu}]\}, \tag{A2}$$

$$e^{S_E(T,m)} \sim m^{c\kappa/6}\{h[(T-T_c)m^{\kappa/\nu}] + m^{-\kappa\omega_3} h_1[(T-T_c)m^{\kappa/\nu}]\} \tag{A3}$$

where f_1, g_1, and h_1 denote scaling functions for correction terms and ω_1, ω_2, and ω_3 are irrelevant exponents.

Let us evaluate the leading scaling parameters in the dodecahedron model by comparing Bayesian scaling analyses [36,37] for Eqs. (8)–(10) and those for Eqs. (A1)–(A3) including the correction terms. Here it should be noted that the fitting results may depend on the range of cutoff dimension m. To check the m dependence, we use two sets of data: one is the set A, $m \in \{120, 240, 500, 800\}$, which contains small m cases, and the other is the set B, $m \in \{500, 600, 700, 800\}$.

Table I summarizes the result of numerical fitting analysis. Since data set A contains small m cases, the estimated κ and c from Eqs. (8)–(10) and ν from Eqs. (A1)–(A3) show a relatively large deviation. Meanwhile, the transition temperature T_c and the exponents κ, ν, β and c obtained from data set B are consistent for the the scaling functions both with and without correction terms. Thus, m in data set B are sufficiently large for the estimation of these values, although the irrelevant exponents ω_1, ω_2, and ω_3 exhibit large m dependencies. Discarding the scaling result from data set A, we obtain the values $T_c = 0.4398(8)$, $\kappa = 0.845(4)$, $\nu = 2.88(8)$, $\beta = 0.21(1)$, $c = 1.99(6)$, which were presented in the main text. We have determined error bars of the final estimation of exponents so as to include the error bars of the fitting results for data set B. Indeed, the scaling plot using the determined exponents in Fig. 5 collapses well to scaling curves.

[1] F. Y. Wu, Rev. Mod. Phys. **54**, 235 (1982).

[2] M. Nauenberg and D. J. Scalapino, Phys. Rev. Lett. **44**, 837 (1980).

[3] J. L. Cardy, N. Nauenberg, and D. J. Scalapino, Phys. Rev. B **22**, 2560 (1980).

[4] A. Patrascioiu and E. Seiler, Phys. Rev. D **64**, 065006 (2001).

[5] R. Krčmár, A. Gendiar, and T. Nishino, Phys. Rev. E **94**, 022134 (2016).

[6] T. Nishino and K. Okunishi, J. Phys. Soc. Jpn. **67**, 1492 (1998).

[7] D. D. Betts, Can. J. Phys. **42**, 1564 (1964).

[8] T. Surungan and Y. Okabe, in *Proceedings of 3rd JOGJA International Conference on Physics* (2012), arXiv:1709.03720.

[9] H. Ueda, K. Okunishi, R. Krčmár, A. Gendiar, S. Yunoki, and T. Nishino, Phys. Rev. E **96**, 062112 (2017).

[10] A. Patrascioiu, J.-L. Richard, and E. Seiler, Phys. Lett. B **241**, 229 (1990).

[11] A. Patrascioiu, J.-L. Richard, and E. Seiler, Phys. Lett. B **254**, 173 (1991).

[12] N. D. Mermin and H. Wagner, Phys. Rev. Lett. **17**, 1133 (1966).

[13] T. Nishino and K. Okunishi, J. Phys. Soc. Jpn. **65**, 891 (1996).

[14] T. Nishino and K. Okunishi, J. Phys. Soc. Jpn. **66**, 3040 (1997).

[15] R. Orus and G. Vidal, Phys. Rev. B **80**, 094403 (2009).

[16] R. J. Baxter, J. Math. Phys. **9**, 650 (1968).

[17] R. J. Baxter, J. Math. Phys. **19**, 461 (1978).

[18] R. J. Baxter, *Ecactly Solved Models in Statistical Mechanics* (Academic Press, London, 1982).

[19] In general, a larger cutoff dimension m is also required for a CTMRG calculation with a large q model.

[20] Message Passing Interface Forum, Intl. J. Supercomput. Appl. High-Perf. Comput. **8**, (3/4) (1994); MPI Forum, https://www.mpi-forum.org/.

[21] T. Sakurai, Y. Futamura, A. Imakura, and T. Imamura, in *Advanced Software Technologies for Post-Peta Scale Computing*, edited by M. Sato (Springer, Singapore, 2019), pp. 37–57.

[22] EigenExa, https://www.r-ccs.riken.jp/labs/lpnctrt/en/projects/eigenexa.

[23] T. Nishino, K. Okunishi, and M. Kikuchi, Phys. Lett. A **213**, 69 (1996).

[24] M. Andersson, M. Boman, and S. Östlund, Phys. Rev. B **59**, 10493 (1999).

[25] L. Tagliacozzo, T. R. de Oliveira, S. Iblisdir, and J. I. Latorre, Phys. Rev. B **78**, 024410 (2008).

[26] F. Pollmann, S. Mukerjee, A. M. Turner, and J. E. Moore, Phys. Rev. Lett. **102**, 255701 (2009).

[27] H. Ueda, K. Okunishi, K. Harada, R. Krčmár, A. Gendiar, S. Yunoki, and T. Nishino, Phys. Rev. E **101**, 062111 (2020).

[28] The term "vertex" used here corresponds to a square on the diagonal square lattice, not to a vertex of the polyhedron.

[29] S. R. White, Phys. Rev. Lett. **69**, 2863 (1992).

[30] S. R. White, Phys. Rev. B **48**, 10345 (1993).

[31] J. Dongarra and R. C. Whaley, A user's guide to the BLACS v1.1, Tech. Rep. No. CS-95-281, Computer Science Dept., University of Tennessee, Knoxville (1995). See also LAPACK Working Note no. 94.

[32] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling,

G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley, *ScaLAPACK Users' Guide* (Society for Industrial and Applied Mathematics, Philadelphia, 1997).

[33] We requested computational time of about a day with 24 × 24 nodes on the K computer, where each node has a process.

[34] G. Vidal, J. I. Latorre, E. Rico, and A. Kitaev, Phys. Rev. Lett. **90**, 227902 (2003).

[35] P. Calabrese and J. Cardy, J. Stat. Mech. (2004) P06002.

[36] K. Harada, Phys. Rev. E **84**, 056704 (2011).

[37] K. Harada, Phys. Rev. E **92**, 012106 (2015).

[38] L. L. Foster, Math. Mag. **63**, 106 (1990).