

Accurate simulation for finite projected entangled pair states in two dimensions

Wen-Yuan Liu,¹ Yi-Zhen Huang,² Shou-Shu Gong,³ and Zheng-Cheng Gu¹

¹*Department of Physics, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong, China*

²*Wilczek Quantum Center and Key Laboratory of Artificial Structures and Quantum Control,
School of Physics and Astronomy, Shanghai Jiao Tong University, Shanghai 200240, China*

³*Department of Physics, Beihang University, Beijing 100191, China*

Based on the scheme of variational Monte Carlo sampling, we develop an accurate and efficient two-dimensional tensor-network algorithm to simulate quantum lattice models. We find that Monte Carlo sampling shows huge advantages in dealing with finite projected entangled pair states, which allows significantly enlarged system size and improves the accuracy of tensor network simulation. We demonstrate our method on the square-lattice antiferromagnetic Heisenberg model up to 32×32 sites, as well as a highly frustrated $J_1 - J_2$ model up to 24×24 sites. The results, including ground state energy and spin correlations, are in excellent agreement with those of the available quantum Monte Carlo or density matrix renormalization group methods. Therefore, our method substantially advances the calculation of 2D tensor networks for finite systems, and potentially opens a new door towards resolving many challenging strongly correlated quantum many-body problems.

PACS numbers:

I. INTRODUCTION

Tensor network states (TNSs) are fundamentally important in modern physics. They provide us a very powerful and efficient way to encode the low-energy physics of complex quantum systems based on their local entanglement structure [1–10], and have led to great success in condensed matter physics and classical statistic physics [11–13], for instance, simulating quantum many-body systems [14–17] and classifying topological phases of quantum matter [18–22]. TNSs have also been extensively applied in many other physical areas including quantum field theories [23–26], quantum circuits [27–31], quantum error corrections [32, 33], holography duality [34, 35], ab-initial calculations [36–38] and others. More recently they have even been expanded to the fields of artificial intelligence such as machine learning [39–45] and language models [46, 47].

Unfortunately, the power of the 2D TNSs, specifically, the projected entangled pair states (PEPS), is seriously hindered due to their complexity, unlike the huge success of one-dimensional matrix product states (MPS) [1, 11, 48]. The first challenge is the extremely expensive computational cost for accurate simulation, whose scaling is at least as high as $O(D^{10})$ on the square lattices in conventional double-layer contraction schemes [5, 49–57], where D is the bond dimension determining the representation ability of PEPS. There are two more difficulties in the practical application of PEPS: general setting of the optimization scheme and evaluation of the expectation values of physical observables such as long-distance correlations [49]. Furthermore, the memory cost is a potential bottleneck because it scales as $O(D^8)$.

To overcome these challenges, various concepts and algorithms have been developed [50–74]. However, the computational cost for accurate PEPS simulation of 2D systems still severely limits the application of PEPS. In particular, as the MPS-based methods strongly suf-

fer from small system size, a breakthrough in PEPS methodology is urgently desired for understanding 2D correlated systems as well as relevant applications such as in machine learning [39–45]. Among various optimization algorithms, variational Monte Carlo (VMC) sampling may provide an elegant framework to overcome the above major difficulties. In the VMC scheme, the computational cost is reduced to $O(D^6)$ and memory cost is reduced to $O(D^4)$, because one only needs to deal with single-layer tensor networks. It also allows a gradient-based method for accurate optimization, and all kinds of physical observables can be evaluated by MC sampling with the aid of massive parallelization [75, 76]. The proof-of-principle combination of VMC and TNS has been demonstrated in one and two dimensions [75, 76]. However, in the last decade this scheme has received little attention [77–80]. Recent studies show one can deal with systems up to 16×16 [78, 79], but it is very challenging to compute larger systems because of the heavy cost of MC sampling. The potential power of this approach is far from clear.

In this paper, we show that the optimized VMC-PEPS method can be very powerful and provides an excellent solution to the above issues. By exploiting the intrinsic advantages of MC sampling, we dramatically speed up the MC sampling efficiency and naturally incorporate it with spin symmetry, resulting in huge improvements for the tensor network calculations. Based on the spin-1/2 J_1-J_2 Heisenberg model on square lattices, we first demonstrate our method in two cases: the unfrustrated case up to 32×32 , which can be unbiasedly simulated by the quantum Monte Carlo (QMC) method [81, 82], and the frustrated case up to 8×28 where QMC fails but the density matrix renormalization group (DMRG) method works well [83]. The obtained results for both frustrated and unfrustrated cases agree excellently with available QMC and DMRG results. Finally, to further demonstrate the power of our method, we simulate the

frustrated case up to 24×24 , where both DMRG and QMC fail. Therefore, our VMC-PEPS method provides a powerful way to resolve many long-standing hard 2D quantum many-body problems as well as machine learning-related problems [39].

The rest of this paper is organized as follows. In Sec.II, we first review the scheme of VMC-PEPS method and introduce the sequentially flipping approach to generate sampling configurations. Next we show an example to compute the ground state of Heisenberg model with 32×32 sites by gradient optimization method, and compare the Heisenberg model results with QMC and analyze the finite size scaling with open boundary conditions. In Sec.III, we apply our algorithm to frustrated models. We first consider the square-lattice frustrated J_1 - J_2 model up to 24×24 sites, and compare the results with DMRG or other available results. Then we present some examples to demonstrate the generality of our method on the frustrated models on triangular and kagome lattices. In Sec.IV, we discuss the advantages of finite PEPS algorithms and potential applications to machine-learning problems.

II. THE ALGORITHM

A. Monte Carlo Sampling

For a quantum spin model on a square lattice with size $N = L_y \times L_x$, the wave function in the form of PEPS with a bond dimension D is given as follows [5]:

$$|\Psi\rangle = \sum_{s_1 \dots s_N=1}^P \text{Tr}(A_1^{s_1} A_2^{s_2} \dots A_N^{s_N}) |s_1 s_2 \dots s_N\rangle, \quad (1)$$

where $A_k^{s_k} = A_k(s_k, l, r, u, d)$ is a rank-5 tensor residing on site k , with one physical index s_k whose degree of freedom is P and four virtual indices l, r, u and d connecting to four nearest-neighbor sites. The dimensions of the virtual indices are D . Without loss of generality, we assume all elements of $A_k^{s_k}$ are real numbers throughout this paper.

For a given TNS, the physical quantities can be computed by using MC sampling over the spin configurations, which was first introduced in Ref. [75, 76], based on MPS or string-bond states. Then this scheme was applied to PEPS[77, 78]. The total energy reads:

$$E_{\text{tot}} = \frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle} = \frac{1}{Z} \sum_S |\Psi(S)|^2 E_{\text{loc}}(S), \quad (2)$$

where the local energy term $E_{\text{loc}}(S)$ is defined as

$$E_{\text{loc}}(S) = \sum_{S'} \frac{\Psi(S')}{\Psi(S)} \langle S' | H | S \rangle. \quad (3)$$

Here $\Psi(S)$ is the coefficient of the configuration $|S\rangle = |s_1 s_2 \dots s_N\rangle$ with the form of

$$\Psi(S) = \langle S | \Psi \rangle = \text{Tr}(A_1^{s_1} A_2^{s_2} \dots A_N^{s_N}), \quad (4)$$

and Z is the normalization factor with $Z = \sum_S |\Psi(S)|^2$.

The energy gradients with respect to tensor elements read

$$\frac{\partial E_{\text{tot}}}{\partial A_{lrud}^{s_k}} = \langle G_{lrud}^{s_k}(S) E_{\text{loc}}(S) \rangle - \langle G_{lrud}^{s_k}(S) \rangle \langle E_{\text{loc}}(S) \rangle, \quad (5)$$

where $\langle \dots \rangle$ denotes the MC average. $G_{lrud}^{s_k}$ is defined as

$$G_{lrud}^{s_k}(S) = \frac{1}{\Psi(S)} \frac{\partial \Psi(S)}{\partial A_{lrud}^{s_k}} = \frac{1}{\Psi(S)} \Delta_{lrud}^{s_k}(S), \quad (6)$$

where $\Delta_{lrud}^{s_k}(S)$ is the element of

$$\Delta^{s_k}(S) = \text{Tr}(\dots A_{k-1}^{s_{k-1}} A_{k+1}^{s_{k+1}} \dots), \quad (7)$$

which is a four-index tensor obtained by contracting a single-layer tensor network, which excludes the tensor $A_k^{s_k}$ located on site k on the fixed configuration $|S\rangle$. Therefore, the energy and its gradients with respect to tensor elements can be evaluated by MC sampling, which can be used to optimize PEPS.

In the MC sampling, we can enforce $\sum_k s_k^z = 0$ for the spin configurations if the ground state of a given spin-1/2 system lives in the $S_{\text{tot}}^z = 0$ sector. Generating configurations plays a crucial role in the efficiency of VMC methods. A popular way to perform the MC sampling is randomly choosing a nearest-neighbor antiparallel spin pair (NNASP) flipping the chosen spin pair to generate trial configurations [84]. Assuming the trial configuration $|S_\beta\rangle$ is obtained by flipping an NNASP which is randomly chosen from the given configuration $|S_\alpha\rangle$, we can get the number of all NNASPs for $|S_\alpha\rangle$ and $|S_\beta\rangle$, supposing they are K_α and K_β , respectively. Because each NNASP of $|S_\alpha\rangle$ will be randomly chosen with the same chance $1/K_\alpha$, the trial configuration $|S_\beta\rangle$ will be accepted with Metropolis' probability

$$P_r = \min \left[1, \frac{|\Psi(S_\beta)|^2 / K_\beta}{|\Psi(S_\alpha)|^2 / K_\alpha} \right]. \quad (8)$$

The next necessary step is to pick a random number r from a uniform distribution on the interval $[0, 1)$ and compare it with P_r . If $r < P_r$, $|S_\beta\rangle$ will be accepted as a new configuration; otherwise $|S_\beta\rangle$ will be rejected and the original configuration $|S_\alpha\rangle$ will be kept. Normally we define a MC sweep as N such flip attempts where N is system size, so that all spins are flipped on average, which can dramatically reduce autocorrelation lengths [84]. The physical quantities such as energy will be measured after each MC sweep. It is easy to verify that the leading cost for generating MC sampling configurations is $O(N^2 D^4 D_c^2)$ in each MC sweep, where N is the system size and D_c is the cutoff bond dimension during the contraction process.

To accelerate the MC sampling, here we introduce a fast configuration-generating method with cost scaling as $O(N D^4 D_c^2)$. Instead of randomly choosing spin pairs to flip, we successively generate configurations by visiting the spin pairs in the order of lattice bonds. In this case, the trial configuration $|S_b\rangle$ obtained by flipping an antiparallel spin pair from the current configuration $|S_a\rangle$ is accepted with a Metropolis probability

$$P_s = \min \left[1, \frac{|\Psi(S_b)|^2}{|\Psi(S_a)|^2} \right]. \quad (9)$$

Similar to the random flipping case, we need pick a random number r from a uniform distribution on the interval $[0, 1)$ and compare it with P_s . If $r < P_s$, $|S_b\rangle$ will be accepted as a new configuration; otherwise $|S_b\rangle$ will be rejected and the original configuration $|S_a\rangle$ will be kept. We can visit all of the bonds sequentially and attempt to flip all encountered antiparallel spin pairs residing on lattice bonds according to the probability Eq. (9). Such a procedure can also be defined as an MC sweep, in which all spins are flipped on average. Then we can measure the observables after each MC sweep to reduce the autocorrelation lengths. Compared with the random visit scheme, the cost of an MC sweep for sequential visiting can be reduced to $O(ND^4D_c^2)$ by storing some auxiliary tensors, which dramatically improves the efficiency. We would like to stress that all kinds of lattice bonds must be swept including both horizontal and vertical bonds, and only sweeping one or the other will not produce correct results. The underlying reason is most likely related to the violation of ergodicity.

As a direct comparison, we show the energy convergence versus MC sweeps of the two different visiting schemes for 8×8 Heisenberg model with $D = 8$, shown in Fig. 1. Here we use $D_c = 24$ to ensure the cutoff convergence for contraction. It shows that in both cases the energies converge very fast, and after about 20000 MC sweeps they converge with errors about 10^{-5} even smaller, and the two schemes give the same converged values -0.619019 after 120000 MC sweeps. This demonstrates the correctness of the sequentially visiting scheme.

On the other hand, we find a not very large D_c can be good enough to produce correct probability distributions when generating configurations, which can further significantly speed up the MC sampling. Our method involves two steps when evaluating observables such as energies: Generating configurations and computing the local energy $E_{\text{loc}}(S)$. The costs of both are dominated by contracting a single-layer tensor network to get $\Psi(S)$ scaling as $O(D^4D_c^2)$, and $D_c = 3D$ always works very well in practical calculations for a spin-1/2 $J_1 - J_2$ model. We observe that the extremely accurate value of the ratio $R = \frac{|\Psi(S_b)|^2}{|\Psi(S_a)|^2}$ in Eq. (9) is not very necessary in the process of generating configurations, since it is merely used to compare with a random number. This exhibits the intrinsic advantage of MC sampling. Generally, we always set $D_{c1} = 2D$ to generate configurations and $D_{c2} = 3D$ to compute $E_{\text{loc}}(S)$ for both unfrustrated and frustrated cases. The convergence of D_{c1} and D_{c2} is shown in the Appendices.

B. Optimization method

Now we turn to the optimization of the PEPS. Optimizing the PEPS wave functions is a very intractable problem. For example, considering a spin-1/2 32×32 square system on the open boundary condition (OBC),

the number of variables in a $D=8$ PEPS wave function is as large as about 7.5×10^6 , which is a great challenge for optimization. Not only the cost is huge, but also it may be trapped into local minima. To overcome this problem, we first use the simple update (SU) imaginary time evolution method to get a rough ground state for initialization [58], and then use stochastic gradient descent method for further optimization to get an accurate ground state [75, 78].

In the stochastic gradient optimization method, each tensor is evolved by a random amount in the opposite direction of the corresponding energy gradient [75, 78],

$$A_{lrud}^{sk}(i+1) = A_{lrud}^{sk}(i) - r \cdot \delta(i) \cdot \text{sgn}\left(\frac{\partial E_{\text{tot}}}{\partial A_{lrud}^{sk}}\right). \quad (10)$$

Here i is the number of evolution step, and r is a random number in the interval $[0, 1)$ for each tensor element A_{lrud}^{sk} . The parameter $\delta(i)$ is the step length, setting the variation range for an element.

Assuming the optimization step number is I and MC sweep number is M , the algorithm structure of the gradient optimization method can be described as follows:

Algorithm 1 VMC-PEPS

```

1: procedure VMC-PEPS( $A$ ) ▷ Input tensors  $A$ 
2:   while  $i \leq I$  do ▷  $I$  is optimization step number
3:     while  $m \leq M$  do ▷  $M$  is MC sweep number
4:       generateConfiguration( $S_m$ ) ▷ Metropolis'
algorithm
5:        $\Psi(S_m) = \text{contractNetwork}(S_m)$  ▷ contracting a
single-layer tensor network with the configuration  $S_m$ 
6:       while  $\langle S'_n | H | S_m \rangle \neq 0$  do ▷ sum over possible
 $S'_n$ 
7:          $\Psi(S'_n) = \text{contractNetwork}(S'_n)$ 
8:          $E_{\text{loc}}(S_m) = E_{\text{loc}}(S_m) + \frac{\Psi(S'_n)}{\Psi(S_m)} \langle S'_n | H | S_m \rangle$  ▷
Compute energy terms
9:       end while
10:       $G_{lrud}^{sk}(S_m) = \text{contractDefectNetwork}(S_m, k)$  ▷
Compute gradient terms
11:       $E_{\text{tot}} = E_{\text{tot}} + E_{\text{loc}}(S_m)$ 
12:       $P_1 = P_1 + G_{lrud}^{sk}(S_m) * E_{\text{loc}}(S_m)$ 
13:       $P_2 = P_2 + G_{lrud}^{sk}(S_m)$ 
14:    end while
15:     $E_{\text{tot}} = E_{\text{tot}} / M$ ;  $P_1 = P_1 / M$ ;  $P_2 = P_2 / M$ 
16:     $P_2 = P_2 * E_{\text{tot}}$ ;  $\frac{\partial E_{\text{tot}}}{\partial A_{lrud}^{sk}} = P_1 - P_2$  ▷ get gradient
for site  $k$ 
17:     $A_{lrud}^{sk}(i+1) = A_{lrud}^{sk}(i) - r * \delta(i) * \text{sgn}\left(\frac{\partial E_{\text{tot}}}{\partial A_{lrud}^{sk}}\right)$  ▷
update tensors
18:  end while
19: end procedure

```

Figure 2 shows the optimization process for a 32×32 square AFM Heisenberg model with $D=8$. We start from the PEPS obtained by the simple update method. The number of MC sweeps is fixed at 45000 for each gradient optimization step. In the first 25 steps, the step length $\delta(i)$ is set as 0.005. We can see the energy

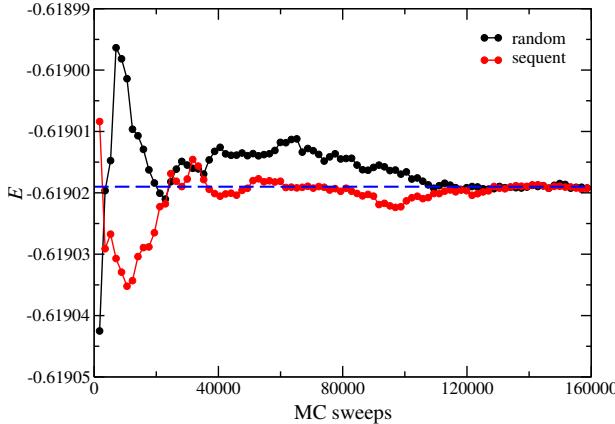


FIG. 1: The comparison of energy convergence versus MC sweeps between randomly visiting (black line) and sequentially visiting (red line) scheme for 8×8 Heisenberg model with $D=8$. The blue dash line denotes the reference energy -0.619019 .

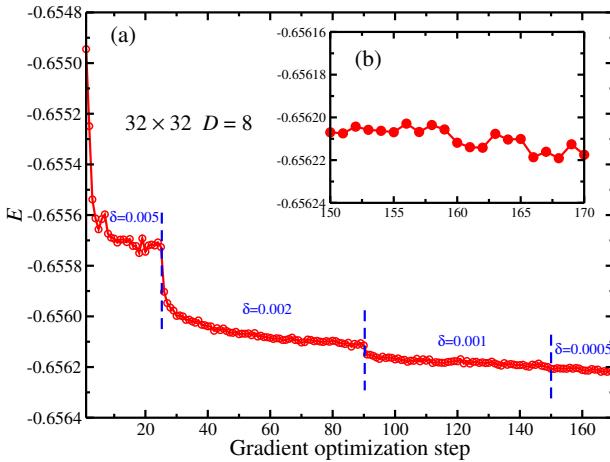


FIG. 2: The energy per site varies with the gradient optimization steps for 32×32 AFM Heisenberg model using PEPS $D = 8$. (a) In the first 25 steps, set the step length $\delta(i)=0.005$; In the next 65 steps, set $\delta(i)=0.002$; From the 91-th step to the 150-th step, set $\delta(i)=0.001$; In the last 20 steps, set $\delta(i)=0.0005$. (b) The energy variation in the last 20 steps.

decreases rapidly at the very beginning, then shows fluctuation. To further decrease the energy, we reduce the step length and keep $\delta(i)=0.002$ until the energy changes slowly. We can continue to reduce $\delta(i)$ to smaller ones such as 0.001 or 0.0005 for further optimization until the step length has no effects on the energy decrease. Fig. 2(b) shows the energy variation for the last 20 steps with $\delta(i)=0.0005$. The optimization process will be stopped when the energy decreases very slowly. It takes about 4 days with 500 Intel E5-2620 cores for the whole optimization process.

Once the optimization process is finished, if one wants to further improve the obtained results, the lattice and

spin symmetries of a given spin-1/2 system can be incorporated into MC sampling [75, 85]. For example, the spin inversion symmetry (SIS) can be used to get a symmetrized wave function $|\Phi\rangle$ when the optimization is finished by $\Phi(S) = \Psi(S) + \Psi(\bar{S})$ where S, \bar{S} denote the configurations $|S\rangle = |s_1 s_2 \cdots s_N\rangle$ and $|\bar{S}\rangle = |-s_1, -s_2, \cdots, -s_N\rangle$, respectively. In general, the results obtained by $|\Phi\rangle$ are slightly more accurate than those of $|\Psi\rangle$. When not otherwise specified, all observables are evaluated with the new weight $|\Phi(S)|^2$ throughout this paper.

C. Heisenberg model on square lattices

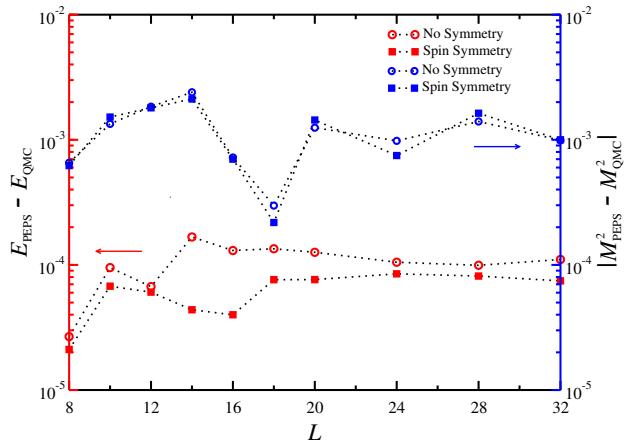


FIG. 3: Errors of ground state energy per site and square magnetization for Heisenberg model for $L = 8 - 32$. The red symbols denote the energy without (open dots) and with (solid squares) spin inversion symmetry. The blue symbols denote corresponding magnetization. QMC and PEPS results are listed in Table. IV.

The Heisenberg model on an $L \times L$ square lattice provides an excellent test bed for benchmark, which is unfrustrated and can be unbiasedly simulated by the QMC method [81, 82]. We compute the ground state energy $E = E_{\text{tot}}/N$ and square magnetization $m^2(\mathbf{k}) = \frac{1}{N^2} \sum_{ij} \langle \mathbf{S}_i \cdot \mathbf{S}_j \rangle e^{i\mathbf{k} \cdot (\mathbf{r}_i - \mathbf{r}_j)}$ with $D=8$ for $L = 8 - 32$. Fig. 3 shows the absolute errors of energy and Néel AFM order $M^2 = m^2(\pi, \pi)$. We can see that the energies obtained with SIS are slightly more accurate than those without SIS, but there is almost no difference for the magnetization. When SIS is used, the largest errors of energy and magnetization up to 32×32 are about 8.0×10^{-5} and 1.0×10^{-3} those of the QMC results, indicating that our results are highly accurate. We note that the errors of energy and magnetization vary randomly with respect to L , possibly because of incomplete optimization of the PEPS, which contains millions of variational parameters. It takes about 5 days with 500 Intel E5-2620 cores to compute the magnetization for the 32×32 lattice with total 50000 MC sweeps.

With OBC, boundary effects may play an important

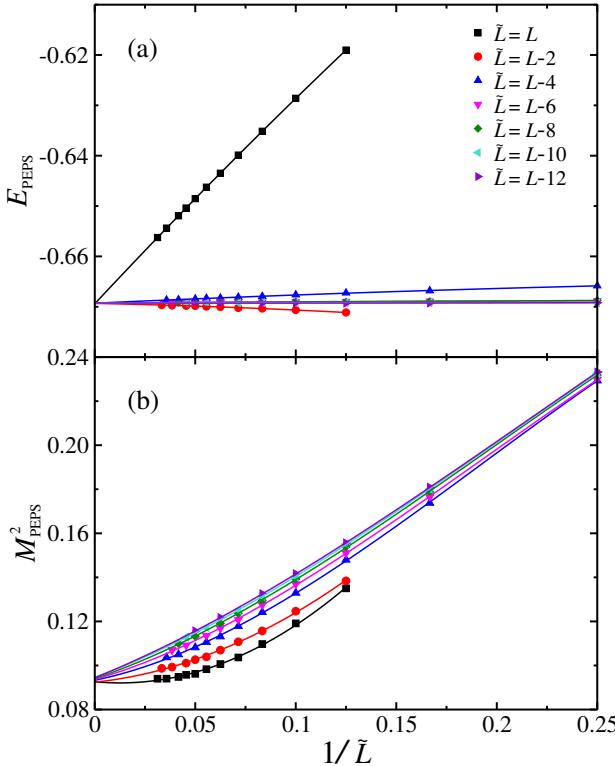


FIG. 4: The finite size scaling of ground state energy and magnetization using different choices of central bulk size $\tilde{L} \times \tilde{L}$ on square Heisenberg model with system size $L = 8 - 32$. (a) The extrapolations of ground state energy with different central bulk choices through a second order polynomial fitting. (b) The extrapolations of magnetization with different central bulk choices. For $\tilde{L} = L$ and $\tilde{L} = L-2$ we use a second order fitting, and for the others a third order fitting is used. Here we use SIS, although one can get similar results without SIS.

role in finite-size scaling (FSS). To reduce boundary effects, one can use the central bulk $\tilde{L} \times \tilde{L}$ of a given $L \times L$ system [78], similar to the standard practice for DMRG calculations when dealing with cylindrical boundary conditions [83]. Various choices are available for the central bulk size \tilde{L} such as $\tilde{L} = L - 2$, $\tilde{L} = L - 4$ and so on. To systematically investigate the influence of \tilde{L} on FSS, we perform the FSS of energy and magnetization for different \tilde{L} choices, as shown in Fig. 4. We first use the whole system size $\tilde{L} = L$ to perform FSS, in which case the extrapolated energy and staggered magnetization are $E_0 = -0.66940(2)$ and $M_0 = 0.304$. Then we choose the central bulk size as $\tilde{L} = L - 2$ and use the corresponding bulk energy and bulk spin order for extrapolation versus the inverse of bulk size $1/\tilde{L}$. Similarly, we can also use other choices such as $\tilde{L} = L - 4$. The extrapolated values with different bulk choices are listed in Table S1 in Supplemental Material. We can see that all of them excellently agree with the standard QMC results using periodic boundary conditions (PBC) with $E_{\text{ex}} = -0.669437$ [81] and $M_{\text{ex}} = 0.3074$ [86]. It is notable that the boundary effects on energy are significantly reduced

when using $\tilde{L} = L - 2$. When smaller central bulks are used, we find that both for the energy and magnetization, the values for different \tilde{L} choices become closer and closer and eventually converge, indicating that boundary effects are gradually eliminated. This is very natural because for a large enough system all bulk choices should give the same value, and specifically for an infinite system there will be no difference among all choices. Additionally, the bulk energy per site for smaller bulk choices, including $\tilde{L} = L - 6$, is approximately -0.6692 and changes very little with respect to system size, providing an efficient way to estimate the energy in the thermodynamic limit.

TABLE I: Ground state energy $E(\infty)$ and square staggered magnetization $M^2(\infty)$ in the thermodynamic limit using different central bulk choices $\tilde{L} \times \tilde{L}$ for extrapolations. The numbers in the brackets are fitting errors. $M(\infty)$ is directly given as the square root of $M^2(\infty)$ without considering fitting errors. The exact results denote QMC results based on periodic boundary conditions [81, 86].

\tilde{L}	$E(\infty)$	$M^2(\infty)$	$M(\infty)$
L	-0.66940(2)	0.0924(11)	0.304
$L-2$	-0.66933(6)	0.0926(10)	0.304
$L-4$	-0.66929(5)	0.0944(06)	0.307
$L-6$	-0.66928(3)	0.0947(08)	0.308
$L-8$	-0.66926(6)	0.0940(13)	0.306
$L-10$	-0.66928(6)	0.0937(13)	0.306
$L-12$	-0.66927(8)	0.0933(12)	0.305
exact	-0.669437 [81]	0.09451 [86]	0.30743(1) [86]

III. APPLICATIONS TO FRUSTRATED MODELS

A. Frustrated model on square lattices

Now we consider the frustrated spin-1/2 J_1 - J_2 square Heisenberg AFM model with size $L_y \times L_x$. In this case, QMC methods suffer the notorious sign problem and we use the DMRG results by keeping 6000 SU(2) states as references [87]. We compute an 8×28 stripe at $J_2/J_1 = 0.5$ and 0.55 with PEPS $D=8$, and the energies are $-0.488958(1)$ and $-0.478997(1)$, which are very close to the DMRG energies of -0.489036 and -0.479085 , respectively; both have an absolute error as small as 8.0×10^{-5} . Fig. 5 compares the spin correlations with those of DMRG at $J_2/J_1=0.5$ and 0.55 . We can see that the PEPS results are in excellent agreement with DMRG. This suggests that our method can also work very well for frustrated models. Here we use 3500000 MC sweeps to obtain long-distance correlation functions with an accuracy of order of 10^{-4} .

To further demonstrate the power of our method, we consider large frustrated systems up to 24×24 at $J_2/J_1=0.5$, on which neither QMC nor DMRG can work. We compare extrapolated ground state energies in the 2D limit with available results obtained by

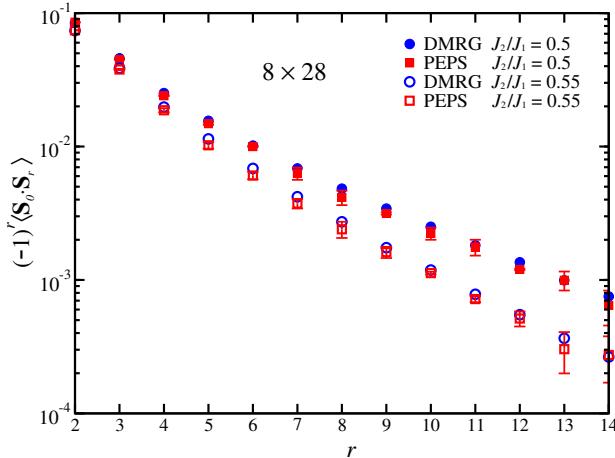


FIG. 5: Comparison of PEPS spin correlations with DMRG for $L_y \times L_x$ systems at $J_2/J_1 = 0.5$ and 0.55 with $L_y=8$ and $L_x=28$. The correlations are measured along the central line $y=4$ and the distance of the reference site away from the left edge is 3 lattice spacings.

DMRG [87] and variational QMC (vQMC) methods [88]. In Fig. 6(a), different central bulk choices $\tilde{L} \times \tilde{L}$ are used for extrapolation, and they all produce almost the same extrapolated energies. Specifically, the choice of $\tilde{L} = L$ gives extrapolated energy $-0.49635(5)$, very close to the extrapolated DMRG energy of -0.4968 and the vQMC plus one Lanczos step energy of -0.4961 , as well as an iPEPS $D = 8$ energy of -0.4964 . The details are shown in Fig. 6(b). We would like to point out that the extrapolated DMRG energy and vQMC energy are both taken from finite systems, so they should be regarded as the corresponding lower bounds of the ground state energies in the 2D limit. In addition, comparing with the energies of different sizes from DMRG and vQMC, bulk energies from OBC show much smaller finite size effects. The above results not only verify that our method can work well on large frustrated systems, but also demonstrate the validity of FSS for both unfrustrated and frustrated systems with OBC. It is notable that previous studies can deal with systems size up to 16×16 and obtain close extrapolated energies [78, 79], but larger sizes calculations up to 24×24 are of great importance to confirm the correctness of those based on smaller systems because they have smaller FSS errors. More results for J_1 - J_2 model are reported elsewhere [89].

To check the convergence with respect to bond dimension D , we calculate the system 24×24 at a highly frustrated point $J_2/J_1 = 0.55$ with D increasing from 4 to 8. For each D , we first use simple update to get an initial state, and then use gradient method for further optimization until the energy varies little. Once optimization is finished, physical observable will be evaluated via MC sampling. Spin orders are computed with single component $m_\alpha^2(\mathbf{k}) = \frac{1}{N^2} \sum_{ij} \langle \mathbf{S}_i^\alpha \cdot \mathbf{S}_j^\alpha \rangle e^{i\mathbf{k} \cdot (\mathbf{r}_i - \mathbf{r}_j)}$ where $\alpha = x, y, z$, as well as the full component one $m^2(\mathbf{k})$. In Table II, we list the ground state energies and

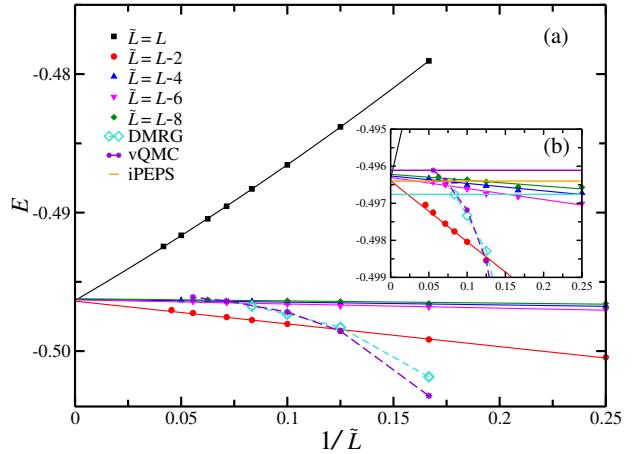


FIG. 6: (a) Ground state energies in the thermodynamic limit at $J_2/J_1=0.5$. Different central bulk choices $\tilde{L} \times \tilde{L}$ are used for extrapolation. The solid lines denote a second order polynomial fitting for $\tilde{L} = L$ or a linear fitting for the other cases, with extrapolated values lying in a small window $[-0.4964, -0.4962]$. Here the error bars from MC sampling and fittings are about 10^{-5} or smaller. The DMRG energies on cylinders with circumference $L_c = 6 - 12$ taken from Ref.[87], and vQMC energies on torus up to 18×18 with one step of Lanczos projection from Ref.[88], are also shown. (b) The results on a more detailed scale. The solid cyan line denotes the DMRG energy of $L_c = 12$, and the solid violet line denotes the vQMC energy of 18×18 sites; both are directly used as extrapolated energies in 2D limit according to Ref.[87] and Ref.[88], respectively. The orange line denotes iPEPS $D = 8$ energy -0.4964 taken from Ref.[90].

spin orders including $M_\alpha^2 = m_\alpha^2(\pi, \pi)$ and $m^2(\mathbf{k})$. With increasing D from 4 to 8, we can see the energy persist E and spin order converge gradually. Notable M_α^2 are getting closer to $\frac{1}{3}M^2$, indicating the SU(2) symmetry is gradually recovered. At the same time, we have $m^2(\pi, 0) = m^2(0, \pi)$ for large D , implying the isotropy of x - and y - axes is also recovered. These results means our method indeed finds the correct ground state subspace.

B. Frustrated models on triangular and kagome lattices

To demonstrate the generality of our method on other frustrated models, we present some results on triangular lattice and kagome lattice. For triangular lattice, we can still define the PEPS on the square lattice. Thus one kinds of the next-nearest neighbor interaction terms along a certain direction in the triangular lattice will be seen as nearest neighbor terms for square-lattice PEPS. Then we can directly apply our method to the triangular models. We compute the fully open spin-1/2 J_1 - J_2 triangular Heisenberg model on a 4×4 square lattice with $D = 8$, shown in Fig. 7. The results for different J_2 are listed in the left part of Tab. III (we set $J_1 = 1$). We can see all the energies are in excellent agreement with

TABLE II: The ground state energies and spin orders for 24×24 at $J_2/J_1=0.55$ with different bond dimension D . Here $M^2 = m^2(\pi, \pi)$ and $M_\alpha^2 = m_\alpha^2(\pi, \pi)$ for $\alpha = x, y, z$, and $\mathbf{k}_x = (\pi, 0)$ and $\mathbf{k}_y = (0, \pi)$.

D	E	M^2	M_x^2	M_y^2	M_z^2	$m^2(\mathbf{k}_x)$	$m^2(\mathbf{k}_y)$
4	-0.479656(3)	$7.46(4) \times 10^{-3}$	$2.12(1) \times 10^{-3}$	$1.96(1) \times 10^{-3}$	$3.38(2) \times 10^{-3}$	$1.07(1) \times 10^{-3}$	$0.96(1) \times 10^{-3}$
5	-0.481498(6)	$6.15(3) \times 10^{-3}$	$2.09(1) \times 10^{-3}$	$1.97(1) \times 10^{-3}$	$2.09(1) \times 10^{-3}$	$0.99(1) \times 10^{-3}$	$1.04(1) \times 10^{-3}$
6	-0.481860(1)	$6.18(8) \times 10^{-3}$	$2.03(4) \times 10^{-3}$	$2.04(2) \times 10^{-3}$	$2.11(2) \times 10^{-3}$	$0.99(3) \times 10^{-3}$	$1.08(2) \times 10^{-3}$
7	-0.481976(3)	$6.24(7) \times 10^{-3}$	$2.06(2) \times 10^{-3}$	$2.06(2) \times 10^{-3}$	$2.12(3) \times 10^{-3}$	$1.03(3) \times 10^{-3}$	$1.07(2) \times 10^{-3}$
8	-0.481989(1)	$6.26(8) \times 10^{-3}$	$2.07(4) \times 10^{-3}$	$2.09(2) \times 10^{-3}$	$2.10(2) \times 10^{-3}$	$1.02(3) \times 10^{-3}$	$1.04(4) \times 10^{-3}$

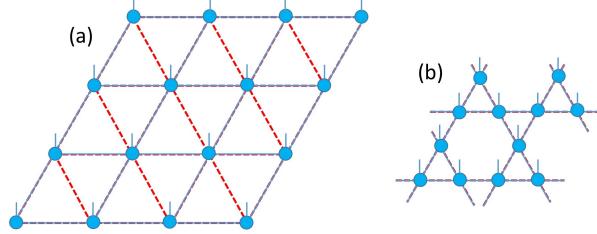


FIG. 7: (a) The square-lattice PEPS representation of triangular-lattice models. (b) The PEPS representation on a kagome torus with 12 sites.

those from exact diagonalization (ED).

For Kagomé lattice, here we consider a small torus with 12 sites, and the PEPS are defined on the lattice. Then we can still use the gradient method to optimize the energy function. Since the system is small, we can use exact summation to replace the MC sampling. We compute the Kagomé antiferromagnetic Heisenberg model with nearest neighbor interactions (set coupling constant $J_1 = 1$). Results of different bond dimension D are listed in the right part of Tab. III. The initialization for different D is always from random tensors. Except the case of $D = 2$, all other cases of $D \geq 3$ are the same with ED results. The above results demonstrate that our method can also work well on triangular and Kagomé lattices. We leave the large-size calculations for further studies.

TABLE III: The left part is comparison between PEPS and ED for the ground state energies of spin-1/2 J_1 - J_2 antiferromagnetic Heisenberg model on triangular lattices. The right part is the ground state energies of a 12-site Kagomé antiferromagnetic Heisenberg model with different bond dimension D .

triangular			kagome	
J_2	PEPS	ED	D	energy
0	-0.481851	-0.4818527	2	-0.4519551985
0.02	-0.479451	-0.4794522	3	-0.4537396014
0.04	-0.477199	-0.4772004	4	-0.4537396014
0.06	-0.475097	-0.4750979	5	-0.4537396014
0.08	-0.473144	-0.4731453	6	-0.4537396014
0.10	-0.471341	-0.4713428	7	-0.4537396014
0.12	-0.469690	-0.4696908	8	-0.4537396014
0.14	-0.468188	-0.4681894	ED	-0.4537396014

IV. CONCLUSION AND DISCUSSION

In summary, we developed an accurate and efficient finite PEPS method based on the scheme of VMC. Our method provides a powerful approach to overcome the core difficulties of tensor network simulation encountered in practical applications, and makes it possible to deal with large quantum systems up to 32×32 sites with high precision. For both unfrustrated and frustrated systems, the obtained results are in excellent agreement with QMC and DMRG. In principle, we can handle larger systems and larger bond dimensions by using more computational resources with massive parallelization, which is the built-in advantage of our method. Our method can also be generalized into fermion systems straightforwardly to deal with Hubbard/t-J model by using Grassmann number TNSs.

We would like to stress that the algorithmic development of finite PEPS ansatz is very necessary. Current progress is mainly focused on infinite systems based on the infinite PEPS (iPEPS) ansatz, with a few tensors in a small unit cell, whereas the finite PEPS ansatz allows a different approach without pre-defining a unit cell, providing the complementary part to the tensor network community. In particular, based on finite 2D systems, it can be directly compared with available DMRG results, which would be very crucial to clarify the nature of some controversial quantum many-body problems [89]. Furthermore, it can also simulate translation-invariance broken systems, including the phases involving short-range or long-range incommensurate orders, as well as trapped cold atoms. It should even allow us to study the real time evolution in the t-VMC scheme [91].

On the other hand, the finite PEPS algorithm can also benefit machine learning field. Current tensor network studies on machine learning is mainly focused on one-dimensional MPS and quasi-one-dimensional tree tensor networks, the related algorithms to which have been well established. Intuitively, PEPS seems to be a more natural approach because in physics it has the same geometric structure as the natural image [92]. Due to its complexity, only until last year did a work using PEPS for supervised learning come up [92]. However, the optimization of PEPS in that work might be still an issue that affects the accuracy [92]. We expect our algorithm would improve the performance of future PEPS application in machine learning models, not limited to

TABLE IV: The ground state energies and square staggered magnetizations for Heisenberg model on $L \times L$ square systems for $L = 8 - 32$ on open boundary conditions. QMC results are obtained by the loop stochastic series expansion (SSE) method with inverse temperature up to $\beta = 120$ [82]. $E_w(\text{PEPS})$ and $M_w^2(\text{PEPS})$ denote PEPS $D = 8$ results with spin inversion symmetry, and $E_o(\text{PEPS})$ and $M_o^2(\text{PEPS})$ denote PEPS $D = 8$ results without spin inversion symmetry.

L	$E(\text{QMC})$	$E_w(\text{PEPS})$	$E_o(\text{PEPS})$	$M^2(\text{QMC})$	$M_w^2(\text{PEPS})$	$M_o^2(\text{PEPS})$
8	-0.619040(1)	-0.619019(1)	-0.619013(1)	0.13565(6)	0.13503(9)	0.13499(0)
10	-0.628667(2)	-0.628600(2)	-0.628572(4)	0.12062(6)	0.11910(0)	0.11928(0)
12	-0.635203(1)	-0.635143(4)	-0.635136(6)	0.1115(2)	0.10974(7)	0.10970(9)
14	-0.639925(3)	-0.639882(3)	-0.639758(4)	0.1058(3)	0.10364(8)	0.10337(9)
16	-0.643528(8)	-0.643489(1)	-0.643398(9)	0.1014(4)	0.10065(9)	0.10062(3)
18	-0.646335(2)	-0.646259(2)	-0.646200(0)	0.0985(2)	0.09830(5)	0.09822(6)
20	-0.648607(9)	-0.648531(5)	-0.648481(0)	0.0976(5)	0.09621(4)	0.09644(2)
22	-0.650463(4)	-0.650387(3)	-0.650336(0)	0.0957(9)	0.09578(8)	0.09573(2)
24	-0.652023(8)	-0.651938(1)	-0.651918(0)	0.0940(8)	0.0948(2)	0.0950(0)
28	-0.654472(8)	-0.654391(4)	-0.654373(5)	0.092(2)	0.0940(1)	0.0938(1)
32	-0.65633(1)	-0.656251(2)	-0.656216(6)	0.095(3)	0.0940(2)	0.0940(0)

supervised learning mentioned above.

V. ACKNOWLEDGMENTS

We thank F. Yang, D. Poilblanc and A. W. Sandvik for helpful discussion. W.-Y. Liu is also indebted to L. He, Y.-J. Han, S.-J. Dong and C. Wang for correlated work. The method is developed with a Fortran library TNSpackage [93]. Z.-C. Gu is supported by Direct Grant No. 4053346 and Group Research Scheme(GRS) No. 3110113 from The Chinese University of Hong Kong; funding from Hong Kong's Research Grants Council (NSFC/RGC Joint Research Scheme No. N-CUHK427/18). S.-S. Gong is supported by NSFC grants No. 11874078, 11834014 and the Fundamental Research Funds for the Central Universities. Numerical calculations were carried out on HPC clusters of CUHK and National Supercomputing Centre in Shenzhen.

Appendix A: Simple update

In the scheme of simple update, the environment of a local tensor is approximated by a series of diagonal matrices, leading to a cheap cost for updating tensors. When dealing with the terms of nearest neighbor (NN) and next-nearest neighbor (NNN) interactions, the cost can be reduced to $O(D^5)$ with the help of QR (LQ) decomposition. Fig. 8 shows how to update tensors for NNN interactions. When an NNN evolution operator acts on the given PEPS, what we need to do is updating several involved tensors, shown in Fig. 8 (a). Fig. 8 (b)-(i) depicts how to update tensors using QR (LQ) decomposition. For detailed explanations please see the caption of Fig. 8.

When using SU method to update tensors, given the bond dimension D , we usually start with a imaginary time step $dt = 0.01$ until the environmental tensors λ_i are converged, i.e., $\frac{1}{P} \sum_{i=1}^P \frac{\|\lambda_i(t+dt) - \lambda_i(t)\|}{\|\lambda_i(t)\|} < 10^{-11}$ where P

is the total number of diagonal matrices. We can decrease the time step to $dt = 0.001$ even smaller for further optimization. In order to speed up the convergence, the large- D states are always initialized by a converged PEPS with small D such as $D=2$ rather than random tensors, and what we need to do is just truncating the increased bond dimension to the desired large D when evolution operators act on the $D=2$ state. Once we get the optimal tensors with SU method, we absorb the environment tensors $\lambda_i^{1/2}$ equally into each local tensor B_i , used as the initial state for gradient optimization.

Appendix B: Boundary-MPS contraction scheme

In our method, we need to contract a $L_y \times L_x$ single-layer tensor network to get the coefficient $\Psi(S)$ for a given configuration $|S\rangle$. Here we adopt the boundary-MPS contraction scheme by contracting the tensor network row by row with treating the first and last row of the tensor network as MPSs and middle rows as MPOs [49]. Shown in Fig. 9(a), the first row is defined as an MPS $|U_1\rangle$ and the k th row as an MPO M_k . In the contraction process, when the MPO M_{k+1} acts on a given MPS $|U_k\rangle$ with a bond dimension D_c , we will get a new MPS $|\tilde{U}_{k+1}\rangle = M_{k+1}|U_k\rangle$ whose bond dimension is DD_c .

In order to avoid the bond dimension increases exponentially during the contraction process, the resulting MPS $|\tilde{U}_{k+1}\rangle$ will be approximated by $|U_{k+1}\rangle$ with a smaller bond dimension D_c by minimizing the cost function $f(T_1, T_2, \dots, T_{L_x}) = \|M_{k+1}|U_k\rangle - |U_{k+1}\rangle\|^2$, where T_i is a rank-3 tensor related to $|U_{k+1}\rangle$ with dimension $D_c \times D \times D_c$ in the middle sites and $D_c \times D \times 1$ on edges, shown in Fig. 9(b). The solution to $f(T_1, T_2, \dots, T_{L_x})$ can be efficiently found by using an iterative sweep algorithm with cost $O(L_x D^4 D_c^2)$ [11]. Therefore we can contract from up to down and we have $|U_k\rangle = M_k M_{k-1} \dots M_2 |U_1\rangle = M_k M_{k-1} \dots M_3 |U_2\rangle = M_k |U_{k-1}\rangle$. Similarly, we can also contract from down to up. Define the last row as $\langle D_{L_y}|$, then we have $\langle D_k| =$

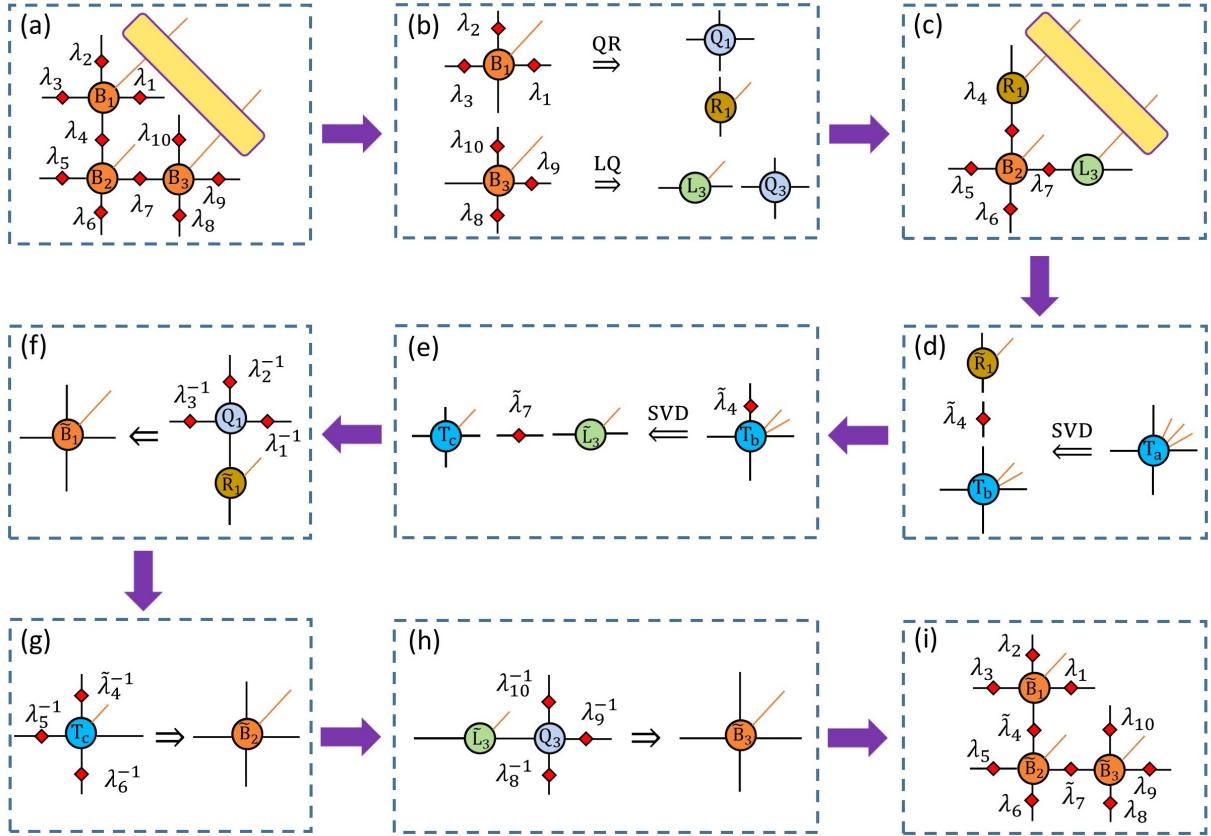


FIG. 8: How to update tensors in the scheme of the SU method with NNN interaction. (a) The NNN evolution operator only acts on several local tensors. In the SU method, PEPS are comprised of tensors B_i living on site i and diagonal matrices λ_i living on the link between nearest-neighbor sites . (b) Contract B_1 (B_3) with corresponding λ s, then do QR (LQ) decomposition on the resulting tensor to get Q_1 (Q_3)and R_1 (L_3). (c) Contract the NNN evolution operator with R_1 , L_3 , B_2 and corresponding λ s, getting a new tensor T_a . (d) Perform SVD on T_a with truncations to get \tilde{R}_1 , $\tilde{\lambda}_4$ and T_b . (e) Contract T_b and $\tilde{\lambda}_4$, then perform SVD with truncations to get \tilde{L}_3 , $\tilde{\lambda}_7$ and T_c . (f) Contract Q_1 with \tilde{R}_1 and the inverse of corresponding λ s to get \tilde{B}_1 . (g) Contract T_c with the inverse of corresponding λ s to get \tilde{B}_2 . (h) Contract Q_3 with \tilde{L}_3 and the inverse of corresponding λ s to get \tilde{B}_3 . (i) A new tensor network state with structure invariance is obtained.

$\langle D_{L_y} | M_{L_y-1} \cdots M_{k+1} M_k = \langle D_{k+1} | M_k$. It is obvious that $\Psi(S) = \langle D_{k+1} | U_k \rangle$ for any $k = 1, 2, \dots, L_y - 1$. We can see the cost of contracting the whole tensor network is $O(N D^4 D_c^2)$ where $N = L_y \times L_x$ is the systems size.

Appendix C: Calculations of physical quantities

Now we turn to how to calculate the energy for a given state. In the MC sampling, we need to compute the local energy $E_{\text{loc}}(S) = \sum_{S'} \frac{\Psi(S')}{\Psi(S)} \langle S' | H | S \rangle$ for a given configuration $|S\rangle = |s_1 s_2 \cdots s_{i-1} s'_i s_{i+1} \cdots s_{j-1} s'_j s_{j+1} \cdots s_N\rangle$. Since the Hamiltonian is comprised of a series of NN and NNN two-body interaction terms, i.e., $H = \sum_{\{ij\}} H_{ij}$ and $H_{ij} = \sum_{s'_i s'_j} \sum_{s_i s_j} (H_{ij})_{s'_i s'_j, s_i s_j} |s'_i s'_j\rangle \langle s_i s_j|$, we have

$$E_{\text{loc}}(S) = \sum_{\{ij\}} \sum_{s'_i s'_j} \frac{\Psi(S')}{\Psi(S)} (H_{ij})_{s'_i s'_j, s_i s_j}, \quad (\text{C1})$$

where $\Psi(S')$ is the coefficient of the configuration $|S'\rangle = |s_1 s_2 \cdots s_{i-1} s'_i s_{i+1} \cdots s_{j-1} s'_j s_{j+1} \cdots s_N\rangle$, and “{}” denotes all NN and NNN spin pairs. The Hamiltonian elements H_{ij} are easily obtained and only nonzero matrix elements contribute to $E_{\text{loc}}(S)$. The main problem for calculating the energy is owing to calculations of $\Psi(S')$ for different spin pairs.

It is notable there are at most 1 spin pair in each configurations $|S'\rangle$ different from the configuration $|S\rangle$. This leads to an efficient evaluation for all different $\Psi(S')$ by storing some auxiliary tensors. Taking the calculation of NN horizontal bond energy terms in the k -th row as an example, which contains $(L_x - 1)$ NN spin pairs. We note $\Psi(S) = \langle D_{k+1} | M_k | U_{k-1} \rangle$ and $\Psi(S') = \langle D_{k+1} | M'_k | U_{k-1} \rangle$, shown in Fig. 10(a) and (b). There are only at most two different tensors between the two MPOs M'_k and M_k , denoted by red balls in Fig. 10 (b).

To get auxiliary tensors, we contract from the first column to the m -th column to get a tensor $L_k^0(m)$, then

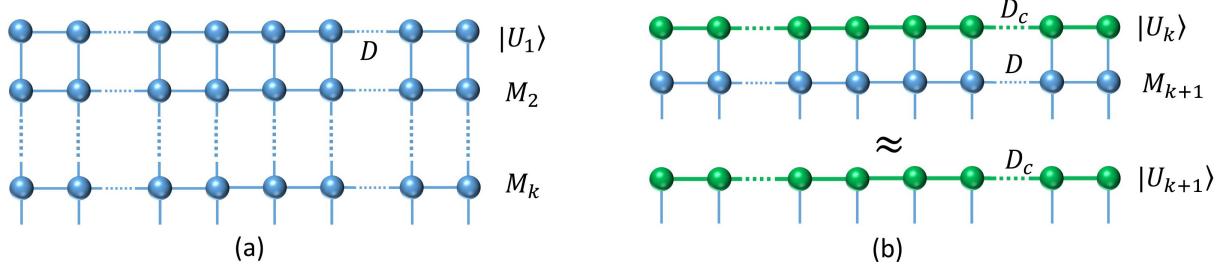


FIG. 9: The boundary-MPS scheme for contracting a tensor network. (a) Treat the first row as an MPS $|U_1\rangle$ and the middle rows as MPOs M_k , and contract the tensor network row by row from up to down. (b) During the contraction process, when the MPO M_{k+1} acts on an MPS $|U_k\rangle$, the resulting MPS will be approximated by $|U_{k+1}\rangle$ with a bond dimension D_c . The approximation is performed with a cost $O(L_x D^4 D_c^2)$, where L_x is the length of the MPS.

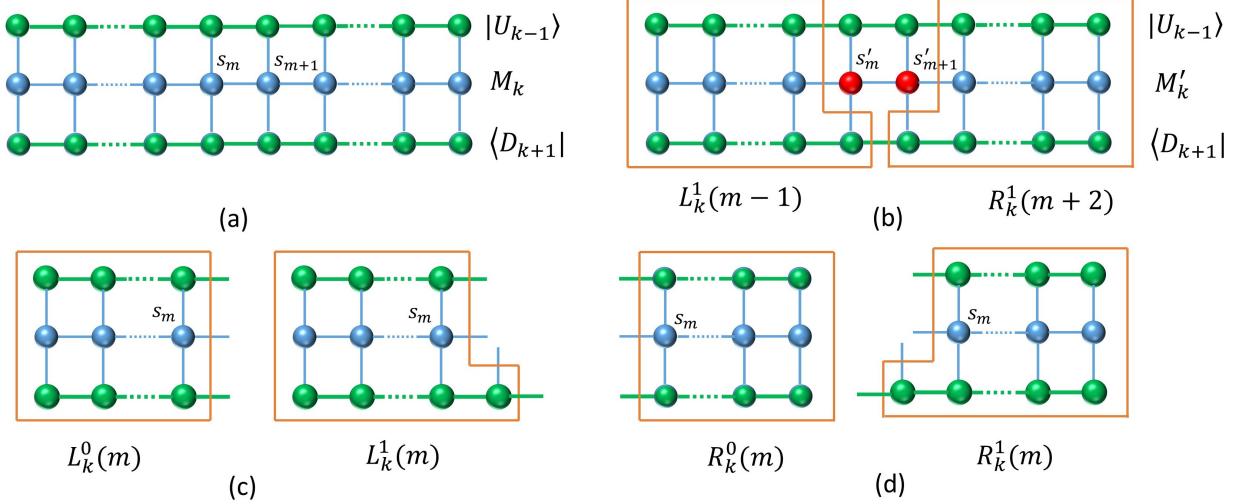


FIG. 10: Use auxiliary tensors to efficiently compute energy terms in the k -th row. (a) Three-line contraction used to get auxiliary tensors. (b) Replace the m -th and $(m+1)$ -th tensors of M_k by the other component tensors correspondingly to get another MPO M'_k . And use auxiliary tensors $L_k^1(m-1)$ and $R_k^1(m+1)$ to get the coefficient $\Psi(S')$ for computing the horizontal bond energy term between the site m and $m+1$. (c) Contract from the first column to the m -th column to get the left auxiliary tensors $L_k^0(m)$, and contract $L_k^0(m)$ with the $(m+1)$ -th tensor in $\langle D_{k+1}|$ to get another auxiliary tensor $L_k^1(m)$. (d) Contract from the last column to the m -th column to get the right auxiliary tensors $R_k^0(m)$, and contract $R_k^0(m)$ with the $(m-1)$ -th tensor in $\langle D_{k+1}|$ to get another auxiliary tensor $R_k^1(m)$.

take $L_k^0(m)$ to contract with the $(m+1)$ -th tensor in $\langle D_{k+1}|$ to get $L_k^1(m)$, shown in Fig. 10(c). Similarly, we contract from the last column to the m -th column and get a tensor $R_k^0(m)$, then use $R_k^0(m)$ to contract with the $(m-1)$ -th tensor in $\langle D_{k+1}|$ to get $R_k^1(m)$, shown in Fig. 10(d). It is easy to verify that $\Psi(S) = \text{Tr}[L_k^0(m)R_k^0(m+1)]$ for $m = 1, 2, \dots, L_x - 1$. With the help of auxiliary tensors including $L_k^0(m)$, $L_k^1(m)$, $R_k^0(m)$ and $R_k^1(m)$, a series of $\Psi(S')$ respectively corresponding to the $(L_x - 1)$ NN spin pairs can be efficiently computed from left to right scaling as $O(L_x D^4 D_c^2)$, shown in Fig. 10(b), then NN horizontal bond energy terms of $E_{\text{loc}}(S)$ in the k -th row for the given configuration $|S\rangle$ are directly obtained. At the same time, using the above auxiliary tensors, we can easily get energy gradients with a leading cost $O(ND^4 D_c^2)$, the same with that of all horizontal bond energy terms. Other energy terms

including vertical bond and NNN terms can be similarly computed.

Appendix D: Convergence of cutoff D_c

Here we consider the effects caused by cutoff D_c . When generating configurations, the Metropolis' probability P_s is used to compare with a random number $r \in [0, 1]$, indicating that it is not necessary to compute the ratio $R = \frac{|\Psi(S_b)|^2}{|\Psi(S_a)|^2}$ in a high precision. That means a relative small D_{c1} which controls the precision of R is good enough to generate configurations. In order to analyse the influence of D_{c1} on the probability distribution, we define the relative error of $\Psi(S_i)$ for a given configuration

$|S_i\rangle$

$$\varepsilon_i = \frac{|\Psi(S_i)_{D_{c1}} - \Psi(S_i)_{\text{ex}}|}{|\Psi(S_i)_{\text{ex}}|}, \quad (\text{D1})$$

where $\Psi(S_i)_{D_{c1}}$ is the value using D_{c1} and $\Psi(S_i)_{\text{ex}}$ denotes the exact value for $|S_i\rangle$. According to Eq.(9), we can estimate the error relative to the exact ratio R_{ex} as

$$\left| \frac{\Delta R}{R_{\text{ex}}} \right| = 2(\varepsilon_a + \varepsilon_b) \approx 4\varepsilon_0, \quad (\text{D2})$$

where $\varepsilon_a(\varepsilon_b)$ is the relative error of $\Psi(S_b)$ ($\Psi(S_b)$), and we assume $\varepsilon_a \approx \varepsilon_b \approx \varepsilon_0$. Then we have

$$P_s = \min[1, (1 \pm 4\varepsilon_0) \cdot R_{\text{ex}}]. \quad (\text{D3})$$

Fig. 11(a) depicts the relative error ε_i of $\Psi(S_i)$ for a 32×32 Heisenberg model, where the configurations $|S_i\rangle$ are generated randomly. We can see the relative errors for different $|S_i\rangle$ are roughly in the same order. For $D_{c1}=8$, the relative error of $\Psi(S_i)$ is about 10%, correspondingly the relative error of R is about 40%. When D_{c1} gets larger such as $D_{c1}=12$, the error $\Psi(S_i)$ is reduced to 1% with the relative error of R being 4%, which makes it possible to produce the correct probability distribution. The relative error of ε_i of $\Psi(S_i)$ for 24×24 at $J_2/J_1=0.5$ shows a similar behaviour, shown in Fig. 12(a). We note the z-component antiferromagnetic spin order $M_z^2 = \frac{1}{N^2} \sum_{ij} (-1)^{i+j} \langle S_i^z S_j^z \rangle$ can be used to detect whether the probability distribution is correctly generated, because M_z^2 only depends on spin configurations. Configurations are generated by different D_{c1} and the corresponding relative error of M_z^2 is computed and depicted as a function of D_{c1} in Fig. 11(b) and Fig. 12(b). We can see the relative error of M_z^2 is almost zero for $D_{c1} \geq 16$, indicating $D_{c1}=16$ is enough to produce a correct probability distribution for 32×32 at $J_2/J_1=0$ and 24×24 at $J_2/J_1=0.5$. Therefore, without loss of generality, we can use $D_{c1} = 2D$ to generate configurations in practical calculations.

We know the precision of physical observables including energy and spin correlations depends on the value of $\frac{\Psi(S')}{\Psi(S)}$ according to Eq.(C1). That means different from generating configurations where a relatively small cutoff D_{c1} can work well, a large cutoff D_{c2} is needed to compute observables if a high precision is required. Taking the energy calculation as an example, Fig. 13(a) shows the energy convergence with respect to D_{c2} on the 32×32 Heisenberg model and 24×24 at $J_2/J_1=0.5$. We can see in both cases when increasing D_{c2} from 8 to 20 the energy converges very fast, while changes very little for $D_{c2} \geq 20$. Compared with the result of $D_{c2} = 32$, the energy errors of $D_{c2} = 24$ is as small as 7.1×10^{-6} for 32×32 and 9.2×10^{-6} for 24×24 lattice, which are the same order of MC sampling errors, show in Fig. 13(b). That means $D_{c2} = 3D$ works quite well to compute physical quantities in the range of allowed MC sampling errors. One can also use larger D_{c1} and D_{c2} , but we do not find improvements in our calculations.

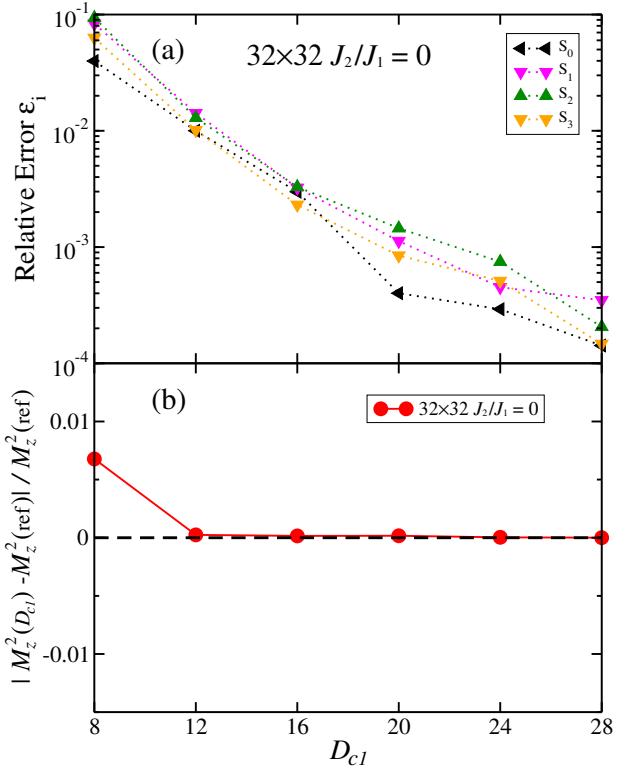


FIG. 11: Convergence of the cutoff D_{c1} used for generating configurations on 32×32 Heisenberg model with $D = 8$. (a) The relatives error ε_i of coefficients $\Psi(S_i)$ versus the cutoff D_{c1} for given spin configurations $|S_i\rangle$. Here the configurations $|S_i\rangle$ are generated randomly and the value $\Psi(S_i)$ of $D_{c1}=200$ is used as the reference. (b) The relative error of z-component spin orders M_z^2 versus the cutoff D_{c1} . $M_z^2(\text{ref})$ denotes the results of $D_{c1} = 32$. M_z^2 is only determined by configurations, therefor measures the correctness of probability distributions of configurations generated with D_{c1} .

Appendix E: MC sampling convergence

We compare the energy convergence with respect to MC sampling on different systems in the scheme of sequentially visiting spin pairs using $D = 8$. Here we use $D_{c1} = 2D$ to generate configurations and $D_{c2} = 3D$ to compute energy $E_{\text{loc}}(S)$ for a given configuration $|S\rangle$. Fig. 14 depicts how the energy per site E changes versus the number of MC sweeps for unfrustrated and frustrated models. To directly compare the energy convergence of different sizes, we subtract their corresponding final energy E_0 which is just the energy with maximal MC sweeps. At $J_2/J_1=0$, 20000 MC sweeps can converge the energy within errors 1.0×10^{-5} , and at $J_2/J_1=0.5$ about 50000 MC sweeps are needed to get the same precision. We note in either unfrustrated or frustrated case, all different systems show the similar convergence behavior, namely, the convergence with respect to MC sweeps is almost size independent. That means our method can be directly applied to larger systems by using

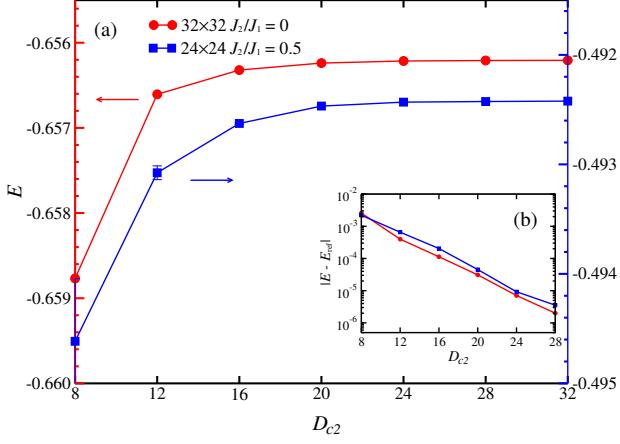
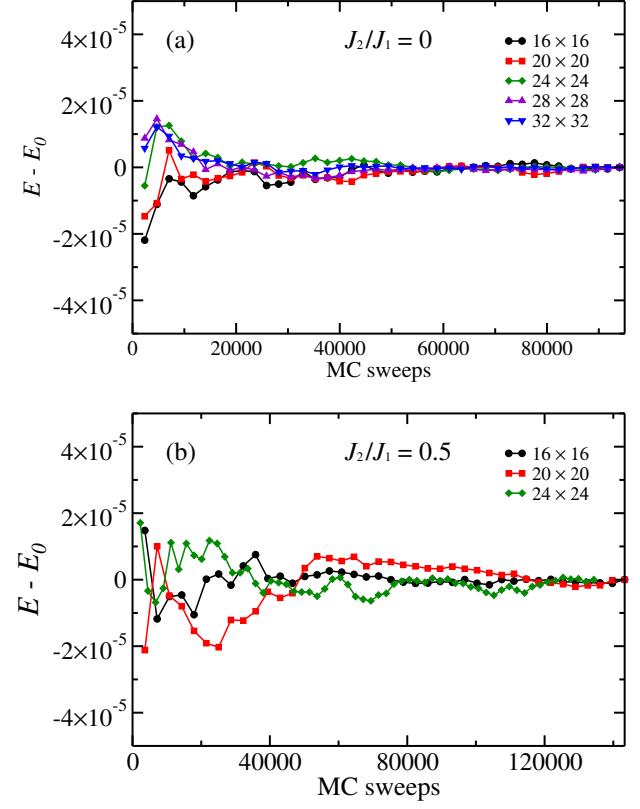
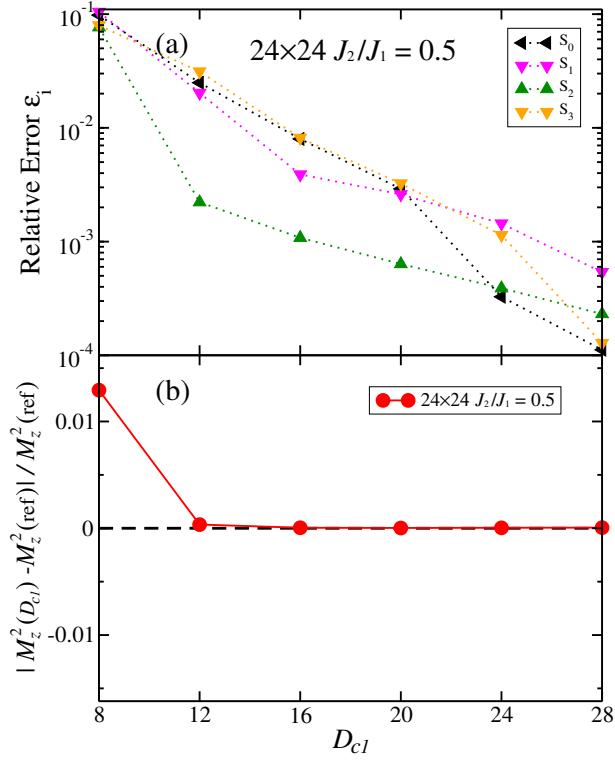


FIG. 13: (a) The convergence of energy versus the cutoff D_{c2} on a 32×32 Heisenberg model and 24×24 $J_2/J_1=0.5$ model. (b) The absolute error of energy versus D_{c2} . E_{ref} denotes the energy using $D_{c2} = 32$.

more computational resources.

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

[2] G. Vidal, Phys. Rev. Lett. **93**, 040502 (2004).

- [3] F. Verstraete, D. Porras, and J. I. Cirac, Phys. Rev. Lett. **93**, 227205 (2004).
- [4] F. Verstraete, M. M. Wolf, D. Perez-Garcia, and J. I. Cirac, Phys. Rev. Lett. **96**, 220601 (2006).
- [5] F. Verstraete and J. I. Cirac, arXiv:0407066 (2004).
- [6] G. Vidal, Phys. Rev. Lett. **99**, 220405 (2007).
- [7] Z. Y. Xie, J. Chen, J. F. Yu, X. Kong, B. Normand, and T. Xiang, Phys. Rev. X **4**, 011025 (2014).
- [8] S. Yang, L. Lehman, D. Poilblanc, K. Van Acoleyen, F. Verstraete, J. Cirac, and N. Schuch, Physical Review Letters **112** (2014).
- [9] C. V. Kraus, N. Schuch, F. Verstraete, and J. I. Cirac, Phys. Rev. A **81**, 052338 (2010).
- [10] R. Orús, Nat. Rev. Phys. **1**, 538 (2019).
- [11] U. Schollwöck, Annals of Physics **326**, 96 (2011).
- [12] R. Orús, Annals of Physics **349**, 117 (2014).
- [13] J. Haegeman and F. Verstraete, Annual Review of Condensed Matter Physics **8**, 355–406 (2017).
- [14] S. R. White and I. Affleck, Phys. Rev. B **54**, 9862 (1996).
- [15] S. R. White and D. J. Scalapino, Phys. Rev. Lett. **91**, 136403 (2003).
- [16] M. Levin and C. P. Nave, Phys. Rev. Lett. **99**, 120601 (2007).
- [17] H. J. Liao, Z. Y. Xie, J. Chen, Z. Y. Liu, H. D. Xie, R. Z. Huang, B. Normand, and T. Xiang, Phys. Rev. Lett. **118**, 137202 (2017).
- [18] Z.-C. Gu, M. Levin, and X.-G. Wen, Phys. Rev. B **78**, 205116 (2008).
- [19] Z.-C. Gu and X.-G. Wen, Phys. Rev. B **80**, 155131 (2009).
- [20] X. Chen, Z.-C. Gu, and X.-G. Wen, Phys. Rev. B **83**, 035107 (2011).
- [21] N. Schuch, I. Cirac, and D. Pérez-García, Annals of Physics **325**, 2153–2192 (2010).
- [22] N. Schuch, D. Pérez-García, and J. I. Cirac, Phys. Rev. B **84**, 165139 (2011).
- [23] F. Verstraete and J. I. Cirac, Phys. Rev. Lett. **104**, 190405 (2010).
- [24] J. Haegeman, T. J. Osborne, H. Verschelde, and F. Verstraete, Phys. Rev. Lett. **110**, 100402 (2013).
- [25] L. Tagliacozzo, A. Celi, and M. Lewenstein, Phys. Rev. X **4**, 041024 (2014).
- [26] Q. Hu, A. Franco-Rubio, and G. Vidal, arxiv:1809.05176 (2018).
- [27] I. L. Markov and Y. Shi, SIAM Journal on Computing **38**, 963 (2008).
- [28] I. Arad and Z. Landau, SIAM Journal on Computing **39**, 3089 (2010).
- [29] I. H. Kim and B. Swingle, arXiv:1711.07500 (2017).
- [30] W. Huggins, P. Patil, B. Mitchell, K. B. Whaley, and E. M. Stoudenmire, Quantum Science and Technology **4**, 024001 (2019).
- [31] F. Pan, P. Zhou, S. Li, and P. Zhang, arxiv:1912.03014 (2019).
- [32] A. J. Ferris and D. Poulin, Phys. Rev. Lett. **113**, 030501 (2014).
- [33] S. Bravyi, M. Suchara, and A. Vargo, Phys. Rev. A **90**, 032326 (2014).
- [34] B. Swingle, Phys. Rev. D **86**, 065007 (2012).
- [35] P. Hayden, S. Nezami, X.-L. Qi, N. Thomas, M. Walter, and Z. Yang, Journal of High Energy Physics **2016**, 9 (2016).
- [36] K. Boguslawski, K. H. Marti, O. Legeza, and M. Reiher, Journal of Chemical Theory and Computation **8**, 1970 (2012).
- [37] N. Nakatani and G. K.-L. Chan, The Journal of Chemical Physics **138**, 134113 (2013).
- [38] S. Szalay, M. Pfeffer, V. Murg, G. Barcza, F. Verstraete, R. Schneider, and O. Legeza, International Journal of Quantum Chemistry **115**, 1342–1391 (2015).
- [39] E. Stoudenmire and D. J. Schwab, in *Advances in Neural Information Processing Systems 29*, edited by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Curran Associates, Inc., 2016), pp. 4799–4807.
- [40] E. M. Stoudenmire, Quantum Science and Technology **3**, 034003 (2018).
- [41] Z.-Y. Han, J. Wang, H. Fan, L. Wang, and P. Zhang, Phys. Rev. X **8**, 031012 (2018).
- [42] Y. Liu, X. Zhang, M. Lewenstein, and S.-J. Ran, arxiv:1803.09111 (2018).
- [43] S. Cheng, L. Wang, T. Xiang, and P. Zhang, Phys. Rev. B **99**, 155131 (2019).
- [44] Z.-F. Gao, S. Cheng, R.-Q. He, Z. Y. Xie, H.-H. Zhao, Z.-Y. Lu, and T. Xiang, arxiv:1904.06194 (2019).
- [45] J. Stokes and J. Terilla, arxiv:1902.06888 (2019).
- [46] A. J. Gallego and R. Orús, arxiv:1708.01525 (2017).
- [47] V. Pestun and Y. Vlassopoulos (2017), arxiv:1710.10248.
- [48] S. R. White, Phys. Rev. B **48**, 10345 (1993).
- [49] F. Verstraete, V. Murg, and J. Cirac, Advances in Physics **57**, 143 (2008).
- [50] J. Jordan, R. Orús, G. Vidal, F. Verstraete, and J. I. Cirac, Phys. Rev. Lett. **101**, 250602 (2008).
- [51] R. Orús and G. Vidal, Phys. Rev. B **80**, 094403 (2009).
- [52] Z. Y. Xie, H. C. Jiang, Q. N. Chen, Z. Y. Weng, and T. Xiang, Phys. Rev. Lett. **103**, 160601 (2009).
- [53] Z. Y. Xie, J. Chen, M. P. Qin, J. W. Zhu, L. P. Yang, and T. Xiang, Phys. Rev. B **86**, 045139 (2012).
- [54] M. Lubasch, J. I. Cirac, and M.-C. Bañuls, Phys. Rev. B **90**, 064425 (2014).
- [55] M. Lubasch, J. I. Cirac, and M.-C. Bañuls, New Journal of Physics **16**, 033014 (2014).
- [56] L. Vanderstraeten, M. Mariën, F. Verstraete, and J. Haegeman, Phys. Rev. B **92**, 201111 (2015).
- [57] S. Yang, Z.-C. Gu, and X.-G. Wen, Phys. Rev. Lett. **118**, 110504 (2017).
- [58] H. C. Jiang, Z. Y. Weng, and T. Xiang, Phys. Rev. Lett. **101**, 090603 (2008).
- [59] P. Corboz, R. Orús, B. Bauer, and G. Vidal, Phys. Rev. B **81**, 165104 (2010).
- [60] L. Wang and F. Verstraete, arxiv:11104362 (2011).
- [61] I. Pižorn, L. Wang, and F. Verstraete, Phys. Rev. A **83**, 052321 (2011).
- [62] P. Corboz, A. M. Läuchli, K. Penc, M. Troyer, and F. Mila, Phys. Rev. Lett. **107**, 215301 (2011).
- [63] H. N. Phien, J. A. Bengua, H. D. Tuan, P. Corboz, and R. Orús, Phys. Rev. B **92**, 035142 (2015).
- [64] R.-Z. Huang, H.-J. Liao, Z.-Y. Liu, H.-D. Xie, Z.-Y. Xie, H.-H. Zhao, J. Chen, and T. Xiang, arXiv:1611.09574 (2016).
- [65] H.-H. Zhao, Z.-Y. Xie, T. Xiang, and M. Imada, Phys. Rev. B **93**, 125115 (2016).
- [66] P. Corboz, Phys. Rev. B **94**, 035133 (2016).
- [67] L. Vanderstraeten, J. Haegeman, P. Corboz, and F. Verstraete, Phys. Rev. B **94**, 155123 (2016).
- [68] Z. Y. Xie, H. J. Liao, R. Z. Huang, H. D. Xie, J. Chen, Z. Y. Liu, and T. Xiang, Phys. Rev. B **96**, 045128 (2017).
- [69] M. T. Fishman, L. Vanderstraeten, V. Zauner-Stauber, J. Haegeman, and F. Verstraete, Phys. Rev. B **98**, 235148

- (2018).
- [70] P. Corboz, P. Czarnik, G. Kapteijns, and L. Tagliacozzo, Phys. Rev. X **8**, 031031 (2018).
 - [71] M. Rader and A. M. Läuchli, Phys. Rev. X **8**, 031030 (2018).
 - [72] L. Vanderstraeten, J. Haegeman, and F. Verstraete, Phys. Rev. B **99**, 165121 (2019).
 - [73] H.-J. Liao, J.-G. Liu, L. Wang, and T. Xiang, Physical Review X **9** (2019).
 - [74] K. Hyatt and E. M. Stoudenmire, arXiv:1908.08833 (2019).
 - [75] A. W. Sandvik and G. Vidal, Phys. Rev. Lett. **99**, 220602 (2007).
 - [76] N. Schuch, M. M. Wolf, F. Verstraete, and J. I. Cirac, Phys. Rev. Lett. **100**, 040501 (2008).
 - [77] L. Wang, I. Pižorn, and F. Verstraete, Phys. Rev. B **83**, 134421 (2011).
 - [78] W.-Y. Liu, S.-J. Dong, Y.-J. Han, G.-C. Guo, and L. He, Physical Review B **95**, 195154 (2017).
 - [79] W.-Y. Liu, S.-J. Dong, C. Wang, Y.-J. Han, H. An, G.-C. Guo, and L. He, Phys. Rev. B **98**, 241109 (2018).
 - [80] S.-J. Dong, C. Wang, Y. Han, G.-C. Guo, and L. He, Phys. Rev. B **99**, 195153 (2019).
 - [81] A. W. Sandvik, Phys. Rev. B **56**, 11678 (1997).
 - [82] O. F. Syljuåsen and A. W. Sandvik, Phys. Rev. E **66**, 046701 (2002).
 - [83] S. R. White and A. L. Chernyshev, Phys. Rev. Lett. **99**, 127004 (2007).
 - [84] A. W. Sandvik, AIP Conference Proceedings **1297**, 135 (2010).
 - [85] A. W. Sandvik, Phys. Rev. Lett. **101**, 140603 (2008).
 - [86] A. W. Sandvik and H. G. Evertz, Phys. Rev. B **82**, 024407 (2010).
 - [87] S.-S. Gong, W. Zhu, D. N. Sheng, O. I. Motrunich, and M. P. A. Fisher, Phys. Rev. Lett. **113**, 027201 (2014).
 - [88] W.-J. Hu, F. Becca, A. Parola, and S. Sorella, Phys. Rev. B **88**, 060402 (2013).
 - [89] W.-Y. Liu, S.-S. Gong, Y.-B. Li, D. Poilblanc, W.-Q. Chen, and Z.-C. Gu, *Gapless quantum spin liquid and global phase diagram of the spin-1/2 j_1 - j_2 square antiferromagnetic heisenberg model* (2020), arXiv:2009.01821.
 - [90] R. Haghshenas and D. N. Sheng, Phys. Rev. B **97**, 174408 (2018).
 - [91] G. Carleo, F. Becca, M. Schiró, and M. Fabrizio, Sci. Rep. **2**, 243 (2012).
 - [92] S. Cheng, L. Wang, and P. Zhang, Phys. Rev. B **103**, 125117 (2021).
 - [93] S.-J. Dong, W.-Y. Liu, C. Wang, Y. Han, G.-C. Guo, and L. He, Computer Physics Communications **228**, 163 (2018).