Improved contraction of finite projected entangled pair states

Markus Scheb *

Department of Physics and Arnold Sommerfeld Center for Theoretical Physics (ASC), Ludwig-Maximilians-University Munich, Theresienstr. 37, D-80333 Munich, Germany

(Received 2 November 2025; revised 1 March 2026; accepted 14 April 2026; published 1 May 2026)

We present an improved version of the algorithm contracting and optimizing finite projected entangled pair states (fPEPS) in conjunction with projected entangled pair operators (PEPOs). Our work has two components to it. First, we explain in detail the characteristic contraction patterns that occur in fPEPS calculations and how to slice them such that peak memory occupation remains minimal while ensuring efficient parallel computation. Second, we combine controlled bond expansion [A. Gleis, J.-W. Li, and J. von Delft, Phys. Rev. Lett. **130**, 246402 (2023)] with randomized singular value decomposition [V. Rokhlin, A. Szlam, and M. Tygert, SIAM J. Matrix Anal. Appl. **31**, 1100 (2009)] and apply it throughout the fPEPS algorithm. We present benchmark results for the Hubbard model for system sizes up to 8×8 and SU(2) symmetric bond dimension of up to $D = 6$ for PEPS bonds and $\chi = 500$ for the environment bonds. Finally, we comment on the state and future of the fPEPS-PEPO framework.

DOI: 10.1103/sfxv-fvyc

I. INTRODUCTION

In the past 30 years, tensor networks have become an increasingly popular tool for calculating quantum many-body systems [1–3]. The prototype of this family of algorithms is the density matrix renormalization group (DMRG) [4,5], which operates on a type of one-dimensional tensor network state called matrix product state (MPS) [6,7]. MPSs are constructed by factorizing and truncating a many-body wavefunction and are most suited for one-dimensional quantum systems. Their natural generalization to two dimensions are projected entangled-pair states (PEPS) [8–11], whose arrangement of tensors matches that of a 2D lattice. While MPSs are convenient to process due to the existence of a canonical gauge, working with a PEPS has proven to be much more complicated due to its loops and the resulting high costs of tensor contractions, the inability to compute expectation values exactly, and the poor convergence of variational calculations.

The most prominent variant of PEPS algorithms is the infinite PEPS (iPEPS) algorithm [12]. It has been successfully applied to various toy models in two dimensions [13–18], but is designed to simulate the thermodynamic limit through a unit cell whose size can not be increased arbitrarily. In addition, PEPSs have been used in more special applications. For instance, imposing unitarity along all virtual bonds yields isometric tensor networks [19–22], which are easier to process but also limited to more exotic phases, such as string-net liquids. Gaussian PEPSs have been used as the starting point for calculating lattice gauge theories [23], d-wave superconductors [24], and U(1)-Dirac spin liquid states [25]. Furthermore, PEPSs have been combined with Monte Carlo methods [26], were shown to represent chiral spin liquids [27], and were applied to thermal states [28].

In this work, we return to the original idea of calculating a grid of unbiased tensors for open boundary conditions without any gauge constraints, called finite PEPS (fPEPS) [29–31]. While the methods listed above are restricted to either small unit cells or limiting gauges and therefore special phases, fPEPSs without gauge constraints have at least the theoretical possibility to describe the entire physical behavior of large, heterogeneous two-dimensional quantum systems in an unbiased fashion. Section II gives a short overview of the fPEPS-PEPO methodology that was detailed in the precursor of this paper [31]. In Sec. III, we describe in detail how to optimally contract the two dominant tensor clusters that occur while optimizing fPEPSs. Afterwards, we combine the controlled bond expansion [32] with the randomized singular value decomposition [33,34] in Sec. IV and apply it to both the environment approximation, as well as the energy minimization within the fPEPS algorithm. Afterwards, we present benchmark results for the Hubbard model in Sec. V and comment on the improvements over the previous version of the algorithm. Finally, in Sec. VI, we comment on the state and future of the fPEPS-PEPO scheme.

*Contact author: mrks.scheb@gmail.com

II. THE FPEPS FRAMEWORK

In the following, we briefly sketch how to conduct energy minimization through finite PEPSs. Figure 1 illustrates the energy functional $E = \langle\psi|H|\psi\rangle$ as a sandwich of a PEPS $|\psi\rangle$, projected entangled pair operator (PEPO) H, and adjoint

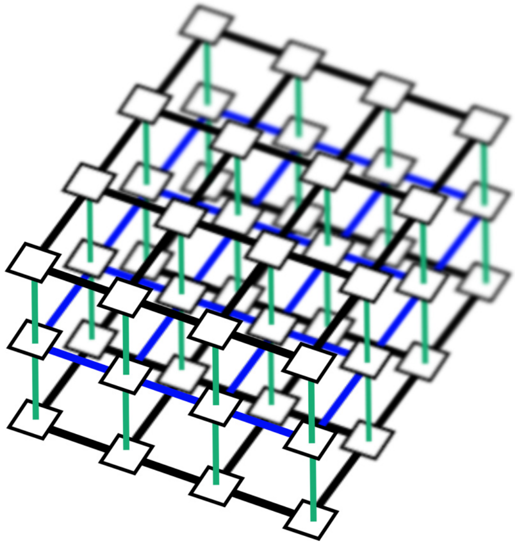

FIG. 1. PEPS-PEPO network for a 4×4 lattice. Black lines connect PEPS tensors, blue lines connect PEPO tensors, and green lines connect PEPS and PEPO tensors.

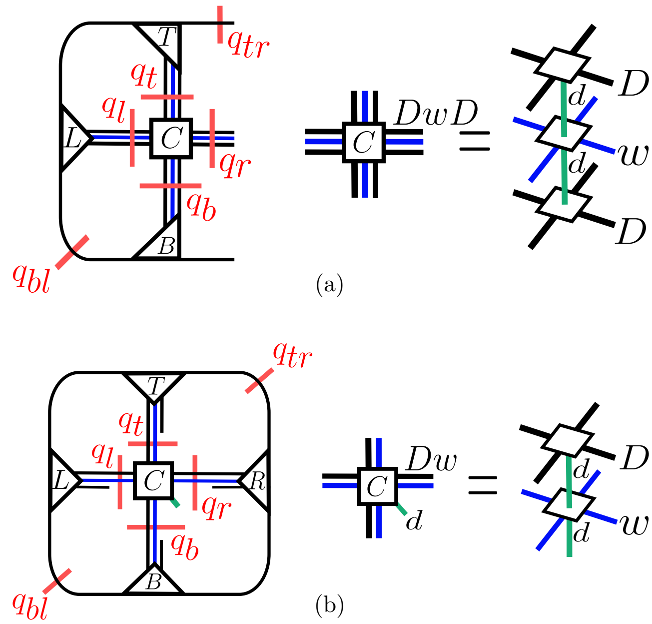

FIG. 2. Characteristic contraction patterns appearing in the fPEPS algorithm, with the environment contraction depicted in (a) and the contraction of an effective single-site Hamiltonian and a PEPS tensor depicted in (b). T, L, B, and R are environment tensors, while the tensor in the center C is a sandwich of a PEPS tensor, PEPO tensor, and adjoint PEPS tensor for (a) and a sandwich of a PEPS tensor and PEPO tensor for (b).

PEPS $\langle\psi|$ for a 4×4 lattice.The PEPS is a representation of the wavefunction for a two-dimensional system, where two adjacent tensors are connected by a black bond of dimension D. The PEPO is the two-dimensional generalization of a matrix product operator (MPO) and stands for a Hamiltonian in 2D. Its construction follows the ideas of Crosswhite *et al.* [35,36], who introduced finite state machines to transform local operators into their tensor network representation. The fermionic sign of the Hubbard model is included via the Jordan-Wigner transformation, which substitutes a cumbersome bookkeeping by parity operators in the Hamiltonian. The resulting PEPO tensors are then connected by blue bonds of dimension w, which remain constant throughout the simulation. The three layers are connected by green bonds representing local Hilbert spaces of dimension d.

Due to the loops inherent to an fPEPS network, the costs of computing expectation values exactly scale exponentially with system size. Therefore, a feasible way of working with fPEPSs includes an environment approximation, in which bundles of three bonds of total dimension $D\,w\,D$ are successively compressed to a cumulative bond of dimension χ. This scheme is structurally identical to the boundary MPO approach [30], where we initialize with $\chi = 1$, start sweeping, successively build up the bond dimension, and find the most optimal symmetry sectors. In practical simulations, $\chi \gg D, w, d$ for stable ground state calculations.

Given this setup, energy minimization is performed by choosing one of two optimization schemes. In the first, called local update, one sweeps over the lattice in both directions to optimize a single tensor plus an adjacent bond. The second, called gradient update, allows one to optimize all PEPS tensors simultaneously while keeping the basis along the bonds fixed.

A thorough explanation of the procedure listed above is given in Ref. [31].

III. OPTIMAL CONTRACTION SEQUENCE

The costs of the fPEPS algorithm are dominated by two characteristic contraction patterns, depicted in Fig. 2. Figure 2(a) consists of three environment tensors (T, L, B) plus a PEPS-PEPO-PEPS sandwich (C), and is computed during the environment approximation, as well as the sweeping process at energy minimization. Figure 2(b) consists of four evironment tensors (T, L, B, R) plus a PEPS-PEPO sandwich (C), and constitutes the $H_{\text{eff}}|\psi\rangle$ operation during the Davidson algorithm, where H_{eff} is the effective Hamiltonian of the single-site Hilbert space and $|\psi\rangle$ is a PEPS tensor. By removing T and B and the bonds attached, one gets the corresponding operations of the DMRG.

Multiplying one tensor after another and as a whole generates giant intermediate contraction results of size $\mathcal{O}(\chi^2(DwD)^2)$, which can exceed the size of all other tensors stored in memory. Therefore, we have developed a strategy for slicing those tensor contractions such that peak memory usage remains as small as possible, without any loss of speed. An overview of this algorithm is given in Algorithm 1.

First, we compute and store the sandwich tensor C. At this point, it is tempting to fuse multiple indices of C to concatenate some of its dense tensor blocks and accelerate future contractions with its neighboring tensors. However, PEPO tensors of local Hamiltonians, which are included in C, do not just observe quantum number conservation, but exhibit even more structure as only a small number of symmetry-preserving hopping terms are actually present. Index fusion therefore leads to large, highly sparse tensor blocks in this

ALGORITHM 1: Sliced tensor contraction.

function maketable(X)
  Initialize table $(q_1, q_2) \to \{X\}_{q_1,q_2}$
  for (q_1, q_2) in X **do**
    Add entry to table $(q_1, q_2) \to \{X\}_{q_1,q_2}$
end function

Calculate C
Initialize table $\{(q_t, q_l)\}_i \to \{(q_b, q_r)\}_i$
for (q_t, q_l, q_b, q_r) *in* C **do**
  Add entry to table $\{(q_t, q_l)\}_i \to \{(q_b, q_r)\}_i$
end

$(q_{tr}, q_t) \to \{T\}_{q_{tr},q_t} \leftarrow$ maketable(T)
$(q_{bl}, q_l) \to \{L\}_{q_{bl},q_l} \leftarrow$ maketable(L)
$(q_{bl}, q_b) \to \{B\}_{q_{bl},q_b} \leftarrow$ maketable(B)
if R *present* **then**
  $(q_{tr}, q_r) \to \{R\}_{q_{tr},q_r} \leftarrow$ maketable(R)
end

$E \leftarrow 0$
for *all* i **do**
  $\{C\}_i \leftarrow \{C\}_{\{(q_t,q_l)\}_i,\{(q_b,q_r)\}_i}$
  for *all* q_{tr} **do**
    for *all* q_{bl} **do**
      if R *present* **then**
        $E+ = ((\{T\}_{q_{tr},q_t} \cdot \{L\}_{q_{bl},q_l}) \cdot \{C\}_i)$
        $\cdot(\{B\}_{q_{bl},q_b} \cdot \{R\}_{q_{tr},q_r})$
      else
        $E+ = ((\{T\}_{q_{tr},q_t} \cdot \{L\}_{q_{bl},q_l}) \cdot \{C\}_i)$
        $\cdot\{B\}_{q_{bl},q_b}$
      end
    end
  end
end

return E

case and we recommend to retain C in the form depicted in Fig. 2(a).

Second, we scan C once and associate bundles of quantum numbers at the top and left (q_t, q_l) with their counterparts at the bottom and right (q_b, q_r). This way, we generate a map $\{(q_t, q_l)\}_i \to \{(q_b, q_r)\}_i$, where the index i designates a set of different quantum numbers whose contraction results may add up to the same final tensor, should q_{tr} and q_{bl} also be equal. The number of these sets determines the number of iterations in the outermost loop of our contraction scheme.

Third, we generate three maps $[(q_{tr}, q_t) \to \{T\}_{q_{tr},q_t}]$, $[(q_{bl}, q_l) \to \{L\}_{q_{bl},q_l}]$, $[(q_{bl}, q_b) \to \{B\}_{q_{bl},q_b}]$, plus a fourth map $[(q_{tr}, q_r) \to \{R\}_{q_{tr},q_r}]$ for the $H_{\text{eff}}|\psi\rangle$ contraction in Fig. 2(b). These additional maps associate the external quantum numbers to the dense blocks inside the environment tensors. After these preparations, we actually calculate the contraction result by nested looping over i, q_{tr}, q_{bl}, and computing $((T \cdot L) \cdot C) \cdot B$ for Fig. 2(a) and $((T \cdot L) \cdot C) \cdot (B \cdot R)$ for Fig. 2(b). Since contractions for different (i, q_{tr}, q_{bl}) do not overlap with each other, this scheme is easily parallelizable.

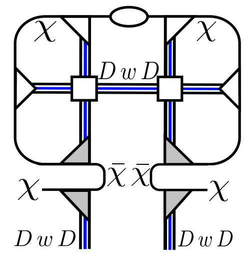

FIG. 3. Controlled bond expansion for the environment approximation.

IV. CONTROLLED BOND EXPANSION VIA RANDOMIZED SINGULAR VALUE DECOMPOSITION

Processing fPEPSs requires two stages at which a bond between two adjacent tensors is optimized. One takes place during the energy minimization of the wavefunction (as is also done in DMRG), the other occurs during the approximation of each environment and is structurally identical to an MPS compression. To circumvent the costs associated with a straightforward 2s-type algorithm, the controlled bond expansion (CBE) [32] allows one to optimize a bond at 1s cost.

Figure 3 illustrates the CBE for the environment approximation.The upper row constitutes the previous environment in a mixed canonical form with bond dimension χ. The middle row is a sequence of PEPS-PEPO-PEPS sandwiches with D as the PEPS dimension and w as the PEPO dimension. The lower row is the new environment to be calculated and is supposed to have maximum overlap with the two rows above at bond dimension χ. The orthogonal projectors at the bottom are defined by the completeness relation in Fig. 4. In the language of projector formalism, χ and $\bar{\chi}$ are the dimensions of kept and discarded space, respectively [37].

Contracting and factorizing the entire cluster in Fig. 3 generates the truncated complement, which contains the most weighty states of the discarded space and is the final output of the CBE. Processing Fig. 3 in this straightforward manner requires operations that are as expensive as performing 2s optimizations, which is why the CBE was introduced in conjunction with the shrewd selection [32], a sequence of contractions and factorizations of smaller tensors. However, as was pointed out by McCulloch *et al.* [38], a more efficient factorization of a large matrix of small rank can be performed using randomized singular value decomposition (RSVD) [33,34]. For the $(\chi DwD) \times (\chi DwD)$ matrix A in Fig. 3, this scheme starts by generating a $(\chi DwD) \times (\tilde{\chi})$ matrix Ω filled with Gaussian random numbers, where $\tilde{\chi} \ll \chi$ is the number of states one wishes to extract from the discarded space. Through repeated application of A and A^T onto Ω, one can extract the dominant subspace within A and perform an

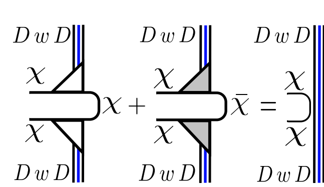

FIG. 4. Completeness relation for environment tensors.

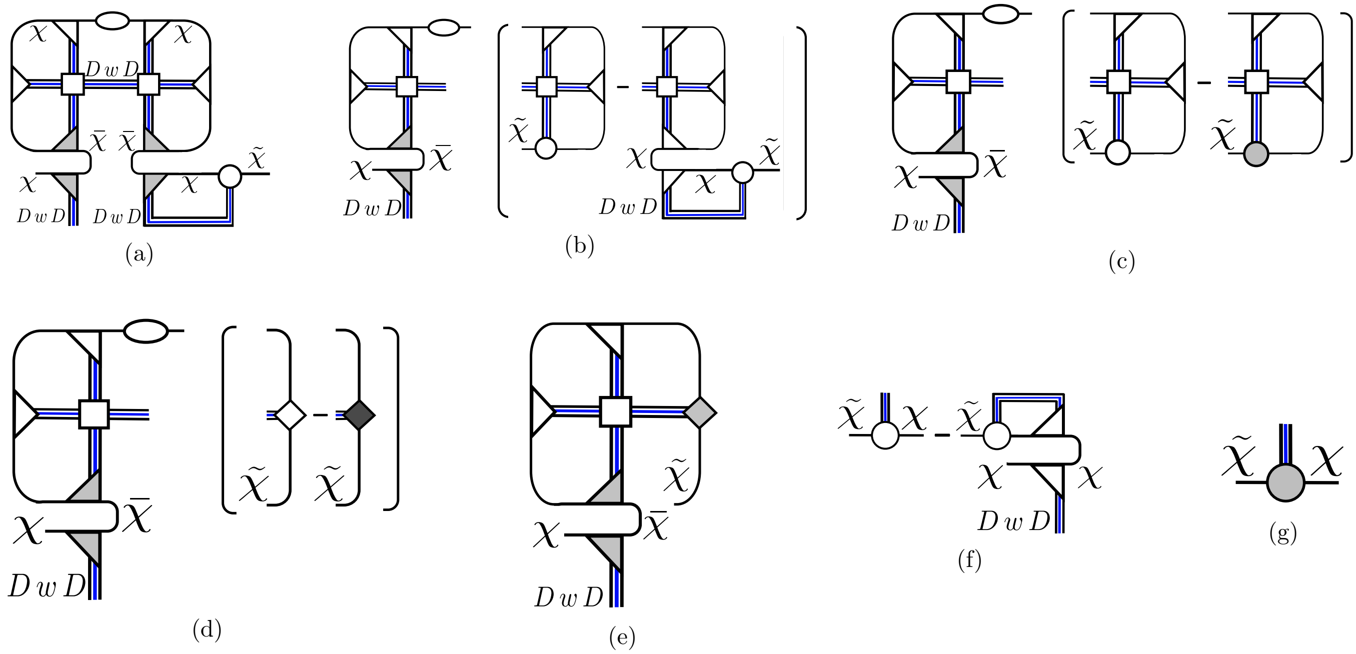

FIG. 5. RSVD for the CBE within environment approximation. (a) Initial setup of $A\,\Omega$. (b) Separation of right orthogonal projector into identity and tangential projector via the completeness relation in Fig. 4. (c) Contraction of tangential projector and Ω. (d) Contraction of both bracketed clusters. (e) Subtraction of both bracketed contraction results. (f) Contraction of upper four tensors from (e) and separation of left orthogonal projector into identity and tangential projector. (g) Final contractions and subtraction.

optimized, truncated factorization. In the context of CBE for fPEPS, $\tilde{\chi} \ll \chi DwD$, which is why we empirically found a single application of A to Ω to be sufficient to extract all relevant contributions from the entire matrix A.

For all steps of the RSVD, we refer to Example 1.6 in Ref. [34]. Here, we only detail the individual operations of $A\,\Omega$ in Fig. 5, which are devised such that the most expensive contraction has $\mathcal{O}(\tilde{\chi}\,\chi^2(DwD)^2)$ cost. Figure 5(a) illustrates the initial setup with the Gaussian matrix Ω as a white circle on the right. First, we note that one should not calculate the orthogonal projector explicitly to avoid generating tensor legs with a dimension of $\bar{\chi}$. Therefore, it is split into the identity and the tangential projector in Fig. 5(b). Afterwards, Ω is contracted with its adjacent environment tensor, leading to two structurally identical clusters in Fig. 5(c). Both are processed according to Fig. 2(a) and subtracted afterwards [Fig. 5(d)]. In Fig. 5(e), the four tensors on top are again contracted according to Fig. 2(a), leaving only a trivial contraction and subtraction as shown in Figs. 5(f) and 5(g). Note that this approach to the CBE renders the operations in Figs. 19 and 20 of Ref. [31] obsolete.

Figure 6 illustrates the CBE for energy minimization. The PEPS tensors are run through the weighted traced gauge [39], such that orthogonal projectors can be constructed [31]. Otherwise the same arguments apply as before and we only need to detail the operations of $A\,\Omega$ in Fig. 7. We again start with the initial setup in Fig. 7(a), where the white circle in the lower right corner constitutes the Gaussian random matrix Ω. Since $\chi \gg D, \tilde{D}, \bar{D}$, the orthogonal projector for PEPS tensors can be calculated directly and is contracted with Ω on the right, leading to Fig. 7(b). The right half of the cluster is calculated according to Fig. 2(a), which leaves the tensors illustrated in Fig. 7(c). The PEPS tensor and its connected projector are dislodged, leaving a cluster that is of the structure $H_{\mathrm{eff}}|\psi\rangle$ and can therefore be contracted according to Fig. 2(b). The remaining, computationally inexpensive operations in Fig. 7(d) yield the final result in Fig. 7(e).

Finally, we note that other less expensive matrix factorizations exist in addition to the RSVD, such as the Golub-Kahan-Lanczos bidiagonalization [40]. However, since the RSVD effectively eliminated the costs of factorizations in the CBE, we felt no need to compare it to other similar procedures.

V. RESULTS

Given the computational improvements laid out in the previous chapters, we present benchmark results for ground state calculations of the two-dimensional Hubbard model. The parameters are similar to those used in the latest ver-

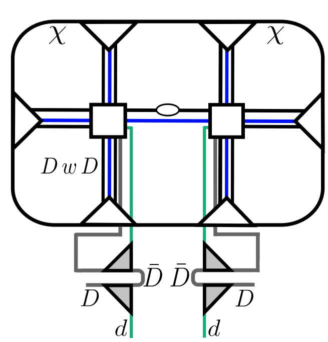

FIG. 6. Controlled bond expansion for energy minimization.

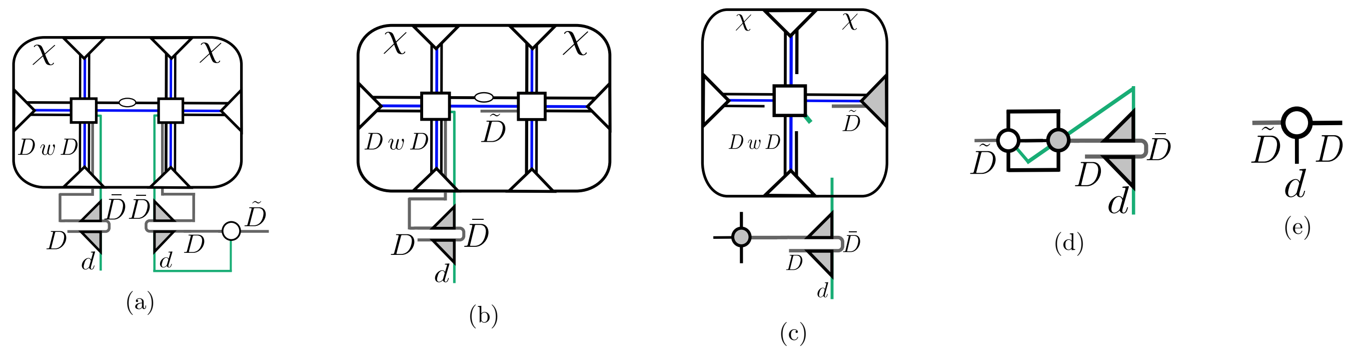

FIG. 7. RSVD for the CBE within energy minimization. (a) Initial setup of $A\,\Omega$. (b) Contraction of right orthogonal projector with Ω. (c) Contraction of four tensors on the right. (d) Contraction of all tensors modulo the lower PEPS layer. (e) Contraction of remaining tensors.

sion of the fPEPS-PEPO algorithm [31]: Hopping is reduced to nearest neighbors only, onsite repulsion is set to $U = 8$, and open boundary conditions are implemented. The simulations for the 4×4 and 6×6 lattices were performed at half-filling, i.e., 16 and 36 electrons, respectively. For the 8×8 lattice, we chose 1/8 filling, i.e., 56 electrons, to induce a stripe structure of the local density. As PEPS calculations without symmetries are intractable in practical simulations, we compared the usage of two different symmetry groups. For $\mathrm{U}(1)_{\mathrm{spin}} \otimes \mathrm{U}(1)_{\mathrm{charge}}$ symmetry, abbreviated as "U(1)," we picked PEPS bond dimensions ranging from $D = 4$ to $D = 8$. For $\mathrm{SU}(2)_{\mathrm{spin}} \otimes \mathrm{U}(1)_{\mathrm{charge}}$ symmetry, abbreviated as "SU(2)," we picked PEPS bond dimensions ranging from $D = 4$ to $D = 6$. In Ref. [31], we empirically found that during ground state calculations the symmetry sectors of virtual bonds did not change significantly after two to three local sweeps, while gradient descent algorithms tend to saturate after circa 100 global sweeps. We therefore opted for alternating between three local sweeps and 100 gradient sweeps, constituting one supersweep. For a proper comparison, we chose the same ratio in this work. All simulations were initialized by PEPSs with bond dimension 1, i.e., product states, such that the first local sweeps would systematically build up the virtual Hilbert spaces and find the best symmetry sectors. While the environment bond dimension ranged from $\chi = 250$ to $\chi = 400$ in Ref. [31], we were able to increase it to $\chi = 500$ for all simulations in this paper, as this was the maximum bond dimension computable. The resulting energies for all lattice sizes were plotted relative to E_0, which is the final energy of a DMRG calculation with $D = 4000$ states. For the 4×4 lattice, E_0 amounts to the exact ground state energy, whereas for the 6×6 and 8×8 lattice, E_0 is an upper bound thereof.

Figure 8 depicts the energy convergence for the 4×4 lattice. The curves show a clear variational behavior, as the energy decreases for higher bond dimensions and flatten for increasing supersweeps. Several notable differences arise when comparing them to Fig. 38 in Ref. [31]: First, the [SU(2),$D = 4$] energies now lie between [U(1),$D = 6$] and [U(1),$D = 7$], whereas previously it seemed to converge to approximately the same value as [U(1),$D = 5$]. Second, for both symmetry groups and most bond dimensions, the energy exhibits a sharp decline at the beginning of the second supersweep, which indicates that after the first batch of gradient sweeps, a set of local sweeps has been overdue to optimize the virtual basis between PEPS tensors. Third, the [SU(2),$D = 6$] run penetrates the 1% barrier after one supersweep, whereas it stayed above it after two supersweeps previously. We attribute all of these differences to the increase of χ from 250, 300, and 350 in Ref. [31] to 500 in this paper. This shows that choosing a χ that is too low can not just lead to numerical instabilities which yield obvious pathological behavior, but distort the convergence in subtle ways that are not detectable in isolation, but only become apparent by comparing different values of χ. While the [SU(2),$D = 6$] run took four days and 150 GB of memory for $\chi = 300$ and two supersweeps, it now takes five days and 26 GB of memory for $\chi = 500$ and 1.5 supersweeps.

We now proceed to the 6×6 lattice in Fig. 9. Again, the energies show a clear variational behavior, although for a lower number of supersweeps due to the larger system size. Simulations for larger bond dimensions had to be terminated

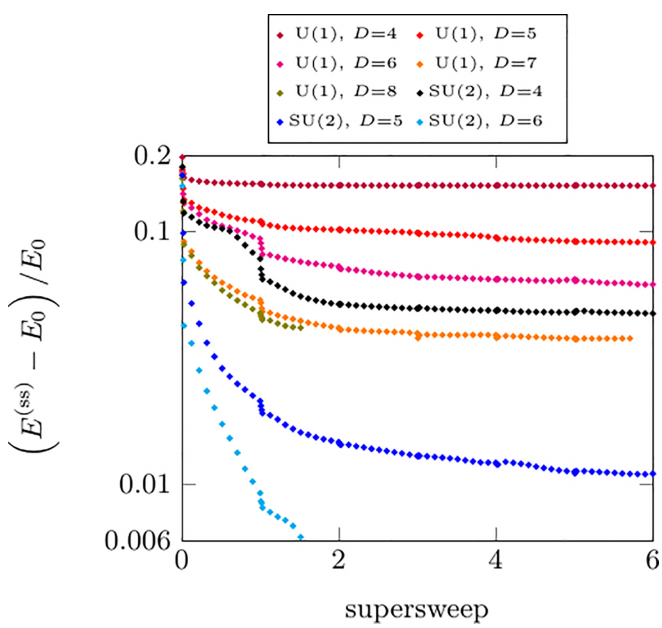

FIG. 8. Relative error in the ground-state energy of the Hubbard model on a 4×4 lattice with open boundary conditions, $U = 8$, $S = 0$, and $N = 16$ (half-filling), calculated with fPEPS with U(1) and SU(2) symmetry.

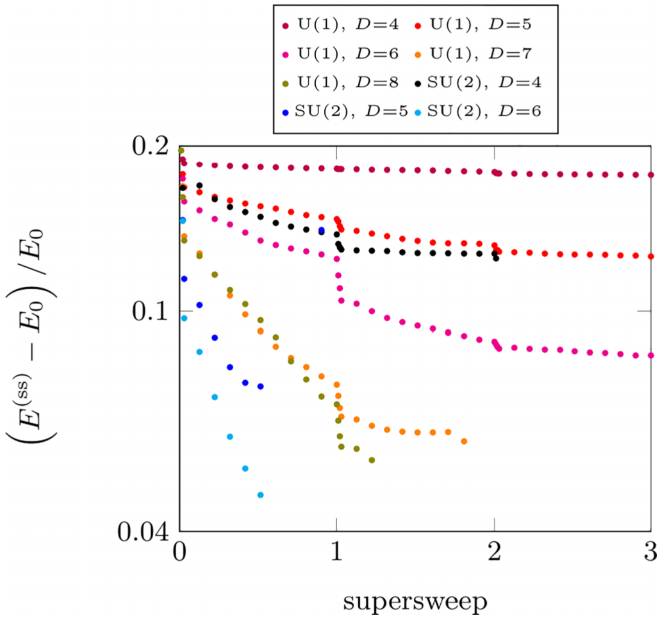

FIG. 9. Relative error in the ground-state energy of the Hubbard model on a 6×6 lattice with open boundary conditions, $U = 8$, $S = 0$, and $N = 36$ (half-filling), calculated with fPEPS with U(1) and SU(2) symmetry.

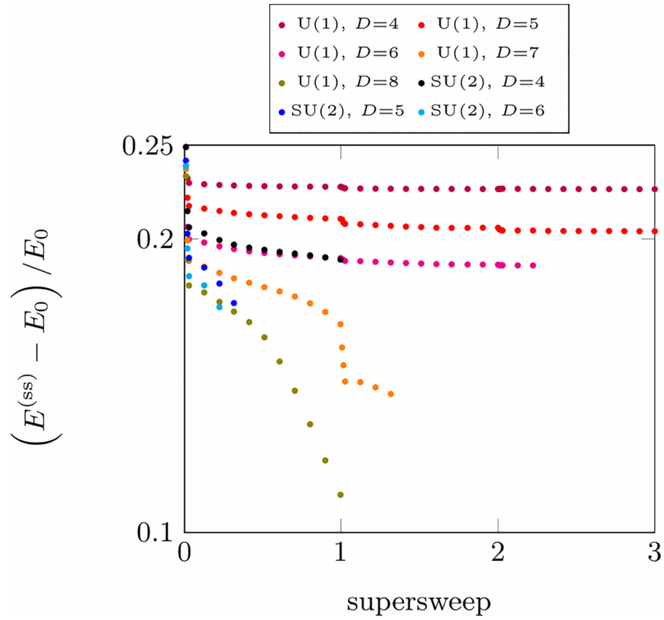

FIG. 10. Relative error in the ground-state energy of the Hubbard model on a 8×8 lattice with open boundary conditions, $U = 8$, $S = 0$, and $N = 56$ (1/8-filling), calculated with fPEPS with U(1) and SU(2) symmetry.

early, as the algorithm became numerically unstable and rerunning those jobs for higher values of χ was not feasible, even with the computational improvements presented above. The dropoff at the beginning of the second supersweep is even more distinct here, indicating that the proper ratio between the number of local sweeps and the number of gradient sweeps ought to be reexamined. The [U(1),$D = 7$] curve exhibits a temporary increase at the end, pointing to a temporary instability from which the algorithm seems to recover afterwards. The most striking improvement compared to Fig. 39 in Ref. [31] is that we were able to execute a [U(1),$D = 8$] simulation, whose energies approximately coincide with those of [U(1),$D = 7$] in the beginning, but then become significantly lower in the second supersweep. While the lowest energy error was previously 6.8% for [SU(2),$D = 6$], we managed to push this number down to 4.7% in this paper, albeit at a higher fidelity due to the larger χ and half the runtime.

Finally, we comment on the energy convergence of the 8×8 lattice at 1/8-filling in Fig. 10. Unfortunately, we were only able to provide a few data points for the SU(2) simulations, as more were either numerically unstable or took more than two weeks of runtime. Unlike the energies depicted in Figs. 8 and 9, the lowest energies are given by the [U(1), $D = 8$] simulation, which after one full supersweep and eight days reached a relative energy error of 11%. This stands in contrast to Ref. [31], where we were only able to reach an error of 16% after half a supersweep and 21 days.

To gain some insight into the physical behavior of the Hubbard model at 1/8-filling, we also present the local density for the U(1) symmetric case at bond dimension 8 in Fig. 11. The z component of the spin is depicted as a blue arrow for positive values and a red arrow for negative values. The hole density is represented by green circles. As expected, we observe the

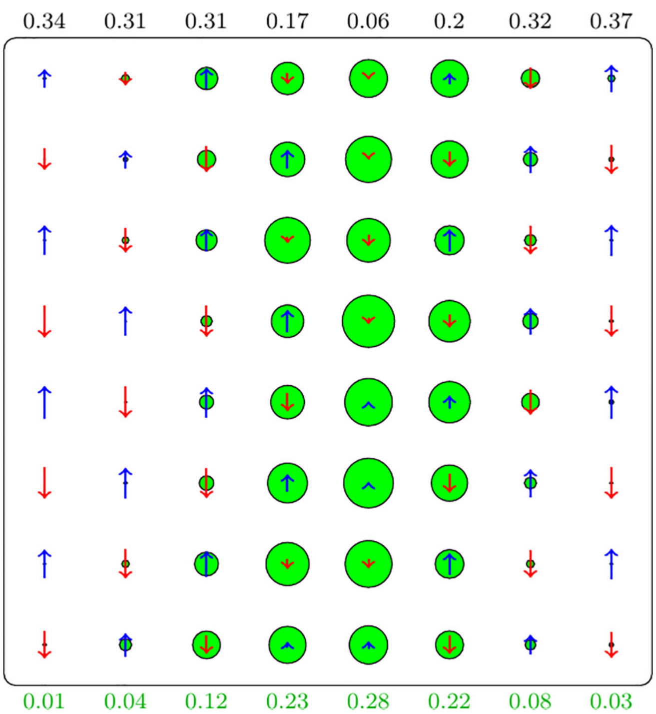

FIG. 11. Local z component of the spin $\langle S_i^z \rangle = \frac{1}{2}(n_{i,\uparrow} - n_{i,\downarrow})$ (size, color, and direction of arrows) and local hole density $1 - \langle n_i \rangle$ (diameter of green-shaded circles) on an 8×8 lattice with open boundary conditions calculated with U(1) symmetry, and bond dimension $D = 8$. Here $U = 8$, $S_z = 0$, and $N = 56$ so that $\langle n \rangle = 0.875$. The black numbers are the average $\langle S_i^z \rangle$ for the column of sites below, and the green numbers on the bottom edge are the average hole densities for the column of sites above.

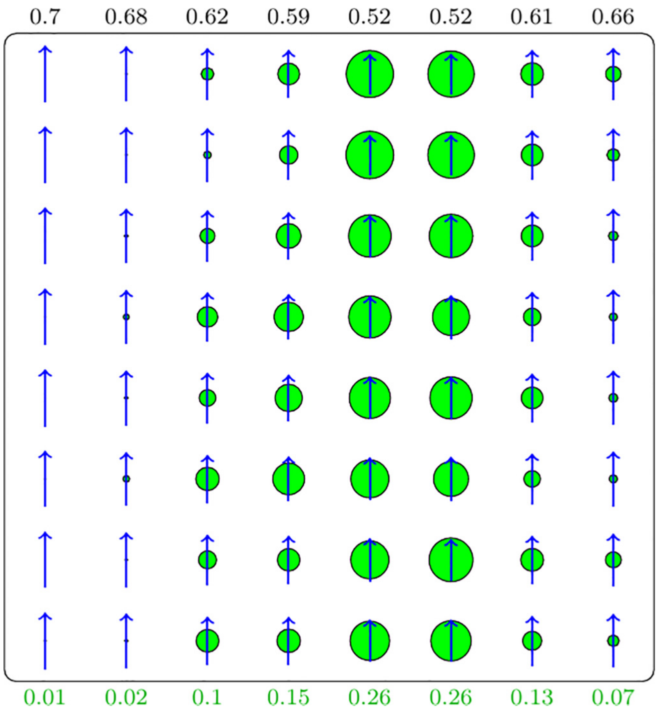

FIG. 12. Local spin density $\langle \mathbf{S}_i^2 \rangle$ (size of blue arrows) and local hole density $1 - \langle n_i \rangle$ (diameter of green-shaded circles) for the Hubbard model on an 8×8 lattice with open boundary conditions and $U = 8$, $S = 0$, and $N = 56$ so that $\langle n \rangle = 0.875$, calculated with SU(2) symmetry, and bond dimension $D = 6$. The black numbers are the average spin density for the column of sites below, and the green numbers on the bottom edge are the average hole densities for the column of sites above.

well-known stripe structure [41] of an oscillating charge density, combined with incommensurate antiferromagnetism. The charge oscillation is edge centered at the top and bottom of the lattice, but appears to be site centered in the middle. Since the algorithm is far from converged, we are unable to determine whether this behavior is closer to the actual physical setup of an 8×8 lattice with open boundary conditions, or a numerical artifact.

We also present the local density for the SU(2) symmetric case in Fig. 12. Since the z component of the spin is zero by construction and antiferromagnetic order is suppressed, we depict the total spin component instead. The charge density exhibits a similar stripe structure as in the U(1) case, whose distribution is a qualitative improvement over the less symmetric distribution in Ref. [31]. We again note that the simulation is far from converged and one should therefore expect the hole density to shift significantly for a more progressed simulation.

VI. SUMMARY AND OUTLOOK

In this paper, we explained in detail how to contract finite PEPSs without any gauge constraints. The first technical section (Sec. III) concerned itself with the optimal contraction of the two dominant contraction patterns and how to slice them such that memory usage remains minimal. The second technical chapter (Sec. IV) illustrated how to combine the CBE with the RSVD and apply this factorization framework to the fPEPS algorithm. Finally, we provided some benchmark results in Sec. V and compared them to the previous version of the fPEPS framework in Ref. [31]. For all three system sizes, we were able to reach lower energies, at a higher χ and lower runtime, therefore justifying the technical improvements presented.

However, as is evident from the data, even these improvements did not yield energies that come close to the upper bounds provided by the DMRG, meaning that in its current form, the fPEPS-PEPO scheme is still not a competitive tool for calculating two-dimensional quantum systems. For future research, the ideas presented in this paper and its precursor [31] have to be combined with other lines of inquiry. One promising option is to incorporate them into the contraction of fPEPS via Monte Carlo methods [26,42,43], which allow for a more efficient tensor contraction at the price of another error. Liu *et al.* [43], in particular, report relative energy errors of less than 1%, which is significantly more accurate than the results presented in this work. We attribute this discrepancy to our attempt at performing exact contractions, as opposed to sampling methods. However, since Ref. [43] also relies on the boundary MPS method, which is structurally identical to our environment approximation, it could possibly benefit from incorporating the CBE+RSVD machinery. Another possibility would be to apply the recently developed belief propagation (BP) [44], which makes an attempt at canonicalizing cyclic tensor networks. Since BP is not guaranteed to converge to an optimal wavefunction for a given bond dimension, neglected contributions can be reincorporated through a loop series expansion [45]. It remains an open question whether these or other potentially new improvements will yield a breakthrough and whether fPEPSs have the practical capacity to describe arbitrary large, heterogeneous, two-dimensional quantum systems.

ACKNOWLEDGMENTS

We thank J. von Delft, A. Gleis, J.-W. Li, and Y. Gao for fruitful discussions. This work was funded in part by the Deutsche Forschungsgemeinschaft under Germany's Excellence Strategy EXC- 2111 (Project No. 390814868). It is part of the Munich Quantum Valley, supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus.

DATA AVAILABILITY

The data that support the findings of this article are not publicly available upon publication because it is not technically feasible and/or the cost of preparing, depositing, and hosting the data would be prohibitive within the terms of this research project. The data are available from the authors upon reasonable request.

[1] R. Orús, Tensor networks for complex quantum systems, Nat. Rev. Phys. **1**, 538 (2019).
[2] J. I. Cirac, D. Pérez-García, N. Schuch, and F. Verstraete, Matrix product states and projected entangled pair states: Concepts, symmetries, theorems, Rev. Mod. Phys. **93**, 045003 (2021).
[3] M. C. Bañuls, Tensor network algorithms: A route map, Annu. Rev. Condens. Matter Phys. **14**, 173 (2023).
[4] S. R. White, Density matrix formulation for quantum renormalization groups, Phys. Rev. Lett. **69**, 2863 (1992).
[5] S. R. White, Density-matrix algorithms for quantum renormalization groups, Phys. Rev. B **48**, 10345 (1993).
[6] S. Rommer and S. Östlund, Class of ansatz wave functions for one-dimensional spin systems and their relation to the density matrix renormalization group, Phys. Rev. B **55**, 2164 (1997).
[7] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, Ann. Phys. **326**, 96 (2011).
[8] F. Verstraete and J. I. Cirac, Renormalization algorithms for quantum-many body systems in two and higher dimensions, arXiv:cond-mat/0407066.
[9] D. Perez-Garcia, F. Verstraete, J. I. Cirac, and M. M. Wolf, PEPS as unique ground states of local Hamiltonians, Quant. Inf. Comp. **8**, 0650 (2008).
[10] N. Schuch, I. Cirac, and D. Pérez-García, PEPS as ground states: Degeneracy and topology, Ann. Phys. **325**, 2153 (2010).
[11] F. Verstraete, M. M. Wolf, D. Perez-Garcia, and J. I. Cirac, Criticality, the area law, and the computational power of projected entangled pair states, Phys. Rev. Lett. **96**, 220601 (2006).
[12] J. Jordan, R. Orús, G. Vidal, F. Verstraete, and J. I. Cirac, Classical simulation of infinite-size quantum lattice systems in two spatial dimensions, Phys. Rev. Lett. **101**, 250602 (2008).
[13] P. Corboz, Variational optimization with infinite projected entangled-pair states, Phys. Rev. B **94**, 035133 (2016).
[14] H. N. Phien, J. A. Bengua, H. D. Tuan, P. Corboz, and R. Orús, Infinite projected entangled pair states algorithm improved: Fast full update and gauge fixing, Phys. Rev. B **92**, 035142 (2015).
[15] P. Corboz, S. R. White, G. Vidal, and M. Troyer, Stripes in the two-dimensional t-J model with infinite projected entangled-pair states, Phys. Rev. B **84**, 041108(R) (2011).
[16] P. Corboz, Improved energy extrapolation with infinite projected entangled-pair states applied to the two-dimensional Hubbard model, Phys. Rev. B **93**, 045116 (2016).
[17] C. Zhang, J.-W. Li, D. Nikolaidou, and J. von Delft, Frustration-induced superconductivity in the t-t' Hubbard model, Phys. Rev. Lett. **134**, 116502 (2025).
[18] J.-W. Li, B. Bruognolo, A. Weichselbaum, and J. von Delft, Study of spin symmetry in the doped $t-J$ model using infinite projected entangled pair states, Phys. Rev. B **103**, 075127 (2021).
[19] M. P. Zaletel and F. Pollmann, Isometric tensor network states in two dimensions, Phys. Rev. Lett. **124**, 037201 (2020).
[20] T. Soejima, K. Siva, N. Bultinck, S. Chatterjee, F. Pollmann, and M. P. Zaletel, Isometric tensor network representation of string-net liquids, Phys. Rev. B **101**, 085117 (2020).
[21] S.-H. Lin, M. P. Zaletel, and F. Pollmann, Efficient simulation of dynamics in two-dimensional quantum spin systems with isometric tensor networks, Phys. Rev. B **106**, 245102 (2022).
[22] W. Kadow, F. Pollmann, and M. Knap, Isometric tensor network representations of two-dimensional thermal states, Phys. Rev. B **107**, 205106 (2023).
[23] P. Emonts and E. Zohar, Fermionic Gaussian projected entangled pair states in $3+1$D: Rotations and relativistic limits, Phys. Rev. D **108**, 014514 (2023).
[24] Q. Yang, X.-Y. Zhang, H.-J. Liao, H.-H. Tu, and L. Wang, Projected d-wave superconducting state: A fermionic projected entangled pair state study, Phys. Rev. B **107**, 125128 (2023).
[25] J.-W. Li, J. von Delft, and H.-H. Tu, U(1)-symmetric Gaussian fermionic projected entangled paired states and their Gutzwiller projection, Phys. Rev. B **107**, 085148 (2023).
[26] T. Vieijra, J. Haegeman, F. Verstraete, and L. Vanderstraeten, Direct sampling of projected entangled-pair states, Phys. Rev. B **104**, 235141 (2021).
[27] J. Hasik, M. Van Damme, D. Poilblanc, and L. Vanderstraeten, Simulating chiral spin liquids with projected entangled-pair states, Phys. Rev. Lett. **129**, 177201 (2022).
[28] A. Sinha, M. M. Rams, and J. Dziarmaga, Efficient representation of minimally entangled typical thermal states in two dimensions via projected entangled pair states, Phys. Rev. B **109**, 045136 (2024).
[29] M. Lubasch, J. I. Cirac, and M.-C. Bañuls, Algorithms for finite projected entangled pair states, Phys. Rev. B **90**, 064425 (2014).
[30] M. Lubasch, J. I. Cirac, and M.-C. Bañuls, Unifying projected entangled pair state contractions, New J. Phys. **16**, 033014 (2014).
[31] M. Scheb and R. M. Noack, Finite projected entangled pair states for the Hubbard model, Phys. Rev. B **107**, 165112 (2023).
[32] A. Gleis, J.-W. Li, and J. von Delft, Controlled bond expansion for density matrix renormalization group ground state search at single-site costs, Phys. Rev. Lett. **130**, 246402 (2023).
[33] V. Rokhlin, A. Szlam, and M. Tygert, A randomized algorithm for principal component analysis, SIAM J. Matrix Anal. Appl. **31**, 1100 (2009).
[34] N. Halko, P. G. Martinsson, and J. A. Tropp, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, SIAM Rev. **53**, 217 (2011).
[35] G. M. Crosswhite, A. C. Doherty, and G. Vidal, Applying matrix product operators to model systems with long-range interactions, Phys. Rev. B **78**, 035116 (2008).
[36] G. M. Crosswhite and D. Bacon, Finite automata for caching in matrix product algorithms, Phys. Rev. A **78**, 012356 (2008).
[37] A. Gleis, J.-W. Li, and J. von Delft, Projector formalism for kept and discarded spaces of matrix product states, Phys. Rev. B **106**, 195138 (2022).
[38] I. P. McCulloch and J. J. Osborne, Comment on "controlled bond expansion for density matrix renormalization group ground state search at single-site costs" (Extended version), arXiv:2403.00562.
[39] G. Evenbly, Gauge fixing, canonical forms, and optimal truncations in tensor networks with closed loops, Phys. Rev. B **98**, 085155 (2018).
[40] G. H. Golub and C. F. Van Loan, *Matrix Computations* (Johns Hopkins University Press, Baltimore, MD, 2013).
[41] B.-X. Zheng, C.-M. Chung, P. Corboz, G. Ehlers, M.-P. Qin, R. M. Noack, H. Shi, S. R. White, S. Zhang, and G. K.-L. Chan, Stripe order in the underdoped region of the two-dimensional Hubbard model, Science **358**, 1155 (2017).
[42] H.-Y. Lin, Y. Guo, R.-Q. He, Z. Y. Xie, and Z.-Y. Lu, Green's function Monte Carlo combined with projected entangled pair

state approach to the frustrated J_1-J_2 Heisenberg model, Phys. Rev. B **109**, 235133 (2024).

[43] W.-Y. Liu, H. Zhai, R. Peng, Z.-C. Gu, and G. K.-L. Chan, Accurate simulation of the Hubbard model with finite fermionic projected entangled pair states, Phys. Rev. Lett. **134**, 256502 (2025).

[44] J. Tindall and M. Fishman, Gauging tensor networks with belief propagation, SciPost Phys. **15**, 222 (2023).

[45] G. Evenbly, N. Pancotti, A. Milsted, J. Gray, and G. K.-L. Chan, Loop series expansions for tensor networks, Phys. Rev. Res. **8**, 013245 (2026).