

Simplified and improved approach to tensor network operators in two dimensionsMatthew J. O'Rourke  and Garnet Kin-Lic Chan*Division of Chemistry and Chemical Engineering, California Institute of Technology, Pasadena, California 91125, USA*

(Received 11 November 2019; revised manuscript received 12 March 2020; accepted 6 April 2020; published 22 May 2020)

Matrix product states and matrix product operators (MPOs) are one-dimensional tensor networks that underlie the modern density matrix renormalization group (DMRG) algorithm. The use of MPOs accounts for the high level of generality and wide range of applicability of DMRG. However, current algorithms for two-dimensional (2D) tensor network states, known as projected entangled-pair states, rarely employ the associated 2D tensor network operators, projected entangled-pair operators (PEPOs), due to their computational cost and conceptual complexity. To lower these two barriers, we describe how to reformulate a PEPO into a set of tensor network operators that resemble MPOs by considering the different sets of local operators that are generated from sequential bipartitions of the 2D system. The expectation value of a PEPO can then be evaluated on the fly using only the action of MPOs and generalized MPOs at each step of the approximate contraction of the 2D tensor network. This technique allows for the simpler construction and more efficient energy evaluation of 2D Hamiltonians that contain finite-range interactions, and provides an improved strategy to encode long-range interactions that is orders of magnitude more accurate and efficient than existing schemes.

DOI: [10.1103/PhysRevB.101.205142](https://doi.org/10.1103/PhysRevB.101.205142)**I. INTRODUCTION**

The density matrix renormalization group (DMRG) algorithm [1,2] is a popular and successful [3] technique for finding the variational ground state of the Schrödinger equation in one spatial dimension (1D). In its modern form, the variational wave function and the Hamiltonian are represented as 1D tensor networks (TNs), namely, matrix product states (MPS) [4–7] and matrix product operators (MPOs) [8–12]. The widespread use of MPOs has allowed for the development of very general, efficient implementations of the algorithm [13], permitting the study of large classes of complex problems in a relatively black-box manner.

However, the two-dimensional (2D) generalization of MPS, known as projected entangled-pair states (PEPS) [14–17], and their associated ground-state algorithms [10,18–22] have not yet come close to the same level of generality or range of applicability. One significant reason for this is that projected entangled-pair operators (PEPOs) [11,23–25], the 2D generalization of MPOs, have been infrequently used in the tensor network literature to date. Despite some applications as an ansatz for 2D mixed states [26–29], PEPOs have not been used as a general representation of 2D Hamiltonians in the context of developing a comprehensive framework for finding ground states. The ground-state optimization algorithms employed by most authors instead utilize a significantly less general representation of the Hamiltonian that is restricted to short-range interactions [10,18–22,30]. We conjecture that this underutilization of PEPOs in favor of simpler operator representations can be attributed to two facts. First, the construction of a PEPO for an arbitrary 2D Hamiltonian is more conceptually complicated than the construction of the MPO

for the analogous Hamiltonian in 1D, which itself is still more complicated than building the local operators currently used in 2D simulations. Second, when compared to the local operators currently used in 2D, the use of PEPOs in a ground-state optimization significantly increases the computational cost of the approximate contraction algorithms for 2D tensor networks in both the finite [31] and infinite (iPEPS) [18,19,32] cases.

In this paper we describe how to overcome both the computational and conceptual complexity of using general tensor network operator representations of the Hamiltonian in 2D algorithms. We focus on developing a simple and efficient framework to evaluate the expectation values of general 2D Hamiltonians, which is compatible with existing gradient-based variational optimization techniques to find their ground states [21,22,33]. To develop this framework, we first briefly summarize the MPO formalism and review some well-known examples that are central ideas in this work (Sec. II). We then introduce a closely related type of tensor network, which we call a generalized MPO (gMPO) (Sec. III A). Next, we show how to reformulate the calculation of the expectation value of a general PEPO into a series of operations involving only MPOs and gMPOs, which we call the boundary gMPO method (Sec. III B). Since the language of MPOs is much better known than that of PEPOs, this reformulation serves to simplify the construction of general 2D Hamiltonians for most readers. In Sec. IV we demonstrate this simplicity by reporting the explicit forms of the gMPOs for various representative types of 2D Hamiltonians. We also show that this scheme sacrifices no accuracy compared to the explicit usage of a PEPO, while providing large speedups in computational time. In addition, a scheme for efficiently constructing and evaluating Hamiltonians with long-range interactions is shown to be many

orders of magnitude more accurate and efficient than existing PEPO-based approaches [25,34,35].

II. MATRIX PRODUCT OPERATORS (MPOs)

Since many detailed and comprehensive presentations of MPOs already exist [7,11,12,23,24], this section will simply contain a brief overview in order to establish notation, as well as some simple examples which we will call upon in later sections.

A. Overview

Consider a 1D system which has been discretized into L localized sites, each with a local Hilbert space \mathcal{H}_i of dimension d_i . A general operator \hat{O} acting on such a system can be written as

$$\hat{O} = \sum_{\{\hat{o}_i\}} O^{o_1 o_2 \dots o_L} \hat{o}_1 \hat{o}_2 \dots \hat{o}_L, \quad (1)$$

where $\{\hat{o}_i\}$ is the set of local operators acting on \mathcal{H}_i and O is a rank- L tensor with indices o_i whose dimensions are equal to the cardinality of their respective set $\{\hat{o}_i\}$. O contains the weights associated with all possible configurations of the local operators \hat{o}_i .

By fixing the indices, a specific element $O^{o_1 o_2 \dots o_L}$ of the tensor O can then be decomposed into a product of matrices $W[i]$:

$$O^{o_1 o_2 \dots o_L} = \sum_{\{\alpha\}} W_{\alpha_1}^{o_1} [1] W_{\alpha_1 \alpha_2}^{o_2} [2] \dots W_{\alpha_{L-1}}^{o_L} [L], \quad (2)$$

where α indexes the so-called ‘‘virtual’’ or ‘‘auxiliary’’ indices which are introduced to perform the matrix multiplication. In Eq. (2) the o_i are simply labels, intended to indicate that each matrix $W[i]$ is chosen specifically so that their product reproduces the element $O^{o_1 o_2 \dots o_L}$. However, if the labels are all reinterpreted as their corresponding indices from Eq. (1), then we see that the full tensor O can be reconstructed as the contraction over rank-3 tensors $W[i]$.

Similar to MPS, this decomposition of a rank- L tensor into L rank-3 tensors is motivated by the fact that most operators of interest do not contain general L -body interactions, but instead are usually limited to few-body terms. This means that, while in general this decomposition could be exponentially expensive, often the O tensor is quite sparse and such a transformation can be a highly efficient way to represent the full tensor.

It is common and frequently useful to associate the operators \hat{o}_i with their corresponding coefficient tensor $W[i]$ according to

$$\hat{W}_{\alpha_{i-1} \alpha_i} [i] = \sum_{o_i} W_{\alpha_{i-1} \alpha_i}^{o_i} [i] \hat{o}_i. \quad (3)$$

This yields matrices $\hat{W}[i]$ in which every element is a $d_i \times d_i$ local operator acting on \mathcal{H}_i . The full operator \hat{O} is thus reconstructed via simple matrix multiplication

$$\hat{O} = \sum_{\{\alpha\}} \hat{W}_{\alpha_1} [1] \hat{W}_{\alpha_1 \alpha_2} [2] \dots \hat{W}_{\alpha_{L-1}} [L], \quad (4)$$

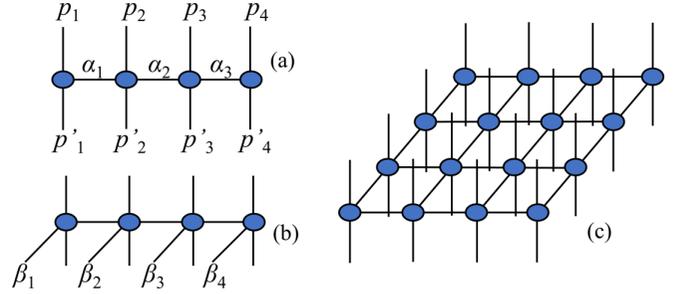


FIG. 1. Tensor network diagrams of (a) an MPO, (b) a gMPO, and (c) a PEPO.

and the set of matrices $\{\hat{W}[i]\}$ are referred to as the MPO representation of \hat{O} . This form of an MPO is commonly used throughout the literature, and will be heavily utilized in the remainder of this work.

We will now relate the MPO form in Eq. (4) to the common diagrammatic representation, as seen in Fig. 1. Since every element of $\hat{W}_{\alpha_{i-1} \alpha_i} [i]$ is itself a $d_i \times d_i$ matrix, each individual numerical element can be exposed by introducing two new indices p_i and p'_i , each of dimension d_i . By fixing each of α_{i-1} , α_i , p_i , and p'_i , the expression $(\hat{W}_{\alpha_{i-1} \alpha_i} [i])_{p_i p'_i}$ yields a single number. More commonly written as $\hat{W}_{\alpha_{i-1} \alpha_i}^{p_i p'_i} [i]$, the correspondence to the rank-4 tensors shown in MPO diagrams becomes apparent. The new indices p_i and p'_i are the so-called ‘‘physical’’ indices, which map the action of the local operators onto the corresponding site tensors of an MPS.

B. Examples

Frequently the operator that one wants to encode as an MPO is a Hamiltonian \hat{H} , so that the DMRG algorithm can be used to find its ground state in the form of an MPS. Here, we will explicitly write out the well-known matrices $\hat{W}[i]$ which make up the MPO representations of several common Hamiltonians consisting of one- and two-body terms. There are multiple techniques that can be used to derive these matrices, each with their own conventions and notation, but in this work we will remain agnostic to these different languages in an attempt to make the presentation in the following sections as conceptually simple and widely accessible as possible. To do so, we will simply refer back to these explicit examples. In lieu of derivations, we will point to helpful references for readers who do not already have a preferred technique for understanding the form of MPO matrices.

1. Nearest-neighbor interactions

Consider a system of L sites, which are indexed by $i \in \{1, 2, \dots, L\}$, and a Hamiltonian consisting of local terms and nearest-neighbor interactions of the form $\hat{H} = \sum_{i=1}^L \hat{C}_i + \sum_{i=1}^{L-1} \hat{A}_i \hat{B}_{i+1}$. In the MPO literature this Hamiltonian is usually written with $\hat{B} = \hat{A}$ so that the interaction is symmetric and \hat{H} is Hermitian, however, in this paper we will always keep the operators distinct for purposes of notational clarity, even though this means that some Hamiltonians under consideration will be non-Hermitian when $\hat{B} \neq \hat{A}$. The MPO

matrices for this Hamiltonian, denoted \hat{W}_{NN} , are given by

$$\begin{aligned}\hat{W}_{\text{NN}}[1] &= (\hat{C} \quad \hat{A} \quad \hat{I}), & \hat{W}_{\text{NN}}[L] &= (\hat{I} \quad \hat{B} \quad \hat{C})^T, \\ \hat{W}_{\text{NN}}[i] &= \begin{pmatrix} \hat{I} & \hat{0} & \hat{0} \\ \hat{B} & \hat{0} & \hat{0} \\ \hat{C} & \hat{A} & \hat{I} \end{pmatrix},\end{aligned}\quad (5)$$

where \hat{I} is the identity operator and $\hat{0}$ is the zero operator.

If instead the interaction is symmetric so that $\hat{H} = \sum_{i=1}^L \hat{C}_i + \sum_{i=1}^{L-1} (\hat{A}_i \hat{B}_{i+1} + \hat{B}_i \hat{A}_{i+1})$, then the MPO matrices ($\hat{W}_{\text{NN-sym}}$) are given by

$$\begin{aligned}\hat{W}_{\text{NN-sym}}[1] &= (\hat{C} \quad \hat{A} \quad \hat{B} \quad \hat{I}), \\ \hat{W}_{\text{NN-sym}}[L] &= (\hat{I} \quad \hat{B} \quad \hat{A} \quad \hat{C})^T, \\ \hat{W}_{\text{NN-sym}}[i] &= \begin{pmatrix} \hat{I} & \hat{0} & \hat{0} & \hat{0} \\ \hat{B} & \hat{0} & \hat{0} & \hat{0} \\ \hat{A} & \hat{0} & \hat{0} & \hat{0} \\ \hat{C} & \hat{A} & \hat{B} & \hat{I} \end{pmatrix}.\end{aligned}\quad (6)$$

In general, for an exact MPO representation of a Hamiltonian \hat{H} , the required bond dimension of the MPO matrices is $D = 2 + br$, where r is the maximum distance over which interactions occur and b is the number of unique operators that act ‘‘first’’ in the interactions. This is reflected in Eq. (5) where $r = 1$ and $b = 1$, and in Eq. (6) where $r = 1$ and $b = 2$. To understand these patterns, as well as the form of the MPO matrices in this section, we recommend Ref. [23].

2. Exponentially decaying interactions

One important exception to the above result is the MPO representation of a Hamiltonian which has long-range interactions that decay exponentially, such as $\hat{H} = \sum_i \hat{C}_i + \sum_{i < j} e^{-\lambda(j-i)} \hat{A}_i \hat{B}_j$. Here, we have introduced a second index j which runs from $i + 1$ to L . Despite the fact that $r = L$ in this case, the Hamiltonian has an exact, compact representation with $D = 3$ MPO matrices (\hat{W}_{exp}) of the form

$$\begin{aligned}\hat{W}_{\text{exp}}[1] &= (\hat{C} \quad \hat{A} \quad \hat{I}), & \hat{W}_{\text{exp}}[L] &= (\hat{I} \quad e^{-\lambda} \hat{B} \quad \hat{C})^T, \\ \hat{W}_{\text{exp}}[i] &= \begin{pmatrix} \hat{I} & \hat{0} & \hat{0} \\ e^{-\lambda} \hat{B} & e^{-\lambda} \hat{I} & \hat{0} \\ \hat{C} & \hat{A} & \hat{I} \end{pmatrix}.\end{aligned}\quad (7)$$

References [11,24,34,36] provide insight into why this is possible for the unique case of exponential interactions.

A special case of this representation, which will prove useful in later sections, is when $\lambda = 0$. The Hamiltonian then has long-range interactions between every pair of sites but the strength of the interactions are all the same, $\hat{H} = \sum_i \hat{C}_i + \sum_{i < j} \hat{A}_i \hat{B}_j$. We will denote this special case with its own MPO notation: \hat{W}_{uniform} .

Much like before, if the interactions are symmetric so that $\hat{H} = \sum_i \hat{C}_i + \sum_{i \neq j} e^{-\lambda|j-i|} \hat{A}_i \hat{B}_j$ (where now both

$i, j \in \{1, \dots, L\}$), the MPO matrices become

$$\begin{aligned}\hat{W}_{\text{exp-sym}}[1] &= (\hat{C} \quad \hat{A} \quad \hat{B} \quad \hat{I}), \\ \hat{W}_{\text{exp-sym}}[L] &= (\hat{I} \quad e^{-\lambda} \hat{B} \quad e^{-\lambda} \hat{A} \quad \hat{C})^T, \\ \hat{W}_{\text{exp-sym}}[i] &= \begin{pmatrix} \hat{I} & \hat{0} & \hat{0} & \hat{0} \\ e^{-\lambda} \hat{B} & e^{-\lambda} \hat{I} & \hat{0} & \hat{0} \\ e^{-\lambda} \hat{A} & \hat{0} & e^{-\lambda} \hat{I} & \hat{0} \\ \hat{C} & \hat{A} & \hat{B} & \hat{I} \end{pmatrix}.\end{aligned}\quad (8)$$

Again, we will give the special case of $\lambda = 0$ its own notation, $\hat{W}_{\text{uniform-sym}}$, which will prove useful in the coming sections.

3. General two-body long-range interactions

As mentioned previously, exact MPO representations of Hamiltonians with general long-range interaction coefficients $\hat{H}_{\text{gen}} = \sum_i \hat{C}_i + \sum_{i < j} V_{ij} \hat{A}_i \hat{B}_j$ require a bond dimension which is proportional to L [24]. However, if V_{ij} is a smoothly decaying function of the distance between two sites, $V_{ij} = f(j - i)$, then highly accurate *approximate* MPO representations of \hat{H}_{gen} can often be found which have finite, constant bond dimensions. The traditional technique is to fit $f(j - i)$ by a sum of exponentials [11,36]

$$f(j - i) = \sum_{k=1}^K a_k e^{-\lambda_k(j-i)}.\quad (9)$$

This yields an MPO representation of \hat{H}_{gen} with bond dimension $K + 2$, where the MPO matrices take the form

$$\begin{aligned}\hat{W}_{K\text{-exp}}[1] &= (\hat{C} \quad a_1 \hat{A} \quad a_2 \hat{A} \quad \dots \quad a_K \hat{A} \quad \hat{I}), \\ \hat{W}_{K\text{-exp}}[L] &= (\hat{I} \quad e^{-\lambda_1} \hat{B} \quad e^{-\lambda_2} \hat{B} \quad \dots \quad e^{-\lambda_K} \hat{B} \quad \hat{C})^T, \\ \hat{W}_{K\text{-exp}}[i] &= \begin{pmatrix} \hat{I} & \hat{0} & \hat{0} & \dots & \hat{0} & \hat{0} \\ e^{-\lambda_1} \hat{B} & e^{-\lambda_1} \hat{I} & \hat{0} & \dots & \hat{0} & \hat{0} \\ e^{-\lambda_2} \hat{B} & \hat{0} & e^{-\lambda_2} \hat{I} & \dots & \hat{0} & \hat{0} \\ \vdots & \vdots & \vdots & \ddots & \hat{0} & \hat{0} \\ e^{-\lambda_K} \hat{B} & \hat{0} & \hat{0} & \dots & e^{-\lambda_K} \hat{I} & \hat{0} \\ \hat{C} & a_1 \hat{A} & a_2 \hat{A} & \dots & a_K \hat{A} & \hat{I} \end{pmatrix}.\end{aligned}\quad (10)$$

The accuracy of the representation $\{\hat{W}_{K\text{-exp}}\}$ is determined by the quality of the fit in Eq. (9).

Although this is often a reasonably accurate approach, several more sophisticated techniques have been developed in recent years which are based on the singular value decomposition (SVD) of blocks of V_{ij} [12,37]. These methods also work most effectively when V_{ij} is a smooth function of the distance, but they are able to fit more general functions f that may be challenging to represent directly with exponentials like those in Eq. (9) [12]. They also can be a bit more efficient, producing a higher accuracy representation of \hat{H}_{gen} with a smaller bond dimension than Eq. (10) [37].

In this work, we utilize the technique described in Ref. [37]. The basic idea is that the MPO matrices \hat{W}_{gen} for the general Hamiltonian \hat{H}_{gen} can be written as

$$\hat{W}_{\text{gen}}[i] = \begin{pmatrix} \hat{I} & \hat{0} & \hat{0} \\ (v_i)_a \hat{B} & (X_i)_{aa} \hat{I} & \hat{0} \\ \hat{C} & (w_i)_a \hat{A} & \hat{I} \end{pmatrix},\quad (11)$$

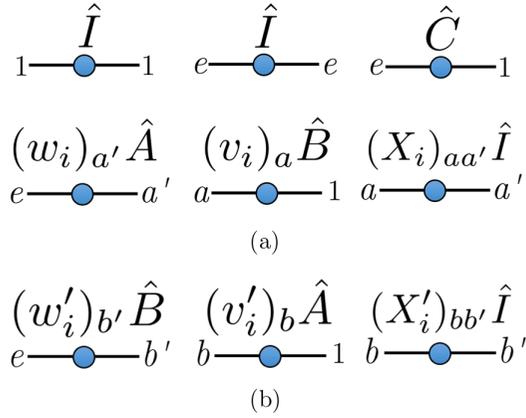


FIG. 2. A set of tensor network diagrams that represent the elements of the operator-valued MPO matrix $\hat{W}_{\text{gen}}[i]$ (a), along with the additional elements needed for $\hat{W}_{\text{gen-sym}}[i]$ (b). Here, we assume $\hat{W}_{\text{gen}}[i]$ is a $(2 + l_i) \times (2 + r_i)$ matrix and $\hat{W}_{\text{gen-sym}}[i]$ is a $(2 + 2l_i) \times (2 + 2r_i)$ matrix. We use the symbol “1” to denote the first value of a given index, e to denote the final value of a given index, a to denote the set of values ranging from 2 to $l_i + 1$, b to denote the set of values ranging from $l_i + 2$ to $2l_i + 1$, a' to denote the set of values ranging from 2 to $r_i + 1$, and b' to denote the set of values ranging from $r_i + 2$ to $2r_i + 1$. This index labeling corresponds directly to the expressions in Eqs. (11) and (12).

where \vec{v}_i is a column vector of coefficients that has length l_i and is indexed by a , X_i is an $l_i \times r_i$ matrix of coefficients indexed by a and a' , and \vec{w}_i is a row vector of coefficients that has length r_i and is indexed by a' , yielding a $(2 + l_i) \times (2 + r_i)$ MPO matrix. We write the indexed elements of \vec{v}_i , \vec{w}_i , and X_i in Eq. (11) to remind the reader of the shape of these quantities. For clarity, tensor network diagrams for this matrix are given in Fig. 2(a). If the coefficients contained in $\hat{W}_{\text{gen}}[i]$ can be, to a good approximation, related to the coefficients contained in $\hat{W}_{\text{gen}}[i + 1]$ by a linear transformation, then the MPO matrices for each site can be successively generated by finding the correct linear transformation on the coefficients contained in the MPO matrix on the previous site. These linear transformations can be found by taking SVDs of certain blocks of the upper triangle of V_{ij} . It is observed in [37] that if V_{ij} is a smooth function of the distance $|j - i|$, the transformations are often compact (i.e., their dimensions do not scale with L) and highly accurate because subblocks of the upper triangle of V_{ij} are low rank. These ideas are developed in detail in the Supplemental Material of Ref. [37],¹ and are expanded upon in terms of hierarchical matrix methods in Ref. [35].

The form of this MPO matrix can be viewed as a direct generalization of \hat{W}_{uniform} . The “coefficients” in adjacent \hat{W}_{uniform} matrices can be related to each other via the simplest possible linear transformation ($X_{1 \times 1} = 1$, $\vec{v} = \vec{w} = 1$) because all the interactions are of identical strength and thus all subblocks of

V_{ij} are rank 1. However, when the interaction coefficients vary with distance and the subblocks of the upper triangle of V_{ij} are rank- l , the single \hat{I} in the center of \hat{W}_{uniform} gets generalized to an $l \times l$ block $X_{l \times l} \hat{I}$ in \hat{W}_{gen} . By extension, \vec{v} and \vec{w} undergo the same generalization. The MPOs \hat{W}_{exp} and $\hat{W}_{K\text{-exp}}$ are special, simple cases of this generalization.

As a final note, if the interactions in the general Hamiltonian \hat{H}_{gen} become symmetric so that $\hat{H} = \sum_i \hat{C}_i + \sum_{i \neq j} V_{ij} \hat{A}_i \hat{B}_j = \sum_i \hat{C}_i + \sum_{i < j} V_{ij} \hat{A}_i \hat{B}_j + \sum_{j < i} V_{ij} \hat{B}_j \hat{A}_i$, then the general MPO matrices become

$$\hat{W}_{\text{gen-sym}}[i] = \begin{pmatrix} \hat{I} & \hat{0} & \hat{0} & \hat{0} \\ (v_i)_a \hat{B} & (X_i)_{aa'} \hat{I} & \hat{0} & \hat{0} \\ (v'_i)_{b'} \hat{A} & \hat{0} & (X'_i)_{bb'} \hat{I} & \hat{0} \\ \hat{C} & (w_i)_{a'} \hat{A} & (w'_i)_{b'} \hat{B} & \hat{I} \end{pmatrix}. \quad (12)$$

Tensor network diagrams representing this matrix are given in Fig. 2. Here, we have introduced the additional indices b and b' to index the new vectors \vec{v}'_i , \vec{w}'_i and the new matrix X'_i , as described in Fig. 2. If we have the additional property that interaction coefficients themselves are symmetric $V_{ij} = V_{ji}$, then the above expression can be simplified according to $\vec{v}'_i = \vec{v}_i$, $\vec{w}'_i = \vec{w}_i$, $X'_i = X_i$.

III. PEPO EXPECTATION VALUE VIA GENERALIZED MPOs

A. Generalized MPOs (gMPOs)

In order to relate the contraction of PEPOs to the well-known 1D MPOs described in Sec. II B, we must first introduce the notion of a generalized MPO (gMPO). In a gMPO, the operator-valued MPO matrices $\hat{W}[i]$ are elevated to rank-3 tensors, which will be indicated by the addition of a virtual index $\beta_i \in \{1, 2, \dots, g\}$. The new operator-valued, rank-3 gMPO tensors will be denoted by $\hat{M}_{\beta_i}[i]$. Exposing all the indices explicitly, this gives a rank-5 tensor $M_{\alpha_{i-1} \alpha_i \beta_i}^{p_i p'_i}[i]$, which is shown in diagrammatic form in Fig. 1.

The basic notion of a gMPO is that for each value of β_i , a different MPO matrix $\hat{W}[i]$ can be encoded in the gMPO tensor. In the simplest case β_i only takes a single value ($g = 1$) and thus every gMPO tensor can only represent a single MPO matrix, reducing the gMPO back to a regular MPO. If instead β_i takes two values ($g = 2$), then every tensor can represent two different MPO matrices, and the gMPO can encode 2^L different 1D MPOs. In practice, however, the β_i are not “free” indices but are instead summed over in the final network just like the α indices in a regular MPO [see Eq. (4)]. The proper notion of a gMPO is therefore as a tensor network operator that can represent a sum of many regular 1D MPOs after the β_i are appropriately summed over. This formulation is useful because it provides a flexible framework in which operators in regular MPOs can be coupled with other operators that act “outside” of the 1D domain of the regular MPO. In general, it allows for the complete coupling of two distinct MPOs into one, however, in this work we only utilize a simpler special case in which specific local operators are coupled together. Much like how a local operator on site i can be coupled to a local operator on site $i + 1$ by summing over the index α_i in a regular MPO, we use the gMPO formalism to couple a local

¹It should be noted that in the referenced paper there is a typo in the explicit expressions for the compressed MPO matrices. The local operators associated with the X block of each matrix should be \hat{I} , not \hat{n} , as in Eq. (11).

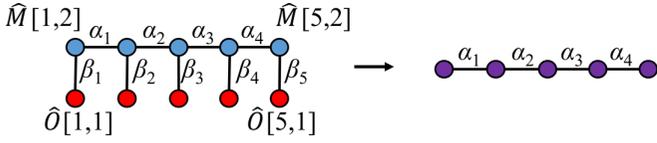


FIG. 3. A gMPO-based representation of the two-row example Hamiltonian in Sec. III A for a 2×5 system. Left: The gMPO tensors (blue) appear on the sites in row 2, while the complementary operator vectors (red) appear on the sites in row 1. Physical indices are suppressed for simplicity. Right: The resulting tensor network along row 2 (an MPO) after contractions over the β indices have been performed.

operator that acts “below” site i to the local operators on site i by performing an appropriate sum over β_i .

For clarity, let us consider a simple example. Given a 2D system of size $2 \times L$ consisting of two rows with L sites each, we can label each site by (i, y) , where $i \in \{1, \dots, L\}$ as usual and $y \in \{1, 2\}$, as depicted in Fig. 3. Consider the Hamiltonian $\hat{H} = \hat{H}_1 + \hat{H}_2 = \sum_i (\hat{A}_{i,1} \hat{B}_{i,2} + \hat{A}_{i,2} \hat{B}_{i+1,2})$, where there are nearest-neighbor interactions between rows 1 and 2 (\hat{H}_1), as well as nearest-neighbor interactions within row 2 (\hat{H}_2). This Hamiltonian can be represented by a simple gMPO (\hat{M}) acting on row 2 along with the complementary operators (\hat{O}) that act locally on the sites in row 1.

Since there are no interactions between sites in row 1, the operators $\{\hat{O}[i, 1]\}$ that are applied in this row take the form of vectors, like those at the ends of a regular MPO, but applied along the β_i index instead of α (see Fig. 3):

$$O_{\beta_i}^{p_i, 1, p_{i,1}}[i, 1] \rightarrow \hat{O}_{\beta_i}[i, 1] = (\hat{I}_{i,1} \quad \hat{A}_{i,1}). \quad (13)$$

To couple these operators with the local operators in row 2, as well as to encode the nearest-neighbor interactions within row 2, gMPO tensors can be used in row 2. They take the form

$$\begin{aligned} \hat{M}_1[i, 2] &= \hat{W}_{\text{NN}}[i], \\ \hat{M}_2[1, 2] &= (\hat{B}_{1,2} \quad \hat{O} \quad \hat{O}), \\ \hat{M}_2[L, 2] &= (\hat{O} \quad \hat{O} \quad \hat{B}_{L,2})^T, \\ \hat{M}_2[1 < i < L, 2] &= \begin{pmatrix} \hat{O} & \hat{O} & \hat{O} \\ \hat{O} & \hat{O} & \hat{O} \\ \hat{B}_{i,2} & \hat{O} & \hat{O} \end{pmatrix}, \end{aligned} \quad (14)$$

where \hat{W}_{NN} is from Sec. II B 1 (with $\hat{C} = \hat{O}$). The reason why the matrix $\hat{M}_2[i, 2]$ takes this form can be understood by explicitly considering what happens during the contraction over β_i for a given column i :

$$\begin{aligned} &\sum_{\beta_i} \hat{O}_{\beta_i}[i, 1] \hat{M}_{\beta_i}[i, 2] \\ &= \hat{I}_{i,1} \cdot \begin{pmatrix} \hat{I}_{i,2} & \hat{O} & \hat{O} \\ \hat{B}_{i,2} & \hat{O} & \hat{O} \\ \hat{O} & \hat{A}_{i,2} & \hat{I}_{i,2} \end{pmatrix} + \hat{A}_{i,1} \cdot \begin{pmatrix} \hat{O} & \hat{O} & \hat{O} \\ \hat{O} & \hat{O} & \hat{O} \\ \hat{B}_{i,2} & \hat{O} & \hat{O} \end{pmatrix} \\ &= \begin{pmatrix} \hat{I} & \hat{O} & \hat{O} \\ \hat{B}_{i,2} & \hat{O} & \hat{O} \\ \hat{A}_{i,1} \hat{B}_{i,2} & \hat{A}_{i,2} & \hat{I} \end{pmatrix}. \end{aligned} \quad (15)$$

The resulting tensor network operator now looks like a regular MPO along row 2 (see Fig. 3), and the form of its matrices looks very similar to \hat{W}_{NN} [Eq. (5)], which encodes nonsymmetric nearest-neighbor interactions. The only difference is that in the place of \hat{C} , the one-body onsite term in Sec. II B 1, there is now the inter-row interaction term \hat{H}_1 for column i . Thus, if these MPO matrices are now all contracted together along the α indices in row 2, we will exactly recover all the terms in our original two-row Hamiltonian.

The function of $\hat{M}_2[i, 2]$ is thus evident: it couples the inter-row interactions into an intrarow MPO matrix in a consistent manner with the structure of the intrarow MPO. Without \hat{M}_2 , the action of $\hat{A}_{i,1}$ could not be selectively coupled into specific matrix elements of \hat{M}_1 . Thus, the form of \hat{M}_2 can be simply determined based on an understanding of the structure of the “in-row” MPO matrix \hat{M}_1 ; namely, to which matrix elements the “external” operators should couple. Although this formalism may appear unnecessarily general in the context of this simple example, its full utility will become apparent in the subsequent sections as more complicated Hamiltonians are considered.

B. Evaluation of PEPO expectation values using gMPOs

To this point, the Hamiltonians under consideration have acted on lattices that are either strictly or quasi one dimensional. In this section we will present an algorithm that utilizes the gMPO formalism to evaluate the expectation value of fully 2D Hamiltonians with the same level of generality as PEPOs, but with simpler and more familiar concepts. This presentation will focus on the case of a finite $L_x \times L_y$ rectangular lattice, but prospects for its extension to the infinite case will be discussed in Sec. V. As is typical in the case of finite PEPS, we make no simplifying assumptions about translation invariance in the wave function, allowing every tensor to be unique.

The concepts for this technique begin with consideration of the three subsets of local operators that are distinguished by a bipartitioning of the system. Namely, given the full system Hamiltonian \hat{H} represented by a localized structure such as a PEPO and a horizontal bipartition of it [as depicted in Fig. 4(a)], all the local operators in \hat{H} can be grouped into three mutually exclusive groups: (i) those for which there are interactions between sites that are all below the line (\hat{H}_{bot}), (ii) all above the line (\hat{H}_{top}), or (iii) those for which interactions occur across the line (\hat{O}_{int}). This decomposition

$$\hat{H} = \hat{H}_{\text{bot}} + \sum_{ij} h_{ij} \hat{O}_i \hat{O}_j + \hat{H}_{\text{top}}, \quad (16)$$

where i indexes sites below the partition, j indexes sites above the partition, and h_{ij} contains the coefficients for the interactions that get “cut,” is a familiar concept in 1D for the analysis of MPOs and is the basis of an efficient implementation of the DMRG algorithm [12]. In 2D, it allows for the evaluation of $\langle \psi | \hat{H} | \psi \rangle$ on the fly using gMPOs.

To see how, first consider the contraction of the finite, two-layer, 2D tensor network corresponding to $\langle \psi | \psi \rangle$ for some PEPS $|\psi\rangle$ using the “boundary MPS” method [31]. Starting from the bottom, the first point of reference is row 1 and as the contraction progresses, it shifts upward to row 2, then row 3, etc. During this process, the Hamiltonian can be

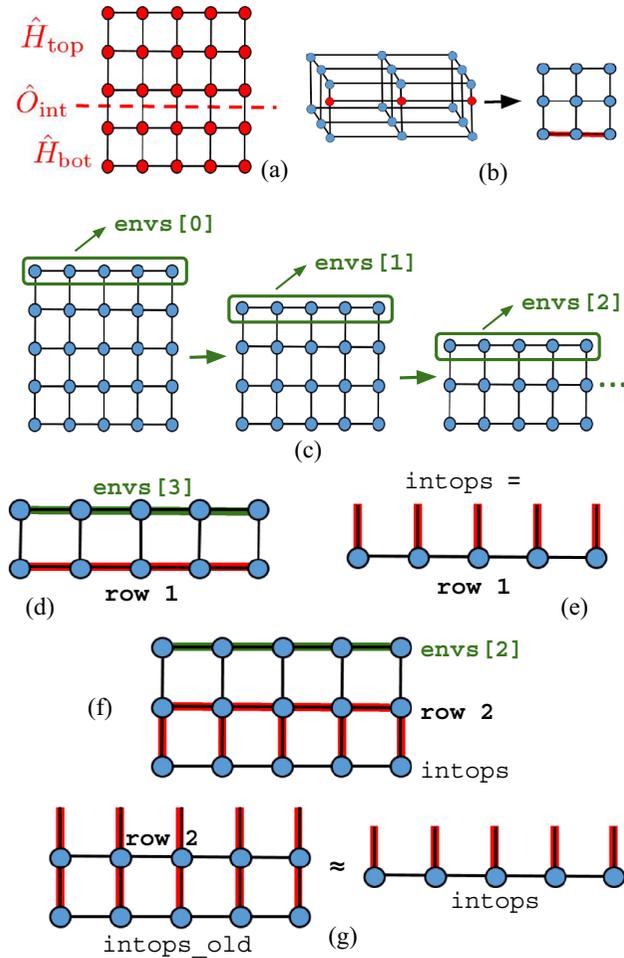


FIG. 4. The first full iteration of the boundary gMPO algorithm for a 5×5 PEPS. (a) A 5×5 PEPO (with physical indices that are suppressed) that is bipartitioned by a cut between rows 2 and 3. (b) A useful diagrammatic definition: when a flat square lattice TN diagram is drawn with some red bonds and some black bonds, the red corresponds to where a tensor network operator has been sandwiched between the bra and ket. Black bonds contain just bra and ket virtual indices. Panels (c)–(g) are the diagrams that directly correspond to the algorithm steps 1–5, respectively (see Sec. III B). Green bonds are used to denote the precomputed environments from step 1.

successively partitioned along with the reference row of the norm contraction, so that the first line lies between rows 1 and 2, then the next is between rows 2 and 3, etc. Using this idea, the total energy $\langle \psi | \hat{H} | \psi \rangle$ can be accumulated as follows (shown graphically in Fig. 4; computational costs given in Appendix A):

(1) Precompute all the partial contractions of $\langle \psi | \psi \rangle$ using the boundary method, starting from the top with row L_y and working downward. They should be stored as $\{\text{envs}[0], \dots, \text{envs}[L_y - 2]\}$ [Fig. 4(c)].

(2) Construct an MPO which contains all the one-body terms in \hat{H} that act locally in row 1 as well as all the interactions between sites in row 1. In other words, this should be the MPO representation of \hat{H}_{bot} when the partition is between row 1 and row 2. Apply this MPO between the

bra and ket tensors of row 1, and evaluate $E_{\text{bot}} = \langle \psi | \hat{H}_{\text{bot}} | \psi \rangle$ by contracting this partial TN with $\text{envs}[L_y - 2]$ [Fig. 4(d)].

(3) Construct complementary operator vectors which contain the local operators \hat{O}_{int} that act in row 1 but have interactions with sites above row 1 (as in Sec. III A). Apply these vectors between the corresponding row 1 ket and bra tensors along the vertical bonds. This partial TN will be called “intops” [Fig. 4(e)].

(4) Shift the partition line up by 1 row (in general, now in-between rows y and $y + 1$). Construct a gMPO to be applied in row y that encodes all the terms in the new \hat{H}_{bot} that have not already been evaluated. Apply the gMPO between the row y bra and ket tensors, and contract this TN with intops (below) and $\text{envs}[L_y - y - 1]$ (above). Add the resulting scalar to E_{bot} to obtain a new E_{bot} , which now accounts for all the terms in $\langle \psi | \hat{H}_{\text{bot}} | \psi \rangle$ given the new partition position. For clarity, the case immediately following step 3 would be when $y = 2$. To accumulate the proper terms, this gMPO should include interactions within row 2, as well as all the interactions between sites in row 2 and sites in the rows beneath it, which is just row 1 for now [$y = 2$ case shown in Fig. 4(f)].

(5) Construct an updated (approximate) intops. This step can be understood as iteratively building up MPOs along the vertical bonds. First, a complementary operator *matrix* (which is just an MPO matrix) is constructed for each column, which relates the \hat{O}_{int} in a given column of row $y - 1$ to the \hat{O}_{int} in the same column of row y . This is exactly like how a regular MPO matrix relates the operators on site $x - 1$ to the operators on site x . Then, these complementary operator matrices are applied between each of the bra and ket tensors of row y along the vertical indices. This row can then be contracted with the old intops and its horizontal bond dimension can be compressed according to the boundary method contraction routine. This yields a new approximate intops that contains the action of all the local operators \hat{O}_{int} that lie below the partition when it is between rows y and $y + 1$ [$y = 2$ case shown in Fig. 4(g)].

(6) Iterate steps 4 and 5 until the top of the PEPS is reached. When the final gMPO is applied to row L_y and contracted with intops, the expectation values of all the terms in \hat{H} will have been tallied in the running total E_{bot} .

Given a Hamiltonian with general interactions of the form $\hat{A}_i \hat{B}_j$, where $i < j$, the big picture of this algorithm (which we will call the “boundary gMPO” method for future reference) can be succinctly summarized as follows: To compute $\langle \psi | \hat{H} | \psi \rangle$, we think about classifying terms in \hat{H} into three nonmutually exclusive groups according to the bipartition of a PEPO between rows y and $y + 1$. Group (1) contains terms where \hat{A} and \hat{B} are both below the partition. Group (2) contains terms where \hat{A} is below the partition but \hat{B} is somewhere above it. Group (3) contains terms where \hat{A} and \hat{B} are both below the previous partition (when it was between rows $y - 1$ and y). At each iteration of the algorithm, we first compute $\langle \psi | \hat{H} | \psi \rangle$ for the set of terms in the difference (1)–(3) by contracting a gMPO with intops, and then we construct a new intops for the next iteration that accounts for all the terms in (2) by slightly modifying the previous intops.

This can be viewed as a “decomposed” contraction of the expectation value of a PEPO. As the partition is iteratively shifted upward, MPOs are sequentially constructed

and applied tensor by tensor along the vertical bonds and gMPOs are applied along the horizontal bonds in order to “extract” the expectation values of the terms in \hat{H}_{bot} , as it is defined based on the current progress of the contraction. When explicitly contracting the expectation value of a PEPO, the boundary tensors accumulate the identical terms but they are not fully evaluated until the entire contraction is complete. By extracting the “completed” terms along the way, the boundary gMPO method allows for the energy evaluation of the same set of general 2D Hamiltonians that can be represented by PEPOs while only invoking MPOs and gMPOs. Since the ideas for constructing MPOs, and thus also gMPOs, are more familiar and well established in the literature than PEPOs, we expect that this will be a useful conceptual simplification.

Additionally, this formulation leads to a reduction in computational cost when compared to the calculation of expectation values using a three-layer network containing a full PEPO. In the case of PEPOs, the algorithm is straightforward, containing only repeated boundary absorption and truncation (analogous to step 5 above). Due to the numerous virtual indices of PEPO tensors, some steps in this procedure have steep scaling with respect to D_{op} , the virtual bond dimension of the PEPO. Namely, during absorption there are operations that have factors of D_{op}^6 , and during truncation (the time-dominant step) there are operations with D_{op}^{10} .² In the boundary gMPO method, since `intops` can always be written with operator virtual indices pointing only in the vertical direction,³ the scalings of the analogous operations (which occur during step 5) with respect to D_{op} are reduced by factors of D_{op}^4 during absorption and D_{op}^9 during truncation. We conclude this discussion of the boundary gMPO algorithm with a few remarks about its properties and practical application: (1) For simplicity, the above discussion was restricted to a Hamiltonian partitioning scheme that started from the bottom of the lattice and ended at the top. However, the concepts can be generalized in a straightforward manner to allow for more detailed partitioning schemes. One possibility is to evaluate some Hamiltonian terms via the bottom-to-top partitions while evaluating other terms with left-to-right partitions. This specific example greatly simplifies the treatment of nearest-neighbor interactions by allowing all terms to be encoded in standard MPOs, eliminating the need for `intops` and gMPOs. Many other schemes for partitioning the Hamiltonian and its terms are also possible. (2) If we restrict our considerations to the special cases of short-ranged Hamiltonians and finite PEPS in which the wave function is constrained to have translation invariance, then in light of (1) the current algorithm has many similarities to that described in Ref. [21]. Detailed comparison of their relative properties and

performance is an interesting research direction. (3) To leading order, the memory consumption of this method is equal to that of existing finite PEPS algorithms for short-ranged Hamiltonians since both require the storage of identical `envs` objects for an efficient implementation [20]. (4) Regarding the optimization of ground-state wave functions, it is natural to use this method in conjunction with direct gradient-based energy minimization techniques [21,22,33] instead of the more common imaginary-time evolution schemes.⁴ Since current formulations of imaginary-time evolution can only efficiently support short-ranged operators, the departure from these techniques is necessary to support general representations of the Hamiltonian.

IV. RESULTS

In this section, we will present the explicit constructions of the MPOs and gMPOs needed to implement the boundary gMPO algorithm described in Sec. III B for various types of 2D Hamiltonians. From the set of Hamiltonians that we explicitly describe, we expect that the construction of most other Hamiltonians of potential interest will be conceptually straightforward. We will also demonstrate the speed and accuracy of the new algorithm, and compare it to the performance of expectation value computations using explicit PEPOs as well as “brute force” application of all the Hamiltonian terms separately (this technique is analogous to the current technique used in 2D simulations, as mentioned in Sec. I).

In our brute force implementations we do not utilize any caching strategies for contraction intermediates that are recyclable between the evaluation of multiple different Hamiltonian terms. This would lead to a faster routine, and might allow for a more direct comparison to the boundary gMPO algorithm since it inherently utilizes a (quite limited) caching strategy. However, while the implementation of the `envs` intermediates in the boundary gMPO method is very straightforward, proper caching for the brute force technique is more complicated, especially for Hamiltonians which include long-range interactions. To keep the results for all Hamiltonians comparable, we thus always refrain from caching in the brute force method.

In all cases we will consider a finite two-dimensional system on a rectangular lattice of $L_x \times L_y$ sites labeled (x, y) , where x indexes the sites in a row and y indexes the sites in a column. By the conventions of the previous sections, $(1,1)$ corresponds to the bottom left corner and (L_x, L_y) to the top right corner. When a sum is taken over all the sites in the lattice using a single index, such as $\sum_{i=1}^{L_x \times L_y}$, the order in which the sites are indexed is such that site $(x+1, y)$ always has a larger label number than site (x, y) , and site $(x, y+1)$ also

²These factors are determined under the assumption that the bond dimension χ of the boundary MPS during boundary method contraction [31] must be proportional to $D^2 D_{\text{op}}$ for accurate results when using a full PEPO [24,25].

³Although this is always possible, it is not required. It may be the case that for some Hamiltonians not explicitly considered in this work, allowing horizontal operator virtual indices in `intops` results in a more efficient representation.

⁴The boundary gMPO algorithm appears to work very nicely within the framework of the newly developed differentiable programming techniques for tensor networks [33]. Since the expectation values of different sets of operators are evaluated during different iterations, each iteration of steps 4 and 5 can be differentiated separately. This allows for the gradient of the energy to also be computed on the fly as the energy itself is being computed, leading to a highly efficient computational formulation.

has a larger label number than (x, y) . This convention will be important when restrictions are placed on the sums, such as the condition $i < j$.

A. Local Hamiltonians

1. Nearest-neighbor interactions

Consider a Hamiltonian with local one-body terms and nonsymmetric nearest-neighbor interactions of the form

$$\hat{H}_{\text{NN}} = \sum_{i=1}^{L_x \times L_y} \hat{C}_i + \sum_{(ij), i < j} \hat{A}_i \hat{B}_j, \quad (17)$$

where both i and j index through all $L_x \times L_y$ sites. The MPO in step 2 of the boundary gMPO algorithm is given by \hat{W}_{NN} from Eq. (5), Sec. II B. The vertical MPOs that are applied tensor by tensor as the algorithm progresses in order to produce intops are given by

$$\begin{aligned} \hat{O}_{\beta_1}[x, 1] &= \begin{pmatrix} \hat{I} & \hat{A} \\ \hat{0} & \hat{0} \end{pmatrix}, \\ \hat{O}_{\beta_{y-1}\beta_y}[x, L_y > y > 1] &= \begin{pmatrix} \hat{I} & \hat{A} \\ \hat{0} & \hat{0} \end{pmatrix}. \end{aligned} \quad (18)$$

Note that here we use the index label β_y to denote its position (y) along the vertical bonds within a column. This is a slight abuse of notation when compared to Sec. III A, where the subscript on β was used to denote its position (x) within a single row. A fully consistent notation would require an x and y subscript on every β , but for all Hamiltonians under consideration in Sec. IV the vertical MPO matrices will be the same for every x , so we always suppress the x label (and sometimes also the y label when the context is unambiguous) on β for simplicity.

Also note that these MPO matrices [Eq. (18)] only need to be of dimension 2×2 because each time a new intops is created, the expectation values of interaction terms with the row above are immediately extracted by contracting it with an appropriate gMPO. Unlike a typical MPO, we therefore never need to “complete” an interaction with a \hat{B} operator in these matrices because that is taken care of in the gMPO. This eliminates the need for the third row and column to account for \hat{B} . In the current case of nearest-neighbor interactions, the bottom row of $\hat{O}_{\beta_{y-1}\beta_y}$ is all $\hat{0}$'s because the action of \hat{A} in row y does not need to be stored once the point of reference is shifted up to row $y + 1$.

The gMPO tensors, used in step 4 to extract the expectation values of terms in \hat{H}_{bot} , were given as the example in Eq. (14). To make the notation consistent with a fully 2D Hamiltonian, the coordinates of the tensors in that expression should be transformed according to $\hat{M}_1[i, 2] \rightarrow \hat{M}_1[x, y > 1]$; $\hat{M}_2[1, 2] \rightarrow \hat{M}_2[1, y > 1]$; $\hat{M}_2[L, 2] \rightarrow \hat{M}_2[L_x, y > 1]$; $\hat{M}_2[i, 2] \rightarrow \hat{M}_2[L_x > x > 1, y > 1]$. Additionally, for generality we do not have $\hat{C} = \hat{0}$ in our current example. In essence, the evaluation of this Hamiltonian's expectation value amounts to performing the same calculation as the one outlined in the example of Sec. III A for every row in the system.

The accuracy of the boundary gMPO algorithm using these tensors to evaluate the expectation value of the given Hamiltonian (with $\hat{C} = \hat{0}$, $\hat{A} = \hat{B} = \sigma_z$) with respect to

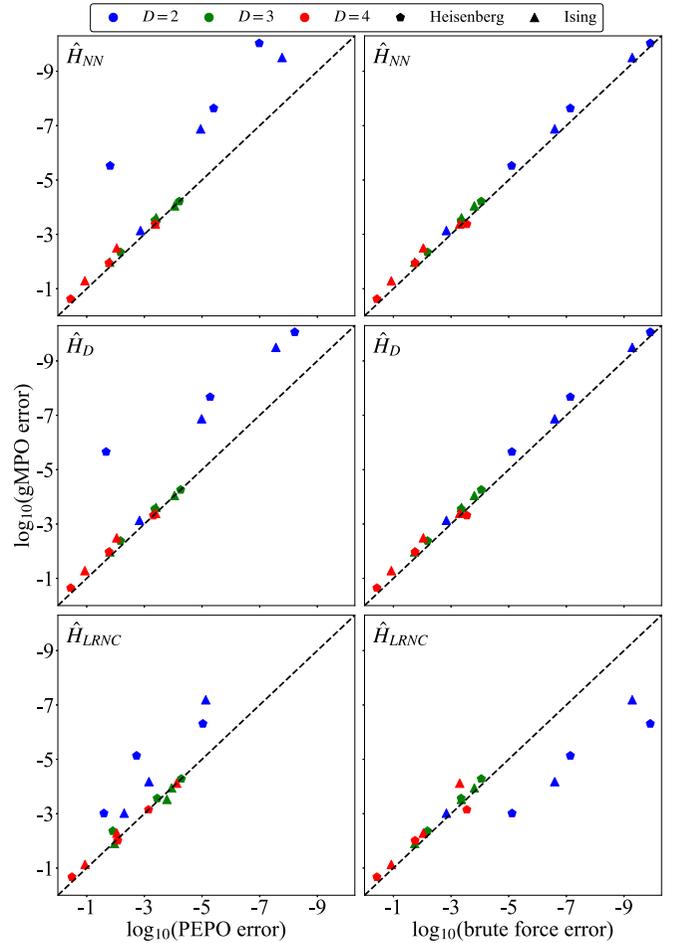


FIG. 5. The relative error of the gMPO-based expectation values (with respect to given trial states) compared to the relative errors obtained using both PEPO-based calculations and the “brute force” technique of evaluating each term in the Hamiltonian separately. For the majority of cases tested, all three techniques exhibit the same level of accuracy. None of these Hamiltonians [Eqs. (17), (19), (22)] contain long-range distance-dependent potentials. The expectation values of all methods are calculated with respect to various 8×8 trial PEPS of bond dimensions $D = 2, 3, 4$. A single point compares the relative error of gMPOs with either PEPOs or brute force, with each technique using the same trial state and χ value (the boundary bond dimension during contraction [31]). All errors are measured with respect to a brute force evaluation that is highly converged in χ . The displayed points are for selected values of χ less than the converged value, in order to compare the levels of accuracy that can be obtained with a given computational effort. For a full picture of the computational effort, these data should be used in conjunction with the speedups reported in Tables I and II, which also include the χ values for each of the points here.

various trial PEPS is shown in Fig. 5. It is almost identically accurate to the brute force scheme and its accuracy is also very similar to the PEPO-based implementation in most cases, with the outliers showing an improved accuracy for the gMPOs. Despite the similar accuracies, using the gMPOs allows for a computational speedup of up to $\sim 30 \times$ over the PEPOs and $\sim 40 \times$ over the brute force implementation, as seen in Tables I and II.

TABLE I. The average computational speedups of the boundary gMPO algorithm over PEPO-based expectation value calculations for a representative set of 2D Hamiltonians [Eqs. (17), (19), (22), and (25)]. The gMPO-based scheme is generically and significantly faster than the PEPOs for all the Hamiltonians except the one with long-range interactions mediated by a distance-dependent potential (LRAC). The reported numbers are averages taken over multiple calculations for each of multiple different trial wave functions: PEPS ground states for the 8×8 AFM Heisenberg model and FM transverse field Ising model ($h = 3.5$). D denotes the bond dimension of the trial PEPSs and χ denotes the maximum boundary bond dimension used during contraction [31]. Both algorithms were implemented in a straightforward manner in order to compare their run times as fairly as possible. These data should be used in conjunction with Fig. 5.

	\hat{H}_{NN}	\hat{H}_D	\hat{H}_{LRNC}	\hat{H}_{LRAC}
$D = 2, \chi = 5$	2.52	3.70	3.66	18.6
$D = 2, \chi = 10$	6.94	11.9	11.6	18.4
$D = 2, \chi = 20$	13.2	27.9	27.2	19.1
$D = 3, \chi = 15$	20.5	39.2	37.5	1.63
$D = 3, \chi = 30$	25.3	52.6	51.7	1.36
$D = 3, \chi = 40$	24.0	50.9	51.2	1.41
$D = 4, \chi = 15$	19.0	33.5	34.2	
$D = 4, \chi = 30$	27.8	59.3	60.3	
$D = 4, \chi = 40$	32.7	62.8	62.5	

2. Diagonal-neighbor interactions

Now, consider a Hamiltonian that has local one-body terms as well as both nearest-neighbor and diagonal-neighbor inter-

TABLE II. The average computational speedups of the boundary gMPO algorithm over “brute force” expectation value calculations for a representative set of 2D Hamiltonians [Eqs. (17), (19), (22), and (25)]. The gMPO-based scheme is significantly faster for all the Hamiltonians under consideration, especially those which contain long-range interactions (LRNC and LRAC). In the brute force technique, the Hamiltonian is evaluated term by term by explicitly applying each pair of local operators. Both algorithms were implemented in a straightforward manner in order to compare their run times as fairly as possible. The reported numbers are averages taken in an identical manner to Table I, and the parameters D and χ are also identically defined. These data should be used in conjunction with Fig. 5.

	\hat{H}_{NN}	\hat{H}_D	\hat{H}_{LRNC}	\hat{H}_{LRAC}
$D = 2, \chi = 5$	60.8	118.4	1066	34.2
$D = 2, \chi = 10$	59.9	107.6	975.7	33.0
$D = 2, \chi = 20$	49.4	84.6	782.2	24.2
$D = 3, \chi = 15$	43.1	72.8	672.3	16.9
$D = 3, \chi = 30$	36.6	64.6	609.6	16.1
$D = 3, \chi = 40$	37.3	65.4	628.9	16.8
$D = 4, \chi = 15$	39.9	61.8	592.6	
$D = 4, \chi = 30$	35.3	63.4	569.8	
$D = 4, \chi = 40$	37.5	68.7	623.8	
$16 \times 16, D = 2, \chi = 20$				296.1

actions, with strengths J_1 and J_2 , respectively,

$$\hat{H}_D = \sum_{i=1}^{L_x \times L_y} \hat{C}_i + J_1 \sum_{\langle ij \rangle, i < j} \hat{A}_i \hat{B}_j + J_2 \sum_{\langle\langle ij \rangle\rangle, i < j} \hat{A}_i \hat{B}_j. \quad (19)$$

Although we are again considering the nonsymmetric Hamiltonian construction (denoted by $i < j$) for simplicity, if the interaction operators are chosen to be symmetric (i.e., $\hat{B} = \hat{A}$), then the given Hamiltonian differs from the truly symmetric one \hat{H}_{sym} (i.e., $i < j \rightarrow i \neq j$) by a factor of 2 in the interaction coefficients $\hat{H}_{\text{sym}}(J_1, J_2) = \hat{H}_D(2J_1, 2J_2)$. If the representation of \hat{H}_{sym} is needed when $\hat{B} \neq \hat{A}$, it can be determined by using the results in this section and following the examples in Sec. II B.

The MPO in step 2 is again given by \hat{W}_{NN} [Eq. (5), Sec. II B], and the vertical MPO matrices are still given by Eq. (18). The gMPO tensors are

$$\begin{aligned} \hat{M}_1[1, y > 1] &= (\hat{C} \quad J_1 \hat{A} \quad \hat{B} \quad \hat{I}), \\ \hat{M}_1[L_x, y > 1] &= (\hat{I} \quad \hat{B} \quad \hat{O} \quad \hat{C})^T, \\ \hat{M}_1[L_x > x > 1, y > 1] &= \begin{pmatrix} \hat{I} & \hat{O} & \hat{O} & \hat{O} \\ \hat{B} & \hat{O} & \hat{O} & \hat{O} \\ \hat{O} & \hat{O} & \hat{O} & \hat{O} \\ \hat{C} & J_1 \hat{A} & \hat{B} & \hat{I} \end{pmatrix}, \quad (20) \\ \hat{M}_2[1, y > 1] &= (J_1 \hat{B} \quad J_2 \hat{I} \quad \hat{O} \quad \hat{O}), \\ \hat{M}_2[L_x, y > 1] &= (\hat{O} \quad \hat{O} \quad J_2 \hat{I} \quad J_1 \hat{B})^T, \\ \hat{M}_2[L_x > x > 1, y > 1] &= \begin{pmatrix} \hat{O} & \hat{O} & \hat{O} & \hat{O} \\ \hat{O} & \hat{O} & \hat{O} & \hat{O} \\ J_2 \hat{I} & \hat{O} & \hat{O} & \hat{O} \\ J_1 \hat{B} & J_2 \hat{I} & \hat{O} & \hat{O} \end{pmatrix}. \end{aligned}$$

These matrices can be understood by noting the similarity between \hat{M}_1 and $\hat{W}_{\text{NN-sym}}$ from Eq. (6). The only difference is that in \hat{M}_1 the entry for \hat{A} in the first column is made to be \hat{O} (and interaction coefficients are included). This is done to prevent symmetric nearest-neighbor interactions of the form $\hat{B}_{x-1,y} \hat{A}_{x,y}$ from being included along the gMPO row (in these coordinates the gMPO is being applied to row y). However, since the sites are ordered in such a way that the (nonsymmetric) diagonal-neighbor interactions occur between site $(x, y - 1)$ and sites $(x - 1, y)$, $(x + 1, y)$, we still want to include the action of \hat{B} “on the left” on site $(x - 1, y)$. This is exactly what the form of $\hat{W}_{\text{NN-sym}}$ is designed to do.

If the \hat{M}_1 matrices were the only ones included in the gMPO, then this action of $\hat{B}_{x-1,y}$ “on the left” would never be utilized due to the \hat{O} in place of $\hat{A}_{x,y}$ in the first column. However, \hat{M}_2 couples the action of $\hat{A}_{x,y-1}$ (from int ops) into the two typical locations of \hat{A} in $\hat{W}_{\text{NN-sym}}$ (and also multiplies by J_2). This allows the “on the left” action of $\hat{B}_{x-1,y}$ to interact with the action of $J_2 \hat{A}_{x,y-1}$, which is exactly the diagonal interaction that we want to include. \hat{M}_2 also couples $J_2 \hat{A}_{x,y-1}$ into the same position as $J_1 \hat{A}_{x,y}$ in \hat{M}_1 , which allows for the nearest-neighbor horizontal interaction and diagonal-neighbor interaction “to the right” to be accounted for simultaneously. Specifically, after the β_{y-1} indices have been appropriately contracted over, the subsequent contraction over an α index

will yield a term like $(J_2\hat{A}_{x,y-1} + J_1\hat{A}_{x,y})\hat{B}_{x+1,y}$. For clarity, in the spirit of the example in Eq. (15), a typical contraction over the β_{y-1} index (with $\hat{C} = \hat{0}$) would look like

$$\begin{aligned} & \sum_{\beta_{y-1}} \hat{O}_{\beta_{y-1}}[x, y-1] \hat{M}_{\beta_{y-1}}[x, y] \\ &= \hat{I} \cdot \begin{pmatrix} \hat{I}_{x,y} & \hat{0} & \hat{0} & \hat{0} \\ \hat{B}_{x,y} & \hat{0} & \hat{0} & \hat{0} \\ \hat{0} & \hat{0} & \hat{0} & \hat{0} \\ \hat{0} & J_1\hat{A}_{x,y} & \hat{B}_{x,y} & \hat{I}_{x,y} \end{pmatrix} \\ &+ \hat{A}_{x,y-1} \cdot \begin{pmatrix} \hat{0} & \hat{0} & \hat{0} & \hat{0} \\ \hat{0} & \hat{0} & \hat{0} & \hat{0} \\ J_2\hat{I}_{x,y} & \hat{0} & \hat{0} & \hat{0} \\ J_1\hat{B}_{x,y} & J_2\hat{I}_{x,y} & \hat{0} & \hat{0} \end{pmatrix} \\ &= \begin{pmatrix} \hat{I} & & \hat{0} & \hat{0} \\ \hat{B}_{x,y} & & \hat{0} & \hat{0} \\ J_2\hat{A}_{x,y-1} & & \hat{0} & \hat{0} \\ J_1\hat{A}_{x,y-1}\hat{B}_{x,y} & J_1\hat{A}_{x,y} + J_2\hat{A}_{x,y-1} & \hat{B}_{x,y} & \hat{I} \end{pmatrix}. \quad (21) \end{aligned}$$

The form of this gMPO, which is the simplest case where \hat{A} can interact with a \hat{B} from a different row *and* column, is the basis for generating all the more complicated finite-range 2D Hamiltonians with interactions between more distant neighbors. In essence, the form of \hat{M}_1 has to be adapted to the desired pattern of operators within the gMPO row, and then $\hat{M}_2, \hat{M}_3, \dots$, etc., take the forms which properly couple the operators from the vertical MPOs (intops) into \hat{M}_1 . For a general construction of this form that includes all nonsymmetric interactions between neighbors up to range R , see Appendix B.

The speed and accuracy of the boundary gMPOs using these tensors (with $\hat{C} = \hat{0}$, $\hat{A} = \hat{B} = \sigma_z$, $J_2 = J_1/2$) is compared to a PEPO-based implementation and a brute force implementation in Fig. 5 and Tables I and II. The gMPOs produce accuracies which are nearly identical to the brute force scheme, but with a computational effort that is $\sim 60\text{--}70\times$ less. When compared to PEPOs, a speedup of up to $\sim 50\times$ is observed and in most cases the gMPOs and PEPOs also produce the same level of accuracy. In cases where they differ, the gMPOs are observed to be more accurate.

B. Long-range Hamiltonians with no coefficients

We will now consider a Hamiltonian which has local one-body terms and nonsymmetric pairwise interactions of equal strength between every site on the lattice. This can be viewed as the 2D version of the Hamiltonian represented by \hat{W}_{uniform} (see Sec. II B 2). We have

$$\hat{H}_{\text{LRNC}} = \sum_i \hat{C}_i + \sum_{i<j} \hat{A}_i \hat{B}_j. \quad (22)$$

The MPO used in step 2 of the boundary gMPO algorithm is given by \hat{W}_{uniform} . The vertical MPOs used for the construction of intops are given by

$$\begin{aligned} \hat{O}_{\beta_1}[x, 1] &= (\hat{I} \quad \hat{A}), \\ \hat{O}_{\beta_{y-1}\beta_y}[x, L_y > y > 1] &= \begin{pmatrix} \hat{I} & \hat{A} \\ \hat{0} & \hat{I} \end{pmatrix}. \quad (23) \end{aligned}$$

These MPO matrices differ from those in Eq. (18) because they “remember” the action of all the local operators in a given column x . In Eq. (18), the contractions over β_{y-1} that are performed in step 5 result in operator vectors of the form $(\hat{I}, \hat{A}_{x,y})$. This was sufficient because the previous Hamiltonians under consideration were local, so the action of the $\hat{A}_{x,y-1}, \hat{A}_{x,y-2}, \dots$, etc. operators had already been completely accounted for by the time the reference row was shifted up by one. However, in our current Hamiltonian the interactions are long-ranged, so the action of all the local operators in a given column must be accounted for in a single intops tensor. This is achieved by the MPO matrices in Eq. (23), for which a contraction over $\beta_1, \beta_2, \dots, \beta_{y-1}$ yields operator vectors of the form $(I, \hat{A}_{x,1} + \hat{A}_{x,2} + \dots + \hat{A}_{x,y})$.

The corresponding gMPO tensors are given by

$$\begin{aligned} \hat{M}_1[1, y > 1] &= (\hat{C} \quad \hat{A} \quad \hat{B} \quad \hat{I}), \\ \hat{M}_1[L_x, y > 1] &= (\hat{I} \quad \hat{B} \quad \hat{0} \quad \hat{C})^T, \\ \hat{M}_1[L_x > x > 1, y > 1] &= \begin{pmatrix} \hat{I} & \hat{0} & \hat{0} & \hat{0} \\ \hat{B} & \hat{I} & \hat{0} & \hat{0} \\ \hat{0} & \hat{0} & \hat{I} & \hat{0} \\ \hat{C} & \hat{A} & \hat{B} & \hat{I} \end{pmatrix}, \\ \hat{M}_2[1, y > 1] &= (\hat{B} \quad \hat{I} \quad \hat{0} \quad \hat{0}), \\ \hat{M}_2[L_x, y > 1] &= (\hat{0} \quad \hat{0} \quad \hat{I} \quad \hat{B})^T, \\ \hat{M}_2[L_x > x > 1, y > 1] &= \begin{pmatrix} \hat{0} & \hat{0} & \hat{0} & \hat{0} \\ \hat{0} & \hat{0} & \hat{0} & \hat{0} \\ \hat{I} & \hat{0} & \hat{0} & \hat{0} \\ \hat{B} & \hat{I} & \hat{0} & \hat{0} \end{pmatrix}. \quad (24) \end{aligned}$$

Note that this result is nearly identical to the gMPO tensors in the previous section for diagonal interactions [Eq. (20)]. The only difference is the replacement of two \hat{O} 's with \hat{I} 's in \hat{M}_1 (and the removal of the interaction coefficients). The reason for this similarity can be understood in two distinct ways. First, the addition of these identities can be viewed as an elevation of the symmetric nearest-neighbor interactions in $\hat{W}_{\text{NN-sym}}$ to symmetric interactions of arbitrary range, which captures all the new terms in \hat{H}_{LRNC} . Second, we can see a direct analogy between the relations of the current \hat{M}_1 to $\hat{W}_{\text{uniform-sym}}$ (Sec. II B 2) and the previous \hat{M}_1 [Eq. (20)] to $\hat{W}_{\text{NN-sym}}$. In other words, in the previous section we argued that because \hat{M}_1 only differed from $\hat{W}_{\text{NN-sym}}$ by a single element, it was clear that it would encode the symmetric nearest-neighbor action of \hat{B} about site (x, y) that was necessary to generate the diagonal interactions. Now in the current case, we replace the modified $\hat{W}_{\text{NN-sym}}$ with an identically modified $\hat{W}_{\text{uniform-sym}}$ to obtain the symmetric action of \hat{B} on *all* sites to the left and right of (x, y) . This is precisely the pattern of operators that needs to be encoded in order to generate all the terms in \hat{H}_{bot} .

The performance of the boundary gMPOs using these tensors (with $\hat{C} = \hat{0}$, $\hat{A} = \hat{B} = \sigma_z$) is compared to a PEPO-based implementation and a brute force implementation in Fig. 5 and Tables I and II. In this case, due to the long-range nature of the interactions, the scaling of our brute force evaluation is $O(N^3)$. While this can be slightly reduced with appropriate caching of contraction intermediates, the gMPO- and PEPO-based techniques only scale as $\sim O(N)$ (where N is the total

number of sites in the system). Thus, in addition to the $\sim 60\times$ speedup over the PEPOs, the gMPOs attain large speedups of $\sim 600\times$ over the brute force algorithm for the $N = 64$ cases that we consider. For larger systems, this speedup will grow rapidly. Given this poor scaling and the fact that the gMPOs can reproduce the accuracy of the brute force calculations in all of the most challenging test cases, it is clear that the brute force technique is not a viable approach to study systems with nonlocal interactions. Of the two viable strategies, gMPOs show very similar accuracy to PEPOs across most of the test cases, as in the previous sections.

C. Long-range isotropic Hamiltonians with approximate coefficients

In the previous section, we demonstrated an exact and compact representation of a long-range interacting 2D Hamiltonian when the interaction coefficients were all the same (this can also be done with a PEPO [25]). Despite this, it is a challenging problem to efficiently⁵ represent a 2D Hamiltonian which has long-range interaction coefficients that depend on the distance between sites, even in an approximate manner [24,25,34,35]. Various solutions to this problem have been proposed recently [25,34,35], but they all require the explicit use of PEPOs, making their computational cost high.

The introduction of the gMPO formalism allows for a different, simpler approach to be derived, which we will show to be many orders of magnitude more accurate and efficient than the PEPO-based approaches. We will consider a restricted case of the general long-range interacting Hamiltonian on the 2D lattice,

$$\hat{H}_{\text{LRAC}} = \sum_i \hat{C}_i + \sum_{i<j} V_{ij} \hat{A}_i \hat{B}_j, \quad (25)$$

where V is a translation-invariant, decaying function of the Euclidean distance between sites i and j (i.e., it is isotropic).

The crux of the long-range interaction problem on the 2D lattice is that functions of the Euclidean distance $f(\sqrt{x^2 + y^2})$, which are necessary for physical potentials V , are difficult to represent efficiently within a tensor network structure [25,34]. Although 1D functions of x and y can be independently constructed with ease (see Sec. II B 2), the known possibilities for combining them within a 2D tensor network ansatz yield functions of the Manhattan distance $f(|x| + |y|)$ or product functions $f(x)g(y)$, but not the desired radially symmetric ones $f(r) = f(\sqrt{x^2 + y^2})$. However, the Gaussian function $e^{-\lambda r^2}$ has the unique property that $f(x)f(y) = e^{-\lambda(x^2+y^2)} = f(r^2)$. This connection allows for a radially symmetric Gaussian function in 2D to be created from the product of two 1D Gaussians $f(x) = e^{-\lambda x^2}$ and $f(y) = e^{-\lambda y^2}$.

This observation can be directly exploited by the gMPO-based algorithm. If the vertical MPOs encode the interactions $\hat{H}_1 = \sum_{i=1}^{L_y} \sum_{j>i} e^{-\lambda(j-i)^2} (\hat{A}_i \hat{I}_j + \hat{A}_i \hat{B}_j)$ and the gMPOs encode horizontal interactions of the form

$\hat{H}_2 = \sum_{i=1}^{L_x} \sum_{j>i} e^{-\lambda(j-i)^2} (\hat{I}_i \hat{B}_j + \hat{B}_i \hat{I}_j + \hat{A}_i \hat{B}_j)$, then they can be combined as a product (as in Secs. IV A 2 and IV B) to make complete interactions of the form $e^{-\lambda(a^2+b^2)} \hat{A}_{x,y} \hat{B}_{x+a,y\pm b} + e^{-\lambda a^2} \hat{A}_{x,y} \hat{B}_{x+a,y} + e^{-\lambda b^2} \hat{A}_{x,y} \hat{B}_{x,y+b}$. The two-dimensional, radially symmetric Gaussians can then be used as a basis to fit the desired long-range potential

$$V(x, y) \approx \sum_{k=1}^K c_k e^{-\lambda_k(x^2+y^2)}, \quad (26)$$

which is a well-studied problem with highly accurate, compact solutions when V smoothly decays with distance [38–40]. The expectation value of the desired Hamiltonian can then be evaluated as the sum over the expectation values obtained using K different sets of vertical MPOs and gMPOs (for the K different values of λ). Since the only requirement of this technique is the representation of 1D Gaussian functions, this basis can be encoded directly within the MPO and gMPO tensors, which completely avoids the conceptual and computational complexity of introducing fictitious superlattices, as in Refs. [25,34].

Unfortunately, there is no known exact, compact representation of a 1D MPO with pairwise Gaussian interactions. However, it can be generated in a nearly numerically exact manner using the method outlined in Sec. II B 3 to create \hat{W}_{gen} . Figure 7(a) shows the required bond dimension for the Gaussian MPO for different values of λ . The result that $D_{\text{op}} = 14$ in the worst case for an accuracy of $\sim 10^{-10}$ is a modest bond dimension for an MPO, which is what makes the current approach of using an exact Gaussian basis amenable to the gMPO algorithm. Although this same scheme could, in principle, be implemented using PEPOs on the same lattice, it would require the use of PEPOs with $D_{\text{op}} = 28$ in the worst case. In practice, the factor of D_{op}^7 in the computational cost of PEPO-based contractions makes a PEPO with a bond dimension of this size unusable. However, since the use of gMPOs reduces the dependence of the cost on the operator bond dimension to at most D_{op}^3 (in step 4), and D_{op}^1 in the most time-intensive step (compression in step 5), using this bond dimension for the vertical MPOs and gMPOs is entirely feasible.

The explicit forms of the tensors in this case can be viewed as a direct generalization of the tensors from the previous section (Sec. IV B), Eqs. (23) and (24). This follows from the discussion in Sec. II B 3 regarding \hat{W}_{gen} and $\hat{W}_{\text{gen-sym}}$ as direct generalizations of \hat{W}_{uniform} and $\hat{W}_{\text{uniform-sym}}$, respectively. Since the tensors in Sec. IV B are derived from $\hat{W}_{\text{uniform(-sym)}}$ and in the current case we want to use tensors based on the $\hat{W}_{\text{gen(-sym)}}$ representation of a Gaussian MPO, the tensors in (23) and (24) generalize to the current case in an analogous way to the $\hat{W}_{\text{uniform(-sym)}} \rightarrow \hat{W}_{\text{gen(-sym)}}$ generalization of Sec. II B 3.

Specifically, the MPO for step 2 is the \hat{W}_{gen} representation of Gaussian interactions with exponential coefficient λ_k , which is determined from the algorithm in Ref. [37]. From this MPO, the data for each \vec{v}_i , \vec{w}_i , and X_i can be extracted [according to Eq. (11)]. These can then be used to construct the other tensors for pairwise interactions mediated by a 2D

⁵Here, we define “efficient” to mean that the computational cost to evaluate the expectation value of the Hamiltonian scales linearly with the number of sites in the system.

Gaussian potential. The vertical MPO tensors are given by

$$\begin{aligned}\hat{O}_{\beta_1}[x, 1] &= (\hat{I} \quad (w_1)_c \hat{A}), \\ \hat{O}_{\beta_{y-1}\beta_y}[x, L_y > y > 1] &= \begin{pmatrix} \hat{I} & (w_y)_c \hat{A} \\ \hat{0} & (X_y)_{c'c} \hat{I} \end{pmatrix},\end{aligned}\quad (27)$$

where c and c' index through the vector \vec{w}_y and matrix X_y , like in Eq. (11). The gMPO tensors are

$$\begin{aligned}\hat{M}_1[1, y > 1] &= (\hat{C}, \quad (w_1)_{a'} \hat{A}, \quad (w_1)_{b'} \hat{B}, \quad \hat{I}), \\ \hat{M}_1[L_x, y > 1] &= (\hat{I}, \quad (v_{L_x})_a \hat{B}, \quad \hat{0}, \quad \hat{C})^T, \\ \hat{M}_1[L_x > x > 1, y > 1] &= \begin{pmatrix} \hat{I} & \hat{0} & \hat{0} & \hat{0} \\ (v_x)_{a'} \hat{B} & (X_x)_{aa'} \hat{I} & \hat{0} & \hat{0} \\ \hat{0} & \hat{0} & (X_x)_{bb'} \hat{I} & \hat{0} \\ \hat{C} & (w_x)_{a'} \hat{A} & (w_x)_{b'} \hat{B} & \hat{I} \end{pmatrix}, \\ \hat{M}_c[1, y > 1] &= ((v_y)_c \hat{B}, \quad (v_y)_c \cdot (w_1)_{a'} \hat{I}, \quad \hat{0}, \quad \hat{0}), \\ \hat{M}_c[L_x, y > 1] &= (\hat{0}, \quad \hat{0}, \quad (v_y)_c \cdot (v_{L_x})_b \hat{I}, \quad (v_y)_c \hat{B})^T, \\ \hat{M}_c[L_x > x > 1, y > 1] &= \begin{pmatrix} \hat{0} & \hat{0} & \hat{0} & \hat{0} \\ \hat{0} & \hat{0} & \hat{0} & \hat{0} \\ (v_y)_c \cdot (v_x)_b \hat{I} & \hat{0} & \hat{0} & \hat{0} \\ (v_y)_c \hat{B} & (v_y)_c \cdot (w_x)_{a'} \hat{I} & \hat{0} & \hat{0} \end{pmatrix}, \\ c &\in \{2, 3, \dots, \text{len}(\vec{w}_{y-1}) + 1\}.\end{aligned}\quad (28)$$

Here, c is used consistently between Eqs. (27) and (28) to index the vertical MPO bond β . In an identical manner to Eq. (12), a, a', b, b' are used to index the coefficient vectors \vec{w}_x, \vec{v}_x and the coefficient matrix X_x . Note that within a given \hat{M}_c matrix, the value of c is fixed while the values of a, a', b, b' range appropriately over the dimensions of the matrix. This means that an expression such as $(v_y)_c \cdot (v_x)_b$ in Eq. (28) is a scalar multiplying a vector.

For these expressions to always make sense, we require $L_x \geq L_y$ so that the bottom MPO is long enough to extract all the necessary coefficient vectors and matrices for the vertical direction. The crucial component of this representation is how \vec{w} and \vec{v} appear in \hat{M}_c . For the \hat{I} operator in the bottom row of the matrix, which couples the action of \hat{A} (from below) to the action of \hat{B} (to the right) in the gMPO row, the ‘‘completion’’ interaction coefficients \vec{v}_y are encoded along the β index while the ‘‘beginning’’ interaction coefficients \vec{w}_x are encoded along the α_x index. Similarly, for the \hat{I} operator in the first column of the matrix, which couples the action of \hat{A} (from below) to the action of \hat{B} (to the left) in the gMPO row, the ‘‘completion’’ interaction coefficients \vec{v}_y are encoded along the β index while the ‘‘completion’’ interaction coefficients \vec{v}_x are encoded along the α_{x-1} index. This formulation allows for vertical interactions of the form $\sum_{i=1}^y e^{-\lambda(y-i)^2} \hat{A}_i \hat{I}_y$ to be ‘‘completed’’ and thus scalar multiplied by ‘‘completed’’ horizontal interactions of the form $\sum_{i=x+1}^L e^{-\lambda(i-x)^2} \hat{I}_x \hat{B}_i$ and $\sum_{i=1}^x e^{-\lambda(x-i)^2} \hat{B}_i \hat{I}_x$, yielding the desired 2D Gaussian potential. The other entries of the tensors can be understood by their analogous form to the previous section and their direct correspondence with $\hat{W}_{\text{gen-sym}}$ [Eq. (12)].

Due to the inherent challenge of explicitly writing and interpreting the algebraic expressions for \hat{M} when the dimension of the β index is greater than 2, it can be more

intuitive to understand the form of these gMPO tensors from a graphical presentation, which is given in Fig. 6. Additionally, a straightforward example implementation of these tensors can be found online [41].

The performance of this scheme relative to the PEPO-based scheme from Ref. [25] for evaluating the expectation value of $\hat{H} = \sum_{i < j} (\sigma_i^z \sigma_j^z) / |\mathbf{r}_i - \mathbf{r}_j|$ is shown in Fig. 7 and Table I. One notable difference between this case and the previous sections is that there is no longer a generic, guaranteed speedup of the gMPOs over PEPOs because the two methods work differently. The PEPOs encode long-range coefficients by introducing a large auxiliary lattice, while the gMPOs do so by using an increased bond dimension. Since these things affect the computational scaling in different ways and their precise costs depend on specific numerical thresholds, one method is not strictly faster than the other.

However, in practice we observe that the gMPOs are many orders of magnitude more computationally efficient than the PEPOs. The simplest way to see this is to first note that for given values of K, χ , and $D > 2$, the CF-PEPO and gMPO schemes require similar levels of computational effort (Table I). Yet, with these same parameters, the gMPOs are approximately four orders of magnitude more accurate than the CF-PEPOs (Fig. 7). This can be extended to recognize that in order to obtain a given level of accuracy, the gMPOs will be many orders of magnitude faster than the PEPOs, or more generally that the gMPOs can obtain a more accurate answer than the PEPOs in less time.

Additionally, the convergence toward high accuracy is faster and more straightforward when using gMPOs than when using PEPOs. In the case of the gMPOs, the accuracy is systematically governed by χ and K [see Figs. 7(b) and 7(c)]. This becomes clear by observing that, for a given trial state,

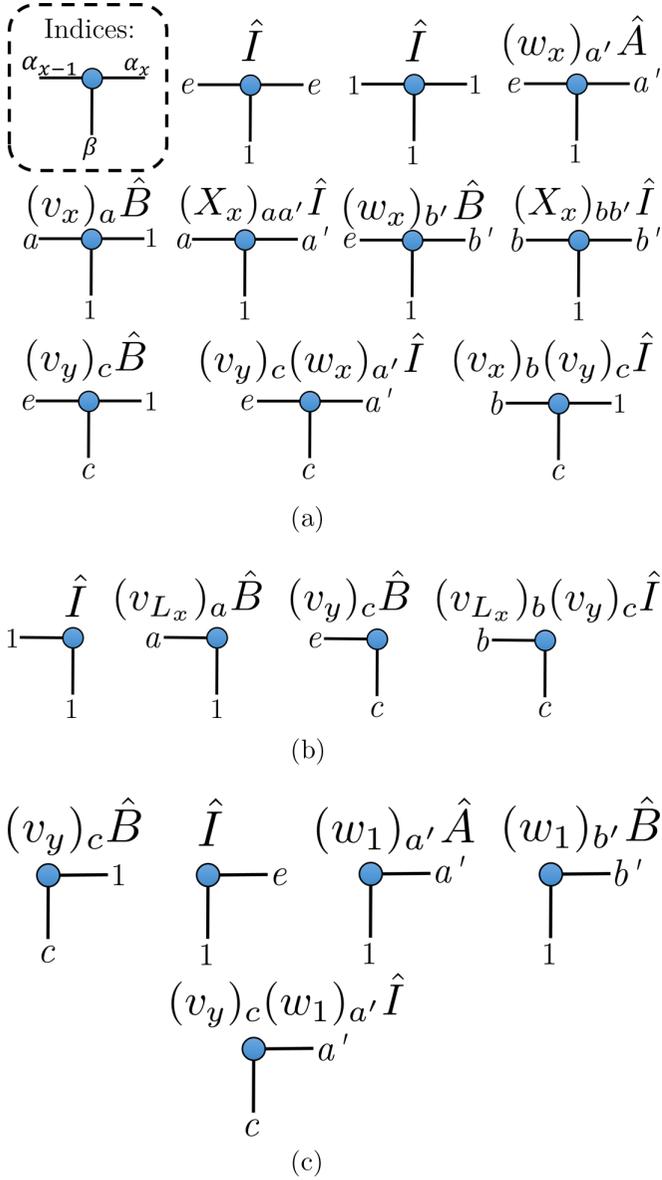


FIG. 6. The gMPO tensors $\hat{M}[x, y > 1]$ for long-range Gaussian interactions [Eq. (28)]. (a) For $1 < x < L_x$, (b) for $x = L_x$, and (c) for $x = 1$. Since the dimension of each bond α_{x-1} , α_x , and β can vary depending on x , y , and the value of the exponential coefficient λ in the Gaussian interaction, we will use symbols to label specific values of the indices. For a horizontal (α) index of bond dimension $2g + 2$ that takes index values $\{1, 2, \dots, 2g + 2\}$, we label the first value by “1,” the next g values by a (if the bond points left) or a' (if the bond points right), the next g values by b (if the bond points left) or b' (if the bond points right), and the final value by e . This is the convention that is explained in Eq. (12) and is also used in Eq. (28). For a vertical (β) index of bond dimension $g + 1$ that takes index values $\{1, 2, \dots, g + 1\}$, we label the first element by “1” and the remaining g elements by c . This corresponds directly with Eqs. (27) and (28).

its curve in either Fig. 7(b) or 7(c) is always decreasing. On the other hand, the convergence of the PEPO curves stall. The medium- and high-accuracy regimes are not bounded by errors due to the basis size or χ , but instead by larger

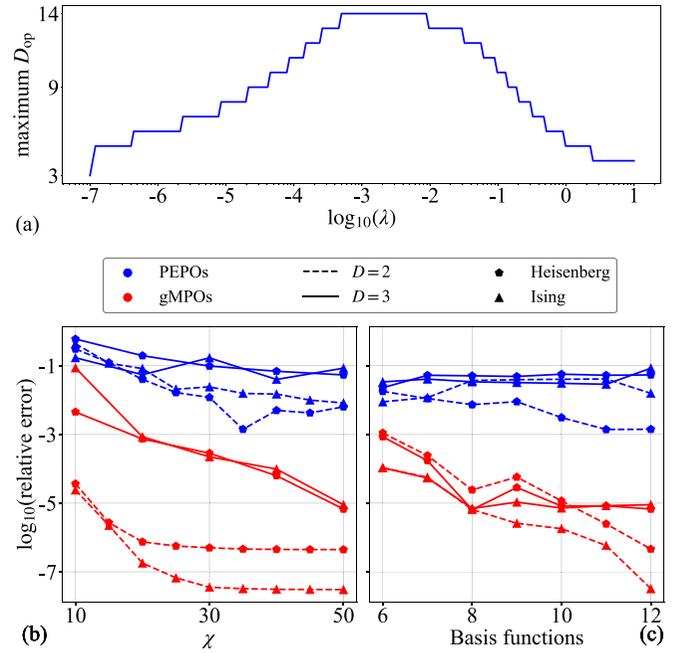


FIG. 7. (a) The maximum bond dimension of a numerically exact $L = 250$ MPO representation of long-range pairwise Gaussian interactions for various values of the exponential coefficient λ . The algorithm from [37] was used with a singular value threshold of 10^{-10} . (b), (c) The relative error in the computed expectation value of $\hat{H} = \sum_{i < j} \sigma_i^z \sigma_j^z / |\mathbf{r}_i - \mathbf{r}_j|$ for 8×8 , $D = 2$ (dashed line) and 3 (solid line) ground states of the AFM Heisenberg model (pentagons) and FM transverse-field Ising model (triangles). We compare the Gaussian gMPO technique (red) with the CF-PEPO technique (blue) from Ref. [25]. In (b) we use 12 basis functions to fit the Coulomb potential and vary the boundary dimension χ of the contraction algorithm [31]. In (c) we fix $\chi = 35$ for $D = 2$, $\chi = 50$ for $D = 3$, and vary the number of basis functions K . The convergence of the gMPOs is rapid and strictly governed by K and χ [for a given trial state, either the curve in (b) or in (c) is always decreasing], while the PEPOs converge slowly and become saturated by other sources of numerical error.

numerical errors stemming from additional complicated parameters involved with making the basis radially symmetric [25,34]. In fact, this is the inherent reason for the major accuracy difference. The gMPO Gaussian basis is radially symmetric up to $\sim 10^{-10}$ (the singular value threshold used in the approximation algorithm), whereas the PEPO bases are only radially symmetric up to significant numerical errors [25,34].

As a final point, we note that a slightly faster implementation of this long-range gMPO scheme is possible. Since the bond dimensions D_{op} reported in Fig. 7(a) are only for \hat{W}_{gen} , the horizontal bond dimension of the gMPO tensors in Eq. (28) is almost twice as large. A factor of ~ 2 speedup can be gained in step 4 of the boundary gMPO algorithm if nonsymmetric gMPO tensors of horizontal dimension D_{op} are used instead, so that the interactions $\hat{A}_{x,y} \hat{B}_{x,y+b} + \hat{A}_{x,y} \hat{B}_{x+a,y} + \hat{A}_{x,y} \hat{B}_{x+a,y+b}$ are encoded in one gMPO and the interactions $\hat{A}_{x,y} \hat{B}_{x-a,y+b}$ in another. The cost of this bond dimension reduction is an increase in the number of gMPOs that need to

be independently evaluated from K to $2K$, but this still leaves a factor of 2 for the speedup because the cost of step 4 depends quadratically on the horizontal bond dimension of the gMPOs.

V. CONCLUSIONS

In this work we have presented an algorithm which can evaluate the expectation value of general 2D Hamiltonians without using a PEPO. To accomplish this, we introduced the formalism of a gMPO and showed how it can be used in combination with MPOs to efficiently compute the energy of a PEPO on the fly. In addition to the conceptual simplification of rewriting PEPOs in terms of the more familiar MPOs, we also showed that computing the energy using this strategy is one to two orders of magnitude faster while being equally as accurate as explicitly using a PEPO. The structure of the algorithm also allows for a different technique to be used for constructing and evaluating 2D Hamiltonians with physical long-range interaction potentials, which we demonstrated to be multiple orders of magnitude more accurate and efficient than existing strategies. We expect that this work will lower the computational and conceptual barriers to using tensor network operators in future PEPS calculations. We hope that this opens the door to the study of new, more complicated Hamiltonians in the tensor network community.

Finally, although this work focused on the specific case of finite systems and rectangular lattices, we expect that the concepts can be generalized in multiple directions. Primarily, extension to the case of infinite rectangular lattices (iPEPS) [18] appears attainable given that the fundamental requirement for applying the algorithm in its current formulation is simply that the contraction method starts from the boundary. Since most prominent contraction methods for iPEPS also begin from the boundary [19,21,22,42] and much is known about infinite MPOs [43–45], many of the necessary components are already understood. There is also potential to extend this work to other regular finite lattices that are not rectangular. In this case, the direct analogies to MPOs used in this work are lost because the lattice connectivity changes the number of tensor indices. However, the general concepts still remain applicable and efficient as long as the structure of the underlying tensor network state is regular and it can be contracted starting from the boundaries.

ACKNOWLEDGMENTS

Primary support for this work was from AFOSR MURI Grant No. FA9550-18-1-0095. M.J.O. acknowledges financial support from a US National Science Foundation Graduate Research Fellowship via Grant No. DEG-1745301. G.K.C. acknowledges support from the Simons Foundation. The authors thank H. Schurkus and Z. Li for helpful feedback on the manuscript.

APPENDIX A: COMPUTATIONAL COSTS

In this Appendix we will describe the computational costs involved with each step of the boundary gMPO algorithm presented in Sec. III B and Fig. 4. As usual with boundary style algorithms, each time that a row is contracted into the

boundary, the bond dimension of the boundary tensors is then truncated to a constant value χ in order to prevent the exponential growth of contraction costs [31]. In this Appendix, we make two typical assumptions: (1) that the physical bond dimensions can be neglected in computing the costs due to their small, fixed size; and (2) that $\chi \sim D^2$ when truncating bonds corresponding to the two-layer tensor network for $\langle \psi | \psi \rangle$. Here, D is the bond dimension of the PEPS $|\psi\rangle$. We also use D_{op} to refer to the virtual bond dimension of the operators such as horizontal and vertical MPOs as well as gMPOs.

Step 1: The leading cost of the contraction and truncation required to absorb the boundary environment at position y (horizontal bond dimension χ) into the row at position $y - 1$ to form the subsequent boundary environment is given by $\sim O(L_x D^5 \chi^3) \approx O(L_x D^{11})$. This scaling is obtained by considering the bra and ket layers of row $y - 1$ separately and performing a truncation after absorbing each part.

Step 2: The leading cost of the contraction depicted in Fig. 4(d) to compute E_{bot} for the first row is $\sim O(L_x D^8 D_{\text{op}}^2)$. This scaling is obtained using a MPS zipper-style contraction.

Step 3: The construction of the first intops simply requires the contraction of an operator vector in-between the bra and ket PEPS. Up to factors of the physical indices, this can be done with a leading cost of $\sim O(L_x D^6 D_{\text{op}})$.

Step 4: In the complete contraction depicted in Fig. 4(f), the horizontal bond dimensions of both intops and env are χ . To perform this contraction exactly, the dominant computational cost is $\sim O(L_x D^8 D_{\text{op}}^3 \chi^2) \approx O(L_x D^{12} D_{\text{op}}^3)$. This contraction can also be performed approximately at reduced cost, although we did not do this in this work.

Step 5: Much like step 1, the leading cost of the contraction and truncation required to move intops (horizontal bond dimension χ) from row y to row $y + 1$ is $\sim O(L_x D^5 D_{\text{op}} \chi^2) \approx O(L_x D^{11} D_{\text{op}})$.

APPENDIX B: MPO AND GMPO FOR GENERAL FINITE-RANGE HAMILTONIAN

In Sec. IV A 2, we reported the exact construction of the vertical MPO matrices and the gMPO tensors for a Hamiltonian that had nonsymmetric “linear” interactions up to distance $R = 1$, and nonsymmetric diagonal interactions up to distance $R = \sqrt{2}$. Following the concepts in that example, and the general ideas behind MPO construction, in this Appendix we will give the exact construction for a Hamiltonian with nonsymmetric linear interactions up to a general distance R , and nonsymmetric diagonal interactions up to $\sqrt{2}R$. The interactions coefficients will be denoted $j_{x,y}$, where x is the horizontal distance between the local operators and y is the vertical distance.

The vertical MPO matrices are size $(R + 2) \times (R + 2)$ and they are given by

$$\hat{O}_{\beta_1}[x, 1] = (\hat{I} \quad \hat{A} \quad [\hat{O}]_{R-1}),$$

$$\hat{O}_{\beta_{y-1}\beta_y}[x, y > 1] = \left(\hat{O} \left| \begin{array}{c} \hat{A} \\ \hat{O} \\ \hat{O} \end{array} \right| \begin{array}{c} \hat{O} \\ \mathbb{1}_{(R-1) \times (R-1)} \hat{I} \\ \hat{O} \end{array} \right). \quad (\text{B1})$$

The gMPO tensors are size $(2R + 2) \times (2R + 2) \times (R + 1)$, where the third dimension is the size of the β index. They are given by

$$\begin{aligned}
\hat{M}_1[1, y > 1] &= (\hat{C}, j_{1,0}\hat{A}, \dots, j_{R,0}\hat{A}, \hat{B}, [\hat{O}]_{R-1} \hat{I}), \\
\hat{M}_1[1, y > 1] &= (\hat{I}, \hat{B}, [\hat{O}]_{2R-1} \hat{C})^T, \\
\hat{M}_1[L_x > x > 1, y > 1] &= \left(\begin{array}{c|ccc|ccc|c} \hat{I} & & & \hat{O} & & & \hat{O} & & & \hat{O} \\ \hat{B} & & & \hat{O} & & & \hat{O} & & & \hat{O} \\ \hat{O} & & \mathbb{1}_{(R-1) \times (R-1)} \hat{I} & & & & \hat{O} & & & \hat{O} \\ \hat{O} & & & \hat{O} & & & \hat{O} & & \mathbb{1}_{(R-1) \times (R-1)} \hat{I} & \hat{O} \\ \hat{O} & & & \hat{O} & & & \hat{O} & & & \hat{O} \\ \hat{C} & j_{1,0}\hat{A}, & \dots, & j_{R-1,0}\hat{A} & & j_{R,0}\hat{A} & \hat{B} & & \hat{O} & \hat{I} \end{array} \right), \\
\hat{M}_k[1, y > 1] &= (j_{0,k-1}\hat{B}, j_{1,k-1}\hat{I}, \dots, j_{R,k-1}\hat{I}, [\hat{O}]_{R+1}), \\
\hat{M}_k[L_x, y > 1] &= ([\hat{O}]_{R+1} j_{1,k-1}\hat{I}, \dots, j_{R,k-1}\hat{I}, j_{0,k-1}\hat{B})^T, \\
\hat{M}_k[L_x > x > 1, y > 1] &= \left(\begin{array}{c|cccc|c} [\hat{O}]_{R+1} & & & & \hat{O} \\ j_{1,k-1}\hat{I} & & & & \hat{O} \\ j_{2,k-1}\hat{I} & & & & \hat{O} \\ \vdots & & & & \hat{O} \\ j_{R,k-1}\hat{I} & & & & \hat{O} \\ j_{0,k-1}\hat{B} & j_{1,k-1}\hat{I}, & j_{2,k-1}\hat{I}, & \dots, & j_{R,k-1}\hat{I} & [\hat{O}]_{R+1} \end{array} \right), \\
k &\in \{2, 3, \dots, R + 1\}. \tag{B2}
\end{aligned}$$

In these expressions, $\mathbb{1}_{N \times N}$ is an $N \times N$ identity matrix. Additionally, when something is enclosed in square brackets and labeled with a subscript n , it means “repeat this n times.” Based on the dimensions of the other blocks, it should be clear which axis it should be expanded along. This is only used in places where it is not otherwise obvious to expand the blocks to match the dimensions of adjacent blocks.

-
- [1] S. R. White, *Phys. Rev. Lett.* **69**, 2863 (1992).
[2] S. R. White, *Phys. Rev. B* **48**, 10345 (1993).
[3] U. Schollwöck, *Rev. Mod. Phys.* **77**, 259 (2005).
[4] M. Fannes, B. Nachtergaele, and R. F. Werner, *Commun. Math. Phys.* **144**, 443 (1992).
[5] M. Fannes, B. Nachtergaele, and R. Werner, *J. Funct. Anal.* **120**, 511 (1994).
[6] S. Östlund and S. Rommer, *Phys. Rev. Lett.* **75**, 3537 (1995).
[7] U. Schollwöck, *Ann. Phys.* **326**, 96 (2011).
[8] F. Verstraete, J. J. Garcia-Ripoll, and J. I. Cirac, *Phys. Rev. Lett.* **93**, 207204 (2004).
[9] I. P. McCulloch, *J. Stat. Mech.* (2007) P10014.
[10] F. Verstraete, V. Murg, and J. Cirac, *Adv. Phys.* **57**, 143 (2008).
[11] B. Pirvu, V. Murg, J. I. Cirac, and F. Verstraete, *New J. Phys.* **12**, 025012 (2010).
[12] G. K.-L. Chan, A. Keselman, N. Nakatani, Z. Li, and S. R. White, *J. Chem. Phys.* **145**, 014102 (2016).
[13] ITensor Library, <http://itensor.org>.
[14] T. Nishino and K. Okunishi, *J. Phys. Soc. Jpn.* **65**, 891 (1996).
[15] F. Verstraete and J. I. Cirac, [arXiv:cond-mat/0407066](https://arxiv.org/abs/cond-mat/0407066).
[16] F. Verstraete, M. M. Wolf, D. Perez-Garcia, and J. I. Cirac, *Phys. Rev. Lett.* **96**, 220601 (2006).
[17] R. Orús, *Ann. Phys.* **349**, 117 (2014).
[18] J. Jordan, R. Orús, G. Vidal, F. Verstraete, and J. I. Cirac, *Phys. Rev. Lett.* **101**, 250602 (2008).
[19] R. Orús and G. Vidal, *Phys. Rev. B* **80**, 094403 (2009).
[20] M. Lubasch, J. I. Cirac, and M.-C. Banuls, *Phys. Rev. B* **90**, 064425 (2014).
[21] P. Corboz, *Phys. Rev. B* **94**, 035133 (2016).
[22] L. Vanderstraeten, J. Haegeman, P. Corboz, and F. Verstraete, *Phys. Rev. B* **94**, 155123 (2016).
[23] G. M. Crosswhite and D. Bacon, *Phys. Rev. A* **78**, 012356 (2008).
[24] F. Fröwis, V. Nebendahl, and W. Dür, *Phys. Rev. A* **81**, 062337 (2010).
[25] M. J. O’Rourke, Z. Li, and G. K.-L. Chan, *Phys. Rev. B* **98**, 205127 (2018).
[26] P. Czarnik and J. Dziarmaga, *Phys. Rev. B* **92**, 035152 (2015).
[27] P. Czarnik, J. Dziarmaga, and A. M. Oleś, *Phys. Rev. B* **93**, 184410 (2016).
[28] A. Kshetrimayum, H. Weimer, and R. Orús, *Nat. Commun.* **8**, 1291 (2017).
[29] A. Kshetrimayum, M. Rizzi, J. Eisert, and R. Orús, *Phys. Rev. Lett.* **122**, 070502 (2019).
[30] R. Haghshenas and D. N. Sheng, *Phys. Rev. B* **97**, 174408 (2018).
[31] M. Lubasch, J. I. Cirac, and M.-C. Banuls, *New J. Phys.* **16**, 033014 (2014).
[32] P. Corboz, T. M. Rice, and M. Troyer, *Phys. Rev. Lett.* **113**, 046402 (2014).
[33] H.-J. Liao, J.-G. Liu, L. Wang, and T. Xiang, *Phys. Rev. X* **9**, 031041 (2019).
[34] Z. Li, M. J. O’Rourke, and G. K.-L. Chan, *Phys. Rev. B* **100**, 155121 (2019).

- [35] L. Lin and Y. Tong, [arXiv:1909.02206](https://arxiv.org/abs/1909.02206).
- [36] G. M. Crosswhite, A. C. Doherty, and G. Vidal, *Phys. Rev. B* **78**, 035116 (2008).
- [37] E. M. Stoudenmire and S. R. White, *Phys. Rev. Lett.* **119**, 046401 (2017).
- [38] D. Braess and W. Hackbusch, *IMA J. Numer. Anal.* **25**, 685 (2005).
- [39] G. Beylkin and L. Monzón, *Appl. Comput. Harmonic Anal.* **19**, 17 (2005).
- [40] G. Beylkin and L. Monzón, *Appl. Comput. Harmonic Anal.* **28**, 131 (2010).
- [41] https://gitlab.com/mattorourke41/lr_gmpo_public.
- [42] M. T. Fishman, L. Vanderstraeten, V. Zauner-Stauber, J. Haegeman, and F. Verstraete, *Phys. Rev. B* **98**, 235148 (2018).
- [43] I. P. McCulloch, [arXiv:0804.2509](https://arxiv.org/abs/0804.2509).
- [44] V. Zauner-Stauber, L. Vanderstraeten, M. T. Fishman, F. Verstraete, and J. Haegeman, *Phys. Rev. B* **97**, 045145 (2018).
- [45] D. E. Parker, X. Cao, and M. P. Zaletel, [arXiv:1909.06341](https://arxiv.org/abs/1909.06341).