Tensor Networks, Matrix Product States and Density Matrix Renormalization Group

Fred Zhang*

December 18, 2018

Abstract

In this note, we introduce the notions of tensor networks and matrix product states (MPS). These objects are particularly useful in describing quantum states of low entanglement.

We then discuss how to efficiently compute the ground states of the Hamiltonians of 1D quantum systems (using classical computers). The density matrix renormalization group (DMRG), due to White (1992, 1993) [Whi92, Whi93], is arguably the most successful heuristic for this problem. We describe it in the language of tensor networks and MPS.

1 Introduction

Recall that in the previous note, we considered the problem of computing the ground state energy a local Hamiltonian. In particular, we discussed some hardness results. Namely, it can be used to encode SAT instances, and we further gave a proof that it is QMA-Complete.

Despite the hardness results, physicists have come up with a variety of heuristics for solving this problem. A typical observation here is that quantum interactions happen locally in many systems. In these cases, we would hope that its ground state has low entanglement and thus admit a succinct classical representation. Algorithmically, this leads to the idea that perhaps we can find such a representation efficiently, using classical computers.

In this note, we will see *tensor networks* and *matrix product states* that formalize the idea of succinctly representing quantum states of low entanglement. As a side remark for the theoretical computer scientists here, one motivation to study tensor network is that it provides a powerful visual tool for thinking about linear algebra. It essentially turns indices into edges in a graph and summations over indices into contractions of edges. In particular, we will soon see that the most useful inequality in TCS and mathematics can be drawn as a cute tensor network.

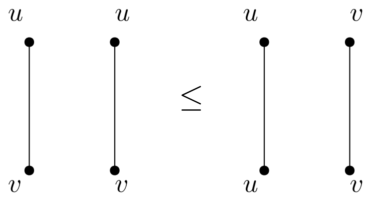


Figure 1: Guess what this is?

In the end, we will discuss the density matrix renormalization group (DMRG), which has established itself as “the most powerful tool for treating 1D quantum systems” over the last decade [FSW07]. For many 1D systems that arise from practice, the heuristic efficiently finds an (approximate) ground state in its matrix product state, specified only by a small number of parameters.

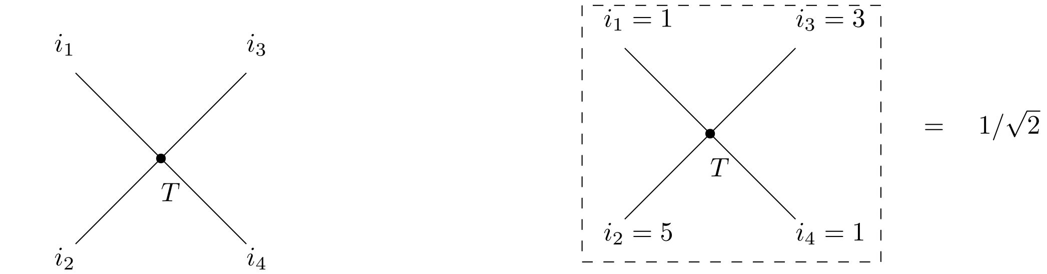
*Harvard University. Email: hzhang@g.harvard.edu

2 Tensor Networks

Now let us discuss our first subject, *tensor networks.* If you have not seen *tensors* before, it is a generalization of matrices. In computer scientists' language, a matrix is a two-dimensional array, and a tensor is a multi-dimensional array. In other words, if we think of a matrix as a square, then a 3 dimensional tensor looks a cube. Formally, a (complex) n dimensional tensor T maps n indices to complex values, namely, to its entries:

$$T : [d_1] \times [d_2] \times \cdots \times [d_n] \to \mathbb{C}.$$

The simplest tensor network is a graphical notation for a tensor. For an n-dimensional tensor T, we draw a star graph and label the center as T and the edges as the indices. To evaluate this tensor network, we put values on the edges, *i.e.*, indices, and then the tensor network would spit out its entry specified by the indices.

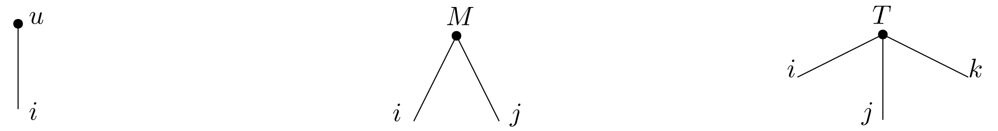


(a) A simple tensor network.

(b) Evaluating a simple tensor network, $T(1, 5, 3, 1) = 1/\sqrt{2}$. The numbers are chosen arbitrarily.

Figure 2: The definition of the simplest tensor network.

Notice that the degree of the center is the number of indices. Hence, a tensor network of degree 1 is a vector, and that of degree 2 is a matrix, and so forth.



(a) A vector u

(b) A matrix M

(c) A 3 dimensional tensor T

Figure 3: Simple tensor networks that represent basic linear algebraic objects.

How is this related to quantum information? For the sake of genearlity we will deal with qudits in $\mathbb{C}^d$, instead of qubits in . Now recall that a quantum state $|\psi_n\rangle$ of n qudits can be encoded as an n dimensional tensor. It can be written as

$$|\psi_n\rangle = \sum_{i_1,\cdots,i_n=0}^{d-1} T(i_1, \cdots, i_n) |i_1, \cdots, i_n\rangle.$$

It is easy to see that all the information, namely, the amplitudes, is just the tensor T. In the later sections, we will see more powerful examples of using tensor networks to represent a quantum state.

So far our discussion is focused merely on these little pictures. The power of tensor networks come from its composition rules, which allow us to join two simple tensor networks together and impose rich internal structures.

2.1 Composition Rules

We introduce two ways of joining two simple tensor networks. Roughly speaking, they correspond to multiplication and summation, and I will give the definitions by showing examples, instead of stating them in the full formalism.

Rule #1: Tensor Product. The product rule allows us to put two tensor networks together and view them as a whole. The resulting tensor is the tensor product of the two if we think of them as vectors. More concretely, consider the following picture.

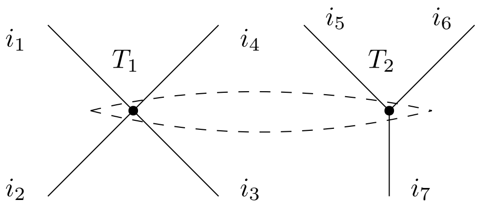
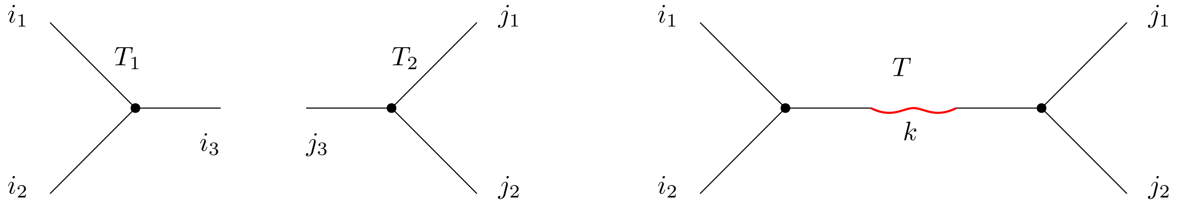


Figure 4: This is viewed as a single tensor network T of 7 edges.

The definition of this joint tensor T is

$$T(i_1, i_2, \cdots, i_7) = T_1(i_1, i_2, i_3, i_4) \cdot T_2(i_5, i_6, i_7).$$

Rule #2: Edge Contractions. At this moment, we can only make up disconnected tensor networks. Edge contractions allow us to link two tensor networks. Suppose we have two 3 dimensional tensor networks. Contracting two edges, one from each, gives us a tensor network of 4 *free edges*. This now corresponds a tensor of 4 dimensions.



(a) Two 3 dimensional tensors.

(b) Contracting i_3 and j_3 together gives a tensor T

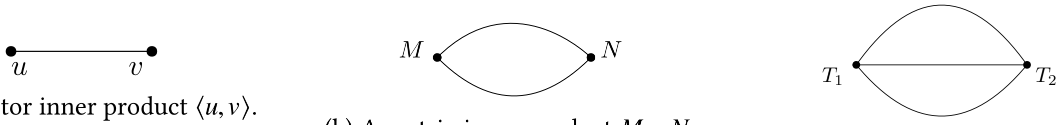
Figure 5: Edge contractions.

We name the contracted edge as k. The definition of T is

$$T(i_1, i_2, j_1, j_2) = \sum_k T_1(i_1, i_2, k)\, T_2(j_1, j_2, k).$$

2.2 Useful Examples

Before we move on, let's take some examples. Keep in mind that the degree of the vertex determines the number of indices (dimensions of this tensor).



(a) A vector inner product $\langle u, v\rangle$.

(b) A matrix inner product $M \circ N$.

(c) A tensor inner product $T_1 \circ T_2$.

Figure 6: Inner products are given by joining all the free edges. Edge labels are omitted.

Here, one simply needs to remember that an edge between two tensor nodes is a summation over the index corresponding to the edge. For example, in Figure 6a, $\langle u, v\rangle = \sum_i u_i \cdot v_i$, where edge is labeled as i. Now you would realize that Figure 1 is the famous

$$\langle u, v\rangle^2 \leq \|u\|^2\|v\|^2. \qquad \text{(Cauchy-Schwarz inequality)}$$

For us, the most important building block is matrix multiplication. Let $H = MN$. Then the definition of matrix multiplication tells us

$$H(i,j) = \sum_k M(i,k)N(k,j).$$

This is precisely encoded in the Figure 7.

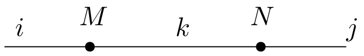


Figure 7: Matrix multiplication, MN.

We are ready to talk about matrix product states. In the language of tensor network, a matrix product state is the following picture.

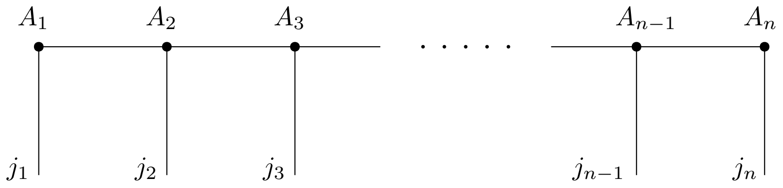


Figure 8: A matrix product state.

As the degrees indicate, the two boundary vertices A_1, A_n represent matrices and the internal vertices represent 3-dimensional tensors. We can view each matrix as a set of (column) vectors and each 3-dimensional tensor as a stack of matrices. Then each one of the free edges picks out a vector or a matrix, and the contracted edges multiply them together which gives out a scalar. If this confused you, move on to the next section. I will introduce the formal definition of matrix product states, and you will see that it is just the picture above.

3 Matrix Product States

Before giving the definition, let's talk about how matrix product state (MPS) naturally arises from the study of quantum states with low entanglement. Matrix product state can be viewed as a generalization of *product state*—(pure) quantum state with no entanglement. Let's consider a simple product state $|\psi\rangle$ of 2 qubits. It can be factorized:

$$.|\psi\rangle = \left(\sum_{i=0}^{1} \alpha_1^i \, |i\rangle\right)\left(\sum_{j=0}^{1} \alpha_2^j \, |j\rangle\right) = \sum_{i,j=0}^{1} \alpha_1^i \alpha_2^j \, |ij\rangle$$

This state is described by 4 complex scalars $\left\{\alpha_1^0, \alpha_1^1, \alpha_2^0, \alpha_2^1\right\}$, and there is nothing quantum about it. However, if the state has entanglement among its qubits, then we know that it is impossible to be factorized and thereby written as (1). MPS generalizes the form of (1) by replacing the scalars with matrices and vectors.

More formally, a matrix product state starts with the following setup. For an n-qudit system, we associate

- a qudit in $\{1, n\}$ with d vectors $\left\{A_1^{j_1}\right\}, \left\{A_n^{j_n}\right\} \in \mathbb{R}^D$; and
- a qudit i in $\{2, 3, \cdots, n-1\}$ with d matrices $\left\{A_i^{j_i}\right\} \in \mathbb{R}^{D \times D}$.

Here, j_i range from 0 to $d-1$, and D is called *bond dimension.* One can think of the set of vectors as a D by d matrix and the set of matrices as a d by D by D three-dimensional tensor. Then let them correspond to the vertices in Figure 8. With this setup, a quantum state is in matrix product state if it can be written as

$$|\psi\rangle = \sum_{j_1, \cdots, j_n = 1}^{n} A_1^{j_1} A_2^{j_2} \cdots A_n^{j_n} \, |j_1 j_2 \cdots j_n\rangle.$$

It is important to keep in mind that $A_1^{j_1}, A_n^{j_n}$ are two vectors, and the other inner terms are matrices, and we get a scalar from the product. Thus, this represents the tensor $T(j_1, j_2, \cdots, j_n) = A_1^{j_1} A_2^{j_2} \cdots A_n^{j_n}$. Now back to Figure 8, notice that each amplitude $A_1^{j_1} A_2^{j_2} \cdots A_n^{j_n}$ from the equation above is an output of the tensor in the picture, where the free edges take values $j_1, j_2, \cdots, j_n$. Also, as discussed earlier, the contracted edges in Figure 8 correspond to matrix and vector multiplications, so the tensor T is precisely represented Figure 8.

The complexity of the MPS is closely related to the bond dimension D. In particular, the number of parameters in this model is $O(ndD^2)$. We would expect that with higher D, we may describe quantum states of more entanglement. In other words, the representation power of an MPS increases with D. In principle, one can represent any quantum state as an MPS; however, D can be exponentially large. See, *e.g.*, Section 4.1.3 of [Sch11] for a proof. On the other extreme, the product state example shows that if $D = 1$, one can represent and *only* represent unentangled states. To summarize, here is the picture you should keep in mind.

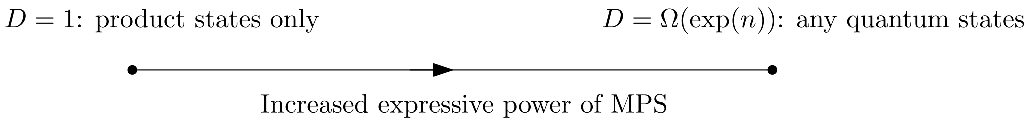

$D = 1$: product states only
$D = \Omega(\exp(n))$: any quantum states
Increased expressive power of MPS

Figure 9: Representation power of MPS increases with bond dimension D.

4 Density Matrix Renormalization Group

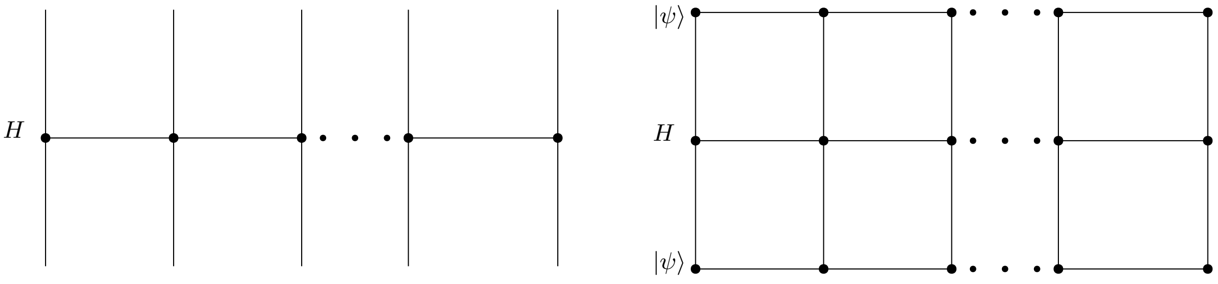
We are now ready to describe Density Matrix Renormalization Group, proposed originally in [Whi92, Whi93]. As mentioned earlier, it does not come with provable guarantees. In fact, one can construct artificial hard instances such that the algorithm get stuck at certain local minima [SCV08]. However, it has remained one of the most successful heuristics for 1D systems. We refer the readers to [Sch11] for a complete survey.

DMRG is a simple alternating minimization scheme for computing the ground state of a 1D Hamiltonian. We start with an arbitrary MPS. Then each step we optimize over the set of matrices $\left\{A_i^{j_i}\right\}_{j_i=0}^{d}$ associated with site i, while fixing everything else.[1], and iterate until convergence.

Formally, the Hamiltonian problem can be phrased as a eigenvalue problem given a Hermitian matrix H, and thus we want to optimize over all $|\psi\rangle$ in MPS of a fixed bond dimension D

$$\min_{|\psi\rangle} \frac{\langle\psi|H|\psi\rangle}{\langle\psi||\psi\rangle}.$$

Here, we assume that the input Hamiltonian is in the product form. In particular, it means that it can be written as a tensor network as Figure 10a so that the numerator of the optimization objective looks like Figure 10b.



(a) Input 1D Hamiltonian is of the particular product form.

(b) The quadratic form $\langle\psi|H|\psi\rangle$

Figure 10: The input and the numerator of the optimization objective

The DMRG works with the Langrangian of the objective (4). For some $\lambda > 0$, we will consider

$$\min_{|\psi\rangle} \langle\psi|H|\psi\rangle - \lambda\langle\psi||\psi\rangle. \tag{1}$$

[1] You may wonder if one can simultaneously optimize over multiple sites. It turns out that it is an NP-hard problem [Eis06].

DMRG optimizes over the set of matrices associated with one qudit. Both terms in of 1 are quadratic in this set of matrices, as one can see from the Figure 11.

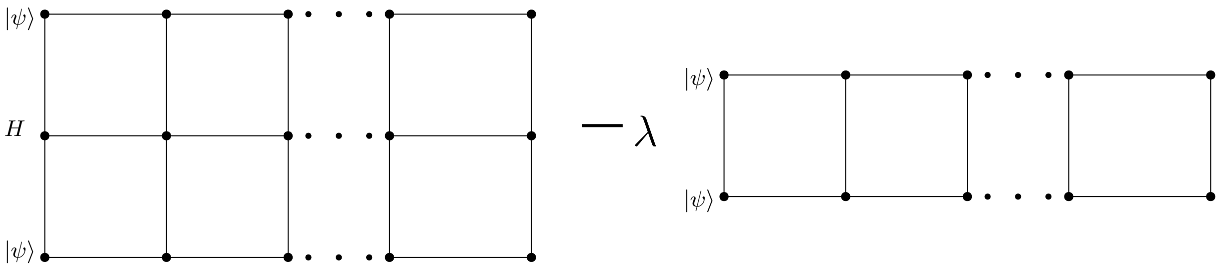


Figure 11: The Langrangian $\langle\psi|H|\psi\rangle - \lambda\langle\psi||\psi\rangle$ as a tensor network.

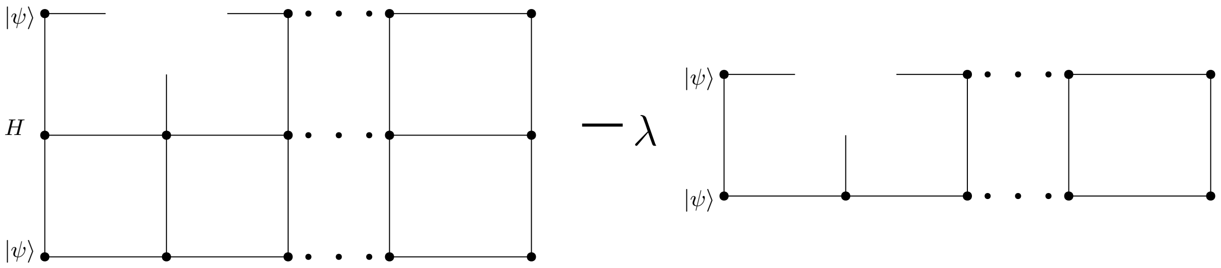


Figure 12: The derivative that we set to 0 and solve.

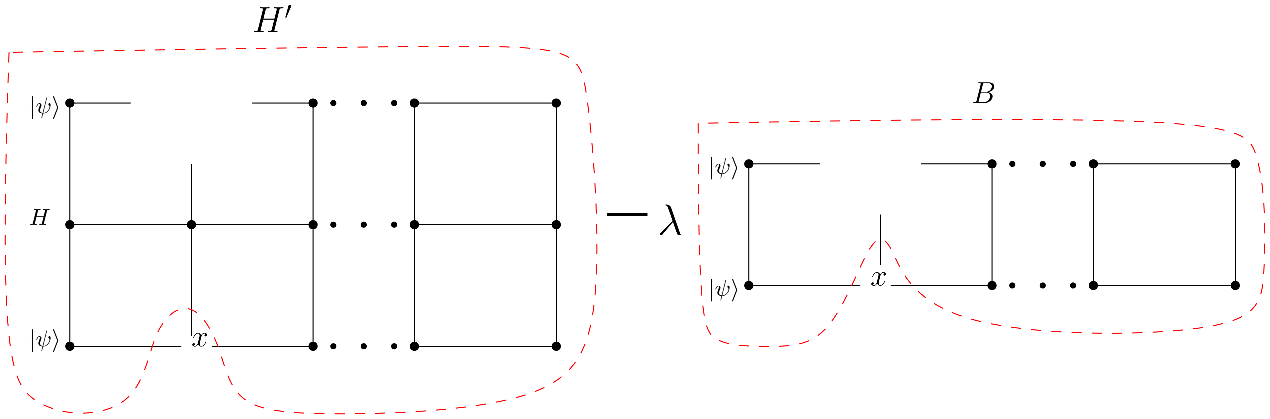


Figure 13: The unknown is named x and the rest of the network is denoted by H' and B for each term.

Now to optimize over the set of parameters associated with one site, calculus tells you (1) to set the (partial) derivative to 0 and (2) the derivative of a quadtric thing is linear. Without going through any algebra, we can guess that the derivative of Figure 11 with respect to a particular site, say the second one, is the same picture except removing the second site on one side—Figure 12. Notice that the unknown is still there, on the bottom side of each term. The trick of DMRG is to view the rest of the network as a linear map applied to the unknown—Figure 13.

Given H' and B, we now have a clean numerical linear algebra problem of solving

$$H'x = \lambda Bx. \tag{2}$$

This is called a generalized eigvenvalue problem, and it is well studied. Importantly, for 1D systems, H' is typically very sparse, which enables very fast solvers in practice. Finally, DMRG sweeps over the sites one after another and stops until convergence is achieved.

5 Concluding Remarks

Our presentation of tensor networks and MPS roughly follows [GHLS15], a nice introductory survey on quantum Hamiltonian complexity.

The notion of tensor network extends well beyond 1D systems, and a generalization of MPS is called tensor product state. It leads to algorithms for higher dimensional quantum systems. One may read [CV09] for a comprehensive survey.

Tensor network has been interacting with other concepts. Within physics, it has been used in quantum error correction [FP14, PYHP15], conformal field theory [Orú14], and statistical mechanics [EV15]. In TCS , we have found its connections with Holographic algorithms [Val08, CGW16], arithmetic complexity [BH07, CDM16, AKK19], and spectral algorithms [MW18]. In machine learning, it has been applied to probabilistic graphical models [RS18], tensor decomposition [CLO$^+$16], and quantum machine learning [HPM$^+$18].

For DMRG, we have only given a rough outline, with many details omitted, such as how to set D and λ and how to obtain the Hamiltonian in the matrix product form, and how to compute the linear maps H' and B for each iteration. An interested reader may read [Sch05, Sch11].

Acknowledgement

I would like to thank Boaz Barak and Tselil Schramm for helpful comments.

References

[AKK19] Per Austrin, Petteri Kaski, and Kaie Kubjas. Tensor network complexity of multilinear maps. In *Proceedings of the 2019 Conference on Innovations in Theoretical Computer Science*. ACM, 2019. 7

[BH07] Martin Beaudry and Markus Holzer. The complexity of tensor circuit evaluation. *Computational Complexity*, 16(1):60–111, 2007. 7

[CDM16] Florent Capelli, Arnaud Durand, and Stefan Mengel. The arithmetic complexity of tensor contraction. *Theory of Computing Systems*, 58(4):506–527, 2016. 7

[CGW16] Jin-Yi Cai, Heng Guo, and Tyson Williams. A complete dichotomy rises from the capture of vanishing signatures. *SIAM Journal on Computing*, 45(5):1671–1728, 2016. 7

[CLO$^+$16] Andrzej Cichocki, Namgil Lee, Ivan V Oseledets, A-H Phan, Qibin Zhao, and D Mandic. Low-rank tensor networks for dimensionality reduction and large-scale optimization problems: Perspectives and challenges part 1. *arXiv preprint arXiv:1609.00893*, 2016. 7

[CV09] J Ignacio Cirac and Frank Verstraete. Renormalization and tensor product states in spin chains and lattices. *Journal of Physics A: Mathematical and Theoretical*, 42(50):504004, 2009. 7

[Eis06] Jens Eisert. Computational difficulty of global variations in the density matrix renormalization group. *Phys. Rev. Lett.*, 97:260501, Dec 2006. 5

[EV15] G. Evenbly and G. Vidal. Tensor network renormalization. *Phys. Rev. Lett.*, 115:180405, Oct 2015. 7

[FP14] Andrew J Ferris and David Poulin. Tensor networks and quantum error correction. *Phys. Rev. Lett.*, 113(3):030501, 2014. 7

[FSW07] Holger Fehske, Ralf Schneider, and Alexander Weiße. *Computational Many-Particle Physics*. Springer, 2007. 1

[GHLS15] Sevag Gharibian, Yichen Huang, Zeph Landau, and Seung Woo Shin. Quantum Hamiltonian complexity. *Foundations and Trends in Theoretical Computer Science*, 10(3):159–282, 2015. 7

[HPM+18] William James Huggins, Piyush Patil, Bradley Mitchell, K Birgitta Whaley, and Miles Stoudenmire. Towards quantum machine learning with tensor networks. *Quantum Science and Technology*, 2018. 7

[MW18] Ankur Moitra and Alexander S Wein. Spectral methods from tensor networks. *arXiv preprint arXiv:1811.00944*, 2018. 7

[Orú14] Román Orús. Advances on tensor network theory: symmetries, fermions, entanglement, and holography. *The European Physical Journal B*, 87(11):280, 2014. 7

[PYHP15] Fernando Pastawski, Beni Yoshida, Daniel Harlow, and John Preskill. Holographic quantum error-correcting codes: Toy models for the bulk/boundary correspondence. *Journal of High Energy Physics*, 2015(6):149, 2015. 7

[RS18] Elina Robeva and Anna Seigal. Duality of graphical models and tensor networks. *Information and Inference: A Journal of the IMA*, page iay009, 2018. 7

[Sch05] Ulrich Schollwöck. The density-matrix renormalization group. *Rev. Mod. Phys.*, 77(1):259, 2005. 7

[Sch11] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, 2011. 5, 7

[SCV08] Norbert Schuch, Ignacio Cirac, and Frank Verstraete. Computational difficulty of finding matrix product ground states. *Phys. Rev. Lett.*, 100(25):250501, 2008. 5

[Val08] Leslie G Valiant. Holographic algorithms. *SIAM Journal on Computing*, 37(5):1565–1594, 2008. 7

[Whi92] Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69:2863–2866, Nov 1992. 1, 5

[Whi93] Steven R. White. Density-matrix algorithms for quantum renormalization groups. *Phys. Rev. B*, 48:10345–10356, Oct 1993. 1, 5