Editors' Suggestion

Equivalence of restricted Boltzmann machines and tensor network states

Jing Chen,[1,2,4] Song Cheng,[1,2] Haidong Xie,[1,2] Lei Wang,[1,*] and Tao Xiang[1,3,†]
[1]*Institute of Physics, Chinese Academy of Sciences, P.O. Box 603, Beijing 100190, China*
[2]*School of Physical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China*
[3]*Collaborative Innovation Center of Quantum Matter, Beijing 100190, China*
[4]*Center for Computational Quantum Physics, Flatiron Institute, New York 10010, USA*

(Received 25 January 2017; revised manuscript received 15 December 2017; published 2 February 2018)

The restricted Boltzmann machine (RBM) is one of the fundamental building blocks of deep learning. RBM finds wide applications in dimensional reduction, feature extraction, and recommender systems via modeling the probability distributions of a variety of input data including natural images, speech signals, and customer ratings, etc. We build a bridge between RBM and tensor network states (TNS) widely used in quantum many-body physics research. We devise efficient algorithms to translate an RBM into the commonly used TNS. Conversely, we give sufficient and necessary conditions to determine whether a TNS can be transformed into an RBM of given architectures. Revealing these general and constructive connections can cross fertilize both deep learning and quantum many-body physics. Notably, by exploiting the entanglement entropy bound of TNS, we can rigorously quantify the expressive power of RBM on complex data sets. Insights into TNS and its entanglement capacity can guide the design of more powerful deep learning architectures. On the other hand, RBM can represent quantum many-body states with fewer parameters compared to TNS, which may allow more efficient classical simulations.

DOI: 10.1103/PhysRevB.97.085104

I. INTRODUCTION

Deep learning is transforming the world with its far-reaching applications in computer vision, speech recognition, natural language processing, recommender systems, etc. [1,2]. At the core of many of these applications are the artificial neural networks which recognize or even discover interesting patterns in the input data [3–5]. In a nutshell, the neural nets act as trainable functional mappings of multiple variables. To design even more powerful and intelligent machines requires one to quantify and extend the expressive power of the neural nets. However, there is a gap between the mathematical foundation and the real-world applications which are largely driven by the engineering practices [2], because it has long been a difficult endeavor to rigorously quantify the expressive power of neural nets over complicated data sets.

Insights into the physical rules governing the neural networks and typical data sets may offer an answer to the great success of deep learning and guide its more fruitful development in the future. For example, statistical physics has a long-standing impact on machine learning [4,6] because both fields concern about collective behavior emerged from a large amount of microscopic degree of freedoms. Moreover, as suggested by Mehta and Schwab [7], there is a connection between the deep learning and the renormalization group. Lin *et al.* [8] also argued that the "unreasonable success" of deep learning can be traced back to the law of physics, which often imposes symmetry, locality, compositionality, polynomial log-probability, and other properties on the input data.

The restricted Boltzmann machine (RBM) is a vivid example of the intrinsic connection between statistical physics and machine learning. An RBM is a special type of neural network which can be better understood as an energy-based model. As illustrated in Fig. 1(a), it consists of a set of interconnected visible and hidden binary variables. These variables are assumed to satisfy the Boltzmann distribution whose energy functional is defined by

$$E(v,h) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_{i,j} v_i W_{ij} h_j, \qquad (1)$$

where $v = \{v_i\}$ and $h = \{h_j\}$ are the visible and hidden binary variables. We denote their number as n_v and n_h, respectively. Parameters a_i, b_j are the biases applied to the visible and hidden units, respectively. W_{ij} is the coupling matrix between these two units.

By integrating out the hidden units, the RBM represents the marginal distribution of the visible variables (omitting an irrelevant normalization factor)

$$\Psi_{\mathrm{RBM}}(v) = \sum_h e^{-E(v,h)} = \prod_i e^{a_i v_i} \prod_j (1 + e^{b_j + \sum_i v_i W_{ij}}). \qquad (2)$$

In the RBM there is no direct connection between the visible units. However, the hidden units generate effective connections or interactions among them. By increasing the number of hidden units and connections, the RBM can in principle parametrize more complex functions of the visible units [9–11].

*wanglei@iphy.ac.cn
†txiang@iphy.ac.cn

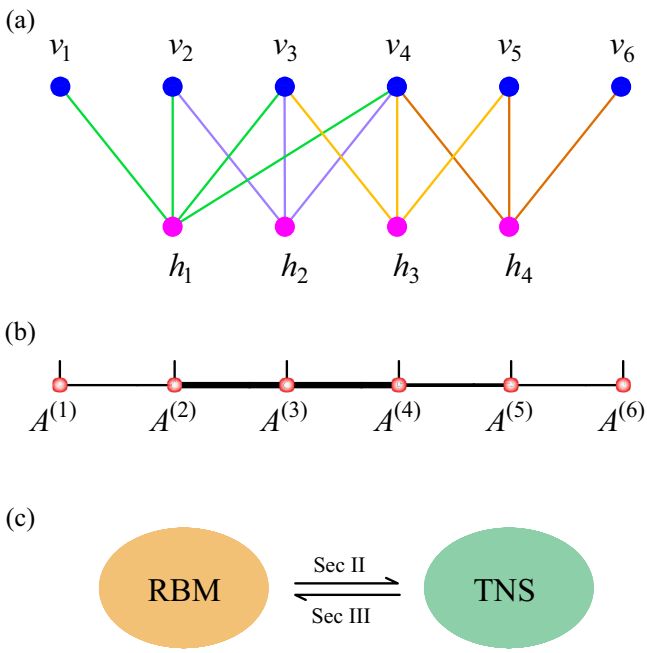

FIG. 1. (a) Graphical representation of the RBM defined by Eq. (2). The blue dots represent visible units v and the magenta dots represent the hidden units h. They are coupled through the connections indicated by the solid lines. (b) MPS defined by Eq. (3). Each red dot denotes a three-index tensor $A^{(i)}$. Throughout this paper, we use dots to represent units of RBM, and ball to represent tensors. Lines in the RBM denote connection weights while lines in the tensor network denote tensor indices. (c) RBM and TNS are two ways to parametrize multivariable functions. With unlimited resources (number of hidden units or bond dimensions) both of them can represent any function to arbitrary accuracy. However, with limited resources they represent two independent sets with overlapping region. Detailed discussions on their relationship are given in Secs. II and III.

One can train an RBM by specifying its parameters such that the probability distribution of the visible units reproduces that of the input data [12,13]. The hidden units of a trained RBM may also reveal correlations in the data with appealing physical meanings. For example, in an RBM trained with a data set of images containing handwritten digits, the connection weight contains the information about the pen stokes [14]. These learning features can be used either for discriminative tasks, such as pattern recognition, or for generative tasks, such as generating more samples according to the learned distribution. RBM has played an important role in the recent renaissance of deep learning [15,16] because of its versatile abilities in feature extraction and dimensionality reduction of complex data sets.

Recently, RBM has attracted great attention in the field of quantum many-body physics. Carleo and Troyer [17] proposed an RBM inspired variational wave function to study quantum many-body systems at or away from equilibrium. Deng *et al.* [18] constructed exact RBM representations for several interesting topological states. Torlai and Melko [19] trained an RBM to reproduce the thermodynamics of a statistical physics model. Huang and Wang [20] used RBM as a recommender system to accelerate Monte Carlo simulation of quantum many-body systems. Liu *et al.* [21] reported similar ideas using classical spin models instead of the RBM.

These developments raise several critical questions about the expressive power of neural nets in the physics contexts. Is RBM more expressive than the standard variational wave functions of quantum states [17]? Can RBM efficiently describe the probability distribution of physical models at criticality [19,20]? Unfortunately, the existing universal approximation theorem [9–11] and its further developments [22–24] are not particularly instructive for practical purpose because they involve exponentially large resources, and it cannot be used as a guiding principle to solve practical physical or industrial problems.

In fact, the quest for more expressive wave function is central to quantum many-body physics. An ideal parametrization of wave function should accurately describe a quantum state with exponentially large degree of freedoms with polynomial resources. Tensor network states (TNS) [25] are promising candidates to meet this demand. Figure 1(b) shows one of the simplest TNS, the matrix product state (MPS) [26], as an example. The MPS parametrizes a wave function of n_v physical variables as

$$\Psi_{\mathrm{MPS}}(v) = \mathrm{Tr} \prod_i A^{(i)}[v_i], \tag{3}$$

where $A^{(i)}$ is a three-index tensor represented by a red dot in Fig. 1(b). For a given value of v_i, which is represented by a dangling vertical bond, $A^{(i)}[v_i]$ is a matrix. The dimension of this matrix is commonly called the virtual bond dimension of the MPS, indicated by the thickness of the horizontal bonds in Fig. 1(b). Connecting these horizontal bonds is to take tensor contraction over all the virtual degree of freedoms. By increasing the bond dimension, MPS can represent with increasing accuracy any complex multivariable functions [26].

MPS representation is equivalent to the tensor train decompositions in the applied math community [27]. Similarly, one can connect higher-order tensors to represent a physical state in a two-dimensional network. This kind of TNS is named projected entangled pair state (PEPS) [28]. A generalization of PEPS to include the entanglement of all particles in a larger unit cell is call projected entangled simplex state (PESS) [29]. In the past decades, solid theoretical understanding and efficient numerical techniques for TNS have been established. See [25,30] for pedagogical reviews on TNS. Moreover, the application of TNS to classical systems also has a long history (see [31–33] for example).

The physics community also has an answer to the "unreasonable effectiveness" of TNS. It relies on the entanglement area law [34], which states that the entanglement entropy scales just linearly with the boundary size separating any two subsystems. The entanglement entropy [35] is a measure of the information content between these two subsystems. Many physical states of practical interests fulfill this area law [34]. It indicates that the degrees of freedom needed to describe a quantum state of physical interest is generally much less than the total degrees of freedom of the whole system. TNS are designed to efficiently represent these quantum states with relatively low entanglement entropy and have achieved remarkable successes in the past decades [36].

RBM and TNS share some similarities in their mathematical structures, especially expressed using the graphical language in Figs. 1(a) and 1(b). As for machine learning, Refs. [2,8]

also suggest that only a tiny fraction of the input data are of practical interests among infinite number of possible inputs. This motivates us to search for a guiding principle from the perspective of quantum information to quantify the expressive power of neural nets used for deep learning [1,2] as well as for quantum and statistical physics problems [17–20].

In this paper, we present a general and constructive connection between RBM and TNS. With this correspondence, many concepts and techniques from deep learning and quantum physics can be exchanged. By transforming an RBM to a TNS and exploiting its entanglement entropy bound, we can quantify the expressibility of RBM for quantum states, for statistical physics models, and for industrial data sets. We also find the necessary and sufficient conditions for transforming a TNS into an RBM with a given structure, and show that RBM can serve as an efficient representation of quantum states.

This paper is arranged as follows. In Sec. II, we present the algorithms to transform an RBM into an MPS or other kind of TNS and discuss their consequences. In Sec. III, we present the sufficient and necessary conditions for a TNS to have an RBM representation with a given architecture. In Sec. IV, we illustrate the intimate connection between RBM and TNS by constructing the fourfold-degenerate ground-state wave functions for the toric code model in both the RBM and TNS representations. In Sec. V, we discuss several applications of the established connection between RBM and TNS for physical and machine learning problems. In Sec. VI, we discuss further implications of our results in a broader context of interdisciplinary research. In Appendix A, we present a sufficient condition to find the RBM representations for some specific TNS and illustrate with statistical Ising model and simple quantum states. Finally, in Appendix B, we discuss the equivalence between more general Boltzmann machines [37,38] and TNS.

II. TNS REPRESENTATION OF RBM

In this section, we discuss the relationship between RBM and TNS via a constructive approach. An important application of the TNS representation for RBM is to provide an upper bound of the entanglement entropy it can capture. To estimate the bound, one only needs structural information but not the detailed parametrization of the RBMs. We first present a simple and intuitive approach, then discuss more sophisticated approaches that can provide tighter or optimal bounds on the tensor bond dimensions. We provide code implementations of the mapping in [39].

Before proceeding, we first clarify a few notations about RBM. First, in the standard machine learning applications and in Refs. [19,20], parameters $\{W_{ij},a_i,b_j\}$ are assumed to be real and the corresponding RBM represents the probability distributions of input data. However, in Refs. [17,18], Eq. (2) is interpreted as the amplitude of quantum mechanical wave function, and $\{W_{ij},a_i,b_j\}$ are generalized to the complex domain. In this paper, we adopt the convention of Refs. [17,18], and assume these parameters to be complex.[1] Second, conventionally one views the RBM as an energy-based model. While for our discussion about the expressibility of Eq. (2), it is sufficient to view it as a function approximator such as a feed-forward neural net [20,40]. Third, although the standard RBM may have dense all-to-all connections between the visible and hidden units, for clarity we will illustrate the transformation using the RBM with sparse connections. Our result nevertheless holds generally and can be applied to RBMs with arbitrarily dense connections.

A. Direct mapping of the RBM to MPS

To start with, let us consider a simple way to map an RBM to an MPS. We summarize the procedure in Algorithm 1. As a concrete example, we consider the RBM defined in Fig. 1(a). The first step is to convert this RBM into a TNS by representing the visible and hidden units as the physical and virtual variables while keeping the network structure unchanged. To do this, we decouple the Boltzmann weights into the terms defined on the vertices and bonds separately by introducing a diagonal tensor, $\Lambda_v^{(i)}$ or $\Lambda_h^{(j)}$, at each visible or hidden site, and a 2×2 matrix $M^{(ij)}$ on each bond linking v_i and h_j:

$$\Lambda_v^{(i)} = \mathrm{diag}(1,e^{a_i}), \tag{4}$$

$$\Lambda_h^{(j)} = \mathrm{diag}(1,e^{b_j}), \tag{5}$$

$$M^{(ij)} = \begin{pmatrix} 1 & 1 \\ 1 & e^{W_{ij}} \end{pmatrix}. \tag{6}$$

This leads to the TNS representation of the RBM shown in Fig. 2(a).

The next step is to map the TNS in Fig. 2(a) to an MPS. We first cut the graphics into n_v pieces, each containing a visible unit [Fig. 2(b)]. The assignment of the hidden units into these pieces can be arbitrary. Contracting all the internal variables within each piece is then equivalent to summing over the hidden units. The MPS, as shown in Fig. 2(c), is obtained by merging all the external connections between different pieces into the virtual bonds. The bond dimension of the MPS is indicated by the thickness of the virtual bonds, which is determined by the number of connections merged.

Here, we should pay more attention to the "long-range" connections that cross two or more vertical cuts. For the RBM shown in Fig. 2(a), the bond that connects v_4 and h_1 is the only long-range connection. In order to map it into the virtual bonds of MPS, we decouple the matrix defined on this connection $M^{(41)}$ into a product of two arbitrary 2×2 matrices, P and Q, subject to the constraint $M^{(41)} = P\cdot Q$ (Fig. 3). This separates effectively the long-range connection into two short ones, whose matrices are defined by P and Q, respectively. These two matrices are then absorbed into the local tensors at v_2 and v_3, i.e., $A^{(2)}$ and $A^{(3)}$, respectively. This long-range connection crosses the vertical bond at v_3, and consequently doubles the bond dimension of $A^{(3)}$. In general, a long-range connection will double the bond dimensions of all tensors it passes by. The dimension of MPS at a particular bond D is determined by the number of connections n one has to cut in order to bipartite the system at that bond, i.e., $D = 2^n$.

[1]References [17,18] used RBM units with ± 1 instead of binary values. This amounts to a simple rescaling and offset of the weights and biases.

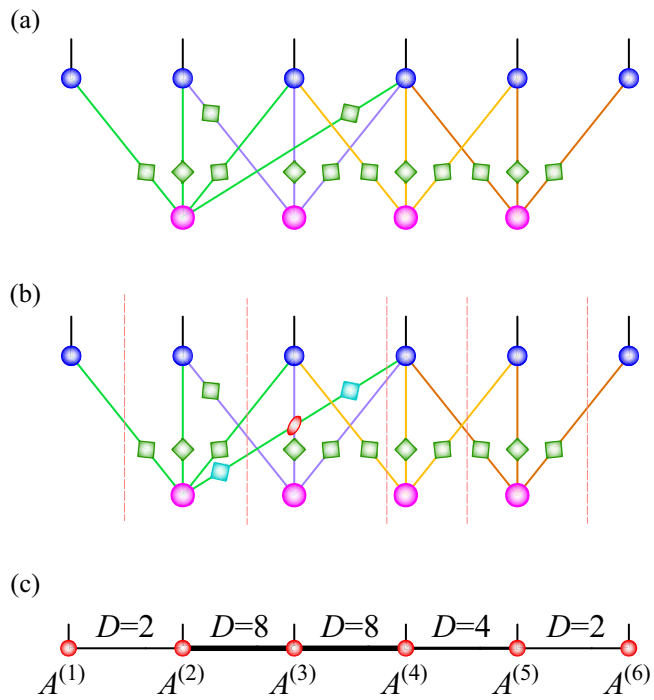

FIG. 2. Steps to map an RBM to an MPS. (a) A TNS representation of the RBM shown in Fig. 1(a). The blue dots represent the diagonal tensors $\Lambda_v^{(i)}$ at the visible units and the magenta dots represent $\Lambda_h^{(j)}$ at the hidden units, defined by Eqs. (4) and (5), respectively. The green squares represent matrix $M^{(ij)}$, defined in Eq. (6). (b) The RBM is cut into n_v (here $n_v = 6$) pieces. Associated with each long-range connection, an identity tensor (red oval) is introduced to break $M^{(ij)}$ into two matrices (see Fig. 3). (c) An MPS representation of the RBM obtained by contracting all hidden units and connection bonds in (b). The bond dimension of the MPS is determined by the number of bonds cut by the corresponding dashed vertical line.

It should also be noted that the MPS obtained from the above mapping process is not unique because the geometrical structure of the hidden units with respect to the visible units can be arbitrarily arranged. No matter how the hidden units are arranged, these different MPS are equivalent. The local tensors obtained with the above approach generally also contain redundant degrees of freedom. They can be gauged into a unique canonical form [41,42] by taking canonical transformations for all local tensors. See Sec. V A for more discussions on the redundancy of TNS and RBM representations.

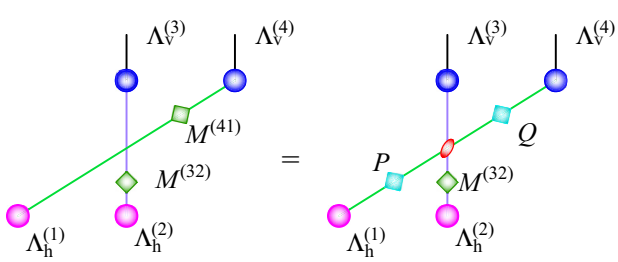

FIG. 3. Matrix $M^{(41)}$, represented by the green square, is decoupled into a product of two matrices, P and Q, denoted by the two cyan squares. The red oval represents direct product of two identity matrices along two crossing directions.

Algorithm 1 Direct mapping an RBM into an MPS.

Input: The connection weight matrix W_{ij} and biases a_i, b_j. The matrix specifies the structure of the RBM.
Output: An MPS with each tensor $A^{(i)}$, $i = 1,2,\ldots,n_v$.

```
The RBM is cut into n_v pieces P_i, i = 1,...,n_v. Each P_i includes a visible unit v_i and several hidden units.
for i = 1,...,n_v do
    Set T_i = ∅.                                  ▷ Contains the tensors in P_i
    Construct Λ_v^(i) according to Eq. (4).
    Add Λ_v^(i) to T_i.
    for h_j ∈ P_i do
        Construct Λ_h^(j) according to Eq. (5).
        Add Λ_h^(j) to T_i.
    end for
end for
for all {(v_i,h_j)|W_ij ≠ 0} do                   ▷ All the connections
    Construct M^(ij) according to Eq. (6).
    Split M^(ij) into products of matrices and add each matrix into the corresponding piece (see Fig. 3).
end for
for i = 1,...,n_v do
    Contract all the internal indices of the tensors within T_i, the result is A^(i).
end for
```

B. Optimal mapping of an RBM to an MPS

The direct mapping method given in the last subsection although intuitive, is not optimal. Here, we present a method to give the MPS representation with optimal bond dimension. RBM is an undirected probabilistic graphical model. For a graph model, we divide all variables into two sets X and Y which are conditionally independent if another set of variables Z are given. This is written as

$$X \perp Y|Z. \tag{7}$$

For the bipartition, we can identify the smallest set Z such that Eq. (7) is satisfied. The degrees of freedom in Z can be treated as virtual bond of an MPS after the translation. The size of Z, denoted by $|Z|$, determines the bond dimension between the variables X and Y:

$$D = 2^{|Z|}. \tag{8}$$

Algorithm II lists the steps to translate an RBM to an MPS with optimal bond dimensions by employing such conditional independence property. We start from left and construct each tensor on the fly all along to the right side with the smallest virtual bond dimension. The virtual bonds of the resulting MPS represent the degrees of freedom of the visible or hidden units of the RBM.

Algorithm 2 Transforming an RBM into an MPS with optimal bond dimensions (see Fig. 4).

Input: The connection weight matrix W_{ij} and biases a_i, b_j. The matrix specifies the structure of the RBM.
Output: An MPS with each tensor $A^{(i)}$, $i = 1,2,\ldots,n_v$.

1: $\mathscr{G}_s = \{(i,j)|W_{ij} \neq 0\}$ ▷ Graph formed by connected units
2: $\mathscr{H}_s = \{j|(i,j) \in \mathscr{G}_s\}$ ▷ All hidden units
3: $Z' = \varnothing$ ▷ The degrees of freedom of the left virtual bond
4: **for** $i = 1,\ldots,n_v$ **do**
5: $\mathscr{G}_t = \varnothing$ ▷ Connections to be counted in tensor $A^{(i)}$
6: $\mathscr{H}_t = \varnothing$ ▷ Hidden units to be traced in tensor $A^{(i)}$
7: $X = Z' \cup \{v_i\}$
8: $Y = \{v_{i+1}, v_{i+2}, \ldots, v_{n_v}\}$ ▷ The remaining physical degrees of freedom
9: Find a minimal set Z such that $X \perp Y|Z$ on the graph $\mathscr{G}_s$.
10: **for** $j \in \mathscr{H}_s$ **do**
11: **if** h_j is not connected to $(Y \setminus Z)$ **then**
12: Move j from $\mathscr{H}_s$ to $\mathscr{H}_t$. ▷ Variable h_j will be traced out in tensor $A^{(i)}$.
13: **end if**
14: **end for**
15: **for** $(k,j) \in \mathscr{G}_s$ **do**
16: **if** v_k and h_j belongs to $X \cup Z \cup \mathscr{H}_t$ **then**
17: Move (k,j) from $\mathscr{G}_s$ to $\mathscr{G}_t$ ▷ The (v_k, h_j) interaction will be included in tensor $A^{(i)}$.
18: **end if**
19: **end for**
20: $A^{(i)}_{Z',Z}[v_i] = \sum_{\{h_{j\in\mathscr{H}_t}\}} e^{a_i v_i + \sum_{(k,j)\in\mathscr{G}_t} v_k W_{kj} h_j + \sum_{j\in\mathscr{H}_t} b_j h_j}$
21: $Z' \leftarrow Z$
22: **end for**

To illustrate the mapping algorithm, we take the RBM in Fig. 1(a) as an example. We start in Fig. 4(a) by considering $X = \{v_1\}$ and $Y = \{v_2, v_3, \ldots, v_6\}$. It is straightforward to see that Z can either be $\{v_1\}$ or $\{h_1\}$ to satisfy the conditional independence $X \perp Y|Z$. Suppose we take $Z = \{v_1\}$, $\mathscr{G}_t = \mathscr{H}_t = \varnothing$, the first tensor $A^{(1)}$ can therefore be chosen as an identity tensor which copies the visible variables $\{v^{(1)}\}$ to the right virtual bond of the matrix $A^{(1)}$.

In Fig. 4(b), we select $X = \{v_1, v_2\}$ and $Y = \{v_3, \ldots, v_6\}$. We can find that the smallest Z can be any of $\{v_1, v_2\}$, $\{v_2, h_1\}$, $\{v_3, v_4\}$, or $\{h_1, h_2\}$. We choose $Z = \{h_1, h_2\}$ for example. Note that the set Z can contain both visible and hidden units. The $\mathscr{G}_t$ in Algorithm II, denoted by the dashed lines in Fig. 4, contains all the connections considered by the tensor $A^{(2)}$. $\mathscr{H}_t = \varnothing$. The left bond of $A^{(2)}$ is the same as the right bond of $A^{(1)}$. And the right bond of $A^{(2)}$ consists of $Z = \{h_1, h_2\}$.

And we go on to Fig. 4(c), where we denote all the interactions which have been taken account as gray lines. In this step, the set $X = \{h_1, h_2\} \cup \{v_3\}$, line 7 of Algorithm II. There are several choices of Z to reach the minimum size $|Z| = 2$. Here, we choose $Z = \{v_3, v_4\}$. Therefore, $\mathscr{G}_t$ consists of all the connections between $\{h_1, h_2\}$ and $\{v_3, v_4\}$. The corresponding interactions are considered by $A^{(3)}$, whose right bond consists of $Z = \{v_3, v_4\}$. $\mathscr{H}_t = \varnothing$, no hidden units needs to be traced out.

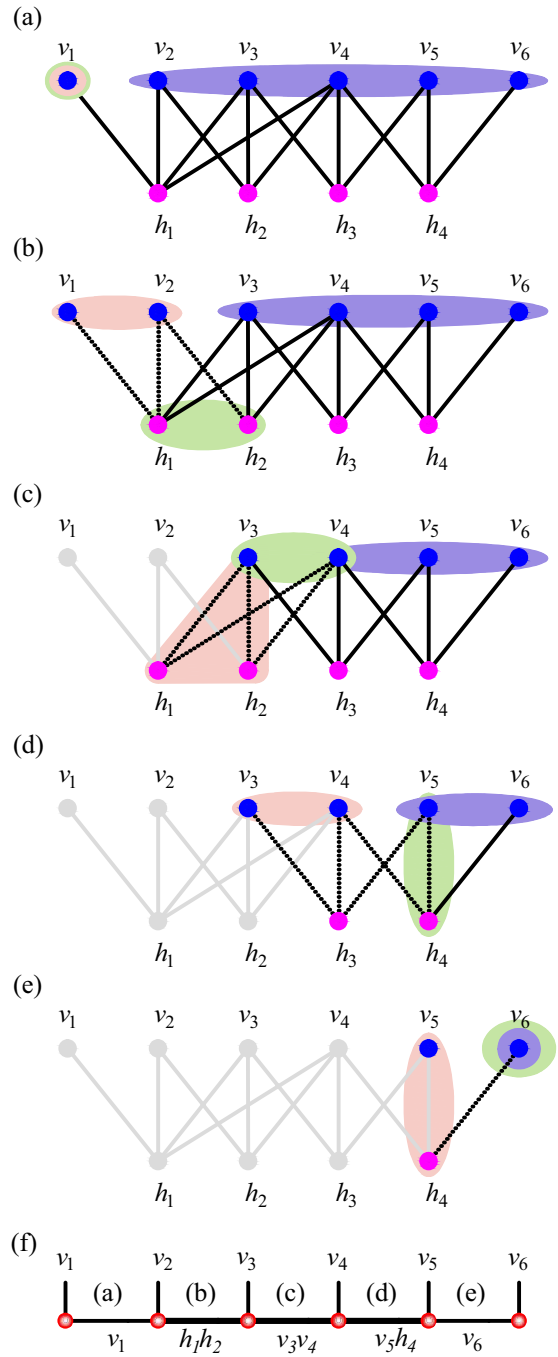

FIG. 4. The optimal mapping from the RBM to the MPS. (a)–(e) Shows each step of the construction. The set X is denoted by pink ellipse and Y in purple ellipse. Z is the minimal set satisfying $X \perp Y|Z$, denoted in light green ellipse. When the variables in Z are given, the RBM function factorizes into product of functions of variables in X and Y. The degrees of freedom in Z are represented by the virtual bond of the MPS. The connection lines in gray represent interactions counted by previous tensors. The dotted line indicates those under consideration at the current step, denoted by $\mathscr{G}_t$ in Algorithm II. The constructed tensors in (a)–(e) are given in (f).

Finally, as we come to Fig. 4(d), Z includes v_5 and h_4. We find $\mathscr{G}_t = \{h_3\}$ has no interaction with the variables in the set Y when Z is given. So, h_3 is traced out when constructing $A^{(4)}$.

Each connection line of the original RBM is considered only once during the construction of the MPS. In this way, we can obtain all the six tensors. We label the bond degrees of freedom in Fig. 4(f). Note that even if one is not running the algorithm numerically, one can still obtain the optimal bond dimension of the corresponding MPS. Moreover, this method can be used in the general undirected graphical model.

C. Implication of the RBM to MPS mapping

The connection between TNS and RBM suggests that we can use the knowledge of TNS, especially entanglement entropy, to quantify the expressibility of RBM. Let us divide the visible units into two parts, denoted as X and Y, respectively. The entanglement entropy of a function Ψ (which can either be an RBM or an MPS) between these two subsystems is then given by [35]

$$S = -\mathrm{Tr}(\rho \ln \rho), \tag{9}$$

where ρ is the reduced density matrix defined by

$$\rho = \sum_{v_Y} \Psi^*(v'_X, v_Y\, \Psi(v_X, v_Y). \tag{10}$$

The matrix is spanned by the visible degrees of freedom in X, while v_Y contains all the visible units in Y. The entanglement entropy characterizes the information content of Ψ, and can be viewed as a proxy of correlations between X and Y. In case X and Y are completely disentangled, the entanglement entropy (9) is zero. While if there are only short-range correlations, the entanglement entropy should depend only on the size of the interface separating X and Y, which is small in comparison with the full volume of the system [43]. The entanglement entropy of MPS can be readily calculated. The maximal entanglement entropy is bounded by the logarithm of the bond dimension, i.e., $\ln D$ [44].

To better assess the expressive power of an RBM one needs to find an equivalent MPS representation with the smallest possible bond dimensions. However, the simple and intuitive approach illustrated in Fig. 2 just provides an upper bound of the bond dimension which is generally higher than what is needed. For example, the bond dimension of the second bond, $D = 8$, in Fig. 2(c) is more than enough to capture the entanglement since there are only two visible units on its left, whose total degrees of freedom just equal four. Below, we present more sophisticated approaches to obtain a tighter bound on the bond dimensions. This improved approach is independent on the assignment of the hidden units.

Figure 5(a) shows an RBM after tracing out all the hidden units. The arcs indicate the interactions between the visible units mediated by the hidden units. If we separate the visible units into two parts, X and $Y(= Y_1 \cup Y_2)$, where the interface region Y_1 contains all the visible units that directly link to X, and Y_2 the rest units, the RBM function [Eq. (2)] can then be expressed as

$$\Psi_{\mathrm{RBM}}(v) = \psi(v_X, v_{Y_1})\phi(v_{Y_1}, v_{Y_2}). \tag{11}$$

Once the visible units in Y_1 are fixed, this RBM becomes a direct product of the visible units in X and Y_2. The rank of this function [45,46], or the entanglement entropy between X and Y, is therefore determined by the total number of visible units in the interface region Y_1, denoted by $|Y_1|$. Hence, the bond dimension of the MPS on the bond separating X and Y is simply given by $D = 2^{|Y_1|}$.

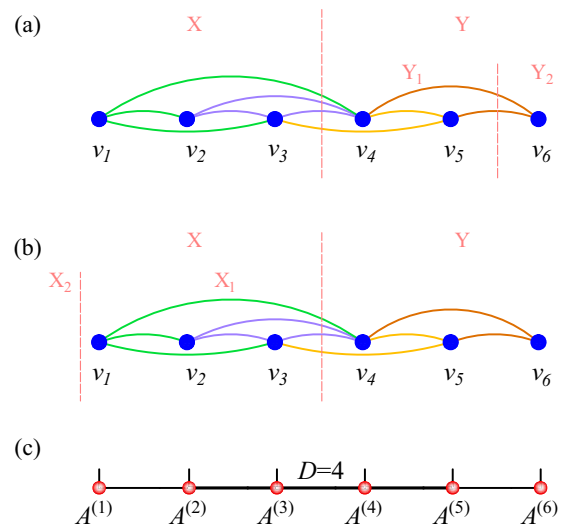

FIG. 5. (a) RBM with all hidden units traced out. The arcs represent the connections between the visible units mediated by the hidden units. The system is divided into two subsystems X and Y. Y is then further divided into two parts, $Y = Y_1 \cup Y_2$, where Y_1 contains all the units that directly connect to the units in X. (b) Alternatively, X is divided into two parts $X = X_2 \cup X_1$, where X_1 contains units that connect directly to the units in Y. (c) The resulting MPS with smaller bond dimensions in comparison with that shown in Fig. 2(c).

Alternatively, one can also divide X into two parts $X = X_2 \cup X_1$ by including all the units that have direct connections with Y in X_1 and the rest units in X_2 [Fig. 5(b)]. Following the argument given above, it can be shown that the entanglement entropy between X and Y is also bounded by the number of units in X_1. Thus, the entanglement entropy between X and Y is bounded by $S_{\max} = \min(|X_1|, |Y_1|) \ln 2$.[2]

Figure 5(c) shows the MPS obtained with this approach. The bond dimensions from left to right are $D = 2,4,4,4,2$, respectively. They are tighter bounds on the bond dimensions of MPS compared to Fig. 2(c). More generally, an even tighter bound on bond dimension can be obtained by fixing a minimal number of units, no matter they are visible or hidden, such that the RBM separates into a bipartite product state. To directly construct the MPS with the optimal smallest bond dimensions, we represent the degrees of freedom in the interface region using the virtual bonds of the MPS. Programming codes to implement these RBM to MPS mapping algorithms are given in [39].

The bond dimensions of the resulting MPS control the maximal entanglement entropies between the visible units. Therefore, the entanglement entropy provides a rigorous quantification on the expressive power of an RBM solely based on its architecture. Estimating these bounds can be done efficiently

[2]Similarly, when modeling the probability using probabilistic graphical models, the upper bound of classical mutual information is given by the size of the interface [43].

with the provided codes [39]. Moreover, canonizing the MPS may further reduce the bond dimensions by removing all unnecessary degrees of freedom and give precise value of the entanglement entropies.

Similarly, one can map an RBM into a PEPS [28] by arranging the visible units on a two-dimensional array. A similar procedure was used in [47] to map a multiscale entanglement renormalization ansatz [48] into a PEPS. This is particularly useful if the original data set represented by the RBM, for example the pixels of image, is defined on a two-dimensional grid.

In general, if the number of units along any direction of the interface region is bounded by m, then the upper bound of the entanglement entropy should scale as

$$S_{\max} \sim m V^{(d-1)/d}, \tag{12}$$

where d is the spatial dimension on which the TNS is defined, and V is the volume of the system. Thus, the maximum entanglement of a sparsely connected RBM [18] satisfies the area law. However, for a densely connected RBM, the interface region extends to the whole system and $m \sim V^{1/d}$, therefore, $S_{\max} \sim V$ scales linearly with the subsystem volume. This suggests that the dense RBM can provide a compact representation for a highly entangled quantum state that does not satisfy the entanglement area law. The number of parameters in this dense RBM just scales polynomially with the system size. However, to describe this state using an MPS or a PEPS, the number of parameters needed scales exponentially with the system size [48,49]. This provides an entanglement entropy justification for the variational calculation of quantum systems using RBM functions [17]. Section V B presents a detailed analysis of the tensor representation of the state used in Ref. [17].

The mapping from RBM to TNS is valid more generally and can be extended to Boltzmann machines without the bipartite graph restriction. In Appendix B, we discuss the general equivalence between Boltzmann machines [37] and TNS.

III. RBM REPRESENTATION OF TNS: SUFFICIENT AND NECESSARY CONDITIONS

We now address the reverse question about how to transform a TNS into an RBM with a given architecture. Here, only a given architecture of RBM is considered because otherwise one can reproduce any function using an RBM with exponentially large resources [9–11]. Again, we present a constructive approach to determine the RBM parameters for a TNS. In Appendix A, a sufficient condition for mapping a TNS to an RBM is discussed and demonstrated with the statistical Ising model and the cluster state [50].

Let us take the six-site MPS shown in Fig. 1(b) as an example to show how to parametrize it into an RBM with the architecture shown in Fig. 6(a). The hidden layer contains $n_h = 4$ units, which factorize the MPS into a product of four tensors, one $2 \times 2 \times 2 \times 2$ tensor $T^{(1)}$, defined at h_1, and three $2 \times 2 \times 2$ tensors $T^{(2,3,4)}$, defined at the remaining three hidden units. Requiring this product to equal the MPS, we have

$$\mathrm{Tr} \prod_i A^{(i)}[v_i] = T^{(1)}_{v_1 v_2 v_3 v_4} T^{(2)}_{v_2 v_3 v_4} T^{(3)}_{v_3 v_4 v_5} T^{(4)}_{v_4 v_5 v_6}. \tag{13}$$

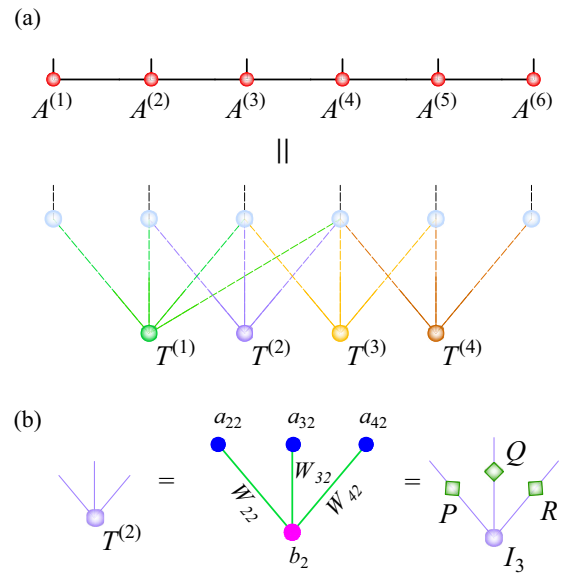

FIG. 6. Graphical representation of (a) Eq. (13) and (b) Eqs. (14) and (15).

Taking the logarithm of this equation, we obtain $2^{n_v} = 64$ linear equations for the $2^4 + 2^3 + 2^3 + 2^3 = 40$ tensor elements. These equations are overdetermined since the number of parameters is less than the number of equations. In order to map a TNS to an MPS, these equations must have a unique solution. If these equations have no solution, then one has to change the architecture of RBM. One can, for example, increase the number of parameters by increasing the number of hidden units and/or the number of connections of the RBM. This set of linear equations may become underdetermined if the number of parameters in the RBM becomes larger than the number of equations.

If Eq. (13) has a unique solution, we then need to decompose each tensor into an RBM with just one hidden unit. For example, for the tensor shown in Fig. 6(b), i.e., $T^{(2)}$, we should decouple it as

$$T^{(2)}_{v_2 v_3 v_4} = \sum_{h_2 \in \{0,1\}} e^{h_2 b_2 + \sum_{i \in \{2,3,4\}} v_i (W_{i2} h_2 + a_{i2})}, \tag{14}$$

where a_{ij} is a partial bias of the ith visible unit contributed by the jth hidden unit. The bias of the ith visible unit is given by the sum of all partial biases contributed by the hidden units connecting this visible unit, $a_i = \sum_j a_{ij}$. For this three-index tensor, seven parameters need to be determined from $2^3 = 8$ equations [Fig. 6(b)]. The number of parameters grows linearly with the order of T, but the number of equations grows exponentially instead. In general, Eq. (14) is overdetermined, and it has solution only in special cases.

In practice, Eq. (14) can be considered as the tensor rank decomposition (CP decomposition) [45,46] of $T^{(2)}$:

$$T^{(2)}_{v_2 v_3 v_4} = \sum_{h_2 \in \{0,1\}} P_{v_2 h_2} Q_{v_3 h_2} R_{v_4 h_2}. \tag{15}$$

The rank of $T^{(2)}$ is 2 because the hidden unit (h_2 here) is a binary variable. P, Q, R are all 2×2 matrices because the visible

units are all binary numbers.[3] For a $2 \times 2 \times 2$ tensor, a rank-2 decomposition always exists in complex field and a rank-4 decomposition in real field [51]. However, for an arbitrary tensor, it is difficult to determine its rank [45]. The high-order singular value decomposition [52] gives the lower bound of the tensor rank as the dimension of the core tensor. If it is larger than 2, the binary condition of the hidden unit cannot be satisfied.

After the decomposition (15), we can further decouple each matrix into a product of three matrices according to Eqs. (4)–(6). For example, we can express matrix P as

$$P = \begin{pmatrix} p & q \\ r & s \end{pmatrix} = p \begin{pmatrix} 1 & \\ & \frac{r}{p} \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & \frac{ps}{qr} \end{pmatrix} \begin{pmatrix} 1 & \\ & \frac{q}{p} \end{pmatrix}. \tag{16}$$

Comparing to Eqs. (4)–(6), we obtain

$$W_{22} = \ln \frac{ps}{qr}, \tag{17}$$

$$a_{22} = \ln \frac{r}{p}, \tag{18}$$

$$b_{22} = \ln \frac{q}{p}. \tag{19}$$

Similar to a_{ij}, b_{ij} is a partial bias to the jth hidden unit imposed by the ith visible unit. The bias of the jth hidden unit is given by $b_j = \sum_i b_{ij}$. In this way, each tensor of Eq. (13) is written in the RBM form, for example.

Thus, the necessary and sufficient condition for mapping an MPS to an RBM is that both Eqs. (13) and (14) have unique solutions. In the case of $n_h = 1$, Eq. (13) merely rephrases the MPS as the wave function itself. The rank-2 decomposition of the tensor, similar to Eq. (14), is generally more difficult to satisfy. By increasing the number of hidden units and connections, one can increase the number of parameters of RBM to ensure both Eqs. (13) and (14) to have solutions. This agrees with the mathematical results stating that RBM can represent any function by employing an exponentially large number of hidden units and connections [9–11].

In practice, a convenient way to quickly check whether a state has a particular RBM representation is to consider the factorization property defined by Eq. (11), namely, to examine whether a TNS can be factorized into a product state by fixing a sequential of visible units.

Using this simple approach, we can actually show that the spin-1 Affleck-Kennedy-Lieb-Tasaki (AKLT) state [53] cannot be represented as an RBM (with ternary visible units) with only short-range connections because of the existence of hidden string order. Expressed in the $(S^z)^{\otimes n_v}$ basis, each component of the wave function looks like "$+\,0\,0\,-\,0\,+\,-\,0\,0\,0\,+$" where $(+, 0, -)$ represent the eigenstates with $S^z = (1,0,-1)$. There is a hidden antiferromagnetic order with arbitrary number of 0's inserted. Even if we fix a sequence of visible units in the middle to be 0, the state is still combination of "$+\,0\,0\,0\ldots0\,-$" and "$-\,0\,0\,0\ldots0\,+$" which cannot be simply expressed as a product state, or an RBM with just local connections. On the other hand, the AKLT state can be written as a $D = 2$ MPS [25,53]. This example shows that the entanglement entropy is not the only variable that quantifies the expressive power of RBM.

[3]If the rank of tensor is higher than 2, then one has to enlarge the basis dimension of hidden units from 2 to a larger number.

IV. EXAMPLE: RBM REPRESENTATION OF THE TORIC CODE GROUND STATES

As a concrete example of the TNS-RBM transformation introduced in the preceding section, we construct the RBM representation of the toric code ground states from the corresponding PEPS wave functions [54]. More examples, including the Ising model and the cluster state [50], are given in the Appendix A.

The toric code [54] is one of the simplest models whose ground states are topological ordered [55] and holds the promise for quantum computation [56]. It is defined by the Hamiltonian

$$\mathcal{H} = -\sum_{+} \mathcal{A}_{+} - \sum_{\square} \mathcal{B}_{\square}, \tag{20}$$

$$\mathcal{A}_{+} = \prod_{i\in+} \sigma_i^z, \quad \mathcal{B}_{\square} = \prod_{i\in\square} \sigma_i^x, \tag{21}$$

where σ_i^x and σ_i^z are Pauli matrices defined on the horizontal and vertical links of the square lattice. $\mathcal{A}_{+}$ consists of the product of four σ_i^z operators connecting to each vertex denoted by $+$, and $\mathcal{B}_{\square}$ consists of the product of four σ_i^x operators on each minimal square plaquette $\square$. All the $\mathcal{A}_{+}$ and $\mathcal{B}_{\square}$ operators commute with each other.

The toric code on a torus has four topologically degenerate ground states. Deng *et al.* [18] have already found an RBM representation for one of them. We now present the RBM representations for all the four ground states using the approach introduced in the preceding section based on their PEPS representations [25,57].

We start from the PEPS representation for one of the ground states of the toric code model [25,57] shown in Fig. 7. The PEPS consists of three kinds of local tensors of dimension 2: a four-index identity tensor I_4 defined at each vertex, a three-index identity tensor I_3 defined at the center of each link where physical operators reside, and a U matrix linking any two neighboring I_3 and I_4:

$$U = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{22}$$

To find its RBM representation, we identify the bond-centered I_3 tensors as the visible units and the vertex-centered tensors I_4 as the hidden units, and decompose U as

$$U = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & \\ & e^0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & e^{i\pi} \end{pmatrix} \begin{pmatrix} 1 & \\ & e^0 \end{pmatrix}. \tag{23}$$

Using Eq. (16), we then find the connection weights and biases of visible and hidden units to be

$$W = i\pi, \tag{24}$$

$$a = 0, \tag{25}$$

$$b = 0. \tag{26}$$

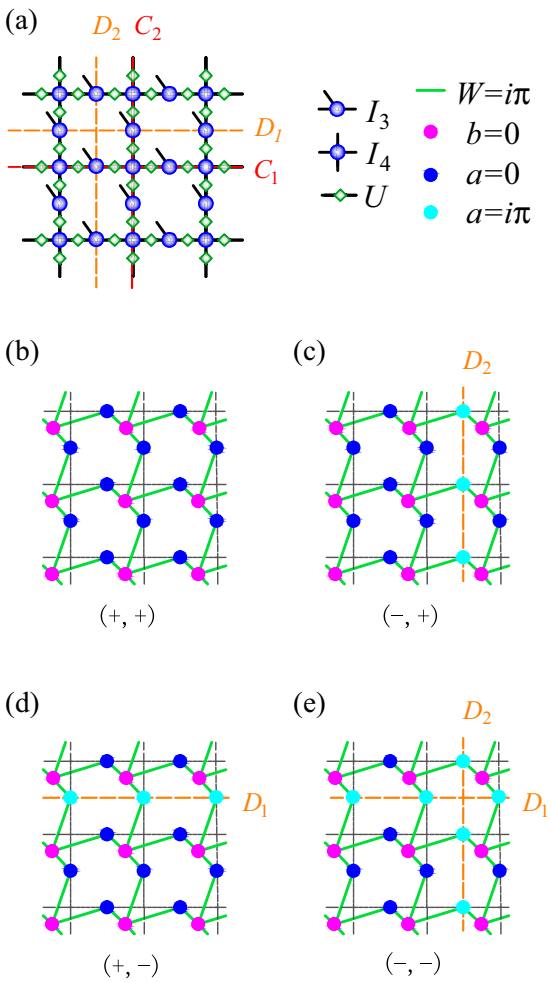

FIG. 7. (a) TNS and (b) its corresponding RBM representations of the toric code ground state at the (+,+)-topological sector on the square lattice. The dashed red and orange lines are the paths of Wilson loops used in Eqs. (28) and (29), respectively. (c)–(e) RBM representations of the toric code ground states in the other three topological sectors. The cyan dots denote the visible units with bias $a = i\pi$.

The resulting wave function is depicted in Fig. 7(b). It is a RBM with only nearest-neighboring connections between visible and hidden units. Each hidden unit couples to four visible units. Tracing out all the hidden units, the RBM becomes

$$\Psi_{\mathrm{TC}}(v) = \prod_{+}(1 + e^{i\pi \sum_{i\in+} v_i}). \tag{27}$$

It represents a quantum state with equal weight superpositions of closed loops where the sum of the visible variables is even on each vertex [54]. The RBM representation Eq. (27) is simpler than that introduced in Ref. [18] where the hidden units are defined on both the vertices and the plaquette centers.

The four ground states of the toric code model belong to four different topological sectors. To differentiate these degenerate states, we first define two Wilson loop operators

$$X_1 = \prod_{i\in C_1} \sigma_i^x, \quad X_2 = \prod_{i\in C_2} \sigma_i^x, \tag{28}$$

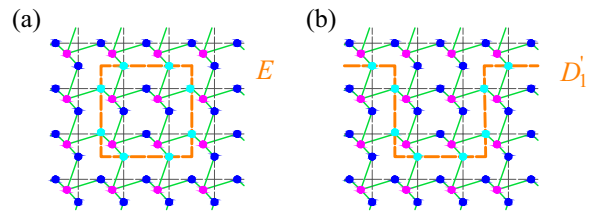

FIG. 8. Demonstration of the gauge invariance of RBM. (a) The wave function obtained by applying four $\mathcal{A}_+$ operators [Eq. (21)] within the region enclosed by the orange path E to the RBM shown in Fig. 7(b). The visible biases are changed from 0 to $i\pi$ on the path, but the two RBM wave functions before and after the transformation are gauge equivalent. (b) The RBM wave function is gauge equivalent to the that shown in Fig. 7(d) by moving the Wilson path from D_1 to D_1'. Please refer to Fig. 7 for the definition of symbols.

where C_1 and C_2 denote the paths along the horizontal and vertical directions of the lattice, indicated by the red lines in Fig. 7(a), respectively. X_1 and X_2 are mutually commuting. They also commute with the Hamiltonian. Thus, we can use their eigenvalues to label the eigenstates.

Both X_1 and X_2 have two eigenvalues, $+1$ and -1. The fourfold-degenerate ground states correspond to the four eigenstates of (X_1, X_2). They can be classified into four topological sectors, labeled by the eigenvalues of (X_1, X_2) as $(\pm,\pm)$. The wave function illustrated in Fig. 7(b) belongs to the $(+,+)$ sector, namely, $\langle v|\Psi^{(+,+)}\rangle = \Psi_{\mathrm{TC}}(v)$.

Now, let us introduce the following two operators along the paths D_1 and D_2 indicated by the dashed orange lines in Fig. 7(a):

$$Z_1 = \prod_{i\in D_1} \sigma_i^z, \quad Z_2 = \prod_{i\in D_2} \sigma_i^z. \tag{29}$$

These two operators do not commute with X_1 and X_2. They transform the states between different topological sectors [58]. For example, $|\Psi^{(-,+)}\rangle = Z_2|\Psi^{(+,+)}\rangle$, $|\Psi^{(+,-)}\rangle = Z_1|\Psi^{(+,+)}\rangle$, and $|\Psi^{(-,-)}\rangle = Z_1 Z_2|\Psi^{(+,+)}\rangle$. In the PEPS representation, $Z_{1,2}$ is to change all the identity tensors I_3 shown in Fig. 7(a) along the $D_{1,2}$ path into diagonal tensors diag$(1, -1)$. In the corresponding RBM representation, this is to change the bias of the visible variables in Eq. (25) from $a = 0$ to $a = \ln(-1) = i\pi$ along the D_1 and D_2 paths [Figs. 7(c)–7(e)].

Note that the Wilson loops D_1 and D_2 are not necessary to be straight lines. But, they must wind around the torus (Fig. 8). The RBM that is obtained by continuously distorting D_1 or D_2 is gauge equivalent to the original one. In other words, to apply the operator X_1 defined on a closed path without winding the torus to any of the ground state will not alter its topological sector. For example, the RBM shown in Fig. 8(a) is gauge equivalent to that shown in Fig. 7(b). They are related by a loop of σ_i^z operators along the path E in Fig. 8(a), which is equal to the product of all $\mathcal{A}_+$ operators enclosed by E. This product of $\mathcal{A}_+$ operators changes the visible biases along the closed loop E from $a = 0$ to $i\pi$. But, the wave function remains in the same topological sector since $\mathcal{A}_+$ at each vertex is conserving. Similarly, the state in Fig. 8(b) is identical to that in Fig. 7(c). More detailed discussion on the redundancy of the RBM parametrization is given in Sec. V A.

V. IMPLICATIONS OF THE RBM-TNS CORRESPONDENCE

A. Optimizing RBM using tensor network methods

Similar to TNS, it is known that RBM or other neural network function contains redundant degrees of freedom [3]. Two RBMs with different connection weights and biases may represent equivalent functions. An RBM can be simplified or optimized by removing redundant degrees of freedom using the well-established tensor network methods.

In one dimension, for example, one can use the canonicalization approach of MPS to optimize an RBM. To do this, we first transform an RBM into an MPS using the algorithm introduced in Sec. II. The MPS is then canonicalized to minimize the bond dimensions for all the local tensors by discarding zero singular vectors [41,42]. This can also partially fix the gauge of the MPS. Finally, we map this optimized MPS back to an RBM using the approach introduced in Sec. III. The RBM such obtained is equivalent to the original one, but is optimized.

To understand this optimization scheme, let us consider the RBM wave function of the 1D cluster state presented in Ref. [18]. This RBM contains equal number of visible and hidden units with each hidden unit connecting to three visible units. It can be mapped onto a $D = 4$ MPS using the simple approach introduced in Sec. II because each bipartition cuts two connections. By taking the canonical transformation, the bond dimension of this MPS is reduced to 2. Mapping this simplified MPS back, we obtain an optimized RBM with each hidden unit just connecting two neighboring visible units. See Appendix A for detailed parametrizations of the simplified RBM.

An RBM defined in higher dimensions can be also simplified by mapping it onto a PEPS or other higher-dimensional TNS. The bond degrees of freedom of the PEPS can be reduced or at least partially reduced (if there are redundancies), by taking higher-order singular value decompositions (or more generally Tucker decompositions) for all the local tensors [59], or just singular value decompositions for all neighboring pairs of local tensors similar to that used in the determination of the ground-state wave function by the simple update method [60].

The redundancy of RBM can be exemplified using the toric code model Eq. (20). The ground-state wave function can be obtained by applying the projection operator

$$\mathcal{P} = \prod_{+} \frac{1+\mathcal{A}_+}{2} \prod_{\square} \frac{1+\mathcal{B}_\square}{2} \tag{30}$$

to an arbitrary product state over the spins $|\psi_0\rangle = \prod_i (\cos\theta\,|\uparrow\rangle + \sin\theta\,|\downarrow\rangle)_i$. Apparently, the PEPS and its corresponding RBM such obtained is not unique. The state shown in Fig. 7(a) corresponds to the choice $\theta = \pi/4$. Figure 9(a) shows another choice with $\theta = 0$ which has four index tensors in the plaquette center. It is the eigenstate of $Z_{1,2}$ in Eq. (29), which is the superposition of the states [Figs. 7(b)–7(e)]. While Fig. 9(b) shows the state obtained with a general θ which has four index tensors defined both in the plaquette center and on the vertices. The corresponding RBM corresponds to [18] and contains more connections than the other two cases. Although these are all ground states of the toric code model, it is impossible to find a local gauge transformation in the internal bonds to connect the local tensors because the ground state is a noninjective Z_2 spin liquid state [61–63]. Exploiting the rich math structures of noninjective PEPS [64], one may further transform or simplify various RBM functions.

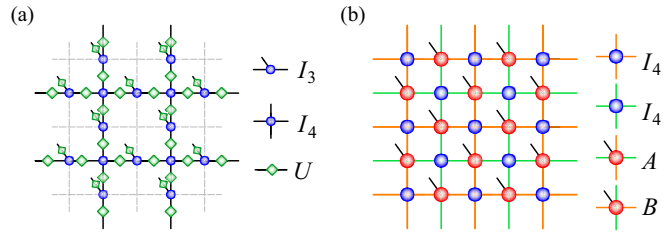

FIG. 9. Two equivalent TNS representations of the toric code ground states obtained by applying the projection operator $\mathcal{P}$ to an initial wave function $|\psi_0\rangle$ defined in the texts. (a) $\theta = 0$. The local identity tensor I_4 is defined at the center of each plaquette, and U is the matrix defined by Eq. (23). (b) θ is arbitrarily chosen. The local tensors are defined on both the original original and dual lattice sites. A and B are two θ-dependent five-index tensors.

In practice, the TNS used in the simplification of an RBM may have a huge bond dimension which is difficult to handle. There are two approaches that can be used to resolve this problem. The first is to dynamically truncate the TNS bond dimensions during the translation from the RBM to TNS. This can avoid the storage of huge TNS tensors. The other is to divide the system into several overlapped pieces, and perform the simplification for each piece separately. In either case, one can simplify the original RBM using the TNS canonicalization technique.

B. TNS representation of the shift-invariant RBM and its entanglement capacity

The variational Monte Carlo study presented in [17] employed a shift-invariant RBM function [65] to enforce the translational invariance of a physical system. The variational ansatz is a product of n_v RBM functions defined in Eq. (2):

$$\Psi(v) = \prod_{\mathcal{T}} \Psi_{\mathrm{RBM}}(\mathcal{T}v), \tag{31}$$

where $\mathcal{T}$ is the translational operator which shifts the visible variables around the periodic spatial direction. Figure 10(a) shows an example of a shift-invariant RBM. Assuming each

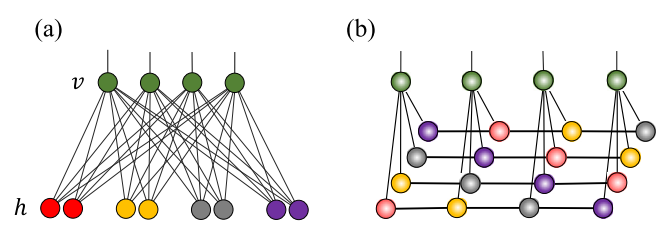

FIG. 10. (a) A shift-invariant RBM of $n_v = 4$ and $n_h = 2$, which is obtained by multiplying n_v copies of RBM with shifted connections. (b) The corresponding MPS can be constructed by connecting n_v copies of the MPS constructed from the corresponding RBM, each tensor offset by one site. The green dots are identity tensors.

RBM wave function $\Psi_{\rm RBM}$ contains n_h hidden units, the shift-invariant RBM contains $n_v n_h$ hidden variables.

This shift-invariant RBM can be also written as an MPS. To do this, we first express each individual factor $\Psi_{\rm RBM}$ as an MPS with bond dimension $D_{\rm RBM}$. For example, one of the RBMs used in Ref. [17] is a fully connected RBM with four hidden units before translational shift. When constructing the MPS, the minimum set C satisfying Eq. (7) contains all the four hidden units. It corresponds to an MPS with bond dimension $D_{\rm RBM} = 2^4 = 16$. Next, we assemble these MPSs into a tensor network as illustrated in Fig. 10(b). This tensor network can be further written as a single MPS. The corresponding bond dimension is $D = (D_{\rm RBM})^{n_v}$, which appears to have a much higher entanglement entropy bound than each factor $\Psi_{\rm RBM}$.

In order to better estimate the expressive power of the shift-invariant RBM, we can directly map it into an MPS using the algorithm introduced in Sec. II B [39]. Using this approach, one first identifies the minimal interface region and determines the optimal bond dimension based on the specific structure of the shift-invariant RBM with enlarged hidden units. For example, the enlarged shift-invariant RBM corresponds to an MPS with bond dimension $D = 2^{n_v/2}$ in the center [see Fig. 10(a)].

The shift-invariant operation plays an important role in improving the accuracy of the variational wave function [17] because it drastically increases the entanglement capability of the wave function without increasing the number of variational parameters. This construction can be generalized to other variational wave functions [28]. Moreover, this trick can be also used to implement other symmetries, such as rotation or inversion symmetries.

From the equivalence between RBM and TNS, we find that there are several guiding principles that can be used to design even more powerful RBM variational ansatz for quantum systems. First, the connection in RBM should be global because otherwise the bond dimension of the corresponding MPS will have finite bond dimensions which do not scale with the system size. Note that this requirement does not mean the RBM connections have to be dense. A sparsely connected RBM with long-range connections can also represent a TNS with large bond dimensions. In fact, the results of Ref. [17] show that many of the optimized RBM connection weights are close to zero. Reference [66] also shows that sparse connected RBM can have good performance in learning real data sets. Second, sharing the same parameters for many hidden and visible units can significantly reduce the number of independent variational parameters without scarifying the entanglement capacity of the ansatz. And, finally, one can connect multiple hidden units to the same set of visible units of RBM to mediate large entanglement between them.

C. An entanglement perspective to unsupervised learning

A natural consequence of the connection between RBM and TNS is a quantum entanglement perspective on unsupervised learning of probabilistic models. The universal approximation theorem in machine learning [9–11] states that there exists an RBM to describe a data set to any accuracy, if there is no limit on the number of hidden units. By introducing the entanglement entropy of a real data set one can better quantify the required resource in terms of the number of hidden neurons and connections of an RBM or, equivalently, the effective bond dimensions of a TNS.

We first clarify the definition of entanglement entropy for a real data set. Assuming the instances of a data set follow a probability distribution $P(v)$, we introduce a probability amplitude $\Psi(v) = \sqrt{P(v)}$ in analog to the quantum mechanical wave function. Using $\Psi(v)$, one can define the reduced density matrix and entanglement entropy of the data set following Eqs. (9) and (10). The entanglement entropy defined in this way is meaningful because it captures the complexity of a real data set similar to classical information theoretical measures [4]. Transferring quantum entanglement perspective to machine learning provides a practically useful way to quantify the difficulty of unsupervised learning and guides future progresses with insights in modeling quantum many-body states. These considerations are relevant to those generative modeling inspired by quantum physics, where one uses a wave-function square to model the probability [67–70].

Consider a data set of natural images: the correlations between pixels are typically dominated by short-range ones which suggests that the entanglement entropy of the data set defined above is relatively small. As a consequence, dense connections in the RBM are not absolutely necessary. In fact, the authors of Ref. [71] showed that a dense RBM still performs well even when 80% of the connections are randomly removed. Reference [66] also proposed an RBM with sparse connections with small-world network structure and found that it performs well compared to a densely connected RBM. Moreover, the distribution of the entanglement is inhomogeneous in various locations of the space. With an entanglement quantification, these features of the data set can be exploited in the neural network structure design.

Another advantage of introducing quantum entanglement for a realistic data set is that the RBM-TNS connection may allow one to adopt the techniques developed in quantum physics directly to machine learning. For example, it is straightforward to estimate the entanglement entropy upper bound of an RBM via counting the bond dimension of its TNS representation. Alternatively, entanglement entropy is a useful characterization of the difficulty of the learning task when directly using TNS to model the data set [69].

Interestingly, the entanglement structure of deep learning was also recently explored by authors in computer science [72]. Their discussions were mainly on feed-forward neural networks, but the key tool for achieving their conclusions is from the techniques developed in tensor networks.

D. Entanglement advantage of deep Boltzmann machines over the shallow ones

The mapping between RBM and TNS discussed in Sec. II is applicable general Boltzmann machines without the bipartite graph restriction. In particular, applying the deep Boltzmann machines (DBM) [38] explains their advantage over the shallow RBMs.

The shallow and deep BM architectures are shown in Fig. 11. They both have n_v visible units, $n_h = 3n_v$ hidden units,

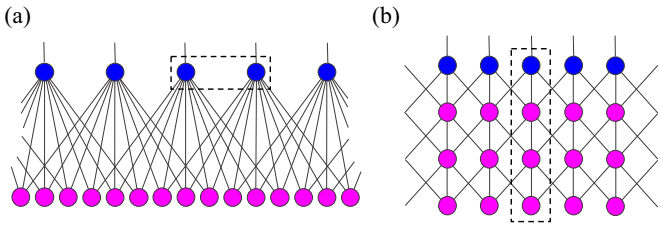

FIG. 11. (a) An RBM and (b) a deep Boltzmann machine with the same number of n_v visible units (blue dots), $n_h = 3n_v$ hidden units (magenta dots), and $9n_v$ connections. According to Sec. II, the corresponding MPS representation of the RBM and DBM had bond dimension $D_{\mathrm{RBM}} = 2^2$ and 2^4, respectively. The dashed lines enclose a minimal number of units which split the network once their values are observed. The comparison shows the that DBM has larger entanglement capacity with the given number of parameters.

and $9n_v$ connections. In contrast to the RBM, the hidden units of the DBM shown in Fig. 11(b) are organized into multilayer structures. Following Sec. II B, the C set units are enclosed by dashed lines, when they are fixed, all the visible units set X are proportional to all the right visible units set Y. The bond dimensions are $D_{\mathrm{DBM}} = 16$ and 4, respectively. Therefore, using the same amount of parameters, the DBM is able to express more complex functions with larger entanglement entropy.

As a concrete example of the application of the considerations above, we consider the 4×4 bars and stripes data set [73] used in [70] as an example. The wave function is the equal superposition of 30 valid configurations. And, the exact entanglement entropy is $\ln 15 - \frac{7}{15} \ln 7 \approx 1.80$, which is larger than $\ln 4$. This implies that the data set can not be captured by the RBM in shown in Fig. 11(a). However, capturing the same distribution using the DBM in Fig. 11(b) is possible.

In general, the visible units in a deep Boltzmann machine can have longer-ranged effective connections mediated by the deep hidden units (cf. Fig. 5), hence, a larger entanglement capacity compared to the RBM with the same number of hidden units and connections. Applying the mapping to TNS thus offers a valuable way to analyze and compare the expressive power of Boltzmann machines with various architectures.

VI. DISCUSSIONS AND OUTLOOKS

We have discussed the general and constructive connection between the RBM and TNS. This equivalence sets up a bridge between the field of deep learning and quantum physics, allowing us to use the well-established entanglement theory of TNS to quantify the expressive power of RBM and obtain lower bound on the required resources compared to previously known results [9–11]. It puts the discovered similarity between the renormalization group and deep learning [7] in a more rigorous manner, and provides a practically useful approach to remove the redundant degrees of freedom in the RBM functions (Sec. V A). Moreover, connections to the TNS identify the shift-invariant construction [65] as a key ingredient in the successful variational calculation [17] and set up useful guiding principles to construct more powerful variational ansatz (Sec. V B). Akin to the success of TNS in quantum physics, our finding suggests that the success of deep learning is related to the relatively low entanglement entropy in the data sets represented by the RBMs, such as natural images and speech signals (Sec. V C). The entanglement entropy also offers a new perspective for using deep Boltzmann machines (Sec. V D).

The correspondence between RBM and TNS suggests that the physical insights and technical methods developed in quantum many-body physics can be exploited in the field of machine learning, and vice versa. In fact, the tensor network methods have already been applied to the pattern recognition [69]. As we discussed in Sec. II, the RBM can represent the quantum state more compactly than TNS, this may offer computational advantages [17]. Deep learning algorithms [2] and industrial software and hardware [74–76] may also be beneficial for quantum many-body physics researches through this connection.

For further investigation, it is interesting to explore the connection between the deep learning architectures, such as the deep Boltzmann machines [38], and the multi-layer TNS, such as the tree tensor networks [77–80] and the multiscale entanglement renormalization ansatz [48]. In passing, we also note the efforts of understanding the expressive power of the deep feed-forward neural networks [81,82]. We believe the insights on quantum entanglement and tensor network states can deepen our understanding on deep learning and guide better neural nets design.

Note added. Recently, there appeared related works exploring representational and entanglement properties of Boltzmann machines [83–89].

ACKNOWLEDGMENTS

We thank G. Carleo, D.-L. Deng, X. Li, C. Fang, X. Gao, E. Miles Stoudenmire, H.-H. Tu, and Y-f. Yang for helpful discussions. This work is supported by the National Natural Science Foundation of China (Grants No. 11474331, No. 11190024, and No. 11774398) and the Ministry of Science and Technology of China (Grant No. 2016YFA0302400).

APPENDIX A: A SUFFICIENT CONDITION FOR RBM REPRESENTATION OF MPS/PEPS AND EXAMPLES

We give a sufficient condition for the MPS or PEPS to have an RBM representation. Many physically interesting thermal states and quantum wave function belong to this class. For example, the toric code model discussed in Sec. IV, the statistical Ising model with external field, and the 1D/2D cluster states discussed in this appendix.

A sufficient condition for an MPS to have the RBM representation is that each tensor has the following form:

$$A_{\alpha\beta}[v] = L_{\alpha v} R_{v\beta}, \tag{A1}$$

where L and R are two 2×2 matrices. As shown in Fig. 12(a), the product of R and L in orange dashed box can be replaced by coupling to the hidden unit of the RBM. The bias for the visible unit (blue dot) is $a/2$ because it is shared by two neighboring boxes. According to Eqs. (6) and (5), we can write the RL into

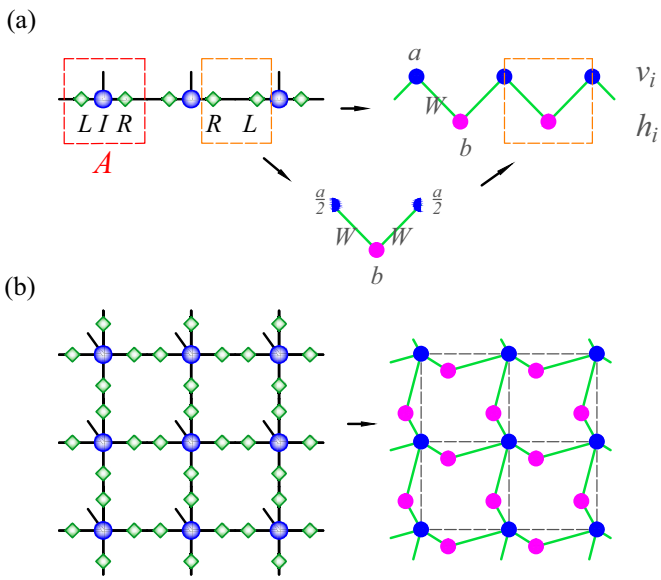

FIG. 12. (a) The tensor A in the dashed red box has the special form Eq. (A1). The green squares denote the L and R matrices, while the blue dots are the three-index identity tensor. We transform the matrix product in the dashed orange rectangular according to Eq. (A2) to obtain the RBM parameters. The half blue dot denotes $a/2$ since the visible unit is shared by two neighboring dashed orange boxes. (b) The PEPS of a similar structure is mapped to an RBM. The meaning of the symbols is the same as (a).

the RBM parameters

$$RL = \begin{pmatrix} 1 & \\ & e^{a/2} \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & e^{W} \end{pmatrix} \begin{pmatrix} 1 & \\ & e^{b} \end{pmatrix} \times \begin{pmatrix} 1 & 1 \\ 1 & e^{W} \end{pmatrix} \begin{pmatrix} 1 & \\ & e^{a/2} \end{pmatrix}. \tag{A2}$$

The decomposition can be arbitrary. Here, we choose a symmetric form for simplicity.

One example of this type is the statistical Ising model with the partition function

$$Z = \sum_{\{s_i\}} \exp\left(K \sum_{\langle i,j \rangle} s_i s_j + H \sum_i s_i \right), \tag{A3}$$

where K is the coupling constant and H is the external field. $s_i = \pm 1$ are the Ising spins. To rewrite the partition function into a summation of the RBM function of the form (2), we introduce binary variables $v_i = (s_i + 1)/2$.

In one dimension, the Ising partition function can be represented as an MPS shown in Fig. 12. The matrix product on each bond reads as

$$RL = \begin{pmatrix} e^{K+H} & e^{-K} \\ e^{-K} & e^{K-H} \end{pmatrix}. \tag{A4}$$

Combining Eqs. (A2) and (A4), we obtain the RBM parameters W,a,b summarized in the first line of Table I.

This procedure can be readily generalized to 2D, where the partition function is represented by a PEPS. The PEPS tensor follows a condition similar to Eq. (A1), as illustrated in Fig. 12(b). We introduce one hidden unit for each bond which couples to the two visible units connected by the lattice bond. The only difference compared to the 1D case is that the one replaces H by $H/2$ in the RL matrix (A4) since each site is shared by four instead of two bonds. Correspondingly, the $a/2$ bias in Eq. (A2) is replaced by $a/4$ because the visible bias is also shared by four connections [Fig. 12(b)]. The RBM parameters for 2D Ising model are summarized in the second line of Table I.

The above results show that a very simple sparse RBM with $n_h = n_v$ (or $n_h = 2n_v$ in 2D) hidden units defined on the bonds can exactly reproduce the thermal distribution of the Ising model. It is remarkable that this is independent of the coupling strength and holds even at the criticality where the correlation between the visible spins are long ranged [19]. Essentially, the effect of the hidden units of the RBM play the role of a Hubbard-Stratonovich auxiliary field which decouple the interaction on the bond [90].

Another example of the TNS satisfying Eq. (A1) is the cluster state [50]. The MPS representation [25,57] has $RL = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ on every bond. We can obtain the RBM parameters W,a,b using Eq. (A2) summarized in Table I. Similar to the Ising model case, the RBM has hidden units coupled to the physical degree of freedoms on each lattice bond. This construction is simpler than the one of Ref. [18] which requires each visible unit to connect to three hidden units. The simplification is due to that we construct the RBM representation from the canonical MPS of the cluster state.

APPENDIX B: GENERAL EQUIVALENCE BETWEEN BOLTZMANN MACHINES AND TNS

The name "restricted" in the RBM means that there are only connections between the visible and hidden units, not within them. The RBM-TNS correspondence can also be generalized to the cases without such restrictions. For example, the deep Boltzmann machines [38] have multilayers of hidden units with interconnections, and the Boltzmann machines (BM) [37] have direct connections within the visible and hidden units.

In general, the BM parametrizes a function in the form

$$\Psi_{\mathrm{BM}}(v) = \sum_h e^{-E(v,h)}, \tag{B1}$$

with the energy function

$$E(x = v \cup h) = -\left(\sum_{i,j} W_{ij} x_i x_j + \sum_k \theta_k x_k \right), \tag{B2}$$

where W_{ij} is the connection weight between the units i and j, and θ_k is the bias of the unit k. One can either view Eq. (B1) as a probability distribution or a complex wave-function amplitude.

To write Eq. (B1) into a TNS, we introduce tensor $M^{(ij)} = \begin{pmatrix} 1 & 1 \\ 1 & e^{W_{ij}} \end{pmatrix}$ on the edges and diagonal tensors $\Lambda^{(k)} = \mathrm{diag}(1, e^{\theta_k})$ on the vertices. For the visible units, the diagonal tensors have an additional dimension corresponding to the external degree of freedoms. Using these tensors, the BM

TABLE I. The RBM parameters for the statistical Ising model and the cluster state. Each hidden unit interacts with two visible units connected by a bond. The parametrizations are not unique and we only list one possible solution. The meaning of the Ising model parameters is given in Eq. (A3).

Model	W	b	a
1D Ising	$\ln(4e^{4K}-2)$	$-\ln(e^{4K}-1)-\ln 4$	$-4K-2H-2\ln 2$
2D Ising	$\ln(4e^{4K}-2)$	$-\ln(e^{4K}-1)-\ln 4$	$-8K-2H-4\ln 2$
1D cluster	$\ln\frac{3}{2}$	$-\ln 2+\mathrm{i}\pi$	$2\ln 2$
2D cluster	$\ln\frac{3}{2}$	$-\ln 2+\mathrm{i}\pi$	$4\ln 2$

Eq. (B1) can be written as a tensor network state

$$\Psi_{\mathrm{TNS}}(v)=\mathrm{Tr}\left(\prod_{i,j}M^{(ij)}\prod_{k}\Lambda^{(k)}\right). \tag{B3}$$

Conversely, one can also attempt to map a TNS back to a BM. First, we prove that any TNS constructed rank-2 tensors only can be directly mapped to a binary BM. The CP decomposition or rank decomposition [45] of $d_1\times d_2\times\cdots\times d_n$ tensor $T^{(i)}$ reads as

$$T^{(i)}_{\alpha_1\alpha_2\ldots\alpha_n}=\sum_{k=1}^{r}P^{(i)}_{\alpha_1 k}Q^{(i)}_{\alpha_2 k}R^{(i)}_{\alpha_3 k}\cdots, \tag{B4}$$

where $P^{(i)},Q^{(i)},R^{(i)},\ldots$ are matrices. The equation holds for a minimal number r, which is the rank of the tensor. For example, $r=1$ if T is a vector and r equals the smaller dimension if T is a matrix. While for $n\geqslant 3$, r is not necessarily smaller than any d_i.

A TNS can be mapped to a BM with binary units if $r=2$ and $d_i=2,\forall\, i$. When contracting the first index of $T^{(i)}$ and the second index of $T^{(j)}$, we can perform rank decomposition and obtain a 2×2 matrix on the connection

$$M^{(ij)}=\left(P^{(i)}\right)^{T}Q^{(j)}=\begin{pmatrix} p & q\\ r & s\end{pmatrix}, \tag{B5}$$

where $P^{(i)}$ ($Q^{(j)}$) is the matrix obtained from the decomposition (B4) of $T^{(i)}$ ($T^{(j)}$), respectively. With the same procedure in Eqs. (16) and (17), we find that $W_{ij}=\ln\frac{ps}{qr}$. The two diagonal matrices in Eq. (16) can be absorbed into the diagonal tensor, which contributes to θ_i.

For more general TNS with larger ranks, we can always perform the CP decomposition following the same procedure and resulting BM will have hidden units with multistates.

[1] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature **521**, 436 (2015).

[2] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning (MIT Press, 2016).

[3] C. M. Bishop, *Neural Networks for Pattern Recognition*, 1st ed. (Clarendon, Oxford, 1996).

[4] S. S. Haykin, *Neural Networks and Learning Machines* (Pearson, Upper Saddle River, NJ, 2009), Vol. 3.

[5] M. A. Nielsen, *Neural Networks and Deep Learning* (Determination Press, 2015).

[6] D. J. Amit, H. Gutfreund, and H. Sompolinsky, Spin-glass models of neural networks, Phys. Rev. A **32**, 1007 (1985).

[7] P. Mehta and D. J. Schwab, An exact mapping between the Variational Renormalization Group and Deep Learning, arXiv:1410.3831.

[8] H. W. Lin, M. Tegmark, and D. Rolnick, Why does deep and cheap learning work so well? J. Stat. Phys. **168**, 1223 (2017).

[9] Y. Freund and D. Haussler, Unsupervised learning of distributions of binary vectors using two layer networks, in *Proceeding NIPS'91 Proceedings of the 4th International Conference on Neural Information Processing Systems* (Morgan Kaufmann, San Francisco, 1994), pp. 912.

[10] N. Le Roux and Y. Bengio, Representational power of restricted Boltzmann machines and deep belief networks, Neural Comput. **20**, 1631 (2008).

[11] G. Montufar and N. Ay, Refinements of universal approximation results for deep belief networks and restricted Boltzmann machines, Neural Comput. **23**, 1306 (2011).

[12] G. E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput. **14**, 1771 (2002).

[13] T. Tieleman, *Training Restricted Boltzmann Machines Using Approximations to the Likelihood Gradient* (ACM, New York, 2008).

[14] http://www.deeplearning.net/tutorial/rbm.html

[15] G. E. Hinton and R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science **313**, 504 (2006).

[16] G. E. Hinton, S. Osindero, and Y. W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. **18**, 1527 (2006).

[17] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, Science **355**, 602 (2017).

[18] D.-L. Deng, X. Li, and S. Das Sarma, Machine learning topological states, Phys. Rev. B **96**, 195145 (2017).

[19] G. Torlai and R. G. Melko, Learning thermodynamics with Boltzmann machines, Phys. Rev. B **94**, 165134 (2016).

[20] L. Huang and L. Wang, Accelerated monte carlo simulations with restricted boltzmann machines, Phys. Rev. B **95**, 035105 (2017).

[21] J. Liu, Y. Qi, Z. Y. Meng, and L. Fu, Self-learning monte carlo method, Phys. Rev. B **95**, 041101 (2017).

[22] G. Montúfar and J. Rauh, Hierarchical models as marginals of hierarchical models, Int. J. Approx. Reason **88**, 531 (2017).

[23] G. F. Montufar, J. Rauh, and N. Ay, Expressive power and approximation errors of restricted Boltzmann machines, in *Advances in Neural Information Processing Systems 24*, edited by J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger (Curran Associates, Red Hook, NY, 2011), pp. 415

[24] L. Younes, Synchronous Boltzmann machines can be universal approximators, Appl. Math. Lett. **9**, 109 (1996).

[25] R. Orús, A practical introduction to tensor networks: Matrix product states and projected entangled pair states, Ann. Phys. (NY) **349**, 117 (2014).

[26] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac, Matrix Product State Representations, Quantum Inf. Comput. **7**, 401 (2007).

[27] I. V. Oseledets, Tensor-train decomposition, SIAM J. Sci. Comput. **33**, 2295 (2011)

[28] F. Verstraete and J. I. Cirac, Renormalization algorithms for Quantum-Many Body Systems in two and higher dimensions, arXiv:cond-mat/0407066.

[29] Z. Y. Xie, J. Chen, J. F. Yu, X. Kong, B. Normand, and T. Xiang, Tensor Renormalization of Quantum Many-Body Systems Using Projected Entangled Simplex States, Phys. Rev. X **4**, 011025 (2014).

[30] F. Verstraete, V. Murg, and J. I. Cirac, Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems, Adv. Phys. **57**, 143 (2008).

[31] T. Nishino, Density matrix renormalization group method for 2d classical models, J. Phys. Soc. Jpn. **64**, 3598 (1995).

[32] V. Murg, F. Verstraete, and J. I. Cirac, Efficient Evaluation of Partition Functions of Inhomogeneous Many-Body Spin Systems, Phys. Rev. Lett. **95**, 057206 (2005).

[33] M. Levin and C. P. Nave, Tensor Renormalization Group Approach to Two-Dimensional Classical Lattice Models, Phys. Rev. Lett. **99**, 120601 (2007).

[34] J. Eisert, M. Cramer, and M. B. Plenio, *Colloquium*: Area laws for the entanglement entropy, Rev. Mod. Phys. **82**, 277 (2010).

[35] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2010).

[36] E. M. Stoudenmire and S. R. White, Studying two-dimensional systems with the density matrix renormalization group, Annu. Rev. Condens. Matter Phys. **3**, 111 (2012).

[37] G. E. Hinton and T. J. Sejnowski, Learning and relearning in boltzmann machines, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (MIT Press, Cambridge, MA, 1986), Vol. 1, pp. 282.

[38] R. Salakhutdinov and G. Hinton, Semantic hashing Int. J. Approx. Reason **50**, 969 (2009).

[39] See https://github.com/yzcj105/rbm2mps for MATLABimplementations of the algorithms to translate an RBM to an MPS. With increasingly sophisticated approaches, the resulting entanglement entropy bounds are tighter until optimal.

[40] B. M. Marlin, K. Swersky, B. Chen, and N. Freitas, Inductive principles for restricted boltzmann machine learning, in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (PMLR, Sardinia, Italy, 2010), Vol. 9, pp. 509–516.

[41] G. Vidal, Efficient Classical Simulation of Slightly Entangled Quantum Computations, Phys. Rev. Lett. **91**, 147902 (2003).

[42] G. Vidal, Efficient Simulation of One-Dimensional Quantum Many-Body Systems, Phys. Rev. Lett. **93**, 040502 (2004).

[43] M. M. Wolf, F. Verstraete, M. B. Hastings, and J. I. Cirac, Area Laws in Quantum Systems: Mutual Information and Correlations, Phys. Rev. Lett. **100**, 070502 (2008).

[44] U. Schollwãck, The density-matrix renormalization group in the age of matrix product states, Ann. Phys. **326**, 96 (2011), Special Issue.

[45] https://en.wikipedia.org/wiki/Tensor_rank_decomposition

[46] F. L. Hitchcock, The expression of a tensor or a polyadic as a sum of products, J. Math. Phys. **6**, 164 (1927).

[47] T. Barthel, M. Kliesch, and J. Eisert, Real-Space Renormalization Yields Finite Correlations, Phys. Rev. Lett. **105**, 010502 (2010).

[48] G. Vidal, Entanglement Renormalization, Phys. Rev. Lett. **99**, 220405 (2007).

[49] G. Evenbly and G. Vidal, Class of Highly Entangled Many-Body States that can be Efficiently Simulated, Phys. Rev. Lett. **112**, 240502 (2014).

[50] H. J. Briegel and R. Raussendorf, Persistent Entanglement in Arrays of Interacting Particles, Phys. Rev. Lett. **86**, 910 (2001).

[51] J. M. Landsberg, *Tensors: Geometry and Applications* (American Mathematical Society, Providence, RI, 2012), Vol. 128.

[52] T. G. Kolda and B. W. Bader, Tensor decompositions and applications, SIAM Rev. **51**, 455 (2009).

[53] I. Affleck, T. Kennedy, E. H. Lieb, and H. Tasaki, Rigorous Results on Valence-Bond Ground States in Antiferromagnets, Phys. Rev. Lett. **59**, 799 (1987).

[54] A. Yu. Kitaev, Fault-tolerant quantum computation by anyons, Ann. Phys. (NY) **303**, 2 (2003).

[55] X. G. Wen, *Quantum Field Theory of Many-Body Systems* (Oxford University Press, Oxford, 2007).

[56] S. B. Bravyi and A. Y. Kitaev, Quantum codes on a lattice with boundary, arXiv:quant-ph/9811052.

[57] F. Verstraete, M. M. Wolf, D. Perez-Garcia, and J. I. Cirac, Criticality, the Area Law, and the Computational Power of Projected Entangled Pair States, Phys. Rev. Lett. **96**, 220601 (2006).

[58] A. Y. Kitaev, A. Shen, and M. N. Vyalyi, *Classical and Quantum Computation*, Graduate studies in mathematics (American Mathematical Society, Providence, RI, 2002).

[59] Z. Y. Xie, J. Chen, M. P. Qin, J. W. Zhu, L. P. Yang, and T. Xiang, Coarse-graining renormalization by higher-order singular value decomposition, Phys. Rev. B **86**, 045139 (2012).

[60] H. C. Jiang, Z. Y. Weng, and T. Xiang, Accurate Determination of Tensor Network State of Quantum Lattice Models in two Dimensions, Phys. Rev. Lett. **101**, 090603 (2008).

[61] N. Schuch, D. Poilblanc, J. I. Cirac, and D. Pérez-García, Resonating valence bond states in the peps formalism, Phys. Rev. B **86**, 115108 (2012).

[62] N. Read and S. Sachdev, Large- N Expansion for Frustrated Quantum Antiferromagnets, Phys. Rev. Lett. **66**, 1773 (1991).

[63] X. G. Wen, Mean-field theory of spin-liquid states with finite energy gap and topological orders, Phys. Rev. B **44**, 2664 (1991).

[64] N. Schuch, I. Cirac, and D. Pérez-García, PEPS as ground states: Degeneracy and topology, Ann. Phys. **325**, 2153 (2010).

[65] K. Sohn and H. Lee, Learning Invariant Representations with Local Transformations, Appears in Proceedings of the 29th

International Conference on Machine Learning (ICML 2012), arXiv:1206.6418.

[66] D. C. Mocanu, E. Mocanu, P. H. Nguyen, M. Gibescu, and A. Liotta, A topological insight into restricted boltzmann machines, Mach. Learn. **104**, 243 (2016).

[67] R. Bailly, Quadratic weighted automata:spectral algorithm and likelihood maximization, in *Proceedings of the Asian Conference on Machine Learning*, edited by C.-N. Hsu and W. S. Lee, Proceedings of Machine Learning Research (PMLR, Taoyuan, Taiwain, 2011), Vol. 20, pp. 147.

[68] M.-J. Zhao and H. Jaeger, Norm-observable operator models, Neural Comput. **22**, 1927 (2010), pMID: 20141473.

[69] E. Stoudenmire and D. J. Schwab, Supervised Learning with Tensor Networks, *Advances in Neural Information Processing Systems 29* (Curran Associates, Inc., 2016), pp. 4799–4807.

[70] Z.-Y. Han, J. Wang, H. Fan, L. Wang, and P. Zhang, Unsupervised Generative Modeling Using Matrix Product States, arXiv:1709.01662.

[71] V. Dumoulin, I. J. Goodfellow, A. Courville, and Y. Bengio, On the Challenges of Physical Implementations of RBMs, *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence* (2014), p. 1199.

[72] Y. Levine, D. Yakira, N. Cohen, and A. Shashua, Deep learning and quantum entanglement: Fundamental connections with implications to network design, arXiv:1704.01552.

[73] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms* (Cambridge University Press, Cambridge, 2003).

[74] https://www.nvidia.com/en-us/deep-learning-ai/developer

[75] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, TensorFlow: A system for large-scale machine learning, arXiv:1605.08695.

[76] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, arXiv:1603.04467.

[77] V. Murg, F. Verstraete, Ö. Legeza, and R. M. Noack, Simulating strongly correlated quantum systems with tree tensor networks, Phys. Rev. B **82**, 205105 (2010).

[78] P. Silvi, V. Giovannetti, S. Montangero, M. Rizzi, J. I. Cirac, and R. Fazio, Homogeneous binary trees as ground states of quantum critical Hamiltonians, Phys. Rev. A **81**, 062335 (2010).

[79] Y.-Y. Shi, L.-M. Duan, and G. Vidal, Classical simulation of quantum many-body systems with a tree tensor network, Phys. Rev. A **74**, 022320 (2006).

[80] L. Tagliacozzo, G. Evenbly, and G. Vidal, Simulation of two-dimensional quantum systems using a tree tensor network that exploits the entropic area law, Phys. Rev. B **80**, 235127 (2009).

[81] N. Cohen, O. Sharir, and A. Shashua, On the expressive power of deep learning: A tensor analysis, in *Proceedings of Machine Learning Research* (PMLR, Columbia University, New York, USA, 2016), Vol. 49, pp. 698–728.

[82] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao, Why and When Can Deep – but Not Shallow – Networks Avoid the Curse of Dimensionality: A Review, arXiv:1611.00740.

[83] D.-L. Deng, X. Li, and S. Das Sarma, Quantum Entanglement in Neural Network States, Phys. Rev. X **7**, 021021 (2017).

[84] X. Gao and L.-M. Duan, Efficient representation of quantum many-body states with deep neural networks, Nat. Commun. **8**, 662 (2017).

[85] Y. Huang and J. E. Moore, Neural network representation of tensor network and chiral states, arXiv:1701.06246.

[86] R. Kaubruegger, L. Pastori, and J. C. Budich, Chiral Topological Phases from Artificial Neural Networks, arXiv:1710.04713.

[87] S. R. Clark, Unifying Neural-network Quantum States and Correlator Product States via Tensor Networks, arXiv:1710.03545.

[88] I. Glasser, N. Pancotti, M. August, I. D. Rodriguez, and J. I. Cirac, Neural-Network Quantum States, String-Bond States, and Chiral Topological States, Phys. Rev. X **8**, 011006 (2018).

[89] Y. Nomura, A. S. Darmawan, Y. Yamaji, and M. Imada, Restricted boltzmann machine learning for solving strongly correlated quantum systems, Phys. Rev. B **96**, 205152 (2017).

[90] J. Hubbard, Calculation of Partition Functions, Phys. Rev. Lett. **3**, 77 (1959).