



*entropy*



Article

---

# An Efficient Algorithmic Way to Construct Boltzmann Machine Representations for Arbitrary Stabilizer Code

---

Yuan-Hang Zhang, Zhian Jia, Yu-Chun Wu and Guang-Can Guo

Special Issue

Quantum Information and Quantum Computation

Edited by

Dr. Zhian Jia and Dr. Dagomir Kaszlikowski



<https://doi.org/10.3390/e27060627>

## Article

# An Efficient Algorithmic Way to Construct Boltzmann Machine Representations for Arbitrary Stabilizer Code

Yuan-Hang Zhang <sup>1,2</sup> , Zhian Jia <sup>3,4,5,6,\*</sup> , Yu-Chun Wu <sup>4,5,\*</sup> and Guang-Can Guo <sup>4,5</sup><sup>1</sup> Department of Physics, University of California, San Diego, CA 92093, USA<sup>2</sup> School of the Gifted Young, University of Science and Technology of China, Hefei 230026, China<sup>3</sup> Centre for Quantum Technologies, National University of Singapore, Singapore 117543, Singapore<sup>4</sup> CAS Key Laboratory of Quantum Information, School of Physics, University of Science and Technology of China, Hefei 230026, China<sup>5</sup> CAS Center For Excellence in Quantum Information and Quantum Physics, University of Science and Technology of China, Hefei 230026, China<sup>6</sup> Microsoft Station Q and Department of Mathematics, University of California, Santa Barbara, CA 93106, USA

\* Correspondence: giannjia@foxmail.com (Z.J.); wuyuchun@ustc.edu.cn (Y.-C.W.)

**Abstract:** Restricted Boltzmann machines (RBMs) have demonstrated considerable success as variational quantum states; however, their representational power remains incompletely understood. In this work, we present an analytical proof that RBMs can exactly and efficiently represent stabilizer code states—a class of highly entangled quantum states that are central to quantum error correction. Given a set of stabilizer generators, we develop an efficient algorithm to determine both the RBM architecture and the exact values of its parameters. Our findings provide new insights into the expressive power of RBMs, highlighting their capability to encode highly entangled states, and may serve as a useful tool for the classical simulation of quantum error-correcting codes.

**Keywords:** neural network quantum state; quantum stabilizer code; restricted Boltzmann machine



Academic Editor: Antonio M. Scarfone

Received: 22 April 2025

Revised: 28 May 2025

Accepted: 9 June 2025

Published: 13 June 2025

**Citation:** Zhang, Y.-H.; Jia, Z.; Wu, Y.-C.; Guo, G.-C. An Efficient Algorithmic Way to Construct Boltzmann Machine Representations for Arbitrary Stabilizer Code. *Entropy* **2025**, *27*, 627. <https://doi.org/10.3390/e27060627>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

To conquer one of the main challenges, the dimensionality problem (also known as Hamiltonian complexity [1,2]), in condensed matter physics, many different representations of quantum many-body states are developed. For example, the well-known tensor network representations [3–5] including density-matrix renormalization group (DMRG) [6], matrix product states (MPS) [5], projected entangled pair states (PEPS) [5,7], folding algorithm [8], entanglement renormalization [9], time-evolving block decimation (TEBD) [10], string-bond state, [11] etc., have gradually become standard methods for solving quantum many-body problems. The efficiency of tensor network representations is known to be partially based on the entanglement properties of the state.

In recent years, a new representation based on a shallow neural network—the restricted Boltzmann machine (RBM)—was introduced by Carleo and Troyer [12]. They demonstrated the expressive power of this representation by computing the ground states and unitary dynamics of the transverse-field Ising model and the antiferromagnetic Heisenberg model. Subsequently, various aspects of the RBM representation have been explored. Deng et al. analyzed the entanglement properties of RBM states [13], while Gao and Duan extended the framework to deep Boltzmann machines (DBM) [14], and the entanglement property of DBM state is given in [15]. The connections between tensor networks and RBM representations have also been investigated in Refs. [14,16–18]. Moreover, many

other neural network architectures, like convolutional neural network, transformer neural network, etc., have been proposed for the efficient representation of quantum many-body states (and density operator), see, e.g., [19–24]; see Refs. [25–28] for comprehensive reviews and outlooks.

A central problem in the study of RBM states is trying to understand their representational power. Although the universality of RBMs has long been established [29], the number of hidden neurons required to represent an arbitrary distribution generally scales exponentially, rendering such constructions impractical. For RBM-based quantum states, despite numerous numerical investigations, analytical results remain scarce.

Notable analytical results on RBM representations for specific quantum states include the toric code state [30], the one-dimensional symmetry-protected topological (SPT) cluster state [30], and the graph state [14]. In Ref. [31], we investigated the RBM representation of the planar code within the stabilizer formalism and explicitly constructed sparse RBM architectures for specific stabilizer groups. However, a general construction remains elusive. Since the publication of the preprint version of this work, several studies have further explored neural network representations of code states of stabilizer codes (or equivalently ground state of local commutative Hamiltonian); see, for example, Refs. [32–34].

In this work, we comprehensively investigate the RBM representation for stabilizer code states [35,36] that cover a larger class of states than studied before. An algorithmic way to construct RBM parameters of an arbitrary stabilizer group is given, which gives a complete solution for the problem of understanding the representational power of RBM in the stabilizer formalism.

The paper is organized as follows. In Sections 2 and 3, we begin by reviewing some basic concepts of RBM states and stabilizer codes. In Section 4, we provide a detailed discussion on how to construct an RBM state for a given stabilizer code. The final section offers concluding remarks and an outlook for future work.

## 2. Preliminary Notions

An RBM is a two-layer neural network consisting of  $n$  visible neurons  $v_i \in \{0, 1\}^n$  and  $m$  hidden neurons  $h_j \in \{0, 1\}^m$ ; see Figure 1. The connections between the visible and hidden layers are specified by a weight matrix  $W_{ij}$ , while  $a_i$  and  $b_j$  denote the bias terms for the visible and hidden neurons, respectively. These parameters define a joint probability distribution over visible and hidden units:

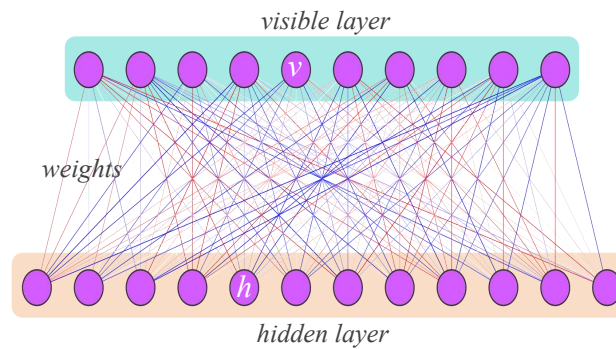
$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} \exp\left(\sum_i a_i v_i + \sum_j b_j h_j + \sum_{i,j} W_{ij} v_i h_j\right), \quad (1)$$

where the partition function  $Z$  is given by

$$Z = \sum_{\mathbf{v}, \mathbf{h}} \exp\left(\sum_i a_i v_i + \sum_j b_j h_j + \sum_{i,j} W_{ij} v_i h_j\right). \quad (2)$$

To represent a quantum many-body state, we map the local degrees of freedom of the quantum system onto the visible neurons and trace out the hidden neurons, yielding

$$\begin{aligned} \psi_{\text{RBM}}(\mathbf{v}) &= \sum_{\mathbf{h}} p(\mathbf{v}, \mathbf{h}) \\ &= \frac{1}{Z} \exp\left(\sum_i a_i v_i\right) \prod_j \left(1 + \exp\left(b_j + \sum_i v_i W_{ij}\right)\right), \quad (3) \\ |\Psi_{\text{RBM}}\rangle &= \sum_{\mathbf{v}} \psi_{\text{RBM}}(\mathbf{v}) |\mathbf{v}\rangle. \end{aligned}$$



**Figure 1.** Illustration of an RBM state. The RBM network consists of a visible layer and a hidden layer. Neurons within the same layer are not connected, while connections exist between neurons in different layers. The visible neurons  $\mathbf{v} = (v_1, \dots, v_n)$  represent the basis states of a given Hilbert space; the output of the RBM represents the coefficients  $\Psi_{RBM}(\mathbf{v})$  for each basis.

The parameters of the RBM can be chosen as complex numbers, in which case the resulting joint distribution  $p(\mathbf{v}, \mathbf{h})$  can take complex values. For more details, see, e.g., Refs. [12,26,31].

A stabilizer group  $\mathbf{S}$  is defined as an Abelian subgroup of the Pauli group  $\mathbf{P}_n = \{I, X, Y, Z\}^{\otimes n} \times \{\pm 1, \pm i\}$  that stabilizes an invariant subspace  $\mathcal{C}$  of the total space  $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$  with  $n$  physical qubits. The space  $\mathcal{C}$  is called the code space of the stabilizer group  $\mathbf{S}$ . More precisely,  $\forall T \in \mathbf{S}, \forall |\Psi\rangle \in \mathcal{C}$ , the equation  $T|\Psi\rangle = |\Psi\rangle$  is always satisfied. Suppose  $\mathbf{S}$  is generated by  $m$  independent operators,  $\mathbf{S} = \langle T_1, T_2, \dots, T_m \rangle$ . It is easy to check the following properties for the stabilizer operators:

1.  $T_j^2 = I$  for all  $j$ ,  $[T_i, T_j] = 0$  and  $-I, \pm iI \notin \mathbf{S}$ .
2.  $\langle T_1, \dots, T_k, \dots, T_m \rangle = \langle T_1, \dots, T_j T_k, \dots, T_m \rangle$ , for any  $j \neq k$ .

We refer the reader to Ref. [37] for more details about the stabilizer code.

Our goal is to find the RBM representation of code states  $|\Psi_L\rangle \in \mathcal{C}$ . We present an explicit algorithm to construct a set of basis code states  $\{|\Psi_L\rangle\}$  that span the code space  $\mathcal{C}$  for an arbitrary stabilizer group. To summarize, we aim to address the following problem:

**Problem 1.** Given a stabilizer group  $\mathbf{S}$  generated by  $m$  independent stabilizer operators  $T_1, \dots, T_m$ , does an efficient RBM representation of the code states  $|\Psi_L\rangle$  exist? If so, how can one determine the corresponding RBM parameters?

To answer this, we first introduce the standard form of a stabilizer code [35–37].

### 3. Standard Form of Stabilizer Code

Every Pauli operator  $T_k$  that squares to identity can be written as  $\alpha_k P(x_1, z_1) \otimes P(x_2, z_2) \otimes \dots \otimes P(x_n, z_n)$ , where  $\alpha_k = \pm 1$  is a phase factor, and  $P(x_i, z_i)$  is one of the Pauli matrices,

$$P(x_i, z_i) = \begin{cases} I & \text{if } x_i = 0, z_i = 0 \\ X & \text{if } x_i = 1, z_i = 0 \\ Y & \text{if } x_i = 1, z_i = 1 \\ Z & \text{if } x_i = 0, z_i = 1 \end{cases}$$

In this way, every stabilizer operator  $T_k$  can be written as the combination of a phase factor  $\alpha_k$  and a binary vector  $a_k = (x_1, \dots, x_n, z_1, \dots, z_n)$ . It is easy to prove that if  $T_l = T_j T_k$ , then  $a_l = a_j \oplus a_k$ , where  $\oplus$  denotes the bitwise addition modulo 2.

For the set of stabilizer generators  $\{T_1, \dots, T_m\}$ , we can stack all binary vectors  $a_k$  together to form an  $m \times 2n$  matrix  $A$ , called the check matrix [37]. Each row of  $A$  is a

vector  $a_k$  that corresponds to a stabilizer operator  $T_k$ . To clarify the notation, we denote  $A = (A_X|A_Z)$ , where  $A_X$  and  $A_Z$  are  $m \times n$  matrices denoting the  $x$  and  $z$  part of the binary vector  $a$ , respectively. Since  $\{T_1, \dots, T_k, \dots, T_m\}$  and  $\{T_1, \dots, T_j T_k, \dots, T_m\}$  generate the same stabilizer group, one can add one row of  $A$  to another row of  $A$  (modulo 2) without changing the code space  $\mathcal{C}$ . Meanwhile, swapping the  $p$ -th and  $q$ -th row of  $A$  corresponds to relabeling the stabilizer generators  $T_p \leftrightarrow T_q$ , and simultaneously swapping the  $p$ -th and  $q$ -th column of both  $A_X$  and  $A_Z$  corresponds to relabeling the qubits  $v_p \leftrightarrow v_q$ .

With adding and swapping operations, we can perform *Gaussian elimination* to the check matrix  $A$  [37,38]. This leads us to the standard form of a stabilizer group. This procedure is well-established, and one can refer to [37] for a detailed explanation. Here, we briefly outline the routine and clarify the notation:

We start from the original check matrix  $A = (A_X|A_Z)$ . Performing Gaussian elimination to  $A_X$ , we obtain

$$\left( \begin{array}{cc|cc} I_p & B & C & D \\ 0 & 0 & E & F \end{array} \right) \tag{4}$$

where  $I_p$  is a  $p \times p$  identity. Note that we must keep track of the phase factors  $\{\alpha_1, \dots, \alpha_m\}$  during this procedure. Further performing Gaussian elimination to  $F$ , we can obtain another identity  $I_q$ :

$$\left( \begin{array}{ccc|ccc} I_p & B_1 & B_2 & C & D_1 & D_2 \\ 0 & 0 & 0 & E_1 & I_q & F_1 \end{array} \right) \tag{5}$$

Finally, we can use  $I_q$  to eliminate  $D_1$ :

$$\left( \begin{array}{ccc|ccc} I_p & B_1 & B_2 & C_1 & 0 & D_3 \\ 0 & 0 & 0 & E_1 & I_q & F_1 \end{array} \right) \tag{6}$$

Equation (6) is called the standard form of a stabilizer code. There are  $p + q$  independent stabilizer generators, and the number of qubits encoded is  $k = n - p - q$ . If the stabilizer generators we start with are not independent of each other, zero rows will be encountered during the elimination, which we can simply discard in order to reach a set of independent generators.

One advantage of Equation (6) is that it is easy to construct logical  $X$  and  $Z$  operators from it. The check matrix for logical  $X$  and  $Z$  operators,  $\bar{X}$  and  $\bar{Z}$ , can be chosen as follows:

$$\begin{aligned} A_{\bar{X}} &= (0 \ F_1^T \ I|D_3^T \ 0 \ 0), \\ A_{\bar{Z}} &= (0 \ 0 \ 0|B_2^T \ 0 \ I). \end{aligned} \tag{7}$$

One can verify that these operators all commute with the stabilizer generators and commute with each other, except that  $\bar{X}_j$  anti-commutes with  $\bar{Z}_j$  [37].

#### 4. RBM Representation for an Arbitrary Stabilizer Group

In this section, we illustrate how to construct the RBM representation for any given stabilizer group.

Suppose the set of stabilizer generators have already been brought into a standard form like Equation (6). To start, we need to specify one code state in the code space  $\mathcal{C}$ . As an example, we choose the logical  $Z$  eigenstate with eigenvalue 1, i.e.,  $\bar{Z}_i|\Psi\rangle = |\Psi\rangle, 1 \leq i \leq k$ . We can see that we are actually treating the logical  $Z$  operators  $\bar{Z}_i$  as new independent stabilizer operators, and the stabilized subspace is narrowed down to containing one state only. The set of independent stabilizer generators now becomes  $\{T_1, \dots, T_m, \bar{Z}_1, \dots, \bar{Z}_k\}$ , with the new check matrix being

$$\left( \begin{array}{ccc|ccc} I_p & B_1 & B_2 & C_1 & 0 & D_3 \\ 0 & 0 & 0 & E_1 & I_q & F_1 \\ 0 & 0 & 0 & G & 0 & I_k \end{array} \right) \tag{8}$$

Upon introducing new independent stabilizer operators, Equation (8) can be further simplified. Eliminating  $D_3$  and  $F_1$  with  $I_k$ , we obtain the final form of the check matrix:

$$\left( \begin{array}{cc|cc} I_p & B & C & 0 \\ 0 & 0 & E & I_r \end{array} \right) \tag{9}$$

where  $r = q + k$ , and  $p + r = n$ . Denote the  $n$  stabilizer generators corresponding to Equation (9) as  $\{\tilde{T}_1, \dots, \tilde{T}_n\}$ . We call  $\tilde{T}_1, \dots, \tilde{T}_p$  X-type stabilizers, denoted by  $\tilde{T}^x$ , and  $\tilde{T}_{p+1}, \dots, \tilde{T}_n$  Z-type stabilizers, denoted by  $\tilde{T}^z$ .

With  $n$  qubits, the  $n$  stabilizer operators uniquely determines one stabilizer code state. Next, we explicitly construct this state and then translate it into the parameters of an RBM.

We start with the full expansion of the quantum state in the Pauli-Z basis

$$|\Psi\rangle = \sum_{\mathbf{v} \in \{0,1\}^n} \psi(\mathbf{v})|\mathbf{v}\rangle \tag{10}$$

where  $\mathbf{v} = (v_1, v_2, \dots, v_n)$ .

Because  $|\Psi\rangle$  is a stabilizer state, every generator  $\tilde{T}_i$  satisfies  $\tilde{T}_i|\Psi\rangle = |\Psi\rangle$ . Applying  $\tilde{T}_i$  on the expansion gives

$$\sum_{\mathbf{v}} \psi(\mathbf{v})|\mathbf{v}\rangle = \sum_{\mathbf{v}} \psi(\mathbf{v})\tilde{T}_i|\mathbf{v}\rangle \tag{11}$$

To begin, we consider a Z-type stabilizer  $\tilde{T}_i^z, i \in \{p + 1, \dots, n\}$ . By construction, it contains only the single-qubit operators  $Z$  or  $I$  and an overall phase  $\alpha_i$ . Therefore, it is diagonal in the computational basis:

$$\tilde{T}_i^z|\mathbf{v}\rangle = \phi_i(\mathbf{v})|\mathbf{v}\rangle, \quad \phi_i(\mathbf{v}) = \alpha_i(-1)^{\sum_{j: Z_j \in \tilde{T}_i^z} v_j} \in \{+1, -1\} \tag{12}$$

Plugging this into Equation (11), we obtain

$$\psi(\mathbf{v}) = \phi_i(\mathbf{v})\psi(\mathbf{v}) \tag{13}$$

Hence, for every basis string either  $\phi_i(\mathbf{v}) = +1$ , in which case  $\psi(\mathbf{v})$  may be non-zero; or  $\phi_i(\mathbf{v}) = -1$ , in which case  $\psi(\mathbf{v})$  must vanish. Each Z-type stabilizer therefore imposes a simple parity rule on the bit string: only those  $\mathbf{v}$  whose selected qubits sum to an even (or odd) parity survive.

After Gaussian elimination, the full check matrix has the block form, Equation (9), and the bottom-right block corresponds to the  $r = n - p$  Z-type stabilizers.

Crucially, each of these  $r$  generators acts with a single  $Z$  on a distinct one of the last  $r$  qubits. Because of this, once the first  $p$  qubits  $(v_1, \dots, v_p)$  are fixed, the parity constraints from the Z-type stabilizers uniquely determine the remaining  $r$  bits.

Using the notation of the check matrix

$$(0 \ 0 \ | \ E \ I_r), \tag{14}$$

the parity constraint can be expressed as

$$\begin{aligned}
 d_j &= \left(1 + \alpha_j (-1)^{\sum_k E_{jk} v_k}\right) / 2 \\
 &= \left(1 + \exp(\log \alpha_j + i\pi \sum_k E_{jk} v_k)\right) / 2 \\
 &= \begin{cases} 1, & \text{constraint satisfied,} \\ 0, & \text{constraint violated.} \end{cases}
 \end{aligned}
 \tag{15}$$

In an RBM, Equation (15) can be implemented by adding a hidden neuron  $h_j$  with bias  $\log \alpha_j$  and weight  $W_{kj} = i\pi$  between  $v_k$  and  $h_j$ .

We will, therefore, label any basis state that does satisfy all Z-type constraints as

$$|v_1 v_2 \cdots v_p \times \cdots \times\rangle, \tag{16}$$

where the “ $\times$ ” symbols stand for the uniquely determined values of qubits  $p + 1, \dots, n$ . All other computational-basis states have an amplitude of zero.

We have already seen that the  $r = n - p$  Z-type generators keep only those basis strings  $|v_1 v_2 \cdots v_p \times \cdots \times\rangle$  that satisfy their parity rules. What remains is to determine the complex amplitude attached to each of those surviving strings. That information is contained in the  $p$  X-type stabilizers  $\tilde{T}_1^x, \dots, \tilde{T}_p^x$ .

In the reduced check matrix

$$(I_p \ B \ | \ C \ 0), \tag{17}$$

- The identity block  $I_p$  says that  $\tilde{T}_j^x$  flips only qubit  $j$ ;
- The matrix  $C$  tells us which qubits carry an extra Z in  $\tilde{T}_j^x$ ;
- The matrix  $B$  flips the “ $\times$ ” qubits that are uniquely determined by the Z-type stabilizers. As stabilizers commute, the flips are consistent with the parity constraints, and we can ignore them for now;
- The zeros in the last  $r$  columns guarantee that none of the “ $\times$ ” qubits contributes any phase.

Because all generators commute, we can build any allowed basis string by starting from the “all-zero” string

$$|00 \cdots 0 \times \cdots \times\rangle, \tag{18}$$

and applying  $\tilde{T}_j^x$  exactly when  $v_j = 1$ .

Acting with  $\tilde{T}_j^x$  flips the bit  $v_j \rightarrow \bar{v}_j = 1 - v_j$  and multiplies the state by a phase

$$c_j(\mathbf{v}) = \alpha_j (-1)^{\sum_k C_{jk} v_k} = \exp(\log \alpha_j + i\pi \sum_k C_{jk} v_k) \in \{\pm 1, \pm i\} \tag{19}$$

where  $\alpha_j \in \{\pm 1, \pm i\}$  is the overall phase appearing in  $\tilde{T}_j^x$ .

Because  $\tilde{T}_j^x |\Psi\rangle = |\Psi\rangle$ , we must have

$$\psi(v_1 \cdots \bar{v}_j \cdots v_p \times \cdots \times) = c_j(\mathbf{v}) \psi(v_1 \cdots v_j \cdots v_p \times \cdots \times) \tag{20}$$

for every admissible  $\mathbf{v}$ .

With an unnormalized state, without loss of generality, we start with the reference amplitude

$$\psi(0, 0, \dots, 0, \times, \dots, \times) \equiv 1 \tag{21}$$

and successively apply Equation (20) for  $j = 1$  up to  $p$ . Let  $\mathbf{v}_j = (v_1, \dots, v_j, 0, \dots, 0, \times, \dots, \times)$ , this procedure yields

$$\begin{aligned} \psi(v_1, \dots, v_p, \times, \dots, \times) &= \prod_{j=1}^p [c_j(\mathbf{v}_{j-1})]^{v_j} \\ &= \exp\left(\sum_j v_j \log \alpha_j + i\pi \sum_{k < j} C_{jk} v_j v_k\right) \end{aligned} \tag{22}$$

In other words, every time a bit  $v_j = 1$ , we pick up a phase factor  $c_j$ . Equation (22) completely specifies the unnormalized wave function on all basis states that survive the Z-type constraints, and is already in the general form of a Boltzmann machine.

Combining Equation (22) with Equation (15), we obtain the full expression of the (unnormalized) wave function:

$$\begin{aligned} \psi(v_1, v_2, \dots, v_n) &= \exp\left(\sum_{j=1}^p v_j \log \alpha_j + i\pi \sum_{k < j} C_{jk} v_j v_k\right) \\ &\quad \cdot \prod_{j=p+1}^n \left(1 + \exp(\log \alpha_j + i\pi \sum_k E_{jk} v_k)\right) \\ |\Psi\rangle &= \sum_{\mathbf{v}} \psi(\mathbf{v}) |\mathbf{v}\rangle \end{aligned} \tag{23}$$

In this procedure, we introduced terms like  $\exp(i\pi v_j)$  and  $\exp(i\pi v_j v_k)$ . In the RBM representation, the former simply corresponds to setting the bias for the visible neuron  $v_j$ , and the latter means that we introduced a connection between visible neurons  $v_j$  and  $v_k$ . Using the conclusion in [14], this corresponds to adding a hidden neuron that connects to  $v_j$  and  $v_k$ , with the connection weights computed from Equation (24):

$$\begin{aligned} \exp(Jv_j v_k) &= \sum_h \exp(a - \ln 2 + b(v_j + v_k)(2h - 1) \\ &\quad + c(2h - 1) + d(v_j + v_k)) \end{aligned} \tag{24}$$

One solution is:

$$a = -d = -J/2, b = -c = -i \arccos(e^{J/2}) \tag{25}$$

In this way, we have finished the construction of the logical Z eigenstate of an arbitrary stabilizer group, and the eigenstate of other logical operators can be constructed in the same way. The number of hidden neurons is at most  $p(p - 1)/2 + r$ , meaning that the representation is efficient. In summary, our method can be organized into Algorithm 1.

**Example 1** ([[5, 1, 3]] code). *As an example, we take the [[5, 1, 3]] code, the smallest quantum error correcting code that can correct an arbitrary single qubit error [39], to illustrate the construction procedure. The stabilizer generators are:*

$$\begin{aligned} T_1 &= X \ Z \ Z \ X \ I \\ T_2 &= I \ X \ Z \ Z \ X \\ T_3 &= X \ I \ X \ Z \ Z \\ T_4 &= Z \ X \ I \ X \ Z \end{aligned} \tag{26}$$

After Gaussian elimination, the stabilizer generators become:

$$\begin{aligned}
 T_1 &= Y & Z & I & Z & Y \\
 T_2 &= I & X & Z & Z & X \\
 T_3 &= Z & Z & X & I & X \\
 T_4 &= Z & I & Z & Y & Y
 \end{aligned}
 \tag{27}$$

Without loss of generality, we construct the eigenstate for the logical X operator with eigenvalue 1. The logical X operator  $\bar{X} = ZIIZX$ . Since  $\bar{X}|\Psi\rangle = |\Psi\rangle$ , treating  $\bar{X}$  as the fifth stabilizer operator  $T_5$  and further carrying out Gaussian elimination using  $T_5$ , we obtain the final form of the stabilizers:

$$\begin{aligned}
 \tilde{T}_1 &= X & Z & I & I & Z \\
 \tilde{T}_2 &= Z & X & Z & I & I \\
 \tilde{T}_3 &= I & Z & X & Z & I \\
 \tilde{T}_4 &= I & I & Z & X & Z \\
 \tilde{T}_5 &= Z & I & I & Z & X
 \end{aligned}
 \tag{28}$$

There are no Z-type stabilizers, and  $\Psi(v_1v_2v_3v_4v_5)$  is obtained by the procedure specified in Algorithm 1. Explicitly writing down every term during the transition from  $|00000\rangle$  to  $|v_1v_2v_3v_4v_5\rangle$ , we obtain the following:

$$\begin{aligned}
 &\psi(v_1v_2v_3v_4v_5) \\
 &= \exp(v_1 \cdot 0) \exp(v_2 \cdot i\pi v_1) \exp(v_3 \cdot i\pi v_2) \\
 &\times \exp(v_4 \cdot i\pi v_3) \exp(v_5 \cdot i\pi(v_1 + v_4)) \\
 &= \exp\left(i\pi(v_1v_2 + v_2v_3 + v_3v_4 + v_4v_5 + v_5v_1)\right)
 \end{aligned}$$

Explicitly converting the visible connections to hidden nodes using Equation (25), the result is:

$$\begin{aligned}
 \psi(v_1v_2v_3v_4v_5) &= \exp\left((i\pi + \alpha) \sum_{i=1}^5 v_i\right) \\
 &\times \prod_{i=1}^5 \left(1 + \exp(i\pi + 2\alpha)(1 - v_i - v_{i+1})\right),
 \end{aligned}
 \tag{29}$$

where  $\alpha = \log(1 + \sqrt{2})$  and  $v_6 = v_1$ . A constant factor is omitted.

The structure of the RBM is shown in Figure 2.

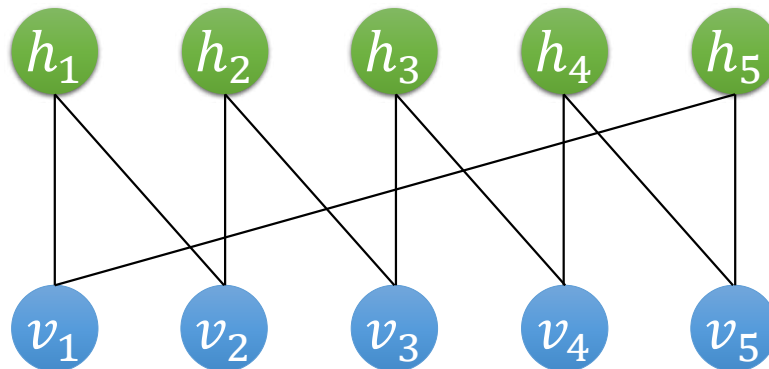


Figure 2. The RBM representation of the logical X eigenstate of the  $[[5,1,3]]$  code.

**Algorithm 1** Constructing RBM representation for the code state of the stabilizer group  $S = \langle T_1, \dots, T_n \rangle$

**Require:** A set of  $n$  commuting stabilizer generators  $G = \{T_1, \dots, T_n\}$  acting on  $n$  qubits.

**Ensure:** RBM parameters  $\Omega = \{a_i, b_\mu, W_{i\mu}\}$  that realize the corresponding code state.

**Initialize**

- 1: Set all visible biases  $a_i \leftarrow 0$ .
- 2: Set the (temporary) visible–visible matrix  $J_{ij} \leftarrow 0$ .
- 3: Set hidden-layer size  $n_h \leftarrow 0$  and initialize  $b_\mu, W_{i\mu}$  to 0.

**Bring the generators into standard form**

- 4: Use Gaussian elimination to put  $G$  into the check-matrix form  $(I_p \ B \ | \ C \ 0) \oplus (0 \ 0 \ | \ E \ I_r)$ , keeping track of the phases  $\{\alpha_1, \dots, \alpha_n\}$ .

**X-type generators**  $\{\tilde{T}_1, \dots, \tilde{T}_p\}$

- 5: **for**  $j = 1$  **to**  $p$  **do**
- 6:      $a_j \leftarrow a_j + \log \alpha_j$  ▷  $\tilde{T}_j$  flips qubit  $j$  only
- 7:     **for**  $k = 1$  **to**  $j - 1$  **do**
- 8:         **if**  $C_{jk} = 1$  **then** ▷  $Z$  on earlier qubit  $k$
- 9:              $J_{jk} \leftarrow J_{jk} + i\pi$
- 10:         **end if**
- 11:     **end for**
- 12: **end for**

**Z-type generators**  $\{\tilde{T}_{p+1}, \dots, \tilde{T}_n\}$

- 13: **for**  $\ell = p + 1$  **to**  $n$  **do**
- 14:     Create a hidden neuron  $h_\ell$ ; set  $n_h \leftarrow n_h + 1$
- 15:      $b_\ell \leftarrow \log \alpha_\ell, W_{\ell\ell} \leftarrow i\pi$
- 16:     **for all**  $k$  with  $E_{\ell k} = 1$  **do** ▷  $Z$  on qubit  $k$
- 17:          $W_{k\ell} \leftarrow i\pi$
- 18:     **end for**
- 19: **end for**

**Remove visible–visible couplings**

- 20: **for all**  $(j, k)$  with  $J_{jk} \neq 0$  and  $j < k$  **do**
- 21:     Introduce a new hidden neuron  $h$
- 22:     Compute  $a_j, a_k, b_h, W_{jh}, W_{kh}$  according to Equations (24) and (25)
- 23:      $J_{jk} \leftarrow 0$
- 24: **end for**
- 25: **return**  $\Omega = \{a_i, b_\mu, W_{i\mu}\}$

**Example 2** (Toric code). *The toric code is a paradigmatic example of a topological quantum error-correcting code, playing a central role in quantum memory, topological quantum computation, and the study of topological phases of matter [40]. In this section, we demonstrate how to obtain the RBM representation of its code states using our proposed algorithm. For simplicity, we consider the toric code defined on a  $2 \times 2$  square lattice with periodic boundary conditions (i.e., a toric code on a torus); see Figure 3. In this setting, there are four vertex operators  $A_v$  and four plaquette operators  $B_p$ :*

$$\begin{aligned} A_1 &= X_1 X_2 X_5 X_8, & A_2 &= X_1 X_4 X_5 X_6, & A_3 &= X_2 X_3 X_7 X_8, & A_4 &= X_3 X_4 X_6 X_7 \\ B_1 &= Z_1 Z_2 Z_6 Z_7, & B_2 &= Z_2 Z_3 Z_5 Z_6, & B_3 &= Z_1 Z_4 Z_7 Z_8, & B_4 &= Z_3 Z_4 Z_5 Z_8 \end{aligned} \tag{30}$$

One can check that  $A_1 A_2 A_3 A_4 = I$  and  $B_1 B_2 B_3 B_4 = I$ , indicating that two operators are redundant. After Gaussian elimination, the independent stabilizer generators are as follows:

$$\begin{aligned} T_1 &= X_1 X_4 X_5 X_6, & T_2 &= X_2 X_4 X_6 X_8, & T_3 &= X_3 X_4 X_6 X_7 \\ T_4 &= Z_3 Z_4 Z_5 Z_8, & T_5 &= Z_2 Z_4 Z_6 Z_8, & T_6 &= Z_1 Z_4 Z_7 Z_8 \end{aligned} \tag{31}$$

As an example, we construct the simultaneous eigenstate of the logical operators  $\bar{X}_1 = X_4 X_8$ ,  $\bar{Z}_2 = Z_3 Z_7$  with eigenvalue 1. Treating  $\bar{X}_1$  and  $\bar{Z}_2$  as additional stabilizer operators and further carrying out Gaussian elimination, we obtain

$$\begin{aligned}
 \tilde{T}_1 &= X_1 X_5 X_6 X_8, & \tilde{T}_2 &= X_2 X_6, & \tilde{T}_3 &= X_3 X_6 X_7 X_8, & \tilde{T}_4 &= X_4 X_8 \\
 \tilde{T}_5 &= Z_1 Z_5, & \tilde{T}_6 &= Z_1 Z_2 Z_3 Z_6, & \tilde{T}_7 &= Z_3 Z_7, & \tilde{T}_8 &= Z_1 Z_3 Z_4 Z_8
 \end{aligned}
 \tag{32}$$

Following Algorithm 1,  $\tilde{T}_1$  through  $\tilde{T}_4$  do not introduce any terms into the RBM, while  $\tilde{T}_5$  through  $\tilde{T}_8$  each introduce a hidden neuron. Explicitly writing out all the terms, we obtain:

$$\begin{aligned}
 \psi(\mathbf{v}) &= \left(1 + \exp(i\pi(v_1 + v_5))\right) \cdot \left(1 + \exp(i\pi(v_1 + v_2 + v_3 + v_6))\right) \\
 &\quad \cdot \left(1 + \exp(i\pi(v_3 + v_7))\right) \cdot \left(1 + \exp(i\pi(v_1 + v_3 + v_4 + v_8))\right)
 \end{aligned}
 \tag{33}$$

The structure of the RBM is shown in Figure 4.

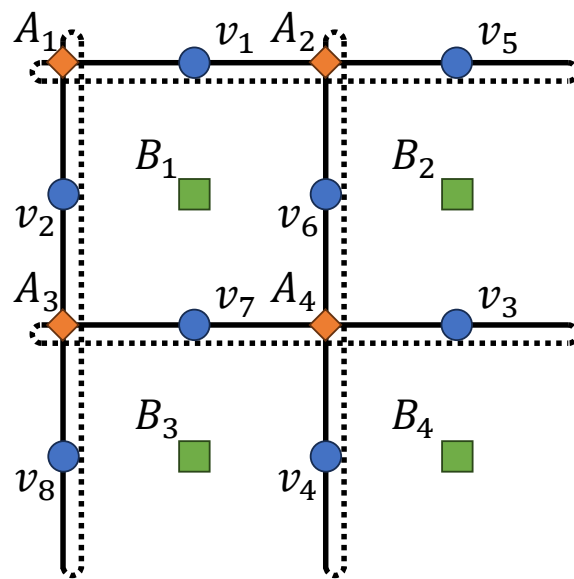


Figure 3. The toric code defined on a  $2 \times 2$  lattice with periodic boundary condition. The indices of the spins are defined such that no column swapping is required during Gaussian elimination.

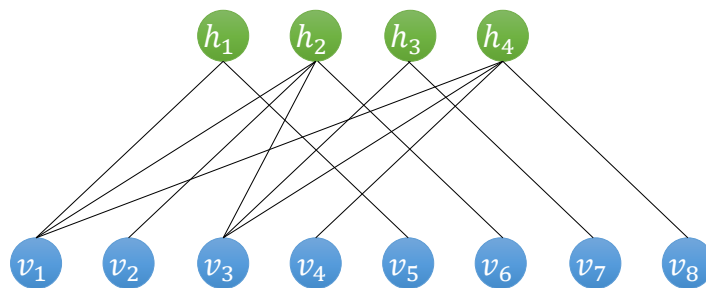


Figure 4. The RBM representation of one eigenstate of the toric code on a  $2 \times 2$  torus lattice.

### 5. Conclusions and Discussions

In this work, we systematically investigate the RBM representations of stabilizer code states and present an algorithmic procedure to construct the RBM parameters for a given stabilizer group  $S$ . To the best of our knowledge, this constitutes the first rigorous proof that RBMs can exactly and efficiently represent arbitrary stabilizer code states. While several related works (e.g., [32,33]) have emerged since our initial preprint appeared on arXiv in 2018 (see also earlier work [13,30,31], where certain special cases were addressed), this topic remains under active investigation.

Our results provide new insights into the representational power of RBMs and offer a theoretical foundation for their empirical success in modeling highly entangled quantum

states. Furthermore, given the central role of stabilizer codes in quantum error correction, our work opens up new possibilities for the classical simulation of quantum error-correcting codes using RBMs, offering a practical tool for initializing code states.

Despite the progress that has been made, several crucial directions remain open for further investigation:

1. In this work, we presented the construction for  $\mathbb{Z}_2$  stabilizer codes. However, its generalization to  $\mathbb{Z}_N$  (or more generally, to finite groups [40–42] and to the (weak) Hopf algebra setting, see e.g., [43–53]) remains largely unexplored. Such a generalization is not only of interest for applications in quantum memory and error correction, but also plays an essential role in understanding quantum phases of matter described by local commuting projector Hamiltonians, where efficient descriptions of ground and excited states are highly desirable.
2. The relationship between RBM representations and tensor network representations has attracted significant attention in recent years (see, e.g., [17,34]). It would be interesting to relate our results to existing approaches for representing quantum codes using tensor networks. Furthermore, exploring connections between RBM representations and other neural network architectures—such as convolutional neural networks, transformers, and others—is another promising direction for future research [25–27].
3. Since the code state can be represented using an RBM, establishing RBM representations of quantum operations—such as measurements and quantum channels—would allow one to embed a quantum code fully within the RBM framework. This may offer new perspectives on leveraging machine learning techniques to assist with quantum error detection and correction.

**Author Contributions:** Conceptualization, Y.-C.W. and G.-C.G.; Investigation, Y.-H.Z. and Z.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by the National Research Foundation in Singapore, the A\*STAR under its CQT Bridging Grant, CQT-Return of PIs EOM YR1-10 Funding and CQT Young Researcher Career Development Grant.

**Data Availability Statement:** The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

**Acknowledgments:** Y.-H. Zhang thanks Xiaodi Wu, Fangjun Hu, and Yuanhao Wang for helpful discussions. Z. Jia acknowledges Liang Kong for discussions during his stay in Yau Mathematical Sciences Center, Tsinghua University, and he also thanks Giuseppe Carleo and Rui Zhai for discussions during the first international conference on “Machine Learning and Physics” at IAS, Tsinghua university. Z. J. also acknowledges Dagomir Kaszlikowski for his support. This paper has been available on arXiv for several years, during which we have received valuable feedback from the community. For this version, we gratefully acknowledge the anonymous referee for suggesting a substantial revision of the references to incorporate many recent developments in the field.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Osborne, T.J. Hamiltonian complexity. *Rep. Prog. Phys.* **2012**, *75*, 022001. [[CrossRef](#)] [[PubMed](#)]
2. Verstraete, F. Quantum Hamiltonian complexity: Worth the wait. *Nat. Phys.* **2015**, *11*, 524. [[CrossRef](#)]
3. Friesdorf, M.; Werner, A.H.; Brown, W.; Scholz, V.B.; Eisert, J. Many-Body Localization Implies that Eigenvectors are Matrix-Product States. *Phys. Rev. Lett.* **2015**, *114*, 170505. [[CrossRef](#)] [[PubMed](#)]
4. Orús, R. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Ann. Phys.* **2014**, *349*, 117–158. [[CrossRef](#)]
5. Verstraete, F.; Murg, V.; Cirac, J. Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. *Adv. Phys.* **2008**, *57*, 143–224. [[CrossRef](#)]

6. White, S.R. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.* **1992**, *69*, 2863–2866. [[CrossRef](#)]
7. Verstraete, F.; Cirac, J.I. Renormalization algorithms for quantum-many body systems in two and higher dimensions. *arXiv* **2004**, arXiv:cond-mat/0407066.
8. Bañuls, M.C.; Hastings, M.B.; Verstraete, F.; Cirac, J.I. Matrix Product States for Dynamical Simulation of Infinite Chains. *Phys. Rev. Lett.* **2009**, *102*, 240603. [[CrossRef](#)]
9. Vidal, G. Entanglement Renormalization. *Phys. Rev. Lett.* **2007**, *99*, 220405. [[CrossRef](#)]
10. Vidal, G. Efficient Classical Simulation of Slightly Entangled Quantum Computations. *Phys. Rev. Lett.* **2003**, *91*, 147902. [[CrossRef](#)]
11. Schuch, N.; Wolf, M.M.; Verstraete, F.; Cirac, J.I. Simulation of Quantum Many-Body Systems with Strings of Operators and Monte Carlo Tensor Contractions. *Phys. Rev. Lett.* **2008**, *100*, 040501. [[CrossRef](#)] [[PubMed](#)]
12. Carleo, G.; Troyer, M. Solving the quantum many-body problem with artificial neural networks. *Science* **2017**, *355*, 602–606. [[CrossRef](#)] [[PubMed](#)]
13. Deng, D.L.; Li, X.; Das Sarma, S. Quantum Entanglement in Neural Network States. *Phys. Rev. X* **2017**, *7*, 021021. [[CrossRef](#)]
14. Gao, X.; Duan, L.M. Efficient representation of quantum many-body states with deep neural networks. *Nat. Commun.* **2017**, *8*, 662. [[CrossRef](#)]
15. Jia, Z.A.; Wei, L.; Wu, Y.C.; Guo, G.C.; Guo, G.P. Entanglement area law for shallow and deep quantum neural network states. *New J. Phys.* **2020**, *22*, 053022. [[CrossRef](#)]
16. Huang, Y.; Moore, J.E. Neural network representation of tensor network and chiral states. *arXiv* **2017**, arXiv:1701.06246. [[CrossRef](#)]
17. Chen, J.; Cheng, S.; Xie, H.; Wang, L.; Xiang, T. Equivalence of restricted Boltzmann machines and tensor network states. *Phys. Rev. B* **2018**, *97*, 085104. [[CrossRef](#)]
18. Glasser, I.; Pancotti, N.; August, M.; Rodriguez, I.D.; Cirac, J.I. Neural-Network Quantum States, String-Bond States, and Chiral Topological States. *Phys. Rev. X* **2018**, *8*, 011006. [[CrossRef](#)]
19. Liang, X.; Liu, W.Y.; Lin, P.Z.; Guo, G.C.; Zhang, Y.S.; He, L. Solving frustrated quantum many-particle models with convolutional neural networks. *Phys. Rev. B* **2018**, *98*, 104426. [[CrossRef](#)]
20. Zhang, Y.H.; Di Ventura, M. Transformer quantum state: A multipurpose model for quantum many-body problems. *Phys. Rev. B* **2023**, *107*, 075147. [[CrossRef](#)]
21. Wei, L.; Jia, Z.; Wang, Y.; Kaszlikowski, D.; Ling, H. Variational Transformer Ansatz for the Density Operator of Steady States in Dissipative Quantum Many-Body Systems. *arXiv* **2025**, arXiv:2502.20723. [[CrossRef](#)]
22. Torlai, G.; Melko, R.G. Latent Space Purification via Neural Density Operators. *Phys. Rev. Lett.* **2018**, *120*, 240503. [[CrossRef](#)] [[PubMed](#)]
23. Hartmann, M.J.; Carleo, G. Neural-Network Approach to Dissipative Quantum Many-Body Dynamics. *Phys. Rev. Lett.* **2019**, *122*, 250502. [[CrossRef](#)] [[PubMed](#)]
24. Nagy, A.; Savona, V. Variational Quantum Monte Carlo Method with a Neural-Network Ansatz for Open Quantum Systems. *Phys. Rev. Lett.* **2019**, *122*, 250501. [[CrossRef](#)]
25. Carleo, G.; Cirac, I.; Cranmer, K.; Daudet, L.; Schuld, M.; Tishby, N.; Vogt-Maranto, L.; Zdeborová, L. Machine learning and the physical sciences. *Rev. Mod. Phys.* **2019**, *91*, 045002. [[CrossRef](#)]
26. Jia, Z.A.; Yi, B.; Zhai, R.; Wu, Y.C.; Guo, G.C.; Guo, G.P. Quantum neural network states: A brief review of methods and applications. *Adv. Quantum Technol.* **2019**, *2*, 1800077. [[CrossRef](#)]
27. Lange, H.; Van de Walle, A.; Abedinnia, A.; Bohrdt, A. From architectures to applications: A review of neural quantum states. *Quantum Sci. Technol.* **2024**, *9*, 040501. [[CrossRef](#)]
28. Melko, R.G.; Carleo, G.; Carrasquilla, J.; Cirac, J.I. Restricted Boltzmann machines in quantum physics. *Nat. Phys.* **2019**, *15*, 887–892. [[CrossRef](#)]
29. Le Roux, N.; Bengio, Y. Representational power of restricted Boltzmann machines and deep belief networks. *Neural Comput.* **2008**, *20*, 1631–1649. [[CrossRef](#)]
30. Deng, D.L.; Li, X.; Das Sarma, S. Machine learning topological states. *Phys. Rev. B* **2017**, *96*, 195145. [[CrossRef](#)]
31. Jia, Z.A.; Zhang, Y.H.; Wu, Y.C.; Kong, L.; Guo, G.C.; Guo, G.P. Efficient machine-learning representations of a surface code with boundaries, defects, domain walls, and twists. *Phys. Rev. A* **2019**, *99*, 012307. [[CrossRef](#)]
32. Pei, M.Y.; Clark, S.R. Compact neural-network quantum state representations of jastrow and stabilizer states. *J. Phys. Math. Theor.* **2021**, *54*, 405304. [[CrossRef](#)]
33. Chen, P.; Yan, B.; Cui, S.X. Representing arbitrary ground states of the toric code by a restricted Boltzmann machine. *Phys. Rev. B* **2025**, *111*, 045101. [[CrossRef](#)]
34. Zheng, Y.; He, H.; Regnault, N.; Bernevig, B.A. Restricted Boltzmann machines and matrix product states of one-dimensional translationally invariant stabilizer codes. *Phys. Rev. B* **2019**, *99*, 155129. [[CrossRef](#)]
35. Gottesman, D. Stabilizer Codes and Quantum Error Correction. Ph.D. Thesis, California Institute of Technology, Pasadena, CA, USA, 1997.
36. Gottesman, D. Theory of fault-tolerant quantum computation. *Phys. Rev. A* **1998**, *57*, 127–137. [[CrossRef](#)]

37. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information*; Cambridge University Press: Cambridge, UK, 2010.
38. Yan, J.; Bacon, D. The k-local Pauli Commuting Hamiltonians problem is in P. *arXiv* **2012**, arXiv:1203.3906.
39. Laflamme, R.; Miquel, C.; Paz, J.P.; Zurek, W.H. Perfect Quantum Error Correcting Code. *Phys. Rev. Lett.* **1996**, *77*, 198–201. [[CrossRef](#)]
40. Kitaev, A. Fault-tolerant quantum computation by anyons. *Ann. Phys.* **2003**, *303*, 2–30. [[CrossRef](#)]
41. Beigi, S.; Shor, P.W.; Whalen, D. The Quantum Double Model with Boundary: Condensations and Symmetries. *Commun. Math. Phys.* **2011**, *306*, 663–694. [[CrossRef](#)]
42. Cong, I.; Cheng, M.; Wang, Z. Hamiltonian and Algebraic Theories of Gapped Boundaries in Topological Phases of Matter. *Commun. Math. Phys.* **2017**, *355*, 645–689. [[CrossRef](#)]
43. Buerschaper, O.; Mombelli, J.M.; Christandl, M.; Aguado, M. A hierarchy of topological tensor network states. *J. Math. Phys.* **2013**, *54*, 012201. [[CrossRef](#)]
44. Jia, Z.; Kaszlikowski, D.; Tan, S. Boundary and domain wall theories of 2d generalized quantum double model. *J. High Energy Phys.* **2023**, *2023*, 1–78. [[CrossRef](#)]
45. Jia, Z.; Tan, S.; Kaszlikowski, D.; Chang, L. On Weak Hopf Symmetry and Weak Hopf Quantum Double Model. *Commun. Math. Phys.* **2023**, *402*, 3045–3107. [[CrossRef](#)]
46. Jia, Z. Generalized cluster states from Hopf algebras: Non-invertible symmetry and Hopf tensor network representation. *J. High Energy Phys.* **2024**, *2024*, 147. [[CrossRef](#)]
47. Jia, Z. Weak Hopf non-invertible symmetry-protected topological spin liquid and lattice realization of (1+1)D symmetry topological field theory. *arXiv* **2024**, arXiv:2412.15336. [[CrossRef](#)]
48. Jia, Z. Quantum Cluster State Model with Haagerup Fusion Category Symmetry. *arXiv* **2024**, arXiv:2412.19657. [[CrossRef](#)]
49. Yan, B.; Chen, P.; Cui, S. Ribbon operators in the generalized Kitaev quantum double model based on Hopf algebras. *J. Phys. Math. Theor.* **2022**, *55*, 185201. [[CrossRef](#)]
50. Meusburger, C. Kitaev lattice models as a Hopf algebra gauge theory. *Commun. Math. Phys.* **2017**, *353*, 413–468. [[CrossRef](#)]
51. Jia, Z.; Tan, S.; Kaszlikowski, D. Weak Hopf symmetry and tube algebra of the generalized multifusion string-net model. *J. High Energy Phys.* **2024**, *7*, 207. [[CrossRef](#)]
52. Jia, Z.; Kaszlikowski, D.; Tan, S. Electric-magnetic duality and  $\mathbb{Z}_2$  symmetry enriched cyclic Abelian lattice gauge theory. *J. Phys. Math. Theor.* **2024**, *57*, 255203. [[CrossRef](#)]
53. Chang, L. Kitaev models based on unitary quantum groupoids. *J. Math. Phys.* **2014**, *55*, 041703. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.