

Quantum circuits for toric code and X-cube fracton model

Penghua Chen¹, Bowen Yan¹, and Shawn X. Cui^{1,2}

¹Department of Physics and Astronomy, Purdue University, West Lafayette

²Department of Mathematics, Purdue University, West Lafayette

We propose a systematic and efficient quantum circuit composed solely of Clifford gates for simulating the ground state of the surface code model. This approach yields the ground state of the toric code in $\lceil 2L + 2 + \log_2(d) + \frac{L}{2d} \rceil$ time steps, where L refers to the system size and d represents the maximum distance to constrain the application of the CNOT gates. Our algorithm reformulates the problem into a purely geometric one, facilitating its extension to attain the ground state of certain 3D topological phases, such as the 3D toric model in $3L + 8$ steps and the X-cube fracton model in $12L + 11$ steps. Furthermore, we introduce a gluing method involving measurements, enabling our technique to attain the ground state of the 2D toric code on an arbitrary planar lattice and paving the way to more intricate 3D topological phases.

1 Introduction

The subject of topological phases of matter (TPMs) has been under extensive study for the past few decades. Topological phases are gapped spin liquids at low temperatures which are not described by the conventional Landau theory of spontaneous symmetry breaking and local order parameters; instead, they are characterized by a new order, *topological order*. The ground states of a topological phase have stable degeneracy and robust long range entanglement. Topological phases in 2D also support quasi-particle excitations with anyonic exchange statistics which make them an appealing platform to fault-tolerantly store and process quantum information. Two peculiar features among others are that the ground state degeneracy is a topological invariant of the underlying system, and that the quasi-particles can freely move without costing energy. A large class of topological phases is realized by exactly solvable spin lattice models with bosonic degrees of freedom. A paradigmatic example in 2D is the toric code, and more generally Kitaev's quantum double model based on finite groups [6, 10], and yet even more generally the Levin-Wen string-net model based on fusion categories [11]. Examples of 3D topological phases include 3D toric model and the Walker-Wang model based on premodular categories [23].

In recent years, more exotic phases in 3D, called fracton phases, have been discovered [8, 21, 22]. Fractons also possess stable ground state degeneracy and long range entanglement. However, the ground state degeneracy of fractons depends on the system size, and hence is not a topological invariant. Moreover, the mobility of excitations is constrained.

Penghua Chen: chen3014@purdue.edu

Bowen Yan: yan312@purdue.edu, The first two authors contributed equally to this work.

Shawn X. Cui: cui177@purdue.edu, Corresponding author.

The excitations can only move in certain subsystems or cannot move at all. Well known examples of fractons include the Haah code [8] and the X-cube model [22]. While regular topological phases are described by topological quantum field theories, it is still an open question what theories mathematically characterize fractons. Since fractons also satisfy the topological order conditions in the sense of [2], we call the ground states of a fracton topologically ordered states, in the same way as those of regular topological phases.

Realizing topological phases in physical systems remains an extremely challenging task. On the other hand, there now exist quantum processors based on a number of platforms such as superconducting qubits [14], Rydberg atomic arrays [7], etc. These devices can host physical qubits at the scale of 10^2 , and this number is expected to increase significantly in the near future. Hence, it is both feasible and interesting to simulate topological phases in quantum processors. Thanks to the intrinsic robustness of topological phases, the simulation is relatively less sensitive to the noises in the current quantum processors. We may also gain more insight in topological phases by engineering them in processors.

The toric code ground states were realized in the superconducting-qubit-based systems [14] and the Rydberg-atom systems [19]. In [14], the authors gave a quantum circuit consisting of Clifford gates to realize the ground states of the planar toric code (a.k.a. surface code [5]). Quantum circuits realizing non-Abelian topological orders such as Levin-Wen string-net model and Kitaev quantum double model have also been studied. See for instance [12, 18, 20, 4, 16, 17], though in these cases, the gates utilized are no longer in the Clifford group and measurements are required.

In this paper, we develop quantum circuits realizing the ground states for a number of topological phases. In [14], only planar toric code is considered where the lattice is defined on a planar surface. Here we generalize their method to apply to a large class of surfaces with or without boundary. The quantum circuit consists of only Clifford gates. In toric code, the Hamiltonian consists of two types of operators, the term A_v for each vertex v and the term B_p for each plaquette p . See Figure 2. The key idea of constructing the ground state in [14] is as follows. Start with the product state $|\phi_0\rangle = |00\dots 0\rangle$ (also written as $|0\rangle^{\otimes}$) which is the +1 eigenstate for all vertex terms. The ground state is then obtained by projecting $|\phi_0\rangle$ to the +1 eigenstate of all plaquette operators,

$$|GS\rangle \sim \prod_p \frac{1+B_p}{2} |\phi_0\rangle. \quad (1)$$

The effect of $\frac{1+B_p}{2}$ acting on certain states can be simulated by an appropriate combination of the Hadamard gate and the CNOT gate. For this method to work, the control qubit for CNOT has to be in the $|0\rangle$ state prior to applying the Hadamard and CNOT. Hence, it is critical to choose the right sequence for the plaquettes so that, immediately before simulating the term corresponding to each plaquette p , there is always at least one edge on the boundary of p with the state $|0\rangle$. When the lattice is a simple planar lattice, the problem can be easily solved by dividing the lattice into several parts and applying the CNOT gates in a specific order. In this paper, since we consider lattices on arbitrary surfaces, this question is much subtler.

Here we provide an explicit algorithm to determine the sequence in which the plaquette operators are simulated. We show that this is always possible for a large class of lattices with or without boundary. The result of the algorithm is a quantum circuit consisting of Clifford gates realizing the ground state of the toric code. Moreover, we also adapt this method to 3D phases including the 3D toric model and the X-cube fracton model. For the X-cube model, we again initialize the state to the product of $|0\rangle$ state and simulate the projectors corresponding to cube terms. A similar issue arises that we need to choose

the correct sequence to simulate the cube terms. We note that the circuit we provide here realizes an *exact* ground state of the X-cube model. By comparison, using cluster states and measurements, the authors in [20] gave an *approximate* realization of the model.

In addition to the above method using only quantum gates, we also provide a different way of realizing the same states. The alternative way, called gluing method, combines Clifford gates and measurement of the Pauli X gate. The resulting circuit has a shorter depth than the first one. Of course, for the toric code or X-cube model, it is possible to only use measurement to obtain the ground state. Considering that frequent measurements in near-term quantum processors are costly, our method is a trade-off between circuit depth and degree of measurements.

2 Realizing ground state of 2D toric code

2.1 Toric code

It is well known that for any Hamiltonian of the form

$$H = - \sum_i P_i, \quad (2)$$

where all elements in P_i are projectors and mutually commuting, $|GS\rangle$ as defined below is a ground state as long as it is non-zero:

$$|GS\rangle = \prod_i P_i |\phi\rangle, \quad (3)$$

where $|\phi\rangle$ stands for an arbitrary state. Specifically, in a given connected lattice Γ , V refers to the set of vertices, P refers to the set of plaquettes and E refers to the set of edges. We define $Bo(p) \subseteq E, p \in P$ to be the set of boundary edges of the plaquette p , $\tau(e) \subseteq P, e \in E$ to be the set of plaquettes for which e is a boundary edge, and $\sigma(e) \in V, e \in E$ to be the set of vertices attached with the edge e . When each edge is associated with a Hilbert space, we may abuse the notation and use e to represent both an edge and the Hilbert space associated to the edge. As an example, if an edge e appears as a subscript of an operator, it means the operator acts on the Hilbert space attached to the edge e .

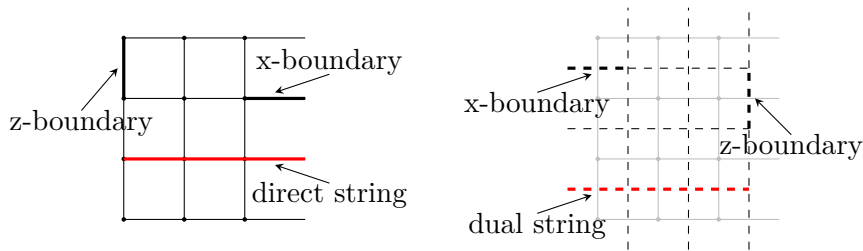


Figure 1: The black solid net on the left represents the lattice Γ and the black dashed net on the right represents the dual of Γ induced by the gray net.

As shown in Figure 1, an edge e is a z -boundary when $\tau(e)$ contains only one element, and it is an x -boundary if $\sigma(e)$ contains only one element (see [5] for details). On the lattice Γ , a direct string S is a series of edges $e_i, i = 1 \cdots n$ such that $\tau(e_i) \cap \tau(e_{i+1}) \neq \emptyset$ for $1 \leq i < n$. A direct string operator $F(S)$ is one operator applies X on all edges along

the string S , thereby generating two electric charges at both ends. Likewise, the concept of a dual string S' can be introduced, representing a direct string in the dual lattice of Γ . A dual ribbon operator $F(S')$ applies Z to all edges intersected by the dual string S' , resulting in the creation of two magnetic charges at its endpoints. It is worth noting that dual string operators that encounter a z -boundary at one end will generate (or annihilate) a magnetic charge at the other end.

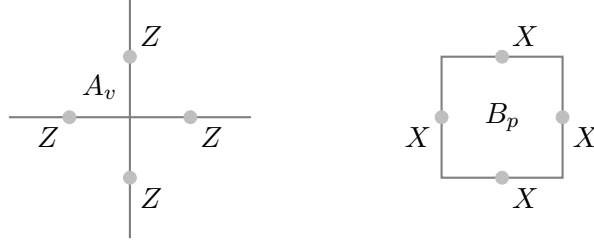


Figure 2: Definitions of A_v and B_p operators in toric code.

The toric code Hamiltonian H is composed of operators defined in Figure 2:

$$H = - \sum_{v \in V} A_v - \sum_{p \in P} B_p. \quad (4)$$

The action of A_v (vertex term) is to apply Pauli matrix Z over edges e if $v \in \sigma(e)$, and B_p (plaquette term) acts to apply Pauli matrix X over edges e if $p \in Bo(e)$. Given that $A_v^2 = B_p^2 = 1$ and $[A_v, B_p] = 0$ for all $v \in V$ and $p \in P$, it can be easily verified that $\frac{1+A_v}{2}$ and $\frac{1+B_p}{2}$ function as projectors. By substituting A_v with $\frac{1+A_v}{2}$ and B_p with $\frac{1+B_p}{2}$ in the Hamiltonian, we achieve a equivalent form that matches the form of Equation 2. This is due to a natural one-to-one correspondence between their spectra. So we get a ground state¹ by Equation 3:

$$|GS\rangle = \prod_{p \in P} \frac{1+B_p}{2} |\phi_0\rangle, \quad (5)$$

where $|\phi_0\rangle = |00\dots 0\rangle$ represents a product state where each qubit is in the $|0\rangle$ state, and we drop all $\frac{1+A_v}{2}$ s due to its trivial action on $|\phi_0\rangle$. This state is non-zero due to the presence of positive coefficients of each components.

2.2 Single plaquette

To systematically introduce our ground state simulation method, we initiate with the most elementary scenario: applying $\frac{1+B_p}{2}$ on a single plaquette, which is the *basic structure* in 2D toric code. A Hadamard gate H is naturally described by $\frac{X+Z}{\sqrt{2}}$, and CNOT gate $C_{i \rightarrow j}$ is defined as

$$C_{i \rightarrow j} |ij\rangle = \left(\frac{1-Z_i}{2} X_j + \frac{1+Z_i}{2} \right) |ij\rangle, \quad (6)$$

where i is the control qubit and j is the target qubit.

In the single plaquette shown in Figure 3, four qubits labeled 1, 2, 3, and 4 are initialized to the state $|0\rangle$. Subsequently, we will systematically implement Hadamard and CNOT

¹The ground state degeneracy of the 2D toric code on a torus is four, and the state $|GS\rangle$ corresponds to $|00\rangle$. A comprehensive explanation can be found in Section 2.5.

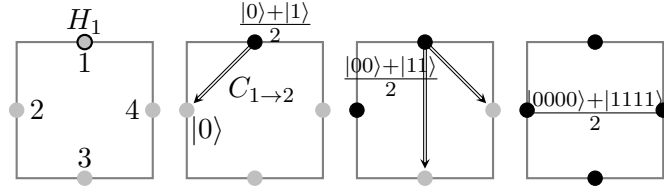


Figure 3: Initially, a qubit in the state $|0\rangle$ is situated at each gray dot. As quantum gates are applied to these qubits, their color changes to black. A circle positioned on a dot signifies the application of a Hadamard gate to the corresponding qubit, while an arrow indicates a CNOT gate, with the arrowhead pointing from the control qubit to the target qubit.

gates in a specific sequence, as outlined in the figure. After the application of H_1 and $C_{1\rightarrow 2}$, we have

$$C_{1\rightarrow 2}H_1|0000\rangle = \left(\frac{1-Z_1}{2}X_2 + \frac{1+Z_1}{2}\right)\frac{X_1+Z_1}{\sqrt{2}}|0000\rangle = \frac{X_1X_2+1}{\sqrt{2}}|0000\rangle. \quad (7)$$

Explicitly, we can insert a $\frac{1+Z}{2}$ into the equation as $\frac{1+Z}{2}|0\rangle = |0\rangle$:

$$\left(\frac{1-Z_1}{2}X_2 + \frac{1+Z_1}{2}\right)\frac{X_1+Z_1}{\sqrt{2}}\frac{1+Z_1}{2} \quad (8)$$

$$= \frac{1}{\sqrt{2}}\left\{\left(X_1\frac{1+Z_1}{2}X_2 + X_1\frac{1-Z_1}{2}\right) + \left(\frac{1-Z_1}{2}X_2 + \frac{1+Z_1}{2}\right)\right\}\frac{1+Z_1}{2} \quad (9)$$

$$= \frac{1}{\sqrt{2}}(X_1X_2+1)\frac{1+Z_1}{2} \quad (10)$$

Notice $\frac{1+Z_1}{2}$ survives within $\{ \}$ in Equation 9. After reverting to the original expression and substituting $\frac{1+Z_1}{2}$ with 1, we verified the accuracy of Equation 7. Importantly, this equation holds for any quantum state $|\phi\rangle$:

$$C_{1\rightarrow 2}H_1|0\rangle \otimes |\phi\rangle = \frac{X_1X_2+1}{\sqrt{2}}|0\rangle \otimes |\phi\rangle, \quad (11)$$

since the key step only requires that the initial state must be the eigenstate of Z_1 with an eigenvalue +1. Finally, applying the other CNOT gates results in

$$\prod_{i=2}^4 C_{1\rightarrow i}H_1|0000\rangle = \frac{X_1X_2X_3X_4+1}{\sqrt{2}}|0000\rangle = \frac{1+B_p}{\sqrt{2}}|0000\rangle, \quad (12)$$

which is the desired ground state. It is important to observe that this procedure remains effective as long as a qubit from $Bo(p)$ is initially in the state $|0\rangle$. We term such qubits as *free qubits*, and their presence is pivotal when considering scenarios involving multiple plaquettes.

2.3 Developing to a surface with boundary

Given a complicated lattice Γ in the state $|\phi_0\rangle$, we need to find a path (termed permissible order in [12]) through all plaquettes p_i , such that $\bigcup_i p_i = P$, using a sequence of edges $e_i \in Bo(p_i)$ where $e_i \notin \bigcup_{j=1}^{i-1} p_j$. Each e_i is then utilized as a free qubit to apply the introduced basic structure, resulting in the accumulation of $\prod_i \frac{1+B_{p_i}}{\sqrt{2}}$ over $|0 \cdots 0\rangle$, which represents the ground state of the toric code on lattice Γ . To illustrate the procedure, we take four plaquettes as an example depicted in Figure 4. A path featuring four free qubits e_1 to e_4 has been chosen, where e_i starts in the state $|0\rangle$ at the onset of every step. Upon completing the path, the desired ground state is eventually obtained.

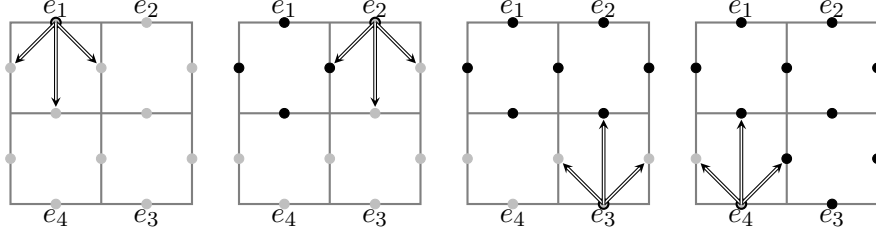


Figure 4: The procedure on the basics structure is applying Hadamard gate on any qubit at $|0\rangle$ first and CNOT gates to other qubits in any order.

2.4 Developing to a surface without boundary

The scenario shifts when dealing with a surface without boundary. While the initial state remains $|\phi_0\rangle$, it becomes impossible to locate a path with sufficient free qubits to cover the entire lattice. Fortunately, as every edge sides two plaquettes, the equation holds:

$$\prod_{p \in P} B_p = 1, \quad (13)$$

which implies that we can intentionally choose a specific B_p as redundant. Consequently, we can select the final plaquette as the redundant one, effectively terminating the path. To illustrate, consider the lattices on a torus shown in Figure 5, there is no need to apply Hadamard and CNOT gates to the bottom left plaquette, as we have previously simulated the toric code's ground state.

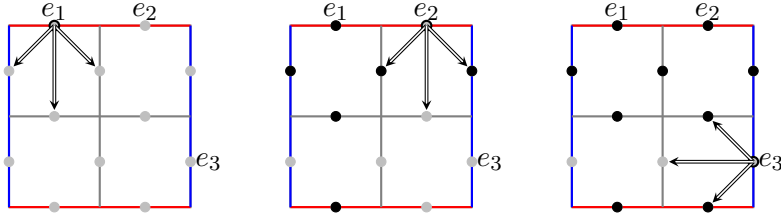


Figure 5: Boundaries with the same color are identified to represent a torus.

This method remains applicable to more intricate 2D surfaces with or without boundary, provided a suitable path can be identified. Additional examples are provided in Appendices A and B. Furthermore, the gluing method described in Section 3 empowers us to simulate ground states on arbitrary planar lattices.

2.5 Simulate arbitrary ground state

As stated in [10], the degeneracy of ground states for 2D toric code on torus is four: $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$. The ground state $|00\rangle$, presented in Section 2.4, is simulated from the initial state ϕ_0 . Due to the properties of logical operators, which can interchange ground states and commute with B_p , it is feasible to apply them to ϕ_0 to obtain the remaining ground states.

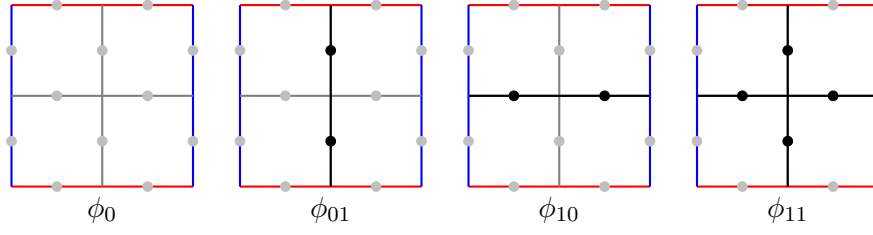


Figure 6: A qubit $|0\rangle$ is placed at each gray dot and the color changes to black when operator X flips the qubit from $|0\rangle$ to $|1\rangle$.

Illustrated in Figure 6, a vertical loop and a horizontal loop of X represent the two logical operators. They are capable of transforming ϕ_0 into ϕ_{01} , ϕ_{10} and ϕ_{11} , which correspond to the initial states of $|01\rangle$, $|10\rangle$ and $|11\rangle$, respectively. Subsequently, we can repeat the same procedure detailed in Section 2.4, but with a modification: utilize $X_i C_{i \rightarrow j} X_i$ instead of $C_{i \rightarrow j}$ when encountering a flipped qubit e_i .

One step further, in order to obtain an arbitrary ground state $\Phi = ae^{i\theta_a}|00\rangle + be^{i\theta_b}|01\rangle + ce^{i\theta_c}|10\rangle + de^{i\theta_d}|11\rangle$, we can implement the unitary operator U outlined in Equation 14 on an adjacent pair of vertical and horizontal edges of ϕ_0 and subsequently utilize CNOT gates to transmit vertically and horizontally to get ϕ . From there, we can proceed by repeating the aforementioned method and we must avoid qubits that have already been utilized, opting instead for free qubits.

$$U_1 = \begin{pmatrix} \frac{a}{\sqrt{a^2+b^2}} & \frac{-b}{\sqrt{a^2+b^2}} & 0 & 0 \\ \frac{b}{\sqrt{a^2+b^2}} & \frac{a}{\sqrt{a^2+b^2}} & 0 & 0 \\ 0 & 0 & \frac{c}{\sqrt{c^2+d^2}} & \frac{-d}{\sqrt{c^2+d^2}} \\ 0 & 0 & \frac{d}{\sqrt{c^2+d^2}} & \frac{c}{\sqrt{c^2+d^2}} \end{pmatrix} \begin{pmatrix} \sqrt{a^2+b^2} & 0 & -\sqrt{c^2+d^2} & 0 \\ 0 & \sqrt{a^2+b^2} & 0 & -\sqrt{c^2+d^2} \\ \sqrt{c^2+d^2} & 0 & \sqrt{a^2+b^2} & 0 \\ 0 & \sqrt{c^2+d^2} & 0 & \sqrt{a^2+b^2} \end{pmatrix},$$

$$U_2 = \begin{pmatrix} e^{i\theta_a} & 0 & 0 & 0 \\ 0 & e^{i\theta_b} & 0 & 0 \\ 0 & 0 & e^{i\theta_c} & 0 \\ 0 & 0 & 0 & e^{i\theta_d} \end{pmatrix} \quad \text{and} \quad U = U_2 U_1. \quad (14)$$

2.6 Quantum circuit depth

To simulate a toric code with length L , using local unitary gates requires at least linear size $O(L)$ depth circuits [3], and constant depth is achievable if measurement operations are allowed [4]. A recent work provided a systematic method to simulate an unknown toric code in $3L + 2$ depth [9, 1]. In comparison, on a $L \times L$ square lattice over a torus, we can simulate a known toric code state like $|00\rangle$ in $2L + 2$ depth and an unknown toric code Φ in $[2L + 2 + \log_2(d) + \frac{L}{2d}]$ depth. Here, the quantum circuit is local and the $CNOT$ gate is restricted to be applied on two qubits with a distance d .

To simulate the state $|00\rangle$, we initiate the process with ϕ_0 and designate the plaquette at the bottom right corner as redundant. Subsequently, we proceed the quantum gates step by step, following the instructions outlined on the left side of Figure 7. On the other hand, as elaborated upon in Section 2.5, an unknown toric code state Φ can be attained by substituting ϕ_0 with ϕ , which is obtained from two logical qubits through a sequence of CNOT gates. This procedure demands $\lceil \log_2(d) + \frac{L}{2d} \rceil$ steps², where d represents the

²For detailed discussion on the local CNOT gate, see Section C.

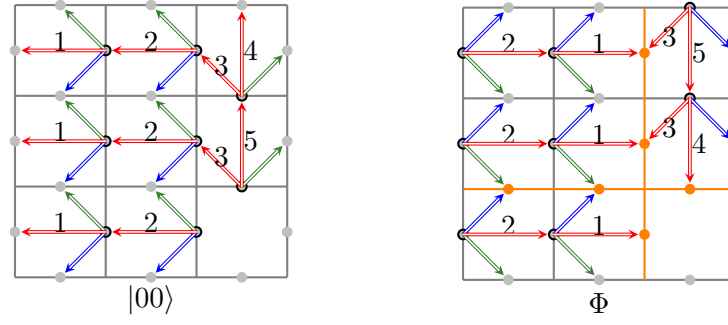


Figure 7: Opposite boundaries are identified and the case of $L=3$ is provided as an example. In both figures, the prescribed order for gate application is as follows: 1 Apply Hadamard gates to the qubits encircled by circles; 2 Execute CNOT gates, indicated by green arrows, followed by those with blue arrows; 3 Implement CNOT gates denoted by red arrows, following their numerical order. In the right figure, orange dots signify qubits that hold the information of Φ .

maximum distance between the two qubits that CNOT gate could apply without breaking locality. Additionally, a slight variation in the order, as demonstrated on the right side of Figure 7, is essential to initiate with ϕ .

3 Gluing method for 2D toric code

3.1 Gluing method for two single plaquettes

The method introduced in the preceding sections is efficient; however, it hinges on the selection of a suitable path. This choice could prove challenging for intricate surfaces. To address this concern, we propose a gluing method designed to overcome this complexity. To exemplify the essence of the gluing approach, we will commence with a straightforward example. To simulate the ground state of toric code on the two plaquettes in Figure 8, we can employ an ancilla qubit to partition it into two independent plaquettes p_1 and p_2 . The edges within $Bo(p_1)$ are denoted by 1, 2, 3, 4, while those within $Bo(p_2)$ are denoted by 5, 6, 7, 8. We initiate the process with ϕ_0 and ignore the overall normalization constant to simplify subsequent calculations.

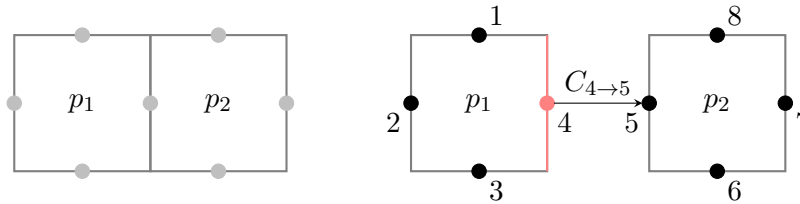


Figure 8: The lattice of two plaquettes is divided into two independent plaquettes by introducing the ancilla qubit in red.

First apply $1 + B_{p_1}$ and $1 + B_{p_2}$ independently to get

$$(1 + X_1 X_2 X_3 X_4)(1 + X_5 X_6 X_7 X_8)|00\dots 0\rangle. \quad (15)$$

Then apply $C_{4 \rightarrow 5}$ and notice this operator commutes with $1 + B_{p_2}$:

$$\begin{aligned} & \left(\frac{1 - Z_4}{2} X_5 + \frac{1 + Z_4}{2} \right) (1 + X_1 X_2 X_3 X_4) (1 + X_5 X_6 X_7 X_8) |00...0\rangle \\ &= (1 + X_1 X_2 X_3 X_4 X_5) (1 + X_5 X_6 X_7 X_8) |00...0\rangle. \end{aligned} \quad (16)$$

Finally, make a measurement over the ancilla qubit with basis $|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$ and $|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$. If we get $+1$, it is equivalent to applying $\frac{1+X_4}{2}$ and thus

$$\begin{aligned} & \frac{1 + X_4}{2} (1 + X_1 X_2 X_3 X_4 X_5) (1 + X_5 X_6 X_7 X_8) |00...0\rangle \\ &= \left(\frac{1 + X_4}{2} \right) (1 + X_1 X_2 X_3 X_5) (1 + X_5 X_6 X_7 X_8) |00...0\rangle. \end{aligned} \quad (17)$$

The ancilla qubit is disentangled, leaving us with the ground state of the two plaquettes. Observing that, when two boundaries e_i and e_j are glued together, all plaquettes terms commute with each other and $C_{i \rightarrow j}$ commutes with all plaquette terms except $1 + B_{p_k}$ where $e_i \in \text{Bo}(p_k)$. This observation underscores that the resultant combination remains a ground state even when multiple plaquettes are fused together concurrently.

On the other hand, if we get -1 , it is equivalent to applying $\frac{1-X_4}{2}$ and thus

$$\begin{aligned} & \frac{1 - X_4}{2} (1 + X_1 X_2 X_3 X_4 X_5) (1 + X_5 X_6 X_7 X_8) |00...0\rangle \\ &= \left(\frac{1 - X_4}{2} \right) (1 - X_1 X_2 X_3 X_5) (1 + X_5 X_6 X_7 X_8) |00...0\rangle, \end{aligned} \quad (18)$$

which is not the expected ground state. It is worth noting that the resulting state is an excited state if a magnetic charge exists at p_1 . Fortunately, we can correct it by applying Z_1 , Z_2 or Z_3 , each of which is a short dual ribbon operator. In the subsequent section, we will establish a proof demonstrating that a correcting operator invariably exists for any planar lattice.

Our method, can be naturally extended to more general scenarios where projectors only involve tensor products of Pauli X (given that tensor products of Pauli Z operators are automatically satisfied by the state $|00...0\rangle$), such as 3D toric model or X cube model to be addressed below. For instance, let us consider two edges from different lattices, labeled as m and n (note that we abuse the notation referring to both edges and lattices). The state of these two lattices can be expressed as $\prod \frac{1+H_m}{2} |0\rangle_m \otimes |0\rangle_{res_m}$ or $\prod \frac{1+H_n}{2} |0\rangle_n \otimes |0\rangle_{res_n}$. Here, $res_{m(n)}$ signifies the remaining system of lattice $m(n)$, and $H_{m(n)}$ denotes the projector onto lattice $m(n)$. Given that $C_{m \rightarrow n}$ only relies on $|m\rangle$, expanding the product of H_m yields

$$C_{m \rightarrow n} \sum_i (X \otimes 1 \otimes A_i + 1 \otimes 1 \otimes B_i) \prod \frac{1 + H_n}{2} |0\rangle_m |0\rangle_n |0\rangle_{res_m} |0\rangle_{res_n}, \quad (19)$$

where A_i and B_i acts only on res_m , and we have not expanded H_n since it has a trivial impact on m . Upon applying $C_{m \rightarrow n}$, we obtain

$$\sum_i (X \otimes X \otimes A_i + 1 \otimes 1 \otimes B_i) \prod \frac{1 + H_n}{2} |0\rangle_m |0\rangle_n |0\rangle_{res_m} |0\rangle_{res_n}. \quad (20)$$

Essentially, this signifies that the CNOT gate transfers all actions from m to n after disentangling m . Akin to Equation 17 and 18, we count the excitations and employ correction operators to obtain the ground state. Consequently, we can attain the expected ground state of the glued lattice by gluing the edges correspondingly, as long as the projectors consist of tensor products of Pauli X operators.

3.2 Gluing method for an arbitrary lattice

When transitioning from the gluing method's application on two single plaquettes to the broader context of numerous arbitrary plaquettes, our focus should not be on the edges measured $+1$, but rather on establishing a systematic method to correct address edges measured -1 .

In the instance presented in Figure 9, if we apply $C_{i \rightarrow j}$ to glue two boundaries and get -1 after measuring qubit e_i , the correcting operator must anti-commute with B_{p_i} and exhibit commutativity with everything else³. One intuitive approach is to apply a dual string operator starting at p_i and ending crossing a z -boundary.

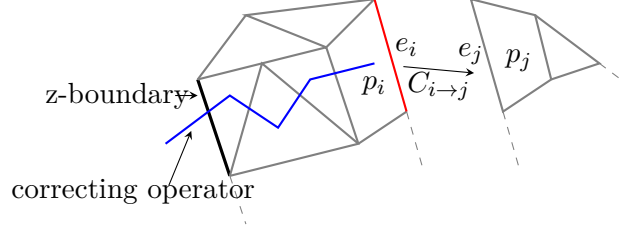


Figure 9: Correcting procedure after gluing two arbitrary plaquettes.

Expanding upon this notion, let us consider a situation involving any connected planar lattice $\gamma = \bigcup_{i=1}^n p_i$ with z -boundary e_0 . For a series $T = p_i$, $Bo(p_i) \cap \bigcup_{j=1}^{i-1} Bo(p_j) \neq \emptyset$ for any $i \in [2, n]$, we can insert ancilla qubits to separate T into multiple plaquettes and subsequently glue them back. To illustrate this idea, let us delve into an example consist of four plaquettes, as depicted in Figure 10.

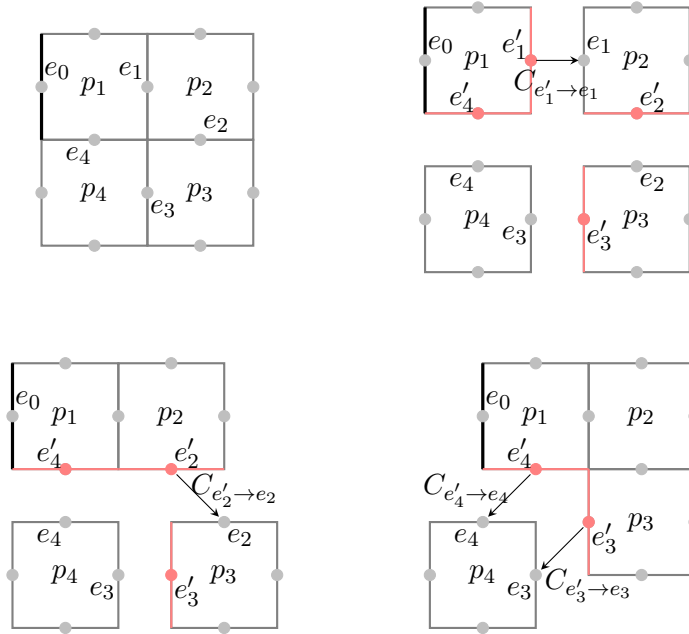


Figure 10: e_0 is a z -boundary and e' in red represents an ancilla qubit.

First, we initiate by utilizing ancilla qubits to fragment the lattice into single plaquettes.

³It is worth noting that this correcting operator effortlessly commutes with all vertex terms.

For $\tau(e_k)$ containing p_i and p_j , where $1 \leq i < j \leq n$ according to the series T , insert an ancilla edge e'_k into p_i while retaining e_k in p_j . Then we apply $1 + B_p$ to every single plaquette $p \in P$. Subsequently, glue them together piece by piece. For $p_i, 1 < i \leq n$, it becomes necessary to apply $C_{e' \rightarrow e}$ to all pairs of $e' \in \bigcup_{j=1}^{i-1} Bo(p_j)$ and $e \in Bo(p_i)$. Finally, we measure and disentangle e' . If we get -1, apply a dual string operator connecting p_i and the z-boundary of p_1 to correct it. It is noteworthy that all plaquettes can be glued simultaneously, allowing a dual string operator to annihilate two magnetic charges by connecting them. In this example, if we get -1 for e'_1 and e'_4 concurrently, the correcting operator will effectively nullify their impact.

In the case of a lattice without boundary, we can choose a specific plaquette p to be redundant, effectively transforming $Bo(p)$ into z-boundaries. Subsequently, this situation mirrors the scenario depicted in the lattice with boundaries, and further details are left for readers to explore. If the lattice solely contains x-boundaries, a viable solution is to consider the dual lattice of it. The process remains unchanged, except for the inversion of plaquette and vertex operators. Thus we can confidently assert that our method is universally capable of simulating the ground state of a toric code on any planar lattice configuration.

In the case of the 2D toric code, the gluing method might initially resemble a simple measurement process, especially when we break down the lattice into pieces, attach ancilla qubits, and then fuse them to obtain the ground state for the entire lattice. However, its capabilities extend significantly when we consider scenarios like 3D models, as discussed thoroughly in Section 4.3, or when we have two lattices in their ground states to be joined. In such cases, a stabilizer measurement can not glue two lattice and get the ground state of the glued lattice.

4 Simulate ground state for 3D models

4.1 3D toric model

The 3D toric model bears strong resemblance to the 2D toric code and is established on an arbitrary 3D lattice. To enhance clarity, a cubic lattice is adopted, as depicted in Figure 11. Within this lattice, V represents the set of all vertices, while P corresponds to the set of all plaquettes; each edge accommodates a single qubit. Moreover, for the sake of convenience, we have affixed labels to each edge, denoting them as x , y , or z based on their alignment with the respective axis (i.e., parallel to the x , y , or z axis). Notice this labeling maintains consistency even when applied to a 3-dimensional torus. The Hamiltonian is defined as

$$H = - \sum_{v \in V} A_v - \sum_{p \in P} B_p, \quad (21)$$

where A_v pertains to the application of the Pauli operator X over six edges connected to the vertex v , and B_p pertains to the application of the Pauli operator Z over the four edges encompassing the plaquette p . It is straightforward to see these new-defined A_v and B_p operators also satisfy $A_v^2 = B_p^2 = 1$ and $[A_v, B_p] = 0$. So this 3D toric Hamiltonian is equivalent to the equation expressed as a summation of local projectors. We get the ground state

$$|GS\rangle = \prod_v \frac{1 + A_v}{2} |\phi_0\rangle, \quad (22)$$

where $|\phi_0\rangle = |00\dots 0\rangle$, and we drop $\frac{1+B_p}{2}$ s since its action on $|\phi_0\rangle$ is +1. It is important to highlight that the constancy of ground state degeneracy endures with fluctuations in

system size, a pivotal characteristic of topological phases of matter. Additionally, Figure 11 presents a comprehensive depiction of a pair of conjugate logical operators. Notice that, the definition of logical operators only depends on the nontrivial loop or non-contractable planes. Consequently, we have three pairs of conjugate logical operators, each acting on edges labeled by x , y , or z respectively.

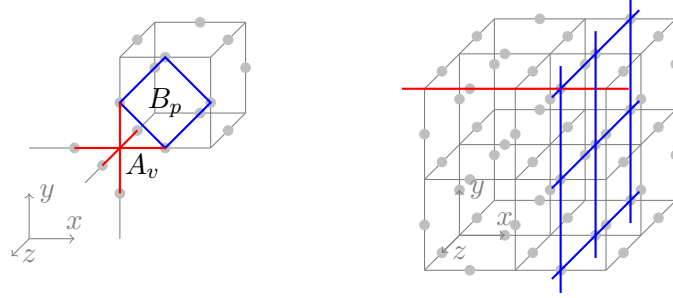


Figure 11: The left sub-figure illustrates the definitions of A_v and B_p operators. Meanwhile, the right sub-figure displays a pair of conjugate logical operators composed of edges labeled by x . The red string is a nontrivial circle parallel to x axis, and a logical Z operator is to apply Pauli Z over edges along the string. Conversely, the blue plane is a non-contractable plane perpendicular to x axis, and a logical X operator is to apply Pauli X over edges within the plane.

We can extend the method of 2D toric code to 3D toric model with boundary directly utilizing a plaquette as the basic structure. It is complicated yet straightforward, so its details are outlined in Section D. However, applying this approach to the 3D toric model without boundaries presents challenges, as the absence of free edges in the final step poses an issue. To circumvent this challenge, we must adopt a basic structure, illustrated in Figure 12. We still initiate the process with $|\phi_0\rangle$. Then we execute the quantum circuits as illustrated in the figure to achieve the action of $\frac{1+A_v}{2}$.

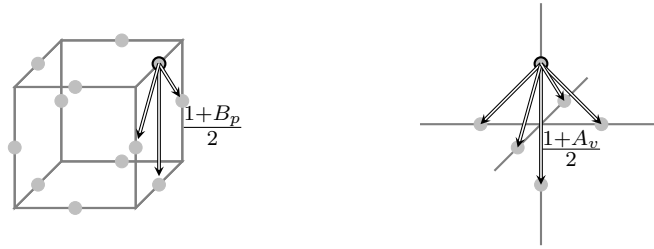


Figure 12: Comparison of two different basic structures: An example consisting of eight cubes with boundary is shown in Section D and a similar example without boundary is shown in Section E.

Using this basic structure to develop the lattice vertex by vertex, we will end with a redundant vertex as

$$\prod_{v \in V} A_v = 1. \quad (23)$$

In Section E, we present a straightforward example comprising eight cubes to illustrate the method. To address the general case, we delineate the procedures required for constructing a quantum circuit for the 3D toric model on an $L \times L \times L$ lattice over a 3-dimensional

torus in Figure 13. The process consists of several carefully orchestrated steps to efficiently realize the circuit, amounting to a total of $3L + 8$ steps. The quantum circuit is purely local since all applied quantum gates are either acting on a single qubit or on nearest two qubits. Certain non-interacting gates offer the potential for further parallelization, but this would only result in a constant difference in circuit depth.

In a manner akin to the procedure detailed in Section 2.5, we employ certain qubits to generate a particular initial state that encodes information about the logical qubits, as depicted in Figure 14. However, it is important to note that these selected qubits are unnecessary when we opt for free qubits during the ground state preparation. In conclusion, our method can prepare an arbitrary ground state of the 3D toric model with linear depth.

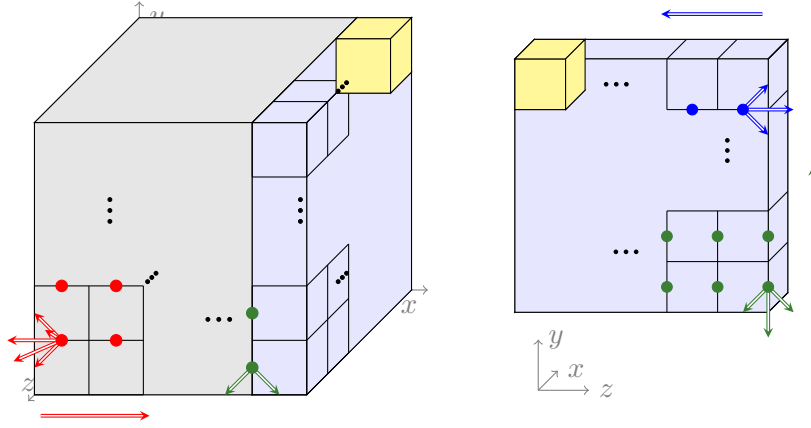


Figure 13: We initiate with slicing the 3D torus into layers along the x -direction and applying H gates to all the colored dots. Subsequently, we apply all CNOT gates from red dots to non-red dots simultaneously and between adjacent red dots layer by layer, which requires $L + 3$ steps. The last layer needs special treatment, which simplifies into a 2D problem after applying 2 CNOT gates from all green and blue dots. Further progression involves applying CNOT gates concurrently from green dots to non-green dots and between adjacent green dots row by row, necessitating $L + 1$ steps. Similarly, another $L + 1$ steps applying CNOT gates from blue dots completes the procedure and leaves a redundant vertex in the yellow cube.

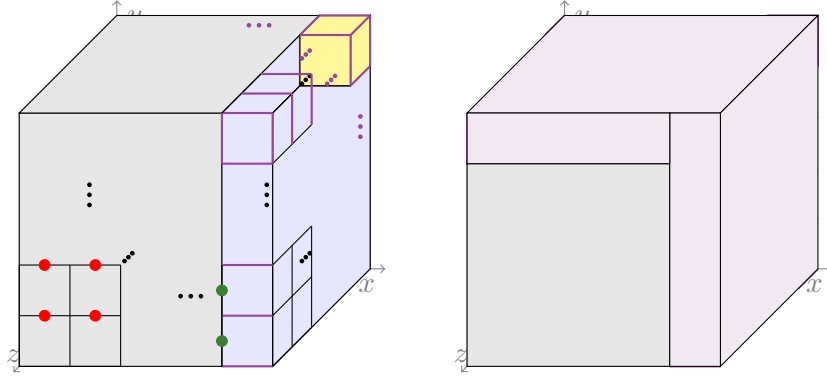


Figure 14: The left sub-figure highlights the employed qubits (indicated by purple edges) that are utilized for the specific initial state within the ground state preparation process. Meanwhile, the right sub-figure sketches all employed edges within a cubic lattice of size $L = 5$. All edges labeled by z in the back layer, x in the right layer, and y in the top layer are employed to encode arbitrary ground states. Importantly, the procedure we introduced earlier remains uninterrupted since we do not designate any of these edges as free qubits.

4.2 X-cube model

The X-cube model is a fracton model defined on a 3D cubic lattice, as visually depicted in Figure 15. Within this lattice, V represents the set of all vertices, while C corresponds to the set of all cubes; each edge accommodates a single qubit. For the sake of convenience, we also affixed labels to each edge, denoting them as x , y , or z based on their alignment with the respective axis. The Halmitonian is defined as

$$H = - \sum_{v \in V} A_v^x + A_v^y + A_v^z - \sum_{c \in C} B_c, \quad (24)$$

where $A_v^i, i = x, y, z$ is defined to implement Pauli operator Z across the four edges oriented vertically to the i axis and attached to vertex v , and B_c is designated to effectuate Pauli operator X across the twelve edges associated with cube c . Again these A_v^i s and B_c s operators satisfy $(A_v^i)^2 = B_c^2 = 1$ and $[A_v^i, B_c] = 0$. We get the ground state

$$|GS\rangle = \prod_c \frac{1+B_c}{2} |\phi_0\rangle, \quad (25)$$

where $|\phi_0\rangle = |00\dots 0\rangle$, and we drop $\frac{1+A_v^i}{2}$ since its action on $|\phi_0\rangle$ is $+1$. It is important to emphasize, however, that the ground state degeneracy experiences exponential growth alongside the system size. Additionally, Figure 15 presents a comprehensive depiction of a pair of logical operators of type W and T ⁴. Similarly, we have three types of logical operator pairs, each acting on edges labeled by x , y , or z respectively. Notably, distinct non-trivial loops exhibit identical homotopy while differ in terms of logical operators. This distinction is a crucial hallmark distinguishing the fracton model from conventional topological orders.

To simulate the ground state for the X-cube model, we outline⁵ the procedures required

⁴The complete set of logical operators are given in [15], but we only use two types of them, which are not conjugate to each other.

⁵We also present a straightforward example comprising eight cubes to illustrate the method, as elaborated in Section F.

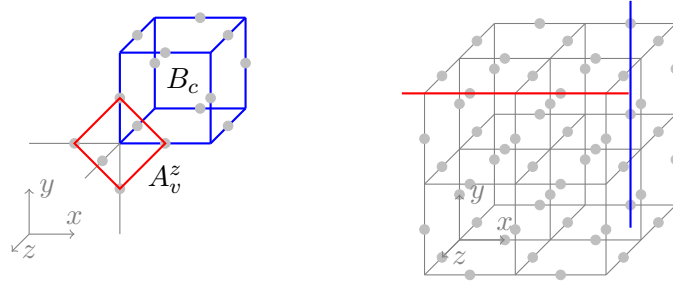


Figure 15: The left sub-figure illustrates the definitions of A_v and B_p operators. Meanwhile, the right sub-figure displays a pair of conjugate logical operators composed of edges labeled by x . The red string is of type W , a nontrivial circle parallel to x axis, and we apply Pauli X over edges along the string. Conversely, the blue string is of type T , a nontrivial circle perpendicular to x axis, and we apply Pauli Z over edges along the string.

to construct a quantum circuit for the X-cube model on an $L \times L \times L$ lattice, over a 3-dimensional torus, in Figure 16. The initial state is $|\phi_0\rangle$, and our target is to find quantum circuit to implement $\prod_c \frac{1+B_c}{2}$. We identify the redundancy by specifically selecting certain cubes, namely the three edges of the cubes in yellow, resulting from the requirement $\prod B_c = 1$ of the involved layer of cubes. Given that layers can be independently sliced in three distinct directions, this selective arrangement of yellow-colored structures⁶ is achieved. To start, we strategically partition the cube into distinct components: a central $(L-1) \times (L-1) \times (L-1)$ cube (colored gray), three $(L-1) \times (L-1) \times 1$ layers of cubes (colored blue), and three rows of redundant cubes (colored yellow). Then we further slice the central cube into $(L-1) \times (L-1) \times 1$ layers. Notice each gray and blue layer has the same boundaries up to rotation. This occurs because those yellow cubes are redundant, and the blue cubes are intended for the application of projectors in other layers. Neither of them interferes with the preparation of the layer structure. Consequently, we treat each layer of cubes as having the same structure, and their corresponding quantum circuits are outlined in Figure 16.

After the initial 9 steps, we apply CNOT from (i, j) to $(i-1, j)$ in the $(3k+10)$ -th step, apply CNOT from (i, j) to $(i, j-1)$ in the $(3k+11)$ -th step, and apply CNOT from (i, j) to $(i-1, j-1)$ in the $(3k+12)$ -th step, where $i+j = k+2$. This allows us to complete the layer structure in a total of $6L+6$ steps. These carefully orchestrated steps efficiently realize the circuit, requiring a total of $12L+11$ steps and the quantum circuit is purely local, similarly to the 3-dimensional toric code case. Certain non-interacting gates offer the potential for further parallelization, reducing the circuit depth by $2\lfloor \frac{2L-3}{2} \rfloor$. It is worth noting that this method can be readily extended to the X-cube model on a lattice of dimensions $L_1 \times L_2 \times L_3$.

The ground state degeneracy can be resolved by the complete set of logical operators [15]. We can readily attain all bases of the ground space of the X-cube model by replacing the initial product state, as demonstrated in Section 2.5 and 4.1. However, it is not straightforward to see whether our method can be applied to prepare arbitrary ground states of the X-cube model. The comprehensive encoding of arbitrary ground states still remains an open question and is left as a topic for future research directions.

⁶While there may exist more redundant cubes, our focus is solely on the chosen ones.

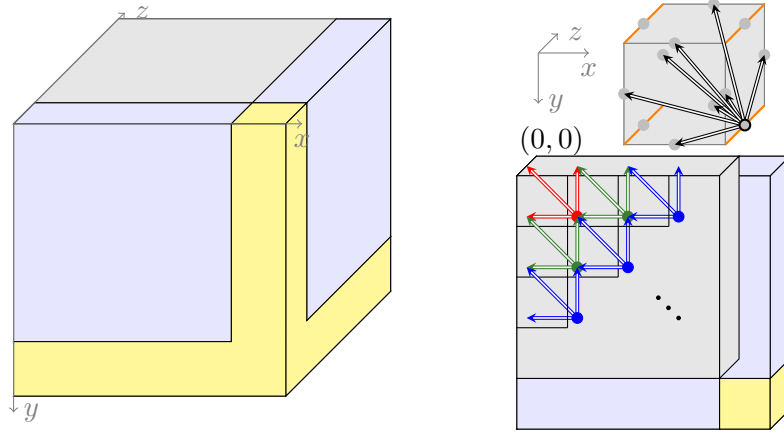


Figure 16: Treating each layer of cubes as the same structure with boundaries, we initiate H gates along the edges in the z -direction of each cube and apply CNOT gates from these edges to the others in the x and y directions, which requires 9 steps. Then we apply CNOT gates diagonally, row by row in different colors, necessitating $6L - 3$ steps. All gray layers are prepared simultaneously, followed by blue layers, leaving behind redundant cubes.

4.3 Gluing method for 3D models

Similar to the scenario in 2D toric code case, we can simulate the ground state of 3D toric model by breaking the lattice into basic structures, simulating on and gluing them back. This results in one redundant vertex term and the excitations are quasi-particles that are able to move freely. The situation is exactly the same as 2D toric code, so we can find correcting operators to annihilate all of the excitations, which is left to readers.

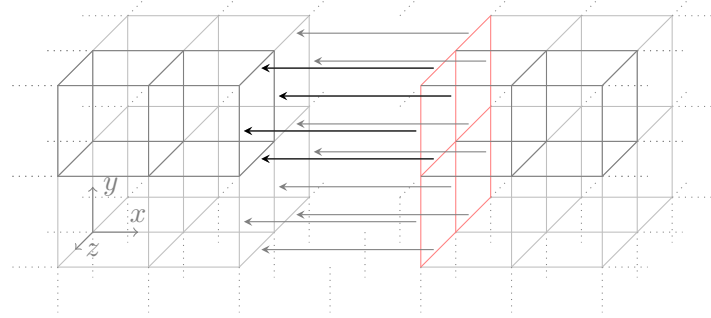


Figure 17: Following the preparation of ground states on the individual lattices, we designate all qubits on one side of the gluing plane as ancilla qubits (represented by red edges). Subsequently, we apply CNOT gates in parallel from these ancilla qubits to the opposite side. This process allows us to obtain the ground state of the fused lattice after appropriately disentangling the ancilla qubits.

Different methods for gluing in the X-cube model exist, and an intuitive one is shown in Figure 17. In this method, the quantum circuits are applied to each of the individual pieces to obtain their respective ground states. Subsequently, CNOT gates are employed along the gluing plane to glue them together. It is essential to note that this process is not a simple measurement, as each edge is influenced by two cube terms. Disentanglement necessitates the implementation of measurements on all red edges and correction operators to eliminate potential excitations based on the measurement outcomes. However, the X-

cube model poses greater complexity as the excitations are fractons. A systematic approach to find correcting operators is based on the following two facts:

1. There are three columns of redundant cubes as shown in Figure 16.
2. The excitation betraying cube terms is a fracton that are not able to move freely. While a *membrane operator* (see [13] for details) creates fractons on four corners of a rectangular.

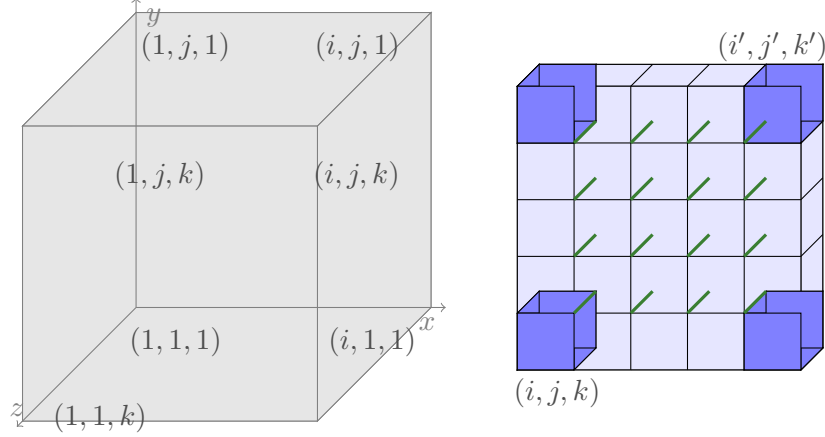


Figure 18: A membrane operator consisting of Z operators on green edges creates fractons at four corners; The correcting operator is a product of three membrane operators.

Illustrated in Figure 18, each cube in the n^3 cubic lattice, underlying the 3D torus topology, is assigned Cartesian coordinates (i, j, k) , where $1 \leq i, j, k \leq n$. Redundant cubes are positioned along three columns, namely $(i, 1, 1)$, $(1, i, 1)$, and $(1, 1, i)$ for $i = 1 \cdots n$. A membrane operator $\mathcal{M}[(i, j, k), (i', j', k')]$, consists of Z operators in the rectangle from (i, j, k) to (i', j', k') creates excitations at the four corners. For instance, when addressing an excitation at $(i, j, 1)$, applying $\mathcal{M}[(1, 1, 1), (i, j, 1)]$ leads to the annihilation of the excitation and the creation of excitations at redundant cubes, which are inconsequential. When dealing with a general excitation at (i, j, k) , with $i, j, k \neq 1$, a multi-step procedure comes into play. Initially, $\mathcal{M}[(1, j, 1), (i, j, k)]$ is applied to eliminate the excitation, generating three additional excitations at $(1, j, 1)$, $(1, j, k)$, and $(i, j, 1)$. Disregarding the one at the redundant cube, the other two are subsequently eliminated by $\mathcal{M}[(1, 1, 1), (1, j, k)]$ and $\mathcal{M}[(1, 1, 1), (i, j, 1)]$, respectively. In essence, the product operator $\mathcal{M}[(1, 1, 1), (1, j, k)] \mathcal{M}[(1, 1, 1), (i, j, 1)] \mathcal{M}[(1, j, 1), (i, j, k)]$ is capable of annihilating general excitations.

5 Conclusion and outlook

In this paper, we propose a method to prepare the ground state of a Hamiltonian consisting of local commuting projectors composed solely of Pauli X and Pauli Z operators. Our approach involves finding an appropriate initial state that serves as the ground state of these projectors and applying a quantum circuit composed solely of Clifford gates to achieve the Hamiltonian's ground state. We demonstrate the effectiveness of our method on 2D toric codes with various surface conditions, both with and without boundaries, as well as on the 3D toric model and the X-cube model. Our method enables the preparation of arbitrary ground states for 2D and 3D toric model with a linear-depth circuit, meeting the

lower bound for preparing ground states in topological phases. It also works for any basis of the ground state in the X-cube model using a linear-depth quantum circuit. We present these results on specific lattices, such as the 2D square lattice or 3D cubic lattice, and introduce a gluing method to facilitate ground state preparation on general 2D and 3D lattices. This gluing method provides a trade-off between measurement usage and circuit depth and can be applied to obtain the ground state of larger lattices by assembling ground states of smaller components.

There are several future directions to proceed from this work. One natural progression involves extending our method to other 3D models of interest. Furthermore, the applicability of our approach to the non-abelian Kitaev model presents a straightforward extension, offering the potential to broaden the scope of its application.

Acknowledgments.

The authors are partially supported by NSF CCF 2006667, Quantum Science Center (led by ORNL), and ARO MURI.

References

- [1] Miguel Aguado and Guifre Vidal. “Entanglement renormalization and topological order”. In: *Physical review letters* 100.7 (2008), p. 070404. DOI: <https://doi.org/10.1103/PhysRevLett.100.070404>.
- [2] Sergey Bravyi, Matthew B Hastings, and Spyridon Michalakis. “Topological quantum order: stability under local perturbations”. In: *Journal of mathematical physics* 51.9 (2010), p. 093512. DOI: <https://doi.org/10.1063/1.3490195>.
- [3] Sergey Bravyi, Matthew B Hastings, and Frank Verstraete. “Lieb-Robinson bounds and the generation of correlations and topological quantum order”. In: *Physical review letters* 97.5 (2006), p. 050401. DOI: <https://doi.org/10.1103/PhysRevLett.97.050401>.
- [4] Sergey Bravyi et al. “Adaptive constant-depth circuits for manipulating non-Abelian anyons”. In: *arXiv:2205.01933* (2022). DOI: <https://doi.org/10.48550/arXiv.2205.01933>.
- [5] Sergey B Bravyi and A Yu Kitaev. “Quantum codes on a lattice with boundary”. In: *arXiv preprint quant-ph/9811052* (1998). DOI: <https://doi.org/10.48550/arXiv.quant-ph/9811052>.
- [6] Eric Dennis et al. “Topological quantum memory”. In: *Journal of Mathematical Physics* 43.9 (2002), pp. 4452–4505. DOI: <https://doi.org/10.1063/1.1499754>.
- [7] Sepehr Ebadi et al. “Quantum phases of matter on a 256-atom programmable quantum simulator”. In: *Nature* 595.7866 (2021), pp. 227–232. DOI: <https://doi.org/10.1038/s41586-021-03582-4>.
- [8] Jeongwan Haah. “Local stabilizer codes in three dimensions without string logical operators”. In: *Physical Review A* 83.4 (2011), p. 042330. DOI: <https://doi.org/10.1103/PhysRevA.83.042330>.
- [9] Oscar Higgott et al. “Optimal local unitary encoding circuits for the surface code”. In: *Quantum* 5 (2021), p. 517. DOI: <https://doi.org/10.22331/q-2021-08-05-517>.
- [10] A Yu Kitaev. “Fault-tolerant quantum computation by anyons”. In: *Annals of Physics* 303.1 (2003), pp. 2–30. DOI: [https://doi.org/10.1016/S0003-4916\(02\)00018-0](https://doi.org/10.1016/S0003-4916(02)00018-0).

- [11] Michael A Levin and Xiao-Gang Wen. “String-net condensation: A physical mechanism for topological phases”. In: *Physical Review B* 71.4 (2005), p. 045110. DOI: <https://doi.org/10.1103/PhysRevB.71.045110>.
- [12] Yu-Jie Liu et al. “Methods for simulating string-net states and anyons on a digital quantum computer”. In: *arXiv:2110.02020* (2021). DOI: <https://doi.org/10.1103/PRXQuantum.3.040315>.
- [13] Abhinav Prem, Jeongwan Haah, and Rahul Nandkishore. “Glassy quantum dynamics in translation invariant fracton models”. In: *Physical Review B* 95.15 (2017), p. 155133. DOI: <https://doi.org/10.1103/PhysRevB.95.155133>.
- [14] KJ Satzinger et al. “Realizing topologically ordered states on a quantum processor”. In: *Science* 374.6572 (2021), pp. 1237–1241. DOI: <https://doi.org/10.1126/science.abi8378>.
- [15] Kevin Slagle and Yong Baek Kim. “Quantum field theory of X-cube fracton topological order and robust degeneracy from geometry”. In: *Physical Review B* 96.19 (2017), p. 195139. DOI: <https://doi.org/10.1103/PhysRevB.96.195139>.
- [16] Nathanan Tantivasadakarn, Ruben Verresen, and Ashvin Vishwanath. “The Shortest Route to Non-Abelian Topological Order on a Quantum Processor”. In: *arXiv:2209.03964* (2022). DOI: <https://doi.org/10.1103/PhysRevLett.131.060405>.
- [17] Nathanan Tantivasadakarn, Ashvin Vishwanath, and Ruben Verresen. “A hierarchy of topological order from finite-depth unitaries, measurement and feedforward”. In: *arXiv:2209.06202* (2022). DOI: <https://doi.org/10.1103/PRXQuantum.4.020339>.
- [18] Nathanan Tantivasadakarn et al. “Long-range entanglement from measuring symmetry-protected topological phases”. In: *arXiv:2112.01519* (2021). DOI: <https://doi.org/10.48550/arXiv.2112.01519>.
- [19] Ruben Verresen, Mikhail D Lukin, and Ashvin Vishwanath. “Prediction of toric code topological order from Rydberg blockade”. In: *Physical Review X* 11.3 (2021), p. 031005. DOI: <https://doi.org/10.1103/PhysRevX.11.031005>.
- [20] Ruben Verresen, Nathanan Tantivasadakarn, and Ashvin Vishwanath. “Efficiently preparing Schrödinger’s cat, fractons and non-Abelian topological order in quantum devices”. In: *arXiv:2112.03061* (2021). DOI: <https://doi.org/10.48550/arXiv.2112.03061>.
- [21] Sagar Vijay, Jeongwan Haah, and Liang Fu. “A new kind of topological quantum order: A dimensional hierarchy of quasiparticles built from stationary excitations”. In: *Physical Review B* 92.23 (2015), p. 235136. DOI: <https://doi.org/10.1103/PhysRevB.92.235136>.
- [22] Sagar Vijay, Jeongwan Haah, and Liang Fu. “Fracton topological order, generalized lattice gauge theory, and duality”. In: *Physical Review B* 94.23 (2016), p. 235157. DOI: <https://doi.org/10.1103/PhysRevB.94.235157>.
- [23] Kevin Walker and Zhenghan Wang. “ $(3+1)$ -TQFTs and topological insulators”. In: *Frontiers of Physics* 7.2 (2012), pp. 150–159. DOI: <https://doi.org/10.1007/s11467-011-0194-z>.

A 2D toric code on sphere

Similar with the example of genus 1 torus, we identify different qubit pairs to change the four plaquettes into a sphere as shown in Figure 19. The bottom right plaquette is chosen to be redundant and two steps will complete the procedure.

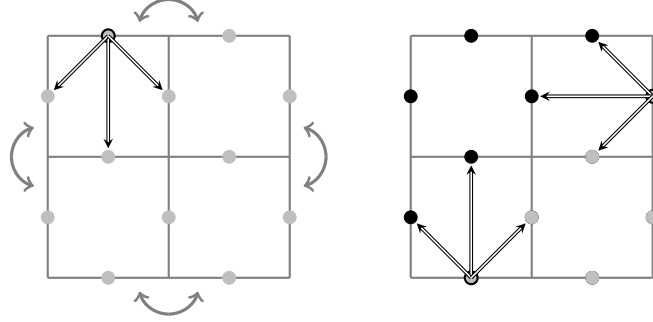


Figure 19: Boundary edges are identified according to the double-headed arrows.

B 2D toric code on genus n surface

Figure 20 shows a genus n surface which is a disk enclosed by a ribbon with identified edges. Beginning with $|\phi_0\rangle$, we develop a disk from inside and leave the ribbon with all identified edges undeveloped. Then we choose one edge in the ribbon to apply the method of basic structure and repeat in clockwise direction. After $2n - 1$ steps for a genus n torus, we will get the ground state of the closed surface.

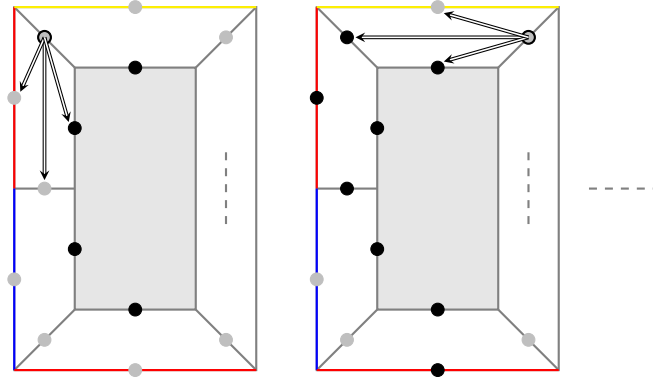


Figure 20: The shaded area represents the developed disk; Boundaries with the same color are identified to change the plaquettes into a genus n torus.

C Local CNOT operation

In the preparation of arbitrary state of 2D toric code, we use *CNOT* to transmit the logical states vertically and horizontally. If we employ non-local CNOT gates, as illustrated on the left side of Figure 21, it takes $\lceil \log_2(L) \rceil$ steps. However, when CNOT gates are constrained to constant distances d , this procedure requires $\lceil \log_2(d) + \frac{L}{2d} \rceil$ steps, as shown on the right side of Figure 21. The distance d is defined such that two qubits are considered to be d apart if the shortest path connecting them contains $d - 1$ qubits.

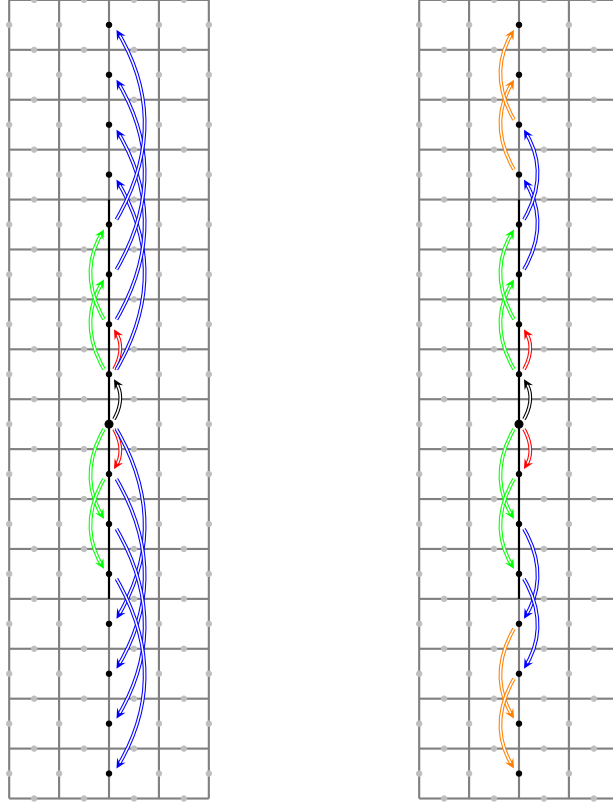


Figure 21: For the case $L = 16$, we illustrate the utilization of CNOT gates to vertically transmit the logical states in the sequence: black, red, green, blue, orange. On the left-hand side, there exists no constraint on the distance d , permitting the use of non-local CNOT gates, resulting in $\log_2(16) = 4$ steps. On the right-hand side, with the restriction of $d = 2$, the process requires $\log_2(2) + \frac{16}{4} = 5$ steps.

D 3D toric model with boundary

The generation from 2D toric code to 3D toric model with boundary is complicated but direct. We can continue to use a plaquette as the basic structure but consider four different types of cubes. Let us take the eight cubes in Figure 22 as an example. We begin with the red cube and develop it into pink cubes. Orange cubes are the next and the yellow cube completes the model. In the following, we will divide the method into four steps, each step describes one type of cubes.

To develop the qubits in the beginning red cube, we need to develop five rather than six faces as the cube is a closed surface with one redundant face. As shown in Figure 23, we develop a face first and choose the four qubits on the opposite face to repeat the basic structure. After that, considering the pink cube shares a face with developed cube, we only need to develop four more faces as the second cube is also a closed surface. We choose the four qubits on the face opposite to the developed cube to repeat the basic structure.

Similarly, we need to develop three faces for the orange cubes and two faces for the yellow cube as shown in Figure 24. The four steps complete the procedure to simulate the ground state of toric model on the eight cubes lattices. And we are able to develop any size cubes with boundary using the method described above.

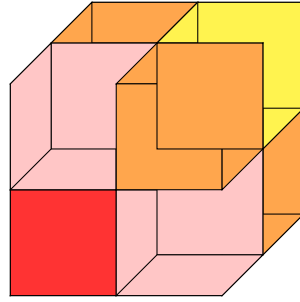


Figure 22: The beginning cube is colored red. The pink, orange and yellow cube represent the cubes connected with one, two or three faces developed.

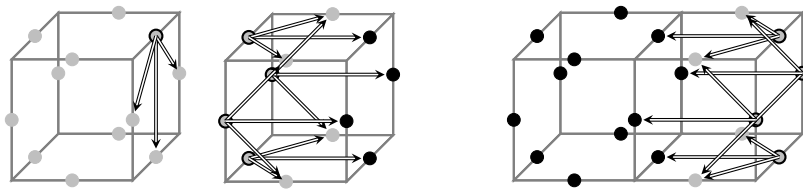


Figure 23: The left two cubes describe the first step to develop the red cube. The right cubes describes the second step to develop the pink cube.

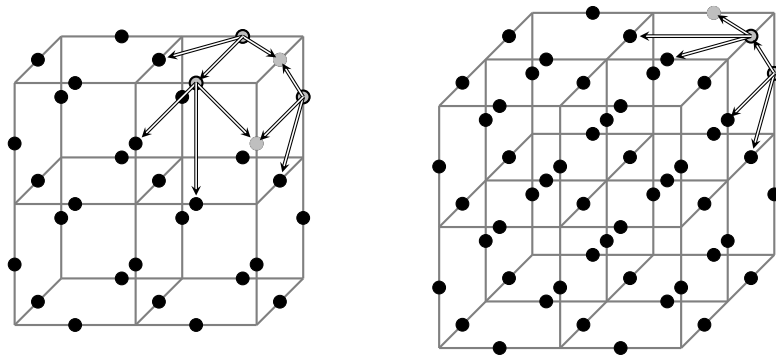


Figure 24: The left figure describes the step of orange cubes, and we need to develop the face in front first. The right figure describes the final step to develop the yellow cube, and we need to develop the face above first.

E 3D toric model without boundary

In Figure 25, the opposite faces are identified together to represent the 3D torus. We begin with $|\phi_0\rangle$ and choose four free qubits in the lower layer to take the procedure in basic structure. After this step and identification of opposite faces, we get the lattice with the middle untouched. Finally, choose three more free qubits to repeat the basic structure and leave a vertex redundant.

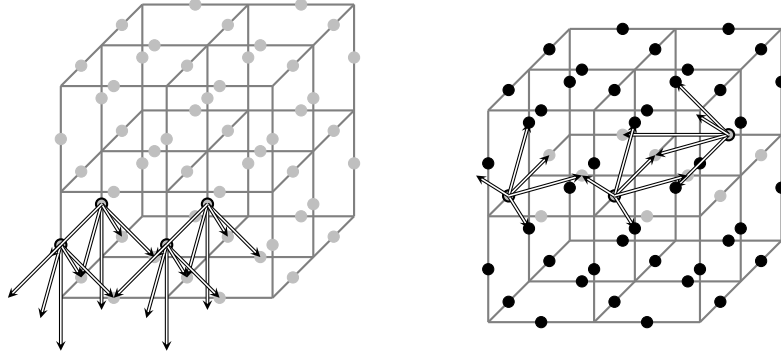


Figure 25: A qubit $|0\rangle$ is placed at each gray dot at the beginning. The color changes to black when a quantum gate is applied on the qubit.

F X-cube model simple example

To illustrate the method, we take the eight cubes case as a simple example shown in Figure 26. Considering the redundant cubes in yellow, we only need to develop four cubes left. The initial state is $|\phi_0\rangle$, and we begin with the cube at the right front higher corner to apply the basic structure. After this step and identifying opposite faces, we get the result on the right-hand side of Figure 26. Then we choose three more free qubits from each cube connecting with the developed cube to repeat the procedure of basic structure and the ground state is completed.

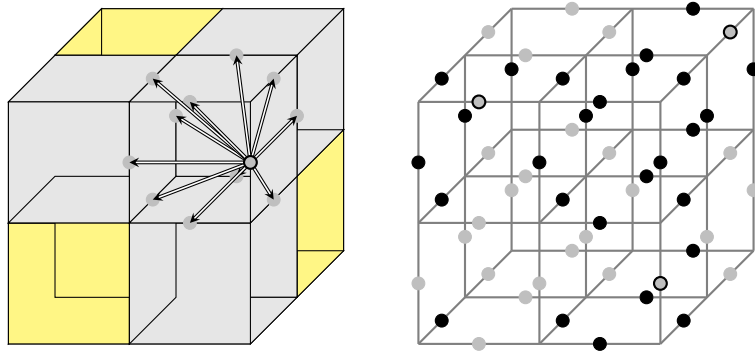


Figure 26: The left figure is an example of X-cube model with opposite faces identified. The right figure shows the result after the first step and the free qubits for next step are circled.