ABSTRACT

Title of Dissertation: Excursions at the Interface of Topological Phases
of Matter and Quantum Error Correction

Ali Lavasani
Doctor of Philosophy, 2022

Dissertation Directed by: Professor Maissam Barkeshli
Department of Physics

Topological quantum error-correcting codes are a family of stabilizer codes that are built using a lattice of qubits covering some manifold. The stabilizers of the code are local with respect to the underlying lattice, and logical information is encoded in the non-local degrees of freedom. The locality of stabilizers in these codes makes them especially suitable for experiments. From the condensed matter perspective, their code space corresponds to the ground state subspace of a local Hamiltonian belonging to a non-trivial topological phase of matter. The stabilizers of the code correspond to the Hamiltonian terms, and errors can be thought of as excitations above the ground state subspace. Conversely, one can use fixed point Hamiltonian of a topological phase of matter to define a topological quantum error-correcting code.

This close connection has motivated numerous studies which utilize insights from one viewpoint to address questions in the other. This thesis further explores the possibilities in this direction. In the first two chapters, we present novel schemes to implement logical gates, which are motivated by viewing topological quantum error-correcting codes as topological phases of

matter. In the third chapter, we show how the quantum error correction perspective could be used to realize robust topological entanglement phases in monitored random quantum circuits. And in the last chapter, we explore the possibility of extending this connection beyond topological quantum error-correcting codes. In particular, we introduce an order parameter for detecting k-local non-trivial states, which can be thought of as a generalization of topological states that includes codewords of any quantum error-correcting code.

Excursions at the Interface of Topological Phases of Matter and Quantum Error Correction

by

Ali Lavasani

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2022

Advisory Committee:
  Professor Maissam Barkeshli, Chair/Advisor
  Professor Sankar Das Sarma
  Professor Jay Deep Sau
  Professor Michael Gullans
  Professor Mohammad Hafezi

Acknowledgments

My time as a graduate student at UMD makes for almost twenty percent of my whole lifetime so far. I think it was an amazing experience, and I am grateful to all those who shaped it through their friendship, support, and mentorship. Although its share of my lifetime will decrease as I move forward through life, I hope its cherished memories will always remain as vivid as they are today.

I would like to thank my advisor, Professor Maissam Barkeshli, for his mentorship and constant support. Each and every project that we worked on was truly insightful and fun for me. I am grateful that his door was always open for any question. He always listened patiently to my half-baked ideas and provided valuable insights on how to move forward. Our discussions invariably left me with a deeper understanding of the problem I was working on.

I would also like to thank Professor Sankar Das Sarma, Professor Mohammad Hafezi, Professor Michael Gullans, and Professor Sagar Vijay for giving me the opportunity to work with them on various interesting projects. I should also thank the wonderful post-docs and graduate students who I have had the chance to collaborate with during my PhD: Yahya Alavirad, Guanyu Zhu, Daniel Bulmash, Hossein Dehghani and Zhu-Xi Luo.

I am thankful to Professor Sankar Das Sarma, Professor Jay Sau, Professor Michael Gullans, and Professor Mohammad Hafezi for agreeing to serve on my thesis committee.

I was privileged to visit the Kavli Institute for Theoretical Physics (KITP) for four months during my Ph.D. as a part of the KITP graduate fellowship program. It was by far the best four contiguous months of my Ph.D. It allowed me to meet many brilliant physicists and engage in several fruitful discussions. I would like to particularly thank Professor Matthew Fisher, Professor Sagar Vijay, Izabella Lovas, Oguzhan Can, David Long, and Zhu-Xi Luo for multiple insightful discussions during my visit.

I am deeply indebted to many fellow graduate students and post-docs at UMD and other Institutions for numerous valuable discussions: Yahya Alavirad, Kasra Hejazi, Alireza Seif, Ali Izadi Rad, Ali Hamed Moosavian, Alireza Parhizkar, Batoul Banihashemi, Masoud Mohammadi, Reza Ebadi, Iman Ahmadabadi, Majid Ekhterachian, Amir Tajdini, Reza Javadi, Hadi Vafaei, Fatholah Salehi, Chun-Xiao Liu, Jedediah Pixley, Su-Kuan Chu, Naren Manjunath, Srivasta Tata, Subhayan Sahu, Fangli Liu, Christopher White, Zhicheng Yang, Abhinav Deshpande, Michael Flynn, Mykhaylo Usatyuk, Bob Knighton and many more who I forgot to name here. I would like to specially thank Yahya Alavirad who was also my closest friend in the graduate school with his desk just three feet away from me.

Last but not least, I would like to thank my family.

Table of Contents

Chapter 1: Introduction

Condensed Matter physics, or at least a fair share of it, is about applying quantum mechanics to understand the physics of *many-body* systems, and it covers many different topics such as the study of fermi-liquids, superconductivity, magnetism, superfluidity, etc. Given the long history of the field, numerous tools and concepts have been developed and perfected over the years to arrive at a better understanding of the relevant underlying many-body physics in these systems.

On the other hand, quantum information utilizes quantum mechanics for information processing tasks like communication and computation. Here, the system of interest is a set of qubits, and ideally, one has complete control to measure and manipulate individual qubits. This naturally leads to the study of *few-body* quantum systems as the main building blocks of the information processing protocols.

Nevertheless, to completely harness the power of quantum mechanics and accomplish tasks beyond the reach of classical protocols, one needs to maintain a high level of control over a large number of qubits. Given the extreme fragility of quantum coherence against noise and perturbation, this is practically impossible unless a systematic mechanism exists to overcome decoherence. Quantum error correction has been developed to achieve just that. Roughly speaking, quantum error-correcting codes protect quantum information against local noise by encoding them in the many-body non-local degrees of freedom. Therefore, it would be fair to say that quantum error correction is fundamentally *a many-body phenomenon*. It is then natural to expect to find close connections between condensed matter physics and quantum error correction.

Topological phases of matter provide a clear example of such a connection. From the condensed matter perspective, topological phases are characterized by properties that are invariant under continuous local deformations of their corresponding Hamiltonian. This robustness against local noise is also the defining feature of quantum error-correcting codes. Thus one may view quantum error-correcting codes and topological phases of matter as two different approaches to the same problem. For example, the Kitaev toric code model [1, 2], which is the archetype of the bosonic topological phase, is probably the most studied quantum error-correcting code in the quantum information literature where it is better known as the surface code [3, 4].

In general, topological phases on non-trivial manifolds like a torus exhibit robust ground state degeneracy, meaning a degeneracy that local perturbations cannot lift. Since different ground states look the same locally and only differ in long-range entanglement patterns, local operators cannot distinguish between them. As such, all matrix elements of any local operator between different ground states are zero [2]. If we treat ground states as codewords of a quantum error-correcting code and local perturbations as errors, this statement basically becomes the Knill-Laflamme quantum error correction condition [5]. This shows that topological phases can be used to define quantum error-correcting codes [6]. Such codes are called topological quantum error-correcting codes.

Various aspects of the connection between topological error-correcting codes and topological phases of matter have already been studied. We list a few in the following. The relation between quantum error correction condition and the stability of topological phases has been studied in Refs. [7–10]. Many studies are focused on finding more efficient quantum error-correcting codes by considering various topological phases on manifolds with different boundary conditions [3,4,11–17]. It is also possible to consider the closely related symmetry-protected topological phases of matter for quantum error correction and quantum computation [18–20]. Another active field of research is finding novel and efficient ways for manipulating logical information encoded in topological quantum error-correcting codes. Code deformation is the idea of performing logical operations in the code space via deformations of the manifold on which the code is defined [3,21–28]. Closely related is the notion of topological quantum computation [2,29,30] where one performs computation via braiding of anyons which are the localized excitations of 2D topologically ordered systems. Studying the relations between error thresholds in topological error-correcting codes and phase transitions in statistical models is also another interesting direction of related research [3,31,32]. Topological phases in higher dimensions have also been studied as potential candidates for self-correcting quantum memories [33,34].

This thesis further explores the interplay of topological phases and quantum error-correcting codes. We have organized our results into four chapters, and each chapter contains an independent topic and can be read on its own.

In Chapter 2 we show how the measurement of carefully chosen string operators can be utilized to perform Clifford gates on different variants of the surface code. This protocol is based on the notion of topological charge measurement in topological phases of matter. An advantage of this method is that it is quite general and can be used in different encoding schemes. We also show how this scheme allows for Clifford gates to be applied on distant logical qubits without bringing them next to each other. It is worth noting that, provided with a supply of specially prepared ancillas known as magic states, Clifford circuits can be used for universal quantum computation. This chapter is based on Ref. [35]

In Chapter 3 we present constant depth quantum circuits to perform a particular type of code deformation known as Dehn twist in topological quantum error-correcting codes defined on manifolds with hyperbolic geometry. The main reason to use hyperbolic geometry is to obtain topological quantum error-correcting codes with a constant encoding rate. Dehn twists correspond to certain unitary operations in the code space of a topological quantum error-correcting code. For certain topological orders, Dehn twists make up a universal gate set, meaning that they can efficiently approximate any other unitary gate. The result of this chapter is based on Ref. [36]. This chapter generalizes the result of Refs. [37,38] which pertains to codes defined on a manifold with Euclidean geometry.

In Chapter 4 we consider emergent topological order in monitored random circuits. Random unitary circuits are often used to model generic unitary evolution, which causes a system to thermalize, resulting in volume law scaling of entanglement entropy for small enough subsystems (compared to the rest of the system). Monitored random unitary circuits are random unitary circuits intercepted by rapid local measurements. Given that measurements can destroy entanglement between non-local degrees of freedom while local unitaries can only generate local entanglement, one would expect that the volume law entanglement vanishes in monitored random circuits, meaning that any finite density of measurements would result in an area law entangled steady state. However, it was discovered that random unitaries could act as encoding circuits of a quantum error-correcting code, which hides a part of non-local entanglement from local measurements. Hence such circuits can sustain volume law entanglement up to some finite non-zero critical measurement rate. As the measurement rate is tuned above the critical rate, the system undergoes an entanglement phase transition to an area law phase where the steady-state looks like a product state. Here, we consider families of monitored random circuits which exhibit emergent topological order. Our circuit model includes multi-qubit measurements, which protect long-range entanglement from local measurements and random unitaries. The multi-qubit measurements are chosen carefully to stabilize the code space of a topological quantum error-correcting code. This chapter is based on Refs. [39, 40].

And Finally, in Chapter 5, we consider the possibility of defining k-local phases of matter. It has been shown that topological phases can be used to construct quantum error-correcting codes. However, not all quantum error-correcting codes correspond to a topological phase. Indeed, many quantum error-correcting codes have algebraic constructions without any reference to a particular geometry. In contrast, topological order is highly dependent on the geometry and space dimension. Given that such codes still satisfy the Knill-Laflamme quantum error correction condition, it is reasonable to ask whether they correspond to some generalized notion of the topological phase. If it is the case, then an order parameter should exist that distinguishes a trivial state from a codeword of a quantum error-correcting code. In Chapter 5, we make progress towards finding such an order parameter. We present a proof of principle based on the notion of irreducible multi-partite correlations. This chapter is based on Ref. [41]

Chapter 2: Clifford gates from joint measurements in variants of surface code

A crucial pillar of universal fault-tolerant quantum computation is the ability to perform quantum error correction [42, 43]. Schematically, given a physical qubit with an error probability p, a quantum error correcting code allows one to reach a target error rate for a logical qubit with error probability $p_{\text{fail}} \sim (p/p_{th})^{d/2}$, where d is the code distance and p_{th} is the error threshold of the code [44]. It is therefore desirable to implement a code which maximizes d and p_{th} to the extent possible for a given set of physical resources. Codes that allow d to be arbitrarily large while maintaining local interactions between the physical qubits are *topological error correcting codes*, which utilize the physics of topological states of matter [2,30,45]. In topological error correcting codes on the Euclidean plane with local interactions, the ratio of the number of physical qubits N_{phys} to the number of logical qubits N_L scales as $N_{phys}/N_L = \mathcal{O}(d^2)$ [46].

The simplest topological error correcting code is known as the $\mathbb{Z}_2$ surface code [3, 4, 11], and possesses a relatively high error threshold; for certain error models the error threshold is quoted to be $p_{th} \sim 1\%$ [47]. Given the rapid experimental advances in qubit technology using various physical platforms, it is reasonable to expect that the $\mathbb{Z}_2$ surface code will play an important role in near-term demonstrations of fault-tolerance. Indeed, there are already preliminary demonstrations of quantum error correction using small surface codes [48–50] as well as limited logical operations on them [51, 52]. Closely related error correcting codes are the color codes [13] and hyperbolic codes [27,53,54]. The color code is effectively two independent copies of the $\mathbb{Z}_2$ surface code [55]; while it has a lower error threshold [56], it allows for transversal implementation of Clifford gates and improves the space-time overhead [24]. The hyperbolic codes are related to the $\mathbb{Z}_2$ surface code on a tiling of hyperbolic space; they allow one to improve the scaling of the ratio N_{phys}/N_L to be independent of d, at the cost of requiring non-local interactions [54].

As we review below, logical qubits can be encoded in the surface code in a number of different ways: through (1) boundary defects, which are domain walls between alternating boundary conditions, (2) holes, or (3) bulk twist defects. Hybrid approaches that combine any or all of the above are also possible.

The set of fault-tolerant logical operations that can be performed using the $\mathbb{Z}_2$ surface code form the Clifford group. In addition to Pauli operations on single qubits, this group is generated by the single qubit Hadamard gate H, phase gate S, and two-qubit CNOT gate. A variety of methods are known for implementing these gates in the $\mathbb{Z}_2$ surface code, however they depend sensitively on the encoding scheme [3,4,23,25,26,28,57–59].

In the context of topological quantum computation, an approach has been developed based on the idea of topological charge measurements. [60,61] to implement logical unitary gates. In particular, Ref. [61] demonstrated that topological charge measurements along certain 'graph' operators could in principle be utilized to implement non-trivial fault-tolerant logical unitary gates (see also Ref. [62]).

In this chapter, we demonstrate how the idea of topological charge measurements in $\mathbb{Z}_2$ surface codes can be utilized to devise low overhead fault-tolerant procedures that implement the full Clifford group. Notably, this approach is not limited to any specific scheme and allows one to implement single qubit Clifford gates and long range CNOT gates in all variants of the surface code, with minimal overhead.

We further apply our methods to both color codes and hyperbolic codes. In the context of color codes, we also propose novel efficient methods for implementing Clifford group operations in a hole based encoding scheme, which provides some advantages over alternate proposals [13, 24, 63].

We note that to obtain universal fault-tolerant quantum computation, the Clifford group must be supplemented with an additional gate, such as the single qubit $\pi/8$ phase gate. In the codes that we study in this chapter, this gate inevitably requires magic state injection and distillation. In this chapter we focus on efficient fault-tolerant implementations of gates in the Clifford group, and do not further consider the $\pi/8$ phase gate.

The rest of this chapter is organized as follows. In Section 2.1 we provide a review of active error correction with the surface code, together with a brief review of the different encoding schemes and proposals for carrying out quantum computation with them. In Section 2.2, we explain the abstract joint measurement circuits that allow implementation of the full Clifford group. In Section 2.3, we demonstrate how to implement these measurement circuits in the surface code using a twist defect logical ancilla. In Sections 2.6 and 2.7 we further apply these results to hyperbolic and color codes.

2.1 Review of logical qubit encodings and Clifford gates in surface code

We begin with a brief review of the various proposals [3,4,23,57,59] for quantum computing with the surface code.

2.1.1 Planar encoding

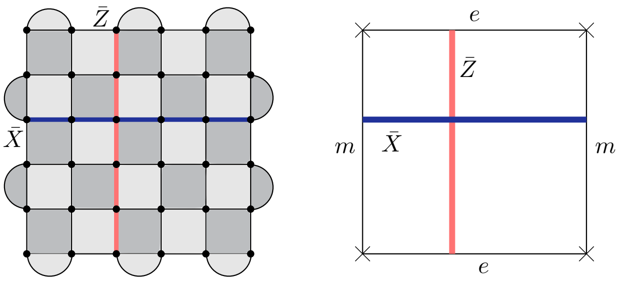
The simplest type of surface code is the planar code based on boundary defects. [3,11,12] We consider a physical qubit at each site of a square lattice, as shown in Fig. 2.1a. Each plaquette p is associated with a stabilizer S_p, with dark plaquettes representing X stabilizers and light plaquettes representing Z stabilizers:

$$S_p = \prod_{i \in \partial p} \sigma_i, \qquad \sigma = \begin{cases} X, & \text{if } p \text{ is dark} \\ Z, & \text{if } p \text{ is light,} \end{cases} \tag{2.1}$$

where ∂p denotes the boundary of the p plaquette. In the bulk, the stabilizers have support on four physical qubits. Violations of X-type stabilizers are referred to as e particles, and violations of Z-type stabilizers are referred to as m particles. Local operators in the bulk can only create e particles in pairs, and similarly for m particles.

On the boundary, the stabilizers, shown as semicircles in Fig. 2.1a, involve two physical qubits. An edge with only Z type stabilizers is referred to as an e boundary, because applying a Z operator on an edge qubit can create a single e particle; therefore, the e particles are 'condensed' on such an edge. Similarly, an edge with only X type stabilizers is referred to as an m boundary (see Fig. 2.1a). To avoid drawing the entire lattice, we use schematic diagrams whenever possible, as shown in Fig. 2.1b. The crosses on the edges, which are domain walls between the two types of boundaries, are referred to as boundary defects.

The stabilizers all commute with each other. The code space $\mathcal{C}$ is defined as the set of states that are eigenvectors of all stabilizer operators with eigenvalue $+1$:



(a) (b)

Figure 2.1: a) A simple distance 6 surface code encoding 1 logical qubit. Black dots are physical qubits. Each dark(light) plaquette represents a $X(Z)$ stabilizer. The semicircles at the boundaries are also stabilizers that involve just two qubits. The logical $\bar{X}$ and $\bar{Z}$ operators are shown by horizontal blue (dark gray) and vertical red (light gray) strings respectively. b) Schematic diagram of the surface code in (a).

$$\mathcal{C} = \{|\psi\rangle : S_p |\psi\rangle = +|\psi\rangle \quad \forall p\}. \tag{2.2}$$

The dimensionality of $\mathcal{C}$ determines how many logical qubits can be encoded in this surface code. For the lattice shown in Fig. 2.1a, there is one less stabilizer than physical qubits. Therefore $\mathcal{C}$ is two-dimensional and corresponds to the encoded logical qubit. It is possible to encode more than one logical qubit in one patch if one uses more defects on the boundary; $2n$ boundary defects can be used to encode $n-1$ logical qubits. However in the planar code, each logical qubit is associated with a separate patch.

Logical operators are associated with those unitary transformations which leave the code subspace $\mathcal{C}$ invariant, but which act non-trivially within $\mathcal{C}$. Logical Pauli operators $\bar{Z}$ and $\bar{X}$ correspond to a tensor product of Pauli operators for each physical qubit along a given string:

$$\bar{Z} = \prod_{i\in l} Z_i, \qquad \bar{X} = \prod_{i\in l'} X_i, \tag{2.3}$$

where l and l' are the light red and dark blue strings, respectively, depicted in Fig. 2.1a. The choice of l and l' is unphysical; any string l that connects the top and bottom edges is sufficient for $\bar{Z}$, and analogously for $\bar{X}$. Physically, $\bar{Z}$ corresponds to an e particle being transported between the top and bottom edge, while $\bar{X}$ corresponds to an m particle being transported between the left and right edge.

The distance of the code, d, is the minimum number of Pauli operators that appear in a nontrivial logical operator. For the code shown in Fig. 2.1, both $\bar{Z}$ and $\bar{X}$ have length 6, hence it is a distance $d = 6$ code.

2.1.1.1 Error correction in surface codes

Here we briefly review the proposal for active quantum error correction using the surface code. In this approach, all of the stabilizers S_p, for every plaquette, are measured in each round of quantum error correction. By constantly measuring the stabilizers S_p for every plaquette, we can ensure that the state of the system remains an eigenstate of each stabilizer.

Consider the planar code shown in Fig. 2.2. Let us say a bit flip error occurs on qubit number 1 and the wave function of the system changes to $X_1 |\psi\rangle$. Now, when we measure the stabilizers, assuming a perfect measurement, all syndromes would be $+1$ except for the measurement outcomes of Z stabilizers marked by blue circles in Fig. 2.2, which will be -1. Thus a single bit flip error creates two adjacent m particles. If instead of a bit flip, a phase flip error had happened, then it would be the X stabilizers marked by red triangles adjacent to qubit 1 that would give different output, giving rise to a pair of adjacent e particles.

An arbitrary single qubit error on the qubit number 1 would change the wave function of the

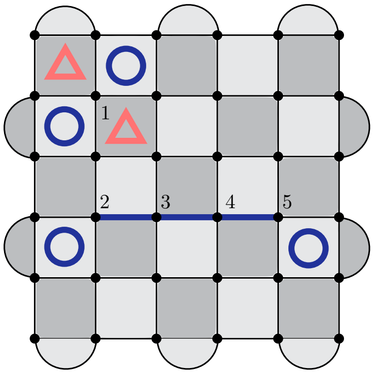


Figure 2.2: Error syndromes can be used to detect and correct errors. A single bit flip error on qubit 1 will create two neighboring m particles indicated here by blue (dark gray) circles whereas a phase flip error will create two e particles (shown as red (light gray) triangles). Error strings like the one shown at the bottom, can create isolated particles.

system to:

$$\begin{aligned} e^{i\theta \boldsymbol{n}\cdot\boldsymbol{\sigma}_1} |\psi\rangle =& \cos(\theta) |\psi\rangle + i \sin(\theta) n_x X_1 |\psi\rangle \\ &+ i \sin(\theta) n_y Y_1 |\psi\rangle + i \sin(\theta) n_z Z_1 |\psi\rangle \,. \end{aligned} \tag{2.4}$$

The four terms on the right hand side of Eq. (2.4) have different error syndromes and after one round of measurement, the system will collapse to one of them. So, from the standpoint of error correction, any single qubit error reduces to a bit flip or phase flip error, or a combination of the two.

If instead of a single qubit error, many adjacent qubits flip at the same time, only the syndromes at the end of the flipped string would give different values. Fig. 2.2 illustrates an example. An error string, such as $E = \prod_{i\in s} X_i$ for some string i, creates an m particle at each end of the string. Similarly an error string consisting of Pauli-Z errors creates a pair of e particles at its ends. However if an error string ends on an appropriate boundary, such as a Z error string that ends on an e boundary, then only a single e particle is created, and therefore only one stabilizer, at the endpoint of the string in the bulk, is violated.

Therefore, when an error occurs in the form of some strings, the only information we get from syndrome measurements is the position of the e and m particles. However, given a set of syndrome measurements (locations of e and m particles), the error string that can create it is non-unique; many different errors can result in the same configuration of particles. The minimum weight perfect-matching method [64, 65] can be used to track back the most likely error strings from the error syndromes. The method finds the set of shortest possible strings that connect a given set of particles. Since longer error strings occur with lesser probability, this algorithm finds the most probable error configuration consistent with the measured syndromes. A logical error occurs when the error string inferred from the minimum weight perfect matching algorithm differs from the correct error string by a non-contractible string. On the other hand, if the inferred error strings are always related to the true error strings by a contractible loop, then that means that we have successfully tracked all of the errors in the software and can compensate for them accordingly. Other variants of the matching algorithm can be used to improve the probability of guessing the true error configuration [66, 67].

Since the standard minimum weight matching algorithm runs in polynomial time in system size l, for large patches of surface code other methods like renormalization-group decoders with $\mathcal{O}(\log l)$ run time could become favourable [68, 69]. Having enough classical resources, one can also solve the minimum weight matching problem in constant time using parallel computing [70].

The probability of a logical error p_{fail} clearly depends on the underlying error model. For uncorrelated single qubit errors, numerical and analytical studies suggest an exponential suppression of p_{fail} with increasing code distance [3,44,71–73]. The rate of exponential decay depends on the physical error probability. Specifically, for fixed d and small probability of physical errors p, p_{fail} is best described by $A(d)(p/p_{\text{th}})^{d/2}$ where p_{th} is called the accuracy threshold [44,71,72,74]. The same form applies for other variants of the surface code but with different values for p_{th}.

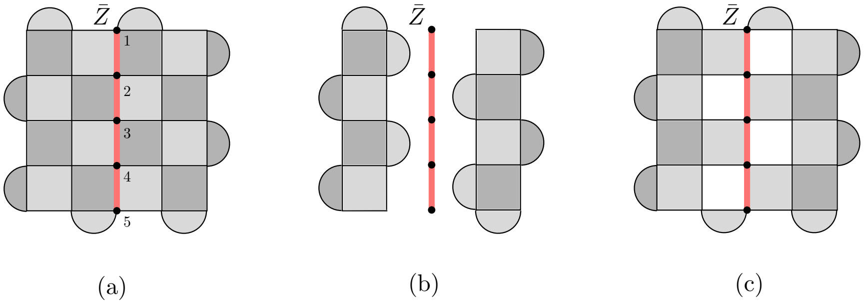
So far we have assumed that the measurement process is perfect. But one also needs to consider the errors that occur in the measurement process. Measurement errors can be addressed by repeating the measurements many times to distinguish the measurement errors from other errors. By repeating the measurement many times, we get a three dimensional map for the position of quasiparticles: two dimensions are used to record the error syndromes in space for each round of measurement and the third dimension is the discrete time. Now, we use the minimal weight perfect-matching algorithm to connect the quasiparticles in this three dimensional lattice together, allowing for the strings to have time segments as well as spatial ones [4,74].

The number of measurement histories that are used for error correction depends on the code distance and the probability of measurement errors. For equal error probability in measurement and storage, $\mathcal{O}(d)$ rounds of previous error syndromes are used to correct the code where d is the code distance [75,76].

2.1.1.2 Measuring string operators in planar codes

Here we will discuss how to fault-tolerantly measure the string operators associated with $\bar{X}$ and $\bar{Z}$. These methods can also be used for initializing logical qubits in the $\bar{X}$ or $\bar{Z}$ basis.

We note that one method to measure $\bar{X}$ and $\bar{Z}$ is to measure all physical qubits in the X or Z basis in order to measure the corresponding logical operator. However since this method is destructive it cannot be used when there are more than one logical qubits encoded in a patch. In



(a) (b) (c)

Figure 2.3: String initialization/measurement method. For initializing the code in an eigenstate of the string operator $\bar{Z}$ shown in a), one can turn off the X stabilizers adjacent to the string and change Z stabilizers next to the string to detach the string and form the code shown in b). Then measure the qubits on the string individually in the Z basis as well as the modified stabilizers for the sake of error correction, which yields the value of $\bar{Z}$. In this step the code effectively looks like (c). After correcting errors, we initialize physical qubits in the $|0\rangle$ state and turn back on all stabilizers in their original form. After d rounds of syndrome measurement and correcting errors, the code is initialized in the $|\bar{0}\rangle$ state.

contrast, the string measurement that we review below can be applied to more general encoding schemes as well.

Suppose for example that we wish to measure the string operator $Z_1Z_2Z_3Z_4Z_5$, shown in Fig. 2.3a. We proceed as follows:

1. We turn off every X stabilizer that shares a qubit with the string operator $\bar{Z}$. We also remove every qubit present in $\bar{Z}$ from all Z stabilizers adjacent to it, thus changing the 4 qubit Z stabilizers adjacent to the string operator to a pair of 2 qubit Z stabilizers. After making these changes, the code would look like Fig. 2.3b. Note that by doing so, we have created a new e edge along $\bar{Z}$.

2. We measure qubits 1-5 individually in the Z basis, in addition to performing the stabilizer measurements. We do d rounds of measurements to make the the procedure fault tolerant. Using the value of the individual qubit measurements, along with the measurement out-

come of the modified stabilizers, we can recover the value of the original Z stabilizers as well. This allows us to track the errors from before the measurement process began.

3. Finally, we turn on all stabilizers and change all the modified stabilizer operators back to their original form. We need to do d rounds of stabilizer measurements to establish stabilizer values and redefine the code space accordingly.

The measurement of $\bar{Z}$ is obtained by multiplying the measured values for the individual Z_i measurements along the string. To make the measurement fault tolerant, it is important to correct any bit flip errors on the $\bar{Z}$ string before using individual measurement outcomes in step 2, and to perform the measurement d times to protect against measurement errors. It is worth noting that one can also measure a ribbon of qubits with thickness d once, instead of measuring a string d times. However, to avoid decreasing the code distance, ribbon measurement requires using larger code patches.

Note that phase flip errors that occur on qubits 1-5 will not change the measurement outcome of $\bar{Z}$ operator.

Measurement in the $\bar{X}$ basis can be done by following similar steps. However, importantly, measurement of $\bar{Y}$ cannot be done in this encoding without introducing additional ingredients, as we describe later.

2.1.1.3 Quantum computing with planar codes

In order to implement universal fault-tolerant quantum computation, we need to implement a universal gate set fault-tolernatly. For the surface code, a natural choice is the Clifford group, together with the T gate, which is the $\pi/8$ single-qubit phase gate. Here we will briefly review the proposals for implementing logical Clifford gates in the encoding described above. The T gate is then implemented fault-tolerantly using magic state distillation.

The Clifford group is generated by the single-qubit Clifford phase gate, $\bar{S} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$, $\bar{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, and the two-qubit CNOT gate.

Note that logical $\bar{Z} = \bar{S}^2$ is easy to implement, as one can implement it transversally by applying the single-qubit Z gates on physical qubits along the $\bar{Z}$ string. $\bar{X} = \bar{H}\bar{Z}\bar{H}$ can be applied similarly.

The logical Hadamard gate, $\bar{H}$, is not as straightforward as $\bar{X}$ and $\bar{Z}$. Although applying the Hadamard gate transversally to each individual physical qubit does exchange eigenstates of $\bar{X}$ and $\bar{Z}$, it will also change the boundary conditions, as an e boundary is converted to an m boundary, and vice versa. Therefore, the transversal Hadamard operation does not yield the original code, but rather yields a $\pi/2$ rotated version of it. One then needs to correct the orientation by code deformation [3, 21, 23]. Code deformation changes the shape of a surface code geometrically by adding physical qubits to the lattice or removing some from it. Adding and removing here does not mean physical changes to the underlying lattice, but it refers to turning on some stabilizers to include some idle physical qubits or turning off some stabilizers to exclude some physical qubits from the code. These additional idle physical qubits add to the spatial overhead required for implementing $\bar{H}$.

The Clifford phase gate $\bar{S}$ is more complicated. Proposals for implementing $\bar{S}$ in *pure* planar encoding require encoding the logical $|Y\rangle$ state in the planar code through state distillation [3, 4] and thus have a large space time over head. However there are proposed hybrid schemes [28] that

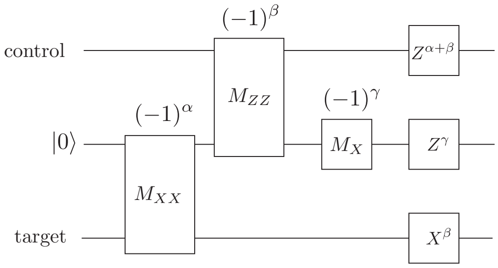


Figure 2.4: Quantum circuit for CNOT. The number above each measurement represents the outcome of that measurement.

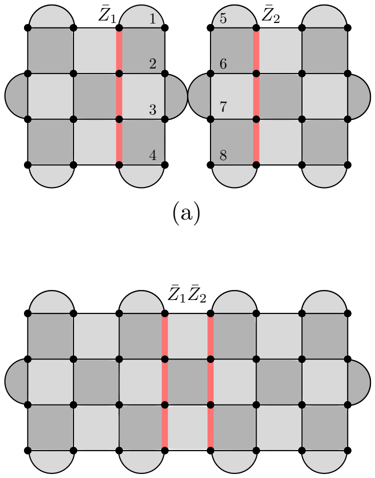
avoid state distillation for $\bar{S}$ gate which we will mention shortly. It is a possible to keep track of single qubit Clifford gates including $\bar{S}$ at the classical level, thus eliminating the need for state distillation by avoiding direct implementation of the $\bar{S}$ gate [25].

The two-qubit logical CNOT gate has been proposed to be implemented as follows. One method is to apply CNOT transversally between every physical qubit in one plane and the corresponding qubit in the other [3, 77]. However this operation is non-local if we limit ourselves to a single-layer two-dimensional layout, and thus will not be further considered.

A method for implementing CNOT using local interactions in planar codes uses a method referred to as lattice surgery [23]. This method utilizes an extra logical ancilla qubit together with the circuit shown in Fig. 2.4 [23]. M_O in the circuit indicates measurement of operator O.

We have already explained how to perform the M_X and M_Z measurements. What remains is to explain how to perform the joint measurement such as M_{XX} and M_{ZZ} in planar codes. It is important to note that measuring $\bar{Z}_1$ and $\bar{Z}_2$ separately and then multiplying the result is not equivalent to a $\bar{Z}_1\bar{Z}_2$ measurement, as the former will project the code into a smaller subspace than intended.

Consider two planar codes next to each other, as in Fig. 2.5a. Note that the neighboring boundaries are both m boundaries. To measure the two body operator $\bar{Z}_1\bar{Z}_2$ we use the following



(a)

(b)

Figure 2.5: a) Two disjoint planar codes with $\bar{Z}_i$ operators shown. b) To measure the parity operator $\bar{Z}_1\bar{Z}_2$, we turn on all Z stabilizers that are between two patches. We also combine two 2-qubit X stabilizers at the boundary into one full stabilizer. After d rounds of syndrome measurement and error correction we can find the value of $\bar{Z}_1\bar{Z}_2$ by multiplying the newly measured Z stabilizers.

steps:

1. We stop measuring the X_2X_3 and X_6X_7 stabilizers and start to measure the combined $X_2X_3X_6X_7$ stabilizer. At the same time, we start measuring two new Z stabilizers $Z_1Z_2Z_6Z_5$ and $Z_3Z_4Z_8Z_7$. This modification effectively merges the two patches together and the code will look like Fig. 2.5b.

2. We wait for d rounds of stabilizer measurements to establish the values of newly added Z stabilizers.

3. We read the value of $\bar{Z}_1\bar{Z}_2$ by multiplying the measurement outcomes of newly added Z stabilizers. After that we stop measuring all three shared stabilizers and turn back on X_2X_3 and X_6X_7 stabilizers to detach the codes again.

If the patches are oriented in such a way that e boundaries are next to each other, we can

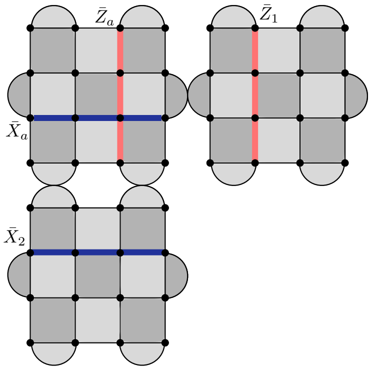


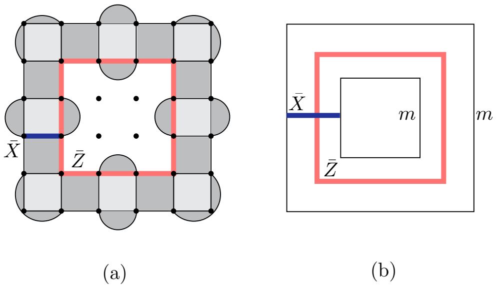
Figure 2.6: Planar code layout to do the CNOT. The patch in the corner encodes the ancilla qubit, the one in the right is the control qubit and the bottom one is the target qubit.

measure the $\bar{X}_1\bar{X}_2$ operator by turning on the shared X stabilizers. The procedure is similar to the M_{ZZ} measurement.

Using the joint measurements, the quantum circuit shown in Fig. 2.4 can be implemented by using the configuration shown in Fig. 2.6. The logical qubit in the corner is the ancilla qubit, the bottom patch encodes the target qubit and the other one is the control qubit. Note that the patches are oriented in such a way to make the joint measurements in Fig. 2.4 possible. In the last step, we need to apply single qubit gates based on the outcome of previous measurements as shown in Fig. 2.4.

2.1.2 Hole encoding

If we start with a planar code and remove some qubits from the bulk, we obtain a hole defect. Fig. 2.7a shows a hole defect that is created by turning off nine stabilizers. Although the qubits inside the hole are completely detached from the code, they are needed for moving the hole. Each hole introduces new edges and like the outer edges, the boundary of a hole can be either an e edge or m edge. In principle a hole can have mixed boundary conditions, but usually uniform boundaries are used.



(a) (b)

Figure 2.7: a) Surface code with hole defect in the bulk. Note that both the outer boundary and the hole's boundary are m boundaries. b) Schematic diagram for the lattice structure shown in (a).

Depending on the boundary type, e or m particles can condense on a hole boundary. This in turn allows one to use hole defects to make new logical operators and thus new logical qubits. The hole defect in Fig. 2.7, for example, encodes one logical qubit.

In general, n holes of the same boundary type define a 2^{n-1} dimensional code subspace. The proposal described in Ref. [4], however, uses a sparse encoding, where each logical qubit is encoded using two holes, as shown in Fig. 2.8. A logical qubit that is defined using a pair of e boundaries is called a X-cut qubit (Fig. 2.8 left). Likewise, Z-cut qubit refers to a logical qubit encoded in a pair of m boundaries (Fig. 2.8 right).

The sparse encoding allows for implementation of logical gates as described below. Our joint measurement technique, described in the subsequent sections, allows Clifford operations to be implemented using denser encodings, and thus may offer advantages in overhead.

2.1.2.1 Quantum computing with hole defects

The measurement and application of the logical $\bar{X}$ and $\bar{Z}$ operators proceeds analogously to the case of the planar encoding. The single qubit Clifford phase gate, $\bar{S}$, is also proposed to be implemented using state injection and state distillation, as in the case of the planar encoding.

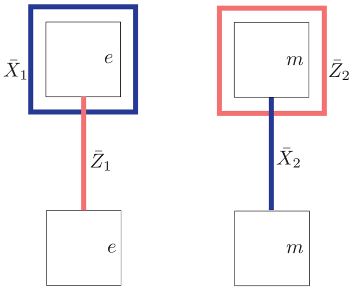
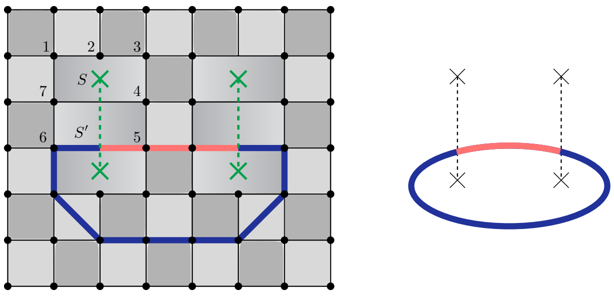


Figure 2.8: Logical qubits encoded in pairs of hole defects alongside the corresponding logical operators. The outer boundary of the code can be far away and is not shown here.

Similar to the planar code, one can circumvent distillation by using the hybrid schemes [28] which we will mention shortly.

The single qubit Hadamard gate $\bar{H}$ is performed through a series of code deformations [4,78], as follows. Assume we have a pair of e holes encoding our logical qubit. Since, unlike the planar code, there is generally more than one logical qubit encoded in a patch, first we isolate the target logical qubit from the rest by measuring a Pauli X string which encircles the hole pair. As was explained in Sec. 2.1.1.2, this would create an m boundary around the two e holes. By expanding the holes one can turn them into e boundaries of the isolated patch, converting the logical qubit to a patch of planar encoding. The Hadamard gate is then applied as it is in the planar code, described above, and then finally the logical qubit is converted back to the hole encoding and merged into the rest of the code.

The logical CNOT operation is quite different in the hole encoding as compared with the planar encoding. If we have a Z-cut qubit and a X-cut qubit, one can show that moving a hole of one qubit around a hole of the other, will perform CNOT between the two [4,78,79]. This process is called hole braiding. However performing CNOT between two qubits with the same type of holes is more complicated, because braiding two holes of the same boundary type is a trivial



(a) (b)

Figure 2.9: (a) A surface code with four twist defects marked by green crosses. The boundary of the surface code is not important and hence has not been shown. Green dashed lines are called dislocation lines. Every twist defect should be connected to another twist defect by a dislocation line. To make a pair of twist defects, we start with a perfect lattice and draw the dislocation line. Then we remove every qubit that lies on the dislocation line. Finally, we combine every two stabilizers that share an edge over the dislocation line into one stabilizer which is given by the product of original stabilizers. Clearly, the contribution of removed qubits have to be omitted. The combined stabilizer is represented by plaquettes with color gradient since they are neither X nor Z stabilizer, but have both operators. It is also possible to create dislocations without removing qubits by having the dislocation lines parallel to the Burgers vectors of the dislocations (not shown here) [15, 80]. (b) Schematic diagram of the surface code shown in (a).

operation in the code subspace. Instead, in this case one needs extra logical ancilla qubits encoded using holes with the other type of boundary. One can then implement the CNOT gate between two hole defects of the same type through a series of hole braidings and measurements [4]. Therefore, to perform a CNOT on two logical qubits requires a total of six holes, if the two logical qubits are both X- or Z- cut qubits.

2.1.3 Dislocation encoding

Making holes inside the bulk is not the only way to introduce non-trivial closed loops in surface codes. Twist defects can also be used to induce topological degeneracies and thus to encode logical qubits. Twist defects have been studied from a number of points of view, using topological field theory (see Sec. V of Ref. [81] and Ref. [80, 82–85]), chiral Luttinger liquid theory [80, 82, 83, 86–89], and in lattice models for topological order [15, 90–92].

Fig. 2.9a illustrates a surface code with four twist defects in the bulk which are marked by green crosses. As is clear from Fig. 2.9a, the physical qubits on the dislocation lines are removed from the lattice and all pairs of stabilizers that share an edge over the dislocation line are combined into one.[1] The stabilizers located on a twist defect involve five physical qubits, and one of the qubits should be measured in the Y basis. For example, the stabilizer S in Fig. 2.9a is defined as:

$$S = X_1 Y_2 Z_3 Z_4 X_7, \tag{2.5}$$

and the stabilizer corresponding to the plaquette just below that is:

$$S' = Z_7 X_4 X_5 Z_6. \tag{2.6}$$

As with hole defects, non-trivial string operators encircling twist defects can be used to define logical qubits. But an important property of dislocation lines is that whenever a string operator passes through them, it changes its type; a Pauli-Z string would change to a Pauli-X string and vice versa. As a result, a closed string operator needs to encircle at least two twist defects. A non-trivial closed string operator is shown as an example at the bottom of Fig. 2.9a and it includes both Pauli-X and Pauli-Z operators:

$$\bar{Z} = \prod_{i \in \text{light red part}} Z_i \prod_{j \in \text{dark blue part}} X_j. \tag{2.7}$$

---

[1] Fig. 2.9 illustrates dislocation lines that run perpendicularly to their Burgers vectors. It is also possible to consider a lattice geometry with dislocation lines parallel to the Burgers vectors, so that qubits do not need to be removed to create a dislocation.

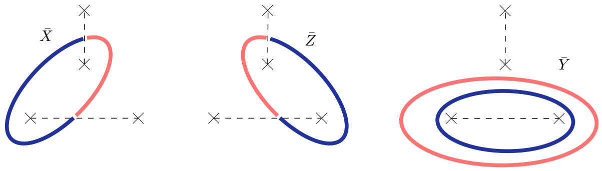
One can easily verify that this operator commutes with every stabilizer but is not a product of stabilizers itself. Fig. 2.9b shows the schematic diagram of the code shown in Fig. 2.9a.

In general, n pairs of twist defects gives rise to 2^{n-1} dimensional code space. In a dense encoding, therefore, there would be one logical qubit for every pair of twist defects (not counting the first pair). Alternatively, sparser encodings are also possible, using three or four twist defects to encode one logical qubit.

A key feature that distinguishes the dislocation code from planar and hole encodings is that the logical $\bar{Y}$ operator is also given by a simple Pauli string. Therefore the $\bar{Y}$ operator can be straightforwardly measured fault-tolerantly using the same methods for measuring $\bar{X}$ and $\bar{Z}$ operators in the planar and hole encodings. Alternatively, logical qubits can be initialized in the $\bar{Y}$ basis straightforwardly. Fig. 2.10 shows the logical $\bar{X}$, $\bar{Y}$ and $\bar{Z}$ operators for a logical qubit encoded in three twist defects. It can be shown that any two loop operators that encircle the same set of twist defects are equal to each other up to multiplication by some set of stabilizers and hence represent the same logical operator. For example, both Pauli-X and Pauli-Z strings shown in Fig. 2.10c are equal to $\bar{Y}$. To prove their equivalence one can use the fact that if a string goes around a single twist twice and closes itself, it acts as the identity on the code subspace (Fig. 2.11).

2.1.3.1 Initialization and measurement

Again, the string initialization and measurement that was described in Section 2.1.1.2 can be used here too. It is notable that by using string initialization, one can prepare the $|Y\rangle$ state without using state injection and state distillation procedures.



(a) (b) (c)

Figure 2.10: Logical operators related to a qubit encoded in three twists. (a) String operator corresponding to $\bar{X}$. It is a mixed Pauli string, with red (light gray) parts corresponding to Pauli X and blue (dark gray) parts corresponding to Pauli Z. Note that whenever the string passes through a dislocation line, it changes color. An equivalent string operator for $\bar{X}$ would be the one with red (light gray) and blue (dark gray) interchanged. All that matters is the defects a string enclose. (b) String operator corresponding to $\bar{Z}$. (c) Two equivalent string operators representing $\bar{Y}$.

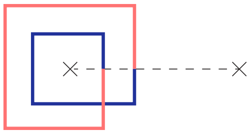


Figure 2.11: The string that encircles a single twist defect twice and encloses itself can be written as a product of stabilizers and hence acts as the identity operator in the code subspace.

2.1.3.2 Quantum Computation by twist defects

The fact that one can measure a logical qubit in the $\bar{Y}$ basis as well as $\bar{X}$ and $\bar{Z}$, allows one to ignore every single-qubit Clifford gate until there is a measurement, and then modify the measurement according to the awaiting gates [57]. For example if we want to apply an S gate on a logical qubit and measure it in the X basis, we can ignore the first gate and instead do the measurement in the $S^\dagger X S = Y$ basis. This point is explained in more detail in Section 2.2.4.

We can also easily implement CNOT using the circuit shown in Fig. 2.4. Joint measurements in dislocation codes are not really different from single qubit measurements. Suppose we want to measure the $\bar{X}_1\bar{X}_2$ operator where $\bar{X}_1$ is given by a string encircling twists 1 and 2 and $\bar{X}_2$ encircles twist defects numbered 3 and 4. It is easy to see that $\bar{X}_1\bar{X}_2$ is given by the simple string that encircles all four twist defects 1 to 4. We will explain this point further in Section 2.3, since joint measurements and twist defects lie at the heart of our method.

2.1.4 Hybrid schemes

Hybrid schemes for encoding logical qubits are also possible [25, 28, 59, 93]. Mixing two schemes also opens up the possibility of new methods to perform logical operations. Furthermore, logical qubits in different encodings can be entangled with each other using hybrid schemes, for example by braiding holes and twist defects.

For example, by converting boundary defects to bulk twist defects, braiding them and converting them back to the boundary, one can implement the $\bar{S}$ gate in planar codes [28] without state distilation. A closely related encoding, known as triangular code [59], can be constructed by starting with the planar encoding, moving one of the boundary defects to the bulk but leaving it there as a twist defect. The S gate in tirangular codes can also be applied without state injection.

2.2 Measurement-based protocols for Clifford gates

Here we explain how one can implement all gates in the Clifford group using circuits based on joint measurements. We only discuss the quantum circuits corresponding to the logical gates, regardless of the underlying setup which is used to encode logical qubits. In the subsequent sections, we will show how one can implement these circuits in surface codes, color codes and hyperbolic codes.

In this section all operators are understood to be logical operators, so we will omit the $\bar{\ }$ notation; all qubits are understood to be logical qubits.

2.2.1 CNOT gate

We have already mentioned the quantum circuit devised to implement CNOT using joint measurements (Fig. 2.4). It is used in many variants of surface codes as well as color codes to implement the CNOT gate [23–25, 57].

2.2.2 S gate

The circuit that is shown in Fig. 2.12 can be used to implement the S gate. Initially, the ancilla qubit is prepared in the $|+\rangle$ state, which is the $+1$ eigenstate of the Pauli X_a operator. Next, the two qubit parity operator ZY_a is measured, followed by a Z_a measurement. The subscript a is used to distinguish the operators associated with the ancilla qubit. The Pauli operators associated with the data qubit will have no subscript.

After the second measurement in Fig. 2.12, the state of the data qubit is given by:

$$\frac{1 - i(-1)^{\alpha+\beta} Z}{\sqrt{2}} |\psi\rangle, \tag{2.8}$$

where $(-1)^{\alpha}$ and $(-1)^{\beta}$ are the the results of first and second measurements. Note that the S gate can be written as:

$$S = e^{i\pi/4} \left(\frac{1 - iZ}{\sqrt{2}} \right). \tag{2.9}$$

Thus, if the outcome of the two measurements have the same sign, the state after the second measurement is, up to an overall phase, $S|\psi\rangle$, and therefore the S gate has been implemented.

On the other hand, if the results of the two measurements are different, the state of the data qubit would be $S^{\dagger}|\psi\rangle$. Since $S = ZS^{\dagger}$, we can recover $S|\psi\rangle$ by applying an additional Z gate

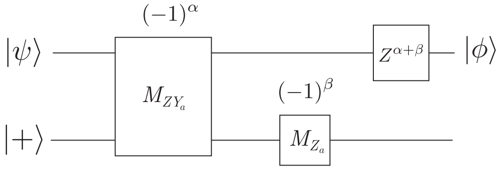


Figure 2.12: Quantum circuit for S gate. The ancilla is prepared in $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ state. The joint measurement corresponds to measuring the parity of the operator ZY_a, where the subscript a stand for ancilla qubit and Z is associated with the data qubit. The outcome of each measurement is written above its box.

to the data qubit.

2.2.3 SHS gate

We need another independent gate to fully implement the Clifford group. Usually it is the Hadamard gate H, but here we choose SHS since it has a simpler circuit. The circuit is shown in Fig. 2.13. Again, it is easy to check that after the second measurement, the state of the data qubit would be:

$$|\phi\rangle = \frac{1 + i(-1)^{\alpha+\beta}X}{\sqrt{2}}|\psi\rangle, \tag{2.10}$$

where $(-1)^\alpha$ and $(-1)^\beta$ are measurement results. Similar to Eqn. 2.9, we have:

$$SHS = \left(\frac{1 + iX}{\sqrt{2}}\right). \tag{2.11}$$

So if the results of the two measurements have the same sign, we get the desired state $SHS\,|\psi\rangle$. Again, if we get different signs, the data qubit would be in the state $S^\dagger HS^\dagger\,|\psi\rangle$. We can then recover $SHS\,|\psi\rangle$ by applying X, because $SHS = iXS^\dagger HS^\dagger$. Hence, at the end of the circuit, we get $|\phi\rangle = SHS\,|\psi\rangle$.

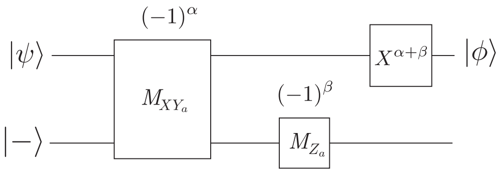


Figure 2.13: Quantum circuit to implement SHS gate. The ancilla is prepared in the $|-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$ state. The joint measurement corresponds to measurement of the parity of the operator XY_a, where subscript a stands for ancilla qubit and X is associated with the data qubit. The outcome of each measurement is written above its box.

2.2.4 Conjugated CNOT circuits

Since single qubit Clifford gates just permute the Pauli matrices (up to a sign), like the H gate that exchanges X and Z, it turns out that one can just keep track of them classically instead of actually applying them at the quantum level. One way to see this is to move all single qubit Clifford gates to the end of the circuit and then modify the final measurements accordingly. The price to pay is that for a general quantum circuit, we need to be able to implement CNOT and the $\pi/8$ phase gate T in any Pauli basis. In other words, we need to be able to implement CNOT and T conjugated by any single-qubit Clifford gate.

As an example, consider the quantum circuit shown in Fig. 2.14a. We use the identity CNOT $S = S\,(S^\dagger$ CNOT $S)$ to move the S gate across CNOT. Also since $S^\dagger H^\dagger \Pi_{\pm,Z} HS = \Pi_{\pm,Y}$, where $\Pi_{\pm,\sigma}$ denote the projection operator onto the $\pm$ eigenspace of σ operator, we can replace the upper Z measurement at the end of the circuit with a Y measurement. So the probabilities for each measurement outcome of this quantum circuit will be equivalent to performing instead $S^\dagger$ CNOT S, followed by measurement in a different basis (Fig. 2.14b). The quantum circuit for $S^\dagger$ CNOT S (Fig. 2.15) can be derived from the CNOT quantum circuit in Fig. 2.4.

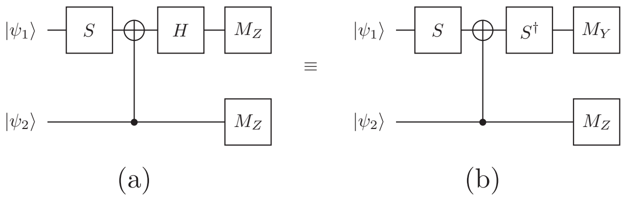


Figure 2.14: The two quantum circuits in (a) and (b) yield equivalent measurement outcomes, provided the measurements are done in different bases.

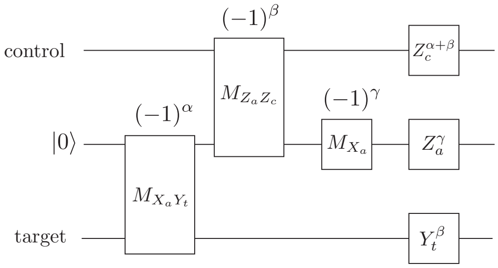
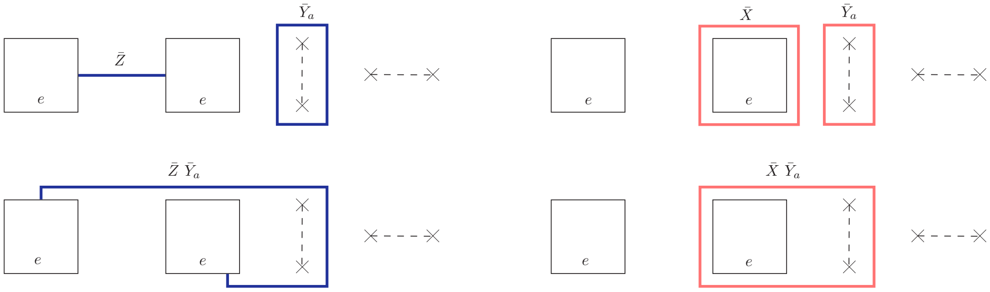


Figure 2.15: $S_t^\dagger$ CNOT S_t circuit. Here the subscripts a, c, and t refer to the ancilla, control, and target qubits, respectively.

2.3 Fault-tolerant joint measurements using twist defects

The measurement circuits for the implementation of the $\bar{S}$, $\bar{S}\bar{H}\bar{S}$, and $\overline{\text{CNOT}}$ gates discussed above require the ability to perform joint measurements of operators such as $\bar{Z}\bar{Z}_a$, $\bar{X}\bar{X}_a$, and $\bar{Z}\bar{Y}_a$. Alternatively, if the logical ancilla can be prepared in the $\bar{Y}$ basis, then we only need the joint measurements $\bar{Z}\bar{Z}_a$ and $\bar{X}\bar{X}_a$. In this section we introduce a method which utilizes twist defects to carry out the required measurements fault-tolerantly and with low overhead.

The main point here is that if the ancilla qubit is encoded using bulk twist defects, the joint measurements could be reduced to measuring simple string operators. Importantly, this is true irrespective of the encoding scheme of the logical data qubits; they could have been encoded using planar encoding, hole encoding or dislocation encoding. In general this procedure works for any encoding scheme as long as the logical $\bar{X}$ and $\bar{Z}$ operators of the logical data qubits can be represented by simple string operators rather than graph operators.



(a) (b)

Figure 2.16: Finding the string corresponding to parity operator measurements in a hole based encoding. Here a pair of Z-cut holes are used to encode a logical qubit. There is also an ancilla qubit encoded in four twist defects. Logical operators of the ancilla qubit are specified by subscript a. In (a) (top panel), one can see the string operators corresponding to $\bar{Z}$ and $\bar{Y}_a$ separately. Both of them are given by a product of Pauli Z operators acting on individual physical qubits. If we want to measure the parity operator $\bar{Z}\,\bar{Y}_a$, we can deform the string corresponding to $\bar{Z}$ in such a way to overlap the left side of the string corresponding to $\bar{Y}_a$. Since $\sigma_Z^2 = 1$ the overlapping part cancels out, and we would get the connected string shown in the lower panel of (a). In (b) we illustrate the string associated with the parity operator $\bar{X}\bar{Y}_a$. Since the logical qubit is encoded in a pair of Z-cut holes, the $\bar{X}$ operator is given by a Pauli-X string operator encircling one of the two holes. To do the joint measurement, we use a Pauli-X string encircling the two twists to represent $\bar{Y}_a$ shown in (b) up; As we explained, it is equivalent to the Pauli Z string encircling the same twists which we used for the $\bar{Z}\,\bar{Y}_a$ measurement. By deforming the $\bar{X}$ string and making an overlap with the $\bar{Y}_a$ string, the shared part cancels out and we get the string shown in the lower panel of (b) for the $\bar{X}\,\bar{Y}_a$ operator.

Suppose that we have some string operator running through some patch of a surface code, corresponding to a logical operator O. It can be a non-trivial loop encircling a hole or some twist defects, or a string that connects two same-type edges in a planar code. Also assume we have a logical qubit encoded in four twist defects in the bulk of the same patch. Imagine we want to measure the parity operator $\bar{O}\,\bar{Y}_a$. One can get a simple string corresponding to this operator by deforming the string corresponding to $\bar{O}$ in a way to also encircle the pair of twists that $\bar{Y}_a$ encircles. Now, if we measure this new single string operator, using the usual procedures used to measure string operators fault tolerantly, we would get the parity value, without measuring each individual logical operator separately. The same procedure works if one wants to measure other logical parity operators like $\bar{O}\,\bar{X}_a$ and $\bar{O}\,\bar{Z}_a$; one just needs to deform the string associated with $\bar{O}$ so it encircles the correct pair of twists. The explicit implementation of this procedure when $\bar{O}$ is the logical X or Z operator of a qubit encoded in a pair of Z-cut holes is shown in Fig. 2.16.

Having the tools, implementing each protocol is quite easy. We only explain the S gate implementation in the context of the hole based encoding, but the procedure is essentially the same for other gates (CNOT and SHS) in other encoding schemes.

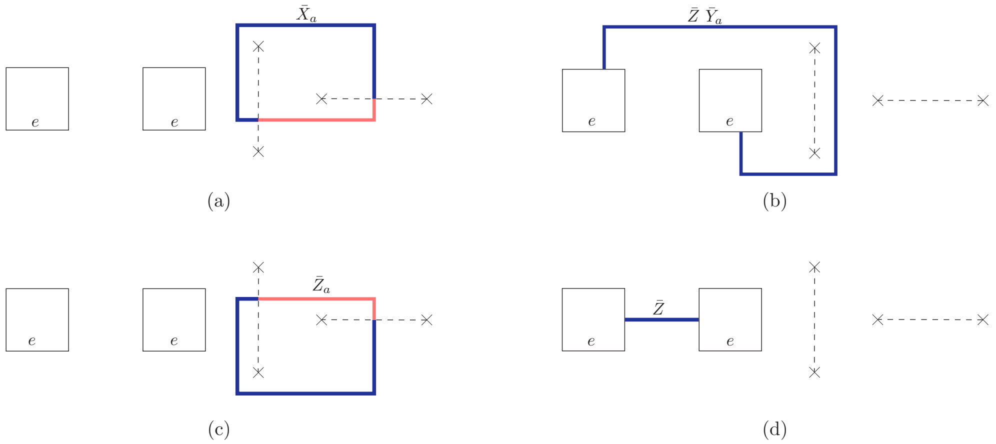
Assume we have a logical qubit $|\psi\rangle$ encoded in a pair of Z cut holes and we want to apply the S gate to it. Assume we have also an ancilla qubit nearby encoded in four twist defects. The following is the step by step description for implementing the S gate:

1. Prepare the ancilla qubit in the $|+\rangle$ logical state (Fig. 2.17a).

2. Measure the $\bar{Z}\,\bar{Y}_a$ string operator shown in Fig. 2.17b using string measurement method explained in Section 2.1.1.2. After reading the measurement result, turn on all the stabilizers and run d rounds of error correction to re-attach the lattice.

3. Measure the $\bar{Z}_a$ string shown in Fig. 2.17c. Again after doing the measurement, turn on all the stabilizers and go through d rounds of error correction.

4. If the results of two measurements had the same sign, the logical qubit has been projected in the $\bar{S}\left|\psi\right\rangle$ state and the procedure has been finished. Otherwise, perform a transversal phase flip $\bar{Z}$ gate along the string shown in Fig. 2.17d to get the desired result.



(a) (b)

(c) (d)

Figure 2.17: $\bar{S}$ gate implementation in hole encoding using a twist defect as ancilla. a) Initialize the ancilla in $\left|+\right\rangle$ state, using the string operator shown above. The red (light gray) and blue (dark gray) parts correspond to X and Z Pauli strings respectively. b) Measure parity operator $\bar{Z}\ \bar{Y}_a$ using the Pauli-Z string shown above. After measurement, glue the patch together by doing d rounds of error correction with full stabilizers. c) Measure $\bar{Z}_a$ for ancilla qubit, using the string operator shown above and again glue the surface together. d) If the results of two measurements in part b and c had different signs, apply $\bar{Z}$ transversally.

The procedures for implementing $\bar{S}\bar{H}\bar{S}$ and $\overline{\text{CNOT}}$ are quite similar to what is described above. One needs only to choose the right string for the measurement, and re-attach the lattice together after each measurement by going through d rounds of error correction.

2.4 Long range joint measurements

An important advantage of joint measurement scheme for Clifford operations is that it allows performing two-qubit gates between far apart logical qubits that are encoded in the same surface

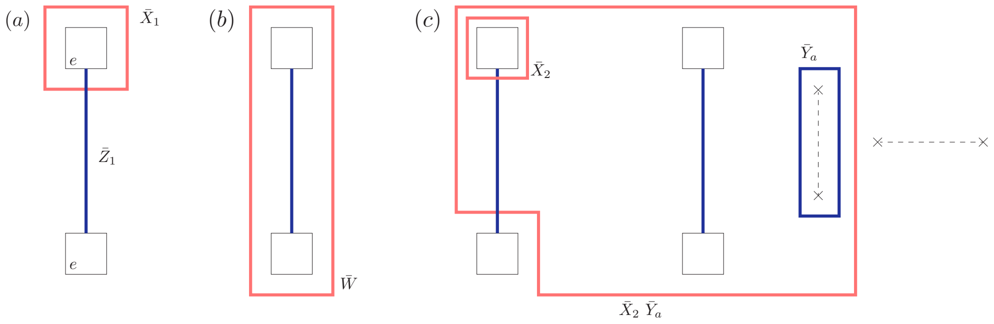


Figure 2.18: A patch of holes encoding four X-cut logical qubits and an ancilla qubit encoded using twist defects. a) Each pair encode a logical qubit with $\bar{Z}$ connecting two holes and $\bar{X}$ encircling the top hole. b) For each pair of holes, the Pauli-X string $\bar{W}$ that encircles both holes in topologically non-trivial but contains no information. We can initialize all such idle strings to $+1$ and use them to extend other strings that need to be measured to maintain the code distance. Note that the Pauli-Z string that encircles the holes can be absorbed into hole boundaries and so is always equal to $+1$. c) How one can utilize idle strings to perform long range joint measurements.

code patch. This is in contrast to proposals which requires one to move two logical qubits next to each other to perform two-qubit gates like CNOT between them. The reason that this is possible is that measuring longer strings do not require deeper circuits. The required depth is rather fixed by the code distance d.

However, it should be noted that the string measurement creates new edges in the system and as such, it can potentially reduce the code distance. To avoid this problem during the string measurements, it is always possible to encode the qubits far enough away from each other. However this could be inefficient since usually it increases the spatial overhead considerably. Other workarounds may be possible in certain cases that will result in no or very small increases in spatial overhead. As an example, we will explain how one can address this issue in sparse hole encodings.

When a hole based encoding is used, typically one places holes on a square lattice with distance d as is depicted in Fig. 2.18, and each pair stores one qubit of information as usual (Fig. 2.18a). However, there are still some nontrivial loops, like the string $\bar{W}$ in Fig. 2.18b, which are not used to encode any information. We call these strings *idle strings* and utilize these unused degrees of freedom to perform long range string measurements without decreasing the code distance. The idea is that we first initialize all idle strings encircling logical qubits to $+1$ and use them to extend the other strings through the code patch. For example, as is shown in Fig. 2.18c, to perform the joint measurement $X_2 Y_a$, where X_2 and Y_a are plotted in the figure, one can use the string that also encircles the logical qubit in between, without affecting the measurement result, which in turn helps to keep the code distance d. More details can be found in the caption.

2.5 Classical tracking of single qubit gates

As was mentioned in Section 2.2.4, instead of applying single qubit gates in a quantum circuit, one can trade CNOT gates in that circuit for conjugated versions of them and modify the final measurements. But this will be useful only if one can implement conjugated versions of CNOT with almost the same number of steps as the CNOT itself. Let us consider the $\bar{S}^\dagger\,\overline{\text{CNOT}}\,\bar{S}$ circuit (Fig. 2.15) as an example. The only non-trivial part of that circuit is the $M_{\bar{X}_a \bar{Y}_t}$ measurement, since this time the Y operator appearing in the operator to be measured is associated with a logical data qubit, whose encoding is arbitrary. For the dislocation code this is clearly not an issue [57] because we can measure the logical qubits in any Pauli basis fault tolerantly and hence the same joint measurement techniques described here can be utilized to measure the $\bar{X}_a \bar{Y}_t$ parity operator.

If the logical qubits are encoded using other types of defects, one needs to find a simple string (as opposed to graph) representation for the parity operator. Remarkably, this can be done in any encoding scheme, as long as there exist ancilla qubits in the twist defect encoding. An example is

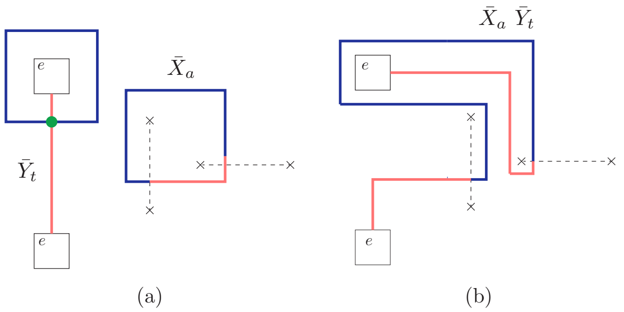


Figure 2.19: a) $\bar{X}_a$ and $\bar{Y}_t$ operators. b) $\bar{X}_a\bar{Y}_t$ operator given by a simple string.

shown for the case of hole encoding in Fig. 2.19 where one can find a simple string representation for the parity operator $\bar{X}_a\bar{Y}_t$. To identify the logical operators in Fig. 2.19 (a) and (b), we have used the fact that the double loop around a single twist defect is a logical identity (see Fig. 2.11). The Y measurement at the end of the modified quantum circuits can also be done similarly, by initializing the ancilla qubit in $|+\rangle$ state and then measuring the $\bar{X}_a\bar{Y}_t$ operator.

.

2.6 Hyperbolic code

The hyperbolic code is another variant of the surface code that uses different tilings of 2D surfaces to improve the encoding rate [54]. Ref. [27] proposes two possibilities for quantum information processing: (1) to perform Dehn twists, which can be used to either move the qubits around in storage, or to perform a logical CNOT between qubits stored in the same handle, and (2) to use lattice surgery to convert encoded information to a surface code, perform the necessary computations, and convert back to the hyperbolic code.

In this section we demonstrate how our methods can be used to implement fault-tolerantly the full Clifford gate set directly within the hyperbolic code, without moving the information into another quantum code patch.

The hyperbolic code is based on a tiling of a closed surface with regular polygons. A specific tiling is described by a set of two numbers $\{p, q\}$, known as Schläfli symbols, which represents a tiling of the plane with regular p-sided polygons such that q of them meet at every vertex. On a Euclidean plane, internal angles of a regular p-sided polygon are equal to $(p-2)\pi/p$. On the other hand if q polygons are to meet at a vertex, the internal angles should be equal to $2\pi/q$. Comparing these two, one can see that only tilings with $1/p + 1/q = 1/2$ can be realized on the Euclidean plane. However, one can use hyperbolic surfaces – surfaces with constant negative curvature – to realize $\{p, q\}$ tilings with $1/p + 1/q < 1/2$, since the sum of the internal angles of a regular polygon on a hyperbolic plane is less than $(p-2)\pi$. Fig. 2.20a illustrates the $\{5, 4\}$ tiling of the hyperbolic plane.

Given a $\{p, q\}$ tiling, one can define a stabilizer code where physical qubits lie on the edges and each vertex (plaquette) represents a Z-type(X-type) stabilizer. A topologically non-trivial closed hyperbolic surface with g handles has $2g$ non-trivial independent loops which can be used to define $2g$ logical qubits that are stabilized by the code. For large distances and fixed number of physical qubits, hyperbolic codes can encode more logical qubits compared to normal surface codes. However in order to realize such codes in an experimental system that is constrained to the Euclidean plane, non-local interactions are required.

If one prefers to work with a form similar to the surface code which was described in Section 2.1, where qubits lie on the lattice sites and all stabilizers are given by plaquette operators, one can use the rectified lattice, denoted by r$\{p, q\}$, which is constructed by connecting the midpoints of the edges in a $\{p, q\}$ lattice(Fig. 2.20b). The rectified lattice tiles the plane with regular p-sided and q-sided polygons. In this new form, qubits lie on the vertices and p and q sided plaquettes represent X and Z stabilizers respectively.

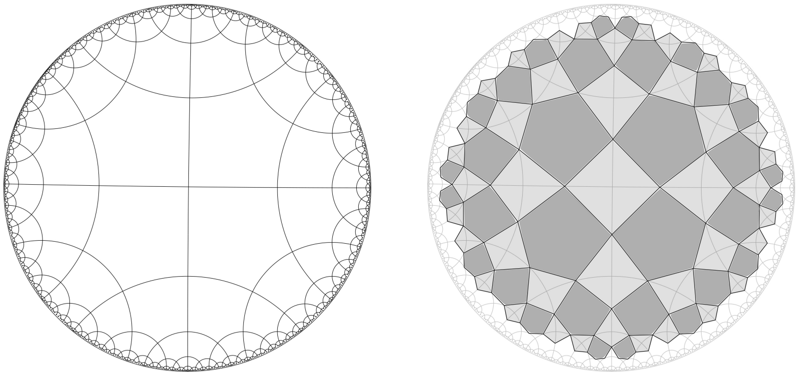

(a) (b)

Figure 2.20: a) $\{5,4\}$ tiling of the hyperbolic plane. b) The rectified tiling r$\{5,4\}$ can be constructed by connecting the midpoints of the edges in the $\{5,4\}$ tiling. The original $\{5,4\}$ tiling is also shown with light solid lines for comparison

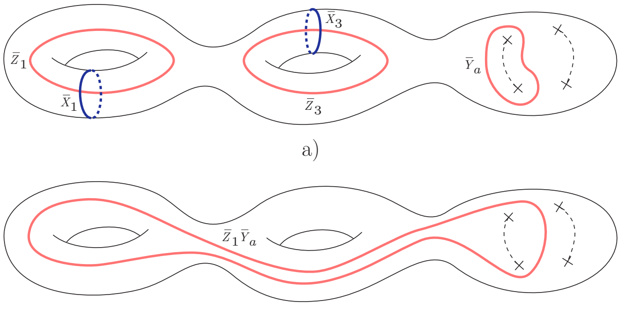


Figure 2.21: a) Hyperbolic surface with 2 handles (genus $g = 2$) encoding 4 logical qubits and four twist defects to encode the ancilla qubit. Logical qubits are encoded by using non-trivial loops on the surface. Some logical operators are shown in the figure as examples. The $\bar{X}_2$ and $\bar{X}_4$ ($\bar{Z}_2$ and $\bar{Z}_4$) operators are given by Pauli-X(Pauli-Z) strings that trace out the same loops as $\bar{Z}_1$ and $\bar{Z}_3$ ($\bar{X}_1$ and $\bar{X}_3$) operators respectively. b) A sample two qubit parity operator.

The joint measurement circuits for implementing Clifford gates can be used in the hyperbolic codes as well. To perform joint measurements fault tolerantly, one can use the original $\{p, q\}$ lattice and create twist defects in the bulk by following a procedure similar to what was described in Section 2.1.3. However one should be careful not to decrease the code distance and to keep track of what happens to other logical qubits. A more straightforward approach would be to select an arbitrary plaquette, divide it into a $2d \times 2d$ square lattice and create a pair of dislocation lines to encode the logical ancilla qubit. Dividing a plaquette by a square lattice clearly does not change the topology of the surface and keeps the code distance fixed.

Having a logical ancilla qubit encoded with twist defects in hand, performing joint measurements and implementing the quantum circuits described in Section 2.2 is straightforward. The single and two qubit parity operators used for Clifford group gates would be given by simple Pauli strings running through the hyperbolic plane (Fig. 2.21a) . Using the string measurement method, these operators can be measured fault-tolerantly using d rounds of error correction. Fig. 2.21b shows a typical Pauli string representing a two-qubit parity operator.

2.7 Color code

Color codes are another form of 2D topological codes, with the advantage of higher encoding rates and also allowing for natural transversal logical operations on the qubits. However, color codes usually have smaller error thresholds compared to surface codes. Nevertheless the trade-off between overhead and error thresholds could potentially favor the color codes in future experiments. Although there are already known methods for fault-tolerant quantum computing with color codes [13,24,94], here we point out that implementing the logical Clifford gates using the joint measurement techniques of this chapter could have its own advantages. Specifically, performing long-range two-qubit gates are more efficient with our method as compared with the transversal or lattice surgery methods. In contrast to the case of the surface code, our joint measurement protocols can be implemented in the color code in the case where the logical ancilla arises from a hole-based encoding.

The color code can be defined on any three-colorable, three-valent lattice. Qubits lie on the lattice sites and each plaquette corresponds to both X and Z stabilizer operators. The lattice structure ensures that all stabilizers commute with each other. Each stabilizer violation corresponds to a particle. To label the particles we use the color and type of the stabilizer it violates. So, r_x denotes a particle detected by a X stabilizer corresponding to a red plaquette. Since we have three different color plaquettes (say red, blue and green), and each plaquette corresponds to two stabilizers, naively there seems to be 6 independent particles in the theory. However, it can be shown that one can annihilate three particles of the same type and different colors with each other [13]. Thus, only 4 out of 6 are really independent particles. By considering composites of these 4 types, we find that there are 16 topologically distinct particles. Indeed, it can be shown that the color code is equivalent (after a finite-depth local unitary transformation) to two copies of the surface code, [55] which has a total of 16 topologically distinct particles.

Just like the surface code, topologically distinct particles always appear in pairs and each pair is connected via Pauli string operators. Since particles carry color, the string operators also would be red, blue or green. For example, a red Pauli-Z string connects two r_x particles. Note that a Pauli-Z string violates X stabilizers at its ends but commutes with Z stabilizers.

Similar to the two topologically distinct e and m boundaries in the surface code, the color code can have 6 topologically distinct types of boundaries, given that it is equivalent to two copies

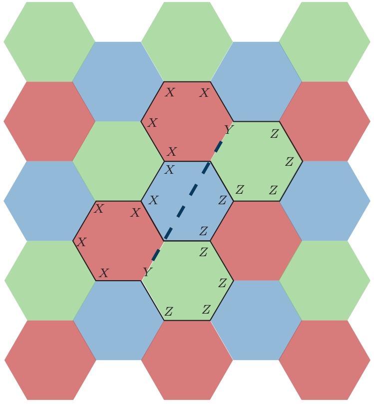


Figure 2.22: Making a twist defect in the color code. To make a twist defect, we first draw the dislocation line (dashed blue line) and remove the qubits which lie on the line. Then we merge pairs of X and Z stabilizers on the sides of the dislocation line into one. Note that in color code each plaquette represents two stabilizers. In this figure half of modified stabilizers are shown. For each stabilizer shown there is another one with X and Z operators exchanged. If the dislocation line passes through a plaquette (like the blue plaquette in the middle), one should merge the X and Z stabilizers corresponding to the same plaquette.

of the surface code (see Ref. [83, 95, 96] for a classification of topologically distinct boundary conditions in topological phases and see Ref. [17] for detailed study of the boundaries in the color code). In particular, the color code can have red, blue and green boundaries where red, blue and green particles can condense.

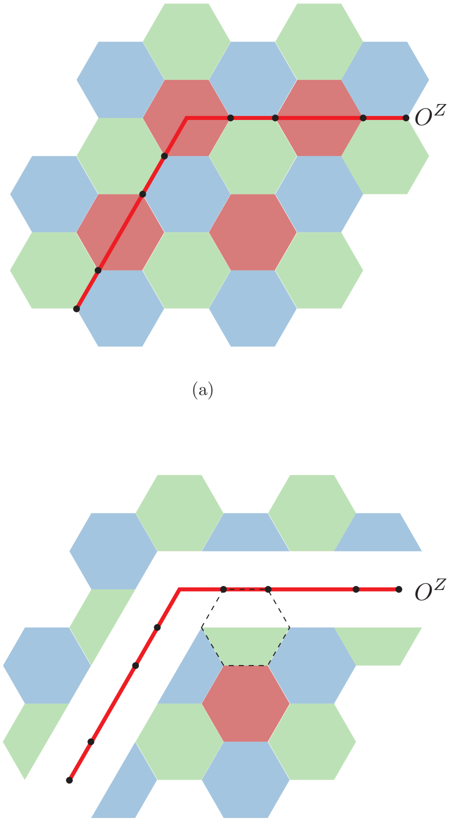
As in the case of the surface code, logical qubits can be defined through boundary defects, holes, bulk twist defects, or by having non-trivial genus [22, 84, 94, 97].

Holes are created by simply not measuring stabilizers within some region. To create a hole with a red boundary, for example, we consider a closed loop of red string, and stop measuring the stabilizers inside the loop. We also modify the stabilizers on the edge accordingly. Then we have created a hole with a red boundary where red strings can start or end on it without violating stabilizers.

The procedure to create bulk twist defects in the color code is similar to the case of the surface code. One chooses a dislocation line, removes the physical qubits over that line and merges pairs of X and Z stabilizers on either sides into one. Twist defects in surface codes transform e particles to m particles and vice versa. Since color codes have more particles, there are more types of twist defects one can create [85]. In Fig. 2.22 we have shown one possible example of a pair of twist defects connected to each other with a dislocation line. The one shown in Fig. 2.22 changes r_x to g_z, r_z to g_x, b_x to b_z and vice versa, as the particles enircle the twist defect.

Based on the underlying encoding scheme, different methods for initialization, measurement and realization of quantum gates can be used. Most of the techniques used in surface codes like lattice surgery and hole braiding have counterparts in color codes [24, 94]. To measure a string operator, one can practically follow the same procedure used in surface codes. An example is shown in Fig. 2.23. Let us say we want to measure the red Pauli-Z string operator O^Z shown in Fig. 2.23a. First, we detach from the stabilizers the qubits that lie on O^Z. This will change some of the green and blue stabilizers from 6 qubit to 4 qubit stabilizers. We also need to turn off all red stabilizers which O^Z passes through. After the modification, the color code will look like Fig. 2.23b. Note that this has effectively created a red boundary along the string, where red error strings can start and end without detection. But, just as in the case of the surface code, these undetected errors will not change the value of O^Z. In the next step, we measure each individual qubit on O^Z in the Z basis and also measure all stabilizers. By combining the outcome of individual measurements and modified stabilizers and comparing them with the value of the complete stabilizers, we can detect any error that happens during the measurement process. After correcting the errors, multiplying the outcome of individual measurements would give the value for the measurement outcome of O^Z.

The joint measurement circuits discussed in this chapter for implementing the Clifford group can also be implemented in color codes. If qubits are encoded in a single patch, for example using



(a)

(b)

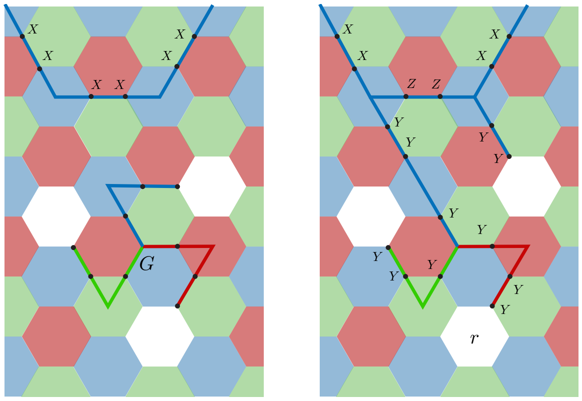
Figure 2.23: String measurement in color codes. a) If we want to measure the red (dark gray) Pauli-Z string operator O^Z, first we detach the qubits of O^Z from neighboring stabilizers and turn off all red (dark gray) stabilizers in its way. b) Then we measure individual qubits in Z basis and multiply the results to find the value of O^Z. One can correct errors by comparing the value of individual measurements and modified stabilizers with the original value of complete stabilizers. For example, the syndrome of the green stabilizer corresponding to the dashed plaquette before cutting the code, should be equal to the product of measurement outcomes of those two individual qubits and the 4 qubit stabilizer bellow them.

holes or dislocations, the quantum circuits described in section 2.2 can be implemented using a single logical ancilla encoded with twist defects. The rest of the protocol is directly analogous to the case of the surface code.

However, unlike the surface code, in color codes we are not restricted to use twist defects as the logical ancilla to implement the joint measurement method. An interesting feature of color codes is that one can measure not only X-type and Z-type strings fault tolerantly, but also Y-type strings. The reason is that in the color code, in contrast to the surface code, for a given plaquette we measure both X and Z stabilizers and the product of these outcomes gives the value of the corresponding Y stabilizer (we need to multiply it by $(i)^n$ where n is the number edges in the plaquette). This feature is a result of the fact that the color code is a CSS code [98] constructed from two *copies* of a single classical code. It is the same property that makes transversal methods natural in this architecture. This in turn enables us to create logical qubits where $\bar{Y}$ is given by a simple Pauli string, without using twist defects.

To implement measurements involving $\bar{Y}$ with a hole encoding, we encode the logical ancilla qubit using three holes associated with different colors, similar to the proposed hole-based encoding in Ref. [94], but with a small modification. Consider the three holes and the graph G that connects them, shown in Fig. 2.24a. We define $\bar{X}$ as the G^X operator, which means the product of Pauli-X operators along the graph, and, similarly, the $\bar{Z}$ as the G^Z operator. Since the graph G consists of an odd number of qubits, G^X anti-commutes with G^Z. The advantage of this scheme is that the logical Y operator would be a Pauli-Y graph operator, denoted G^Y (in contrast to the proposed method in Ref. [94]) and can be measured fault tolerantly.

If we encode the ancilla qubit in the aforementioned three hole structure, no matter how the data qubits are encoded, as long as the logical X and Z operators of the data qubits are given



(a)

(b)

Figure 2.24: a) Three holes and the G graph that connects them encode the logical ancilla qubit. Blank plaquettes represent holes where we do not measure any stabilizer. $\bar{X}_a$, $\bar{Z}_a$ and $\bar{Y}_a$ are given by G^X, G^Z and G^Y respectively. At the top, part of a Pauli string is shown which is a portion of $\bar{X}$ operator related to another logical qubit. b) To measure two qubit parity operator $\bar{Y}_a\bar{X}$ one should deform G in such a way to overlap with the string related to $\bar{X}$ and measure the resulting string.

by deformable strings, we can use joint measurement for quantum computation. The idea is similar to what was described in surface codes. For parity measurements, we deform the strings to overlap and measure the resulting string (Fig. 2.24b). Since the G graph has all three different colors, we can deform it to overlap with any other string operator along a line. Then, the string measurement method can be used to find the parity value fault tolerantly.

Chapter 3: instantaneous Dehn twists for hyperbolic quantum codes

The significant effects of decoherence on quantum systems require that a fault-tolerant quantum computer appropriately encode logical quantum information and furthermore apply logical gates directly on the encoded qubits [99, 100]. However it is currently an open question to understand the ultimate asymptotic resource costs required for performing quantum error correction and fault-tolerant quantum computation. In this chapter we presents a protocol which improves upon the known optimal asymptotic space-time resource costs by proposing constant depth topologically protected unitary circuits that implement a universal set of logical gates for the Tuarev-Viro codes [101] defined on hyperbolic surfaces. Our protocols demonstrate that protected universal gate sets can be applied in parallel and with constant space overhead using constant depth unitary circuits.

The fault-tolerant storage and processing of quantum information come at significant resource costs in both space overhead, n/k, and the time overhead for implementing logical gates. Families of QECCs that are known to possess a finite error threshold, code distance d growing with n, and constant space overhead, where n/k is a constant independent of code distance d, have been proposed through two basic constructions. The first is in terms of topological codes defined on the cellulation of hyperbolic space [102, 103]. The second is various types of tensor product codes [104–107][1].

---

[1] Many other QECCs with constant space overhead are also known (see e.g. Ref. [108, 109]), however it is not

While constant space overhead is known to be possible, it is unclear what the fundamental time overhead must be for performing a universal set of logical gates. Universal logical gate sets on encoded qubits can in general be implemented through three known methods: (1) state distillation and gate implementation through measurements [99, 112, 113], (2) code switching [114], and (3) braiding or Dehn twist operations in appropriate classes of topological codes [2, 60, 61, 101, 115–117].

The methods (1) - (3) necessarily require either (a) measurements, which are non-trivial to take fault-tolerantly, together with follow-up operations that depend on the results of those measurements, or (b) unitary circuits whose depth grows linearly with the code distance. An exception is the class of protocols introduced in Ref. [37, 38], which demonstrate how universal gate sets can be applied through constant depth unitary circuits (with the depth independent of n and d).

To date, it has not been demonstrated how methods (2) or (3) can be combined with QECCs that have constant space overhead. Moreover, to our knowledge the optimal space-time overhead achieved using method (1) is due to a proposal of Gottesman [112], who showed that constant space and time overhead are simultaneously achievable for *sequential* quantum circuits on Pauli stabilizer codes. This means that a generic quantum circuit of depth D, which can be implemented in $\mathcal{O}(D)$ time using parallel gate operations, could take up to time $\mathcal{O}(kD)$ if implemented sequentially. Note that for quantum codes with constant encoding rate, k is proportional to n and as such, the circuit depth of the sequential implementation could grow linearly with the system size.

---

clear whether they possess a finite error threshold, as they are not low density parity check (LDPC) codes [110]. The LDPC property guarantees that a stabilizer code with $d \sim n^{\alpha}$ for $\alpha > 0$ will have finite error threshold. [111, 112].

In this chapter we consider the most general class of topological codes, which we refer to as Turaev-Viro codes. These are based on Turaev-Viro-Barrett-Westbury topological quantum field theories (TQFTs) [118, 119] and are stabilized by the Levin-Wen string-net Hamiltonians [116]. This class of TQFTs include, as special cases, the theories that describe the Kitaev surface code and quantum double models [2]. The use of these TQFTs and the associated Levin-Wen Hamiltonians as quantum codes was discussed explicitly in Ref. [101].

Here we consider Turaev-Viro codes defined on triangulations of hyperbolic surfaces. As we review below, the constant negative curvature of the hyperbolic surface allows for a finite encoding rate and thus constant space overhead for the associated topological code [102]. Such hyperbolic codes can be implemented in a flat two-dimensional layout of physical qubits by allowing long-range couplings between the physical qubits – a possibility allowed by a variety of quantum computing architectures (e.g. ion traps [120, 121], modular architectures of superconducting cavity networks [122–125], Rydberg atoms [126–129] and silicon photonics [130]), and a necessity for any QECC with constant space overhead. In particular, experimental realization of hyperbolic circuit QED lattices [131, 132] is achieved by utilizing the feature that the cavity quantum bus (used to connect superconducting qubits) can have a wide range of length scales.

The main result of our chapter is an explicit protocol for implementing Dehn twist operations in hyperbolic Turaev-Viro codes through constant depth unitary circuits. Here constant depth refers to the fact that the depth of the circuit is independent of n and d. We note that throughout this chapter, by constant depth unitary circuit, we assume implicitly that every gate in the circuit also acts on a constant number (independent of d and n) of physical qubits. In contrast to the proposal of Ref. [112] which implements logical gates sequentially via gate teleportation, our approach allows parallel gate operations on encoded qubits via unitary circuits. Our circuit takes the form of a permutation on qubits, followed by a constant depth unitary circuit which is local with respect to the hyperbolic metric that abstractly defines the code. The permutation can be implemented with a depth-two circuit by applying long-range SWAP operations in parallel throughout the code. These results generalize the proposals of Refs. [37, 38, 133] demonstrating that braids and Dehn twists in general topological codes associated with triangulations of Euclidean space can be implemented by similar constant depth unitary circuits. The extension to hyperbolic space described here implies that these protocols are also compatible with having constant space overhead. Our results demonstrate explicitly an advantage in terms of space complexity for implementing the mapping class group of closed manifolds in topological codes, as compared with the more well-studied braid group of punctures [2, 30, 134].

Our protocols are inherently protected from errors in the sense that all error strings that are introduced to the system by faulty physical operations have $\mathcal{O}(1)$ length; moreover, all pre-existing error strings grow by at most an $\mathcal{O}(1)$ factor. Stated differently, our circuits map local operators to local operators: an operator with support in a local region $\mathcal{R}$ is mapped to an operator with support in a local region $\mathcal{R}'$, such that the area of $\mathcal{R}$ and $\mathcal{R}'$ are related by a constant factor independent of d. However, since this constant factor is greater than unity, our circuits may grow error strings by a constant factor that is larger than unity. In the presence of syndrome measurement errors we thus require $\mathcal{O}(d)$ rounds of error correction for every $\mathcal{O}(\log d)$ logical gates that are applied.

Our protocols demonstrate how to apply logical gate operations in hyperbolic codes without increasing the space overhead. The constant depth logical gates developed here for the general hyperbolic Turaev-Viro code can also be applied to the hyperbolic surface code as a specific case, where a subset of Clifford logical gates [133] can be implemented through constant depth unitary circuits. We note that Ref. [135] also discussed Dehn twists in hyperbolic stabilizer codes, however constant depth protocols that can preserve the constant space overhead were not presented.

As we discuss, our protocols have important implications for improving the asymptotic scaling of the space-time overhead required for fault-tolerant universal quantum computation. In particular, they suggest that universal fault-tolerant quantum computation with constant space overhead and a time cost of $\mathcal{O}(dD/\log d)$ is possible for a logical quantum circuit of depth D.

This chapter is organized as follows. In Section 3.1, we review the construction of hyperbolic Turaev-Viro codes. In Section 3.2, we sketch the two key steps for how to implement logical operations corresponding to Dehn twists in hyperbolic Turaev-Viro codes. The first step requires explicit maps representing Dehn twists on hyperbolic surfaces, which we present in Section 3.3. The second step, explained in Section 3.4 demonstrates how Turaev-Viro codes associated with different triangulations of hyperbolic space can be related to each other through local constant depth circuits. In Section 3.5 we discuss the fault-tolerance of these protocols and and their implications for asymptotic space-time resource costs for universal fault-tolerant quantum computation.

3.1 Hyperbolic Turaev-Viro code

The Turaev Viro code [101, 116] is a quantum error correcting code which is defined based on a given unitary fusion category $\mathcal{C}$ and is constructed using qudits that reside on the edges of a triangulation Λ of a surface Σ. Below we briefly review the construction of such codes.

Consider a unitary fusion category [117] $\mathcal{C}$ with N simple objects. We associate a vector

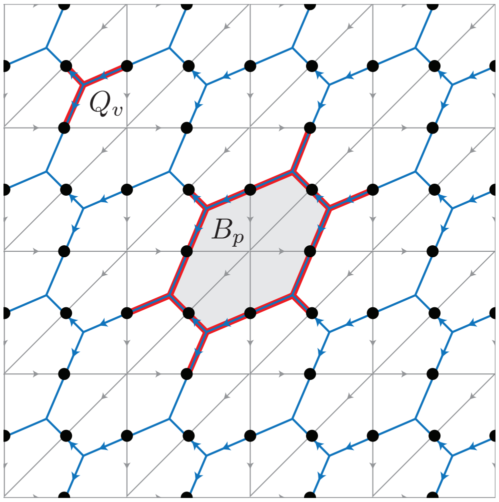


Figure 3.1: A triangulation Λ drawn by light gray solid lines and its dual trivalent graph $\hat{\Lambda}$ drawn by dark blue lines. The arrows on the edges define the branching structure. Black dots represent the physical qubits. Examples of vertex operators Q_v and flux operators B_p and their support are also illustrated.

space V_{ab}^{c} to any triplet of simple objects (a, b, c), whose dimension corresponds to the fusion rules $N_{ab}^{c} = \dim V_{ab}^{c}$. We may write fusion rules formally as:

$$a \times b = \sum_{c} N_{ab}^{c}\, c. \tag{3.1}$$

Associativity of the fusion rules gives a constraint on the fusion coefficients N_{ab}^{c}. In particular, the vector spaces $\bigoplus_e V_{ab}^{e} \otimes V_{ec}^{d}$ and $\bigoplus_f V_{bc}^{f} \otimes V_{af}^{d}$ are isomorphic. The unitary map between these two vector spaces are the F-symbols of the theory,

$$F_d^{abc} : \bigoplus_e V_{ab}^{e} \otimes V_{ec}^{d} \mapsto \bigoplus_f V_{bc}^{f} \otimes V_{af}^{d} \tag{3.2}$$

Consider a surface Σ together with a triangulation Λ (see Fig. 3.1) of Σ. The triangulation is also equipped with a branching structure (i.e. a local ordering of vertices). To each edge of the triangulation we assign an N-state qudit with states labeled $|a_i\rangle$, where a_i are the simple objects in $\mathcal{C}$. For simplicity of the construction we assume that the fusion rules N_{ab}^{c} are 0 or 1, although this can be easily generalized by including additional degrees of freedom at the vertices.

To define the code, it is simpler to work with the cellulation dual to Λ, which we denote by $\hat{\Lambda}$. Note that since Λ is a triangulation of the surface, $\hat{\Lambda}$ is a trivalent graph. If a qudit is in state $|a\rangle$, we say a type a string is passing through the corresponding edge on $\hat{\Lambda}$ and label the edge by a. The wave functions of the code space then can be seen as superpositions of string-net configurations that are consistent with certain string branching rules [116].

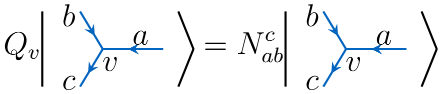
The code space $\mathcal{H}_{\Lambda}(\Sigma)$ is a subspace of the full Hilbert space of the physical qudits. The topological nature of the code guarantees that different choices of triangulations Λ and branching structures yield isomorphic code subspaces.

In particular $\mathcal{H}_{\Lambda}(\Sigma)$ corresponds to the ground state subspace of a local Levin-Wen Hamiltonian [116]:

$$H_{\hat{\Lambda}} = -\sum_{v} Q_v - \sum_{p} B_p, \tag{3.3}$$

where $\sum_v$ and $\sum_p$ sum over all vertices and plaquettes of $\hat{\Lambda}$ respectively (see Fig. 3.1). One can think of this model as a generalization from the abelian surface code to arbitrary (abelian and non-abelian) non-chiral topological orders in 2D. We note that the Levin-Wen Hamiltonian as defined in Ref. [116] requires a certain tetrahedral symmetry for the F-symbols, which makes the branching structure of the triangulation unnecessary, although the construction can be generalized to relax this condition.

The vertex operator Q_v is a local projection operator which ensures that the ground state wave function is consistent with the branching rules of the theory. The action of Q_v on a state $|\psi\rangle$ depends only on the states of qudits that reside on the edges which are incident to v and is defined as:



(3.4)

The action of the flux operator B_p on a wave function depends only on the edges which are incident on the vertices of p and is defined using the F-symbols. It can be thought of as the generalization of the local plaquette operators in $\mathbb{Z}_2$ surface codes. Its exact form is rather complicated and we refer the interested reader to References [101, 116] for a complete and through review of the Turaev-Viro codes.

Just like the surface code, the eigenstates of the Hamiltonian (3.3) can be realized in an active error correction approach by repeated measurement of all Q_v and B_p operators, each of which can be performed by a local constant depth circuit together with an ancilla. The measurement results are then used to detect and correct possible errors. Designing explicit quantum circuits for syndrome measurements and finding fault tolerant error correction schemes for specific variants of Turaev-Viro codes are subjects of ongoing research [136–141].

A hyperbolic Turaev-Viro code is a Turaev-Viro code which is defined on a triangulation of a closed hyperbolic surface Σ. A closed hyperbolic surface is a closed surface endowed with a Riemannian metric of constant curvature -1. Due to the Gauss-Bonnet theorem, the area of such a surface can be found from its genus g:

$$A_\Sigma = 4\pi(g-1). \tag{3.5}$$

The number of logical qubits (i.e. $\log[\dim \mathcal{H}_\Lambda(\Sigma)]$) for a topological quantum code on a closed genus g surface is proportional to g. On the other hand, for a fine triangulation with bounded geometry, where by bounded geometry we mean that the edge lengths and angles are bounded from above and below, the number of physical qubits n is proportional to the surface area A_Σ and the code distance d scales as $\log(n)$ [102]. Therefore, according to (3.5), by using hyperbolic surfaces of increasing genus we can construct a family of hyperbolic Turaev-Viro codes with constant encoding rate and increasing code distance. Note that the encoding rate for a quantum code defined on a Euclidean surface is $\mathcal{O}(1/d^2)$ and goes to 0 as one goes to large distances [46].

We further note that the existence of a lower bound on the angles in the fine triangulation also ensures that the associated error-correcting code is in essence a low-density parity check (LDPC) code, which has a low-weight (upper-bounded) plaquette syndrome B_p on the dual trivalent graph $\hat{\Lambda}$ (the vertex operator Q_v is always weight-3 due to the definition of a triangulation or equivalently the trivalent structure of its dual graph). As is the case with stabilizer codes [111], the LDPC property is important for the possibility of an error threshold in the

We note that numerical evidence for a finite error threshold in hyperbolic surface (stabilizer) codes was established in Ref. [54, 135].

3.2 Geometric gate sets for hyperbolic Turaev-Viro codes

Consider a Turaev-Viro code defined using a unitary fusion category $\mathcal{C}$ on a triangulation Λ of a closed surface Σ with genus g. It is well-known that the code space $\mathcal{H}_\Lambda(\Sigma)$ forms a non-trivial representation of the mapping class group (MCG) of the surface Σ. In other words, elements of the MCG implement certain non-trivial operations on the code space. Recall that the MCG is the group of homeomorphisms of the surface modulo those homeomorphisms that are continuously connected to the identity. We call the set of such operations the *geometric gate set*. Gates corresponding to MCG operations are naturally topologically protected (and thus can be made fault-tolerant) and can be implemented through a variety of methods [30, 38, 61, 101, 117, 133]. For certain codes, such as the Fibonacci Turaev-Viro code, the geometric gate set forms a universal gate set [115].

Here we consider a way of implementing MCG elements in terms of constant depth unitary circuits that is closely related to the method proposed in [38], although our presentation below is somewhat different and more general. This method can then be applied to the case of hyperbolic codes which yield constant space overhead.

Let U be a mapping class group element of the surface Σ. We denote its representation on the code space $\mathcal{H}_\Lambda(\Sigma)$ by $\mathcal{U}$. For a given U, one can implement $\mathcal{U}$ using the following procedure:

- (Step 1) Let $f_U : \Sigma \to \Sigma$ be a specific homeomorphism representing U. We move the vertices of Λ using f_U, and connect them as they were connected originally to get a new triangulation of Σ which we denote by Λ'. This operation corresponds to a permutation of the physical qubits. If the qubits are mobile, this transformation can be carried out by shuttling the qubits around. Otherwise, it can be implemented as a depth-two circuit by using long-range SWAPs in parallel throughout the system [38].

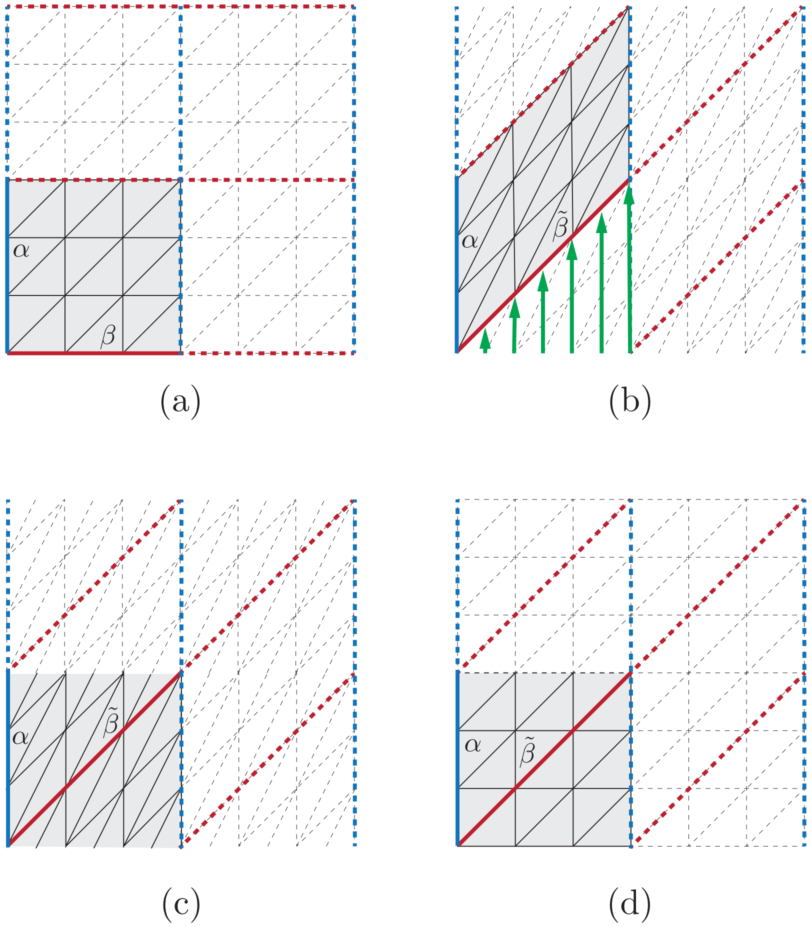
- (Step 2) Since the Turaev-Viro code was defined using the triangulation Λ, after the first step the wave function of the system would no longer be in the original code space $\mathcal{H}_\Lambda(\Sigma)$. Rather it would be associated to the code space $\mathcal{H}_{\Lambda'}(\Sigma)$ of a different triangulation Λ'. To remedy this, we apply a local quantum circuit that effectively implements a local geometry deformation and transforms the code defined on the Λ' triangulation back to the one

> defined on the original Λ triangulation. We will show in subsequent sections that this retriangulation can be performed via a *constant depth* local quantum circuit. If we regard this transformation as a homeomorphism of the surface Σ, it would be equivalent to the trivial element of MCG, which ensures that it will not result in another nontrivial transformation on top of the map U. Details of the geometry deformation circuit are explained in Section 3.4.

As an example, take Σ to be the torus T^2 with a regular triangulation Λ which is used to define the Turaev-Viro code. To construct the torus, we can take a square of side 1 and identify the opposite sides.

Alternatively, we can start with the complex plane $\mathbb{C}$ and identify points according to equivalence relations $z \sim z + 1$ and $z \sim z + i$. We can use these identification rules to define a universal covering map from $\mathbb{C}$ to T^2. A triangulation of T^2 then translates to a triangulation of the complex plane (see Fig. 3.2a).

Let $U = D_\alpha$ be the Dehn twist along α, the meridional loop of the torus(for a brief review of Dehn twists, see Appendix A). Consider the shearing map $f_D(x + iy) = x + i(x + y)$. It is easy to verify that this map respects the equivalence relations and corresponds to a Dehn twist along the α loop. If we move (permute) the qubits according to f_D, we get the configuration shown in Fig. 3.2b. Note that as a result of this map, the string along the β loop now encircles both handles while the string along the α loop remains unchanged, as one would expect form a Dehn twist along α. As a result of the previous step, the triangulation of the torus has been changed, as one can see by comparing Fig. 3.2c and Fig. 3.2a. To compensate for that, we will apply a local unitary circuit, which corresponds to the trivial element of the MCG, to restore the original



(a) (b)

(c) (d)

Figure 3.2: (a) The complex plane $\mathbb{C}$ as the universal cover of the torus T^2. Blue and red lines correspond to the canonical loops on the torus. The triangulation Λ of the torus is shown as well. The shaded region can be taken as the fundamental domain of the covering. By applying $f_U(x + iy) = x + i(x + y)$ (green arrows), the shaded region in (a) maps to the sheared region in (b) and the β loop maps to $\tilde{\beta}$ while the α loop remains intact. By looking at the original fundamental domain as in (c) it becomes clear that $\tilde{\beta}$ goes around both handles. Application of the local geometry deformation circuit then recovers the original triangulation as shown in (d) and maps the wave function back into the original code space.

triangulation without applying any further logical gate. The final result is shown in Fig. 3.2d.

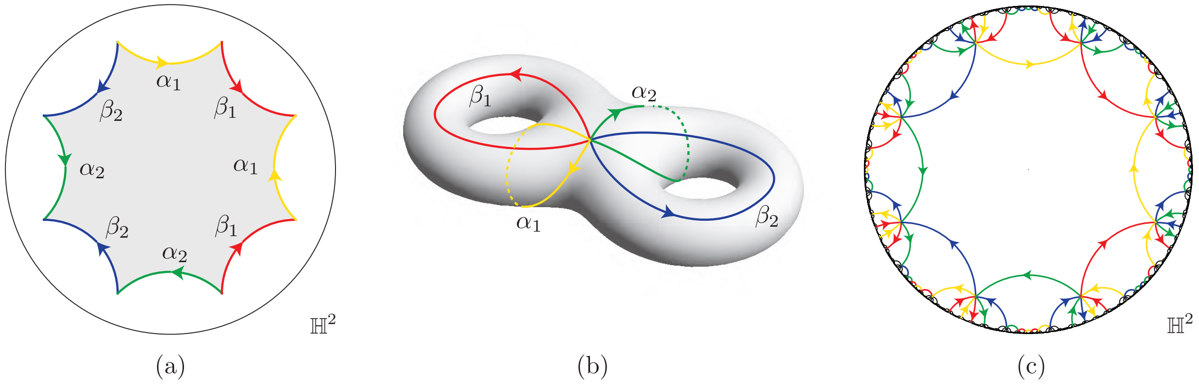
In Ref [37, 38], the above procedure has been used extensively to implement MCG elements by finite depth quantum circuits in QECCs which are defined on Euclidean surfaces. The main result of this chapter is that the same basic idea can be used to implement logical gates in QECCs which are defined on a hyperbolic surface. In the following we are going to explain in detail how one can implement geometric gates in hyperbolic Turaev-Viro codes.

Let Σ denote a hyperbolic surface that is used to define the hyperbolic Turaev-Viro code. Since the MCG can be generated by the Dehn twists around the handles of Σ, to implement an arbitrary geometric gate it suffices to be able to implement Dehn twists around the handles of Σ [38].

In Section 3.3, we construct specific diffeomorphisms that correspond to the basic Dehn twists, which then can be used to carry out Step 1 of the above procedure. Next, in Section 3.4, we introduce the local finite depth quantum circuit that converts two given triangulations to one another. By combining the results of these two sections and following the above procedure, one can implement the representation of any basic Dehn twists on the code space $\mathcal{H}_\Lambda(\Sigma)$, and hence implement any geometric gate by a constant depth unitary circuit.

3.3 Continuous Maps for Dehn Twists

First we concentrate on the $g = 2$ case. After developing the maps for the simplest case, we show how these maps can be generalized for a surface of arbitrary genus.



(a)

(b)

(c)

Figure 3.3: (a) The canonical octagon plotted on the Poincare disk model of the hyperbolic plane. Edges with the same color (label) have to be identified such that the arrows on the two edges line up to each other. (b) The Σ_2 surface resulting from identifying the edges of the canonical octagon. (c) Tilling of the hyperbolic plane by the canonical octagon.

3.3.1 Dehn Twists on a Double torus

Let $\tilde{\Sigma}_2$ be an arbitrary genus 2 surface. As a surface with negative Euler characteristic, $\chi(\tilde{\Sigma}_2) = -2$, it admits a hyperbolic metric, i.e. a complete finite area Riemannian metric of constant negative curvature -1.

One way to define a hyperbolic metric on $\tilde{\Sigma}_2$ is to start with the regular hyperbolic octagon on $\mathbb{H}^2$ whose interior angles sum to 2π, known as the Fricke canonical polygon [142, 143]. If we identify every other edge as shown in Fig. 3.3a with the arrows specifying how the edges should be lined up, we obtain a genus $g = 2$ hyperbolic surface Σ_2 shown in Fig. 3.3b, which is homeomorphic to $\tilde{\Sigma}_2$. According to Ref. [144], the homeomorphism can be upgraded to a diffeomorphism and thus induces a hyperbolic metric on $\tilde{\Sigma}_2$. From now on, we concentrate on Σ_2, knowing that our statements about Σ_2 can be generalized to the $\tilde{\Sigma}_2$ surface using the aforementioned diffeomorphism.

As a result of the identification scheme, all vertices of the canonical polygon represent a single point on Σ_2 and thus the sides of the polygon correspond to closed loops with a common base point on Σ_2 (see Fig. 3.3b). Conversely, if we start with the double torus in Fig. 3.3b and cut the surface along these loops, we will obtain the octagon in Fig. 3.3a. In fact, these loops can be taken as the generators of the fundamental group $\pi_1(\Sigma_2)$,

$$\pi_1(\Sigma_2) = \langle \alpha_1, \beta_1, \alpha_2, \beta_2 \,|\, \prod_{i=1}^{2} \alpha_i \beta_i \alpha_i^{-1} \beta_i^{-1} = 1 \rangle. \tag{3.6}$$

An element of the MCG will take these loops to some other loops on Σ_2, and thus naturally induces a map over $\pi_1(\Sigma_2)$. More precisely, it can be shown [145, 146] that the mapping class group is isomorphic to the group of outer automorphisms of the fundamental group,

$$\text{MCG}(\Sigma_2) \approx \text{Out}(\pi_1(\Sigma_2)) = \text{Aut}(\pi_1(\Sigma_2))/\text{Inn}(\pi_1(\Sigma_2)), \tag{3.7}$$

where Aut(G) and Inn(G) denote the automorphism group and inner automorphism group of G respectively.

Let U denote an arbitrary element of the mapping class group and hence an equivalence class of homeomorphisms on Σ_2. Due to Eq. (3.7), U also corresponds to an equivalence class of automorphisms of $\pi_1(\Sigma_2)$. In the rest of this chapter, in an abuse of notation we use the same symbol U to denote both the equivalence classes of homeomorphisms and Out$(\pi_1(\Sigma_2))$ and also representatives of these classes.

Consider the canonical octagon on the hyperbolic plane. By attaching a copy of the octagon on each edge according to the identification rules and continuing this procedure indefinitely for the edges of the newly added octagons, one will end up with the $\{8, 8\}$ tiling of the hyperbolic plane (see Fig. 3.3c). The result can be used to define a covering map $p : \mathbb{H}^2 \longrightarrow \Sigma_2$ which along with $\mathbb{H}^2$ makes up the universal cover of Σ_2.

In what follows we provide explicit expressions for the homeomorphisms of Σ_2 to itself corresponding to the Dehn twists along the primary loops of Σ_2.

3.3.1.1 Dehn twists along the α and β loops

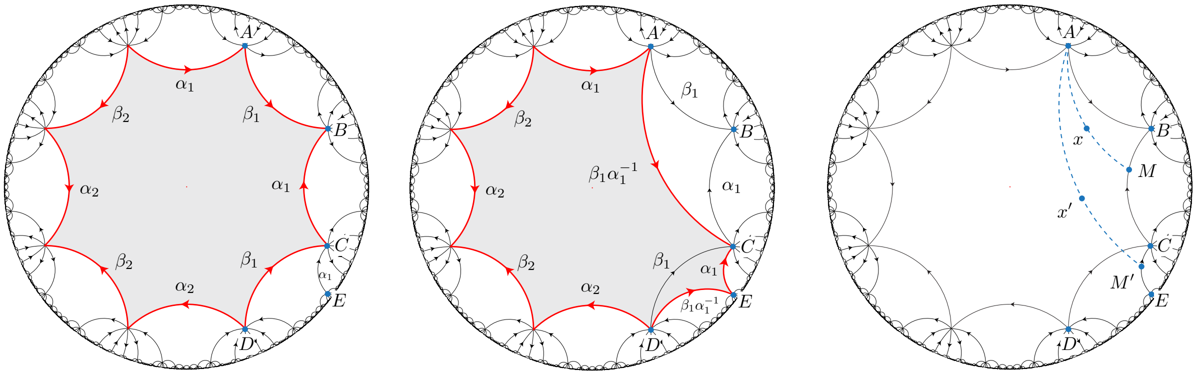
We start with the Dehn twist along α_1.

$$D_{\alpha_1} : \Sigma_2 \longrightarrow \Sigma_2 \tag{3.8}$$

To find an element of Aut$(\pi_1(\Sigma_2))$ which represents D_{α_1}, it is enough to see how it acts on the canonical loops of the Σ_2 surface. To this end we can use the Dehn surgery method described in Appendix A. However, for simplicity, first we push the α_1 slightly to the left to detach it from the α_2 and β_2 loops and consider the twist map along this new loop. Note that the Dehn twist D_{α_1} depends only on the isotopy class of α_1. Then, as one can verify by looking at Fig. 3.3b and using the Dehn surgery method, D_{α_1} maps β_1 to $\beta_1\alpha_1^{-1}$ and leaves all the other canonical loops invariant. Note that we use the left to right convention for loop multiplication; if f and g are two loops with a common base point, fg corresponds to a loop that traces f first and then g.

Instead of specifying D_{α_1}, we provide an explicit form for its lift to the covering space $D^*_{\alpha_1} : \mathbb{H}^2 \longrightarrow \mathbb{H}^2$ such that the diagram below commutes:

$$\begin{tikzcd}
\mathbb{H}^2 \arrow[r, "D^*_{\alpha_1}"] \arrow[d, "p"] & \mathbb{H}^2 \arrow[d, "p"] \\
\Sigma_2 \arrow[r, "D_{\alpha_1}"] & \Sigma_2
\end{tikzcd} \tag{3.9}$$



(a) (b) (c)

Figure 3.4: (a) The fundamental domain (the shaded region) can be identified with Σ_2. We define the action of the shearing map on the points in this region. The action of the map on the other points is defined according to the identification scheme. (b) Image of the fundamental domain under the shearing map $D^*_{\alpha_1}$. (c) Example of the action of the map: the point x inside the triangle ΔABC maps to the point x' inside ΔACE. See the text for details.

To this end, we define how $D^*_{\alpha_1}$ acts on the points of a fundamental domain. Its action on the other points of $\mathbb{H}^2$ are defined accordingly to ensure commutativity of the diagram in Eq. (3.9).

We take the fundamental domain to be the canonical octagon (shaded region in Fig. 3.4a). The action of $D^*_{\alpha_1}$ on the fundamental domain gives the sheared octagon shown in Fig. 3.4b. As one can easily verify, it transforms the β_1 loop to $\beta_1\alpha_1^{-1}$ as desired, while all the other loops remain unchanged. In particular note that the α_1 loop is mapped to itself and thus has not changed.

We define the map $D^*_{\alpha_1}$ more precisely as follows. We map ΔABC in Fig. 3.4b to ΔACE and ΔACD to ΔAED, where ΔABC denotes the hyperbolic triangle made by connecting A,B and C via geodesics on $\mathbb{H}^2$. All the other regions in the octagon are left untouched. Mapping ΔABC to ΔACE is done as follows. Consider an arbitrary point, x, inside ΔABC (see Fig. 3.4c). To find its image $x' = D^*_{\alpha_1}(x)$, first draw the geodesic line that passes through A and x, and continue it to find its crossing point M with the line BC. Now choose M' on line CE such that $|CM'|/|CE| = |BM|/|BC|$. By $|PQ|$ we mean the length of the geodesic line connecting P

and Q, measured using the hyperbolic metric. Finally, we choose x' on the AM' line such that $|Ax'|/|AM'| = |Ax|/|AM|$. ΔACD is mapped similarly to ΔAED.

It is clear that this map is continuous for the points inside the fundamental domain. It is also straightforward to check that this map is consistent with edge identification rules and thus is continuous throughout the $\mathbb{H}^2$ plane. Moreover, the diagram in Eq. (3.9) commutes by construction and hence D_{α_1} is continuous on Σ_2. Furthermore, as we explicitly verify in Appendix B, this map does not change the area of any region by more than a constant factor. This is an important property which allows the second step of our Dehn twist protocol, as we explain in subsequent sections.

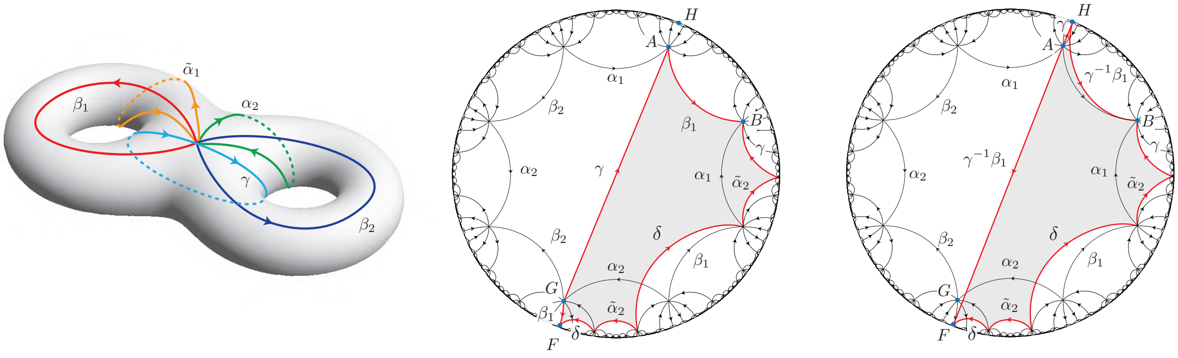
Dehn twists along β_1, α_2 and β_2 are defined in a similar manner.

3.3.1.2 Dehn twists along the γ loop

To generate all elements of the MCG we need the Dehn twist along the γ loop as well (see Fig. 3.5a). The canonical octagon which we used to define D_{α_1} has two important features: first, D_{α_1} only changes the β_1 sides while leaving other sides of the octagon invariant; second, α_1 and β_1 were neighboring sides of the octagon. However, since both β_1 and β_2 transform non-trivially under D_γ and since the γ loop is not one of the polygon's sides, the action of D_γ on the canonical octagon is not as simple as D_{α_1} and looks rather complicated. Therefore to construct D_γ, it is easier to work with a different fundamental domain. To find the appropriate fundamental domain, we cut Σ_2 along a new set of loops rather than the standard α's and β's.

Let $\tilde{\alpha}_i$ denote the α_i loop translated through β_i:

$$\tilde{\alpha}_i = \beta_i \alpha_i \beta_i^{-1}. \tag{3.10}$$



(a) (b) (c)

Figure 3.5: (a) $\tilde{\alpha}_1 = \beta_1\alpha_1\beta_1^{-1}$ and γ loops on the Σ_2 surface. (b) The shaded region can be taken as the fundamental domain of the covering map. (c) Image of the the shaded region in (b) under the shearing map D_γ^*.

$\tilde{\alpha}_1$ is illustrated in Fig. 3.5a. Note that we can write γ as:

$$\gamma = \alpha_2^{-1}\tilde{\alpha}_1 = \alpha_2^{-1}\beta_1\alpha_1\beta_1^{-1}. \tag{3.11}$$

We also define δ as:

$$\delta = \beta_2\beta_1, \tag{3.12}$$

which represents a loop that encircles both holes of Σ_2.

As was the case for the α_1 Dehn twist, to find an automorphism of $\pi_1(\Sigma_2)$ corresponding to D_γ first we push the γ loop shown in Fig. 3.5a slightly to the right and then use the Dehn surgery method to find its action on various loops. We remark that if we used another loop, e.g. if we pushed the γ loop slightly to the left instead, we would find the same automorphism up to an action of Inn(π_1). Note that due to Eq. (3.7), all such maps represent the same element of the mapping class group.

The representative automorphism induced by D_γ on $\pi_1(\Sigma_2)$ can then be summarized in the following four equations:

$$\begin{aligned}
&D_\gamma(\alpha_1) = \alpha_1, \quad D_\gamma(\beta_1) = \gamma^{-1}\beta_1, \\
&D_\gamma(\tilde{\alpha}_2) = \tilde{\alpha}_2, \quad D_\gamma(\beta_2) = \beta_2\gamma.
\end{aligned} \tag{3.13}$$

Note that D_γ leaves the δ loop invariant. To find the appropriate fundamental domain, we trade the $\{\alpha_1, \beta_1, \alpha_2, \beta_2\}$ loops with $\{\gamma, \beta_1, \tilde{\alpha}_2, \delta\}$. The group relation in Eq. (3.6) can be expressed in terms of these loops as well:

$$\tilde{\alpha}_2^{-1}\delta^{-1}\tilde{\alpha}_2\delta\beta_1^{-1}\gamma\beta_1\gamma^{-1} = 1. \tag{3.14}$$

Note that $\{\gamma_1, \beta_1\}$ as well as $\{\tilde{\alpha}_2, \delta\}$ have algebraic intersection 1 while the two sets are mutually detached, i.e. the algebraic intersection number of a loop from the first set and a loop from the second set is 0. So the $\{\gamma, \beta_1, \tilde{\alpha}_2, \delta\}$ loops could have been taken as the primary loops of Σ_2 in the first place.

This in turn suggests using an octagon with its sides following the δ,$\tilde{\alpha}_2$,β_1 and γ loops. Such an irregular octagon is shown in Fig. 3.5b. As one can easily verify, this can be taken as the fundamental domain of the mapping p. Moreover, it has the features we are looking for: under the action of D_γ, only the β_1 loop gets deformed and, furthermore, the γ and β_1 loops are represented by neighboring sides of the polygon.

D^*_γ (the lift of D_γ to $\mathbb{H}^2$) shears the octagon shown in Fig. 3.5b to the one shown in Fig. 3.5c. More precisely, ΔFGA is mapped to ΔFAH and ΔFAB is deformed to ΔFHB. Mapping the triangles is done through the same procedure described in Section 3.3.1.1. The action of D^*_γ on the other points of $\mathbb{H}^2$ is then defined according to the identification rules. It is straightforward to verify continuity of the map. Moreover, as we verify in Appendix B, these maps do not change the area of any region of the surface by more than a constant factor.

Any element of the MCG of the double torus can be generated using D_{α_i}, D_{β_i} and D_γ. In the next section we discuss how these constructions generalize to higher genus surfaces.

3.3.2 Dehn Twists on Genus g surface

A hyperbolic genus g surface can be obtained by identifying every other edge of a $4g$-gon, whose angles sum to 2π, in hyperbolic space. The space of different hyperbolic metrics, Teichmüller space, corresponds to inequivalent choices of the locations of the vertices of the $4g$-gon [147]. Here we consider the canonical $4g$-gon, i.e. a regular $4g$-gon on $\mathbb{H}^2$. The sides of the polygon can be used to generate the fundamental group of the Σ_g surface:

$$\pi_1(\Sigma_g) = \langle \alpha_1, \beta_1, \cdots, \alpha_g, \beta_g \,|\, \prod_{i=1}^{g} \alpha_i \beta_i \alpha_i^{-1} \beta_i^{-1} = 1 \rangle. \tag{3.15}$$

The MCG of Σ_g can be generated by Dehn twists along α_i and β_i for $i = 1, \cdots, g$ and γ_i for $i = 1, \cdots, g-1$. γ_i can be written as,

$$\gamma_i = \alpha_{i+1}^{-1} \tilde{\alpha}_i. \tag{3.16}$$

where $\tilde{\alpha}_i$ is defined as in (3.10). Since our maps for the Dehn twists on the double torus modify only a specific corner of the polygon while leaving the other parts of it fixed, they generalize naturally to maps on the $4g$-gons. Also as in the previous $g = 2$ case, to construct D_{γ_i}, it is easier to work with an irregular g-gon. As an example, the $g = 3$ case is analyzed in more detail in the

Appendix C.

Furthermore, in Appendix B we show that this map does not change the area of any region by more than a constant bounded factor, even in the limit $g \to \infty$. As explained in the next section, this feature is important to ensure that the depth of the local geometry change circuit remains constant as one increases the code distance.

3.4 Change of triangulation

As described in Section 3.2, step (1) of our protocol permutes the qubits by applying the continuous shear map of Section 3.3 to the triangulation Λ. After the permutation, the original triangulation Λ is changed to a sheared triangulation Λ'. In order to return to the original Hilbert space $\mathcal{H}_\Lambda$ and hence reach a non-trivial unitary map preserving the code space, we need to switch the triangulation back from Λ' to Λ.

In this section, we devise a local unitary circuit to switch a Turaev-Viro code between two arbitrary triangulations Λ and Λ'. We consider Λ and Λ' to have the same number of vertices and edges for any given region, up to at most a constant factor, c [as illustrated in Fig. 3.6a]. Since the switching circuit can be parallelized by acting throughout the whole space at once, the depth of the circuit only depends on c. To present our algorithm it is more convenient to show the switch between the two dual trivalent graphs instead, as indicated by the thick blue lines in Fig. 3.6a.

For clarity, we drop the branching structure (previously indicated by arrows on the edges) of the graphs in this section. Many theories of interest, such as the Ising code and the Fibonacci code, do not require the branching structure.

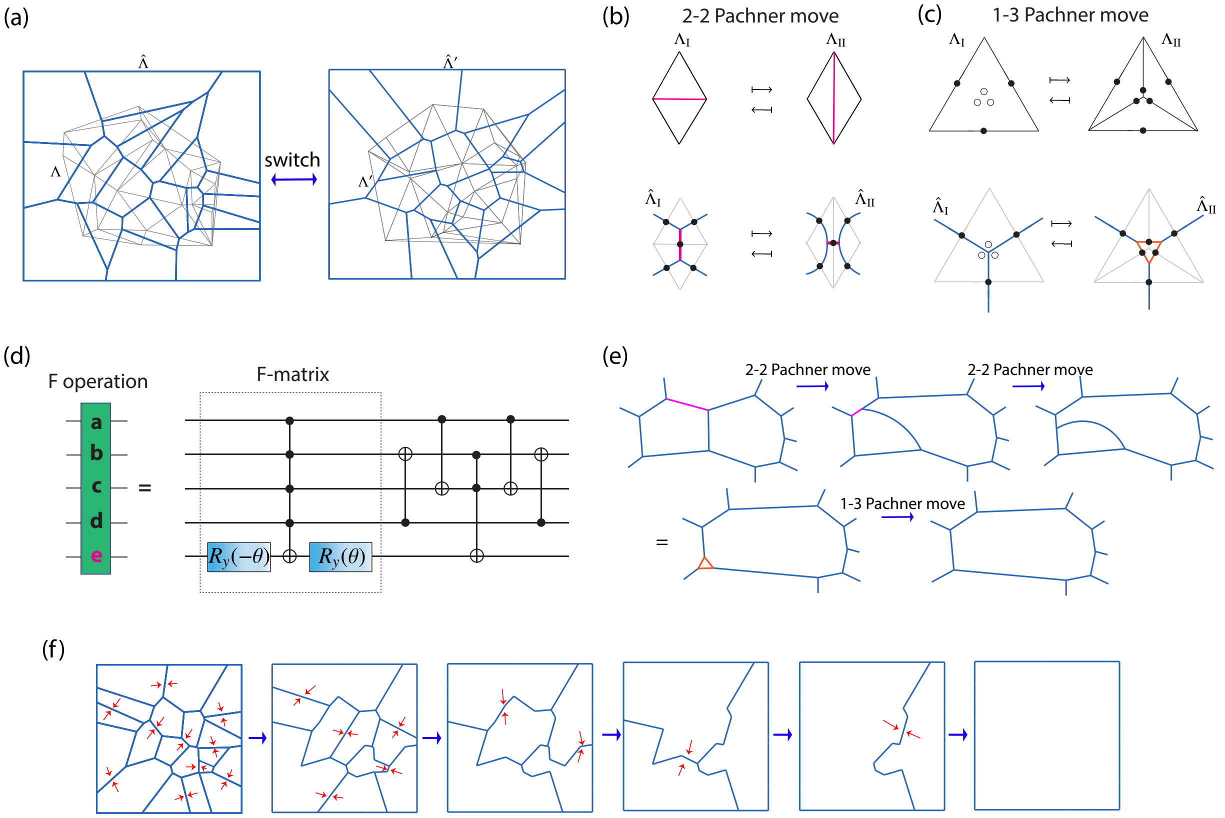
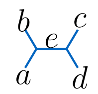
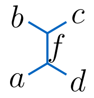


Figure 3.6: (a) Example of two different triangulations Λ and Λ' (and their dual graphs) with the same average vertex density, which we wish to switch between. (b,c) Elementary gadgets: 2-2 and 1-3 Pachner moves on the triangulation (represented by $\hat{\Lambda}_I$ to $\hat{\Lambda}_{\mathrm{II}}$ in the top row) and its dual trivalent graph (represented by $\hat{\Lambda}_I$ to $\hat{\Lambda}_{\mathrm{II}}$ in the bottom row). The black dots represent data qubits, while the white dots represent ancilla qubits initialized to $|0\rangle$. (d) The quantum circuit implementing the 2-2 Pachner move (F operation) in the Fibonacci model. (e) The "edge-sweep" algorithm to merge two edge-sharing plaquettes on a trivalent graph by $s-3$ steps of 2-2 Pachner moves followed by a 1-3 Pachner move in the end. (f) A parallel merging algorithm.

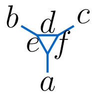
3.4.1 Gadgets

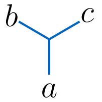
The elementary re-triangulation gadgets we use are associated with the 2-2 and 1-3 Pachner moves, shown in Fig. 3.6b and Fig. 3.6c. They correspond to unitary transformations that take the wave function defined on a triangulation $\Lambda_{\rm I}$ to the wave function defined on a different triangulation $\Lambda_{\rm II}$, which differs locally:





$$\Psi_{\hat{\Lambda}_{\rm I}}\left(\quad\right) = \sum_f F^{abc}_{def}\,\Psi_{\hat{\Lambda}_{\rm II}}\left(\quad\right) \tag{3.17}$$





$$\Psi_{\hat{\Lambda}_{\rm I}}\left(\quad\right) = [F^{abd}_{fce}]^{*}\sqrt{\frac{d_d d_f}{d_c}}\,\Psi_{\hat{\Lambda}_{\rm II}}\left(\quad\right), \tag{3.18}$$

where F^{abc}_{def} is the F-symbol that defines the Turaev-Viro code, and d_c, d_d, and d_f are the quantum dimensions of anyons labeled by c, d and f. For the 2-2 Pachner move [Eq. (3.17)], the state labels on the four external legs (a,b,c and d) are fixed, while the internal edge is flipped (both in the triangulation and trivalent graph) with the state labels changing from e to f. The 2-2 Pachner move can be implemented by unitary gates acting on physical qubits as will be illustrated below. For the 1-3 Pachner move [Eq. (3.18)], the state labels on the three external legs (a,b,c and d) are fixed, while a triangle with three new edges (e, d and f) is added at the center of the trivalent graph (from $\hat{\Lambda}_I$ to $\hat{\Lambda}_{\rm II}$). Correspondingly, in the original triangulation, a three-legged vertex is added in the center of a triangle (from Λ_I to $\Lambda_{\rm II}$) [see Fig. 3.6c]. These new edges come from ancilla qubits initialized at $|0\rangle$, which then get entangled into the code by the 1-3 Pachner move, which can be considered as a fine graining procedure. The reverse of this process is a coarse graining procedure. The 1-3 Pachner move can also be implemented by unitary gates, which can be decomposed into 2-2 Pachner moves and two other simple unitary gates (see Ref. [37,38] for details).

As an example we may consider the Fibonacci Turaev-Viro code. In this code, we have two simple objects in the unitary fusion category, labeled 0 and 1, with fusion rules $1 \times 1 = 0 + 1$, and the only non-trivial F-matrix is:

$$F_1^{111} = \begin{pmatrix} \phi^{-1} & \phi^{-\frac{1}{2}} \\ \phi^{-\frac{1}{2}} & -\phi^{-1} \end{pmatrix}, \tag{3.19}$$

where $\phi = \frac{\sqrt{5}+1}{2}$ is the golden ratio. All other F-symbols are either 1 or 0, depending on whether they are consistent with the fusion rules and Eq. (3.17). A specific quantum circuit implementing the 2-2 Pachner move (F-operations) in the Fibonacci code was presented in Ref. [136] and is shown in Fig. 3.6d. The circuit inside the dashed box is composed of a 5-qubit Toffoli gate sandwiched by two single-qubit rotations, which implements the F-matrix in (3.19). Here, $R_y(\pm\theta) = e^{\pm i\theta\sigma_y/2}$ represents single-qubit rotations about the y-axis with angle $\theta{=}\tan^{-1}(\phi^{-\frac{1}{2}})$. All the other maps are taken care of by the rest of the quantum circuit in panel (d). For the other widely considered case, the $\mathbb{Z}_2$ hyperbolic surface codes, the F-symbols and Pachner moves can be implemented via only CNOTs (see Ref. [38] for details).

Based on these gadgets, we introduce the following two lemmas about the trivalent graphs, which serve as additional gadgets for the main algorithm.

Lemma 3.1. *Two edge-sharing plaquettes on a trivalent graph can be merged into a single plaquette using $s - 2$ steps of Pachner moves, where s is the number of edges in the smaller*

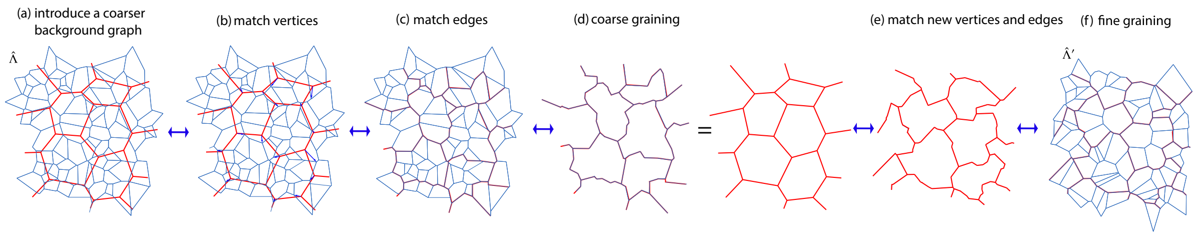


Figure 3.7: The main switching algorithm between trivalent graphs $\hat{\Lambda}$ and $\hat{\Lambda}'$.

plaquette (i.e., plaquette with fewer edges).

An "edge-sweeping" algorithm implements the above statement as shown in Fig. 3.6e with $s-3$ steps of 2-2 Pachner moves and a $1-3$ Pahcner move in the end.

Using the above plaquette-merging gadget, we can also demonstrate the following lemma when considering merging many plaquettes in parallel, as illustrated in Fig. 3.6f. In particular, one merges all mergeable neighboring pairs in each step.

Lemma 3.2. *A collection of m contiguous plaquettes can be merged with $\mathcal{O}\left(\log_2(m)\right)$ rounds of merging of the edge-sharing plaquettes. The depth of the algorithm is upper bounded by $\mathcal{O}\left(l\ \log_2(m)\right)$ steps of Pachner moves. Here, l is chosen to be the number of edges of the largest plaquette in this collection.*

3.4.2 The main algorithm

Here, we present the main algorithm to switch between the triangulations Λ and Λ' (and equivalently the dual graphs $\hat{\Lambda}$ and $\hat{\Lambda}'$). For the clarity of presentation, we choose a particular order by switching from $\hat{\Lambda}$ to $\hat{\Lambda}'$, although the algorithm and corresponding quantum circuit are reversible. The detailed algorithm is as follows:

1. We first introduce a trivalent graph $\hat{\Lambda}_C$ coarser than the two fine trivalent graphs $\hat{\Lambda}$ (starting

graph) and $\hat{\Lambda}'$ (target graph) we want to switch between. In particular, we consider a coarser graph $\hat{\Lambda}_C$ that encircles at most m plaquettes of the starting finer graph $\hat{\Lambda}$. We note that m needs to be bounded and independent of code distance d. We also require that any plaquette in the coarser graph $\hat{\Lambda}_C$ contains at least one plaquette of the finer graphs, as illustrated in Fig. 3.7a.

Now the coarser graph $\hat{\Lambda}_C$ also encircles at most m' plaquettes of the target finer graph $\hat{\Lambda}'$. As discussed in Section 3.3 and Appendix B, the area ratio for a given infinitesimal code patch before and after the application of the shear maps (belonging to $\hat{\Lambda}$ and $\hat{\Lambda}'$ respectively) is a bounded constant independent of d (and thus also independent of the number of physical qubits n). This ensures that the ratio m/m' is also bounded and independent of d. Since m is bounded and independent of d, m' is also a bounded constant independent of d. This is a crucial property to ensure the $\mathcal{O}(1)$ depth of the switching circuit and the logical gates.

2. We now match the vertices (v_C) of the coarser graph $\hat{\Lambda}_C$ with vertices (v) on the starting graph $\hat{\Lambda}$ by pinning the vertices of $\hat{\Lambda}_C$ to the closest vertices to them on $\hat{\Lambda}$, as shown in Fig. 3.7b. We can now bend each edge (e_C) on the coarser graph $\hat{\Lambda}_C$ to match with multiple edges (e) on the starting graph, as shown in Fig. 3.7c. In this way, each plaquette in the fine graph $\hat{\Lambda}$ is strictly enclosed in only one plaquette in the coarser graph $\hat{\Lambda}_C$. This step does not require any quantum operations and is done entirely in the classical software.

   Note that the deformation of the coarser graph $\hat{\Lambda}_C$ leads to a slight change of the maximal number of enclosed fine plaquettes in any coarser plaquette, i.e., m and m', into $\mathcal{O}(m)$ and $\mathcal{O}(m')$.

3. We coarse grain the finer graph $\hat{\Lambda}$ into the coarser graph $\hat{\Lambda}_C$ by using the parallel plaquette merging algorithms introduced in **Lemma 3.2**, as shown in Fig. 3.7d. This procedure takes $\mathcal{O}(\log_2(m))$ merging steps with $\mathcal{O}(l\ \log_2(m))$ time steps of Pachner moves, where l is the largest number of edges among all plaquettes in the starting finer graph $\hat{\Lambda}$.

4. We match vertices (v_C) and the edges (e_C) of the coarser graph $\hat{\Lambda}_C$ with those (v' and e') in the finer graph $\hat{\Lambda}'$ using the same procedure as above, as shown in Fig. 3.7e.

5. We fine grain the coarser graph $\hat{\Lambda}_C$ into the target finer graph $\hat{\Lambda}'$ by reversing the parallel merging algorithms in **Lemma 3.2**, as shown in Fig. 3.7f. This procedures takes $\mathcal{O}(\log_2(m'))$ splitting steps with $\mathcal{O}(l'\ \log_2(m'))$ time steps of Pachner moves, where l' is the largest number of edges among all plaquettes in the target finer graph $\hat{\Lambda}'$.

As we see the total time complexity of the switching algorithm is $\mathcal{O}[\max(l \log_2(m), l' \log_2(m'))]$, i.e., dominated by the larger complexity from the coarse graining and fine graining process. Since m, m', l, l' can all be bounded values independent of the code distance d (or equivalently graph size), we reach the following theorem:

Theorem 3.1. *If two triangulations Λ and Λ' have bounded ratios in terms of their vertices, edges, and plaquettes per unit area, then there exists a bounded depth circuit to convert between them with Pachner moves. The depth of the circuit is independent of the area of the surface, and therefore independent of the code distance d.*

We emphasize again that the requirement of the bounded ratios of vertices, edges and plaquettes per unit area is ensured by the bounded ratio of the areas corresponding to the infinitesimal code patches before and after the shearing maps, as discussed in Section 3.3 and Appendix B.

We also note that in the above discussions, we focused on proving the existence of a constant depth local unitary circuit, rather than giving the most efficient switching algorithm. To be more efficient, one does not need to follow the fine→coarse→fine pattern, but can rather directly find the shortest circuit to directly switch between the two fine graphs via Pachner moves.

3.5 Fault tolerance and space-time overhead

So far we have shown that a generating set of Dehn twists for the MCG of a genus g hyperbolic surface can be implemented by a constant depth unitary circuit, where the depth is independent of the code distance and therefore also the number of physical and logical qubits, n and k. In this section, we briefly discuss fault tolerance of these circuits.

Our circuit breaks up into two basic pieces: A permutation of the physical qubits and a local constant depth circuit that implements the retriangulation. Since local unitary circuits have a linear light cone, the latter, i.e. the local constant depth circuit, is intrinsically fault tolerant. So, to ensure the fault tolerance of our procedure, we concentrate on the first part which permutes the physical qubits.

The permutation in our maps requires qubits to be permuted over long distances. Due to its non-local nature, there are two main concerns in regard to the propagation of errors that we need to address: (1) What happens to the pre-existing local error strings? Is it possible for them to be enlarged to lengths of $\mathcal{O}(d)$? (2) Is it possible to introduce new non-local ($\mathcal{O}(d)$ long) error strings by a noisy implementation of the permutation circuit, e.g. noisy SWAP gates?

For a generic non-local permutation, both issues mentioned above could possibly arise. Nevertheless the permutations that we utilize have a special structure. To address the first issue, note that the continuous maps introduced in Section 3.3, take two points which are $\mathcal{O}(1)$ apart to new points which are still $\mathcal{O}(1)$ apart. This can be seen from the analysis carried out in Appendix B as well. However, to enlarge a local error string to a large error string of length $\mathcal{O}(d)$ we have to separate its endpoints by a factor of $\mathcal{O}(d)$. Thus, we can conclude that after the permutation, all pre-existing local error strings would remain local; at worst their length will be increased by a constant factor independent of d. Stated differently, our constant depth circuits map any operator with support in a spatial region $\mathcal{R}$ to an operator with support in a spatial region $\mathcal{R}'$, where the area of $\mathcal{R}$ and $\mathcal{R}'$ are related by a constant (independent of code distance) factor. Here the areas are with respect to the hyperbolic metric. In this sense, the whole circuit is also "locality-preserving," although strictly speaking the phrase "locality-preserving" is often reserved for the case where the constant factor is unity [134, 148].

Now we consider the possibility of introducing new non-local error strings during a noisy implementation of the permutation. Let's say the permutation is implemented by a set of noisy long range SWAPs. The important point to note is that while the SWAP operations are long ranged, they are still *low weight* operators. In particular, each SWAP operation acts on 2 qubits. On the other hand, a logical error string is a high weight operator, consisting of $\mathcal{O}(d)$ single qubit errors. So, if we assume the errors occur independently on different SWAPs, the possibility of introducing a logical error by a set of noisy SWAPs is still exponentially small in d. Therefore the second question can be answered in the negative as well.

Therefore, our Dehn twists and the corresponding logical gates are inherently protected from errors, in the sense that they do not stretch error strings by more than a constant factor, nor can they introduce error strings that have length more than a constant, independent of code distance.

However, if we apply the same collection of Dehn twists (logical gates) repetitively in the same region of the manifold, in the worst case the length of error strings could grow exponentially with the number of logical gates being applied. In the absence of measurement noise, the error string can be decoded and corrected in $\mathcal{O}(1)$ time [138] immediately after the application of a single logical gate (here we have ignored the classical computation time of the decoder which still typically scales with system size). Hence the computation scheme will have an $\mathcal{O}(1)$ (constant) time overhead. However, in the presence of measurement noise, the error string cannot be immediately decoded and corrected in $\mathcal{O}(1)$ time, so the growth of such a string would be inevitable. After performing $\mathcal{O}(\log d)$ logical gates in the same region without any measurement or error correction in between, the error string may grow to a length of $\mathcal{O}(d)$, which will cause the decoder to fail. Therefore, one has to insert $\mathcal{O}(d)$ rounds of measurements, decoding, and error corrections for every $\mathcal{O}(\log d)$ of logical gates in the same region. This suggests a sub-linear overhead $\mathcal{O}(d/\log d)$ in the computational time when repetitively applying logical gates in the same region, if the measurement error is taken into account. It may be possible to further reduce such overhead by some additional tricks, at least for certain types of logical circuits, but there may still be such a sub-linear overhead in the most generic situation. We note that the above $\mathcal{O}(d/\log d)$ time overhead is an estimate suggested by the considerations stated above; further work is required to develop efficient decoders to concretely demonstrate the validity of this estimate.

The above statements for the asymptotic space-time resource costs have assumed that the codes have a finite error threshold and that an efficient decoder exists and requires $\mathcal{O}(d)$ rounds of syndrome measurements to decode errors. We note that finding such a decoder is still an active filed of research [138, 141, 149].

Chapter 4: Measurement induced topological entanglement phase transitions

Generic unitary dynamics drive quantum many-body systems into highly entangled states characterized by volume-law scaling of subsystem entanglement entropies. When this dynamics is intercepted by rapid local measurements, individual quantum trajectories are expected to collapse into low entanglement states characterized by area-law scaling of subsystem entanglement entropies. Recently, it was discovered that, at least in a class of models, these two phases are separated by a scale-invariant "critical point" at a finite measurement rate [150–152]. Several aspects of this transition and its generalizations have been studied recently [150–190].

In the limit of infinitely rapid local measurements, the state of the system crucially depends on the choice of measurement basis. Assuming one measures only commuting single-qubit operators, the wave-function collapses into an unentangled trivial product-state. However, if one chooses to measure a set of stabilizer operators that stabilize a topological or a symmetry protected topological (SPT) wave-function, the resulting state, despite having area-law scaling of entanglement as well, would be topologically distinct from the product state [191, 192].

This raises the question of whether the notion of a topological phase is well-defined in random quantum circuits that include both unitary dynamics and local measurements. Given that the symmetry plays an important role in the topological classification of phases of matter, we consider two different setups of hybrid random circuit. First we investigate the question of whether one can sustain symmetry protected topological order in hybrid random circuit models. To this end, we consider a (1+1)D quantum random circuit model which is constrained by a $\mathbb{Z}_2 \times \mathbb{Z}_2$ symmetry. The symmetry is chosen so the circuit dynamics could sustain a specific SPT state known as the cluster state. In the next part, we consider (2+1)D hybrid random circuits without any symmetry constraint which are designed to be able to sustain long range topological order similar to that of the toric code.

Entanglement phase transitions involving topological or SPT phases, also seem to be closely related to quantum error correction. In particular, the rapid stabilizer measurements are reminiscent of syndrome measurements in active error correction schemes. Moreover, random single qubit measurements can be viewed as faulty syndrome measurements or qubit decoherence, while unitary dynamics models the random noise affecting the qubits. In this context, "entanglement phase transitions" could be related to "error-thresholds" beyond which the long range entanglement structure of the code space, which is responsible for the topological protection of the encoded information, is entirely lost, hence rendering recovery of logical information impossible. Within this framechapter, our results might have natural applications to quantum error correcting codes. Note that this is a different analogy to quantum error correction than the one presented in Ref. [154, 157], where the volume law phase is considered to be a quantum error correcting code.

4.1 (1+1)D symmetric monitored random quantum circuits

4.1.1 Circuit Model

We study a family of (1+1)D random quantum circuits that realize quantum trajectories extrapolating between wave functions in an SPT phase, a trivial product state, and a volume-law

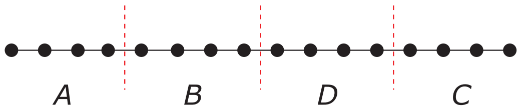


Figure 4.1: The 1D chain cutting used to define the generalized topological entanglement entropy.

entangled phase.

We take our SPT to be the $\mathbb{Z}_2 \times \mathbb{Z}_2$ symmetry protected phase realized by the cluster model defined on an open chain of N qubits (we take N even throughout) in (1+1)D [193, 194],

$$H_0 = -\sum_{i=2}^{N-1} X_{i-1} Z_i X_{i+1}, \tag{4.1}$$

where X_i and Z_i denote Pauli matrices. Note that all terms commute with each other and therefore this model is exactly solvable. This model realizes a SPT phase [195–197] protected by the $\mathbb{Z}_2 \times \mathbb{Z}_2$ symmetry generated by

$$G_1 = \prod_{i \text{ is even}} Z_i \; ; \; G_2 = \prod_{i \text{ is odd}} Z_i. \tag{4.2}$$

We say an eigenstate of H_0 is a symmetry invariant eigenstate if it is an eigenstate of all terms in H_0 as well as G_1 and G_2. All symmetry invariant eigenstates within the same symmetry sector can be related to each other by a symmetry-preserving constant depth local unitary circuit.

On an open chain, a particular generalization of the topological entanglement entropy [19, 194, 198, 199] can be used as an order parameter for this SPT phase. Consider dividing the system as shown in Fig. 4.1. The generalized topological entanglement entropy S_{topo} is defined as

$$S_{\text{topo}} \equiv S_{AB} + S_{BC} - S_B - S_{ABC}. \tag{4.3}$$

S_{AB} stands for the von Neumann entanglement entropy of the region $A \cup B$ in the chain. Other terms are defined similarly. One can show that for all symmetry invariant eigenstates of H_0, $S_{\text{topo}} = 2$.

To realize a wave function in this SPT phase, that is, a symmetry invariant eigenstate of H_0, we can for example use a quantum circuit that starts with an arbitrary eigenstate of G_1 and G_2 and then proceed to measure all stabilizer operators $g_i \equiv X_{i-1}Z_iX_{i+1}$.

To realize wave functions in the trivial phase, we use a quantum circuit that measures all single qubit operators in the Z_i basis. The choice of the single qubit measurement basis Z_i is fixed by demanding all measurement operators commute with the symmetry generators G_1 and G_2. All wave functions in the trivial phase have $S_{\text{topo}} = 0$.

To realize wave functions in the volume law phase, we use random Clifford unitary gates that are allowed by the symmetry. The simplest class of gates to consider would be two qubit nearest-neighbor random unitaries. However, due to the symmetry restrictions, this set is not effective in entangling the qubits. Ergo, we work with three-qubit random unitary gates.

We are now in a position to construct our full quantum circuit model: We start with the $|0\rangle^{\otimes N}$ state. In each updating step we either: (a) apply a random 3-qubit Clifford unitary between qubits $i-1$, i and $i+1$ with probability p_u, for a random i drawn from $2, \cdots, N-1$, (b) measure the single qubit operator Z_i with probability p_s, for a random i drawn from $1, \cdots, N$, or (c) measure the stabilizer $g_i \equiv X_{i-1}Z_iX_{i+1}$ with probability $p_t = 1 - p_s - p_u$, for a random i drawn from $2, \cdots, N-1$. A time step is defined as N consecutive updating steps. A typical snapshot of the circuit is shown in Fig. 4.2a.

In the limiting case $p_u = 1$ and $p_s = 0$, the random unitary circuit drives the system into a volume law phase, whereas for the other two limiting cases, i.e. $p_u = 0, p_s = 0$ and $p_u = 0, p_s =$

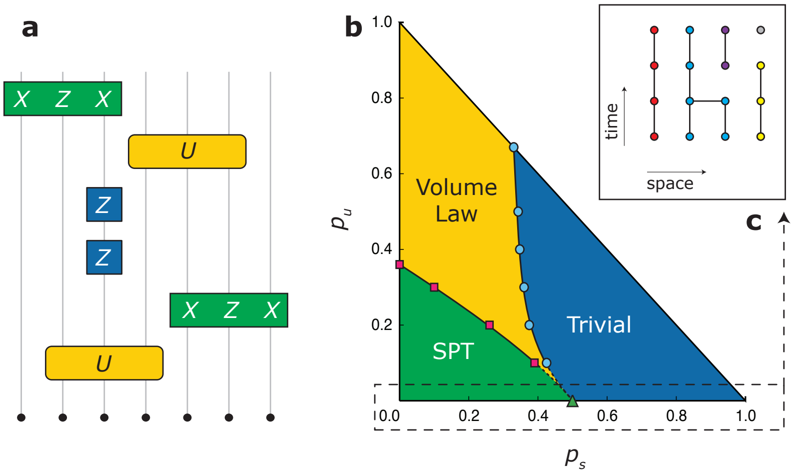


Figure 4.2: Schematic of the circuit and its corresponding phase diagram. **a**, Schematic diagram of a typical quantum circuit. Yellow (light) boxes corresponds to a three qubit random Clifford unitary, blue and green boxes represent projective measurements. **b**, The phase diagram describing the entanglement structure of the steady state. Red squares and blue circles are obtained from numerical simulations, while the rest of the phase boundaries are extrapolated. **c**, Mapping the dynamics of the random circuit on the $p_u = 0$ axis to the 2D percolation on a square lattice.

1, the system is in an area law phase, one with SPT order and the other without. Using suitably defined order parameters, we discover study the rest of the phase diagram.

4.1.2 Order Parameters

We detect the presence of the different phases in several distinct ways. First, at each time step we calculate S_{topo}, averaged over quantum trajectories, and run the circuit until a steady state value is obtained.

In addition to S_{topo}, to detect the phase transition from the area to volume law phase we extensively use the order parameter originally introduced in Ref. [155]. To do so, first we run the circuit for time $2N$ to reach the steady state. Then, we entangle an ancilla qubit to the two qubits in the middle of the chain by measuring the following stabilizers,

$$Z_{N/2-1}Z_a, \quad Z_{N/2+1}Z_a, \quad X_{N/2-1}X_aX_{N/2+1}, \tag{4.4}$$

where X_a and Z_a act on the ancilla qubit. Note that all three stabilizers commute with the symmetry generators G_1 and G_2. Next, we let the circuit run for an extra $\mathcal{O}(N)$ time steps, and then measure the entanglement entropy of the ancilla qubit. As shown in Ref. [155], if the system is in the area law phase, the ancillla's entanglement entropy S_a should be zero by the time we measure it while in a volume law phase, the ancilla should be still entangled with the system.

We also use a slightly modified version of the ancilla order parameter [155], which we call the scrambled ancilla order parameter denoted by $\tilde{S}_a$, such that instead of 1 ancilla we use 10 and instead of measuring the stabilizers listed in equation (4.4) the ancillas are entangled to the system via 10 time steps of a scrambling circuit, where at each updating step a random (non-symmetric) 3-qubit Clifford gate is applied to three randomly drawn qubits. As was the case for S_a, we measure the entropy of the ancilla subsystem after the qubit chain evolves $O(N)$ time steps under the symmetric random circuit. While in the trivial phase the ancilla subsystem would have been entirely disentangled from the qubit chain, giving $\tilde{S}_a = 0$, in the SPT phase the ancilla subsystem should have remained entangled to the two edge degrees of freedom which are protected by the symmetry, resulting in $\tilde{S}_a = 2$. In the volume law phase the ancilla subsystem should remain entangled to the bulk as well and hence $\tilde{S}_a > 2$.

It turns out that compared to $\tilde{S}_a$ and S_{topo}, S_a shows a sharper SPT to volume law phase transition when $p_u > 0$ –and hence it is used to extract the corresponding critical exponents– but is unable to detect the topological phase transition at $p_u = 0$. On the other hand, $\tilde{S}_a$ can be used as an experimentally accessible probe to detect the phase transition at $p_u = 0$.

We note that a type of Edwards-Anderson glass order parameter can also be used to detect the topological phase, although it cannot distinguish the trivial and volume law phases.

Finally, we note that the random quantum circuits studied here, viewed as a quantum channel, eventually transform the initial state of the system into the maximally mixed state allowed by the symmetry (see Appendix E for a proof and a bound on how fast this happens). Therefore, the steady state expectation value of *any* operator stays the same throughout the phase diagram and thus cannot serve as an order parameter.

4.1.3 Mapping The Case Without Unitary Dynamics $p_u = 0$ to Classical Percolation

Here we show how to map the entire $p_u = 0$ line in the random circuit presented above to two copies of a classical $2D$ percolation problem on a square lattice. This percolation model is non-standard, although our numerical results indicate that it has the same critical properties as the standard classical percolation model on the square lattice.

Let us divide the operators measured by the random circuit into two sets. One set, which we call the odd site operators, is comprised of single qubit operators Z_i for odd i alongside the stabilizers g_j which *end* on the odd sites, i.e. for even j. The even site operators are defined analogously. Note that each member of one set commutes with all elements of the other set.

Let us focus on the measurements of odd site operators. Consider the $N/2 \times M$ square lattice as shown in Fig. 4.2c, where M is the total number of updating steps in the circuit. We call this lattice the odd sites' percolation lattice. The $N/2$ vertices on each row corresponds to the odd sites of the system and we label them accordingly. The vertical (horizontal) links ending (residing) on the m'th row are related to the Z_i (g_j) measurements in the m'th step of the circuit in the following way: if Z_i *is not* measured at updating step m, we draw a vertical link between the $(i, m-1)$ and (i, m) vertices. Also if the stabilizer g_j *is* measured at step m, we draw a horizontal link between the $(j-1, m)$ and $(j+1, m)$ vertices. At the end, we assign a unique color to each connected cluster of vertices. We construct the even sites' percolation lattice analogously. The randomness of the quantum circuit translates into random connections in the percolation lattices.

The entanglement structure of the system at step M can be extracted from the colors of the vertices on the last row of the two aforementioned percolation lattices. As the following proposition makes precise, qubits of the same color make up their own SPT state:

Proposition 4.1. *Group the qubits based on their color on the last row of the percolation lattice. Let* $A^j = \{q_i\}_{i=1}^n$ *denote the ordered set of qubit indices corresponding to* j*'th color; that is, the* q_i *label a set of qubits all with the same color at step* M*. Then, up to a minus sign, the operators that stabilize the state of the system at step* M *are of the following form,*

$$\prod_{i=1}^{n} Z_{q_i} \quad \textit{and} \quad g_{q_i,q_{i+1}} \quad \textit{for } i = 1, 2, \cdots, n-1, \tag{4.5}$$

where $g_{q_i,q_{i+1}}$ *is defined as*

$$g_{i,j} = X_i \left[\prod_{k=0}^{\frac{j-i}{2}-1} Z_{i+2k+1} \right] X_j. \tag{4.6}$$

By considering similarly defined stabilizer operators for all different colors (A^j's with different j), we get a complete set of stabilizers that specify the state of the system. The proof of Proposition 4.1 is left for Appendix F.

As shown in Lemma D.1 in Appendix D, the minus sign ambiguity in Proposition 4.1 has no bearing on the entanglement spectrum of the system's state. Thus the percolation lattices exactly determine the (von Neumann or Rènyi) entanglement entropy for any subset of qubits.

4.1.4 The duality mapping p_s to $1 - p_s$ and vice versa

When $p_u = 0$, there is a local duality between circuits with paramter p_s and circuits with parameter $1 - p_s$. As we will see, this duality explains why the phase transition point is pinned at $p_c = 1/2$.

For simplicity, consider the system with periodic boundary conditions. Let us define the Clifford unitary U_d such that for $i = 1, \cdots, N$,

$$U_d \, X_i \, U_d^\dagger = X_i \tag{4.7}$$

$$U_d \, Z_i \, U_d^\dagger = X_{i-1} \, Z_i \, X_{i+1} \tag{4.8}$$

Note that under U_d, the stabilizer g_i transforms as

$$U_d \, g_i \, U_d^\dagger = Z_i. \tag{4.9}$$

equation (4.9) and equation (4.8) together show that the ensemble of random quantum circuits at p_s (and $p_u = 0$) is mapped to the ensemble of random quantum circuit at $1 - p_s$ (and $p_u = 0$) under U_d. However, the unitary U_d is not local, i.e. it cannot be written as the tensor product of on-site unitaries and therefore does not keep the entanglement structure invariant. Nonetheless, it is clear from equation (4.7) and equation (4.8) that U_d maps local stabilizers to local stabilizers.

Since the entanglement in stabilizer states is related to the number of independent stabilizers that traverse the boundary of a region [200], one can still say that U_d maps a state with the area-law entanglement to an area-law entangled state. Hence, if there exists a continuous phase transition which has logarithmic entanglement scaling, it has to occur at $p_s = p_c = 1/2$.

4.1.5 Numerical Results

We start by briefly reviewing the quantities we numerically calculate to obtain the phase diagram and to characterize the critical phase boundaries.

A signature of criticality in (1+1)D systems is the logarithmic scaling of the entanglement entropy. Thus, we calculate the entanglement entropy at the tth time step (which corresponds to tN updating steps), $S(x, L; t)$ of a subsystem of length x for a system of total length $L = N$, averaged over all of the quantum trajectories of the circuit.

In the large time limit, this averaged entanglement entropy saturates to a logarithmic form at the phase transitions as in (1+1)D CFTs [201]:

$$S(x, L) = a_x \log\left(\frac{L}{\pi} \sin \frac{\pi x}{L}\right) + b. \tag{4.10}$$

We can also characterize the entanglement growth with time. At criticality, for timescales much smaller than the saturation time we have,

$$S(x, L; t) = a_t \log(t) + b'. \tag{4.11}$$

Note that as opposed to unitary CFTs the coefficient of the logarithmic scaling a_x is not given by the central charge of any underlying CFT. In the context of the area law to volume law transition, Ref. [202] provides an appealing interpretation of a_x and a_t as universal quantities given by the scaling dimension of certain "boundary condition changing" operators. b and b' are non-universal constants

Throughout the phase boundaries, we find $a_x = a_t$ within the margin of error, which is consistent with a dynamical exponent $z = 1$, as the entanglement growth rate is similar along time and space directions.

We can use the averaged topological entanglement entropy, S_{topo} as the order parameter to distinguish the three different phases: S_{topo} would be extensive in the volume law phase, while in the thermodynamic limit it should converge to values 2 and 0 in the topological and trivial phases respectively. Let $S_{\text{topo}}(p, L)$ denote the steady state value of S_{topo} when some tuning parameter (e.g. single qubit measurement probability) is p and system size is L. On general grounds, we expect the following scaling form in the vicinity of the critical point,

$$S_{topo}(p, L) = F((p - p_c)L^{1/\nu}), \tag{4.12}$$

where $F(x)$ is some unknown function, p_c is the critical value of tuning parameter p, and ν is the correlation length critical exponent, $\xi \propto |p - p_c|^{-\nu}$.

As explained in Section 4.1.2, the entanglement entropy of a suitably entangled ancilla system, S_a or $\tilde{S}_a$ can also be used as the order parameter to distinguish the volume law phase from the other two area law phases. Assuming the dynamical exponent $z = 1$, for the ancilla entropy S_a we have [155],

$$S_a(p, L, t) = G((p - p_c)L^{1/\nu}, t/L), \tag{4.13}$$

where $G(x)$ is some unknown function. $\tilde{S}_a$ has a similar scaling form.

We now present our numerical results. We study system sizes up to 512 qubits and average over 10^5 random quantum trajectories. We start with the $|0\rangle^{\otimes N}$ state and let the circuit run for $2N$ time steps for the system to reach the steady state. We have explicitly verified that saturation is reached before $t = 2N$. After entangling the ancilla qubit, we simulate the system for an additional $\mathcal{O}(N)$ time steps to calculate S_a (as explained above).

Fig. 4.3 shows numerical results along the $p_u = 0$ line. Fig. 4.3a and c show the steady state value of S_{topo} and $\tilde{S}_a$ versus p_s for different system sizes. As is evident from both diagrams, there is a clear continuous phase transition at $p_c = 1/2$ in the thermodynamic limit. This is consistent with what we expected from the duality argument presented in Section 4.1.4. Interestingly we find that S_a seems to be unable to capture the area-law to area-law phase transition at $p_u = 0$, at least for numerically accessible systems sizes. On the other hand, From collapsing the data near the critical point $p_c = 1/2$, we find $\nu = 4/3$ results in a near perfect collapse (see Fig. 4.3c and d).

Fig. 4.3e shows the steady state value of entanglement entropy $S(x)$ of the subregion $[1, x]$ at the critical point $p_u = 0$ and $p_s = 1/2$, for $L = 512$. As shown, the entanglement entropy fits the CFT form of equation (4.10) with $a_x = 0.20(1)$.

Fig. 4.3f shows the entanglement entropy of the half chain versus time at $p = p_c$ for different chain sizes. The entanglement entropy grows logarithmically with time, until the finite size effects show up. By comparing the corresponding fitted analytical expressions we find $a_t = a_x = 0.20(1)$.

We now proceed to the case with unitary dynamics $p_u \neq 0$. Fig. 4.4 shows S_{topo}, S_a and $\tilde{S}_a$ versus p_s for the fixed value of $p_u = 0.3$. For $p_s = 0$, the system is in the topological phase as

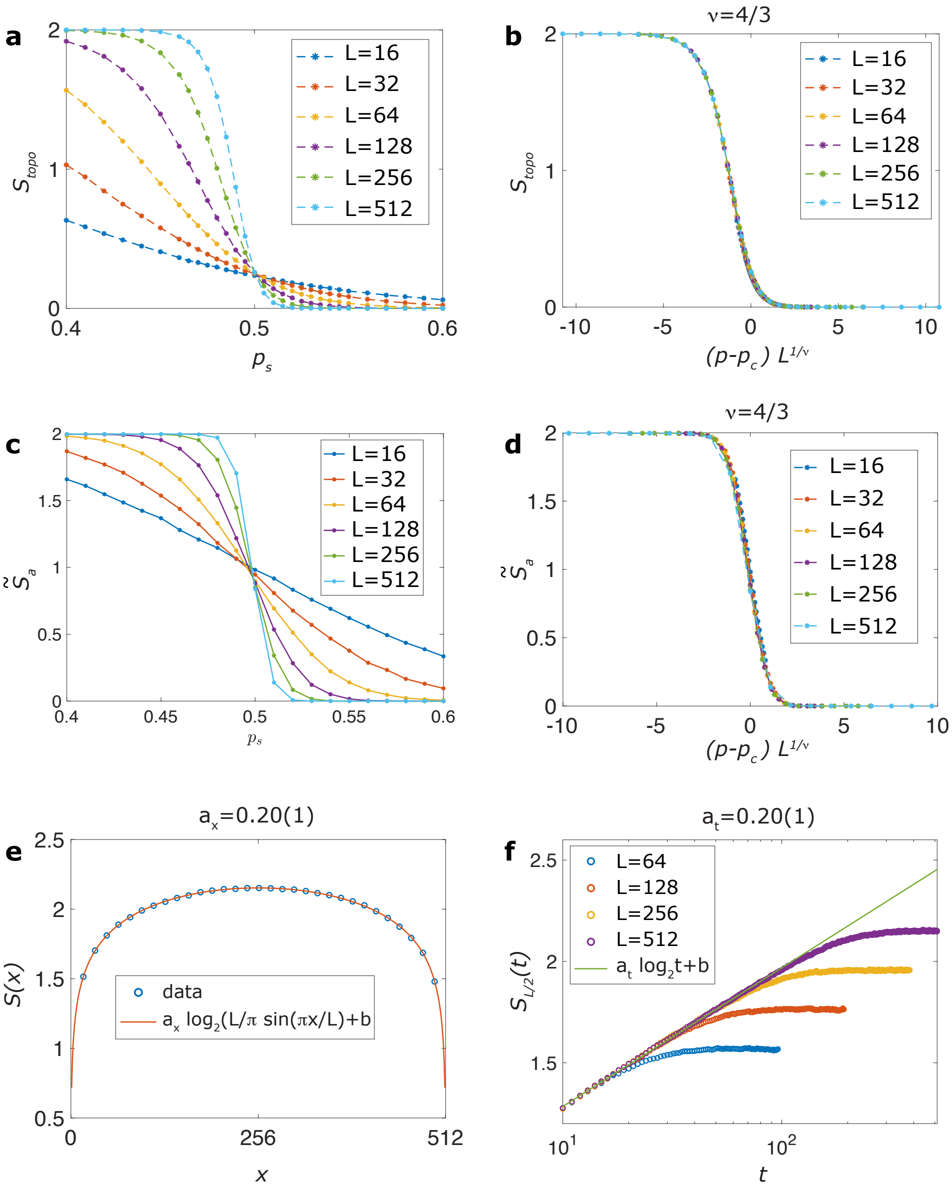


Figure 4.3: The numerical results pertaining to the tricritical point at $p_u = 0$. **a**, Topological entanglement entropy S_{topo} near the tricritical point versus single qubit measurement probability p_s. **b**, Scaling collapse of the data in panel a. **c**, Ancilla entropy $\tilde{S}_a$ measured $t = N$ time steps after scrambling. **d**, Scaling collapse of the data in panel c. **e**, The entanglement entropy of the $[0, x]$ segment of the chain, $S(x, L)$, at late times for $p = p_c$ and $L = 512$. **f**, The entanglement entropy of the half-chain versus time for $p_s = p_c$. All entropies are in units of $\log 2$. See Appendix F for an analytical derivation of the a coefficient using the percolation map.

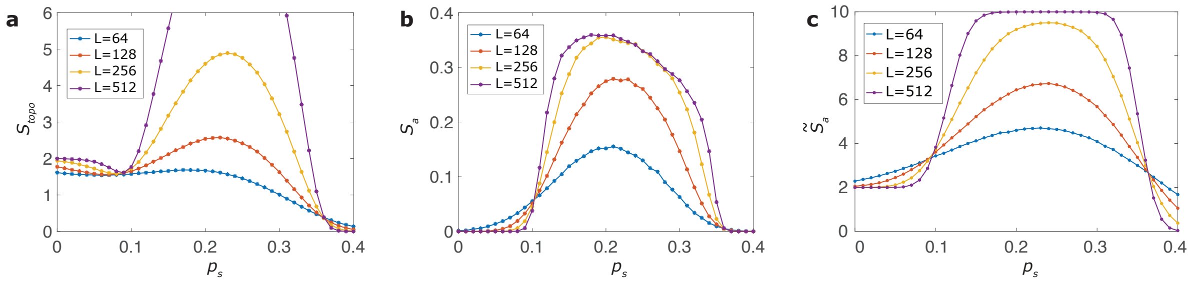


Figure 4.4: The phase transitions across the $p_u = 0.3$ line. **a**, S_{topo} verses p_s. **b**, The ancilla entropy S_a measured $t = N$ time steps after it was entangled, versus p_s. **c**, The ancilla entropy S_a measured $t = N$ time steps after it was entangled, versus p_s. In all three panels, the first crossing corresponds to the phase transition from the SPT phase into the volume law phase while the second crossing corresponds to the phase transition from the volume law phase to the trivial phase. The critical points are marked on the phase diagram in Fig.4.2b as well.

can be seen from Fig. 4.4a. By increasing p_s, the entropies exhibit a continuous phase transition to the volume law phase at first and then another continuous phase transition to the trivial phase.

By using analogous plots for different values of p_u, we can determine the 2D phase diagram in the (p_s, p_u) space. The result is illustrated in Fig. 4.2b. Note that since the probability of measuring a stabilizer is $1 - p_u - p_s$, the phase diagram is restricted to the region $p_u + p_s \leq 1$. The data points on the plot have been extracted using numerical simulations and then the schematic phase diagram is drawn based on them.

The SPT/volume law phase boundary intersects the p_u axis at $p_u = 0.355(3)$ and the volume law/trivial phase boundary ends at $p_u = 0.663(4)$ on the $p_u + p_s = 1$ line. Our numerical simulations demonstrate that the volume law phase still exists for p_u as low as 0.1. Unfortunately, clearly detecting the SPT to volume law transition requires increasingly large system sizes as p_u is lowered. Therefore, we extrapolate the phase diagram for smaller values of p_u. By following the trend of the data points, it appears that the volume law phase survives all the way down to $p_u = 0$, hence suggesting that the critical point at $p_s = 0.5$ and $p_u = 0$ is actually a tricritical point. This in turn means that at $p_s = 1/2$, arbitrarily sparse random Clifford gates in the quantum circuit can still drive the system into the volume law phase.

By using the scaling form in equation (4.12) and collapsing the data, we can extract the correlation length critical exponent ν along the phase boundaries. Taking into account the margins of error, our numerical results are consistent with $\nu = 4/3$ everywhere along the phase boundaries. However $a_x = a_t$ changes significantly along the phase boundaries at the largest system sizes we have studied. If the $a_x = a_t$ that we extract are indeed close to their values in the thermodynamic limit, this suggests that the volume to area law critical lines may be related to two copies of the classical percolation fixed point by marginal deformations.

4.2 (2+1)D topological monitored random quantum circuits

In the second part of the chapter, we consider a class of $(2+1)$D random quantum circuits that extrapolate between (1) a topologically ordered phase, characterized by non-zero topological entanglement entropy (TEE) [203, 204] and realized by measuring the $\mathbb{Z}_2$ toric code stabilizers [2], (2) a volume law entangled phase realized by random Clifford unitaries and, (3) the trivial, area law phase realized by single-site measurements. In contrast to the previous section, no symmetry restriction has been assumed.

4.2.1 Circuit Models

We consider $N = L^2$ qubits laid on the vertices of a two dimensional periodic square lattice of linear length $L_x = L_y = L$. Three different sets of gates are considered where each gate set, when applied exclusively, drives the system into one of distinct phases discussed above.

For the topological phase, we consider measurements corresponding to toric code stabilizers,

$$g_{i,j} = \begin{cases} X_{i,j}\ X_{i+1,j}\ X_{i,j+1}\ X_{i+1,j+1} & i+j \text{ is even} \\ \\ Z_{i,j}\ Z_{i+1,j}\ Z_{i,j+1}\ Z_{i+1,j+1} & i+j \text{ is odd} \end{cases}, \tag{4.14}$$

where (i, j) denotes the coordinates and $X_{i,j}$ and $Z_{i,j}$ are the Pauli operators acting on the corresponding qubit. We denote the set of all $g_{i,j}$ operators as $\mathcal{M}_g$.

For the trivial phase, we can pick any set of single qubit measurements. We use $\mathcal{M}_P$ to denote the set of single qubit Pauli-P operators (P could be either X,Y or Z). For the volume law phase, we use the set $\mathcal{C}_4$ consisting of four qubit Clifford unitaries $U_{i,j}$, acting on neighboring qubits located at (i, j), $(i + 1, j)$, $(i, j + 1)$ and $(i + 1, j + 1)$.

We study two types of random circuits. First, we consider measurement-only random circuits comprised of only measurements. More specifically, we start with the product state $|0\rangle^{\otimes N}$ and at each updating step, we measure an operator which is chosen uniformly at random from either $\mathcal{M}_Z$ with probability p_z, $\mathcal{M}_Y$ with probability p_y or $\mathcal{M}_g$ with probability $p_g = 1 - p_z - p_y$. Each time step is defined as N consecutive updating steps. A typical example of such a circuit is shown in Fig. 4.5a.

We also consider hybrid random circuits, which are comprised of unitary gates as well as measurements. We start with $|0\rangle^{\otimes N}$ and at each updating step we either apply a gate chosen uniformly at random from $\mathcal{C}_4$ with probability p_u or measure an operator chosen uniformly at random from $\mathcal{M}_Z$ or $\mathcal{M}_g$ with probabilities p_z and $p_g = 1 - p_u - p_z$ respectively. A typical example of such a circuit is shown in Fig. 4.5c.

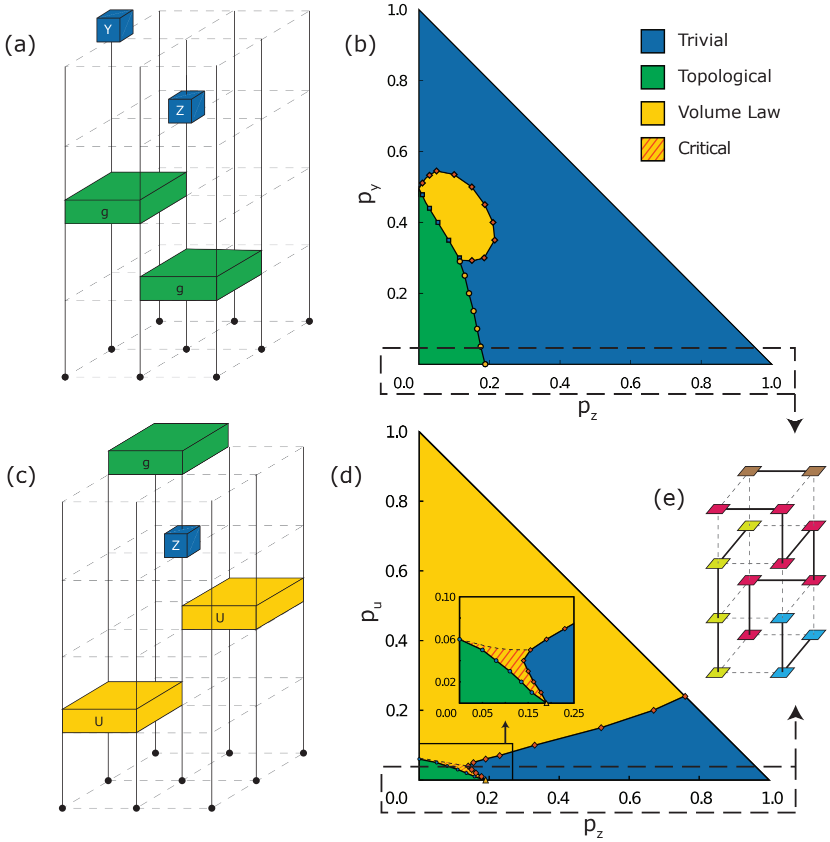


Figure 4.5: (a) A typical measurement-only random circuit. (b) Phase diagram of (2+1)D measurement-only random circuits. (c) A typical hybrid random circuit. (d) Phase diagram of (2+1)D hybrid random circuits. (e) Entanglement dynamics at the $p_y = 0$ line of measurement-only random circuits (as well as the $p_u = 0$ line of hybrid random circuits) maps to a classical bond percolation problem on a cubic lattice.

4.2.2 Order Parameters

One can use TEE [203, 204] denoted by to distinguish phases. In an abuse of notation, we denote TEE by S_{topo}, although it is different from the generalized topological entanglement entropy introduced in the previous section. S_{topo} equals 1 for the eigenstates of the toric code Hamiltonian while it is 0 for quantum states in the trivial phase. As for the volume law phase, the contribution which is proportional to the size of each region cancels out and one may expect S_{topo} to vanish in this phase as well. However, the $(1+1)$D results [169, 188] suggest that the entanglement entropy of a region has sub-extensive contributions [186, 188] in the volume law phase, which results in a system-size dependent value for S_{topo}. Our numerical results support this scenario.

We also use the scrambled ancilla order parameter introduced in section 4.1.2. Since it will be the only ancilla order parameter that we use in this section, we simply call it the ancilla order parameter and drop the "scrambled" part. We utilize $N_a = 10$ ancilla qubits in addition to the system qubits, as follows. First a random local Clifford unitary circuit of depth $O(N)$ is applied to the entire set (system + ancilla) of qubits, which results in a maximally entangled stabilizer state of all qubits. Next, the system qubits are evolved under the random quantum circuit of interest for T time-steps and then the entanglement entropy of the set of all ancilla qubits, denoted by $S_a(T)$, is measured. For large enough system sizes and at $T = \mathcal{O}(L)$, $S_a(T)$ will be 0, N_L and N_a in trivial, topological and volume law phases respectively, where N_L denotes the number of logical qubits in the topological phase ($N_L = 2$ for the torus topology). We assumed that $N_L \ll N_a \ll L^2$.

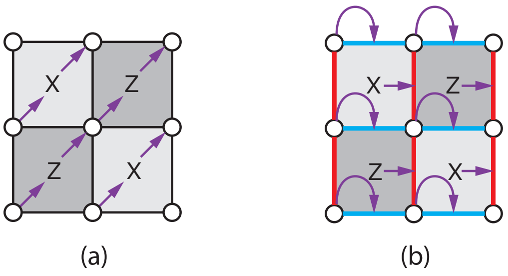


Figure 4.6: The diagrammatic illustration of the action of the duality map (a) U defined by Eqs.(4.15) and (b) $\tilde{U}$ defined by Eqs.(4.16). The white dots correspond to single qubit Y operators and squares correspond to g stabilizers. The vertical red lines and horizontal blue lines in panel b represent XX and ZZ operators respectively

4.2.3 Duality Mappings for the $p_z = 0$ line

Here, we present the two dualities related to the projective random circuit model with $p_z = 0$. First duality maps the family of circuits at p_y to the ones at $1-p_y$, hence fixing any possible phase transition at $p_y = 1/2$. The second duality maps the g and Y operators to the gauge operators of the Bacon-Shor code [205, 206], which might be helpful in understanding the numerical results which will be presented later. For simplicity, we consider the infinite plane geometry.

Consider the Clifford unitary U transforming stabilizer and Y operators in the following way

$$\begin{aligned} U^\dagger\, Y_{i,j}\, U &= g_{i,j} \\ U^\dagger\, g_{i,j}\, U &= Y_{i+1,j+1}, \end{aligned} \tag{4.15}$$

where $g_{i,j}$ is defined in the main text, and $i, j \in \mathbb{Z}$. Diagrammatically, it acts as a half-translation in both i and j directions (see Fig. 4.6a) such that U^2 is just the lattice translation $(i, j) \mapsto (i+1, j+1)$.

It is straightforward to verify that this transformation yields the right commutation relations for the images of g and Y operators. Clearly, the images of Y operators commute among themselves, and similarly for the images of the g operators. On the other hand, note that a stabilizer operator g anti-commutes only with the four single qubit Y operators acting on its corners. The U transformation maps g to a single qubit Y operator and maps the four Y operators to the four neighboring stabilizers, keeping the anti-commutation relations.

To uniquely specify the unitary U, one has to define its action on a complete basis of Pauli strings. The set of Y and g operators is not a complete basis for Pauli strings on an infinite plane and as such, the transformation in (4.15) does not fully specify U. However, since the projective random quantum circuits at $p_z = 0$ are only comprised of g and Y measurements, no matter how one extends Eq.(4.15) to a complete basis, the Clifford U maps a projective random quantum circuit chosen with probability distribution corresponding to $p_z = 0$ and $p_y = p$ to a projective random quantum circuit chosen according to the probability distribution corresponding to $p_z = 0$ and $p_y = 1 - p$.

Moreover, if the stabilizer set describing the state of the system is generated only by operators comprised of g and Y operators, Eq.(4.15) is enough to specify the image of the wave function under U transformations. It also ensures that U keeps the local entanglement structure of the state intact, i.e. changing the entanglement of a region by at most a term proportional to the region's area. These considerations then enforce the $p_z = 0$ line of the phase diagram to be symmetric around $p_y = 0.5$ point.

The second duality maps the Y and g operators to the gauge operators of the Bacon-Shor code. More specifically, consider the Clifford unitary $\tilde{U}$ which transforms the stabilizer and Y

operators as

$$\begin{aligned}
\tilde{U}^\dagger \, Y_{i,j} \, \tilde{U} &= Z_{i,j} Z_{i+1,j} \\
\tilde{U}^\dagger \, g_{i,j} \, \tilde{U} &= X_{i+1,j} X_{i+1,j+1}.
\end{aligned} \tag{4.16}$$

It is illustrated diagrammatically in Fig. 4.6b. As before, we consider the infinite plane geometry. It is easy to verify that $\tilde{U}$ preserves the commutation relations and hence could be extended to a complete unitary. We note that the measurements of g and Y operators, when viewed in the dual picture, resembles the syndrome measurements of the Bacon-Shor subsystem code in the active error correction scheme.

4.2.4 Numerical results

We start by studying the phase diagram of the measurement-only circuits. First, we focus on the $p_y = 0$ line. Notably, as shown in Appendix F, there is an exact mapping which maps the entanglement dynamics at this line of the phase diagram to a classical bond percolation problem on a 3D cubic lattice. Fig. 4.7a and b show the TEE and the ancilla order parameter as a function of p_z. As can be seen from the plots, there exists a stable topological phase extending up to $p_c \approx 0.2$, at which point a continuous phase transition takes the system to the trivial phase.

On general grounds, we may assume the following scaling forms governing the order param-

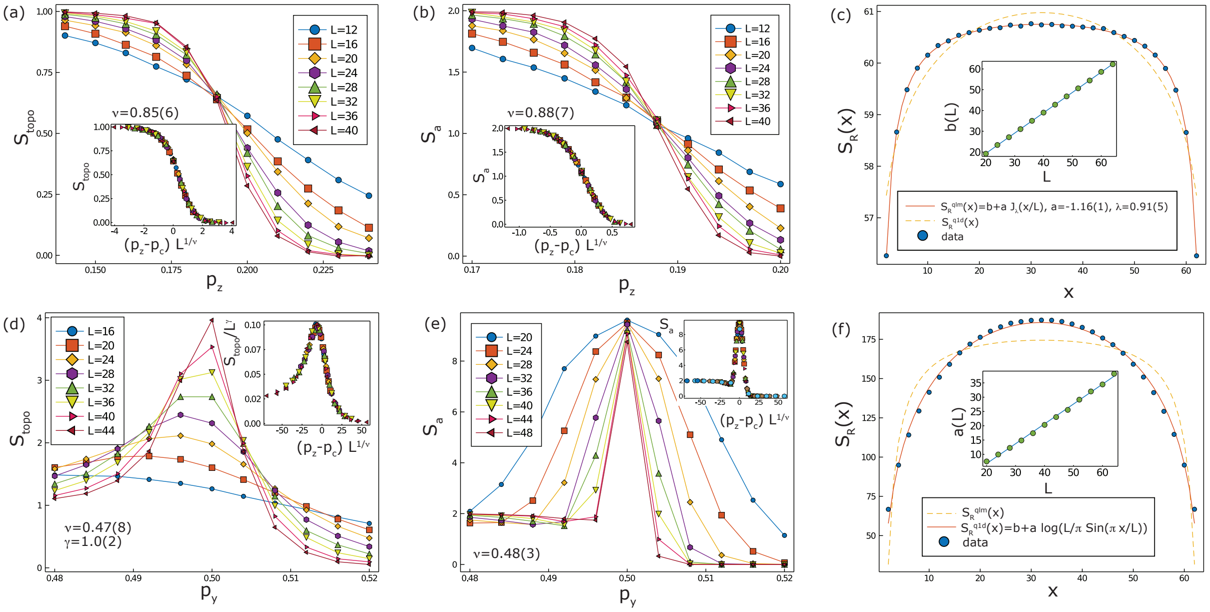


Figure 4.7: Phase transitions across the $p_y = 0$ (top row) and $p_z = 0$ (bottom row) lines of the phase diagram for the measurement-only circuit: (a) S_{topo} and (b) S_a measured at $t = 4L$ versus p_z for fixed $p_y = 0$. Insets show the corresponding data collapse. (c) $S_R(x)$ for system size $L = 64$ at the percolation critical point $(p_z, p_y) = (0.188, 0)$, with the best fit of scaling functions $S^{\text{qlm}}(x)$ (solid line) and S^{q1D} (dashed line). The inset is the best fit value of the b parameter in Eq.(4.20) as a function of L. (d) S_{topo} and (e) S_a measured at $t = 0.6\ L^{1.46}$ versus p_y for fixed $p_z = 0$. Insets show the corresponding data collapse. (f) $S_R(x)$ for system size $L = 64$ at the self-dual critical point $(p_z, p_y) = (0, 0.5)$, with the best fit of scaling functions S^{q1D} (solid line) and $S^{\text{qlm}}(x)$ (dashed line). The inset shows the linear dependence of the best fit value of the a parameter in Eq.(4.19).

eters near the phase transition

$$S_{\text{topo}}(p; L) = L^{\gamma} F((p - p_c) L^{1/\nu}), \tag{4.17}$$

$$S_a(p, t, L) = G((p - p_c) L^{1/\nu}, t/L^z), \tag{4.18}$$

where $F(x)$ and $G(x)$ are arbitrary functions and ν and z are the correlation length critical exponent and dynamical critical exponent respectively. We find our data for the percolation critical point to be consistent with setting γ to 0. By collapsing S_{topo} near the critical point for different system sizes, we find $p_c = 0.188(2)$ and $\nu = 0.85(6)$. Note that ν is consistent with the values obtained from numerical simulation of classical percolation in 3D [207]. By investigating the time dependence of the ancilla order parameter S_a at $p = p_c$, we find it to be consistent with $z = 1$. Collapsing S_a at $t = O(L)$ then yields $\nu = 0.88(7)$, in agreement with the value found via collapsing S_{topo}.

Another quantity of interest is the scaling form of the entanglement entropy with sub-system size at the critical point. We consider the cylindrical region R with a smooth boundary, which has length x in one direction and goes all the way around the torus in the other direction. Let $S_R(x)$ denote its EE. Note that the boundary length $|\partial R|$ is $2L$, independent of x. As is discussed in Appendix F, in the percolation picture this quantity is related to the number of clusters with shared support on region R and its complement.

For a conventional CFT in $(2+1)$D, the non-universal leading area-law term scales with $|\partial R| = 2L$. The sub-leading term for a cylindrical subregion is less well-understood and several forms have been suggested, among which two are of particular interest. One is a quasi-$(1+1)$D scaling function, inspired by the exact form found in $(1+1)$D CFTs, which seems to decently capture entanglement entropy scaling in certain (2+1)D gapless models [208]

$$S_R^{\mathrm{q1d}}(x) = b + a\ \log\left(\frac{L}{\pi}\sin\left(\frac{\pi\ x}{L}\right)\right), \tag{4.19}$$

where b contains the non-universal area law term. The other relevant scaling form was originally derived for the quantum Lifshitz model [209] but was found to describe the entanglement entropy scaling in various other (2+1)D gapless models as well, including some (2+1)D CFTs [210].

$$S_R^{\mathrm{qlm}}(x) = b + aJ_\lambda(x/L) \tag{4.20}$$

$$J_\lambda(u) = \log\left(\frac{\theta_3(i\lambda u)\theta_3(i\lambda(1-u))}{\eta(2iu)\eta(2i(1-u))}\right), \tag{4.21}$$

where $\theta_3(z)$ and $\eta(z)$ are the Jacobi theta function and the Dedekind eta function respectively. b contains the non-universal area-law contribution and λ is a model parameter, which we will use to find the best fit.

Fig. 4.7c shows $S_R(x)$ for system size $L = 64$ at p_c alongside the best fit of the scaling functions. As can be seen from the graph, $S_R^{\mathrm{qlm}}(x)$ results in a good fit (solid line), while S_R^{q1D} cannot capture the scaling form. Moreover, we find that the best fit values of $a = -1.16(1)$ and $\lambda = 0.91(5)$ for S_R^{qlm} remain constant for different system sizes within the margin of error. As is shown in the inset, the b parameter scales linearly with system size, which shows that the leading term scales with $|\partial R|$.

We now turn our attention to the $p_z = 0$ line. Here the circuit has a self-duality mapping $p_y \to 1 - p_y$. Note that along this line, the system has $2L$ subsystem symmetries generated by the product of Y (or stabilizer) operators along horizontal or vertical loops, e.g. $\prod_j Y_{i,j}$. On a related note, as was shown in Section 4.2.3, there is a unitary transformation which maps the g and Y operators to the gauge operators of the $2D$ Bacon-Shor subsytem code [211, 212] on a square lattice. Under this unitary, the subsytem symmetries are maped to the stabilizers of the Baco-Shor code.

By examining the TEE S_{topo}(Fig. 4.7d), we find the topological phase to be extended up to the self-dual point $p_y = 0.5$. However, at $p_y = 0.5$, S_{topo} grows with system size, which suggests a non-zero γ exponent. Collapsing S_{topo} data near the critical point yields $p_c = 0.502(1)$, $\gamma = 1.0(2)$ and $\nu = 0.47(8)$, which shows that this critical point is distinct from the percolation fixed point. Moreover, by looking at the time dependence of the ancilla order parameter at $p_y = 0.5$, we find that, in contrast to the percolation critical point, the best fit to the scaling form in Eq. 4.17 corresponds to $z = 1.46(8)$. Accordingly, by collapsing $S_a(p, t, L)$ data at $t = O(L^{1.46})$ (Fig. 4.7e), we find $\nu = 0.48(3)$, in agreement with the result obtain from collapsing S_{topo}.

As for the cylindrical subregion entanglement entropy $S_R(x)$ (Fig. 4.7f), we find that the quasi-1d scaling form $S^{\text{q1d}}(x)$ – rather than $S^{\text{qlm}}(x)$ – fits the data. However, as is shown in the inset, the a parameter in Eq.(4.20) is not constant, but has a linear dependence on system size L, demonstrating that the leading term scales as $L \log L$ rather than L as is expected in an area law state. The origin of the $L \log L$ violation is unclear; it may be related to the existence of subsystem symmetries along the $p_z = 0$ axis, which translates to the stabilizers of the Bacon-Shor code under the aforementioned duality map.

The rest of the phase diagram can be determined analogously (Fig. 4.5a). We find that the percolation critical point is part of a critical line that persists up to some finite non-zero value of p_y, while the self-dual critical point at $(p_z, p_y) = (0, 0.5)$ splits into two critical lines with an intermediate volume law entangled phase in between, making it tricritical. Interestingly,

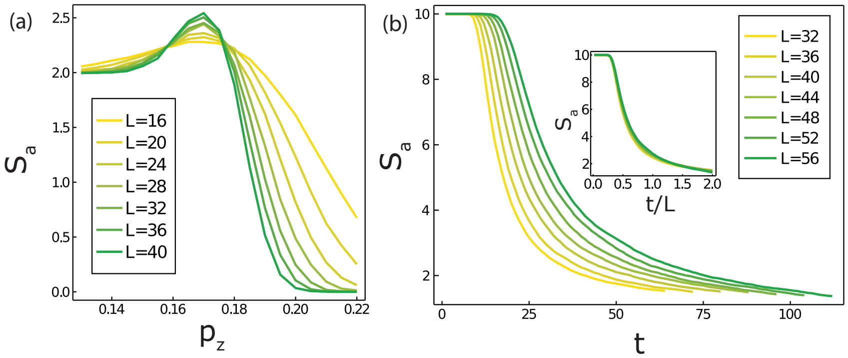


Figure 4.8: (a) The ancilla entanglement entropy S_a measured at $t = L$ as a function of p_z for fixed $p_u = 0.01$ in the hybrid random circuit. (b) S_a as a function of time at $(p_z, p_u) = (0.17, 0.01)$ in the hybrid random circuit. The inset is the same, plotted as a function of t/L.

the numerical data for all critical points that we considered, other than $(p_z, p_y) = (0, 0.5)$, are consistent with $z = 1$ and $\gamma = 0$, with ν remaining close to 0.8, similar to the percolation critical point. Their entanglement entropy scaling is given by $S^{\text{qlm}}(x)$ as well, with an area law scaling leading term. Remarkably, this makes the self-dual point special in this regard, as it is the only point in the phase diagram with $L \log L$ violation of area law, as well as quite different ν and γ exponents. We also note that the extracted a and λ parameters in $S^{\text{qlm}}(x)$ change throughout the phase diagram.

lastly we present the numerical results for the hybrid random circuit which has unitary dynamics. The $p_u = 0$ line of phase diagram is exactly the same as $p_z = 0$ line of the measurement-only random circuit. Fig. 4.8a shows the ancilla order parameter along $p_u = 0.01$, which signals the emergence of an intervening phase between topological and trivial phases, suggesting that the percolation critical point is actually a tricritical point in this phase diagram. In the intermediate phase, S_a does not saturate to $N_a = 10$, as is expected to be the case in the volume law phase, but rather increases weakly with system size, showing indications that it may saturate at a finite value less than 10. Indeed, for a point in the intermediate phase and for large systems, $S_a(t; L)$

seems to be a function of only t/L (Fig. 4.8b) which is a signature of a critical phase with $z = 1$ (see Eq. 4.18). Moreover, we find that in the intermediate phase, $S^{\text{qlm}}(x)$ fits the entanglement entropy of a cylindrical subregion as well. These points suggest that the intermediate phase is a critical region. Nonetheless, we remark that the observed behavior could be just related to finite size effects and the proximity to the critical lines.

The critical region extends to the p_u axis, ending at $p_u \approx 0.06$, which appears to be a tricritical point, although within the precision of this study, we cannot rule out the existence of a narrow critical region around $p_u = 0.06$. By collapsing the S_a data along the p_u axis, we find $p_c = 0.059(1)$ with critical exponent $\nu = 0.78(8)$. On the right, the critical region ends on the boundary of the trivial phase and the volume law phase. The trivial phase itself ends at $p_u = 0.238(2)$ along the $p_u + p_z = 1$ line. We find $\nu = 0.80(7)$ at the corresponding phase transition. The overall phase diagram of the hybrid circuit in 2D is illustrated in Fig. 4.5b.

Chapter 5: Irreducible multi-partite correlations as an order parameter for k-local nontrivial states

Understanding and classifying quantum phases of matter is one of the major goals of theoretical condensed-matter physics. Non-trivial quantum phases are sometimes characterized using the "quantum circuit definition": States belonging to a d dimensional non-trivial phase cannot be obtained from the product state using a finite depth geometrically local (d-dimensional) unitary quantum circuit [213, 214].

However, intriguingly, for non-invertible topologically ordered states the minimum circuit depth required remains infinite even if we replace the condition of geometric locality and demand only the much weaker condition of k-locality [215, 216]. That is, if we only consider unitary gates that act on at most k qubits at a time, without any restriction on their relative placement (as opposed to unitary gates that act on a geometrically local region), we still cannot transform trivial states into topological states in a finite depth. This suggests that the classification of quantum phases of matter can be extended beyond the geometrically-local phases to the k-local setting. In addition to conventional topological states, examples of nontrivial k-local states without any underlying geometrical structure are prevalent in the quantum information context, e.g. quantum LDPC codes [217, 218] and random quantum codes [180, 219].

Given their relevance to both condensed matter and quantum information physics, it is interesting to see how much of the tools and concepts which were developed for the study of topological phases of matter can be generalized to study 'k-local' non-trivial states. In particular, we ask: is it possible to construct an entirely non-geometric order parameter for such k-local non-trivial states? For example, is it possible to distinguish the ground state of the toric code from a typical state in the trivial phase without having access to the underlying geometry, i.e. by removing the position label of all qubits? Or, is it possible to distinguish an encoded state of a quantum code from a typical trivial state without having access to the underlying interaction graph?

In this chapter we show that, at least in a large class of examples, this question can be answered in the positive. We argue that irreducible multi-partite correlations, i.e. correlations that can be inferred only by accessing the global density matrix as opposed to its marginals, amongst random subsystems at a suitably chosen sampling rate can serve as an order parameter for such k-local nontrivial states (precise definitions are provided in the body of the text). We will use this feature to define a non-geometric order parameter for k-local nontrivial states and numerically demonstrate that it can be used to detect a phase transition from a nontrivial phase to a trivial phase.

Using a combination of numerical and analytical methods, we study several examples representing a wide class of k-local non-trivial states including geometric topological phases (toric code), random stabilizers states, quantum LPDC codes (quantum expander codes [220]), and Holographic error correcting codes (the holographic hexagon state [221]).

5.1 Irreducible multipartite correlations

Consider a density matrix ρ shared between a set of M non-overlapping parties $A_1, \cdots, A_M$. The marginal density matrix ρ_j is defined as the reduced density matrix obtained by tracing out A_j, i.e. $\rho_j = \mathrm{Tr}_{A_j}\, \rho$. The parties $A_1, \cdots, A_M$ have non-zero irreducible M-partite correlations if there is more information in ρ than in the set of its marginals $\{\rho_j\}_{j=1}^{M}$.

Let $\tilde{\rho}$ denote the maximum entropy state consistent with all marginals of ρ, meaning $\mathrm{tr}_{A_j}\, \tilde{\rho} = \rho_j$ for all j. As discussed in Refs. [222–225], a quantitative measure of irreducible M-partite correlations can then be defined as:

$$D_M(\rho, \mathcal{A}) = S(\tilde{\rho}) - S(\rho), \tag{5.1}$$

where the function $S(\rho) = -\mathrm{Tr}(\rho \log_2 \rho)$ is the von-Neumann entanglement entropy and $\mathcal{A} = \{A_1, \cdots, A_M\}$ denotes the partitioning. Intuitively, this object quantifies how many bits of information in ρ cannot be read-off unless *all* M parties are accessed. Perhaps, the simplest example of a state with $D_M > 0$ is provided by the GHZ state [222, 226].

5.1.1 Irreducible multipartite correlations in stabilizer states

Computing D_M is not easy in general. However, for stabilizer states it can be efficiently computed. Let A denote a set of n_A qubits in a mixed stabilizer state specified by the stabilizer density matrix ρ_A given as

$$\rho_A = \frac{1}{2^{n_A}} \sum_{g \in G_A} g. \tag{5.2}$$

where G_A is a the stabilizer group corresponding to ρ_A. The entanglement entropy of ρ can be related to its stabilizer group as follows [200] (see also Appendix D),

$$S(\rho) = n_A - \dim G_A. \tag{5.3}$$

Let $\mathcal{A}$ denote a partitioning of A into a set of M non-overlapping parties (subsystems) $A = \cup_{j=1}^{M} A_j$. D_M can then be equivalently written as [224, 226],

$$D_M(\rho, \mathcal{A}) = \text{dim } G_A - \text{dim } G_{\mathcal{A};(M-1)}, \tag{5.4}$$

where $G_{\mathcal{A};(M-1)}$ is defined as the stabilizer group generated by stabilizers in G_A which are shared by at most $(M-1)$ parties in $\mathcal{A}$. The generator set for $G_{\mathcal{A};(M-1)}$ can be written as $\cup_{j=1}^{M} G_{A \setminus A_j}$.

In the case where the system studied is pure $S(\rho_A) = 0$, it was shown in Ref. [226] that $D_M(\rho, \mathcal{A})$ is equal to the M partite GHZ extraction yield of the stabilizer state with respect to $\mathcal{A}$, that is, the maximum number of M partite GHZ states that can be extracted by local unitaries (local to each party). We further remark that if the M-partite GHZ extraction yield of stabilizer states is non-zero for a macroscopic M, i.e. for M growing with system size, it immediately follows that the state cannot be prepared with a constant depth k-local unitary circuit and is therefore k-local nontrivial.

To develop intuition, let us consider the multipartite correlations shared between different regions (subsystems) of the ground state of the Kitaev's toric code on a torus (or any other $2D$ topological phase), with logical stabilizers going around both the non-contractible loops of the torus. As a simple example, divide the torus into a set of M equal size "cake-slices" correspond-

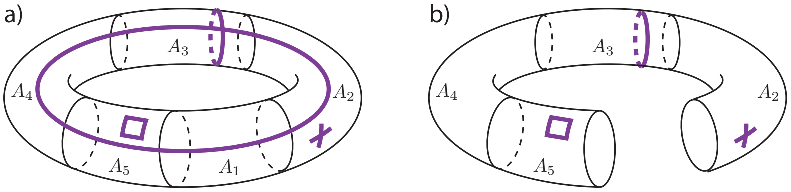


Figure 5.1: a) The toric code state on a torus. Thick lines represent some of its stabilizers. The system is partitioned into five disjoint parties $A = \cup_{j=1}^{5} A_j$. b) The marginal $A \setminus A_1$. All stabilizers except the one that goes around the large handle of the torus can be read from the marginals, resulting in $D_5 = 1$.

ing to $A_1, \cdots, A_M$ (see Fig. 5.1). Apart from the logical qubit that goes around the torus, all of the information is encoded locally and therefore present in $M-1$ party density matrices (all the other stabilizer generators or Wilson loops can be taken to have support only in one region or in the boundary of two). However, the logical operator that goes around the cake cannot be accessed without having all M of cake slices (cannot be contracted to any $M-1$ of the subsystems). We, therefore, have $D_M = 1$.

5.1.2 Upper bound from conditional quantum mutual information

We can use the strong subadditivity of von-Neumann entropy to upper bound D_M. Consider dividing the M parties in A into any three disjoint sets B, C and D such that BC, CD and BD are all subsets of marginals of A - e.g. $B = \cup_{i=1}^{M-2} A_i$, $C = A_{M-1}$ and $D = A_M$. Let $\tilde{\rho}$ be the maximum entropy state consistent with all marginals of ρ. Strong subadditivity for $\tilde{\rho}$ entails,

$$S(\tilde{\rho}) \leq S(\mathrm{tr}_C(\tilde{\rho})) + S(\mathrm{tr}_D(\tilde{\rho})) - S(\mathrm{tr}_{CD}(\tilde{\rho})). \tag{5.5}$$

Since $\tilde{\rho}$ has the same marginals as ρ, we can compute the entropies on the right hand side on ρ instead of $\tilde{\rho}$. Therefore, we have the following upper bound on D_M

$$\begin{aligned} D_M(\rho, \mathcal{A}) &\leq S(BD) + S(BC) - S(B) - S(BCD) \\ &= I(C : D|B), \end{aligned} \tag{5.6}$$

where $I(C : D|B)$ is the conditional quantum mutual information, and in an abuse of notation, we have used $S(X)$ to denote $S(\operatorname{tr}_{A \setminus X} \rho)$ for a subset X of the qubits. Note that all entropies on the right hand side correspond to ρ rather than $\tilde{\rho}$ and hence are easily computable.

5.2 Irreducible multipartite correlations between random subsystems in different quantum phases of matter

We now define our proposed quantity of interest. We start with the explicit definition and then explain the intuition behind it.

Consider a quantum state of N qubits. For a given M and q such that $0 < Mq \leq 1$, choose the subset A and its M-partitioning as follows; Label every qubit with a random number x_i chosen uniformly from $[0, 1]$. Define A_j as the set of qubits with labels between $(j-1)q$ and $j\,q$ and let A denotes union of A_1 through A_M, i.e.

$$A_j = \{i : \; (j-1)\,q < x_i < j\,q\} \quad \text{for } j = 1, \cdots, M \;, \tag{5.7}$$

$$A = \cup_{j=1}^{M} A_j, , \quad \mathcal{A} = \{A_1, \cdots, A_M\}. \tag{5.8}$$

Let ρ denote the reduced density matrix of the system on A. We define $C(N, M, q)$ as the average of $D_M(\rho, \mathcal{A})$ over different random labelings of the qubits,

$$C(N, M, q) = \mathbb{E}[D_M(\rho, \mathcal{A})]. \tag{5.9}$$

Roughly speaking $C(N, M, q)$ measures irreducible multipartite correlations between M randomly chosen subsystems, each of size $\sim qN$. In this chapter, we study the behavior of $C(N, M, q)$ as one changes q and M (the number of parties), and discuss how it can be used as an order parameter for k-local nontrivial states.

The basic idea is that the presence of generic irreducible multipartite entanglement for a macroscopic number of parties in a quantum state suggests that the state is non-trivial. To gain a better understanding of this fact, it is helpful to see what can be said about $C(N, M, q)$ on general grounds, before considering specific examples.

Consider evaluating $C(N, M, q)$ for a state which belongs to a particular k-local nontrivial state. The set A in Eq. (5.8) can be thought of as the random subset which includes each qubit with probability $p = Mq$. In the small sampling rate limit $p \simeq 0$, we expect A to be a set of sparse uncorrelated qubits and hence ρ (the reduced density matrix on A), to be indistinguishable from a typical state in the trivial phase, which in turn implies we can not distinguish the underlying phase of matter from the trivial phase. However, for large sampling rates $p \simeq 1$, we expect the global state to be approximately recoverable from ρ as we have access to global information, and hence, we can determine the underlying phase of matter. This observation suggests a phase transition in the complexity of the density matrix of a random subsystem [227] as a function of sampling rate p. The quantity $C(N, M, q)$ is designed to capture this phase transition.

To see this, consider fixing $M \gg 1$ and changing q from 0 to $1/M$. For $p = Mq$ near zero, as mentioned before, the density matrix ρ is approximately a tensor product of independent density matrices which implies $C(N, M, q) \approx 0$. As we increase q, ρ starts to see some correlations between few A_js, but since D_M catches only *irreducible* M-partite correlations and since $M \gg 1$, we still get $C(N, M, q) \approx 0$. However, as soon as $p = Mq$ becomes large enough that the global entanglement structure of the state can be read off from ρ, C becomes non-zero. On the other hand, as we increase q further, the global information would now become accessible by the marginals of ρ as well and therefore C drops to zero again. This discussion suggests two phase transitions happen as one tunes q from 0 to $1/M$; first, when ρ starts to see the global entanglement at $Mq = p_c$, and the second, when this information becomes accessible to the marginals of ρ as well, i.e. at $(M-1)q = p_c$.

We will back up the scenario described in the previous paragraph by (1) Computing C in various specific examples in the next section, (2) Providing a geometrical picture in terms of bond percolation in the case of geometric topological phases and (3) Providing a more quantitative argument in the case of error correcting codes with a finite erasure threshold.

5.3 Examples

5.3.1 Random Haar and random stabilizer states

In this subsection, we study the behavior of $C(N, M, q)$ for random Haar and random stabilizer states. The entanglement structure of subsystems of random Haar and random stabilizer states is well studied. In particular, if $|\psi\rangle$ is a random state of N qubits and if ρ denotes its reduced density matrix on a subset A of qubits with size $n_A < N/2$, it is known that the average entanglement entropy of ρ is exponentially close to its maximum possible value, i.e. n_A [228, 229]:

$$\mathbb{E}[S(\rho)] = n_A - O(2^{-2(N/2-n_A)}). \tag{5.10}$$

As we shall explain below, it follows then that $C(N, M, q) = 0$ for $Mq < 1/2$ as well as for $(M-2)q > 1/2$, up to exponentially small corrections in N.

First consider the $Mq < 1/2$ case. Let A be the random subset of qubits as defined in Eq. (5.8) and let ρ denote the corresponding reduced density matrix. Note that $n_A/N \simeq Mq$ and that the equality becomes exact in the $N \to \infty$ limit. This shows $\mathbb{E}[S(\rho)] = n_A - O(2^{-N(1/2-Mq)})$. Since for any density matrix $\tilde{\rho}$ on n_A qubits we have $S(\tilde{\rho}) \leq n_A$, it follows from Eq. (5.4) that $C(N, M, q) = O(2^{-N(1/2-Mq)})$.

For the $(M-2)q > 1/2$ case, let $B = \cup_{j=1}^{M-2} A_i, C = A_{M-1}, D = A_M$ denote three non-overlapping subsets of A such that $A = B \cup C \cup D$. Note that $|B|/N \simeq (M-2)q > 1/2$ and that the equality becomes exact in the $N \to \infty$ limit. Roughly speaking, in this case B, C, D form an approximate quantum Markov chain, and therefore one could recover the whole state of A just from its marginals [230, 231]. To get a more precise formulation, note that B, C, D satisfy the condition of the strong subadditivity bound in Eq. (5.6). The value of the upper bound can be found using Eq. (5.10),

$$\begin{aligned} I(C : D|B) &= S(BC) + S(BD) - S(B) - S(BCD) \\ &= O(2^{-\varepsilon N}). \end{aligned} \tag{5.11}$$

Hence we get $C(N, M, q) = O(2^{-\varepsilon N})$.

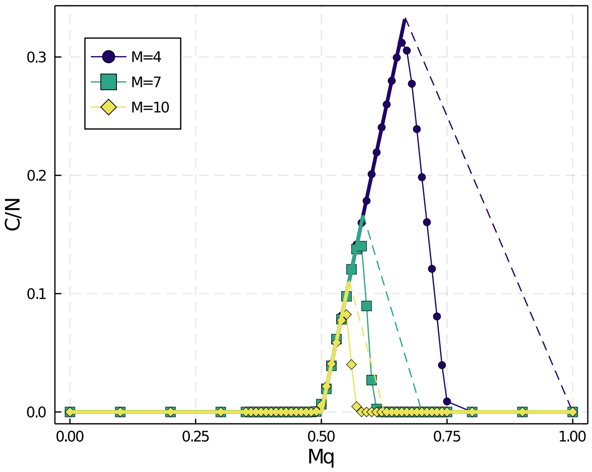


Figure 5.2: $C(N, M, q)/N$ evaluated numerically for a random stabilizer state, on a fixed system size $N = 5184$ and for different M's as a function of $p = Mq$. The thick solid lines correspond to the analytic result. The dashed lines correspond to the analytic upper bound in Eq. (5.13)

We now turn to the regime where $Mq > 1/2$ but $(M-1)q < 1/2$. First we consider the random stabilizer states. Since $(M-1)q < 1/2$ it follows that for large enough N, $|A \setminus A_j|/N < 1/2$ for any $j = 1, \cdots, M$. Therefore, comparing Eq. (5.10) and Eq. (5.3) shows $\mathbb{E}[\dim G_{A\setminus A_j}]$ is exponentially small, which in turn means the contribution of $\mathbb{E}[\dim G_{A;(M-1)}]$ in Eq. (5.4) is exponentially small. On the other hand, $Mq > 1/2$ implies that $|A^c|/N < 1/2$, where A^c stands for the complement of A. Therefore we can find $\dim G_A$ from Eq. (5.3) by noting $S(A) = S(A^c)$ and using Eq. (5.10) to find $S(A^c)$. Hence, we find $\mathbb{E}[\dim G_A]/N \simeq 2Mq - 1$, where the equality becomes exact in the thermodynamic limit. This shows $C(N, M, q)/N = 2Mq - 1$ up to small errors vanishing in the thermodynamic limit. We expect the same behavior in random Haar states because in the intermediate regime, all $M-1$ marginal density matrices are exponentially close to the maximally mixed state and therefore the maximum entropy state consistent with all of them is expected to have the maximum entropy allowed by that system size, i.e. $S(\tilde{\rho}) \simeq NMq$, which implies $C(N, M, q)/N \simeq 2Mq - 1$, similar to the random stabilizer states.

In the final regime where $(M-1)q > 1/2$ but $(M-2)q < 1/2$, we can only upper bound $C(N, M, q)$. Note that the subadditivity bound Eq. (5.6) with the choice of regions $B = \cup_{j=1}^{M-2} A_i, C = A_{M-1}, D = A_M$ gives (following the same logic as the $(M-2)q > 1/2$ regime),

$$\begin{aligned} I(C : D|B) &= S(BC) + S(BD) - S(B) - S(BCD) \\ &= N - 2n_B + O(2^{-\varepsilon N}). \end{aligned} \tag{5.12}$$

This implies that, in the regime $(M-1)q > 1/2$ but $(M-2)q < 1/2$, we have $C(N, M, q)/N < 1 - 2(M-2)q$.

We remark that the combination of this bound with the results in the regime $Mq > 1/2$, and, $(M-1)q < 1/2$, already implies that $C(N, M, q)/N$ is non-analytic at $(M-1)q = p_c = 1/2$ (this is the second transition discussed in Section 5.2). Numerically, we find that the bound above is not tight in this regime.

Putting everything together, for the random stabilizer state we have the following result:

$$\begin{cases} C/N = 0, & Mq < \frac{1}{2} \\ C/N = 2Mq - 1, & (M-1)q < \frac{1}{2} < Mq \\ C/N < 1 - 2(M-2)q, & (M-2)q < \frac{1}{2} < (M-1)q \\ C/N = 0, & 1/2 < (M-2)q \end{cases} \tag{5.13}$$

up to small corrections vanishing in large N.

Our numerical, as well as analytical results, are displayed in Fig. 5.3.1. The thick solid lines correspond to the analytic expressions and the dashed lines represent the analytical upper bound. The numerical result are computed for a system size $N = 5184$. The slight discrepancy between the numerical and analytical results is due to finite size effects and will vanish in the thermodynamic limit [232]. Consistent with the discussion of Section 5.2, we find (at least) two phase transitions where the behavior of $C(N, M, q)$ is non-analytic - One where the set of all M parties start to have access to global information about the state $Mq = p_c = 1/2$ and a second one where the set of $M - 1$ parties starts to have access to global information $(M - 1)q = p_c = 1/2$.

In analogy with the usual theory of critical phenomena, close to the phase transition at $Mq = 1/2$, we find the scaling ansatz

$$\frac{C(N, M, q = p/M)}{N} = (M-1)^{-\beta} f((p - p_c)(M-1)^{\alpha}), \tag{5.14}$$

will collapse the data perfectly with $p_c = 1/2$, $\alpha = 1$, and $\beta = 1$. In the following subsections, we successfully apply the same scaling ansatz to study several analogous phase transitions.

5.3.2 Topological Phases of Matter

We now consider the behavior of $C(N, M, q)$ in the conventional topological phases of matter. We will focus on the toric code model, but we expect a similar result to hold for all topological phases.

Consider the toric code model on a $L \times L$ torus. In Section 5.1.1 we discussed the behavior of D_M for a particular choice of subsystems in the toric code. Here we discuss the behavior of D_M for generic subsystems. Let $\mathcal{A} = \{A_i\}_{i=1}^{M}$ denote a set of disjoint subsystems of the qubits on the torus. Their union $A = \cup_{i=1}^{M} A_i$, can be viewed as a collection of patches with some holes on the torus. Let ρ denote the reduced density matrix of the system on A. Stabilizers that stabilize ρ

can be taken to be either (1) normal 4-qubit toric code stabilizers, or (2) stabilizer loops around the holes in A, or (3) logical loops around the torus. If we take $M > 4$, $D_M(\rho, \mathcal{A})$ can only have contributions from the stabilizers of type (2) and (3). Moreover, such stabilizers contribute only if they are not stabilizing any of its marginals. Hence, we see that $D_M(\rho, \mathcal{A})$ counts the number of independent non-trivial loops in A that are not present in any of its $(M-1)$-party subsystems. In the context of more general CSS quantum codes, this can be put more formally as a question about how much of the homology/cohomology properties of A, thought of as cellulation of a manifold or more generally a chain complex can be read-off from its subsystems.

When the A_i subsystems are chosen randomly as described in Section 5.2, one can gain further insights into the behavior of D_M from the theory of bond percolation. We can view A as the set of qubits on the torus each chosen with probability $p = Mq$. Similarly, any $(M-1)$-party union set, i.e. $A \setminus A_j$ for any j, can be viewed as the set of qubits on the torus each chosen with probability $p' = (M-1)q$. Now if $p > p_c$ while $p' < p_c$ (with $p_c = 1/2$ denoting the bond percolation threshold on the square lattice), the set A percolates with probability one while no $(M-1)$-party union set does. Hence, while the two logical qubits are accessible to A, their value can not be deduced from any of the $(M-1)$-party marginals. This ensures that $C(N, M, q) \geq 2$ for $\frac{p_c}{M} < q < \frac{p_c}{(M-1)}$. Note that this region of q shrinks as M increases, which implies that for large Ms we are close to the percolation critical point and should therefore be careful with the order of limits and finite-size effects. Proximity to the percolation threshold $Mq \approx p_c$, in turn, implies that A is likely to include macroscopically large holes. Each one of these large punctures (with perimeter larger than M) has some chance of contributing to $D_M(\rho, \mathcal{A})$ depending on whether it is present in $M-1$ partite systems or not (note that these contributions are also present on surfaces without a logical qubit, e.g. a sphere).

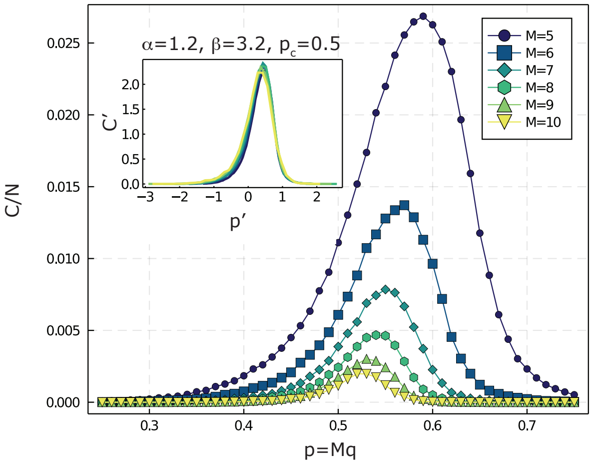


Figure 5.3: $C(N, M, q)/N$ for the toric code on a torus. Plotted at a fixed system size $N = 72\times72$ and for different Ms as a function of $p = Mq$. Inset: the best data collapse for the toric code around the point $Mq = 1/2$, using the scaling ansatz of Eq. (5.14) with parameters $p_c = 0.5$, $\alpha = 1.2$ and $\beta = 3.2$.

Putting everything together, we expect $C(N, M, q)$ to be small unless we choose our parameters such that A and its $(M - 1)$-party subsystems are on different percolation phases $\frac{p_c}{M} < q < \frac{p_c}{(M-1)}$. Furthermore, the calculation of $C(N, M, q)$ can be mapped into a calculation of the distribution of large punctures in the percolation theory near the critical point. However, we do not attempt this analytical computation here and instead use numerical results to find a phenomenological form for $C(N, M, q)$ and discuss its universal properties which should be entirely set by the percolation CFT. We expect a similar picture to hold in all topological phases (with stabilizers replaced by generic Wilson loop operators).

Before proceeding further, we'd like to emphasize that the discussion above was simplified by the fact that both X and Z stabilizers of the toric code live on a square lattice (square lattice is self dual). In the case of more generic lattices, or more generic CSS codes, one needs to carefully separate the discussion of X and Z stabilizers to be precise.

We now proceed to present the numerical results for the case of the toric code on a torus (square lattice): Based on numerics, we find that in the large system size limit $N \gg 1$, $C(N, M, q)$ becomes proportional to the system size N. This is consistent with the percolation picture because the distribution of punctures is proportional to the area of the system. In Fig. 5.3, we have plotted $C(N, M, q)/N$ at fixed system size $N = 72 \times 72$ as a function of $p = Mq$ for a number of different Ms. Consistent with our expectation, we find that $C(N, M, q)$ is nonzero in a region of width $\Delta p \approx 1/M$ around $p_c = 1/2$. Furthermore, the inset shows that we can collapse the data around the $Mq = 1/2$ point, using the scaling ansatz of Eq. (5.14). We find that $p_c = 0.5$, $\alpha = 1.2$ and $\beta = 3.2$ results in the best collapse.

5.3.2.1 Using $C(N, M, q)$ to detect a phase transition

To claim that the behavior of $C(N, M, q)$ can be used to distinguish trivial and non-trivial states we need to understand how $C(N, M, q)$ behaves across a phase transition. As an example, we look into the phase transition from the toric code fixed point into the trivial phase. In the Hamiltonian setting, this can be done by following the ground state of the troic code in an external magnetic field [194]. This approach is not suitable for us since our computational power in computing is $C(N, M, q)$ is limited to stabilizer states. As we saw in Chapter 4, in the context of monitored random quantum circuits we can stabilize topological phases and study their transitions entirely within the stabilizer formalism. This convenient feature makes these models suitable to study using our computational method.

In particular, we shall use a simplified monitored random circuit model which is equivalent to a special case of the measurement-only circuit introduced in Section 4.2. We consider a radom circuit on a 2D square lattice of qubits (on a torus) where at each step, with probability p_y a

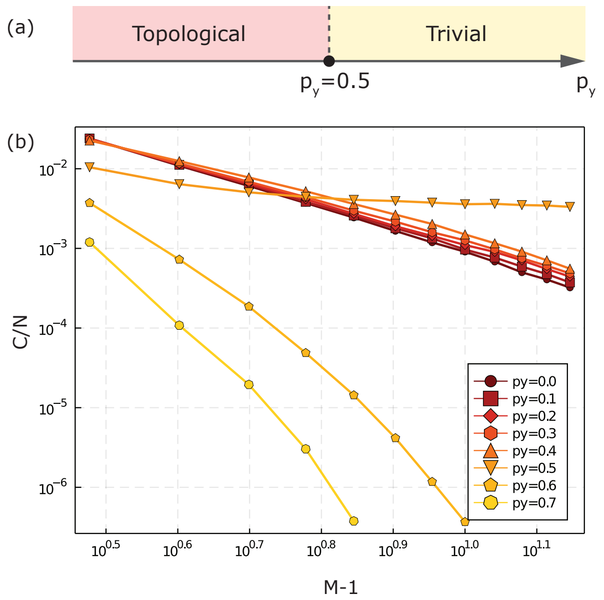


Figure 5.4: a) The phase diagram of the random quantum circuit, consisted of measuring either single qubit Y with probability p_y or four qubit toric code stabilizers with probability $1-p_y$. b) $C(N, M, q)$ at $Mq = p_c = 0.5$ and $N = 56 \times 56$ as a function of M evaluated on the late time states of the random quantum circuit and average over many realizations.

randomly chosen single qubit is measured in the Y basis, and with probability $1-p_y$ a random stabilizer of the toric code is measured. The entanglement structure of the late time state of this circuit exhibits an entanglement phase transition at $p_y = 1/2$, as shown in Fig. 5.4a. We refer the reader to Section 4.2 for further details about this phase transition.

We study the behavior of the $C(N, M, q)$ averaged over different circuit realizations as we change p_y across the phase transition. In particular, for each p_y, we look at $C(N, M, q)$ at $Mq = 0.5$ as a function M. Based on the ansatz in Eq. (5.14), we expect $C(N, M, q = p_c/M)$ to be a power law as a function of $M - 1$ in the topological phase. On the other hand, in the trivial phase, we expect C to vanish faster than any power-law due to the absence of long-range entanglement. The numerical results are presented in Fig. 5.4b. As evident from the plot, $C(N, M, q = p_c/M)/N$ behaves as a power-law throughout the topological phase. The expo-

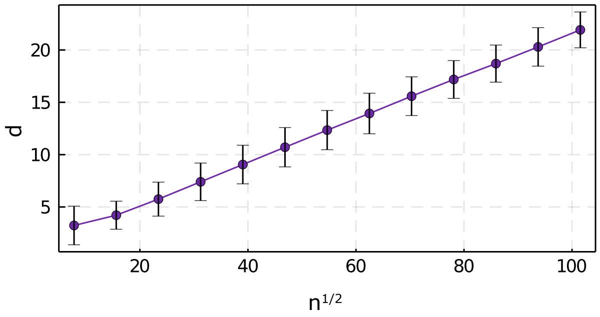


Figure 5.5: Average distance and its variance of quantum expander code with n qubits as a function of $\sqrt{n}$, based on random bi-partite graphs with left and right degrees 5 and 6 respectively.

nent also seems to remain constant (within the error bar), suggesting that the power-law shape of $C(N, M, q = \frac{1/2}{M})$ as well its exponent are universal in the topological phase. In the trivial phase $p_y > 0.5$, we find that $C(N, M, q = \frac{1/2}{M})$ decays faster than any power law (shown in Fig. 5.4b) - establishing a sharp, qualitative difference between the behavior $C(N, M, q = p_c/M)$ in trivial and topological phases.

5.3.3 Quantum Expander Code

Code states of a $[\![n, k, d]\!]$ quantum error correcting code with $k > 0$ and d growing with system size can not be prepared via constant-depth k-local unitary circuits [215, 216]. Therefore, it is interesting to study the irreducible multipartite entanglement in the quantum states which correspond to code words of a quantum error correcting code.

In particular, we will focus on the quantum expander code family [220]. Quantum expander codes have finite rate and their distance scales as $\sqrt{n}$, so they make a family of $[\![n, k = \Theta(n), d = \Omega(\sqrt{n})]\!]$ quantum codes. A brief review of the quanutm expander code construction is proided in Appendix G. In what follows, we consider quantum expander codes on n qubits which are based on classical random bi-partite graphs with left and right degrees $\Delta_L = 5$ and $\Delta_R = 6$ respectively.

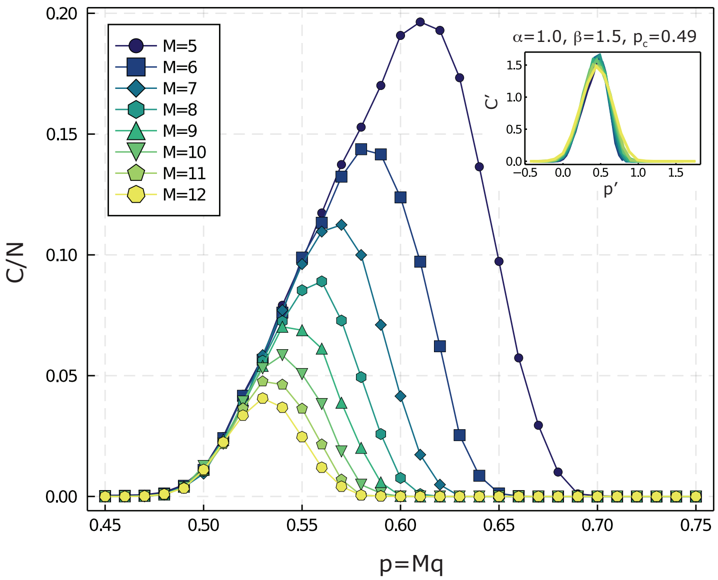


Figure 5.6: $C(n, M, q)$ as a function of $p = Mq$ for different Ms at fixed $n = 2196$. The inset shows the collapsed data corresponding to the first phase transition, where the horizontal axis is rescaled to $p' = (Mq - p_c)(M - 1)^{\alpha}$ and the vertical axis is rescaled to $C' = C\,(M - 1)^{\beta}$, with $p_c = 0.49$, $\alpha = 1.0$ and $\beta = 1.5$.

It can be shown [233] that with probability $P > 1 - O(n^{-\beta})$, the resulting quantum code has distance $d > \alpha\sqrt{n}$ for some positive constants α and β and encodes $k = \frac{(\Delta_R - \Delta_L)^2}{\Delta_R^2 + \Delta_L^2} n = \frac{1}{61} n$ logical qubits. This result in an upper bound on the erasure threshold of $e_{\text{th}} \leq (1 - 1/61)/2 \simeq 0.49$ [234]. Fig. 5.5 shows the average code distance of this family and its variance as a function of $\sqrt{n}$.

We now present the numerical results corresponding to the irreducible many-partite correlations for code states of the aforementioned quantum expander codes. All results are averaged over different selections of random partition as well as different realizations of random quantum expander codes. Fig. 5.6 shows $C(n, M, q)$ as a function of $p = Mq$ for different values of M at fixed system size of $n = 2196$. As expected from general considerations, C vanishes for small and large values of p and becomes nonzero in a window of width $\Delta p \sim \frac{1}{M}$ near $p \sim 1/2$. The

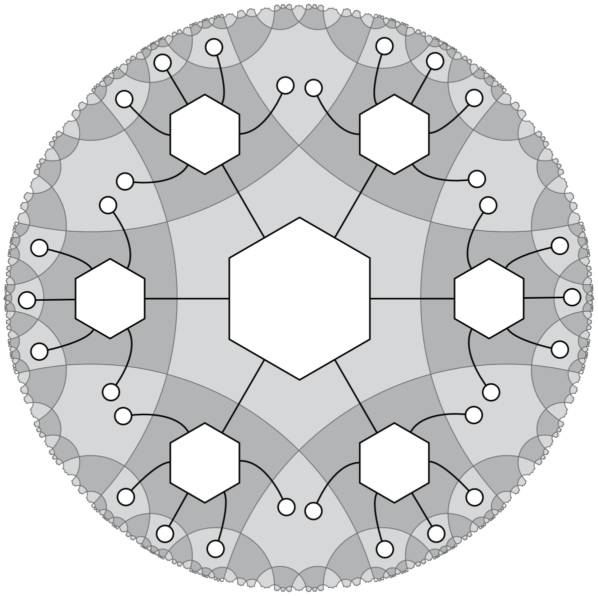


Figure 5.7: The holographic state on $n = 30$ qubits constructed by two layers of 6-legged perfect tensors. The white dots denote the physics qubits.

inset shows the collapsed data according to the scaling ansatz in Eq. (5.14) We find $p_c = 0.49$, $\alpha = 1.0$ and $\beta = 1.5$ collapses different curves on top of each other.

5.3.4 Holographic states

The holographic quantum error correcting codes and the closely related holographic states were introduced as simple toy models of the AdS/CFT correspondence [221, 235]. These constructions utilize a special type of tensor, known as a perfect tensor. The tensor network which results from contracting a finite number of perfect tensors according to a compatible tessellation of the hyperbolic plane represents a holographic quantum error correcting code or a holographic state depending on whether the tensors in the bulk have any uncontracted leg left or not. The dangling legs on the boundary correspond to physical qubits while the uncontracted bulk legs (if any) correspond to logical qubits. Here, we focus on a holographic state which is defined by

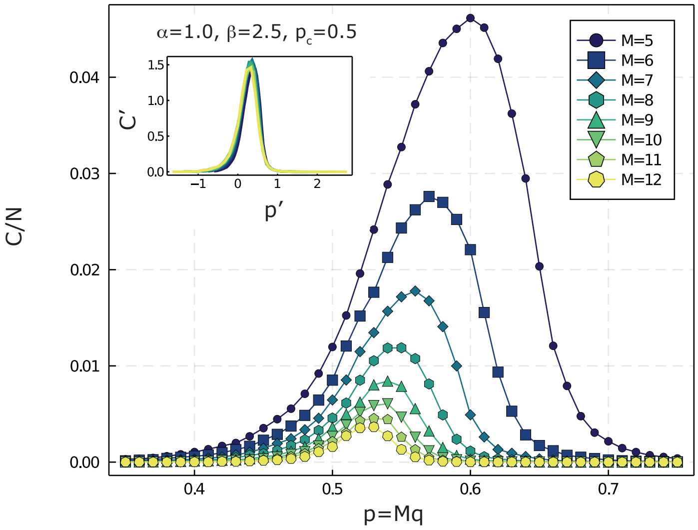


Figure 5.8: $C(N, M, q)$ as a function of Mq for different Ms, computed for the the holographic hexagon state with $N = 1590$ physical qubits. The inset shows the data collapse according to the scaling ansatz of Eq. (5.14), with $p_c = 0.5$, $\alpha = 1.0$ and $\beta = 2.5$. The horizontal axis of inset is $p' = (Mq - p_c)(M-1)^\alpha$ and the vertical axis is $C' = (M-1)^\beta C$.

contracting 6-legged perfect tensors according to the $\{6, 4\}$-tiling of the hyperbolic plane (see Fig. 5.7). The perfect tensor in this construction corresponds to a $[\![6, 0]\!]$ stabilizer code which is closely related to the well-known 5-qubit code. This makes the resulting holographic state, a stabilizer state as well (see Ref. [221] for more details). In what follows we refer to this state as the holographic hexagon state.

Fig. 5.8 shows $C(n, M, q = p/M)$ as a function of $p = Mq$, for fixed $n = 5934$ and different Ms. This particular number of n corresponds to contracting $r = 6$ layers of edge-adjacent hexagons on the hyperbolic plane (see Fig. 5.7 for $r = 2$). Similar to other cases, C vanishes for small and large ps and it peaks near $p = 0.5$. The inset of the plot, shows the same data but collapsed according to the scaling ansatz of Eq. (5.14), with parameters $p_c = 0.5$, $\alpha = 1.0$ and $\beta = 2.5$.

5.4 Relation to erasure threshold

In this subsection, we discuss the relation between $C(N, M, q)$ and the erasure threshold in quantum error correcting codes. Let ρ be a given code state in a quantum error correcting code $\mathcal{Q}$. Consider the situation in which the subset E of the qubits are erased. The quantum error correcting code $\mathcal{Q}$ can correct for this erasure error if and only if it can reconstruct ρ using only the remaining qubits. We say an error correcting code has a finite erasure threshold $e_{\text{th}} > 0$ if with probability one it can correct random erasure errors below a certain rate $e < e_{\text{th}}$, where e is the probability of a given qubit being erased. Equivalently, in a quantum error correcting code with erasure threshold e_{th}, if we consider a subset A of qubits where each qubit is kept at random with probability $p > (1 - e_{\text{th}})$, it includes all the information about the state of the logical qubits with probability one. By no-cloning theorem, this implies that the erased subsystem includes no information about the state of the logical qubits.

Consider the stabilizer state $\rho = |0_L\rangle\langle 0_L|$ in a stabilizer code $\mathcal{Q}$ with a finite erasure threshold e_{th}. Let Z_L denote the logical operator associated with $|0_L\rangle$, which along with other stabilizers, generates the stabilizer group of ρ. Consider the closely related stabilizer state $\tilde{\rho} = \frac{1}{2}(|0_L\rangle\langle 0_L| + |1_L\rangle\langle 1_L|)$. The generators for the stabilizer group of $\tilde{\rho}$ can be taken to be the same as the generators of the stabilizer group for ρ but with Z_L dropped. For a given M and q, consider A to be the randomly chosen subsystem defined in Section 5.2 and let $\rho_A = \mathrm{tr}_{A^c}(\rho)$ and $\tilde{\rho}_A = \mathrm{tr}_{A^c}(\tilde{\rho})$ denote the associated reduced density matrices on A. From the discussion in the previous paragraph, we know that if $Mq < e_{th}$, with probability one ρ_A does not include Z_L or a logically equivalent operator (for brevity, in the rest of this paragraph we drop the phrase "or a logically equivalent operator"). We also know that if $Mq > 1 - e_{th}$, with probability one ρ_A does include Z_L. Now consider the smallest $Mq = p_c$ such that ρ_A has a nonzero probability to include Z_L. Note that at this point, marginals of ρ_A do not include Z_L, therefore, all marginals of ρ_A and $\tilde{\rho}_A$ would be equal. In that case, the stabilizer group associated with ρ_A has a finite chance to have one extra generator which is not present in the stabilizer group of $\tilde{\rho}_A$, which in turn entails that there is a finite chance to get $S(\tilde{\rho}) > S(\rho_A)$ (see Eq. (5.3)). Therefore we are guaranteed to get $C(N, M, q) > 0$ for the interval $(M-1)q < p_c < Mq$ (Note that $e_{th} < p_c < 1 - e_{th}$). Roughly speaking, this implies that all states which belong to an error correcting code with a finite erasure threshold have long-range irreducible multipartite correlations amongst their random subsystems (for suitably chosen subsystem sizes).

5.5 Final remarks and future directions

In this chapter, we proposed a non-geometric order parameter for k-local nontrivial phases and demonstrated its utility by studying a wide variety of examples. In this section, we conclude by speculating about the behavior of $C(N, M, q)$ in generic quantum systems and list several interesting future directions.

The following is a summary of the expected behavior of $C(N, M, q)$ in generic systems: (1) $C(N, M, q)$ is expected to be nonzero in all geometric topological phases, including states defined on surfaces without a ground state degeneracy, e.g. the surface of a sphere. This is because even though the contribution from the logical qubit vanishes, the large holes close to percolation transition still contribute to $C(N, M, q)$ (see Section 5.3). Therefore, a nonzero value of $C(N, M, q)$ is not necessarily tied with the existence of a protected logical qubit. (2) $C(N, M, q)$ is expected to be nonzero in all "k-local" quantum error-correcting codes with a finite erasure threshold (see Section 5.4). (3) $C(N, M, q)$ is expected to be zero in some gapless phases. In particular, $C(N, M, q)$ is expected to vanish in eigenstates of free systems. This is because in these systems, Wick's theorem implies that two partite density matrices include all the global information. i.e. $C(N, M > 2, q) = 0$. (4) $C(N, M, q)$ is expected to be nonzero in gapless phases that have error correcting properties. Examples include the Motzkin chain [236, 237] and CFTs important in AdS/CFT correspondence [221, 238]. It would be intriguing to understand the behavior of C in these systems.

The results of this chapter demonstrate the basic utility of $C(N, M, q)$ in studying k-local systems - However, much work remains to be done. In particular: (1) It would be interesting if the computation scheme used in this chapter can be extended to states beyond the stabilizer states. (2) While we demonstrated, the applicability of the scaling ansatz in Eq. (5.14) to many examples, we do not have a clear understanding of the meaning of these exponents and the underlying field theory. In particular, it is plausible that the value of these critical exponents encodes some universal information (e.g. dimensionality) of the underlying k-local phase. (3) It would be interesting if we can understand the relation between $C(N, M, q)$ and the circuit complexity. In particular, it is intriguing to see if it is possible to prove the non-triviality of a state that has power-law decaying $C(N, M, q)$ (as a function of M). (4) It is interesting to understand if our order parameter can be used to differentiate distinct volume law phases.

Appendix A: Dehn Twist

Consider the annulus A shown in Fig. A.1a which consists of the points in the (r, θ)-plane with $1 \leq r \leq 2$ and $0 \leq \theta < 2\pi$. Let $T : A \to A$ denote the twist map given as:

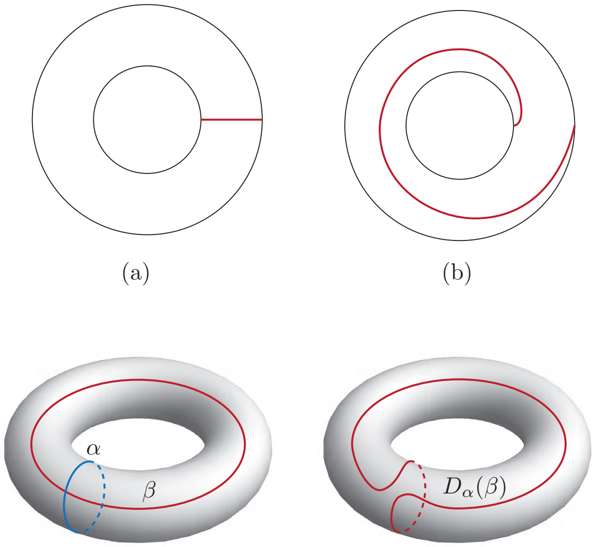
$$T(r, \theta) = (r, \theta + 2\pi(r - 1)). \tag{A.1}$$

Note that T is an orientation preserving homeomorphism which reduces to the identity map on the boundaries of A. Fig. A.1b shows the image of a line in A under T.

Now, let Σ denote an arbitrary oriented surface and let α be a simple closed curve on Σ. To define the Dehn twist around α, first we choose a regular neighborhood $\mathcal{A}$ of α on Σ which is homeomorphic to A. Let $\phi : A \to \mathcal{A}$ denote an orientation preserving homeomorphism from A to $\mathcal{A}$. We define the twist map around α by the following homeomorphism $T_\alpha : \Sigma \to \Sigma$

$$T_\alpha(x) = \begin{cases} \phi \circ T \circ \phi^{-1}(x) & x \in \mathcal{A} \\ x & x \notin \mathcal{A} \end{cases} \tag{A.2}$$

T_α clearly depends on the choice of N, ϕ and α. We define the *Dehn Twist* around α, denoted by D_α to be the isotopy class of T_α, i.e. the class of homeomorphisms of Σ to itself that can be deformed continuously to T_α. We remark that D_α depends only on the isotopy class of α.



(a) (b)

(c) (d)

Figure A.1: (a) Annulus A with a typical line connecting its boundaries together. (b) Image of the region A under the twist map. (c) and (d) show the β loop and its image under the Dehn twist around the α loop, respectively.

As an example, one can consider the Dehn twist around the α loop on the simple torus shown in Fig.A.1c. Figure A.1d shows how the β loop gets deformed by the action of D_α.

A simple way to find the image of a given loop like β under the action of the Dehn twist around another loop like α, called the Dehn surgery, is as follows: We start by tracing out the β loop until we hit an intersection with the α loop. Then we turn left to trace out the α loop until we return to the intersection point, where we turn right to continue tracing out the β loop. We need to do the same at any intersection of α and β until we get back to the starting point. Note that with the above definition the Dehn twist around α does not depend on any direction that the α loop might have.

Appendix B: Area scaling calculation

In this appendix we analyze the shearing maps introduced in Section 3.3 to see by how much they scale the area locally. Because these maps do not have any singularity on a genus g surface, the local area scaling is bounded from below and above as one considers a fixed g. But more importantly, we will show that if we consider a family of maps on surfaces with increasing genus g, the local area scaling remains finite and bounded from below. Note that this result is crucial for the retriangulation circuit described in Sec. 3.4 to be constant depth.

As in Chapter 3, we use the Poincare disk model for $\mathbb{H}^2$, namely the open unit disk in $\mathbb{C}$ with the Reimannian metric:

$$\mathrm{d}s^2 = 4\frac{\mathrm{d}x^2 + \mathrm{d}y^2}{(1-r^2)^2}. \tag{B.1}$$

In this model, the geodesics connecting two points would be either circular arcs perpendicular to the unit disk or straight lines passing through the origin. It would be useful to note that a line reflection in $\mathbb{H}^2$ looks like circle inversion in the unit disk model.

A genus g hyperbolic surface can be obtained by compactifying a canonical N-gon for $N = 4g$. A canonical N-gon plotted in the Poincare disk model and centered at the origin would have its vertices at points:

$$z_j = \sqrt{\cos\left(\frac{2\pi}{N}\right)} e^{i2\pi j/N}, \quad j = 0, \cdots, N-1. \tag{B.2}$$

The length of the sides, as measured using the hyperbolic metric, would be (see Fig.B.1a):

$$|AB| = 4\tanh\left(\frac{\cos(\pi/N) - \sin(\pi/N)}{\sqrt{\cos(2\pi/N)}}\right). \tag{B.3}$$

The $\{N, N\}$-tiling of $\mathbb{H}^2$ can be obtained by reflecting the canonical N-gon with respect to its sides and repeating this procedure indefinitely.

We start by analyzing the α_1 Dehn twist in detail. With minor modifications, the same calculation applies to the other maps as well, so we will only mention the end results for the other maps.

For the points in the shaded region but outside the $ABCD$ hyperbolic 4-gon in Fig. 3.4b, D_{α_1} acts as identity and hence it is preserves area locally. Next we consider the points inside ΔABC which are mapped to points inside ΔACE (See Fig.B.1).

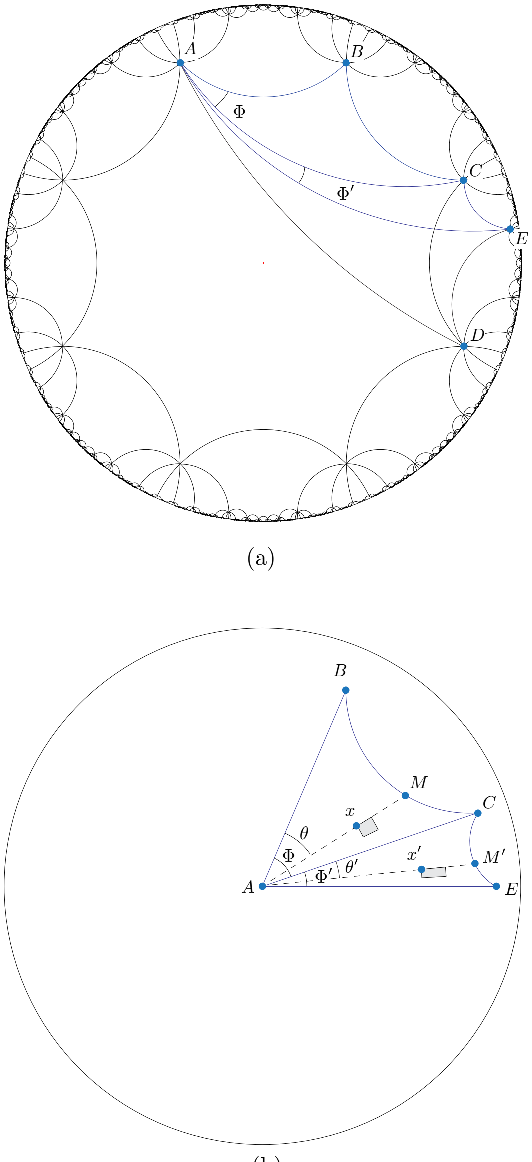
We start by calculating some lengths and angles. As in the main text, $|\,|$ denotes the hyperbolic length. $|AC|$ can be obtained using the hyperbolic law of cosines (see Fig.B.1a):

$$\cosh(|AC|) = \cosh(|AB|)^2 - \sinh(|AB|)^2 \cos(2\pi/N), \tag{B.4}$$

where we used the fact $|BC| = |AB|$. Then, we can calculate the angle $\Phi \equiv \angle BAC = \angle BCA$ by the hyperbolic law of sines:

$$\sin(\Phi) = \frac{\sinh(|AB|)}{\sinh(|AC|)} \sin(2\pi/N). \tag{B.5}$$

Note that $|CE| = |AB|$ and $\angle ECD = 2\pi/N$. Therefore, $|AE|$ and $\Phi' \equiv \angle CAE$ can be



(a)

(b)

Figure B.1: (a) $D^*_{\alpha_1}$ maps the ΔABC and ΔACD triangles to the ΔACE and ΔAED triangles respectively. (b) By moving the origin of the Poincarè disk to the point A, the geodesics AB, AC and AE turn into straight lines while the angles between them remains the same. A typical area element at point x alongside its image at x' is shown on the figure.

computed using the hyperbolic laws of cosines and sines respectively:

$$\begin{aligned}\cosh(|AE|) = \ & \cosh(|AB|)\cosh(|AC|) \\ & - \sinh(|AB|)\sinh(|AC|)\cos(4\pi/N - \Phi)\end{aligned} \tag{B.6}$$

$$\sin(\Phi') = \frac{\sinh(|AB|)}{\sinh(|AE|)}\sin(4\pi/N - \Phi). \tag{B.7}$$

To make calculations simpler, we place the origin of the Poincarè disk on point A; by doing so, the geodesics that come out of the point A will look like straight lines on the disk. In Fig. B.1b, ΔABC and ΔACE are plotted in this new coordinate frame.

Consider an infinitesimal area element $\mathrm{d}s$ at $x = (r, \theta)$ inside ΔABC which is mapped to the area element $\mathrm{d}s'$ at $x' = (r', \theta')$ inside ΔACE; r and r' are the Euclidean distance from origin in Fig. B.1b and the θ and θ' angles are measured with respect the lines AB and AC respectively. we define $\lambda(r, \theta)$ as the scale factor of the area:

$$\lambda_N(r, \theta) \equiv \frac{\mathrm{d}s'}{\mathrm{d}s} = \sqrt{\frac{g(r')}{g(r)}}\frac{r'\mathrm{d}r'}{r\mathrm{d}r}\frac{\mathrm{d}\theta'}{\mathrm{d}\theta}, \tag{B.8}$$

where $g(r)$ is the determinant of the hyperbolic metric:

$$g(r) = \frac{16r^2}{(1 - r^2)^4}. \tag{B.9}$$

We want to show that λ is bounded from above and below:

$$L < \lambda_N(r, \theta) < U, \tag{B.10}$$

for some real positive L and U. First, we need to find r' and θ' in terms of r and θ.

Using the law of cosines for the angles in ΔABM (see Fig. B.1b) gives:

$$\begin{aligned}\cos(\angle AMB) = &-\cos(\theta)\cos\left(\frac{2\pi}{N}\right)\\ &+\sin(\theta)\sin\left(\frac{2\pi}{N}\right)\cosh(|AB|)\end{aligned} \tag{B.11}$$

And we can use this result to find $|BM|$ and $|AM|$:

$$\sinh(|AM|) = \frac{\sin(2\pi/N)}{\sin(\angle AMB)}\sinh(|AB|) \tag{B.12}$$

$$\sinh(|BM|) = \frac{\sin(\theta)}{\sin(\angle AMB)}\sinh(|AB|) \tag{B.13}$$

$D^*_{\alpha_1}$ is defined such that:

$$|CM'| = \frac{|BM|}{|BC|}|CE|. \tag{B.14}$$

Note that in this case $|BC| = |CE|$ and hence:

$$|CM'| = |BM|. \tag{B.15}$$

We can find $|AM'|$ by using the law of cosines in $\Delta ACM'$:

$$\cosh(|AM'|) = \cosh(|CM'|)\cosh(|AC|) \tag{B.16}$$

$$-\sinh(|CM'|)\sinh(|AC|)\cos\left(\frac{4\pi}{N}-\Phi\right). \tag{B.17}$$

It is then straight forward to find r' and θ'. According to the way the map $D^*_{\alpha_1}$ is defined, we have:

$$|Ax'| = \frac{|AM'|}{|AM|}|Ax|. \tag{B.18}$$

Note that $r' = \tanh(|Ax'|/2)$ and $|Ax| = 2\,\mathrm{arctanh}(r)$. So we have:

$$r' = \tanh(\rho\,\mathrm{arctanh}(r)), \tag{B.19}$$

where ρ is equal to $\frac{|AM'|}{|AM|}$ and depends only on θ(not r). By taking the derivative of (B.19) with respect to r we get:

$$\frac{\mathrm{d}r'}{\mathrm{d}r} = \rho(\theta)\frac{1-r'^2}{1-r^2} \tag{B.20}$$

θ' can also be obtained by using the hyperbolic law of sines in $\Delta ACM'$:

$$\sin(\theta') = \frac{\sinh(|CM'|)}{\sinh(|AM'|)}\sin\left(\frac{4\pi}{N} - \Phi\right) \tag{B.21}$$

By plugging (B.19),(B.20) and (B.9) into (B.8) we find:

$$\lambda_N(r,\theta) = \left(\frac{\tanh(\rho\,\mathrm{arctanh}(r))}{r}\right)^2\left(\frac{1+r}{1+\tanh(\rho\,\mathrm{arctanh}(r))}\right)\left(\frac{1-r}{1-\tanh(\rho\,\mathrm{arctanh}(r))}\right)\rho(\theta)\frac{\mathrm{d}\theta'}{\mathrm{d}\theta}. \tag{B.22}$$

First, we fix θ and see how λ changes as one varies r in the range $[0, r_{\text{max}} = \tanh(|AM|/2)]$. It is straight forward to show that $\tanh(\rho\,\mathrm{arctanh}(r))/r$ is bounded by 1 and ρ. The second parentheses in (B.22), which is equal to $(1+r)/(1+r')$, also is bounded by $1/2$ and 2. The third parenthesis in (B.22) is a monotonic function of r, as one can verify by taking the derivative, and so is bounded by 1 and $(1-\tanh(|AM|/2))/(1-\tanh(|AM|'/2))$. Therefore, to make sure that $\lambda_N(r,\theta)$ is bounded, it suffices to show that $\rho(\theta)=\frac{|AM|'}{|AM|}$, $(1-\tanh(|AM|/2))/(1-\tanh(|AM|'/2))$ and $\frac{\mathrm{d}\theta'}{\mathrm{d}\theta}$ remain finite as one changes θ from 0 to Φ.

So far, all expressions were exact. But, since we are interested in the $N\to\infty$ limit, the calculation can be simplified greatly by computing the large N expansion of each expression. In particular, (B.3),(B.4),(B.6),(B.5) and (B.7) have the following asymptotic forms:

$$\begin{aligned}
|AB| &= -2\ln\left(\frac{\pi}{2N}\right)-\frac{7\pi^2}{6N^2}-\frac{487\pi^4}{720N^4}+\mathcal{O}(1/N^5)\\
|AC| &= -2\ln\left(\frac{\pi}{4N}\right)-\frac{61\pi^2}{24N^2}+\mathcal{O}(1/N^3)\\
|AE| &= -2\ln\left(\frac{\pi}{14N}\right)+\mathcal{O}(1/N)\\
\Phi &= \frac{\pi}{2N}+\frac{3\pi^3}{8N^3}+\mathcal{O}(1/N^4)\\
\Phi' &= \frac{\pi}{14N}+\mathcal{O}(1/N^2)
\end{aligned}\tag{B.23}$$

To find the asymptotic form of functions that involve θ, first we trade θ for $\eta\equiv\frac{\theta}{\Phi}$. The reason is that θ varies in the range $[0,\Phi]$ and so has an implicit $1/N$ dependence because of Φ. By using η instead of θ, the only small parameter of our expressions would be $1/N$. Note that $0\le\eta\le 1$. In terms of η, we get the following asymptotic forms for the expressions in B.12,B.16 and B.21:

$$|AM| = -\ln\left(\frac{\pi\sqrt{\eta(1-\eta)}}{4N}\right)+\mathcal{O}(1/N)\tag{B.24}$$

$$|AM'| = -\ln\left(\frac{\pi\sqrt{\eta(1-\eta)}}{(16+33\eta)N}\right)+\mathcal{O}(1/N)\tag{B.25}$$

$$\theta' = \frac{7\pi\eta}{(32+66\eta)N}+\mathcal{O}(1/N^2)\tag{B.26}$$

The first two expressions are only valid for $0 < \eta < 1$; for $\eta = 0, 1$ we have to use the expressions listed in (B.23) instead. Note that for $\eta = 0$, we have $|AM| = |AB|$ and $|AM'| = |AC|$. Similarly, $|AM| = |AC|$ and $|AM'| = |AE|$ when $\eta = 1$. It follows then,

$$\begin{aligned}
&\lim_{N\to\infty} \rho(\eta) = 1 \\
&\lim_{N\to\infty} \frac{1-\tanh(|AM|/2)}{1-\tanh(|AM|'/2)} = \frac{4}{16+33\eta} \\
&\lim_{N\to\infty} \frac{\mathrm{d}\theta'}{\mathrm{d}\theta} = \frac{122}{(16+33\eta)^2},
\end{aligned} \tag{B.27}$$

and clearly all of them are bounded as a function of η. Thus we conclude that the inequality (B.10) holds and therefore, when mapping ΔABC to ΔACE, local area scaling is finite and bounded form above and belllow.

It remains to show that the same holds when mapping ΔACD to ΔAED. The steps are quite the same. First we calculate the side lengths and angles of these triangles. Note that $|DE| = |AC|$,$|CD| = |AB|$ and $\angle ACD = 2\pi/N - \Phi$. $|AD|$ and $\Phi'' \equiv \angle CAD$ can then be obtained using the hyperbolic law of cosines and sines respectively:

$$\begin{aligned}
\cosh(|AD|) = {} & \cosh(|AC|)\cosh(|AB|) \\
& - \sinh(|AC|)\sinh(|AB|)\cos\left(\frac{2\pi}{N} - \Phi\right),
\end{aligned} \tag{B.28}$$

$$\sin(\Phi'') = \frac{\sinh(|AB|)}{\sinh(|AD|)} \sin\left(\frac{2\pi}{N} - \Phi\right). \tag{B.29}$$

Let M be a point on the side CD and M' its image which will be on ED. According to the definition of the map,

$$|EM'| = \frac{|DE|}{|CD|}|CM| = \frac{|AC|}{|AB|}|CM| \tag{B.30}$$

We define ρ, θ, θ', and η similar to the previous case: $\theta \equiv \angle MAC$, $\theta' \equiv \angle M'AE$, $\rho(\theta) \equiv |AM'|/|AM|$ and $\eta \equiv \theta/\Phi''$. As in the previous section, we only need to compute the corresponding limits listed in (B.27). The calculation follows the same steps and at the end we will find that:

$$\lim_{N\to\infty} \rho(\eta) = 1 \tag{B.31}$$

$$\lim_{N\to\infty} \frac{1-\tanh(|AM|/2)}{1-\tanh(|AM|'/2)} = \frac{8}{49-45\eta} \tag{B.32}$$

$$\lim_{N\to\infty} \frac{\mathrm{d}\theta'}{\mathrm{d}\theta} = \frac{112}{(49-45\eta)^2}. \tag{B.33}$$

Since all of them are bounded for $0 \leq \eta \leq 1$ it follows that the area elements in ΔACD also expand or shrink by a finite factor under this transformation.

Putting everything together, we conclude that $D^*_{\alpha_1}$ has finite local area scaling over the entire polygon and this ends the proof.

The situation for β Dehn twists are exactly the same up to some relabeling and needs no more analysis. One can also carry out similar calculations to check that the same remains true for the γ Dehn twists. Note that D^*_γ is essentially the same as D^*_α; the only difference is that the polygon is no longer regular.

Appendix C: Dehn Twists on Σ_3

Although we used a $g = 2$ hyperbolic surface to describe our procedure, as we mentioned in the main text the protocols can be easily generalized to higher genus surfaces. In this section, we briefly explain how the generalization applies in the $g = 3$ case. The genus 3 surface Σ_3, shown in Fig. C.1, can be constructed by identifying every other edge of a 12-gon (see Fig. C.2a). The space of possible hyperbolic metrics – Teichmüller space – corresponds to inequivalent choices for the locations of the vertices of the 12-gon. Here we consider a regular 12-gon for simplicity.

The sides of the 12-gon then correspond to the canonical closed loops of the closed surface which generate the fundamental group of Σ_3:

$$\pi_1(\Sigma_3) = \langle \alpha_1, \beta_1, \cdots, \alpha_3, \beta_3 \mid \prod_{i=1}^{3} \alpha_i \beta_i \alpha_i^{-1} \beta_i^{-1} = 1 \rangle. \tag{C.1}$$

Similar to the genus 2 case, we can tile the hyperbolic plane by attaching copies of the standard 12-gon appropriately together, which then specifies a universal covering for Σ_3. We can choose one of the standard 12-gons as the fundamental domain of the covering. Such a tiling is plotted in Fig.C.2a.

The mapping class group of Σ_3 can be generated by the Dehn twists around the α and β loops

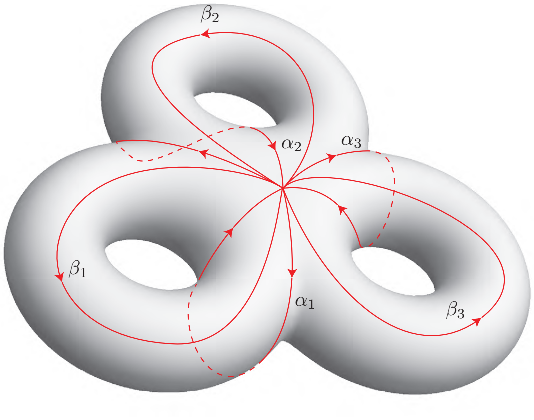


Figure C.1: Σ_3 surface and generators of its fundamental group.

plus the Dehn twists along γ_1 and γ_2 loops, where:

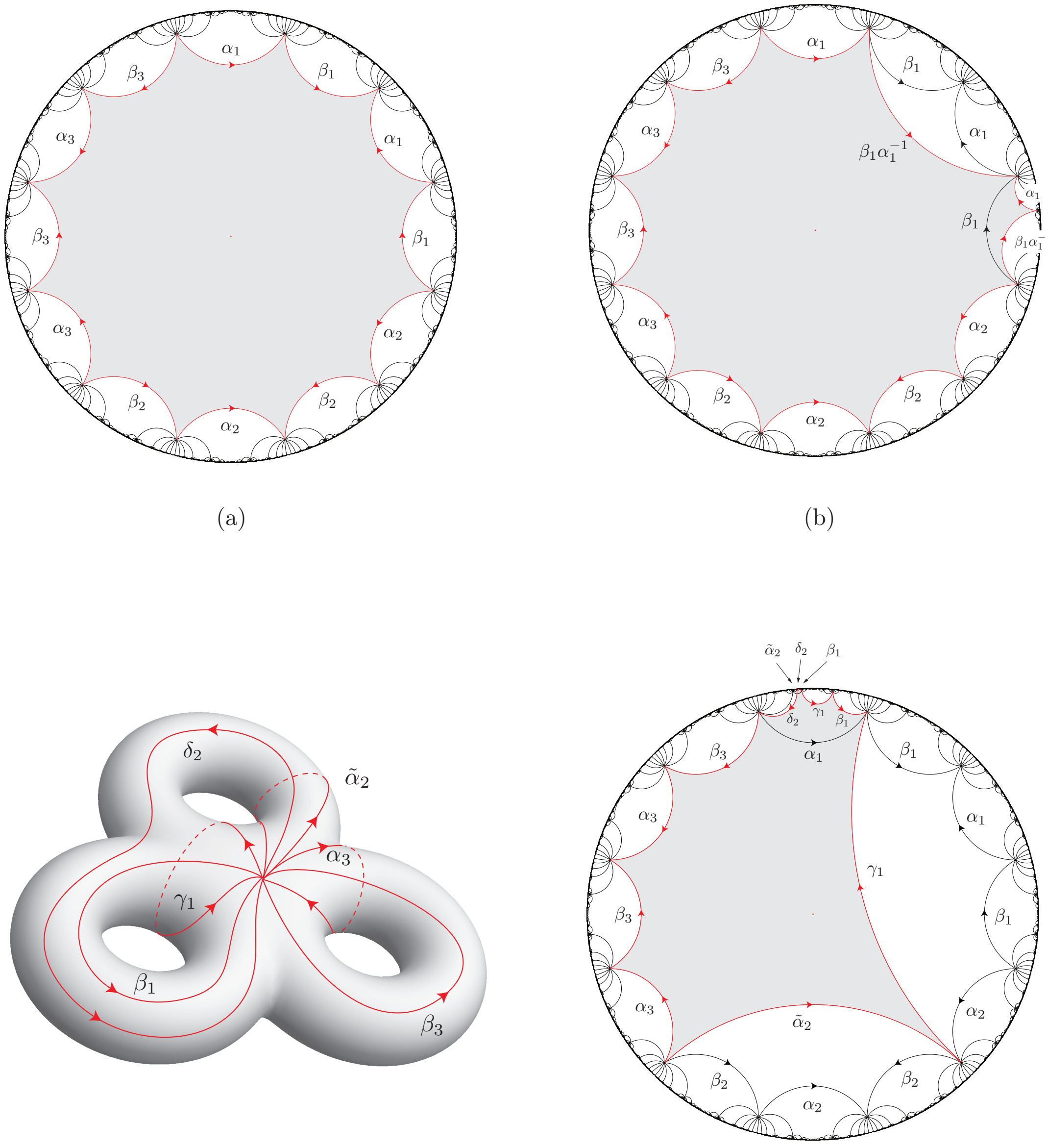
$$\gamma_i = \alpha_{i+1}^{-1}\tilde{\alpha}_i, \qquad i = 1, 2 \tag{C.2}$$

and where $\tilde{\alpha}_i$ is the α_i but transported along β_i:

$$\tilde{\alpha}_i = \beta_i \alpha_i \beta_i^{-1} \tag{C.3}$$

To implement the Dehn twists, it suffices to find appropriate shearing maps to permute the qubits accordingly. The subsequent step, which is the re-triangulation, has already been described in the general case in Section 3.4. The representative maps for the Dehn twists along α and β loops are essentially the same as the ones described in Section 3.3 for the $g = 2$ case. As an example, Fig. C.2b shows how the fundamental domain is transformed by $D^*_{\alpha_1}$ in this case.

The representative maps for Dehn twists along γ_is are also straightforward generalizations of the map described in Section 3.3; the trick is to choose a 12-gon as the fundamental domain such



(a)

(b)

(c)

(d)

Figure C.2: (a) The regular 12-gon corresponding to the Σ_3 surface. (b) The image of the fundamental domain shown in (a) under the action of $D^*_{\alpha_1}$. (c) New set of loops on the surface of Σ_3 (d) The fundamental domain that is used to define $D^*_{\gamma_1}$ which can be obtained by cutting the Σ_3 surface along the loops shown in (c). The sides associated with $\tilde{\alpha}_2$, β_1, δ_1 marked by the arrows could not be drawn completely due to size constraint.

that it has the γ_i loop as two of its sides and only the sides neighboring γ_i transform non-trivially by the representative Dehn twist map. Here we consider the Dehn twist along γ_1. The Dehn twist along γ_2 follows similarly. The action of D_{γ_1} on α_1, α_2, β_1 and β_2 follows from the expressions in (3.13) by replacing γ with γ_1. Furthermore, it keeps α_3 and β_3 invariant. The loop $\delta_2 = \beta_2\beta_1$ also remains invariant under D_{γ_1}. Just like (3.14) for the $g = 2$ case, we can rewrite the group relation in (C.1) as:

$$\delta_2^{-1}\tilde{\alpha}_2\delta_2\beta_1^{-1}\gamma_1\beta_1\gamma_1^{-1}\tilde{\alpha}_2^{-1}\alpha_3\beta_3\alpha_3^{-1}\beta_3 = 1. \tag{C.4}$$

This form then suggests trading $\{\alpha_1, \beta_1\}$ with $\{\gamma_1, \beta_1\}$ and $\{\alpha_2, \beta_2\}$ with $\{\tilde{\alpha}_2, \delta_2\}$. Fig. C.2c shows these new set of loops on the surface of Σ_3. If we cut Σ_3 along these loops, we end up with the the shaded irregular 12-gon shown in Fig. C.2d, which can be taken as the fundamental domain of the covering map. Note that in this 12-gon all sides remain invariant by D_{γ_1} except the two sides labeled by β_1. Moreover, the β_1 sides are neighboring the γ_1 sides as was the case in $g = 2$. Therefore a shearing map directly analogous to D^*_γ in Section 3.3 would work here as well.

Appendix D: Entanglement entropy in stabilizer formalism

A stabilizer state $|\psi\rangle$ over N qubits is specified by a set $\mathcal{S} = \{s_1, \cdots, s_N\}$ of N independent, mutually commuting Pauli strings operators s_i, such that

$$s_i |\psi\rangle = |\psi\rangle . \tag{D.1}$$

Clearly, there are many equivalent choices of $\mathcal{S}$ that result in the same stabilizer state $|\psi\rangle$. However, given a stabilizer set $\mathcal{S}$, the elements s_i generate an abelian group $\mathcal{G}_\psi = \langle s_1, s_2, \cdots, s_N\rangle$ under multiplication, which is determined uniquely by the stabilizer state $|\psi\rangle$.

The density matrix of the system in the stabilizer state $|\psi\rangle$ can be written as [200]

$$\rho = |\psi\rangle\langle\psi| = \frac{1}{2^N} \sum_{g \in \mathcal{G}_\psi} g. \tag{D.2}$$

Given a bipartition of the qubits into two sets A and B, the reduced density matrix of ρ over A can be obtained by tracing equation (D.2) over B, which yields

$$\rho_A = \frac{1}{2^{n_A}} \sum_{g \in \mathcal{G}_{A,\psi}} g, \tag{D.3}$$

where n_A is the number of qubits in A and $\mathcal{G}_{A,\psi} \subseteq \mathcal{G}_\psi$ is the subgroup of the stabilizers which are entirely contained in A, i.e. they act as identity on the qubits outside A. The von Neumann entanglement entropy of ρ_A is given by

$$S_A(|\psi\rangle) = n_A - \log_2 |\mathcal{G}_{A,\psi}|, \tag{D.4}$$

where $|\mathcal{G}|$ stands for the number of elements in group $\mathcal{G}$ [200]. Moreover, $R_\alpha(\rho_A)$, the Renyi entropy of order α, is actually independent of α and is equal to von Neumann entanglement entropy.

Let $|\psi\rangle$ be a stabilizer state specified by the stabilizer set $\mathcal{S} = \{s_1, s_2, \cdots, s_N\}$. Consider a closely related stabilizer state $|\psi'\rangle$ which is specified by the stabilizer set $\mathcal{S}' = \{(-1)^{n_1}s_1, (-1)^{n_2}s_2, \cdots, (-1)^{n_N}$ where each n_i is either 0 or 1. The following Lemma shows that $|\psi\rangle$ and $|\psi'\rangle$ are indistinguishable as far as the entanglement entropy is concerned.

Lemma D.1. *For $|\psi\rangle$ and $|\psi'\rangle$ defined as above and for any subset A of the qubits,*

$$S_A(|\psi\rangle) = S_A(|\psi'\rangle). \tag{D.5}$$

Proof. Let $\mathcal{G}\psi$ and $\mathcal{G}_{\psi'}$ denote the stabilizer groups associated with $|\psi\rangle$ and $|\psi'\rangle$ respectively. Consider the group homomorphism h between $\mathcal{G}_\psi$ and $\mathcal{G}_{\psi'}$ defined by its action on the generators of $\mathcal{G}_\psi$ as

$$h : \mathcal{G}_\psi \longmapsto \mathcal{G}_{\psi'}$$

$$h(s_i) = (-1)^{n_i} s_i.$$

Since h maps a generator set to another, it is bijective. Moreover, it is straightforward to verify that for any subset A of qubits, h maps $\mathcal{G}_{A,\psi}$ to $\mathcal{G}_{A,\psi'}$. The Lemma's claim then follows immediately from equation (D.4). □

Given a stabilizer state $|\psi\rangle$, one has the freedom to choose any N independent elements from $\mathcal{G}_\psi$ to form the stabilizer set $\mathcal{S}$. We can use this gauge freedom to impose certain conditions on the elements of $\mathcal{S}$.

Define the left (right) endpoint of a stabilizer s to be the first (last) site on which s acts nontrivially. Given a set of stabilizers $\mathcal{S}$, let $\rho_l(i)$ denote the the number of stabilizers whose left endpoint resides on site i and define $\rho_r(i)$ similarly with regard to the right end points. As is shown in Ref. [239], one can always choose $\mathcal{S}$ such that

1. For all sites we have $\rho_r(i) + \rho_l(i) = 2$.
2. If $\rho_l(i) = 2$ (or $\rho_r(i) = 2$) for a site i, the two corresponding stabilizers have a different Pauli operator at i.

Such a stabilizer set $\mathcal{S}$ is said to be in the clipped gauge [153]. The utility of the clipped gauge is that the entanglement entropy has a simple form in this gauge. In particular, if the stabilizer set $\mathcal{S}$ is in the clipped gauge, the entanglement entropy of a contiguous region A equals to half the number of stabilizers in $\mathcal{S}$ which have one endpoint in A and another in its complement [153].

Appendix E: Time dynamics of the density matrix in monitored Clifford random quantum circuit

Let $\mathcal{Q}$ denote a specific realization of the quantum circuit laid out in the main text. If we fix the initial state to be $|0\rangle^{\otimes N}$ and run the same circuit many times, due to the randomness in the measurement outcomes, the final state of the system could be different each time. Instead of considering quantum trajectories, we can calculate the expectation value of operators over different runs by viewing the measurements in $\mathcal{Q}$ as quantum channels. Accordingly, the entire circuit can be described as a quantum channel $\mathcal{E}_{\mathcal{Q}}$, which transforms the initial pure density matrix $\rho_0 = (|0\rangle\langle 0|)^{\otimes N}$ to a mixed final density matrix $\rho_* = \mathcal{E}_{\mathcal{Q}}(\rho_0)$.

ρ_* can be used to compute the expectation values of measurements which are averaged over many runs of the circuit without post-selection on the measurement outcomes. In particular, if we run the circuit $\mathcal{Q}$ many times, measure a fixed operator O each time and average the result over a large number of runs, the value we get would be

$$\bar{O} = \mathrm{tr}(O\rho_*). \tag{E.1}$$

Here we show that with probability one, ρ_* is actually independent of the underlying circuit. In other words, if $\mathcal{Q}$ is any fixed quantum circuit chosen with the distribution associated with probabilities $0 < p_s, p_t < 1$ and $p_u < 1$, the final density matrix of the system is always given by

$$\rho_* = \mathcal{E}_{\mathcal{Q}}(\rho_0) = \frac{1}{2^{N-2}} \Pi_{G_1,+} \Pi_{G_2,+}. \tag{E.2}$$

where $\Pi_{G_i,+}$ is the projection operator on the $G_i = 1$ subspace. Note that, not only does ρ_* not depend on the specific realization $\mathcal{Q}$, but it is also independent of p_s and p_u, which means that as far as the expectation value of operators is concerned, the entire phase space looks the same. Moreover, we show that the time it takes for the density matrix to reach the steady state is constant for $p_u = 0$ while it is at most $\mathcal{O}(N)$ for $p_u \neq 0$.

For a general Pauli string operator S, the quantum channel corresponding to its measurement is given by

$$\mathcal{E}_S(\rho) = \Pi_{S,+}\, \rho\, \Pi_{S,+} + \Pi_{S,-}\, \rho\, \Pi_{S,-} \tag{E.3}$$

where $\Pi_{\pm}$ denote the projectors onto $S = \pm 1$ subspaces, i.e.

$$\Pi_{S,\pm} = \frac{1}{2}(\mathbb{1} \pm S). \tag{E.4}$$

By using the explicit form of the projectors $\Pi_{S,\pm}$, equation (E.3) can be written as

$$\mathcal{E}_S(\rho) = \frac{1}{2}(\rho + S\, \rho\, S). \tag{E.5}$$

Consider a mixed stabilizer state ρ

$$\rho(\mathcal{G}) = \frac{1}{2^N} \sum_{g \in \mathcal{G}} g, \tag{E.6}$$

for a Pauli group $\mathcal{G} = \langle e_1, \cdot, e_n \rangle$ with $n \leq N$ independent generators. According to equation (E.5), under the measurement of a Pauli string S, we have

$$\mathcal{E}_S(\rho) = \frac{1}{2^{N+1}}(\sum_{g\in\mathcal{G}} g + \sum_{g\in\mathcal{G}} SgS) \tag{E.7}$$

$$= \frac{1}{2^N} \sum_{g\in C_{\mathcal{G}}(S)} g \tag{E.8}$$

$$= \rho(C_{\mathcal{G}}(S)). \tag{E.9}$$

Here $C_{\mathcal{G}}(S)$ is the centralizer of S in $\mathcal{G}$. If S commutes with all elements in $\mathcal{G}$, clearly $C_{\mathcal{G}}(S) = \mathcal{G}$. Otherwise, without loss of generality, we can assume S commutes with all generators of $\mathcal{G}$ except one of them, say e_n. Thus $C_{\mathcal{G}}(S) = \langle e_1, \cdots, e_{n-1} \rangle$.

The above analysis shows that for a mixed stabilizer state, whenever a Pauli string is measured, it either leaves the density matrix untouched or takes it to another mixed stabilizer state with one less generator, depending on whether the measured Pauli string commutes with the corresponding Pauli group or not.

In our case, the initial state of the system is given by

$$\rho(\mathcal{G}_0) = \frac{1}{2^N} \sum_{g\in\mathcal{G}_0} g, \quad \mathcal{G}_0 = \langle G_1, G_2, Z_2, \cdots, Z_{N-1} \rangle. \tag{E.10}$$

Let us first consider the $p_u = 0$ case. Based on the discussion above, it is clear that Z_i measurements never change ρ and thus can be ignored for our purpose. Each time a stabilizer g_i is measured, $\rho(\mathcal{G})$ is transformed to $\rho(C_{\mathcal{G}}(g_i))$. Note that in general we have

$$C_{C_{\mathcal{G}}(S_1)}(S_2) = C_{\mathcal{G}}(\{S_1, S_2\}). \tag{E.11}$$

Thus after all stabilizers g_i have been measured at least once, the density matrix of the system would be given by:

$$\begin{aligned}\rho(C_{\mathcal{G}_0}(\{g_2, \cdots, g_{N-1}\})) &= \rho(\langle G_1, G_2\rangle) \\ &= \frac{1}{2^{N-2}}\Pi_{G_1,+}\Pi_{G_2,+} = \rho_*. \end{aligned} \tag{E.12}$$

Let m_j denote the updating step at which g_j is measured for the first time. It is easy to show that $\mathbb{E}[m_j] = (N-2)/p_t$, where $\mathbb{E}[X]$ denotes the expectation value of X. Therefore, the average time it takes for the system to reach the steady state ρ_* would be

$$\tau_* = \frac{1}{N}\mathbb{E}[\max_j(m_j)] = \frac{1}{N}\max_j(\mathbb{E}[m_j]) = \mathcal{O}(1), \tag{E.13}$$

where the pre-factor $1/N$ is there to convert updating steps to time steps.

Now consider the $p_u \neq 0$ case. Again, we start by the same initial density matrix given by equation (E.10). Each time a measurement is performed, either Z_i or g_i, the Pauli group associated with the density matrix of the system either remains the same or shrinks to one of its subgroups with one less generator, as was explained above. On the other hand, whenever a Clifford unitary U is applied, it just transform $\rho(\mathcal{G})$ to $\rho(U^\dagger \mathcal{G} U)$ with the same number of generators. Now, note that any Pauli group that commutes with every element in the set $\mathcal{M} = \{Z_i\}_{i=1}^{N} \cup \{g_i\}_{i=2}^{N-1}$ should be a subgroup of $\langle G_1, G_2\rangle$ (or the ones which are obtained by substituting G_i with $-G_i$). Therefore, for any Pauli group $\mathcal{G}$ with more than two generators, there is at least one element of $\mathcal{M}$ that does not commute with $\mathcal{G}$. Ergo, at each updating step with probability of at least min$(p_s/N, p_t/(N-2))$, the Pauli group associated with the density matrix would shrink to a subgroup with one less generator, until only two generators G_1 and G_2 remain. Thus, on average, at most it takes $\mathcal{O}(N)$ updating steps until a stabilizer is measured which decreases the number of generators by one. Since we start with N generators, the average time it takes to reach the steady state with only G_1 and G_2 as generators, i.e. ρ_*, would be:

$$\tau_* \leq \frac{1}{N}(N-2)\mathcal{O}(N) = \mathcal{O}(N). \tag{E.14}$$

Appendix F: Mapping the entanglement dynamics to the classical bond percolation

In this section we provide the details regarding the percolation mappings discussed in Chapter 4. The first part pertains to the (1+1)D symmetric random circuit studied in Section 4 and shows how the entanglement dynamics can be related to two decoupled 2d classical bond percolation problems. this section also includes the proof of Proposition 4.1. In the next part, we consider the (2+1)D monitored circuit of Section 4.2 and we show how one can map the entanglement dynamics to a 3d classical bond percolation problem.

F.1 (1+1)D symmetric monitored random quantum circuit

F.1.1 Graphical representation of the state

In this section we develop a graphical description to follow the system's state as it evolves under the random quantum circuit described in the main text for $p_u = 0$. Moreover, this graphical representation provides the basic intuition behind the percolation mapping.

The initial state of the system is given by the stabilizer set $\mathcal{S}_0 = \{Z_1, Z_2, \cdots, Z_N\}$. At each step of the circuit, we measure either a stabilizer g_i or a single qubit operator Z_i on a qubit and update the stabilizer set accordingly. Let $\mathcal{S}_m$ denote the stabilizer set that corresponds to the system's state after m updating steps.

First, we prove the following Lemma:

Lemma F.1. *$\mathcal{S}_m$ can be chosen such that each of its elements, up to a minus sign, is in one of the following forms:*

$$\begin{aligned} Z_{2j+1,2k+1} &\equiv \prod_{i=j}^{k} Z_{2i+1}, \\ Z_{2j,2k} &\equiv \prod_{i=j}^{k} Z_{2i}, \\ g_{2j+1,2k+1} &\equiv \prod_{i=j+1}^{k} g_{2i} = X_{2j+1} Z_{2j+2,2k} X_{2k+1}, \\ g_{2j,2k} &\equiv \prod_{i=j}^{k-1} g_{2i+1} = X_{2j} Z_{2j+1,2k-1} X_{2k}, \end{aligned} \tag{F.1}$$

for some integers j and k.

Proof. We prove the lemma by induction. The claim is clearly true for $\mathcal{S}_0$. Assume it is true for $\mathcal{S}_m$. First consider the case where we measure Z_{2j+1} in the next step. We follow the procedure proscribed by the Gottesman-Knill theorem [240, 241] to obtain the stabilizer set $\mathcal{S}_{m+1}$. If Z_{2j+1} commutes with every stabilizer in $\mathcal{S}_m$, then nothing happens by measuring Z_{2j+1}, hence $\mathcal{S}_{m+1} = \mathcal{S}_m$. So consider the case where some elements of $\mathcal{S}_m$ anti-commute with Z_{2j+1}.

Any element of $\mathcal{S}_m$ that does not commute with Z_{2j+1} has either the form $g_{2j+1,2k+1}$ or $g_{2k+1,2j+1}$ for some k. If there is only one of them, then one only needs to replace it with $\pm Z_{2j+1}$ (with the sign chosen arbitrarily) to obtain $\mathcal{S}_{m+1}$. If there are more than one, we replace the first one with $\pm Z_{2j+1}$, again with the sign chosen arbitrarily, and multiply the others with the stabilizer that was replaced, to get $\mathcal{S}_{m+1}$. In either cases, $\mathcal{S}_{m+1}$ will have the stated form.

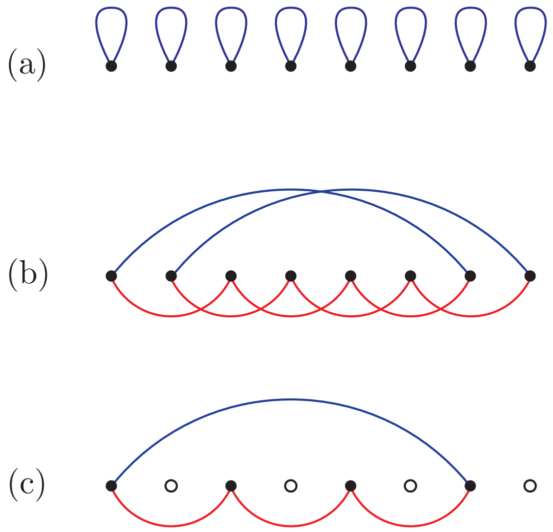


Figure F.1: The diagrammatic representation of the quantum state. **a**, The diagrammatic representation of the product state $|0\rangle^N$. **b**, The diagrammatic representation of the stabilizer state specified by $\mathcal{S} = \{g_i\}_{i=1}^N$. **c** the same as b, but including just the odd sites.

The other possibilities, i.e. measuring other operators at step $m + 1$, can be treated similarly.

$\square$

Based on Lemma F.1, we can use a diagrammatic notation to specify $\mathcal{S}_m$; we put N dots along a line representing the qubits, as is shown in Fig. F.1. Then, for every $g_{a,b}$ element in $\mathcal{S}_m$, we draw a line between sites a, b from below and for every $Z_{a,b}$ element in $\mathcal{S}_m$ draw a line from above. Fig. F.1a and Fig. F.1b show the diagrams corresponding to $\mathcal{S} = \{Z_i\}_{i=1}^N$ and $\mathcal{S} = \{g_i\}_{i=1}^N$ respectively, with g_1 and g_N defined as $g_1 \equiv G_1$ and $g_N \equiv G_2$.

The form of the stabilizers listed in Lemma F.1 suggests a decomposition of the system into odd and even sites. Note that if we measure, for example, Z_{2i+1}, the only stabilizers that could be replaced are in the form $g_{2j+1,2k+1}$ while the $g_{2j,2k}$ stabilizers whose endpoints reside on even sites remain unchanged. Also, if one measures $g_{2i-1,2i+1} = g_{2i}$ whose ends points are on odd

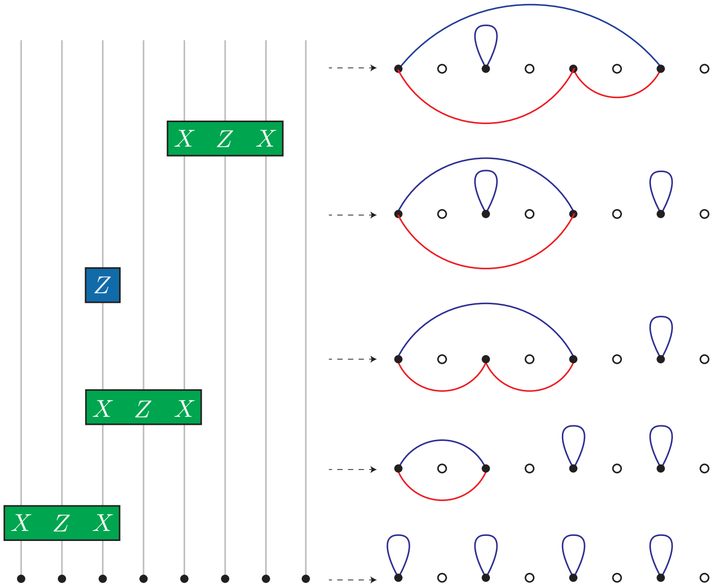


Figure F.2: Step by step evolution of the system under the quantum circuit shown on the left. The diagrammatic representation of the system's state is shown on the right after each measurement.

sites, the only stabilizers that could change have the form $Z_{2j+1,2k+1}$. So, if the stabilizer we are measuring has endpoints on odd sites, we only need to know about the stabilizers in $\mathcal{S}_m$ which also end on odd sites to find $\mathcal{S}_{m+1}$. In other words, we can keep track of the set of stabilizers that start and end on odd sites, without knowing anything about the other stabilizers which start and end on the even sites and vice versa. This allows us to consider odd sites and even sites separately. Fig. F.1c shows the same state as in Fig. F.1b but restricted to odd sites only. For simplicity, we will only consider odd sites in what follows, while similar statements hold for even sites as well.

Using this diagrammatic formalism, it is easy to track $\mathcal{S}_m$. Fig. F.2 shows a typical quantum circuit and the step by step evolution of the system's stabilizer set using the diagrammatic notation developed above.

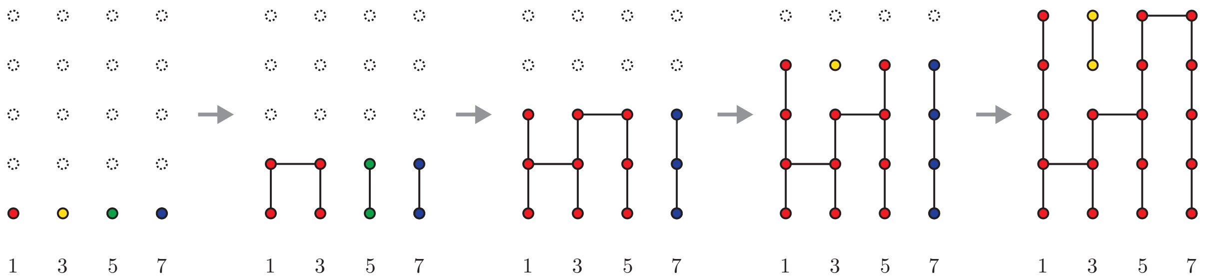


Figure F.3: Step by step evolution of the system under the quantum circuit shown in Fig. F.2 in the percolation picture for the odd sites. At each step, the sites with the same color on the last row represent an isolated SPT phase. There is an analogous diagram for the even sites. The two diagrams together fully specify the entanglement structure of the system.

F.1.2 Proof of Proposition 4.1

We start by noting that for the circuit shown in Fig. F.2, the state of the system can always be described as a collection of isolated SPT states and decoupled qubits, as can be seen from the accompanying diagrammatic representation. For example, in the final state, qubits 1, 5 and 7 form an isolated SPT state, while qubit 3 is decoupled. This observation is indeed true in general. We start by putting forward a precise definition of an isolated SPT state and then show that there is an efficient description of $\mathcal{S}_m$ as a partition of $\{1, 2, \cdots, N\}$.

Definition F.1. *Consider a set of numbers $A = \{q_i\}_{i=1}^n$, such that,*

$$1 \leq q_1 < q_2 < \cdots < q_n \leq N. \tag{F.2}$$

Assume that all numbers are either odd or even. Define its associated stabilizer set, denoted by $\mathcal{S}(A)$, as

$$\mathcal{S}(A) = \{g_{q_i,q_{i+1}}\}_{i=1}^{n-1} \cup \{\prod_{i=1}^{n} Z_{q_i}\}. \tag{F.3}$$

We call such a stabilizer set, an isolated SPT state.

Note that the g stabilizers in $\mathcal{S}(A)$ generate the set of all g strings between any two points in A. Also, note that the stabilizers in equation (F.3) are the same as the ones appearing in Proposition 1.

Lemma F.2. *Let $\mathcal{S}_m$ denote the stabilizer set that corresponds to the system's state after m updating steps. $\mathcal{S}_m$ can always be chosen such that, up to minus signs,*

$$\mathcal{S}_m = \cup_i \mathcal{S}(A_i), \tag{F.4}$$

where A_is correspond to a partition of the qubits into disjoint sets,

$$\cup_i A_i = \{1, 2, \cdots, N\}. \tag{F.5}$$

and $\mathcal{S}(A_i)$ denotes the isolated SPT state corresponding to subset A_i.

Proof. We prove it by induction. It is obviously true for $\mathcal{S}_0$ with $A_i = \{i\}$ for $i = 1, \cdots, N$.

Now assume it is true for $\mathcal{S}_m$, so there exists a partition of qubits given by $\{1, 2, \cdots, N\} = \cup_i A_i$ such that $\mathcal{S}_m = \cup_i \mathcal{S}(A_i)$. First consider the case in which a single qubit operator Z_{2j+1} is measured in the next step. Suppose $2j+1$ is in subset A_k for some k. Note that Z_{2j+1} commutes with any element in $\mathcal{S}(A_{k'})$ with $k' \neq k$. If A_k is the single element set $\{2j+1\}$ (which means $\mathcal{S}(A_k) = \{Z_{2j+1}\}$) then Z_{2j+1} is already in S_m and thus $S_{m+1} = S_m$. If A_k has more than one element, we will show that measuring Z_{2j+1} corresponds to breaking A_k to two subsets of $A_k \setminus \{2j+1\}$ and $\{2j+1\}$.

Note that Z_{2j+1} anti-commutes only with the $g_{a,b}$ elements in $\mathcal{S}(A_k)$ where either a or b equals

$2j+1$. If $2j+1$ is the smallest or largest number in A_k, there is only one such element and by measuring Z_{2j+1}, we just need to replace that element by Z_{2j+1} (with an arbitrary sign) to get the updated stabilizer set S_{m+1}. If $2j+1$ is neither the smallest nor the largest number in A_k, there are two such elements, $g_{a,2j+1}$ and $g_{2j+1,b}$ for some odd numbers a and b. Thus by measuring Z_{2j+1}, one is replaced by Z_{2j+1} (with an arbitrary sign) and the other by $g_{a,2j+1}g_{2j+1,b} = g_{a,b}$ to get the updated stabilizer set. It is easy to verify that in both cases, S_{m+1} is equivalent to the stabilizer set obtained by the union of isolated SPT states corresponding to the same partitioning as for S_m, but with A_k broken to two sets of $A_k \setminus \{2j+1\}$ and $\{2j+1\}$.

Next consider the case where an stabilizer $g_{2j-1,2j+1} = g_{2j}$ is measured in the next step. If $2j-1$ and $2j+1$ belong to the same subset in the partition, nothing happens. If not, let say one belongs to A_k and the other to $A_{k'}$, then $g_{2j-1,2j+1}$ anti-commute with the two Z chains in $\mathcal{S}(A_k)$ and $\mathcal{S}(A_{k'})$ and commutes with everything else in $\mathcal{S}_m$. Therefore, by measuring $g_{2j-1,2j+1}$, we replace one of the Z chains with $\pm g_{2j-1,2j+1}$ (with an arbitrary sign) and the other with the product of the two Z chains, which is just the Z chain over $A_k \cup A_{k'}$, to get S_{m+1}. It is straightforward to verify that S_{m+1} is equivalent to the stabilizer set obtained by union of isolated SPT states corresponding to the same partitioning as for S_m, but by merging the two subsets A_k and $A_{k'}$ into a single subset $A_k \cup A_{k'}$. □

Based on Lemma F.2, there is a one-to-one mapping between partitions of $\{1, 2, \cdots, N\}$ and the state of the system. Moreover, as can be seen from the Lemma's proof, the dynamics of the system can be translated into merging and splitting of the subsets.

Let us specify a partition by assigning unique colors to the qubits in the same subset. Then, whenever a Z operator is measured, a new unique color should be assigned to the corresponding qubit to account for the new single element subset that is created in the new partitioning. On the other hand, when a g operator with end points in different subsets is measured, the two subsets merge together which translates into assigning the same color to qubits in either one. The dynamics we have just described emerges naturally in the percolation model and thus can be used to map the quantum circuit to an instance of percolation on the square lattice. We use a $N/2 \times M$ square lattice, where M is the total number of updating steps. The m'th row of the lattice corresponds to the state of the system after the updating step m. We start by $N/2$ dots with distinct colors at the lowest row which corresponds to the initial product state. If Z_{2i+1} is measured at step m, we leave the vertical link between $(2i+1, m)$ vertex and its history at $(2i+1, m-1)$ broken, while connecting all the other vertical links between the rows m and $m-1$. By doing so, the $(2i+1, m)$ vertex gets a new color, while all the other vertices retain their color form the previous row, which agrees with the aforementioned splitting. On the other hand, if a stabilizer $g_{2i-1,2i+1}$ is measured at step m, we first connect all the vertical links between the rows m and $m+1$, and then connect the vertices at $(2i-1, m)$ and $(2i+1, m)$ to enforce their colors to be the same, thus accounting for the aforementioned merging. Therefore, in each step, the colors of the last updated row can be used to find the partitioning of qubits mentioned in Lemma F.2, which completes the proof of Proposition 1.

As an example, Fig. F.3 shows the step by step development of the circuit described in Fig. F.2, in the percolation picture.

It is worth noting that the stabilizer set given in Lemma F.2 is already in the clipped gauge (see Supplementary Section D for the definition of clipped gauge). Therefore, the entanglement structure can be inferred readily from the percolation picture. In particular, the entanglement entropy $S(x)$ of the region $[1, x]$, is equal to the number of clusters with support on both inside and outside of the region $[1, x]$ on the top row of the percolation lattices. Such a quantity can be computed using the percolation CFT and the coefficient of the logarithm turns out to be:

$$a_x = \frac{\ln(2)\sqrt{3}}{2\pi} \simeq 0.191 \tag{F.6}$$

See for example equation (3) in Ref. [242] (see also [168, 243]). The $2\ln(2)$ discrepancy between equation (F.6) and equation (3) of Ref. [242] is due to the fact that we have two copies of percolation and the logarithm in our definition of entanglement entropy is in base 2.

F.2 (2+1)D topological monitored random quantum circuit

In this section we present the mapping between the random quantum circuit with only stabilizers and single qubit Z measurements to a classical bond percolation problem on the 3D cubic lattice. The quantum circuit could be viewed as either the $p_y = 0$ line of the projective random quantum circuit or the $p_u = 0$ line of the hybrid random quantum circuit.

For simplicity, we consider the infinite plane geometry so we can ignore the non-trivial cycles of the torus as well as the subtleties arising near the boundary. Moreover, we work with the standard version of toric code where the qubits are on the edges and the X and Z stabilizers correspond to the star and plaquette operators respectively.

We take the initial state to be the eigenstate of all the star and plaquette operators. Note that since single qubit measurements are only in Z direction, the system will remain to be an eigenstate of plaquette operators. Therefore, we ignore all plaquette operator measurements in what follows.

We place a square on each vertex which represents the corresponding star operator (Fig.F.4a).

We start with all squares having a unique color. At each time step, the colors are updated as follows

- Whenever a star operator is measured, it acquires it own unique color (Fig.F.4b)
- Whenever a single qubit is measured, the two adjacent squares acquire the same color (Fig.F.4c)

As we will show, these rules represent how the stabilizer state of the system evolves. On the other hand, these set of rules arise naturally in a percolation picture. Consider a $3D$ cubic lattice, henceforth called the percolation lattice, and place a square on each vertex. The vertical direction corresponds to time. If a star operator is *not* measured at time t, we connect the corresponding squares on the $(t-1)$'th and t'th layers. And if a qubit is measured at time t, we connect the corresponding adjacent squares on the t'th layer. At the end, we assign a unique color to each cluster of connected squares. The coloring that will arise is exactly what one would have found if the rules listed above were followed (see Fig.F.4).

The following proposition shows how the stabilizers specifying the state can be inferred from a given coloring configuration.

Proposition F.1. *Let $A^j = \{s_i\}_{i=1}^n$ denote the set of squares corresponding to j'th color, in some arbitrary order. Up to a minus sign, the following operators stabilize the state:*

$$\overline{\mathcal{X}}_j = \prod_{i=1}^{n} \mathcal{X}_{s_i} \qquad \textit{and} \qquad Z_{s_i,s_{i+1}}, \tag{F.7}$$

where $\mathcal{X}_{s_i}$ is the star operator at square s_i and $Z_{s_i,s_{i+1}}$ is the Pauli Z string operator which starts on s_i and ends on s_{i+1}.

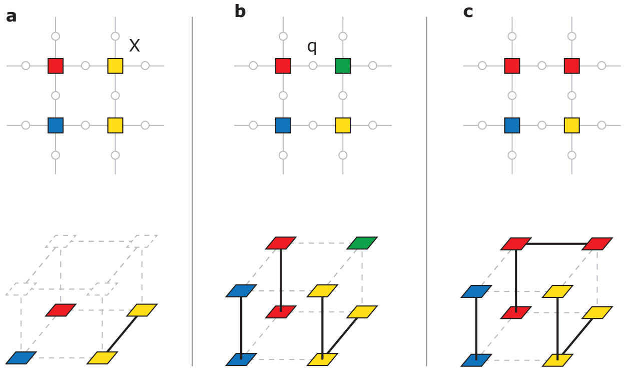


Figure F.4: Diagrammatic representation of coloring rules outlined in the text arise naturally in the 3D percolation picture. (a) The initial coloring of the squares (top) and the corresponding state in the percolation lattice (bottom). (b) The state of the system after the star stabilizer marked by a X in panel "a" is measured. (c) the state of the system after the qubit marked by q in panel b is measured in Z basis.

Note that, because the quantum state is an eigenstate of every plaquette operator and because the space manifold has trivial topology, we don't need to specify the exact path that $Z_{s_i,s_{i+1}}$ string operator takes.

Proof of Proposition F.1. We prove by induction. At $t = 0$ it is clearly the case.

Now assume it is true at step m. Let's say we measure the star operator at some vertex s_1 at step $m + 1$. If the corresponding square has a unique color already, nothing happens and the statement holds trivially afterwards. So let us consider the case where there are more than one squares with the same color as s_1, denoted by the set A^1. Based on the induction assumption, the quantum state is an eigenstate of $\prod_{s\in A^1} \mathcal{X}_s$ operator at step m. Therefore, after measuring $\mathcal{X}_{s_1}$, the system will be an eigenstate of $\mathcal{X}_{s_1}$ as well as $\mathcal{X}_{s_1} \times \prod_{s\in A^1} \mathcal{X}_s = \prod_{s\in A^1\setminus\{s_1\}} \mathcal{X}_s$. Moreover, any $Z_{i,j}$ string operator that doesn't end on s_1 commutes with $\mathcal{X}_{s_1}$ and as such, measurement of $\mathcal{X}_{s_1}$ has no bearing on the system being an eigenstate of it or not. As for the string operator stabilizers starting or ending on s_1, one of them will be replaced with $\mathcal{X}_{s_1}$ stabilizer, and the other with the product of the two which will be an string operator starting and ending on the set $A^1 \setminus \{s_1\}$. Ergo the statement holds at step $m+1$ as well.

Now consider the case in which, at step $m+1$, we measure a single qubit – say qubit q which is between squares s_1 and s_2 – in the Z basis. If s_1 and s_2 have the same color, it results from the induction hypothesis that Z_q already stabilizes the state and the statement trivially holds at step $m+1$. So let us consider the case where s_1 and s_2 have different colors, such that $s_1 \in A^1$ and $s_2 \in A^2$. Since $\prod_{s\in A^1\cup A^2} \mathcal{X}_s$ stabilizes the state at step m and commutes with Z_q, it stabilizes the state at step $m+1$ as well. Based on the induction assumption, the state is stabilized by any $Z_{i,j}$ string that has both end points either on A^1 or A^2. Moreover, it is now stabilized by the string operator which connects $s_1 \in A^1$ and $s_2 \in A^2$, namely Z_q. Therefore the state of the system at step $m+1$ is stabilized by any $Z_{i,j}$ string that has end points on $A^1 \cup A^2$. Hence the statement holds at step $m+1$ as well. □

Proposition F.1 allows us to study the entanglement structure of the steady state, using the percolation picture. In particular, the entanglement entropy of a rectangular region A is equal to the number of clusters which have shared support in A and A^c, minus one.

Proposition F.2. *Let A be a rectangular region with smooth boundary and let $B = A\backslash\partial A$ denote its bulk (see Fig.F.5). For a given coloring configuration of the squares (star operators), let C_A denote the set of colors that appear inside A. Also define C_B^{ex} to be the colors that exclusively appear in the bulk. Then, for the stabilizer state which is associated with that coloring, the entanglement entropy of the subset of qubits inside A can be expressed as,*

$$S_A = |C_A \setminus C_B^{ex}| - 1. \tag{F.8}$$

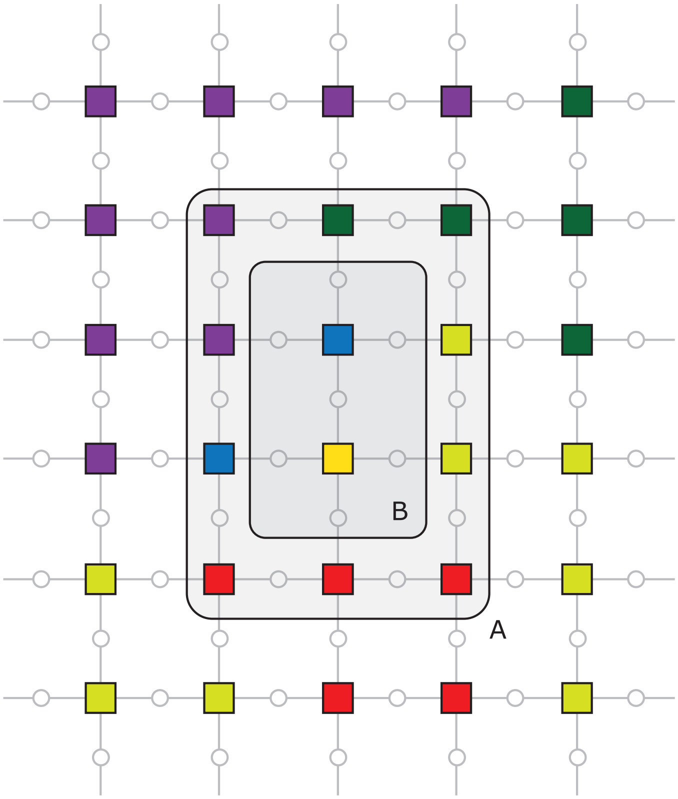


Figure F.5: An example of a rectangular region A with sides $l_x = 2$ and $l_y = 3$. Its bulk is marked by B. In the example shown above, C_A and C_B^{ex} have 6 and 1 elements respectively.

Proof. Let A be a $l_x \times l_y$ rectangular region. By an abuse of notation, we use A to denote the set of qubits which reside inside region A as well. We prove this claim by direct calculation of the entanglement. For stabilizers states we have

$$S_A = n_A - \dim G_A, \tag{F.9}$$

where n_A is the number of qubits in A and G_A is the subgroup of stabilizers which act trivially on the qubits in A^c. It is easy to see that G_A can be generated by Z plaquette stabilizers, $Z_{i,j}$ string stabilizers and $\overline{\mathcal{X}}_j$ stabilizers (see Proposition F.1 for their definition) which are themselves contained in A. Let $n_\square$, n_S and n_+ denote the number of independent Z plaquette stabilizer, $Z_{i,j}$

string stabilizer and $\overline{\mathcal{X}}_j$ stabilizers which are contained entirely within A, respectively. Then we have,

$$S_A = n_A - n_\square - n_+ - n_S \tag{F.10}$$

Now we compute each quantity separately. finding n_A and $n_\square$ is quite easy,

$$n_A = 2l_x l_y + l_x + l_y, \tag{F.11}$$

$$n_\square = l_x l_y. \tag{F.12}$$

Also we have $n_+ = |C_B^{\text{ex}}|$, because any $\overline{\mathcal{X}}_j$ operator which has a star operator outside B has nontrivial support in A^c. For a given color c, define $m_{c,A}$ to be the number of squares in A with color c. Then we can write n_S as,

$$n_S = \sum_{c \in C_A} (m_{c,A} - 1) = \sum_{c \in C_A} m_{c,A} - |C_A|. \tag{F.13}$$

Now, the $\sum_{c \in C_A} m_{c,A}$ sum is just the total number of squares inside A, which is $(l_x + 1)(l_y + 1)$. Thus we have

$$n_S = l_x l_y + l_x + l_y + 1 - |C_A|. \tag{F.14}$$

Putting everything together we get,

$$S_A = |C_A| - |C_B^{\text{ex}}| - 1 = |C_A \setminus C_B^{\text{ex}}| - 1, \tag{F.15}$$

where in the last part we used the fact that $C_B^{\text{ex}} \subseteq C_A$. $\square$

Appendix G: Classical and quantum expander codes

Classical linear codes - A classical linear code $\mathcal{C}$ over n bits can be defined as the kernel of a $m \times n$ parity check matrix $H \in \mathbb{F}_2^{m\times n}$, i.e. $c \in \mathbb{F}_2^n$ is a code word in $\mathcal{C}$ iff $Hc = 0$. A classical code associated with the parity check matrix H encodes $k = \dim(\ker(H)) = n - \mathrm{rank}(H)$ logical bits and has distance d where d is the minimum Hamming distance between pairs of different code words in $\mathcal{C}$. For future convenience, we set $d = \infty$ for codes with $k = 0$. The standard notation $\mathcal{C} = [n, k, d]$ is used to compactly specify the code properties. Note that $k \geq n - m$ since $\mathrm{rank}(H) \leq m$. Furthermore, different parity check matrices can give rise to the same classical code $\mathcal{C}$.

A classical code can also be specified by a Tanner graph. For a given classical linear code and its parity check matrix $H \in \mathbb{F}_2^{m\times n}$, its Tanner graph is a bipartite graph with n bit vertices on the left and m check vertices on the right. A check vertex i is connected to a bit vertex j iff $H_{i,j} = 1$. Since the parity check matrix of a code is not uniquely defined, so neither is its Tanner graph. As an example, the Tanner graph of the $[7, 4, 3]$ Hamming code with the parity check matrix of,

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{G.1}$$

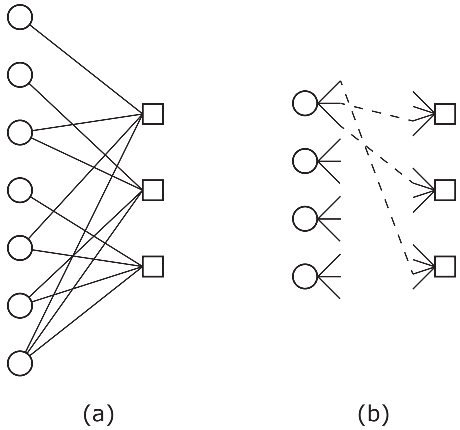


Figure G.1: (a) The Tanner graph associated with the 7-bit Hamming code, given by the parity check matrix in Eq.(G.1). (b) Random construction of biregular graphs for $(\Delta_L, \Delta_R) = (3, 4)$.

is shown in Fig.G.1a. Conversely, given a bipartite graph, one can define a parity check matrix H accordingly and associate a classical code $\mathcal{C}$ to it.

For a given classical code $\mathcal{C} = [n, k, d]$ and its parity check matrix $H \in \mathbb{F}_2^{m\times n}$, the transpose code of $\mathcal{C}$, denoted by $\mathcal{C}^T = [m, \tilde{k}, \tilde{d}]$ is defined as the classical code associated to the parity check matrix H^T which is the matrix transpose of H. Note that $\mathcal{C}^T$ depends on the choice of parity check matrix for $\mathcal{C}$ and is not uniquely defined by $\mathcal{C}$. Since $\mathrm{rank}(H) = \mathrm{rank}(H^T)$, it follows that $\tilde{k} = k - (m - n)$. The Tanner graph for $\mathcal{C}^T$, is the same as $\mathcal{C}$ with bit and check vertices interchanged.

Classical Expander Code - Consider a bipartite graph $G = (V, C, E)$ with V and C denoting the set of vertices on left and right respectively and E denoting the set of edges between V and C. The degree of a vertex is the number of edges connected to it. A bipartite graph is said to be biregular if all vertices on each part have the same degree. Let G be a biregular graph with left and right degrees Δ_L and Δ_R respectively, i.e. all vertices in V have degree d_L and all vertices in C have degree d_R. We say G is $(\alpha, \delta)-$left expanding if for any subset of left vertices $S \subset V$ for which $|S| \leq \alpha|V|$, we have $|\Gamma(S)| \geq (1 - \delta)\Delta_L|S|$ where $\Gamma(S)$ is the set of neighbors of S on the graph G. The classical code whose Tanner graph is (α, δ)-left expanding for some positive α and $\delta < 1/2$, is an instance of classical expander code. The distance of such a code is lower bounded by αn [233].

A biregular graph G with good expansion may be obtained via random construction. Imagine we want to construct a biregular graph G with n vertices of degree Δ_L on the left and $m = \frac{\Delta_L}{\Delta_R} n$ vertices of degree Δ_R on the right. We start by putting n vertices on the left with Δ_L stubs connected to each and putting m vertices on the right, each with Δ_R stubs connected to it. Note that there are $N = n\Delta_L = m\Delta_R$ stubs on each side. Now, we choose a random permutation π of numbers $1, \cdots, N$ uniformly at random and for each i, we connect the i'th stub on left to the π_i'th stub on the right (see Fig.G.1b). It can be shown that for any $\delta > 1/\Delta_L$, there exists a strictly positive α such that the resulting graph is (α, δ)-expanding with a probability that approaches 1 for large enough ns [233].

Hypergraph product code - The hyper graph product code is a quantum code built out of two classical codes. Let $\mathcal{C}_1 = [n_1, k_1, d_1]$ and $\mathcal{C}_2 = [n_2, k_2, d_2]$ denote two classical codes corresponding to two given parity check matrices $H_1 \in \mathbb{F}_2^{m_1 \times n_1}$ and $H_2 \in \mathbb{F}_2^{m_2 \times n_2}$ respectively. The hypergraph product code of $\mathcal{C}_1$ and $\mathcal{C}_2$ is a CSS stabilizer code $\mathcal{Q}$ defined on $n_1 n_2 + m_1 m_2$ qubits. The X and Z stabilizers of the quantum code $\mathcal{Q}$ is given by the parity matrices H_X and H_Z which are defined as

$$H_X = \left(I_{n_1} \otimes H_2 \mid H_1^T \otimes I_{m_2}\right) \tag{G.2}$$

$$H_Z = \left(H_1 \otimes I_{n_2} \mid I_{m_1} \otimes H_2^T\right), \tag{G.3}$$

where I_n is the n-dimensional identity matrix and $(A \mid B)$ is the matrix which results from the horizontal concatenation of A and B matrices. It can be shown that $\mathcal{Q}$ encodes $k_{\mathcal{Q}} = k_1 k_2 + \tilde{k}_1 \tilde{k}_1$ logical qubits and its distance is lower bounded by $\min(d_1, d_2, d_1^T, d_2^T)$, or in the standard notation

$$\mathcal{Q} = [\![n_1 n_2 + m_1 m_2, k_1 k_2 + k_1^T k_2^T, \min(d_1, d_2, d_1^T, d_2^T)]\!].$$

Quantum expander code - Quantum expander code is the hypergraph product code of a classical expander code with itself [220].

Bibliography

[1] A Yu Kitaev. Quantum error correction with imperfect gates. In *Quantum communication, computing, and measurement*, pages 181–188. Springer, 1997.

[2] A.Yu. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2 – 30, 2003.

[3] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. Topological quantum memory. *Journal of Mathematical Physics*, 43(9):4452–4505, 2002.

[4] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A*, 86:032324, Sep 2012.

[5] Emanuel Knill and Raymond Laflamme. Theory of quantum error-correcting codes. *Physical Review A*, 55(2):900, 1997.

[6] Yang Qiu and Zhenghan Wang. Ground subspaces of topological phases of matter as error correcting codes. *Annals of Physics*, 422:168318, 2020.

[7] Sergey Bravyi, Matthew B Hastings, and Spyridon Michalakis. Topological quantum order: stability under local perturbations. *Journal of mathematical physics*, 51(9):093512, 2010.

[8] Sergey Bravyi and Matthew B Hastings. A short proof of stability of topological order under local perturbations. *Communications in mathematical physics*, 307(3):609–627, 2011.

[9] Spyridon Michalakis and Justyna P Zwolak. Stability of frustration-free hamiltonians. *Communications in Mathematical Physics*, 322(2):277–302, 2013.

[10] Bruno Nachtergaele, Robert Sims, and Amanda Young. Quasi-locality bounds for quantum lattice systems. part ii. perturbations of frustration-free spin models with gapped ground states. In *Annales Henri Poincaré*, volume 23, pages 393–511. Springer, 2022.

[11] S.B. Bravyi and A. Yu. Kitaev. Quantum codes on a lattice with boundary. 1998.

[12] Michael H. Freedman and David A. Meyer. Projective plane and planar quantum codes. *Foundations of Computational Mathematics*, 1(3):325–332, 2001.

[13] H. Bombin and M. A. Martin-Delgado. Topological quantum distillation. *Phys. Rev. Lett.*, 97:180501, Oct 2006.

[14] Hector Bombin and Miguel-Angel Martin-Delgado. Topological computation without braiding. *Physical review letters*, 98(16):160502, 2007.

[15] H. Bombin. Topological order with a twist: Ising anyons from an abelian model. *Phys. Rev. Lett.*, 105:030403, Jul 2010.

[16] Alexei Kitaev and Liang Kong. Models for gapped boundaries and domain walls. *Communications in Mathematical Physics*, 313(2):351–373, 2012.

[17] Markus S Kesselring, Fernando Pastawski, Jens Eisert, and Benjamin J Brown. The boundaries and twist defects of the color code and their applications to topological quantum computation. *Quantum*, 2:101, 2018.

[18] Andrew C Doherty and Stephen D Bartlett. Identifying phases of quantum many-body systems that are universal for quantum computation. *Physical review letters*, 103(2):020506, 2009.

[19] Bei Zeng and Duan-Lu Zhou. Topological and error-correcting properties for symmetry-protected topological order. *EPL (Europhysics Letters)*, 113(5):56001, 2016.

[20] Robert Raussendorf, Cihan Okay, Dong-Sheng Wang, David T Stephen, and Hendrik Poulsen Nautrup. Computationally universal phase of quantum matter. *Physical review letters*, 122(9):090501, 2019.

[21] H Bombin and MA Martin-Delgado. Quantum measurements and gates by code deformation. *Journal of Physics A: Mathematical and Theoretical*, 42(9):095302, 2009.

[22] Hector Bombin. Clifford gates by code deformation. *New Journal of Physics*, 13(4):043005, 2011.

[23] Clare Horsman, Austin G Fowler, Simon Devitt, and Rodney Van Meter. Surface code quantum computing by lattice surgery. *New Journal of Physics*, 14(12):123011, 2012.

[24] Andrew J. Landahl and Ciaran Ryan-Anderson. Quantum computing by color-code lattice surgery. 2014.

[25] Daniel Litinski and Felix von Oppen. Lattice surgery with a twist: Simplifying clifford gates of surface codes. 2017.

[26] Christophe Vuillot, Lingling Lao, Ben Criger, Carmen García Almudéver, Koen Bertels, and Barbara M Terhal. Code deformation and lattice surgery are gauge fixing. *New Journal of Physics*, 21(3):033028, 2019.

[27] Nikolas P Breuckmann, Christophe Vuillot, Earl Campbell, Anirudh Krishna, and Barbara M Terhal. Hyperbolic and semi-hyperbolic surface codes for quantum storage. *Quantum Science and Technology*, 2(3):035007, 2017.

[28] Benjamin J. Brown, Katharina Laubscher, Markus S. Kesselring, and James R. Wootton. Poking holes and cutting corners to achieve clifford gates with the surface code. *Phys. Rev. X*, 7:021029, May 2017.

[29] Michael Freedman, Alexei Kitaev, Michael Larsen, and Zhenghan Wang. Topological quantum computation. *Bulletin of the American Mathematical Society*, 40(1):31–38, 2003.

[30] Chetan Nayak, Steven H. Simon, Ady Stern, Michael Freedman, and Sankar Das Sarma. Non-abelian anyons and topological quantum computation. *Rev. Mod. Phys.*, 80:1083, 2008.

[31] Helmut G Katzgraber, H Bombin, and MA Martin-Delgado. Error threshold for color codes and random three-body ising models. *Physical review letters*, 103(9):090501, 2009.

[32] Christopher T Chubb and Steven T Flammia. Statistical mechanical models for quantum codes with correlated noise. *Annales de l'Institut Henri Poincaré D*, 8(2):269–321, 2021.

[33] Robert Alicki, Michal Horodecki, Pawel Horodecki, and Ryszard Horodecki. On thermal stability of topological qubit in kitaev's 4d model. *Open Systems & Information Dynamics*, 17(01):1–20, 2010.

[34] Hector Bombin, Ravindra W Chhajlany, Michał Horodecki, and Miguel-Angel Martin-Delgado. Self-correcting quantum computers. *New Journal of Physics*, 15(5):055023, 2013.

[35] Ali Lavasani and Maissam Barkeshli. Low overhead clifford gates from joint measurements in surface, color, and hyperbolic codes. *Physical Review A*, 98(5):052319, 2018.

[36] Ali Lavasani, Guanyu Zhu, and Maissam Barkeshli. Universal logical gates with constant overhead: instantaneous dehn twists for hyperbolic quantum codes. *arXiv preprint arXiv:1901.11029*, 2019.

[37] Guanyu Zhu, Ali Lavasani, and Maissam Barkeshli. Universal logical gates on topologically encoded qubits via constant-depth unitary circuits. *Physical Review Letters*, 125(5):050502, 2020.

[38] Guanyu Zhu, Ali Lavasani, and Maissam Barkeshli. Instantaneous braids and dehn twists in topologically ordered states. *Physical Review B*, 102(7):075105, 2020.

[39] Ali Lavasani, Yahya Alavirad, and Maissam Barkeshli. Topological order and criticality in (2+ 1) d monitored random quantum circuits. *Physical review letters*, 127(23):235701, 2021.

[40] Ali Lavasani, Yahya Alavirad, and Maissam Barkeshli. Measurement-induced topological entanglement transitions in symmetric random quantum circuits. *Nature Physics*, pages 1–6, 2021.

[41] Yahya Alavirad and Ali Lavasani. Irreducible multi-partite correlations as an order parameter for k-local nontrivial states. *arXiv preprint arXiv:2106.05269*, 2021.

[42] Barbara M. Terhal. Quantum error correction for quantum memories. *Rev. Mod. Phys.*, 87:307–346, Apr 2015.

[43] Earl T. Campbell, Barbara M. Terhal, and Christophe Vuillot. Roads towards fault-tolerant universal quantum computation. *Nature*, 549(7671):172–179, 09 2017.

[44] Austin G. Fowler. Analytic asymptotic performance of topological codes. *Phys. Rev. A*, 87:040301, Apr 2013.

[45] Zhenghan Wang. *Topological Quantum Computation*. American Mathematical Society, 2008.

[46] Sergey Bravyi, David Poulin, and Barbara Terhal. Tradeoffs for reliable quantum information storage in 2d systems. *Physical review letters*, 104(5):050503, 2010.

[47] Ashley M Stephens. Fault-tolerant thresholds for quantum error correction with the surface code. *Physical Review A*, 89(2):022321, 2014.

[48] Google Quantum AI. Exponential suppression of bit or phase errors with cyclic error correction. *Nature*, 595(7867):383, 2021.

[49] Youwei Zhao, Yangsen Ye, He-Liang Huang, Yiming Zhang, Dachao Wu, Huijie Guan, Qingling Zhu, Zuolin Wei, Tan He, Sirui Cao, et al. Realizing an error-correcting surface code with superconducting qubits. *arXiv preprint arXiv:2112.13505*, 2021.

[50] Sebastian Krinner, Nathan Lacroix, Ants Remm, Agustin Di Paolo, Elie Genois, Catherine Leroux, Christoph Hellings, Stefania Lazar, Francois Swiadek, Johannes Herrmann, et al. Realizing repeated quantum error correction in a distance-three surface code. *arXiv preprint arXiv:2112.03708*, 2021.

[51] JF Marques, BM Varbanov, MS Moreira, Hany Ali, Nandini Muthusubramanian, Christos Zachariadis, Francesco Battistel, Marc Beekman, Nadia Haider, Wouter Vlothuizen, et al. Logical-qubit operations in an error-detecting surface code. *Nature Physics*, 18(1):80–86, 2022.

[52] Alexander Erhard, Hendrik Poulsen Nautrup, Michael Meth, Lukas Postler, Roman Stricker, Martin Stadler, Vlad Negnevitsky, Martin Ringbauer, Philipp Schindler, Hans J Briegel, et al. Entangling logical qubits with lattice surgery. *Nature*, 589(7841):220–224, 2021.

[53] M. H. Freedman, D. A. Meyer, and F. Luo. Z2-systolic freedom and quantum codes. In *Mathematics of Quantum Computation*, pages 287–320. Chapman and Hall, London, U.K., 2002.

[54] N. P. Breuckmann and B. M. Terhal. Constructions and noise threshold of hyperbolic surface codes. *IEEE Transactions on Information Theory*, 62(6):3731–3744, June 2016.

[55] Aleksander Kubica, Beni Yoshida, and Fernando Pastawski. Unfolding the color code. *New Journal of Physics*, 17(8):083026, 2015.

[56] Ashley M Stephens. Efficient fault-tolerant decoding of topological color codes. *arXiv preprint arXiv:1402.3037*, 2014.

[57] M. B. Hastings and A. Geller. Reduced space-time and time costs using dislocation codes and arbitrary ancillas. 2014.

[58] Daniel Litinski. A game of surface codes: Large-scale quantum computing with lattice surgery. *Quantum*, 3:128, 2019.

[59] Theodore J. Yoder and Isaac H. Kim. The surface code with a twist. *Quantum*, 1:2, April 2017.

[60] Parsa Bonderson, Michael Freedman, and Chetan Nayak. Measurement-only topological quantum computation via anyonic interferometry. *Annals of Physics*, 324(4):787 – 826, 2009.

[61] Maissam Barkeshli and Michael Freedman. Modular transformations through sequences of topological charge projections. *Phys. Rev. B*, 94:165108, Oct 2016.

[62] Iris Cong, Meng Cheng, and Zhenghan Wang. Topological quantum computation with gapped boundaries. 2016.

[63] Daniel Litinski and Felix von Oppen. Braiding by majorana tracking and long-range cnot gates with color codes. 2017.

[64] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17(3):449–467, 1965.

[65] Jack Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of Research of the National Bureau of Standards B*, 69(125-130):55–56, 1965.

[66] Bettina Heim, Krysta M Svore, and Matthew B Hastings. Optimal circuit-level decoding for surface codes. *arXiv preprint arXiv:1609.06373*, 2016.

[67] P Baireuther, MD Caio, B Criger, CWJ Beenakker, and TE O'Brien. Neural network decoder for topological color codes with circuit level noise. *arXiv preprint arXiv:1804.02926*, 2018.

[68] Guillaume Duclos-Cianci and David Poulin. Fast decoders for topological quantum codes. *Physical review letters*, 104(5):050504, 2010.

[69] Guillaume Duclos-Cianci and David Poulin. Fault-tolerant renormalization group decoder for abelian topological codes. *arXiv preprint arXiv:1304.6100*, 2013.

[70] Austin G Fowler. Minimum weight perfect matching of fault-tolerant topological quantum error correction in average $o(1)$ parallel time. *arXiv preprint arXiv:1307.1740*, 2013.

[71] Chenyang Wang, Jim Harrington, and John Preskill. Confinement-higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory. *Annals of Physics*, 303(1):31–58, 2003.

[72] Austin G. Fowler. Proof of finite surface code threshold for matching. *Phys. Rev. Lett.*, 109:180502, Nov 2012.

[73] Fern HE Watson and Sean D Barrett. Logical error rate scaling of the toric code. *New Journal of Physics*, 16(9):093045, 2014.

[74] Austin G Fowler, Adam C Whiteside, Angus L McInnes, and Alimohammad Rabbani. Topological code autotune. *Physical Review X*, 2(4):041003, 2012.

[75] Robert Raussendorf, Jim Harrington, and Kovid Goyal. Topological fault-tolerance in cluster state quantum computation. *New Journal of Physics*, 9(6):199, 2007.

[76] Austin G Fowler, Adam C Whiteside, and Lloyd CL Hollenberg. Towards practical classical processing for the surface code. *Physical review letters*, 108(18):180501, 2012.

[77] Daniel Gottesman. Theory of fault-tolerant quantum computation. *Physical Review A*, 57(1):127, 1998.

[78] Austin G Fowler, Ashley M Stephens, and Peter Groszkowski. High-threshold universal quantum computation on the surface code. *Physical Review A*, 80(5):052312, 2009.

[79] R. Raussendorf, J. Harrington, and K. Goyal. A fault-tolerant one-way quantum computer. *Annals of Physics*, 321(9):2242 – 2270, 2006.

[80] Maissam Barkeshli and Xiao-Liang Qi. Topological nematic states and non-abelian lattice dislocations. *Phys. Rev. X*, 2:031013, Aug 2012.

[81] Maissam Barkeshli and Xiao-Gang Wen. $u(1) \times u(1) \rtimes z_2$ chern-simons theory and z_4 parafermion fractional quantum hall states. *Phys. Rev. B*, 81:045323, 2010. See Sec. V. A.

[82] Maissam Barkeshli, Chao-Ming Jian, and Xiao-Liang Qi. Twist defects and projective non-abelian braiding statistics. *Phys. Rev. B*, 87:045130, Jan 2013.

[83] Maissam Barkeshli, Chao-Ming Jian, and Xiao-Liang Qi. Theory of defects in abelian topological states. *Phys. Rev. B*, 88:235103, Dec 2013.

[84] Jeffrey C.Y. Teo, Abhishek Roy, and Xiao Chen. Unconventional fusion and braiding of topological defects in a lattice model. 2013.

[85] Maissam Barkeshli, Parsa Bonderson, Meng Cheng, and Zhenghan Wang. Symmetry, defects, and gauging of topological phases. 2014.

[86] David J. Clarke, Jason Alicea, and Kirill Shtengel. Exotic non-abelian anyons from conventional fractional quantum hall states. *Nature Comm.*, 4:1348, 2013.

[87] Meng Cheng. Superconducting proximity effect on the edge of fractional topological insulators. *Phys. Rev. B*, 86:195126, Nov 2012.

[88] Netanel H. Lindner, Erez Berg, Gil Refael, and Ady Stern. Fractionalizing majorana fermions: Non-abelian statistics on the edges of abelian quantum hall states. *Phys. Rev. X*, 2:041002, Oct 2012.

[89] Jason Alicea and Paul Fendley. Topological phases with parafermions: Theory and blueprints. *Annual Review of Condensed Matter Physics*, 7(1):119–139, 2016.

[90] Alexei Kitaev and Liang Kong. Models for gapped boundaries and domain walls. *Comm. Math. Phys.*, 313(2):351–373, 2012.

[91] Yi-Zhuang You and Xiao-Gang Wen. Projective non-abelian statistics of dislocation defects in a F_N rotor model. *Phys. Rev. B*, 86:161107, Oct 2012.

[92] Yi-Zhuang You, Chao-Ming Jian, and Xiao-Gang Wen. Synthetic non-abelian statistics by abelian anyon condensation. *Phys. Rev. B*, 87:045106, Jan 2013.

[93] Nicolas Delfosse, Pavithran Iyer, and David Poulin. Generalized surface codes and packing of logical qubits. 2016.

[94] Austin G. Fowler. Two-dimensional color-code quantum computation. *Phys. Rev. A*, 83:042310, Apr 2011.

[95] Michael Levin. Protected edge modes without symmetry. *Phys. Rev. X*, 3:021009, May 2013.

[96] Maissam Barkeshli, Chao-Ming Jian, and Xiao-Liang Qi. Classification of topological defects in abelian topological states. *Phys. Rev. B*, 88:241103(R), Dec 2013.

[97] Jacob C Bridgeman, Stephen D Bartlett, and Andrew C Doherty. Tensor networks with a twist: Anyon-permuting domain walls and defects in peps. *arXiv preprint arXiv:1708.08930*, 2017.

[98] Michael A Nielsen and Isaac L Chuang. Quantum computation and information theory, 2010.

[99] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[100] Barbara M Terhal. Quantum error correction for quantum memories. *Rev. Mod. Phys.*, 87(2):307–346, April 2015.

[101] Robert Koenig, Greg Kuperberg, and Ben W Reichardt. Quantum computation with Turaev-Viro codes. *Annals of Physics*, 325(12):2707–2749, December 2010.

[102] Michael H Freedman, David A Meyer, and Feng Luo. Z2-systolic freedom and quantum codes. *Mathematics of quantum computation, Chapman & Hall/CRC*, pages 287–320, 2002.

[103] Larry Guth and Alexander Lubotzky. Quantum error correcting codes and 4-dimensional arithmetic hyperbolic manifolds. *Journal of Mathematical Physics*, 55(8):082202, August 2014.

[104] J.-P Tillich and Gilles Zémor. Quantum ldpc codes with positive rate and minimum distance proportional to the square root of the blocklength. *Information Theory, IEEE Transactions on*, 60:1193–1202, 02 2014.

[105] Anthony Leverrier, Jean-Pierre Tillich, and Gilles Zémor. Quantum expander codes. pages 810–824, 2015.

[106] Nikolas P Breuckmann and Jens N Eberhardt. Balanced product quantum codes. *IEEE Transactions on Information Theory*, 67(10):6653–6674, 2021.

[107] Pavel Panteleev and Gleb Kalachev. Quantum ldpc codes with almost linear minimum distance. *IEEE Transactions on Information Theory*, 68(1):213–229, 2021.

[108] A. R. Calderbank and Peter W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54:1098–1105, Aug 1996.

[109] Sergey Bravyi and Matthew B. Hastings. Homological product codes. *Proc. of the 46th ACM Symposium on Theory of Computing (STOC 2014)*, pages 273–282, 2014.

[110] Nikolas P Breuckmann and Jens Niklas Eberhardt. Quantum low-density parity-check codes. *PRX Quantum*, 2(4):040101, 2021.

[111] Alexey A. Kovalev and Leonid P. Pryadko. Fault tolerance of quantum low-density parity check codes with sublinear distance scaling. *Phys. Rev. A*, 87:020304, Feb 2013.

[112] Daniel Gottesman. Fault-Tolerant Quantum Computation with Constant Overhead. *Quantum Information & Computation*, 14(15-16):1338–1371, November 2014.

[113] Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal clifford gates and noisy ancillas. *Phys. Rev. A*, 71:022316, Feb 2005.

[114] Adam Paetznick and Ben W Reichardt. Universal Fault-Tolerant Quantum Computation with Only Transversal Gates and Error Correction. *Phys. Rev. Lett.*, 111(9):090505–5, August 2013.

[115] M H Freedman, M Larsen, and Z H Wang. A modular functor which is universal for quantum computation. *Communications in Mathematical Physics*, 227(3):605–622, June 2002.

[116] Michael A. Levin and Xiao-Gang Wen. String-net condensation: a physical mechanism for topological phases. *Phys. Rev. B*, 71:045110, Jan 2005.

[117] Zhenghan Wang. *Topological Quantum Computation*. American Mathematics Society, 2010.

[118] V.G. Turaev and O.Y. Viro. State sum invariants of 3-manifolds and quantum 6j-symbols. *Topology*, 31:865–902, 1992.

[119] John W. Barrett and Bruce W. Westbury. Invariants of piecewise-linear 3-manifolds. *Trans. Amer. Math. Soc.*, 348:3997–4022, 1996.

[120] N M Linke, D Maslov, M Roetteler, S Debnath, C Figgatt, K A Landsman, K Wright, and C Monroe. Experimental Comparison of Two Quantum Computing Architectures. *PNAS 13, 3305–3310*, February 2017.

[121] Bjoern Lekitsch, Sebastian Weidt, Austin G. Fowler, Klaus Mølmer, Simon J. Devitt, Christof Wunderlich, and Winfried K. Hensinger. Blueprint for a microwave trapped ion quantum computer. *Science Advances*, 3(2), 2017.

[122] P. Campagne-Ibarcq, E. Zalys-Geller, A. Narla, S. Shankar, P. Reinhold, L. Burkhart, C. Axline, W. Pfaff, L. Frunzio, R. J. Schoelkopf, and M. H. Devoret. Deterministic remote entanglement of superconducting circuits through microwave two-photon transitions. *Phys. Rev. Lett.*, 120:200501, May 2018.

[123] Philipp Kurpiers, Paul Magnard, Theo Walter, Baptiste Royer, Marek Pechal, Johannes Heinsoo, Yves Salathé, Abdulkadir Akin, Simon Storz, J-C Besse, et al. Deterministic quantum state transfer and remote entanglement using microwave photons. *Nature*, 558(7709):264, 2018.

[124] Christopher J. Axline, Luke D. Burkhart, Wolfgang Pfaff, Mengzhen Zhang, Kevin Chou, Philippe Campagne-Ibarcq, Philip Reinhold, Luigi Frunzio, S. M. Girvin, Liang Jiang, M. H. Devoret, and R. J. Schoelkopf. On-demand quantum state transfer and entanglement between remote microwave cavity memories. *Nature Physics*, 14(7):705–710, apr 2018.

[125] K S Chou, J Z Blumoff, C S Wang, Reinhold, P. C., C J Axline, Y Y Gao, L Frunzio, M H Devoret, Liang Jiang, and R J Schoelkopf. Deterministic teleportation of a quantum gate between two logical qubits. January 2018.

[126] M Saffman, T G Walker, and K Mølmer. Quantum information with Rydberg atoms. *Rev. Mod. Phys.*, 82(3):2313–2363, August 2010.

[127] Daniel Comparat and Pierre Pillet. Dipole blockade in a cold Rydberg atomic sample [Invited]. *Journal of the Optical Society of America B*, 27(6):A208–A232, June 2010.

[128] K M Maller, M T Lichtman, T Xia, Y Sun, M J Piotrowicz, A W Carr, L Isenhower, and M Saffman. Rydberg-blockade controlled-not gate and entanglement in a two-dimensional array of neutral-atom qubits. *Physical Review A*, 92(2):022336–12, August 2015.

[129] Hannes Pichler, Guanyu Zhu, Alireza Seif, Peter Zoller, and Mohammad Hafezi. Measurement Protocol for the Entanglement Spectrum of Cold Atoms. *Physical Review X*, 6(4):041033–12, November 2016.

[130] David A Herrera-Martí, Austin G Fowler, David Jennings, and Terry Rudolph. Photonic implementation for the topological cluster-state quantum computer. *Physical Review A*, 82(3):032332–6, September 2010.

[131] Alicia J. Kollár, Mattias Fitzpatrick, and Andrew A. Houck. Hyperbolic lattices in circuit quantum electrodynamics. *Nature*, 571(7763):45–50, 2019.

[132] Igor Boettcher, Przemyslaw Bienias, Ron Belyansky, Alicia J Kollár, and Alexey V Gorshkov. Quantum simulation of hyperbolic space with circuit quantum electrodynamics: From graphs to geometry. *Physical Review A*, 102(3):032208, 2020.

[133] Guanyu Zhu, Mohammad Hafezi, and Maissam Barkeshli. Quantum Origami: Transversal Gates for Quantum Computation and Measurement of Topological Order. *arXiv:1711.05752*, November 2017.

[134] Michael E Beverland, Oliver Buerschaper, Robert Koenig, Fernando Pastawski, John Preskill, and Sumit Sijher. Protected gates for topological quantum field theories. *Journal of Mathematical Physics*, 57(2):022201–40, February 2016.

[135] Nikolas P Breuckmann, Christophe Vuillot, Earl Campbell, Anirudh Krishna, and Barbara M Terhal. Hyperbolic and semi-hyperbolic surface codes for quantum storage. *Quantum Science and Technology*, 2(3):035007–21, August 2017.

[136] N E Bonesteel and D P DiVincenzo. Quantum circuits for measuring Levin-Wen operators. *Physical Review B*, 86(16):165113, 2012.

[137] Weibo Feng. Non-abelian quantum error correction. *Ph.D. Thesis, The Florida State University*, 2015.

[138] Simon Burton, Courtney G Brell, and Steven T Flammia. Classical simulation of quantum error correction in a Fibonacci anyon code. *Physical Review A*, 95(2):580–10, February 2017.

[139] Guillaume Dauphinais, Laura Ortiz, Santiago Varona, and Miguel Angel Martin-Delgado. Quantum error correction with the semion code. *New Journal of Physics*, 21(5):053035, 2019.

[140] Julio Carlos Magdalena de la Fuente, Nicolas Tarantino, and Jens Eisert. Non-pauli topological stabilizer codes from twisted quantum doubles. *Quantum*, 5:398, 2021.

[141] Alexis Schotte, Guanyu Zhu, Lander Burgelman, and Frank Verstraete. Quantum error correction thresholds for the universal fibonacci turaev-viro code. *Physical Review X*, 12(2):021012, 2022.

[142] Robert Fricke and Felix Klein. *Vorlesungen uber die Theorie der automorphen Funktionen*. Johnson Reprint, 1897.

[143] Linda Keen. Canonical polygons for finitely generated fuchsian groups. *Acta Mathematica*, 115(1):1–16, 1966.

[144] James Munkres. Obstructions to the smoothing of piecewise-differentiable homeomorphisms. *Annals of Mathematics*, pages 521–554, 1960.

[145] Max Dehn. *Papers on group theory and topology*. Springer Science & Business Media, 2012.

[146] Jakob Nielsen. Untersuchungen zur topologie der geschlossenen zweiseitigen flächen. *Acta Mathematica*, 50(1):189–358, 1927.

[147] Benson Farb and Dan Margalit. *A primer on mapping class groups (pms-49)*. Princeton University Press, 2011.

[148] Sergey Bravyi and Robert König. Classification of Topologically Protected Gates for Local Stabilizer Codes. *Phys. Rev. Lett.*, 110(17):170503–5, April 2013.

[149] Guillaume Dauphinais and David Poulin. Fault-Tolerant Quantum Error Correction for non-Abelian Anyons. *Communications in Mathematical Physics*, 355(2):519–560, July 2017.

[150] Brian Skinner, Jonathan Ruhman, and Adam Nahum. Measurement-induced phase transitions in the dynamics of entanglement. *Physical Review X*, 9(3):031009, 2019.

[151] Yaodong Li, Xiao Chen, and Matthew PA Fisher. Quantum zeno effect and the many-body entanglement transition. *Physical Review B*, 98(20):205136, 2018.

[152] Amos Chan, Rahul M. Nandkishore, Michael Pretko, and Graeme Smith. Unitary-projective entanglement dynamics. *Phys. Rev. B*, 99:224307, Jun 2019.

[153] Yaodong Li, Xiao Chen, and Matthew PA Fisher. Measurement-driven entanglement transition in hybrid quantum circuits. *Physical Review B*, 100(13):134306, 2019.

[154] Michael J. Gullans and David A. Huse. Dynamical purification phase transition induced by quantum measurements. 2019.

[155] Michael J. Gullans and David A. Huse. Scalable probes of measurement-induced criticality. *Phys. Rev. Lett.*, 125:070606, Aug 2020.

[156] Romain Vasseur, Andrew C. Potter, Yi-Zhuang You, and Andreas W. W. Ludwig. Entanglement transitions from holographic random tensor networks. *Phys. Rev. B*, 100:134203, Oct 2019.

[157] Soonwon Choi, Yimu Bao, Xiao-Liang Qi, and Ehud Altman. Quantum error correction in scrambling dynamics and measurement-induced phase transition. *Phys. Rev. Lett.*, 125:030505, Jul 2020.

[158] M. Szyniszewski, A. Romito, and H. Schomerus. Entanglement transition from variable-strength weak measurements. *Phys. Rev. B*, 100:064204, Aug 2019.

[159] Qicheng Tang and W. Zhu. Measurement-induced phase transition: A case study in the nonintegrable model by density-matrix renormalization group calculations. *Phys. Rev. Research*, 2:013022, Jan 2020.

[160] Chao-Ming Jian, Yi-Zhuang You, Romain Vasseur, and Andreas W. W. Ludwig. Measurement-induced criticality in random quantum circuits. *Phys. Rev. B*, 101:104302, Mar 2020.

[161] Xiangyu Cao, Antoine Tilloy, and Andrea De Luca. Entanglement in a fermion chain under continuous monitoring. *SciPost Phys.*, 7:24, 2019.

[162] Javier Lopez-Piqueres, Brayden Ware, and Romain Vasseur. Mean-field entanglement transitions in random tree tensor networks. *Phys. Rev. B*, 102:064202, Aug 2020.

[163] Yimu Bao, Soonwon Choi, and Ehud Altman. Theory of the phase transition in random unitary circuits with measurements. *Phys. Rev. B*, 101:104301, Mar 2020.

[164] Lorenzo Piroli, Christoph Sünderhauf, and Xiao-Liang Qi. A random unitary circuit model for black hole evaporation. *Journal of High Energy Physics*, 2020(4):63, Apr 2020.

[165] Aidan Zabalo, Michael J. Gullans, Justin H. Wilson, Sarang Gopalakrishnan, David A. Huse, and J. H. Pixley. Critical properties of the measurement-induced transition in random quantum circuits. *Phys. Rev. B*, 101:060301, Feb 2020.

[166] Davide Rossini and Ettore Vicari. Measurement-induced dynamics of many-body systems at quantum criticality. *Phys. Rev. B*, 102:035119, Jul 2020.

[167] Ruihua Fan, Sagar Vijay, Ashvin Vishwanath, and Yi-Zhuang You. Self-organized error correction in random unitary circuits with measurement. 2020.

[168] Adam Nahum and Brian Skinner. Entanglement and dynamics of diffusion-annihilation processes with majorana defects. *Phys. Rev. Research*, 2:023288, Jun 2020.

[169] Yaodong Li, Xiao Chen, and Matthew PA Fisher. Measurement-driven entanglement transition in hybrid quantum circuits. *Physical Review B*, 100(13):134306, 2019.

[170] Michael J Gullans and David A Huse. Dynamical purification phase transitions induced by quantum measurements. *arXiv preprint arXiv:1905.05195*, 2019.

[171] Michael J. Gullans and David A. Huse. Scalable probes of measurement-induced criticality. *Phys. Rev. Lett.*, 125:070606, Aug 2020.

[172] Jason Iaconis, Andrew Lucas, and Xiao Chen. Measurement-induced phase transitions in quantum automaton circuits, 2020.

[173] Yohei Fuji and Yuto Ashida. Measurement-induced quantum criticality under continuous monitoring. *Phys. Rev. B*, 102:054302, Aug 2020.

[174] Nicolai Lang and Hans Peter Büchler. Entanglement transition in the projective transverse field ising model. *Phys. Rev. B*, 102:094204, Sep 2020.

[175] Shengqi Sang and Timothy H Hsieh. Measurement protected quantum phases. *arXiv preprint arXiv:2004.09509*, 2020.

[176] Xiao Chen, Yaodong Li, Matthew P. A. Fisher, and Andrew Lucas. Emergent conformal symmetry in nonunitary random dynamics of free fermions. *Phys. Rev. Research*, 2:033017, Jul 2020.

[177] Adam Nahum, Sthitadhi Roy, Brian Skinner, and Jonathan Ruhman. Measurement and entanglement phase transitions in all-to-all quantum circuits, on quantum trees, and in landau-ginsburg theory, 2020.

[178] Oliver Lunt and Arijeet Pal. Measurement-induced entanglement transitions in many-body localized systems. *Phys. Rev. Research*, 2:043072, Oct 2020.

[179] Marcin Szyniszewski, Alessandro Romito, and Henning Schomerus. Universality of entanglement transitions from stroboscopic to continuous measurements, 2020.

[180] Michael J. Gullans, Stefan Krastanov, David A. Huse, Liang Jiang, and Steven T. Flammia. Quantum coding with low-depth random circuits, 2020.

[181] Adam Nahum and Brian Skinner. Entanglement and dynamics of diffusion-annihilation processes with majorana defects. *Phys. Rev. Research*, 2:023288, Jun 2020.

[182] Ori Alberton, Michael Buchhold, and Sebastian Diehl. Trajectory dependent entanglement transition in a free fermion chain – from extended criticality to area law, 2020.

[183] Xhek Turkeshi, Rosario Fazio, and Marcello Dalmonte. Measurement-induced criticality in $(2 + 1)$-dimensional hybrid quantum circuits. *Phys. Rev. B*, 102:014315, Jul 2020.

[184] Lukasz Fidkowski, Jeongwan Haah, and Matthew B. Hastings. How dynamical quantum memories forget, 2020.

[185] Sagar Vijay. Measurement-driven phase transition within a volume-law entangled phase, 2020.

[186] Ruihua Fan, Sagar Vijay, Ashvin Vishwanath, and Yi-Zhuang You. Self-organized error correction in random unitary circuits with measurement, 2020.

[187] Matteo Ippoliti, Michael J. Gullans, Sarang Gopalakrishnan, David A. Huse, and Vedika Khemani. Entanglement phase transitions in measurement-only dynamics, 2020.

[188] Yaodong Li and Matthew P. A. Fisher. Statistical mechanics of quantum error-correcting codes, 2020.

[189] Mathias Van Regemortel, Ze-Pei Cian, Alireza Seif, Hossein Dehghani, and Mohammad Hafezi. Entanglement entropy scaling transition under competing monitoring protocols. *arXiv preprint arXiv:2008.08619*, 2020.

[190] Oles Shtanko, Yaroslav A. Kharkov, Luis Pedro Garcí-Pintos, and Alexey V. Gorshkov. Classical models of entanglement in monitored random circuits, 2020.

[191] Matthew B. Hastings. Topological order at nonzero temperature. *Phys. Rev. Lett.*, 107:210501, Nov 2011.

[192] Xie Chen, Zheng-Cheng Gu, Zheng-Xin Liu, and Xiao-Gang Wen. Symmetry protected topological orders and the group cohomology of their symmetry group. *Physical Review B*, 87(15):155114, 2013.

[193] Robert Raussendorf and Hans J Briegel. A one-way quantum computer. *Physical Review Letters*, 86(22):5188, 2001.

[194] Bei Zeng, Xie Chen, Duan-Lu Zhou, and Xiao-Gang Wen. *Quantum information meets quantum matter*. Springer, 2019.

[195] Wonmin Son, Luigi Amico, Rosario Fazio, Alioscia Hamma, Saverio Pascazio, and Vlatko Vedral. Quantum phase transition between cluster and antiferromagnetic states. *EPL (Europhysics Letters)*, 95(5):50001, 2011.

[196] Luiz H Santos. Rokhsar-kivelson models of bosonic symmetry-protected topological states. *Physical Review B*, 91(15):155150, 2015.

[197] Lokman Tsui, Yen-Ta Huang, Hong-Chen Jiang, and Dung-Hai Lee. The phase transitions between zn× zn bosonic topological phases in 1+ 1d, and a constraint on the central charge for the critical points between bosonic symmetry protected topological phases. *Nuclear Physics B*, 919:470–503, 2017.

[198] Bei Zeng and Xiao-Gang Wen. Gapped quantum liquids and topological order, stochastic local transformations and emergence of unitarity. *Physical Review B*, 91(12):125121, 2015.

[199] P. Fromholz, G. Magnifico, V. Vitale, T. Mendes-Santos, and M. Dalmonte. Entanglement topological invariants for one-dimensional topological superconductors. *Phys. Rev. B*, 101:085136, Feb 2020.

[200] David Fattal, Toby S Cubitt, Yoshihisa Yamamoto, Sergey Bravyi, and Isaac L Chuang. Entanglement in the stabilizer formalism. *arXiv preprint quant-ph/0406168*, 2004.

[201] Pasquale Calabrese and John Cardy. Entanglement entropy and conformal field theory. *Journal of Physics A: Mathematical and Theoretical*, 42(50):504005, 2009.

[202] Yaodong Li, Xiao Chen, Andreas W. W. Ludwig, and Matthew P. A. Fisher. Conformal invariance and quantum non-locality in hybrid quantum circuits. 2020.

[203] Alexei Kitaev and John Preskill. Topological entanglement entropy. *Phys. Rev. Lett.*, 96:110404, Mar 2006.

[204] Michael Levin and Xiao-Gang Wen. Detecting topological order in a ground state wave function. *Phys. Rev. Lett.*, 96:110405, Mar 2006.

[205] Cenke Xu and J. E. Moore. Strong-weak coupling self-duality in the two-dimensional quantum phase transition of $p + ip$ superconducting arrays. *Phys. Rev. Lett.*, 93:047003, Jul 2004.

[206] Zohar Nussinov and Jeroen van den Brink. Compass models: Theory and physical motivations. *Rev. Mod. Phys.*, 87:1–59, Jan 2015.

[207] Junfeng Wang, Zongzheng Zhou, Wei Zhang, Timothy M. Garoni, and Youjin Deng. Bond and site percolation in three dimensions. *Phys. Rev. E*, 87:052107, May 2013.

[208] Hyejin Ju, Ann B. Kallin, Paul Fendley, Matthew B. Hastings, and Roger G. Melko. Entanglement scaling in two-dimensional gapless systems. *Phys. Rev. B*, 85:165121, Apr 2012.

[209] Jean-Marie Stephan, Hyejin Ju, Paul Fendley, and Roger G Melko. Entanglement in gapless resonating-valence-bond states. *New Journal of Physics*, 15(1):015004, 2013.

[210] Xiao Chen, Gil Young Cho, Thomas Faulkner, and Eduardo Fradkin. Scaling of entanglement in 2+ 1-dimensional scale-invariant field theories. *Journal of Statistical Mechanics: Theory and Experiment*, 2015(2):P02010, 2015.

[211] Dave Bacon. Operator quantum error-correcting subsystems for self-correcting quantum memories. *Phys. Rev. A*, 73:012340, Jan 2006.

[212] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493–R2496, Oct 1995.

[213] Xie Chen, Zheng-Cheng Gu, and Xiao-Gang Wen. Local unitary transformation, long-range quantum entanglement, wave function renormalization, and topological order. *Phys. Rev. B*, 82:155138, Oct 2010.

[214] Bei Zeng, Xie Chen, Duan-Lu Zhou, and Xiao-Gang Wen. *Quantum information meets quantum matter*. Springer, 2019.

[215] Dorit Aharonov, Itai Arad, and Thomas Vidick. Guest column: the quantum pcp conjecture. *Acm sigact news*, 44(2):47–79, 2013.

[216] Dorit Aharonov and Yonathan Touati. Quantum circuit depth lower bounds for homological codes, 2018.

[217] Daniel Gottesman. Fault-tolerant quantum computation with constant overhead, 2014.

[218] Nikolas P. Breuckmann and Jens Niklas Eberhardt. Ldpc quantum codes, 2021.

[219] Winton Brown and Omar Fawzi. Short random circuits define good quantum error correcting codes. In *2013 IEEE International Symposium on Information Theory*, pages 346–350. IEEE, 2013.

[220] Anthony Leverrier, Jean-Pierre Tillich, and Gilles Zémor. Quantum expander codes. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 810–824. IEEE, 2015.

[221] Fernando Pastawski, Beni Yoshida, Daniel Harlow, and John Preskill. Holographic quantum error-correcting codes: Toy models for the bulk/boundary correspondence. *Journal of High Energy Physics*, 2015(6):1–55, 2015.

[222] N. Linden, S. Popescu, and W. K. Wootters. Almost every pure state of three qubits is completely determined by its two-particle reduced density matrices. *Phys. Rev. Lett.*, 89:207901, Oct 2002.

[223] Kohtaro Kato, Fabian Furrer, and Mio Murao. Information-theoretical analysis of topological entanglement entropy and multipartite correlations. *Phys. Rev. A*, 93:022317, Feb 2016.

[224] D. L. Zhou. Irreducible multiparty correlations in quantum states without maximal rank. *Phys. Rev. Lett.*, 101:180505, Oct 2008.

[225] D. L. Zhou. Irreducible multiparty correlations can be created by local operations. *Phys. Rev. A*, 80:022113, Aug 2009.

[226] Sergey Bravyi, David Fattal, and Daniel Gottesman. Ghz extraction yield for multipartite stabilizer states. *Journal of Mathematical Physics*, 47(6):062106, 2006.

[227] Matthew B. Hastings. Topological order at nonzero temperature. *Phys. Rev. Lett.*, 107:210501, Nov 2011.

[228] Patrick Hayden, Debbie W Leung, and Andreas Winter. Aspects of generic entanglement. *Communications in mathematical physics*, 265(1):95–117, 2006.

[229] Graeme Smith and Debbie Leung. Typical entanglement of stabilizer states. *Phys. Rev. A*, 74:062314, Dec 2006.

[230] Patrick Hayden, Richard Jozsa, Denes Petz, and Andreas Winter. Structure of states which satisfy strong subadditivity of quantum entropy with equality. *Communications in mathematical physics*, 246(2):359–374, 2004.

[231] Omar Fawzi and Renato Renner. Quantum conditional mutual information and approximate markov chains. *Communications in Mathematical Physics*, 340(2):575–611, 2015.

[232] This can be numerically verified by considering a slightly different way of choosing A_i subsystems which converges to the thermodynamic limit much faster: instead of assigning iid labels to each qubit which results in $|A_i|/N = q + O(1/\sqrt{N})$, assign distinct random sets of $\lfloor qN \rfloor$ qubits to each set A_i.

[233] Tom Richardson and Ruediger Urbanke. *Modern coding theory*. Cambridge university press, 2008.

[234] Nicolas Delfosse and Gilles Zémor. Upper bounds on the rate of low density stabilizer codes for the quantum erasure channel. *arXiv preprint arXiv:1205.7036*, 2012.

[235] Alexander Jahn and Jens Eisert. Holographic tensor network models and quantum error correction: A topical review. *arXiv preprint arXiv:2102.02619*, 2021.

[236] Ramis Movassagh and Peter W Shor. Supercritical entanglement in local systems: Counterexample to the area law for quantum matter. *Proceedings of the National Academy of Sciences*, 113(47):13278–13282, 2016.

[237] Fernando G. S. L. Brandão, Elizabeth Crosson, M. Burak Şahinoğlu, and John Bowen. Quantum error correcting codes in eigenstates of translation-invariant spin chains. *Phys. Rev. Lett.*, 123:110502, Sep 2019.

[238] Ahmed Almheiri, Xi Dong, and Daniel Harlow. Bulk locality and quantum error correction in ads/cft. *Journal of High Energy Physics*, 2015(4):163, 2015.

[239] Adam Nahum, Jonathan Ruhman, Sagar Vijay, and Jeongwan Haah. Quantum entanglement growth under random unitary dynamics. *Physical Review X*, 7(3):031016, 2017.

[240] Daniel Gottesman. The Heisenberg representation of quantum computers. In *22nd International Colloquium on Group Theoretical Methods in Physics*, pages 32–43, 7 1998.

[241] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002. See Section 10.6.3.

[242] John Cardy. Conformal invariance and percolation. *arXiv preprint math-ph/0103018*, 2001.

[243] John Cardy. Linking numbers for self-avoiding loops and percolation: Application to the spin quantum hall transition. *Phys. Rev. Lett.*, 84:3507–3510, Apr 2000.