

Notes on Quantum Error Correction

Cheng-Yu Liu

316497z@gmail.com

Abstract

Quantum error correction (QEC) protects against both coherent and incoherent errors by encoding logical quantum information into a larger Hilbert space of physical qubits, where errors can be detected and corrected without disturbing the logical state. In this note, we provide short introductions to key concepts in QEC, including the stabilizer formalism, code concatenation, and subsystem codes, with illustrative examples. Finally, using Python and the Qiskit framework, we reproduce a simulation of the Bacon–Shor code on a trapped-ion chain as presented in [6], and discuss the results in relation to the error models analyzed in that work.

1

2 Contents

3	1 Introduction	2
4	2 Stabilizer Formalism	2
5	3 Homomorphic Logical Measurements (Notes on the Talk and Paper [5, 7])	4
6	3.1 Surface code	4
7	3.2 Quantum LDPC (Low-Density Parity-Check) code	4
8	3.3 Logical measurements of Shor and Steane type	5
9	3.4 Binary vector spaces	8
10	3.5 Algebraic Topology	9
11	3.5.1 Homology Groups	11
12	3.6 Homomorphic logical measurements	12
13	3.7 Homomorphic measurements on surface codes	13
14	3.8 Homomorphic gadgets for covering spaces	15
15	3.9 Fault tolerance	17
16	3.10 Joint measurement	17
17	4 Summary	17

18

19

20 1 Introduction

21 Quantum algorithms potentially speed up calculations exponentially but at the same time require thousands
22 of gate operations. High-fidelity gates can be realized in experiments [13], but accumulation of errors can
23 be drastic in even larger systems as required in many applications [2]. Assuming a simple model where each
24 gate has independent stochastic errors with fidelity f , the probability that a circuit of m gates has no errors
25 is f^m , which can be around 50% with 140 consecutive gates with fidelity $f = 99.5\%$. The probability of
26 error-free execution drops rapidly as the circuit depth increases. For practical algorithms requiring thousands
27 or millions of operations, such raw physical error rates are clearly insufficient. This motivates the use of
28 quantum error correction (QEC), in which logical qubits are redundantly encoded into multiple physical
29 qubits to actively detect and correct errors. The threshold theorem ensures that if the physical error rate is
30 below a certain threshold value, then logical errors can be suppressed arbitrarily by increasing the code size.
31 For surface codes, one of the most studied QEC schemes, this threshold is on the order of 1% and follows
32 the relation [4]:

$$P_L \sim \left(\frac{P}{P_{\text{thre}}} \right)^{\frac{d+1}{2}}$$

33 to suppress the logical error rate P_L per stage, where d is the code distance, P is the physical error rate
34 per stage, and P_{thre} is the error threshold. This relation also shows how quantum error correction mitigates
35 logical errors P_L by mapping physical operations into logical operations.

36 Suppose one physical qubit in the Shor code suffers a small coherent error:

$$U_X(\theta) = e^{-i\frac{\theta}{2}X} = \cos\left(\frac{\theta}{2}\right)I - i\sin\left(\frac{\theta}{2}\right)X.$$

37 For small θ , this can be approximated as

$$U_X(\theta) \approx I - i\frac{\theta}{2}X.$$

38 Action on an encoded state

39 Let $|\psi_L\rangle$ denote the encoded logical state. After the error, the state becomes

$$U_X(\theta)|\psi_L\rangle = \cos\left(\frac{\theta}{2}\right)|\psi_L\rangle - i\sin\left(\frac{\theta}{2}\right)X_j|\psi_L\rangle,$$

40 where X_j denotes a bit-flip on physical qubit j .

41 Thus, the corrupted state is a superposition of:

- 42 • the “no error” branch with amplitude $\cos(\frac{\theta}{2})$, and
- 43 • the “error on qubit j ” branch with amplitude $-i\sin(\frac{\theta}{2})$.

44 In this article, we will first highlight the importance of quantum error correction. We will then examine
45 the work based mainly on [6], aiming to provide explanations, reproduce key results, and draw inspiration
46 from their findings.

47 2 Stabilizer Formalism

48 A well-developed framework used to efficiently characterize quantum error correction codes is the *stabilizer*
49 *formalism*. It describes a code space as the simultaneous +1 eigenspace of a set of commuting Pauli operators,
50 called *stabilizers*. Errors are detected by measuring these stabilizers: if an error anticommutes with a
51 stabilizer, the corresponding measurement outcome flips, providing an error syndrome.

52 **Stabilizer code.** A *stabilizer code* on n qubits is specified by an abelian subgroup $\mathcal{S} \subseteq \mathcal{P}_n$ (the n -qubit
53 Pauli group) that does not contain $-I$. The *codespace* \mathcal{C} is the joint $+1$ eigenspace of all elements of \mathcal{S} :

$$\mathcal{C} = \{ |\psi\rangle \in (\mathbb{C}^2)^{\otimes n} : S|\psi\rangle = |\psi\rangle \quad \forall S \in \mathcal{S} \}.$$

54 **Dimension.** If the code encodes k logical qubits into n physical qubits (an $[[n, k, d]]$ code), then with $n - k$
55 independent stabilizer generators we have

$$\dim \mathcal{C} = \frac{2^n}{2^{n-k}} = 2^k.$$

56 **Logical basis inside the codespace.** Because $\dim \mathcal{C} = 2^k$, we may choose an orthonormal *logical basis*
57 of \mathcal{C} ,

$$\mathcal{B}_L = \{ |x_L\rangle : x \in \{0, 1\}^k \},$$

58 such that each basis vector lies in the codespace (hence is stabilized):

$$S|x_L\rangle = |x_L\rangle \quad \forall S \in \mathcal{S}, \forall x \in \{0, 1\}^k.$$

59 **Arbitrary logical state (spanning by the logical basis).** Any generic codespace vector $|\psi\rangle \in \mathcal{C}$ can
60 be expressed *in the logical basis* as

$$|\psi\rangle \equiv |\psi_L\rangle = \sum_{x \in \{0, 1\}^k} \alpha_x |x_L\rangle, \quad \sum_x |\alpha_x|^2 = 1.$$

61 Thus the logical basis $\{|x_L\rangle\}$ spans the same subspace that was defined abstractly by the condition $S|\psi\rangle =$
62 $|\psi\rangle$.

63 **Expansion in the physical (computational) basis.** Each logical basis vector is itself a vector in the
64 n -qubit Hilbert space and typically expands as a superposition of computational basis states:

$$|x_L\rangle = \sum_{i=0}^{2^n-1} c_i^{(x)} |i\rangle,$$

65 with coefficients $\{c_i^{(x)}\}$ constrained by the stabilizer conditions $S|x_L\rangle = |x_L\rangle$ for all $S \in \mathcal{S}$. These constraints
66 select which computational-basis components may appear and with what relative phases or amplitudes.

67 **Remark (stabilizers vs. logical operators).** Stabilizers act *trivially* on every codespace vector (eigen-
68 value $+1$) and thus define \mathcal{C} . By contrast, *logical operators* act *nontrivially* within \mathcal{C} ; they lie in the normalizer
69 $N(\mathcal{S})$ of \mathcal{S} in \mathcal{P}_n but not in \mathcal{S} itself.

70 There are $n - k$ independent stabilizer generators, which generate the full stabilizer group of size $|\mathcal{S}| =$
71 2^{n-k} . These $n - k$ constraints reduce the full 2^n -dimensional Hilbert space to the 2^k -dimensional code space.
72 In other words:

- 73 • n physical qubits provide a Hilbert space of dimension 2^n ,
- 74 • $n - k$ stabilizer constraints remove $n - k$ degrees of freedom,
- 75 • leaving k logical qubits, i.e. a code space of dimension 2^k (same as logical state dimension).

76 Within this framework, many of the most important quantum error correction codes—including repetition
77 codes, concatenated codes (e.g., Shor code), the color code of Hamming codes (e.g., Steane code), surface
78 codes, and subsystem codes (e.g., Bacon-Shor code)—can be described in a unified and elegant way.

79 **3 Homomorphic Logical Measurements (Notes on the Talk and**
80 **Paper [5, 7])**

81 **3.1 Surface code**

82 Each stabilizer act on neighbor local qubits. The error threshold is low but the code distance cannot be well
83 increased even with larger physical qubits number. The relation write $kd^2 = O(n)$ with n, k, d being the
84 typical $[[n, k, d]]$ definition error code. Here we can see that if restricting on encoding rates $\frac{k}{n} \sim 1$, code
85 distance d scales as $O(1)$. Noted that for linear code, $n \geq k + d - 1$

86 **3.2 Quantum LDPC (Low-Density Parity-Check) code**

87 Decoding time is large to cost computation delays, while fast decoding is an essential ingredient to fault-
88 tolerant computation. Sparce stabilizers (low weight hamming weight) can improve the problem [11]. Quan-
89 tum LDPC code provide nonlocal stabilizers, measurements. The code distance d can be increased faster not
90 following $kd^2 = n$ (code rate: $\frac{k}{n}$). In addition, one motivation comes from when standard Shor and Steane
91 style logical measurement cannnot be performed on large quantum LDPC code.

92 For typical surface code, code rate scales asymptotically to zero and with square root of code distance
93 when enlarging code block. Improvement gives nonvanishing encoding rate for different surfaces (more non-
94 trivial loops), but with code distance logarithmic in the blocklength. **Hypergraph product construction**
95 improve this problem: First of all, we have

$$\text{Toric code} \subset \text{Homological codes} \subset \text{Hypergraph product codes} \subset \text{Stabilizer codes}.$$

96 Noted that homological codes belong to mutually orthogonal binary codes, and stabilizer codes belong to
97 additive self-orthogonal code over GF (4) with respect to the trace Hermitian inner product

98 **Theorem 1:** it guarantees that from any full-rank classical LDPC parity-check matrix H , you can system-
99 atically build a quantum LDPC code whose parameters are exactly those given.

Classical	Quantum (constructed)	Notes
Code $[n, k, d]$	$\rightarrow [[n^2 + (n-k)^2, k^2, d]]$	Quantum code parameters
LDPC (sparse) row weight i , column weight j	\rightarrow LDPC (row weight $\approx i+j$)	Sparsity preserved
Parity-check matrix H	$\rightarrow (H_X, H_Z)$ built from $H \otimes I, I \otimes H^T$	CSS-type stabilizers
Distance d	\rightarrow Distance d	Same as classical code
Rate k/n	$\rightarrow \frac{(k/n)^2}{1 + (1 - k/n)^2}$	Quantum rate expression

100 **LDPC codes** linear codes with sparse parity check matrix and can also be described by Tanner graph
101 denoted by bipartite $\mathcal{T}(V, C, E)$. For $H = \mathbb{F}_2^{r \times n}$, $V = 1, \dots, n$ (called variable nodes) is the columns of H and
102 $C = \otimes_1, \dots, \otimes_r$ (check nodes) with column indices i and row indices j . There is an edge set E when $H_{ij} = 1$

103 **Generalizations from Toric code** An $m \times m$ toric code (V, E) can be represented as $\mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$, where
104 the two-dimensional vertex set consists of coordinates (x, y) with each coordinate ranging over $\{0, 1, 2, \dots, m-1\}$. The vertex-edge incidence matrix \mathbf{H}_1 is defined such that $(\mathbf{H}_1)_{ij} = 1$ if vertex i is incident to edge j .
105 Each i -th row of \mathbf{H}_1 corresponds to a vertex (an X -stabilizer), and each j -th column corresponds to an
106 edge, which represents a physical qubit. Pictorially, for a four qubits repetition code (building block of toric
107 code) can be denoted as in Table 1. Let $H_1 \in \{0, 1\}^{r_1 \times n_1}$ and $H_2 \in \{0, 1\}^{r_2 \times n_2}$ be classical parity-check
108 matrices. Define identity matrices I_a of the indicated sizes, and use the Kronecker product \otimes . Then, the
109 CSS stabilizer matrices are given by
110

$$H_X = [H_1 \otimes I_{n_2} \mid I_{r_1} \otimes H_2^T], \quad H_Z = [I_{n_1} \otimes H_2 \mid H_1^T \otimes I_{r_2}].$$

X stabilizer (row)	edge ₀	edge ₁	edge ₂	edge ₃
X ₀	1	1	0	0
X ₁	0	1	1	0
X ₂	0	0	1	1
X ₃	1	0	0	1

Table 1: Toric code H_r matrix for 4 edges

111 **Toric code as a special case.** If both classical codes are chosen as the length- L repetition code with
 112 parity-check $H_r \in \{0, 1\}^{L \times L}$ (representing a cyclic ring), then the toric-code stabilizer matrices become

$$H_X = [H_r \otimes I_L \mid I_L \otimes H_r^\top], \quad H_Z = [I_L \otimes H_r \mid H_r^\top \otimes I_L].$$

113 Here the rows of H_X correspond to plaquette (face) X -stabilizers and the rows of H_Z correspond to vertex
 114 Z -stabilizers, while the columns index the $2L^2$ edge qubits of the lattice.

115 3.3 Logical measurements of Shor and Steane type

116 Standard approach will encounter two possible limitations. First, if an error occur on the ancilla qubits, the
 117 error will propagate to data qubits and cause higher weight errors. Below shows a graph of common error
 118 propagations extracted from [12]

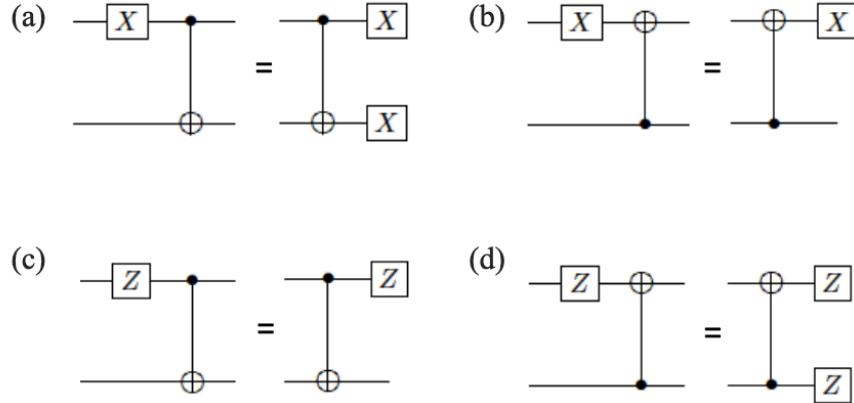


Fig. 3. Propagation of X and Z errors through the CNOT gates.

Figure 1:

119 Shor's fault-tolerant logical measurements are implemented by applying transversal gates between data
 120 qubits and ancilla GHZ (cat) states. The procedure requires multiple rounds, where each GHZ ancilla

121 interacts transversally with the data qubits and is then measured in the X (Z) basis, corresponding to initial
122 input state $|\overline{+}\rangle$ ($|\overline{0}\rangle$).

123 These repeated measurements allow one to perform majority voting on the syndrome outcomes, thereby
124 suppressing the effect of measurement errors. Fault tolerance requires that errors arising at any stage do not
125 propagate uncontrollably to the data qubits. Figure 2 illustrates this process.

126 One potential issue is that ancilla faults during syndrome extraction can propagate in such a way that
127 errors mimic measurement errors. To avoid this mixing, each round of syndrome extraction must itself be
128 implemented fault-tolerantly. By performing fault tolerant error correction in each state, a single fault can
129 only corrupt the outcome and then be fixed during that round. This guarantees that majority voting across
130 repeated rounds of cat state measurements produces valid syndrome information.

Shor's method requires repetitions of each stage to alleviate an probability

$$P = \frac{1}{2} - (1 - 2p)^d = \frac{1}{2} - \Delta$$

131 of logical error occurs, where p is the single qubit error probability and d is the circuit depth of each stage.
132 The majority vote requires $O(\epsilon^{2d})$ repetitions.

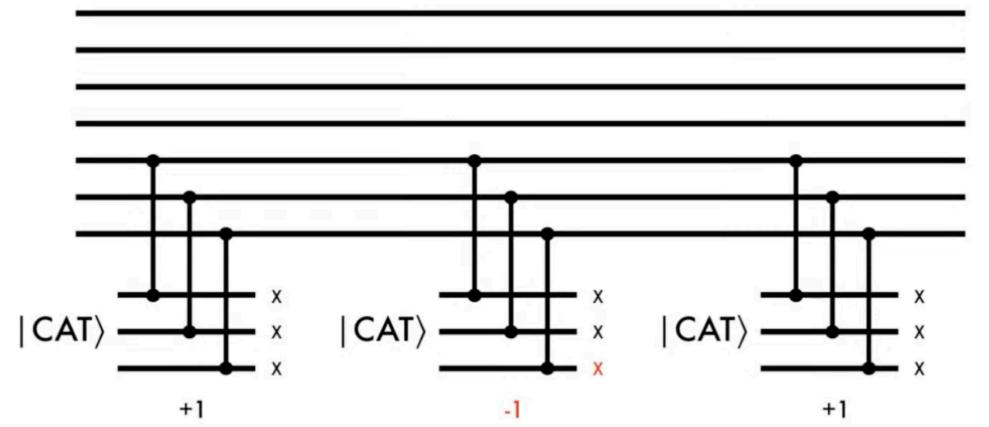


Figure 2:

133 In Steane method, logical measurement is performed by preparing an ancilla block encoded in the same
134 CSS code (e.g., $|0_L\rangle = \frac{1}{\sqrt{2}}(|++\rangle + |--\rangle)^{\otimes 3}$ or $|+_L\rangle$ for the Shor code.) and coupling it to the data
135 block with transversal CNOTs, realizing

$$\text{CNOT}^{\otimes n} = \overline{\text{CNOT}}^{\otimes k}$$

136 for an $[n, k, d]$ CSS code. This is in fact mapping the measurement outcome from data code block to ancilla
137 code block:

138 Let the data block be $|\psi\rangle = \alpha|0_L\rangle + \beta|1_L\rangle$ and the ancilla be $|0_L\rangle$. After transversal CNOTs: $|\psi\rangle|0_L\rangle \rightarrow$
139 $\alpha|0_L\rangle|0_L\rangle + \beta|1_L\rangle|1_L\rangle$. Measuring the ancilla block in the Z basis reveals the eigenvalue of Z_L on the data
140 block, while collapsing it into $|0_L\rangle$ or $|1_L\rangle$ accordingly.

Unlike Shor's cat-state method, which measures stabilizers one by one, Steane's method allows all stabilizers of one error type (either X -type or Z -type) to be extracted in a single round. One can imagine that once an measurement error occurs in Steane's parity checks, more parity checks outcome can be used to infer the syndromes compared with Shor's method that require more qubits for multiple stages measurements for one set of syndrome in each stage. The logical error rate yields

$$P = \mathcal{O} \left(p^{\frac{d-1}{2}} \right)$$

141 . Here, d is the code distance.

142 For a general $[n, k, d]$ code, it is easily to generalize the ancilla states to $|0\rangle = \overline{|+1\dots0_i\dots+k\rangle}$ for a Z
 143 type measurements, since $X_j|+_k\rangle$ leaves no change of the state, the measurement of Z_i will only extract
 144 information from i qubits (the state $|+_k\rangle$, which treats all Z measurement outcomes on an equal footing).
 145 $|0_i\rangle$ is just the ancilla state (measurement state) for Z_i stabilizers. Dimensions of $|0_i\rangle$ state is the weight of
 146 Z_i stabilizers. Here, we have noted that for even weight of stabilizers, they are related by local Hadamard
 147 gate, called Clifford-equivalent. Also, the choice of codewords are designed by both logical operators and
 148 stabilizers. A density matrix for logical state of one-qubit encoding can be found as [1]

$$|0_L\rangle\langle 0_L| = \frac{1}{2^n} (I + \overline{Z}_L) \prod_j (I + S_j).$$

149 For $|1_L\rangle$, one can change the plus sign to minus sign.

150 Problem in Steane code could be ancilla states $|0_L\rangle$ preparation [9]. It can be comprised of a non-fault
 151 tolerant preapartion process combined with a verification stage. The verification stage Fig. 3 requires and
 152 additional ancilla qubit to flag a successful preparation, like post-process. The whole process then become
 153 fault-tolerant but with successful rate e^{-np} , with n being number of gates and p the succesful probability
 154 of each gate. For state other than $|0_L\rangle$ can be prepared combined with Clifford operations. Noted that
 155 apart from Clifford operations, magic state injection ($T|+\rangle$) is also required to fulfill the universal quantum
 156 operations. Similarly, by state distillation or code concatenation, desired ancilla qubit states can be obtained
 157 but with large overheads [14].

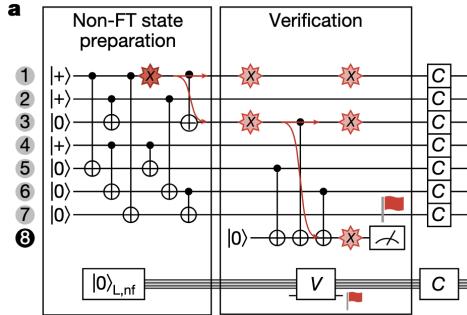


Figure 3:

158 Another trick for ancilla states creation in Steane code is by performing X-type measurements. It seems
 159 like we can create the codewords $|0_L\rangle$ by following a state projection from logical operators and stabilizers:

$$\frac{1}{2^J} (I + \overline{Z}_L) \prod_j^J (I + S_j).$$

160 If we perform X type measurements (mathematically described by above formula, while S_j being Z type
 161 measurements can be trivial), the mathematical description of this projecting process can be:

$$|0_L\rangle = \underbrace{\frac{1}{2^{J/2}} (I + \overline{Z}_L)}_{j \in X\text{-type}} \prod_j (I + S_j) |0^{\otimes n}\rangle + \underbrace{\frac{1}{2^{J/2}} (I + \overline{Z}_L)}_{j \in Z\text{-type}} \prod_j (I + S_j) \overline{|0^{\otimes n}\rangle}. \quad (1)$$

162 Noted that Z type measurements and logical Z_L measurement act trivially on $|0^{\otimes n}\rangle$ (They are already
 163 in stabilizer group or commute with stabilizers). The formula requires projective operator which could only
 164 be done unitarily. More explicity, an arbitrary state can be written as combination of projective states with
 165 different observables, hence we could write $|0^{\otimes n}\rangle = \frac{I+S_j}{2}|0^{\otimes n}\rangle + \frac{I-S_j}{2}|0^{\otimes n}\rangle$. This also demonstrate that

166 Eq. 1 have implicitly selectively choose the projective states $\frac{I+S_i}{2}|0^{\otimes n}\rangle$ with some probability. For Steane
 167 code, this probability is $(\frac{1}{2})^3 = \frac{1}{8}$ for three consecutive projecting process.

168 From $|0^{\otimes n}\rangle = \frac{I+S_i}{2}|0^{\otimes n}\rangle + \frac{I-S_i}{2}|0^{\otimes n}\rangle$, we could infer that the correponding error correction of Z-type
 169 could fix the problem when projecting into wrong states. Hence, the process require further fault-tolerant
 170 error correction (FTEC) following the stabilizers measurement to deterministically generate logical $|0_L\rangle$
 171 state(Above are my current understanding which may not correspond to what paper really trying to convey.).
 172 There is no need post-selection for Steane's ancilla qubit preparation as claimeed in the video for logical
 173 qubits number $k = 1$. Also, for $k > 1$ the process can be used to generate $|0^{\otimes k}\rangle$ (all Z measurement at
 174 once) but not $|\overline{+}_1 \dots \overline{0}_i \dots \overline{+}_k\rangle$ (If we want particular Z_i measurement). The reason is that $|\overline{+}_1 \dots \overline{0}_i \dots \overline{+}_k\rangle$ are
 175 not easily prepared anymore. This may make the whole preparation process as hard as directly measuring
 176 logical operators in data block.

177 A natural thoughts then will be can a new choice of ancilla code such that it can achieve a LDPC
 178 measurement on particular logical qubit. The next question is, is there any other choices of ancilla code to
 179 achieve non-postselection, no repition like Steane's method for an $[[m,1,d]]$ (Steane ancilla code: $[[n,1,d]]$ or
 180 $[[n, k ,d]]$)ancilla code. Here, the speaker aims to build a new code that could perform with $m < d$ that
 181 could be more resource freindly.

182 The speaker introduced a measurement process called *homomorphic measurement*. A toric code is an

$$[[n, k, d]] = [[2L^2, 2, L]]$$

183 defined on a torus, which can be represented as a square sheet with periodic boundary conditions. The
 184 stabilizers all commute, and the corresponding logical operators are shown in Fig. 10. The horizontal loops
 185 $\overline{X}_1, \overline{Z}_2$ and the vertical loops $\overline{Z}_1, \overline{X}_2$ correspond to logical operators that wrap around the torus in the
 186 horizontal or vertical directions.

187 3.4 Binary vector spaces

188 They construct the homomorphism between data qubits and the ancilla qubits by using CSS codes chain
 189 complexes.

190 An $r \times n$ binary matrix defines a linear map

$$H : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^r,$$

191 where $\mathbb{F}_2 = \{0, 1\}$ with addition and multiplication modulo 2. The transpose H^T is the $n \times r$ matrix with
 192 rows and columns swapped. The kernel (null space) is

$$\ker(H) = \{v \in \mathbb{F}_2^n : Hv = 0\},$$

193 the image (column space) is

$$\text{im}(H) = \{Hv : v \in \mathbb{F}_2^n\},$$

194 and the row space is the span of the rows of H , denoted $\text{rs}(H)$. Note that $\dim(\text{im}(H)) = \dim(\text{rs}(H)) =$
 195 $\text{rank}(H)$.

196 Given a finite set S , the vector space $\mathbb{F}_2[S]$ consists of all formal binary sums of elements in S ,

$$v = \sum_{e \in S} v_e e, \quad v_e \in \mathbb{F}_2,$$

197 which can be naturally identified with subsets of S (element e is present if $v_e = 1$). If $H : \mathbb{F}_2[A] \rightarrow \mathbb{F}_2[B]$,
 198 then the transpose defines a map $H^T : \mathbb{F}_2[B] \rightarrow \mathbb{F}_2[A]$ under the corresponding bases.

199 As an example, consider

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

200 The image is spanned by the columns $(1, 0)^T$, $(0, 1)^T$, and $(1, 1)^T$, which generate all of \mathbb{F}_2^2 . The row space
201 is spanned by $(1, 0, 1)$ and $(0, 1, 1)$, giving the subspace

$$\{(0, 0, 0), (1, 0, 1), (0, 1, 1), (1, 1, 0)\} \subseteq \mathbb{F}_2^3.$$

202 In the language of quantum error correction, the row space $\text{rs}(H)$ often corresponds to the stabilizer
203 group (constraints on codewords), the image $\text{im}(H)$ corresponds to possible syndrome outcomes, and the
204 kernel $\ker(H)$ corresponds to valid codewords with no detected error.

205 A CSS code with stabilizer X type and Z type will have corresponding stabilizer group isomorphic to
206 $\text{rs}(H_X)$ and $\text{rs}(H_Z)$.

207 The quantum code can be described using two families of Pauli stabilizers. The X-type stabilizer group
208 corresponds to parity checks that involve X operators, and it is isomorphic to the row space of H_X . Similarly,
209 the Z-type stabilizer group corresponds to parity checks that involve Z operators, and it is isomorphic to
210 the row space of H_Z .

211 The X-type logical operators are elements of $\ker(H_Z)$ (like centralizer), meaning they commute with all
212 Z-type checks and therefore preserve the Z-stabilizer constraints. Likewise, the Z-type logical operators are
213 elements of $\ker(H_X)$, since they commute with all X-type checks since a logical operator will stay in the
214 codespace.

215 The number of encoded logical qubits is the number of independent logical degrees of freedom that remain
216 after imposing all stabilizer constraints:

$$k = \dim(\ker(H_X)/\text{rs}(H_Z)) = \dim(\ker(H_Z)/\text{rs}(H_X))$$

217 (quotient subgroup: The elements of the quotient space V/W are the cosets of W . Each coset is of the
218 form $v + W$ for some $v, w \in V$. Algebraically, forming the quotient space V/W , V/W means we treat all
219 vectors that differ by an element of W as equivalent. Topologically, V/W is like shrinking W space into a
220 point. It is also like finding logical qubits dimension using $\dim(2^n/2^{n-k}) = k$). This formula says that logical
221 qubits live in the space of operators that preserve one type of stabilizer (the kernel) but are not redundant
222 with the other type (the row space). The X distance d_X measures how resilient the code is against bit-flip
223 (X-type) errors: it is the minimum number of qubits that must be flipped to implement a nontrivial logical
224 X operation. Formally,

$$d_X := \min\{|c| : c \in \ker(H_Z) \setminus \text{rs}(H_X)\}.$$

225 Similarly, the Z distance d_Z quantifies protection against phase-flip (Z-type) errors:

$$d_Z := \min\{|c| : c \in \ker(H_X) \setminus \text{rs}(H_Z)\}.$$

226 Finally, the overall code distance is

$$d = \min\{d_X, d_Z\},$$

227 which sets the maximum number of arbitrary single-qubit errors the code can reliably detect and correct.
228 Physically, the larger the distance, the more robust the code is against noise.

229 Quantum error correction uses this framework because stabilizers operators naturally form abelian groups
230 modulo phases (self-commute).

231 3.5 Algebraic Topology

232 Notes from the lecture [3], the **2-dimensional disk** is defined as

$$D^2 = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 \leq 1\}.$$

233 It consists of all points in the plane whose distance from the origin is less than or equal to 1. **Interior and**
234 **boundary**

$$\text{Int}(D^2) = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 < 1\}, \quad \partial D^2 = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 = 1\} = S^1. (\partial D^n = S^{n-1})$$

²³⁵ **Topological Meaning** In a CW complex, D^2 serves as a **2-cell**. Attaching a 2-cell means gluing a copy of D^2 along its boundary S^1 via a continuous map:

$$f : S^1 \rightarrow X^1.$$

²³⁷ For example:

- ²³⁸ S^2 is formed by attaching one D^2 to a point ($X^1 = X^0$ here, one D^0 zero D^1 , one $D^{2]}$, $\chi(S^2) = 2$ (χ defined below)) .
- ²⁴⁰ A torus T^2 is formed by attaching D^2 along a loop that winds in two directions (X^0 : a point, X^1 add ²⁴¹ two D^1 lines, $f : \partial D^1 = S^0 \rightarrow X^0$. X^2 : add a D^2 two dimensional disk $f : S^1 \rightarrow X^1$) ($X^2 : ab^{-1}a^{-1}b$, ²⁴² the direction of loop are glued will result in different shape, if $X^2 : ab^{-1}ab$ is a Klein bottle). $\chi(T^2) = 0$

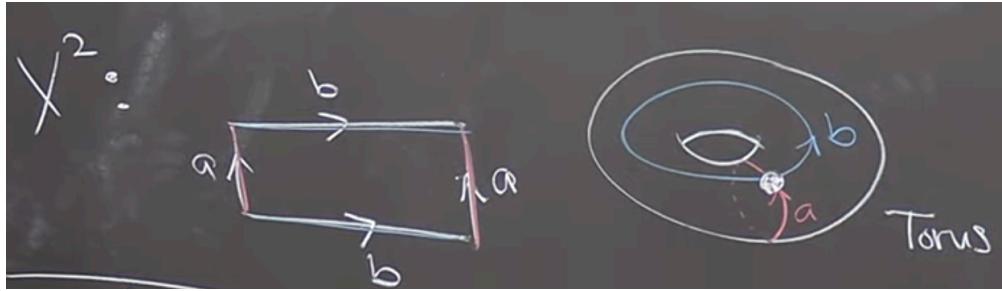


Figure 4: S^1 for torus. X^1 : add

²⁴³ **Generalization** The n -dimensional disk is

$$D^n = \{(x_1, \dots, x_n) \in \mathbb{R}^n \mid x_1^2 + \dots + x_n^2 \leq 1\},$$

$$f : S^{n-1} \rightarrow X^{n-1}.$$

²⁴⁵ **Euler characteristic** Vertices $D^0 = V$, Edges $D^1 = E$, Faces $D^2 = F, \dots$

$$\chi = \#\text{even dim}(D) - \#\text{odd dim}(D)$$

$$\chi(T^2) = V - E + F = 1 - E + 1 = 0 \Rightarrow E = 2$$

²⁴⁶ or a torus can be build from $V = 4$, $E = 8$, $F = 4$ and similar goes for S^2 but with χ fixed.

²⁴⁷ **Product and homology.** Noted that D^n is contractible and S^n are not.
²⁴⁸ **homotopy** \simeq

Spaces (X, Y)	Relationship	Intuition
D^n and a point *	$D^n \simeq *$	A disk can be shrunk to a point (contractible).
S^1 and a circle-shaped wire loop	$S^1 \simeq$ any loop	All circles have the same homotopy type
S^1 and a torus (T^2)	Not homotopy equivalent	A torus has more “holes.”
\mathbb{R}^n and a point	$\mathbb{R}^n \simeq *$	Can contract the entire space to a point.
A hollow cylinder and a circle	$S^1 \times I \simeq S^1$	The cylinder retracts onto its circular core.

²⁵⁰ The torus (solid torus: $D^2 \times S^1$) is defined as $T^2 = S^1 \times S^1$ and its *fundamental group* is $\pi_1(T^2) \cong \mathbb{Z} \times \mathbb{Z}$.
²⁵¹ In contrast, for the circle we have $\pi_1(S^1) \cong \mathbb{Z}$. Since the integer group \mathbb{Z} is not isomorphic to the product

252 group $\mathbb{Z} \times \mathbb{Z}$, it follows that $T^2 \not\simeq S^1$. Geometrically, if one tries to shrink the torus T^2 into a circle S^1 , one
 253 must collapse or “break” one of the gluing directions that form T^2 . Since this cannot be done continuously
 254 without tearing the surface, T^2 and S^1 are not homotopy equivalent. $D^1 \times D^2$: a solid cylinder (sphere)
 255 Some identities:

$$D^n \times D^m = D^{n+m}$$

$$\partial(X \times Y) = (\partial X \times Y) \cup (X \times \partial Y)$$

256 \cup is called union. For example, calculate $\partial(D^2 \times [1, 0]) = (\partial D^2 \times [1, 0]) \cup (D^2 \times \partial[1, 0]) =$
 257 $(S^1 \times [1, 0]) + D^2 \times \{0, 1\}$ It is exactly the surface of the cylinder. Or simply, $\partial(D^2 \times [1, 0]) = \partial D^3 // = S^2$
 258 So calculate $\partial(S^1 \times S^1 \times [1, 0]) = S^1 \times S^1 \times \{0, 1\}$ is two copies of torus surface.

259 Another example: $S^3 = \partial(D^4) = \partial(D^2 \times D^2) = S^1 \times D^2 \cup D^2 \times S^1$ (two tori formed by looping around
 260 different directions, pictorially, draw S^1 first for first qubit and then draw D^2 connected on S^1 similar for
 261 second torus but with opposite order.). Union can be think of gluing, hence gluing two tori is S^3 .

262 Examples of **Quotients** in topology: $D^1/S^0 = S^1$, $D^2/S^1 = S^2$, $S^2/S^1 = S^1 \vee S^1$, \vee (pronounce:
 263 wedge), also examples in Fig. 5

264 **Homology** group is used to describe

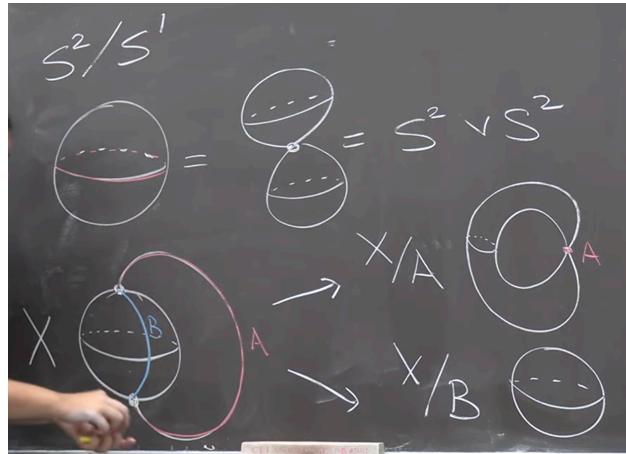


Figure 5:

265 3.5.1 Homology Groups

266 Vector spaces over \mathbb{F}_2 are abelian groups C_i under addition. Boundary operators ∂_i are group homomorphisms
 267 (Like in toric code, logical Z_L is noncontractible loops around the torus.). Groups here are in topological
 268 sense not the same as Stabilizers group in physical Pauli sense. A chain complex is just a sequence of abelian
 269 groups with compatible homomorphisms, typically written as

$$C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0, \quad \text{with } \partial_1 \circ \partial_2 = 0.$$

270 In CSS codes, $H_X = \partial_1$ and $H_Z^T = \partial_2$ naturally satisfy $H_X H_Z^T = 0$ is the stabilizers. The above
 271 **homology group** is used to describe data qubits C_1 and logical operators ($\ker(\partial_1)/\text{im}(\partial_2)$)

272 Logical operators are homology classes H_i . They are cycles Z_n (commute with stabilizers) but not
 273 boundaries B_n themselves (not product of stabilizers). Mathematically:

$$H_n = Z_n / B_n \tag{2}$$

$$Z_n := \ker \partial_n := \{ c \in C_n \mid \partial_n(c) = 0 \} \tag{3}$$

$$B_n := \text{im } \partial_{n+1} := \{ \partial_{n+1}(c) \mid c \in C_{n+1} \} \tag{4}$$

274 This correspond to $\dim(\ker(\partial_n)/\text{im}(\partial_{n+1})) = \dim(\ker(H_z)/\text{rs}(H_x)) = k$. The algebra links with Fig. 10 toric
 275 code. The toric code can be expressed as the chain complex $C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0$, where qubits live on edges
 276 (C_1), X -stabilizers are associated with vertices (C_0), and Z -stabilizers with faces (C_2). The logical operators
 277 are characterized by the first homology group

$$H_1 = \ker(\partial_1)/\text{im}(\partial_2) \cong \mathbb{Z}_2 \oplus \mathbb{Z}_2 = (0,0), (1,0), (0,1), (1,1),$$

278 which corresponds to the two nontrivial loops around the torus that encode the logical qubits, while torus
 279 requires two loops to decribe its topology. In general,

$$\ker(\partial H_Z)/\text{im}(\partial H_X)$$

280 corresponds to the X -type logical operators, while

$$\ker(\partial H_X)/\text{im}(\partial H_Z)$$

281 corresponds to the Z -type logical operators. Pictorially, one can imagine that all closed loops of errors on
 282 qubits in Fig. 10 lie in $\ker(\partial H_Z)$, but many of them can also be formed as products of X stabilizers. The
 283 only exceptions are loops that connect opposite edges (loop around) of the torus, which give nontrivial errors
 284 that cannot be detected and by design act as logical Z operators.

285 3.6 Homomorphic logical measurements

286 As we have elaborated on Shor and Steane measurement downsides and limitations, here we dorectly go
 287 to the arthor main points, homomorphic logical measurements. They are trying to find a new code $[[m, 1
288 ,d]]$ that could unifying or improve before mentioned downsides. The process first start from preparing 1.
 289 preparing ancilla in $|0^{\otimes k}\rangle$ 2. perform interaction Γ between ancilla and data block. 3. measured Z basis on
 290 ancilla block.

291 Data–Ancilla Interaction

292 Applying the homomorphism for CSS codes into their ancilla code construction by considering possible
 293 interaction between data-ancilla interaction (typically utlising similar mathematical but applying on different
 294 purposes.):
 295 We have two CSS codes: - Data: (H_X, H_Z) of length n , - Ancilla: (H'_X, H'_Z) of length m . Before interaction,
 296 stabilizer groups are written as

$$T_Z = \text{rs} \begin{pmatrix} H_Z & 0 \\ 0 & H'_Z \end{pmatrix}, \quad T_X = \text{rs} \begin{pmatrix} H_X & 0 \\ 0 & H'_X \end{pmatrix}.$$

297 After Interaction (Γ a gate matrix $\Gamma : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^n$ (CNOTs)), stabilizer groups can be written as

$$T'_Z = \text{rs} \begin{pmatrix} H_Z & 0 \\ H'_Z \Gamma^T & H'_Z \end{pmatrix}, \quad T'_X = \text{rs} \begin{pmatrix} H_X & H_X \Gamma \\ 0 & H'_X \end{pmatrix}.$$

298 To explain T'_Z further, it is like the outcome of H'_Z on ancilla qubits, is determined not only by the state
 299 initial state $|0_L\rangle$ lie in ancilla block but also $H'_Z \Gamma^T$ when performing interaction, which is like a different
 300 mapping other than Steane style, from my understanding, Steane style measurement follows $H'_Z \Gamma^T = H'_Z$
 301 (Since Γ here is like identity for transversal gates in Steane measurement) and also $H'_Z = H_Z$ since they
 302 are using same logical codewords, hence same stabilizers. Just like the author mentioned, for Shor's style
 303 measurment, $H'_Z \neq H_Z$ since H_Z should correspond to cat states stabilzers (1D).

304 The role interchange between target and controlled of X type and Z type errors can be explained by the
 305 error propagation shown in Fig. 1.

306 We also required conditions such $T'_Z = T_Z$, $T'_X = T_X$, i.e.

$$\text{rs}(H'_Z \Gamma^T) \subseteq \text{rs}(H_Z), \quad \text{rs}(H_X \Gamma) \subseteq \text{rs}(H'_X).$$

307 This ensures the interaction preserves the stabilizer groups.

308 **Definition (Homomorphic gadget).** An $[[m, k', d']]$ homomorphic gadget (H'_X, H'_Z, Γ) for an $[[n, k, d]]$
309 CSS code (H_X, H_Z) consists of: (i) an ancilla $[[m, k', d']]$ CSS code with checks (H'_X, H'_Z) ; (ii) a gate matrix
310 $\Gamma : \mathbb{F}_2^m \rightarrow \mathbb{F}_2^n$; such that

$$\text{rs}(H'_Z \Gamma^T) \subseteq \text{rs}(H_Z), \quad \text{rs}(H_X \Gamma) \subseteq \text{rs}(H'_X). \quad (5)$$

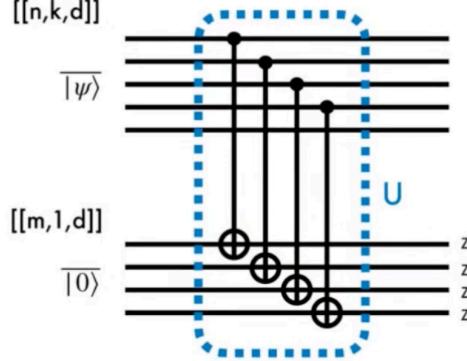


Figure 6:

311 To summarize, a stabilizer element (in fact also logical error) $v \in \ker H'_Z$ are transformed as $\Gamma v \oplus v$, acted
312 on data block (Γv) and ancilla block (v), since ancilla state is prepared in logical $|0^{\otimes k}\rangle$, the outcome will be
313 Γv . There are two cases: where $v \in \text{rs}(H'_Z)$ or $v \notin \text{rs}(H'_Z)$, the former under homomorphic gadget setting
314 will preserve the structure of $v \in \text{rs}(H'_Z)$ and act as a X error detection for data block. The latter are in
315 fact mapping logical Z operation into ancilla block. As mentioned, the outcome will be Γv (it is measured
316 in ancilla block but in fact bring based on data block information. One can simply assume a vector acting
317 by matrix T'_Z to see this) which will isomorphic to $\Gamma \ker(H_X)$ (seems might encounter vector space outside
318 $\Gamma \ker(H_X)$)

319 3.7 Homomorphic measurements on surface codes

320 Surface codes are defined as cellulations of a manifold $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$, where the boundary maps $\partial_2 : \mathcal{F} \rightarrow \mathcal{E}$
321 and $\partial_1 : \mathcal{E} \rightarrow \mathcal{V}$ obey the CSS code condition $\partial_1 \partial_2 = 0$. (Can LDPC CSS codes, such as hypergraph product
322 codes, have different homomorphic gadgets?) Linear maps $\gamma : \mathcal{A} \rightarrow \mathcal{D}$ connect the ancilla and data surface
323 codes. In fact, the gate matrix is given by $\Gamma = \gamma_1$ in the paper, where $\gamma_1 : \mathcal{E}' \rightarrow \mathcal{E}$ is the linear map between
324 qubits. The data and ancilla surface codes are defined respectively as $\mathcal{D} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$, and $\mathcal{A} = (\mathcal{V}', \mathcal{E}', \mathcal{F}')$.
325 Explicitly, the relation between data block and ancilla block:

$$\begin{array}{ccccc} \mathbb{F}_2[\mathcal{F}'] & \xrightarrow{\partial'_2} & \mathbb{F}_2[\mathcal{E}'] & \xrightarrow{\partial'_1} & \mathbb{F}_2[\mathcal{V}] \\ \downarrow \gamma_2 & & \downarrow \gamma_1 & & \downarrow \gamma_0 \\ \mathbb{F}_2[\mathcal{F}] & \xrightarrow{\partial_2} & \mathbb{F}_2[\mathcal{E}] & \xrightarrow{\partial_1} & \mathbb{F}_2[\mathcal{V}] \end{array}$$

326 The above relation naturally gives homomorphic gadget conditions shown in Eq. 5 , as $\gamma_1 \partial'_2 = \partial_2 \gamma_2 \subseteq \partial_2$ and
327 $\gamma_0 \partial'_1 = \partial'_1 \gamma_1 \subseteq \partial_1$. The paper seems like weakening the global homeomorphism constraints of a usual linear
328 map Γ (γ_i) such as Steane or Shor to local homeomorphism. This generalization gives more degree of freedom

329 to represent logical operators to a single non-contractable loop in a new manifold. This generalization do
 330 not preserve transversal gates, as we can see that γ_i local homeomorphism, or covering spaces can be many-
 331 to-one linear maps. There are also certain boundaries for manifold M , with two rough boundaries and two
 332 smooth boundaries is the planar surface codes [4].

333 The paper constructs homomorphic gadgets into two categories: **subspaces of data code space** \mathcal{D} and
 334 **covering space** of \mathcal{D} . For the first one, it is natural that homomorphic gadget can be constructed given Γ
 335 is injective (one-to-one, hence transversal and fault tolerant. $\mathcal{A}(\mathcal{V}', \mathcal{E}', \mathcal{F}') \in \mathcal{D}(\mathcal{V}, \mathcal{E}, \mathcal{F})$). Shor code can be
 336 thought of as $A = l \subseteq (\mathcal{V}, \mathcal{E})$ and l loops not intersecting (loops here generally mean logical operators, so
 337 not restricted on toric code loops, if loop intersects, it could involve two logical operators which is not in cat
 338 state gadget.), with $F = \emptyset$ and this indicates repetition code $0_L = \frac{1}{\sqrt{2}}(|+++...+|---...)$) will have only
 339 X stabilizers, for repetition code of Z stabilizers, one use T'_X which interchange the controlled and target
 340 qubits between data and ancilla qubits.

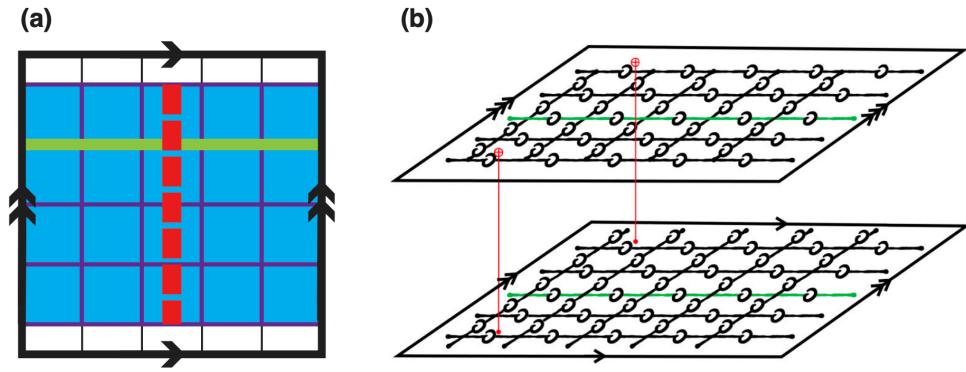


Figure 7:

341 Fig. 7 (a) describes a toric code \mathcal{D} and the blue region describes surface code with smooth boundaries.
 342 Red line connecting two boundaries are then logical X (or by seeing red lines crossing sets of Z stabilizers.).
 343 The logical operator dimension of \mathcal{A} are then reduced from four (complete mapping correspond to Steane
 344 measurement) to two. This ensures simpler preparation of ancilla states as mentioned in 3.3, which is also a
 345 problem associated with Steane measurement if utilising a complete mapping between data (might involve
 346 two or more logicals) and ancilla block.

347 Here we move on to homomorphic gadgets from **covering spaces**. Emphasizing the motivation again,
 348 if we want to perform a single-shot nondestructive logical CSS measurements on multiple logical operators
 349 (ancilla block), then the direct mapping such as Steane code or $A \in l$ inevitably support two logicals degree
 350 of freedom but with overlapped qubits, furthermore, multiple logical qubits ancilla is hard to prepare. The
 351 idea is to unfold the manifold to make logical operators uniquely represented by a non-intersecting loop in
 352 ancilla sheets. This resolves all of the problems mentioned.

353 **Groups acting on spaces:** The infinite *simply connected* covering space U (for example, \mathbb{R}^2) is equipped
 354 with a regular tiling (cellulation: divided into cells like vertices, edges, faces, $[i, i+1] \times [j, j+1]$ for $(i, j) \in \mathbb{Z}^2$).
 355 A group G of symmetries (leave the grid-structure intact), such as translations or rotations, acts on U , and
 356 for each point $u \in U$, its orbit $Gu = \{g(u) \mid g \in G\}$ consists of all symmetry-related copies of u (all Gu
 357 collapses to one point.). The orbit space U/G is the corresponding quotient manifold (for instance, a torus),
 358 and the quotient map (continuous and open) $p_G : U \rightarrow U/G$ sends each point u to its orbit Gu . The resulting
 359 manifold $M = U/G$ is the compact surface on which the surface code is defined. Because each element of G
 360 preserves the tiling of U , the quotient map p_G induces a cellulation of M ; that is, every k -cell in U maps to
 361 a k -cell in M , preserving the lattice structure.

362 As discussed in Sec. 3.5 and illustrated in Fig. 4, one can regard the **first example: torus** as the
 363 quotient of the real plane $\mathcal{U} = \mathbb{R}^2$ by the integer translation group $G \cong \mathbb{Z} \times \mathbb{Z}$

364 (translations $t_{r,s}(x,y) : (x,y) \rightarrow (x+dr, y+ds)$ for an $[2d^2, 2, d]$ toric code). Intuitively, this corresponds to
 365 identifying points that differ by integer shifts, i.e., taking 0 and 1 as the same point in each direction. The
 366 quotient $\mathbb{R}^2/(\mathbb{Z} \times \mathbb{Z})$ can thus be represented by the unit square $[0,1] \times [0,1]$, where opposite edges—labeled
 367 a and b in Fig. 4—are glued together to form the torus topologically. Because the torus is constructed as
 368 this quotient, its *fundamental group* is isomorphic to the translation group itself,

$$\pi_1(T^2) \cong \mathbb{Z} \times \mathbb{Z},$$

369 with each generator corresponding to one of the two noncontractible loops along the a and b directions.

370 We can also consider **second example: hyperbolic surface codes**, with the universal $\mathcal{U} = \mathbb{H}^2$ which
 371 are defined on regular tilings characterized by a Schläfli symbol $\{r, s\}$ (note that this is unrelated to the
 372 integer coordinates (r, s) used earlier). Here, r indicates that each face (tile) is a regular polygon with r
 373 sides, and s means that s such faces meet at each vertex. The pair $\{r, s\}$ determines both the curvature of
 374 the surface and the stabilizer structure: if $(r-2)(s-2) < 4$, the surface is spherical; if $(r-2)(s-2) = 4$,
 375 it is Euclidean (flat, as in the toric code); and if $(r-2)(s-2) > 4$, it is hyperbolic. In the code, each
 376 Z -type stabilizer acts on r qubits (around a face), and each X -type stabilizer acts on s qubits (around a
 377 vertex). The **Coxeter group** $G_{r,s}$ preserve the tiling structure. Group G is chosen as the normal subgroup
 378 of $G_{r,s}$ (like relation between Pauli group and Clifford group). The parameters $\llbracket n, k, d \rrbracket$ satisfy $k = O(n)$
 379 and $d = O(\log n)$.

380 (This passage formalizes how one can form a quotient manifold \mathcal{U}/G by identifying points under a group of
 381 **local homeomorphism**, in a way that preserves the cellulation and thus the qubit and stabilizer structure
 382 of the original topological code.)

383 The *image* of N_u , $g(N_u)$ is the set of all points in \mathcal{U}/G that are reached when applying the map p_G to every
 384 point in $N_v : p_G(N_u) = \{p_G(x) \mid x \in N_u\}$. Therefore, if N_v is a small open patch around v in the original
 385 space \mathcal{U} , then the set $N_v := p_G(N_v)$ is the corresponding small open patch and disjoint in the quotient space
 386 \mathcal{U}/G . N_v and $g(N_u)$ are homeomorphic. Also, no nontrivial $g(u) = u$ (Not fixed points mean mapping all
 387 of the points to N_v , bijective: injective and surjective)

388 **Third example: $\llbracket 2d^2, 2, d \rrbracket$ toric code.** if we choose U_u for any $u = (x, y) \in \mathcal{U} = \mathbb{R}^2$

389 The *lifting property* is the key topological feature they rely on, since it allows any logical operator—
 390 represented by a noncontractible loop on the base surface to be lifted to a non-self-intersecting path on a
 391 multi-sheeted covering manifold. The loop on \mathcal{U} starts at u and ends at some translated copy of $g(u)$.

392 **Fourth example: $\llbracket 2d^2, 2, d \rrbracket$ toric code.** Lifting a horizontal loop l on \mathcal{U}/G to \tilde{l} . Mathematically,
 393 denoted as $g(u) = t_{10}(u)$, where $u = (0, 0) \in \mathcal{U}$. One can imagine logical operator correspond to \mathcal{U}/G is like
 394 viewing $(0, y)$ and (d, y) as same point. Also, \tilde{l} is guaranteed to be a loop if and only if l is contractible on
 395 \mathcal{U}/G given U is simply connected.

396 Consider another covering map $p_G^H : \mathcal{U}/H \rightarrow \mathcal{U}/G$ defined as $p_G^H H(u) = G(u)$. When $H = \langle t_{1,0} \rangle$
 397 (horizontal translations), the intermediate covering space \mathcal{U}/H is an infinite *cylinder*, obtained by identifying
 398 points along the horizontal direction of the universal cover $U = \mathbb{R}^2$. The base space \mathcal{U}/H , where $G =$
 399 $\langle t_{1,0}, t_{0,1} \rangle$, is the *torus*, obtained by identifying both horizontal and vertical directions. On the torus \mathcal{U}/H ,
 400 the horizontal and vertical logical loops correspond to $t_{1,0}$ and $t_{0,1}$, respectively. When lifted to the cylinder
 401 \mathcal{U}/H , the horizontal loop remains closed since $t_{1,0} \in H$, while the vertical loop becomes an open segment as
 402 $t_{0,1} \notin H$. This pictorizes the general relation

$$g \in H \iff \text{the lifted loop } \ell \text{ is closed on } \mathcal{U}/H.$$

403 **Fifth example: $\llbracket 2d^2, 2, d \rrbracket$ toric code** is same as the previous example for relation
 404 $g \in H \iff \text{the lifted loop } \ell \text{ is closed on } \mathcal{U}/H$.

405

406 3.8 Homomorphic gadgets for covering spaces

407 Now we can start to construct homomorphic gadgets for covering spaces. Until now, we make some remarks:
 408 $g(u)$ lives in \mathcal{U} and $p_G(u)$ lives in \mathcal{U}/G . For loops, $p_G(g(u)) = p_G(u)$ on \mathcal{U}/G but $g(u) \neq u$ on \mathcal{U} could

409 be possible. This could directly be seen $p_G(u) = Gu = \{g(u) \mid g \in G\}$ while p_G represented all possible
 410 $g(u) \in \mathcal{U}$ and collapse to one point in space \mathcal{U}/G by definition.

411 The task is to find $\mathcal{A} \subseteq \tilde{D} = \mathcal{U}/H$ (where H is defined previously) such that $\mathcal{A} \subset l'$ and satisfies $d_{\mathcal{A}} = d_{\mathcal{D}}$.

412 As discussed before, the subgroup $H \supseteq G$, and $p = p_G^H$ is the covering map from \mathcal{U}/H to \mathcal{U}/G . If we pick
 413 $H = \langle g \rangle$ ($g \in G$), then all the loops are unfolded except the loop l corresponding to the g -translation.

414 Specifically, we map two non-contractible loops to one non-contractible loop in the ancilla block $A \subseteq$
 415 $\tilde{D} = \mathcal{U}/H$. This ensures that we only have one unique logical operator in \mathcal{U}/H , where H is chosen to be
 416 $\langle t_{1,1} \rangle$ (i.e., no overlapping qubits like in the toric code with different logicals). This unique logical operator
 417 can be designed to represent $\overline{Z_1 Z_2}$, enabling single-shot measurement. Noted that ancilla block \mathcal{A} is chosen
 418 such that $d_A = d_D$ (minimum weight of a nontrivial X logical operator of \mathcal{A} , is the red line part in Fig. 9(c))

419 The induced homomorphic gadget (not necessarily transversal for covering maps) is induced by map
 420 $\gamma := p \circ \tilde{\gamma}$, where $\tilde{\gamma} : \mathcal{A} \rightarrow \tilde{D}$, $\gamma : \mathcal{A} \rightarrow \mathcal{D}$

421 One can obtain a clearer physical picture from Fig. 9. Panel (a) shows the data-qubit manifold $\mathcal{U}/G = \mathbb{T}^2$,
 422 where the green loops represent the logical operators $\overline{Z_1 Z_2}$. In panel (b), the corresponding logical loop ℓ
 423 is lifted to the covering space \mathcal{U} , forming a path that connects the two points (x, y) and $(x + d, y + d)$
 424 (connecting two grids), pictorially, imagine two grids collapse into one grid due to translation symmetry,
 425 then the green lines in Fig. 9(b) become Fig. 9(a). Finally, panel (c) illustrates the ancilla-qubit manifold
 426 \mathcal{U}/H , with $H = \langle t_{1,1} \rangle$ which takes the form of a cylinder. The covering spaces correspond to Fig. 9(c) is
 427 shown in Fig. 8. Noted that red line in Fig. 9(c) is logical X operator, when depicting in Fig. 8, it will
 428 become a line connected two smooth boundaries.

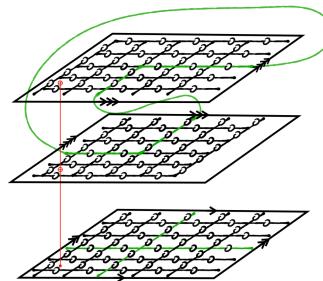


Figure 8:

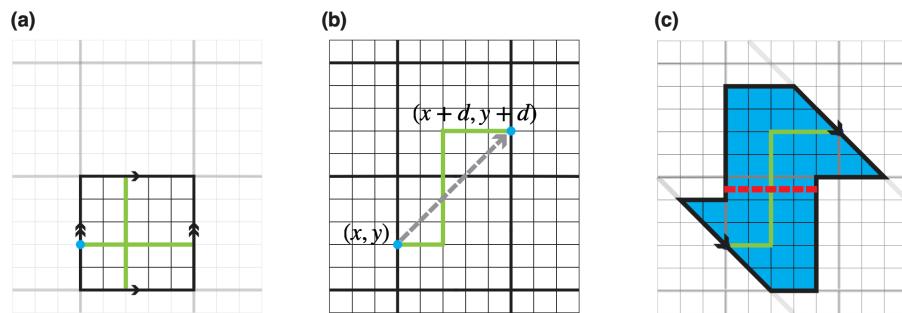


Figure 9: (a) Data-qubit manifold $\mathcal{U}/G = \mathbb{T}^2$, where the green loops represent the logical operators $\overline{Z_1 Z_2}$.
 (b) The covering space \mathcal{U} , showing the lifted path connecting (x, y) and $(x + d, y + d)$. (c) The ancilla-qubit
 manifold \mathcal{U}/H , which is topologically equivalent to a cylinder.

429 **3.9 Fault tolerance**

430 Since there is no transversal mapping for $\gamma := p \circ \tilde{\gamma}$, while homeomorphism between data sheets and ancilla
 431 sheets in standard measurement method is levergaed to local homeomorphism between them. The mapping
 432 between edges then might encounter many-to-one coupling, $\gamma_1^T(e) \in E'$. Even under these correlations, it is
 433 shown it still have fault tolerance with X error $\min\{d_{\mathcal{A}}, d_{\mathcal{D}}\}$.

434 **3.10 Joint measurement**

435 Considering two disjoint loops l_1 and l_2 on \mathcal{U}/G , if the manifold \mathcal{M} is path connnected, then logical operator
 436 can be $l_1 p l_2 p^{-1}$.

437 For two separate codes, say two ancilla blocks $\mathcal{A}_1, \mathcal{A}_2$, in order to prepare ancilla, one uses a lattice
 438 surgery approach to entangle two blocks from the initial state $|+\rangle_1 |+\rangle_2$ into the logical Bell state $|+\rangle_L =$
 439 $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ by measuring $Z_{\mathcal{A}_1} Z_{\mathcal{A}_2}$ with some surface code A' satisfying $\partial A' = l'_1 \cup l'_2$. (Note that the
 440 results will be either $|+\rangle_L = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ or $|-\rangle_L = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$. One then applies X_1 for correction.)
 441 Just like the layer of ancilla blocks depicted in Fig. 8, $Z_{\mathcal{A}_i}$ can be a closed loop on the boundary, $l'_i \subseteq \partial \mathcal{A}_i$.
 442 After ancilla preparation, one could construct homomorphic gadget (entangle data block and ancilla block)
 443 and perform logical measurement afterwards. Ancilla states can be prepared *offline*.

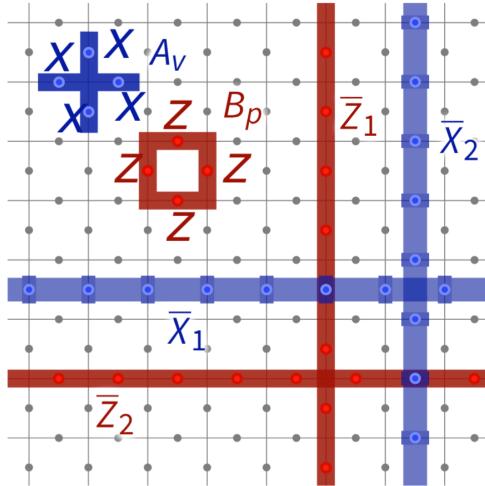


Figure 10:

444 **4 Summary**

445 They first establish the algebraic conditions under which the interaction matrix $\Gamma = \gamma_1$ between a data
 446 block $\llbracket n, k, d \rrbracket$ and an ancilla block $\llbracket n', k', d' \rrbracket$ preserves the stabilizer structure. Motivated by topological
 447 intuition, the authors represent intersecting logical loops as a single noncontractible loop. This construction
 448 achieves two goals: (1) it enables single-shot measurement of multiple logical operators, and (2) it simplifies
 449 the logical state preparation of the ancilla block.

450 The intuition is formalized through *covering map* between the topological structures (vertices, edges, and
 451 faces) of the data and ancilla codes. Such a map induces corresponding linear mappings between their chain
 452 complexes, ensuring that the homomorphic gadget conditions are automatically satisfied.

453 In this framework, the Steane measurement corresponds to a homeomorphic (one-to-one) chain map,
 454 while the homomorphic logical measurement generalizes it to a *covering map* (locally bijective but globally

455 many-to-one). This broader formulation naturally supports more general and scalable constructions of logical
456 measurements across CSS codes.

457 References

- 458 [1] <https://sites.google.com/site/danbrownneucl/teaching/lectures-on-topological-codes-and-quantum-computation>.
- 460 [2] Quantum error correction below the surface code threshold. *Nature*, 638(8052):920–926, 2025.
- 461 [3] Anthony Bosman. Algebraictopology, 2023.
- 462 [4] Austin G Fowler, Matteo Mariantoni, John M Martinis, and Andrew N Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A—Atomic, Molecular, and Optical Physics*, 86(3):032324, 2012.
- 465 [5] Shilin Huang. Homomorphic logical measurements | qiskit seminar series with shilin huang. 2023.
- 466 [6] Shilin Huang, Kenneth R. Brown, and Marko Cetina. Comparing shor and steane error correction using
467 the bacon-shor code. *Science Advances*, 10(45):eadp2008, 2024.
- 468 [7] Shilin Huang, Tomas Jochym-O'Connor, and Theodore J Yoder. Homomorphic logical measurements.
469 *PRX Quantum*, 4(3):030301, 2023.
- 470 [8] CY Liu, CG Feyisa, Muhammad S Hasan, and HH Jen. High-fidelity multipartite entanglement creation
471 in non-hermitian qubits. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 58(7):075501,
472 2025.
- 473 [9] Lukas Postler, Sascha Heußen, Ivan Pogorelov, Manuel Rispler, Thomas Feldker, Michael Meth,
474 Christian D Marciak, Roman Stricker, Martin Ringbauer, Rainer Blatt, et al. Demonstration of fault-
475 tolerant universal quantum gate operations. *Nature*, 605(7911):675–680, 2022.
- 476 [10] AV Rynbach, A Muhammad, AC Mehta, J Hussmann, and J Kim. A quantum performance simulator
477 based on fidelity and fault-path counting. *arXiv preprint arXiv:1212.0845*.
- 478 [11] Jean-Pierre Tillich and Gilles Zémor. Quantum ldpc codes with positive rate and minimum distance pro-
479 portional to the square root of the blocklength. *IEEE Transactions on Information Theory*, 60(2):1193–
480 1202, 2013.
- 481 [12] Andre Van Rynbach, Ahsan Muhammad, Abhijit C Mehta, Jeffrey Hussmann, and Jungsang
482 Kim. A quantum performance simulator based on fidelity and fault-path counting. *arXiv preprint
483 arXiv:1212.0845*, 2012.
- 484 [13] Ye Wang, Stephen Crain, Chao Fang, Bichen Zhang, Shilin Huang, Qiyao Liang, Pak Hong Leung, Ken-
485 neth R Brown, and Jungsang Kim. High-fidelity two-qubit gates using a microelectromechanical-system-
486 based beam steering system for individual qubit addressing. *Physical Review Letters*, 125(15):150505,
487 2020.
- 488 [14] Yi-Cong Zheng, Ching-Yi Lai, and Todd A Brun. Efficient preparation of large-block-code ancilla states
489 for fault-tolerant quantum computation. *Physical Review A*, 97(3):032331, 2018.