2 Stabilizer Formalism

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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 select which computational-basis components may appear and with what relative phases or amplitudes.

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

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

- n physical qubits provide a Hilbert space of dimension 2^n,

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

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

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

3.1 Surface code

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

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

Decoding time is large to cost computation delays, while fast decoding is an essential ingredient to fault-tolerant computation. Sparce stabilzers (low weight hamming weight) can improve the problem [12]. Quantum LDPC code provide nonlocal stabilzers, measurements. The code distance d can be increased faster not following $kd^2 = n$ (code rate: $\frac{k}{n}$). In addition, one motivation comes from when standard Shor and Steane style logical measurement cannnot be performed on large quantum LDPC code.

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

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

Noted that homological codes belong to mutually orthogonal binary codes, and stabilizer codes belong to additive self-orthogonal code over GF (4) with respect to the trace Hermitian inner product
Theorem 1: it guarantees that from any full-rank classical LDPC parity-check matrix H, you can systematically build a quantum LDPC code whose parameters are exactly those given.

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

LDPC codes linear codes with sparce parity check matrix and can also be described by Tanner graph denoted by bipartite $\mathcal{T}(V, C, E)$. For $H = \mathbb{F}_2^{r \times n}$, $V = 1, ...n$ (called varaible nodes) is the columns of H and $C = \otimes_1, ... \otimes r$ (check nodes) with column indices i and row indices j. There is an edge set E when $H_{ij} = 1$v
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 the two-dimensional vertex set consists of coordinates (x, y) with each coordinate ranging over $\{0, 1, 2, \ldots, m-$

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. Each i-th row of $\mathbf{H_1}$ corresponds to a vertex (an X-stabilizer), and each j-th column corresponds to an edge, which represents a physical qubit. Pictorially, for a four qubits repetition code (building block of toric 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

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

matrices. Define identity matrices I_a of the indicated sizes, and use the Kronecker product $\otimes$. Then, the CSS stabilizer matrices are given by

$$H_X = \left[\, H_1 \otimes I_{n_2} \;\middle|\; I_{r_1} \otimes H_2^{\top} \,\right], \qquad H_Z = \left[\, I_{n_1} \otimes H_2 \;\middle|\; H_1^{\top} \otimes I_{r_2} \,\right].$$

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

$$H_X = \left[\, H_r \otimes I_L \;\middle|\; I_L \otimes H_r^{\top} \,\right], \qquad H_Z = \left[\, I_L \otimes H_r \;\middle|\; H_r^{\top} \otimes I_L \,\right].$$

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

3.3 Logical measurements of Shor and Steane type

Standard approach will encounter two possible limitations. First, if an error occur on the ancilla qubtis, the error will propagate to data qubtis and cause higher weight errors. Below shows a graph of common error propgations extracted from [13]

Shor's fault-tolerant logical measurements are implemented by applying transversal gates between data qubits and ancilla GHZ (cat) states. The procedure requires multiple rounds, where each GHZ ancilla interacts transversally with the data qubits and is then measured in the X (Z) basis, corresponding to initial input state $|\overline{+}\rangle$ ($|\overline{0}\rangle$).

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

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

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

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

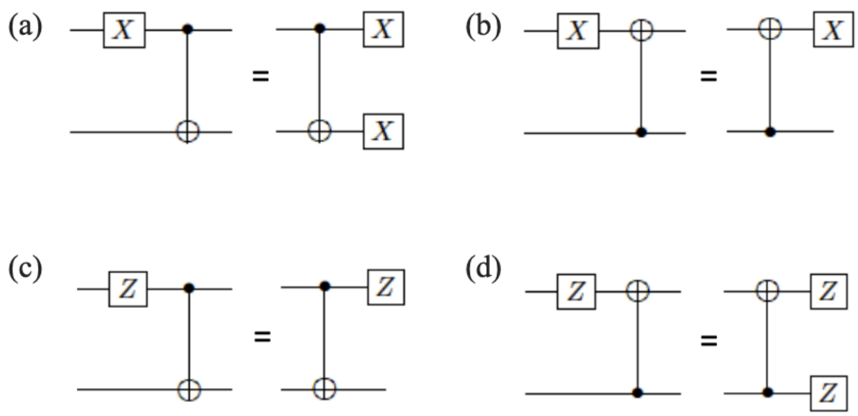

Fig. 3. Propagation of X and Z errors through the CNOT gates.

Figure 1:

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

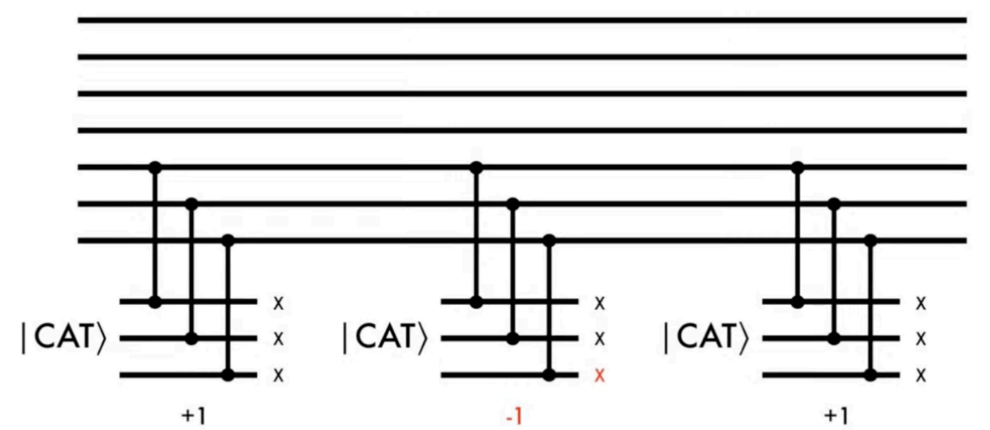

Figure 2:

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

$$\mathrm{CNOT}^{\otimes n} \;=\; \overline{\mathrm{CNOT}}^{\otimes k}$$

for an $[\![n, k, d]\!]$ CSS code. This is in fact mapping the measurement outcome from data code block to ancilla code block:
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: $|\psi\rangle|0_L\rangle \mapsto \alpha|0_L\rangle|0_L\rangle + \beta|1_L\rangle|1_L\rangle$. Measuring the ancilla block in the Z basis reveals the eigenvalue of Z_L on the data 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 ocurrs in Steane's parity checks, more parity checks outsome 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

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

. Here, d is the code distance.

For an general $[\![n,k,d]\!]$ code, it is easily to generalize the ancilla states to $|0\rangle = \overline{|+_1 ... 0_i ... +_k\rangle}$ for a Z type measurements, since $X_j|+_k\rangle$ leaves no change of the state, the measurement of Z_i will only extract information from i qubits (the state $|+_k\rangle$, which treats all Z measurement outcomes on an equal footing). $|0_i\rangle$ is just the ancilla state (measurement state) for Z_i stabilizers. Dimensions of $|0_i\rangle$ state is the weight of Z_i stabilizers. Here, we have noted that for even weight of stabilizers, they are related by local Hadamard gate, called Clifford-equivalent. Also, the choice of codewords are designed by both logical operators and stabilzers. A density matrix for logical state of one-qubit encoding can be found as [1]

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

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

Problem in Steane code could be ancilla states $|0_L\rangle$ preparation [10]. It can be comprised of a non-fault tolerant preapation process combined with a verification stage. The verification stage Fig. 3 requires and additional ancilla qubit to flag a successful preparation, like post-process. The whole process then become fault-tolerant but with successful rate e^{-np}, with n being number of gates and p the successtul probability of each gate. For state other than $|0_L\rangle$ can be prepared combined with Clifford operations. Noted that apart from Clifford operations, magic state injection ($T|+\rangle$) is also required to fulfill the universal quantum operations. Similarly, by state distillation or code concatenation, desired ancilla qubit states can be obtained but with large overheads [17].

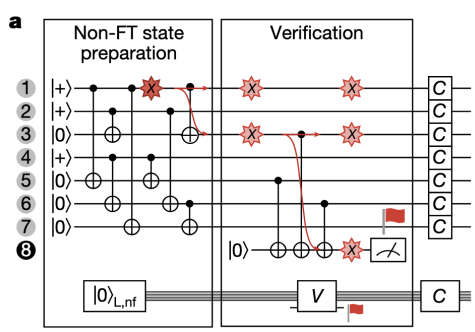

Figure 3:

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

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

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

$$|0_L\rangle = \frac{1}{2^{J/2}}\cancel{(I+\overline{Z}_L)}\prod_{j\in X\text{-type}}(I+S_j)\,|0^{\otimes n}\rangle + \xcancel{\frac{1}{2^{J/2}}(I+\overline{Z}_L)\prod_{j\in Z\text{-type}}(I+S_j)\,|0^{\otimes n}\rangle}. \tag{1}$$

Noted that Z type measurements and logical Z_L measurement act trivially on $|0^{\otimes n}\rangle$ (They are already in stabilizer group or commute with stabilizers). The formula requires projective operator which could only be done unitarily. More explicity, an arbitrary state can be written as combination of projective states with 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 Eq. 1 have implicitly selectively choose the projective states $\frac{I+S_j}{2}|0^{\otimes n}\rangle$ with some probability. For Steane code, this probability is $(\frac{1}{2})^3 = \frac{1}{8}$ for three consecutive projecting process.

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

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

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

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

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

3.4 Binary vector spaces

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

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

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

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

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

the image (column space) is

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

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

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

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

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

As an example, consider

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

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 is spanned by $(1,0,1)$ and $(0,1,1)$, giving the subspace

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

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

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

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

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

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

$$k = \dim\bigl(\ker(H_X)/\mathrm{rs}(H_Z)\bigr) = \dim\bigl(\ker(H_Z)/\mathrm{rs}(H_X)\bigr)$$

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

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

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

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

Finally, the overall *code distance* is

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

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

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

3.5 Algebraic Topology

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

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

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

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

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:

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

For example:

- 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)) .
- 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 two D^1 lines, $f : \partial D^1 = S^0 \to X^0$. X^2: add a D^2 two dimensional disk $f : S^1 \to X^1$) ($X^2 : ab^{-1}a^{-1}b$, the direction of loop are glued will result in different shape, if $X^2 : ab^{-1}ab$ is a Klein bottle). $\chi(T^2) = 0$

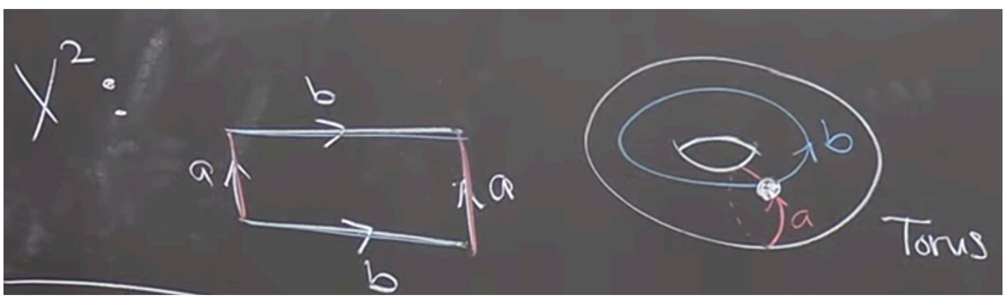

Figure 4: S^1 for torus. X^1: add

Generalization The n-dimensional disk is

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

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

Euler characteristic Vertices $D^0 = V$, Edges $D^1 = E$, Faces $D^2 = F$,...

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

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

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

Product and homology. Noted that D^n is contractible and S^n are not.

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.

The torus (solid torus: $D^2 \times S^1$) is defined as $T^2 = S^1 \times S^1$ and its *fundamental group* is $\pi_1(T^2) \cong \mathbb{Z} \times \mathbb{Z}$. 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 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 must collapse or "break" one of the gluing directions that form T^2. Since this cannot be done continuously without tearing the surface, T^2 and S^1 are not homotopy equivalent. $D^1 \times D^2$: a solid cylinder (sphere) Some identities:

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

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

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

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 different directions, pictorially, draw S^1 first for first qubit and then draw D^2 connected on S^1 similar for second torus but with opposite order.). Union can be think of gluing, hence gluing two tori is S^3.

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: wedge), also examples in Fig. 5

Homology group is used to describe

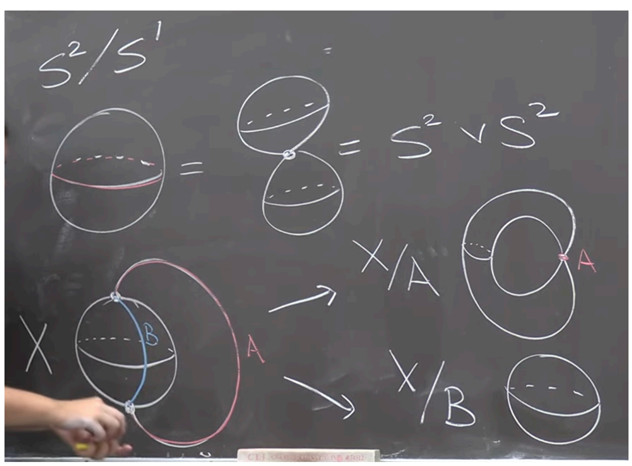

Figure 5:

3.5.1 Homology Groups

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

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

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 **homology group** is used to describe data qubits C_1 and logical operators $(\ker(\partial_1)/\mathrm{im}(\partial_2))$

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

$$H_n = Z_n / B_n \tag{2}$$
$$Z_n := \ker \partial_n := \{\, c \in C_n \mid \partial_n(c) = 0 \,\} \tag{3}$$
$$B_n := \mathrm{im}\, \partial_{n+1} := \{\, \partial_{n+1}(c) \mid c \in C_{n+1} \,\} \tag{4}$$

This correspond to $\dim(\ker(\partial_n)/\mathrm{im}(\partial_{n+1})) = \dim(\ker(H_z)/\mathrm{rs}(H_x)) = \mathrm{k}$. The algebra links with Fig. 10 toric 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 (C_1), X-stabilizers are associated with vertices (C_0), and Z-stabilizers with faces (C_2). The logical operators are characterized by the first homology group

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

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

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

corresponds to the X-type logical operators, while

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

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

3.6 Homomorphic logical measurements

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

Data–Ancilla Interaction

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

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

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

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

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

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

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

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

This ensures the interaction preserves the stabilizer groups.

Definition (Homomorphic gadget). An $[[m, k', d']]$ homomorphic gadget (H'_X, H'_Z, Γ) for an $[[n, k, d]]$ 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 $\Gamma : \mathbb{F}_2^m \to \mathbb{F}_2^n$; such that

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

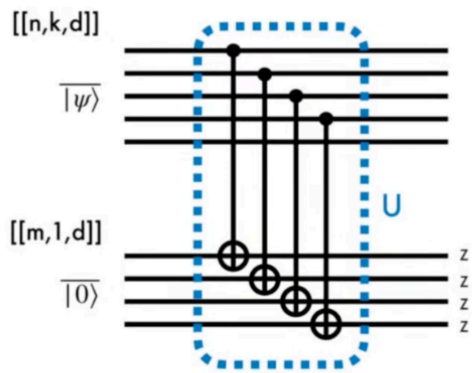

Figure 6:

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

3.7 Homomorphic measurements on surface codes

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

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

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 $\gamma_0\partial_1' = \partial_1'\gamma_1 \subseteq \partial_1$. The paper seems like weakening the global homeomorphism constraints of a usual linear map Γ (γ_i) such as Steane or Shor to local homeomorphism. This generalization gives more degree of freedom to represent logical operators to a single non-contractable loop in a new manifold. This generalization do not preserve transversal gates, as we can see that γ_i local homeomorphism, or covering spaces can be many-to-one linear maps. There are also certain boundaries for manifold M, with two rough boundaries and two smooth boundaries is the planar surface codes [4].

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

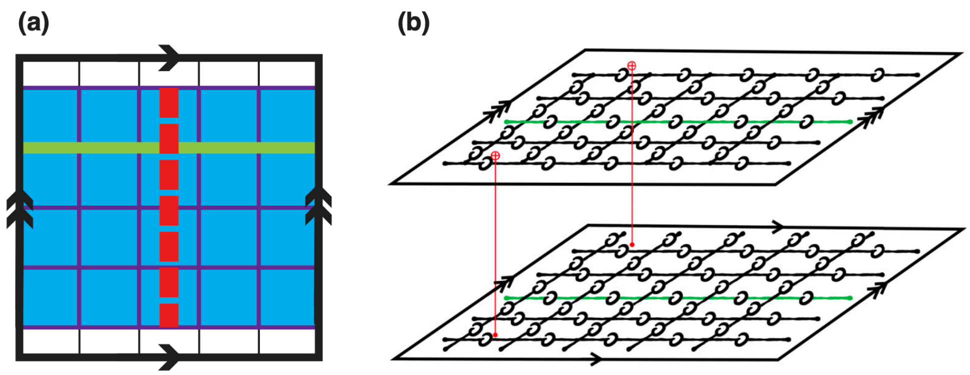

Figure 7:

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

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

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

As discussed in Sec. 3.5 and illustrated in Fig. 4, one can regard the **first example: torus** as the quotient of the real plane $\mathcal{U}=\mathbb{R}^2$ by the integer translation group $G\cong\mathbb{Z}\times\mathbb{Z}$ (translations $t_{r,s}\,(x,y):(x,y)\to(x+dr,y+ds)$ for an $[2d^2,2,d]$ toric code). Intuitively, this corresponds to identifying points that differ by integer shifts, i.e., taking 0 and 1 as the same point in each direction. The 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 a and b in Fig. 4—are glued together to form the torus topologically. Because the torus is constructed as this quotient, its *fundamental group* is isomorphic to the translation group itself,

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

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

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

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

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 point in $N_\nu : p_G(N_u)=\{\,p_G(x)\mid x\in N_u\,\}$. Therefore, if N_ν is a small open patch around u in the original space $\mathcal{U}$, then the set $N_v := p_G(N_\nu)$ is the corresponding small open patch and disjoint in the quotient space $\mathcal{U}/G$. N_ν and $g(N_u)$ are homeomorphic. Also, no nontrivial $g\,(u)=u$ (Not fixed points mean mapping all of the points to N_ν, bijective: injective and surjective)

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

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

Fourth example: [[2d^2, 2, d]] toric code. Lifting a horizontal loop l on $\mathcal{U}/G$ to $\tilde{l}$. Mathematically, 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 viewing $(0,y)$ and (d,y) as same point. Also, $\tilde{l}$ is guranteed to be a loop if and only if l is contractble on $\mathcal{U}/G$ given U is simply connected.

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

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

Fifth example: [[2d^2, 2, d]] toric code is same as the previous example for relation $g \in H \iff$ the lifted loop ℓ is closed on $\mathcal{U}/H$.

3.8 Homomorphic gadgets for covering spaces

Now we can start to construct homomorphic gadgets for covering spaces. Until now, we make some remarks: $g(u)$ 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 be possible. This coule directly be seen $p_G(u) = Gu = \{\, g(u) \mid g \in G \,\}$ while p_G represented all possible $g(u) \in \mathcal{U}$ and collapse to one point in space $\mathcal{U}/G$ by definition.
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 $d_{\mathcal{A}} = d_{\mathcal{D}}$.

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 $H = \langle g \rangle$ ($g \in G$), then all the loops are unfolded except the loop l corresponding to the g-translation.

Specifically, we map two non-contractible loops to one non-contractible loop in the ancilla block $A \subseteq \tilde{D} = \mathcal{U}/H$. This ensures that we only have one unique logical operator in $\mathcal{U}/H$, where H is chosen to be $\langle t_{1,1} \rangle$ (i.e., no overlapping qubits like in the toric code with different logicals). This unique logical operator can be designed to represent $\overline{Z_1 Z_2}$, enabling single-shot measurement. Noted that ancilla block $\mathcal{A}$ is chosen 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)

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

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

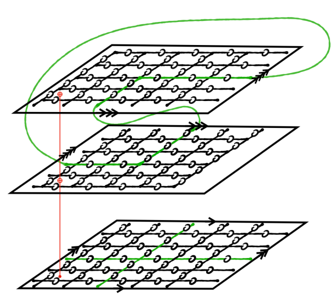

Figure 8:

3.9 Fault tolerance

Since there is no transversal mapping for $\gamma := p \circ \tilde{\gamma}$, while homeomorphism between data sheets and ancilla sheets in standard measurement method is levergaed to local homeomorphism between them. The mapping

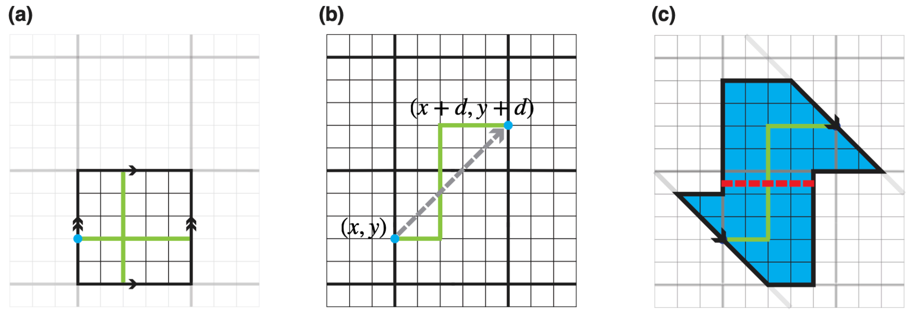

Figure 9: (a) Data-qubit manifold $\mathcal{U}/G = \mathbb{T}^2$, where the green loops represent the logical operators $\overline{Z_1 Z_2}$. (b) The covering space $\mathcal{U}$, showing the lifted path connecting (x, y) and $(x + d, y + d)$. (c) The ancilla-qubit manifold $\mathcal{U}/H$, which is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

between edges then might encounter many-to-one coupling, $\gamma_1^T(e) \in E'$. Even under these correlations, it is shown it still have fault tolerance with X error $\min\{d_{\mathcal{A}}, d_{\mathcal{D}}\}$.

3.10 Joint measurement

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

For two separate codes, say two ancilla blocks $\mathcal{A}_1$, $\mathcal{A}_2$, in order to prepare ancilla, one uses a lattice surgery approach to entangle two blocks from the initial state $|+\rangle_1 |+\rangle_2$ into the logical Bell state $|+\rangle_L = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ by measuring $Z_{\mathcal{A}_1} Z_{\mathcal{A}_2}$ with some surface code A' satisfying $\partial \mathcal{A}' = l_1' \cup l_2'$. (Note that the 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.) Just like the layer of ancilla blocks depicted in Fig. 8, $Z_{\mathcal{A}_i}$ can be a closed loop on the boundary, $l_i' \subseteq \partial \mathcal{A}_i$. After ancilla preparation, one could construct homomorphic gadget (entangle data block and ancilla block) and perform logical measurement afterwards. Ancilla states can be prepared *offline*.

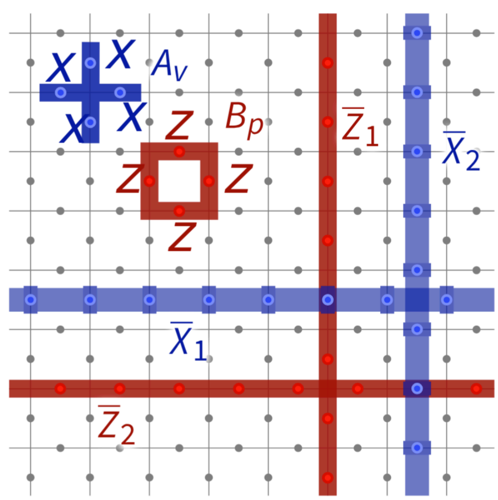

Figure 10:

4 Summary

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

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

In this framework, the Steane measurement corresponds to a homeomorphic (one-to-one) chain map, while the homomorphic logical measurement generalizes it to a *covering map* (locally bijective but globally many-to-one). This broader formulation naturally supports more general and scalable constructions of logical measurements across CSS codes.