PHYS 7810 (Fall 2026)
Quantum Error Correction

Andrew Lucas[1,*] and ChatGPT 5.6 Pro/Codex 5.6 Sol Ultra

[1]Department of Physics and Center for Theory of Quantum Matter,
University of Colorado, Boulder, CO 80309, USA

These are AI-generated transcriptions (with light human edits) of human-developed handwritten lecture notes into more readable LaTeX. These lecture notes do not contain references.

Contents

* andrew.j.lucas@colorado.edu

1 Quantum channels

1.1 How to encode classical information

Classical information is usually transmitted as a sequence of bits. The environment may corrupt some of those bits before the message reaches its destination: see Figure 1.1. The basic idea of error correction is to encode information redundantly before exposing it to noise.

Example 1.1. The simplest example is the three-bit repetition code:

$$0 \longmapsto 000, \qquad 1 \longmapsto 111. \tag{1.1}$$

Here the bit on the left is the *logical bit*, while the three bits on the right are the *physical bits*. A majority-vote decoder assigns

$$\begin{aligned} &000,\ 001,\ 010,\ 100 \longmapsto 0, \\ &111,\ 110,\ 101,\ 011 \longmapsto 1. \end{aligned} \tag{1.2}$$

Thus the desired logic is schematically

$$\text{decoder}(\text{noise}(\text{information})) = \text{information}\,. \tag{1.3}$$

Modern communication networks use more sophisticated versions of this idea (i.e. more sophisticated codes). We'll describe them later in Section **??**.

No code is perfect: some errors are decodable, while sufficiently large errors can change the logical information. For example, $0 \longmapsto 000 \longmapsto 001 \longmapsto 0$ is a decodable error, whereas $0 \longmapsto 000 \longmapsto 011 \longmapsto 1$ is an undecodable, or logical, error. For the three-bit repetition code, every single-bit flip is correctable, but two or more bit flips can make the majority vote return the wrong logical bit. In Section 6, the quality of a code will be quantified by the smallest physical error capable of producing a logical error.

1.2 Why encoding quantum information is harder

A logical qubit is a normalized superposition

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \qquad |\alpha|^2 + |\beta|^2 = 1. \tag{1.4}$$

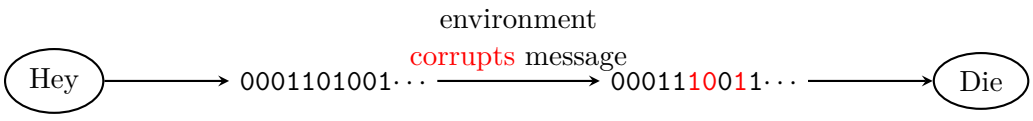

Figure 1.1: A classical "text message", which is first encoded into bits, can be altered by noise during transmission.

Unlike a classical bit, a qubit contains continuous parameters α and β. We must preserve the entire superposition, not merely distinguish the basis states $|0\rangle$ and $|1\rangle$.

1.2.1 Challenge 1: the no-cloning theorem

Theorem 1.2. There is no linear operation Q that clones every qubit according to

$$Q(\alpha|0\rangle + \beta|1\rangle) = (\alpha|0\rangle + \beta|1\rangle) \otimes (\alpha|0\rangle + \beta|1\rangle) \tag{1.5}$$

for arbitrary α and β.

Proof. If Q clones the computational-basis states, then

$$Q|0\rangle = |0\rangle \otimes |0\rangle, \qquad Q|1\rangle = |1\rangle \otimes |1\rangle. \tag{1.6}$$

Linearity would therefore imply

$$Q(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle \otimes |0\rangle + \beta|1\rangle \otimes |1\rangle. \tag{1.7}$$

By contrast, the desired cloned state expands as

$$(\alpha|0\rangle + \beta|1\rangle) \otimes (\alpha|0\rangle + \beta|1\rangle) = \alpha^2|00\rangle + \alpha\beta|01\rangle + \alpha\beta|10\rangle + \beta^2|11\rangle, \tag{1.8}$$

which is generically different from Eq. (1.7). Thus quantum redundancy cannot be created by simply copying an unknown state. □

1.2.2 Challenge 2: bit and phase errors

The Pauli operators act on the computational basis as

$$X|0\rangle = |1\rangle, \qquad X|1\rangle = |0\rangle, \qquad Z|0\rangle = |0\rangle, \qquad Z|1\rangle = -|1\rangle. \tag{1.9}$$

For three qubits, subscripts specify the qubit on which an operator acts; for example,

$$X_1 = X \otimes \mathbb{I} \otimes \mathbb{I}, \qquad Z_2 = \mathbb{I} \otimes Z \otimes \mathbb{I}. \tag{1.10}$$

Consider the encoded superposition $\alpha|000\rangle + \beta|111\rangle$. This is basically taking the three-bit repetition code and trying to make it quantum. A bit flip on the first qubit gives

$$X_1(\alpha|000\rangle + \beta|111\rangle) = \alpha|100\rangle + \beta|011\rangle, \tag{1.11}$$

which moves the state into a distinguishable error space. A phase flip gives

$$Z_1(\alpha|000\rangle + \beta|111\rangle) = \alpha|000\rangle - \beta|111\rangle, \tag{1.12}$$

which remains inside the same two-dimensional codespace and is therefore undetectable by this repetition code.

A measurement can also act as an error. If an eavesdropper measures Z_1, then

$$\alpha\,|000\rangle + \beta\,|111\rangle \xrightarrow{\text{measure } Z_1} \begin{cases} |000\rangle\,, & \text{with probability } |\alpha|^2, \\ |111\rangle\,, & \text{with probability } |\beta|^2. \end{cases} \tag{1.13}$$

The relative phase and coherent superposition have been destroyed.

1.3 Open quantum systems

To describe generic quantum error processes, we need the mathematical framework of open quantum systems. For a closed system, a pure state is represented by a vector. For a single qubit,

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}. \tag{1.14}$$

An open system is generally described by a mixed state, or density matrix,

$$\rho = a\,|0\rangle\langle 0| + b\,|1\rangle\langle 1| + c\,|0\rangle\langle 1| + \overline{c}\,|1\rangle\langle 0| = \begin{pmatrix} a & c \\ \overline{c} & b \end{pmatrix}. \tag{1.15}$$

In these lectures, we use $\overline{c}$ to denote the complex conjugate of c. A valid density matrix satisfies

$$\rho = \rho^\dagger, \qquad \operatorname{spec}(\rho) \subseteq [0,1], \qquad \operatorname{tr}(\rho) = 1. \tag{1.16}$$

Since ρ is Hermitian, it admits a spectral decomposition

$$\rho = \sum_n p_n\,|\psi_n\rangle\langle\psi_n|\,, \qquad \langle\psi_n|\psi_{n'}\rangle = \delta_{nn'}. \tag{1.17}$$

We may interpret this as being in the pure state $|\psi_n\rangle$ with probability p_n. This decomposition is not unique if one does not insist that the states in the ensemble be orthogonal.

Equivalently, one often states the density-matrix condition as $\rho \succeq 0$ together with $\operatorname{tr}\rho = 1$. Positivity and unit trace automatically place every eigenvalue in the interval $[0,1]$.

1.4 Projective measurements

Suppose an observable A has eigenvalue a, and let P_a project onto the corresponding eigenspace, so that

$$AP_a = aP_a. \tag{1.18}$$

If outcome a is observed, Born's rule gives the conditional post-measurement state

$$\rho_a = \frac{P_a \rho P_a}{\mathrm{tr}(P_a \rho P_a)} \tag{1.19}$$

with probability

$$p_a = \mathrm{tr}(P_a \rho P_a). \tag{1.20}$$

The conditional update in Eq. (1.19) is nonlinear because of its normalization. If the outcome is measured and then discarded, however, the averaged state is

$$\rho_{\text{ignore}} = \sum_a p_a \rho_a = \sum_a \mathrm{tr}(P_a \rho P_a) \frac{P_a \rho P_a}{\mathrm{tr}(P_a \rho P_a)} = \sum_a P_a \rho P_a =: \mathcal{N}_A(\rho), \tag{1.21}$$

which is linear. A useful model of noise is therefore: the environment measures A, but we never learn the outcome.

Because $P_a^2 = P_a$ and the trace is cyclic, $p_a = \mathrm{tr}(P_a \rho P_a) = \mathrm{tr}(P_a \rho)$. The form in Eq. (1.20) makes the cancellation in Eq. (1.21) especially transparent.

Example 1.3. Start from

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \qquad \rho_0 = |\psi\rangle\langle\psi| = \begin{pmatrix} |\alpha|^2 & \alpha\overline{\beta} \\ \overline{\alpha}\beta & |\beta|^2 \end{pmatrix}. \tag{1.22}$$

If the environment measures Z and discards the outcome, then

$$\mathcal{N}_Z(\rho_0) = \begin{pmatrix} |\alpha|^2 & 0 \\ 0 & |\beta|^2 \end{pmatrix}. \tag{1.23}$$

The off-diagonal coherences vanish, and the state is now mixed.

1.5 Quantum channels act on density matrices

A quantum channel is the most general physical – i.e. linear – operation that maps density matrices to density matrices:

$$\mathcal{C} : \{\text{density matrices}\} \longrightarrow \{\text{density matrices}\}. \tag{1.24}$$

It must satisfy three requirements.

1. It is trace preserving:

$$\mathrm{tr}(\mathcal{C}(\rho)) = \mathrm{tr}(\rho) = 1. \tag{1.25}$$

2. It is linear on probabilistic mixtures:

$$\mathcal{C}(p_1\rho_1 + p_2\rho_2) = p_1\mathcal{C}(\rho_1) + p_2\mathcal{C}(\rho_2). \tag{1.26}$$

This is required because a mixed state can represent classical uncertainty about which state was prepared. Also, although we introduced $\mathcal{C}$ as a map from density matrices to density matrices, linearity extends it to the full operator space. This extension is used below when $\mathcal{C}$ acts on operators such as $|\alpha\rangle\langle\alpha'|$ in (1.37).

3. It is completely positive. If the system S is entangled with an arbitrary reference system R, then

$$(\mathcal{C}_S \otimes \mathbb{I}_R)(\rho_{SR}) \succeq 0 \tag{1.27}$$

for every positive state ρ_{SR}.

The last requirement is essential because a system of interest may be entangled with degrees of freedom that are not explicitly included in its description.

Amazingly, all quantum channels (on finite-dimensional Hilbert spaces, which is basically what we'll be focusing on) can be completely classified by the following theorem.

Theorem 1.4 (Kraus representation)**.** In finite dimensions, $\mathcal{C}$ is completely positive and trace preserving if and only if it can be written as

$$\mathcal{C}(\rho) = \sum_i K_i \rho K_i^\dagger, \qquad \sum_i K_i^\dagger K_i = \mathbb{I}. \tag{1.28}$$

The operators K_i are called Kraus operators.

Proof. First suppose Eq. (1.28) holds. Trace preservation follows from cyclicity of the trace:

$$\mathrm{tr}(\mathcal{C}(\rho)) = \mathrm{tr}\left(\rho \sum_i K_i^\dagger K_i\right) = \mathrm{tr}(\rho\mathbb{I}) = 1. \tag{1.29}$$

For any positive ρ_{SR} and any vector $|\Psi_{SR}\rangle$,

$$\begin{aligned} \langle\Psi_{SR}|\,(\mathcal{C}_S \otimes \mathbb{I}_R)(\rho_{SR})\,|\Psi_{SR}\rangle &= \sum_i \langle\Psi_{SR}|\,(K_i \otimes \mathbb{I}_R)\rho_{SR}(K_i \otimes \mathbb{I}_R)^\dagger\,|\Psi_{SR}\rangle \\ &= \sum_i \langle\Psi_i|\,\rho_{SR}\,|\Psi_i\rangle \geq 0, \qquad |\Psi_i\rangle := (K_i^\dagger \otimes \mathbb{I}_R)\,|\Psi_{SR}\rangle\,. \end{aligned} \tag{1.30}$$

Thus $\mathcal{C}$ is completely positive.

Conversely, suppose $\mathcal{C}$ is CPTP. Take $\dim R = \dim S$ and choose bases $\{|\alpha\rangle_S\}$ and $\{|\alpha\rangle_R\}$.

Define the Bell-like vector

$$|\Omega\rangle = \sum_{\alpha} |\alpha\rangle_S \otimes |\alpha\rangle_R \,, \qquad \rho_\Omega = |\Omega\rangle\langle\Omega| \,. \tag{1.31}$$

Complete positivity implies that

$$\sigma_{SR} := (\mathcal{C}_S \otimes \mathbb{I}_R)(\rho_\Omega) \succeq 0. \tag{1.32}$$

Therefore the positive operator σ_{SR} can be decomposed as

$$\sigma_{SR} = \sum_i |\sigma_i\rangle\langle\sigma_i| \,. \tag{1.33}$$

Expanding ρ_Ω in the chosen basis gives

$$\sigma_{SR} = \sum_{\alpha,\alpha'} \mathcal{C}(|\alpha\rangle\langle\alpha'|) \otimes |\alpha\rangle\langle\alpha'| \,. \tag{1.34}$$

For an arbitrary $|\psi\rangle = \sum_\alpha \psi_\alpha |\alpha\rangle$, define its basis-wise conjugate in R by

$$\left|\overline{\psi}\right\rangle_R := \sum_\alpha \overline{\psi_\alpha} |\alpha\rangle_R \,, \tag{1.35}$$

and define

$$|k_i(\psi)\rangle_S := (\mathbb{I}_S \otimes \left\langle\overline{\psi}\right|_R) |\sigma_i\rangle_{SR} \,. \tag{1.36}$$

Then Eqs. (1.33) and (1.34) imply

$$\sum_i |k_i(\psi)\rangle\langle k_i(\psi)| = (\mathbb{I}_S \otimes \left\langle\overline{\psi}\right|_R)\sigma_{SR}(\mathbb{I}_S \otimes \left|\overline{\psi}\right\rangle_R) = \sum_{\alpha,\alpha'} \psi_\alpha \overline{\psi_{\alpha'}} \mathcal{C}(|\alpha\rangle\langle\alpha'|) = \mathcal{C}(|\psi\rangle\langle\psi|). \tag{1.37}$$

The conjugation in Eq. (1.35) makes $|k_i(\psi)\rangle$ a linear function of $|\psi\rangle$. Hence there is an operator K_i such that

$$|k_i(\psi)\rangle = K_i |\psi\rangle \,. \tag{1.38}$$

Equation (1.37) therefore yields

$$\mathcal{C}(|\psi\rangle\langle\psi|) = \sum_i K_i |\psi\rangle\langle\psi| K_i^\dagger. \tag{1.39}$$

By linearity, this extends to every density matrix:

$$\mathcal{C}(\rho) = \sum_i K_i \rho K_i^\dagger. \tag{1.40}$$

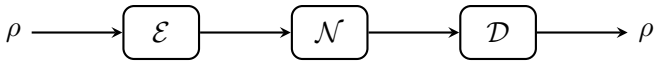

Figure 1.2: The encoder, noise, and decoder are all quantum channels. Remember that the order of operations in the quantum channels in (1.43) is read from right to left!

Finally, trace preservation implies

$$\sum_i K_i^\dagger K_i = \mathbb{I}, \tag{1.41}$$

as in the forward direction. □

The operator σ_{SR} in Eq. (1.32) is the Choi matrix of the channel, up to the normalization convention for $|\Omega\rangle$. The proof is constructive: a decomposition of the Choi matrix into rank-one positive terms directly produces a set of Kraus operators.

1.6 Looking forward: the goal of quantum error correction

Let ρ denote the logical quantum data. An encoder $\mathcal{E}$ maps it into a larger physical Hilbert space, a noise channel $\mathcal{N}$ acts, and a decoder $\mathcal{D}$ attempts to recover the logical state. The goal is

$$\mathcal{D}(\mathcal{N}(\mathcal{E}(\rho))) = \rho. \tag{1.42}$$

Equivalently, for every noise channel in the chosen correctable family,

$$\mathcal{D} \circ \mathcal{N} \circ \mathcal{E} = \mathcal{I} \tag{1.43}$$

on the logical state space: pictorially, see Figure 1.2.

The essence of QEC can be foreshadowed in three steps.

1. **Encode logical information in a subspace.** For the three-qubit repetition code, the encoder $\mathcal{E}$ acts as

$$\mathcal{E} : \alpha\,|0\rangle + \beta\,|1\rangle \longmapsto \alpha\,|000\rangle + \beta\,|111\rangle\,, \tag{1.44}$$

so the codespace is

$$\mathcal{H}_{\text{code}} = \text{span}\{|000\rangle\,, |111\rangle\}. \tag{1.45}$$

Single bit flips move the two logical basis states into distinct basis states:

$$\begin{aligned} |000\rangle &\xrightarrow{X_1,X_2,X_3} \{|100\rangle\,, |010\rangle\,, |001\rangle\}, \\ |111\rangle &\xrightarrow{X_1,X_2,X_3} \{|011\rangle\,, |101\rangle\,, |110\rangle\}. \end{aligned} \tag{1.46}$$

Entanglement is needed in order to preserve a generic superposition while distributing the logical information among several physical qubits. For generic nonzero α and β, the encoded state in Eq. (1.44) is entangled. The three-qubit repetition code nevertheless corrects an arbitrary single-qubit X error but no Z error; later codes combine complementary checks so that both kinds of error become detectable.

2. **Correctable noise $\mathcal{N}$ moves the state into distinguishable error spaces.** The error diagnosis should identify the occupied error space without revealing the logical state.

3. **Decode.** The decoder $\mathcal{D}$ infers the error and reverses it, returning the state to the codespace and recovering the logical data.

2 Shor code

Recall that the goal of quantum error correction is

$$(\mathcal{D} \circ \mathcal{N} \circ \mathcal{E})(\rho) = \rho, \tag{2.1}$$

where $\mathcal{E}$ encodes the logical qubit, $\mathcal{N}$ describes the noise or error, and $\mathcal{D}$ decodes the state. It is often useful to work entirely within the physical Hilbert space. If

$$\rho_L = \mathcal{E}(\rho) \in \mathcal{C} \tag{2.2}$$

is an encoded state in the codespace, then a recovery channel $\mathcal{R}$ should obey

$$(\mathcal{R} \circ \mathcal{N})(\rho_L) = \rho_L. \tag{2.3}$$

We will study the simplest quantum code that detects and corrects single-qubit Pauli X- and Z-type errors.

2.1 The three-qubit repetition code for X errors

For three physical qubits, the Hilbert space is

$$\mathcal{H} = (\mathbb{C}^2)^{\otimes 3} = \text{span}\{|000\rangle, |001\rangle, \ldots, |111\rangle\}. \tag{2.4}$$

2.1.1 The codespace and error syndromes

The repetition-code subspace is

$$\mathcal{C} = \text{span}\{|000\rangle, |111\rangle\}, \qquad |0_L\rangle := |000\rangle, \qquad |1_L\rangle := |111\rangle. \tag{2.5}$$

Thus a generic encoded qubit is

$$|\psi_L\rangle = \alpha |000\rangle + \beta |111\rangle. \tag{2.6}$$

A single Pauli X error acts as

$$X_1 |\psi_L\rangle = \alpha |100\rangle + \beta |011\rangle \in \mathcal{B}_1, \tag{2.7a}$$
$$X_2 |\psi_L\rangle = \alpha |010\rangle + \beta |101\rangle \in \mathcal{B}_2, \tag{2.7b}$$
$$X_3 |\psi_L\rangle = \alpha |001\rangle + \beta |110\rangle \in \mathcal{B}_3. \tag{2.7c}$$

The corresponding orthogonal sectors are

$$\mathcal{B}_1 = \text{span}\{|100\rangle, |011\rangle\}, \tag{2.8a}$$

$$\mathcal{B}_2 = \mathrm{span}\{|010\rangle, |101\rangle\}, \tag{2.8b}$$
$$\mathcal{B}_3 = \mathrm{span}\{|001\rangle, |110\rangle\}, \tag{2.8c}$$

and therefore

$$\mathcal{H} = \mathcal{C} \oplus \mathcal{B}_1 \oplus \mathcal{B}_2 \oplus \mathcal{B}_3. \tag{2.9}$$

The decoding strategy is to determine which of these four sectors contains the state and then apply X_j if the state lies in $\mathcal{B}_j$. Crucially, measuring the sector does not determine α or β.

Let $P_{\mathcal{C}}$ denote the projector onto the codespace. On computational-basis states,

$$P_{\mathcal{C}} |x_1 x_2 x_3\rangle = \begin{cases} |x_1 x_2 x_3\rangle, & x_1 = x_2 = x_3, \\ 0, & \text{otherwise.} \end{cases} \tag{2.10}$$

Let $P_{\mathcal{B}_j}$ be the projectors onto the three error sectors, defined analogously. The recovery channel is

$$\mathcal{R}(\rho) = P_{\mathcal{C}} \rho P_{\mathcal{C}} + \sum_{j=1}^{3} X_j P_{\mathcal{B}_j} \rho P_{\mathcal{B}_j} X_j. \tag{2.11}$$

This is a valid quantum channel. Indeed, it has Kraus operators

$$K_0 = P_{\mathcal{C}}, \qquad K_j = X_j P_{\mathcal{B}_j} \quad (j = 1, 2, 3), \qquad \mathcal{R}(\rho) = \sum_{j=0}^{3} K_j \rho K_j^\dagger, \tag{2.12}$$

and they satisfy

$$\sum_{j=0}^{3} K_j^\dagger K_j = P_{\mathcal{C}}^2 + \sum_{j=1}^{3} P_{\mathcal{B}_j} X_j^2 P_{\mathcal{B}_j} = P_{\mathcal{C}} + \sum_{j=1}^{3} P_{\mathcal{B}_j} = \mathbb{I}. \tag{2.13}$$

2.1.2 Stabilizer measurements and the error syndrome

Definition 2.1. A *stabilizer* is an observable S that obeys $S|\psi\rangle = |\psi\rangle$ for every $|\psi\rangle \in \mathcal{C}$. We will usually specify a small set of independent stabilizer generators. We will revisit this concept more seriously in Section 7.

For the three-qubit repetition code, take

$$S_1 = Z_1 Z_2, \qquad S_2 = Z_2 Z_3. \tag{2.14}$$

Both logical basis states have eigenvalue $+1$ under both operators:

$$\begin{aligned} S_1 |000\rangle &= (+1)^2 |000\rangle = |000\rangle, & S_2 |000\rangle &= |000\rangle, \\ S_1 |111\rangle &= (-1)^2 |111\rangle = |111\rangle, & S_2 |111\rangle &= |111\rangle. \end{aligned} \tag{2.15}$$

Checking the remaining computational-basis states gives

sector	S_1	S_2	correction
$\mathcal{C}$	$+1$	$+1$	$\mathbb{I}$
$\mathcal{B}_1$	-1	$+1$	X_1
$\mathcal{B}_2$	-1	-1	X_2
$\mathcal{B}_3$	$+1$	-1	X_3

. (2.16)

The two stabilizer outcomes identify the error sector without resolving the logical state. Their ordered pair is called the *syndrome*. In particular, the codespace projector can be written as

$$P_{\mathcal{C}} = \frac{\mathbb{I} + S_1}{2} \frac{\mathbb{I} + S_2}{2}. \tag{2.17}$$

Thus measuring the stabilizers implements the sector measurement needed by the recovery channel.

2.1.3 Logical operators and detectable errors

Logical Pauli operators may be chosen as

$$X_L = X_1 X_2 X_3, \qquad Z_L = Z_1. \tag{2.18}$$

Their action on the encoded state is

$$X_L \left|\psi_L\right\rangle = \beta \left|000\right\rangle + \alpha \left|111\right\rangle, \tag{2.19a}$$

$$Z_L \left|\psi_L\right\rangle = \alpha \left|000\right\rangle - \beta \left|111\right\rangle. \tag{2.19b}$$

The first swaps $|0_L\rangle$ and $|1_L\rangle$, while the second applies a phase to $|1_L\rangle$. We must not measure X_L or Z_L, because doing so would measure and generally destroy the encoded quantum information.

The key property is that the logical operators commute with the stabilizers:

$$[X_L, S_j] = [Z_L, S_j] = 0, \qquad j = 1, 2. \tag{2.20}$$

For example,

$$X_L S_1 = X_1 X_2 X_3 Z_1 Z_2 = (XZ) \otimes (XZ) \otimes X = (-ZX) \otimes (-ZX) \otimes X = S_1 X_L. \tag{2.21}$$

Here we used the single-qubit anticommutation relation

$$XZ = -ZX. \tag{2.22}$$

It can be checked directly on the computational basis:

$$\begin{aligned} XZ\,|0\rangle &= X\,|0\rangle = |1\rangle\,, & ZX\,|0\rangle &= Z\,|1\rangle = -\,|1\rangle\,, \\ XZ\,|1\rangle &= -X\,|1\rangle = -\,|0\rangle\,, & ZX\,|1\rangle &= Z\,|0\rangle = |0\rangle\,. \end{aligned} \tag{2.23}$$

An error need not commute with the logical operators. For example,

$$[X_1, Z_L] \neq 0. \tag{2.24}$$

It is detectable because it fails to commute with at least one stabilizer.

Proposition 2.2. Suppose an error E obeys

$$ES_j = \eta_j S_j E, \qquad \eta_j \in \{+1, -1\}. \tag{2.25}$$

Then, on any codeword, measurement of S_j returns the deterministic outcome η_j.

Proof. For $|\psi\rangle \in \mathcal{C}$, we have $S_j\,|\psi\rangle = |\psi\rangle$. Therefore

$$S_j E\,|\psi\rangle = \eta_j E S_j\,|\psi\rangle = \eta_j E\,|\psi\rangle\,. \tag{2.26}$$

Thus the errored state is an eigenstate of S_j with eigenvalue η_j. □

For example,

$$X_1 S_1 = -S_1 X_1, \qquad X_1 S_2 = S_2 X_1, \tag{2.27}$$

so an X_1 error has syndrome $(-1, +1)$. This matches what we saw earlier in (2.16). Linearity makes the result valid for every superposition in the codespace.

2.2 The three-qubit repetition code for Z errors

A complementary repetition code detects phase errors rather than bit flips. Its stabilizers and logical operators are

$$S_1 = X_1 X_2, \qquad S_2 = X_2 X_3, \qquad X_L = X_1, \qquad Z_L = Z_1 Z_2 Z_3. \tag{2.28}$$

The roles of X and Z can be exchanged by the Hadamard gate,

$$H = \frac{X + Z}{\sqrt{2}}, \qquad H^\dagger X H = Z, \qquad H^\dagger Z H = X. \tag{2.29}$$

Thus applying $H^{\otimes 3}$ maps the bit-flip repetition code to the phase-flip repetition code. We choose the logical basis

$$|0_L\rangle = \frac{1}{2}(|000\rangle + |011\rangle + |101\rangle + |110\rangle), \tag{2.30a}$$

$$|1_L\rangle = \frac{1}{2}(|100\rangle + |010\rangle + |001\rangle + |111\rangle). \tag{2.30b}$$

This convention also includes a logical Hadamard. In particular,

$$H^{\otimes 3}|000\rangle = \frac{|0_L\rangle + |1_L\rangle}{\sqrt{2}}. \tag{2.31}$$

For example, the stabilizer and logical actions are

$$S_1|0_L\rangle = \frac{1}{2}(|110\rangle + |101\rangle + |011\rangle + |000\rangle) = |0_L\rangle, \tag{2.32a}$$

$$X_L|0_L\rangle = \frac{1}{2}(|100\rangle + |111\rangle + |001\rangle + |010\rangle) = |1_L\rangle. \tag{2.32b}$$

A phase error produces a detectable pattern of relative signs:

$$Z_1|0_L\rangle = \frac{1}{2}(|000\rangle + |011\rangle - |101\rangle - |110\rangle). \tag{2.33}$$

Indeed, the errored state is orthogonal to the original codeword:

$$\langle 0_L| Z_1 |0_L\rangle = \frac{1}{4}(1 + 1 - 1 - 1) = 0. \tag{2.34}$$

Equivalently, the anticommutation of Z_1 with S_1 gives a nontrivial syndrome.

2.3 Code concatenation and the nine-qubit Shor code

We now combine the two repetition codes to obtain a genuinely quantum code that protects against both X- and Z-type errors. Begin with the phase-flip codeword written as three outer qubits,

$$|0_L\rangle_{\text{outer}} = \frac{1}{2}(|0\rangle\otimes|0\rangle\otimes|0\rangle + |0\rangle\otimes|1\rangle\otimes|1\rangle + |1\rangle\otimes|0\rangle\otimes|1\rangle + |1\rangle\otimes|1\rangle\otimes|0\rangle). \tag{2.35}$$

Replace each outer qubit by a logical qubit of the bit-flip repetition code:

$$|0\rangle \longmapsto |000\rangle, \qquad |1\rangle \longmapsto |111\rangle. \tag{2.36}$$

The result is the nine-qubit Shor code,

$$|0_L\rangle = \frac{1}{2}(|000\,000\,000\rangle + |000\,111\,111\rangle + |111\,000\,111\rangle + |111\,111\,000\rangle), \tag{2.37a}$$

$$|1_L\rangle = \frac{1}{2}(|111\,000\,000\rangle + |000\,111\,000\rangle + |000\,000\,111\rangle + |111\,111\,111\rangle). \tag{2.37b}$$

Writing out the codewords becomes cumbersome very quickly. The stabilizer description is much more economical.

2.3.1 Stabilizers of the Shor code

The two outer phase-flip stabilizers become products of logical X operators on neighboring three-qubit blocks, while each inner block retains its two Z-type stabilizers. A set of eight stabilizer generators is

$$\begin{aligned} S_1 &= X_1X_2X_3X_4X_5X_6, & S_2 &= X_4X_5X_6X_7X_8X_9, \\ S_3 &= Z_1Z_2, & S_4 &= Z_2Z_3, \\ S_5 &= Z_4Z_5, & S_6 &= Z_5Z_6, \\ S_7 &= Z_7Z_8, & S_8 &= Z_8Z_9. \end{aligned} \tag{2.38}$$

The logical operators transform to

$$X_L = X_1X_2X_3, \qquad Z_L = Z_1Z_4Z_7. \tag{2.39}$$

They obey

$$[X_L, S_j] = [Z_L, S_j] = 0, \qquad [S_j, S_{j'}] = 0, \qquad j, j' = 1, \ldots, 8. \tag{2.40}$$

For every pair consisting of an X-type generator and a Z-type generator, the supports overlap on either zero or two qubits. Each overlap contributes one minus sign when the operators are exchanged, so the total sign is always $+1$. This is a general property we return to in Section 7.

Because all eight stabilizers commute, they can be measured simultaneously to diagnose errors without measuring the logical operators. Again, we'll revisit this more carefully in Section 7.

2.3.2 Correcting bit-flip errors

To detect X errors, measure the six inner-block stabilizers

$$Z_1Z_2, \quad Z_2Z_3, \quad Z_4Z_5, \quad Z_5Z_6, \quad Z_7Z_8, \quad Z_8Z_9. \tag{2.41}$$

Within each block, the syndrome is exactly the one in Eq. (2.16); hence one can detect and correct at most one X error in each block. For example, the computational-basis state $|001\,000\,010\rangle$ has

$$\begin{aligned} (Z_2Z_3)\,|001\,000\,010\rangle &= -\,|001\,000\,010\rangle\,, \\ (Z_7Z_8)\,|001\,000\,010\rangle &= -\,|001\,000\,010\rangle\,, \\ (Z_8Z_9)\,|001\,000\,010\rangle &= -\,|001\,000\,010\rangle\,. \end{aligned} \tag{2.42}$$

All other Z-stabilizers have $+1$ eigenvalues in this state. The first syndrome locates the flip on qubit 3, while the last two locate the flip on qubit 8.

We must stress that (2.42) is not analyzing the syndrome on a codeword. It would be rather tedious to write out such a calculation for each term in the codeword's wave function – this is why the stabilizer framework will be critical to us as we develop the formalism of quantum error correction!

2.3.3 Correcting phase-flip errors

To detect a Z error, measure the two outer stabilizers S_1 and S_2. For a phase error in the first block,

$$Z_j S_1 = -S_1 Z_j, \qquad Z_j S_2 = S_2 Z_j, \qquad j = 1, 2, 3. \tag{2.43}$$

The syndrome identifies the block but not the particular qubit within that block. This is sufficient because the three errors act identically on the codespace. For example, if $|\psi\rangle \in \mathcal{C}$, then

$$Z_1 |\psi\rangle = Z_1 (Z_1 Z_2) |\psi\rangle = Z_2 |\psi\rangle . \tag{2.44}$$

The same reasoning gives

$$Z_1 |\psi\rangle = Z_2 |\psi\rangle = Z_3 |\psi\rangle , \qquad Z_4 |\psi\rangle = Z_5 |\psi\rangle = Z_6 |\psi\rangle , \qquad Z_7 |\psi\rangle = Z_8 |\psi\rangle = Z_9 |\psi\rangle . \tag{2.45}$$

More generally, if S stabilizes the codespace, then E and ES have the same action on every codeword. Errors related in this way are called *stabilizer-equivalent*; a syndrome need only identify their equivalence class. It is enough to use Z_1, Z_4, and Z_7 as the three representative phase-error corrections. Their syndromes are

representative error	Z_1	Z_4	Z_7
(S_1, S_2)	$(-1, +1)$	$(-1, -1)$	$(+1, -1)$

. (2.46)

2.3.4 Combined bit-flip and phase-flip errors

Finally,

$$Y_j = -\mathrm{i} Z_j X_j . \tag{2.47}$$

A Y_j error therefore produces both the X_j-type and Z_j-type syndrome information. Detecting and correcting those two components separately corrects the Y_j error as well.

3 Knill-Laflamme conditions

We have seen a specific example of a quantum code. We now develop a general, abstract theory of quantum error correction. The abstraction is worthwhile: relatively simple conditions will guarantee protection against all quantum errors in a prescribed error space.

3.1 Abstract setup

Let $\mathcal{H}$ be a discrete Hilbert space, let $\mathcal{C} \subseteq \mathcal{H}$ be the codespace that we wish to protect, and let P denote the orthogonal projector onto $\mathcal{C}$. Thus a state supported in the codespace obeys

$$\rho = P\rho P. \tag{3.1}$$

The goal is to find a recovery channel $\mathcal{R}$ such that

$$(\mathcal{R} \circ \mathcal{N})(\rho) = \rho \tag{3.2}$$

for every channel of the form

$$\mathcal{N}(\rho) = \sum_i K_i \rho K_i^\dagger, \qquad K_i \in \text{span}\{E_a\}. \tag{3.3}$$

The important point is that the Kraus operators need not themselves be chosen from a discrete list. For example, the error space may contain both X_j and Z_ℓ, and hence also every linear combination (e.g. $\alpha X_j + \beta Z_\ell$).

Theorem 3.1 (Knill–Laflamme conditions)**.** Let $\{E_a\}$ be a basis for a linear space of errors. This error space is correctable on $\mathcal{C}$ if and only if there is a Hermitian matrix α such that

$$PE_a^\dagger E_b P = \alpha_{ab} P \qquad \text{for every } a, b. \tag{3.4}$$

In particular,

$$\alpha_{ab} = \overline{\alpha_{ba}}. \tag{3.5}$$

The rest of the lecture proves this theorem and pauses along the way to explain its physical content. First, we may change the basis of the error space by diagonalizing α, and then rescale every basis vector corresponding to a nonzero eigenvalue. We therefore split the resulting basis into operators $E_{\widetilde{a}}$ and $E_{\bar{a}}$ such that

$$PE_{\widetilde{a}}^\dagger E_{\widetilde{b}} P = \delta_{\widetilde{a}\widetilde{b}} P, \tag{3.6a}$$

$$PE_{\bar{a}}^\dagger E_b P = 0 \qquad \text{for every } b. \tag{3.6b}$$

The tilded indices label the nonzero block of α, while the barred indices label its zero block. Some

barred errors may annihilate the codespace; we return to them below.

The rescaling above is well defined because α is automatically positive semidefinite. Indeed, for $F = \sum_a v_a E_a$, the Knill–Laflamme condition gives

$$PF^\dagger FP = \left(\sum_{a,b} \overline{v_a} \alpha_{ab} v_b \right) P \geq 0. \tag{3.7}$$

Thus the nonzero eigenvalues of α are positive and can be normalized to one.

3.2 Sufficiency: constructing a recovery channel

The first part of the proof of Theorem 3.1 shows that (3.4) is sufficient to construct $\mathcal{R}$.

3.2.1 Orthogonal error sectors

Lemma 3.2. For each normalized error $E_{\widetilde{a}}$, there is a unitary $U_{\widetilde{a}}$ such that

$$E_{\widetilde{a}} P = U_{\widetilde{a}} P. \tag{3.8}$$

If

$$P_{\widetilde{a}} := U_{\widetilde{a}} P U_{\widetilde{a}}^\dagger, \tag{3.9}$$

then the rotated codespaces are mutually orthogonal:

$$P_{\widetilde{b}} P_{\widetilde{a}} = \delta_{\widetilde{a}\widetilde{b}} P_{\widetilde{a}}. \tag{3.10}$$

Proof. For $|\psi\rangle, |\phi\rangle \in \mathcal{C}$, Eq. (3.6a) implies

$$\langle \psi | \phi \rangle = \langle \psi | P | \phi \rangle = \langle \psi | P E_{\widetilde{a}}^\dagger E_{\widetilde{a}} P | \phi \rangle = \langle E_{\widetilde{a}} \psi | E_{\widetilde{a}} \phi \rangle . \tag{3.11}$$

Hence $E_{\widetilde{a}} P$ is an isometry on the codespace. Extending this isometry to the rest of the finite-dimensional Hilbert space gives a unitary $U_{\widetilde{a}}$ obeying Eq. (3.8).

Using both Eq. (3.8) and its adjoint, we find

$$P_{\widetilde{b}} P_{\widetilde{a}} = U_{\widetilde{b}} P U_{\widetilde{b}}^\dagger U_{\widetilde{a}} P U_{\widetilde{a}}^\dagger = U_{\widetilde{b}} (P E_{\widetilde{b}}^\dagger E_{\widetilde{a}} P) U_{\widetilde{a}}^\dagger = \delta_{\widetilde{a}\widetilde{b}} U_{\widetilde{a}} P U_{\widetilde{a}}^\dagger = \delta_{\widetilde{a}\widetilde{b}} P_{\widetilde{a}}, \tag{3.12}$$

which gives (3.10). □

We include the no-error operator and choose

$$E_{\widetilde{0}} = \mathbb{I}, \qquad U_{\widetilde{0}} = \mathbb{I}, \qquad P_{\widetilde{0}} = P. \tag{3.13}$$

Each detectable error therefore maps the codespace to a different orthogonal syndrome sector,

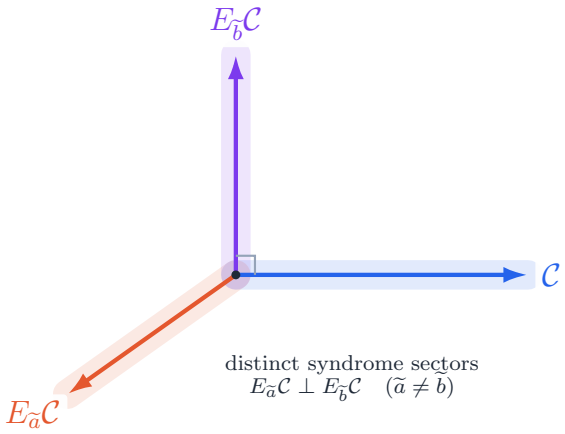

Figure 3.1: Each distinct detectable error $E_{\widetilde{a}}$ takes the codespace $\mathcal{C}$ into an orthogonal subspace.

exactly as in the Shor code (Section 2.3). As illustrated in Figure 3.1, each class of correctable error $E_{\widetilde{a}}$ kicks us into orthogonal subspaces $E_{\widetilde{a}}\mathcal{C}$; projector $P_{\widetilde{a}}$ is a projector onto only this subspace.

3.2.2 The decoder

Define the projector onto all states not reached by the normalized error operators:

$$P_\perp := \mathbb{I} - \sum_{\widetilde{a}} P_{\widetilde{a}}. \tag{3.14}$$

The recovery channel is

$$\mathcal{R}(\rho) = \sum_{\widetilde{a}} U_{\widetilde{a}}^{-1} P_{\widetilde{a}} \rho P_{\widetilde{a}} U_{\widetilde{a}} + P_\perp \rho P_\perp. \tag{3.15}$$

Operationally, the projectors measure which syndrome sector $\widetilde{a}$ is occupied, and $U_{\widetilde{a}}^{-1}$ rotates that sector back to the codespace. The last term specifies an arbitrary harmless action on states that the correctable errors cannot reach.

The Kraus operators of Eq. (3.15) are

$$L_{\widetilde{a}} = U_{\widetilde{a}}^{-1} P_{\widetilde{a}}, \qquad L_\perp = P_\perp. \tag{3.16}$$

They satisfy

$$\sum_{\widetilde{a}} L_{\widetilde{a}}^\dagger L_{\widetilde{a}} + L_\perp^\dagger L_\perp = \sum_{\widetilde{a}} P_{\widetilde{a}} U_{\widetilde{a}} U_{\widetilde{a}}^{-1} P_{\widetilde{a}} + P_\perp = \sum_{\widetilde{a}} P_{\widetilde{a}} + P_\perp = \mathbb{I}, \tag{3.17}$$

so $\mathcal{R}$ is a quantum channel.

Lemma 3.3 (Correction of the normalized basis errors)**.** For every $\widetilde{a}$ and every $\rho = P\rho P$,

$$\mathcal{R}(E_{\widetilde{a}}\rho E_{\widetilde{a}}^{\dagger}) = \rho. \tag{3.18}$$

Proof. The corrupted state must begin in the codespace. For every $\widetilde{b}$,

$$U_{\widetilde{b}}^{-1}P_{\widetilde{b}}E_{\widetilde{a}}P = PU_{\widetilde{b}}^{\dagger}E_{\widetilde{a}}P = PE_{\widetilde{b}}^{\dagger}E_{\widetilde{a}}P = \delta_{\widetilde{a}\widetilde{b}}P. \tag{3.19}$$

The $P_{\perp}$ term vanishes, and therefore

$$\mathcal{R}(E_{\widetilde{a}}P\rho PE_{\widetilde{a}}^{\dagger}) = \sum_{\widetilde{b}} U_{\widetilde{b}}^{-1}P_{\widetilde{b}}E_{\widetilde{a}}P\rho PE_{\widetilde{a}}^{\dagger}P_{\widetilde{b}}U_{\widetilde{b}} = \sum_{\widetilde{b}} \delta_{\widetilde{a}\widetilde{b}}P\rho P = P\rho P = \rho, \tag{3.20}$$

which gives (3.18). □

3.2.3 Arbitrary channels in the error span

Lemma 3.4 (Linearity of correctable errors)**.** The same recovery channel corrects every channel

$$\mathcal{N}(\rho) = \sum_i K_i\rho K_i^{\dagger} \tag{3.21}$$

whose Kraus operators obey

$$K_i \in \operatorname{span}\{E_{\widetilde{a}}\}. \tag{3.22}$$

Proof. By linearity, it is enough first to take one operator

$$K = \sum_{\widetilde{a}} k_{\widetilde{a}}E_{\widetilde{a}}. \tag{3.23}$$

Repeating the calculation above gives

$$\mathcal{R}(KP\rho PK^{\dagger}) = \sum_{\widetilde{b}} PE_{\widetilde{b}}^{\dagger}KP\rho PK^{\dagger}E_{\widetilde{b}}P = \sum_{\widetilde{a},\widetilde{a}',\widetilde{b}} k_{\widetilde{a}}\overline{k_{\widetilde{a}'}}\delta_{\widetilde{a}\widetilde{b}}\delta_{\widetilde{a}'\widetilde{b}}P\rho P = \left(\sum_{\widetilde{a}} |k_{\widetilde{a}}|^2\right) P\rho P. \tag{3.24}$$

Consequently, $(\mathcal{R}\circ\mathcal{N})(P\rho P)$ is proportional to $P\rho P$. Since both $\mathcal{R}$ and $\mathcal{N}$ are channels, and (3.24) applies to every K_i in (3.21), their composition is trace preserving, so the proportionality constant is one:

$$(\mathcal{R}\circ\mathcal{N})(P\rho P) = P\rho P. \tag{3.25}$$

This completes the proof. □

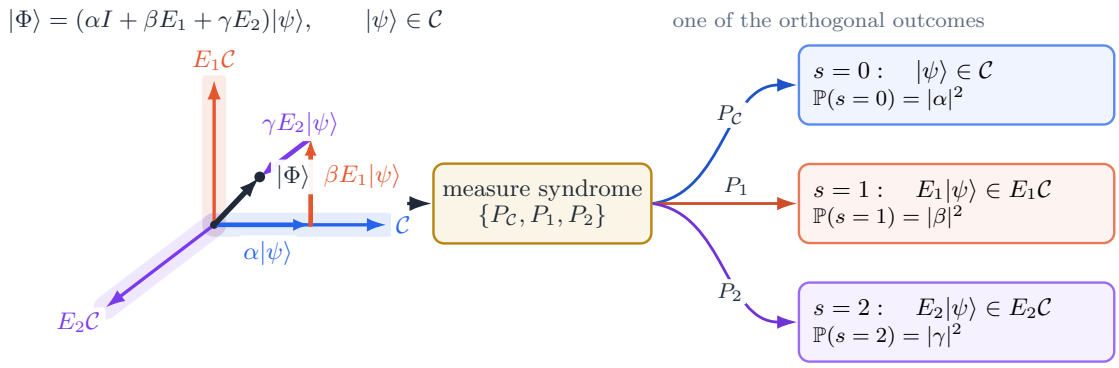

Figure 3.2: Measuring which syndrome sector we are in collapses the quantum state and "digitizes" a continuum of errors into a discrete set, which we can then correct using $\mathcal{R}$.

3.2.4 Why a continuum of errors reduces to discrete syndromes

The preceding calculation is fast, but its physical content is crucial. Consider a coherent error acting on $|\psi\rangle \in \mathcal{C}$: for example,

$$(\alpha\mathbb{I} + \beta E_{\tilde{1}} + \gamma E_{\tilde{2}})\,|\psi\rangle = \alpha\,|\psi\rangle + \beta E_{\tilde{1}}\,|\psi\rangle + \gamma E_{\tilde{2}}\,|\psi\rangle\,. \tag{3.26}$$

The three terms lie in orthogonal syndrome sectors. A syndrome measurement therefore returns

sector	probability	post-measurement state
P	$\lvert\alpha\rvert^2$	$\lvert\psi\rangle$
$P_{\tilde{1}}$	$\lvert\beta\rvert^2$	$E_{\tilde{1}}\lvert\psi\rangle$
$P_{\tilde{2}}$	$\lvert\gamma\rvert^2$	$E_{\tilde{2}}\lvert\psi\rangle$

(3.27)

for normalized coefficients. Lemma 3.3 then rotates the observed sector back to the codespace.

In this respect, quantum error correction resembles digital classical error correction: the decoder only needs to fix a discrete set of syndromes. The continuum of possible coherent errors changes the probabilities of those syndromes, rather than requiring a continuum of distinct correction operations. This is illustrated in Figure 3.2.

3.2.5 Errors that annihilate the codespace

Lemma 3.5 (Null directions in the error space)**.** The recovery above still works when the error space includes operators $E_{\bar{a}}$ obeying Eq. (3.6b).

Proof. Taking $b = \bar{a}$ in Eq. (3.6b) gives

$$0 = PE_{\bar{a}}^\dagger E_{\bar{a}} P = (E_{\bar{a}}P)^\dagger (E_{\bar{a}}P), \tag{3.28}$$

so

$$E_{\bar{a}} P = 0. \tag{3.29}$$

For a general operator in the enlarged span,

$$K = \sum_{\widetilde{a}} k_{\widetilde{a}} E_{\widetilde{a}} + \sum_{\bar{a}} k_{\bar{a}} E_{\bar{a}}, \tag{3.30}$$

we therefore have

$$KP = \left(\sum_{\widetilde{a}} k_{\widetilde{a}} E_{\widetilde{a}} \right) P. \tag{3.31}$$

The problem reduces to Lemma 3.4. □

Example 3.6. A common null error has the form

$$\widetilde{E}(S - \mathbb{I}), \tag{3.32}$$

where S is a stabilizer, because $(S - \mathbb{I})P = 0$. For example, the Shor code corrects Z_1, Z_2, Z_3, although these errors are not distinguishable within the first three-qubit block. Since $Z_1 Z_2$ is a stabilizer,

$$Z_2 = Z_1 + Z_1(Z_1 Z_2 - \mathbb{I}). \tag{3.33}$$

The second term annihilates the codespace, so Z_2 has the same action on codewords as the representative Z_1.

3.3 Necessity: exact recovery implies Knill–Laflamme

The preceding lemmas prove that the Knill–Laflamme conditions imply quantum error correction. It remains to prove the converse.

3.3.1 Finishing the proof of Theorem 3.1

Lemma 3.7 (Necessity of the Knill–Laflamme conditions)**.** Suppose a channel

$$\mathcal{N}(\rho) = \sum_i K_i \rho K_i^\dagger \tag{3.34}$$

is exactly corrected on $\mathcal{C}$ by some recovery $\widetilde{\mathcal{R}}$:

$$(\widetilde{\mathcal{R}} \circ \mathcal{N})(P\rho P) = P\rho P. \tag{3.35}$$

Then there are constants α_{ij} such that

$$PK_i^\dagger K_j P = \alpha_{ij} P. \tag{3.36}$$

Proof. By Theorem 1.4, we may write the recovery channel as

$$\widetilde{\mathcal{R}}(\sigma) = \sum_\mu D_\mu \sigma D_\mu^\dagger. \tag{3.37}$$

The exact-recovery assumption becomes

$$\sum_{\mu,i} D_\mu K_i P \rho P K_i^\dagger D_\mu^\dagger = P\rho P. \tag{3.38}$$

We first claim that

$$D_\mu K_i P = c_{\mu i} P \tag{3.39}$$

for constants $c_{\mu i}$. To see this, choose $|\psi\rangle \in \mathcal{C}$ and any $|\phi\rangle \perp |\psi\rangle$. Applying Eq. (3.38) to $\rho = |\psi\rangle\langle\psi|$ gives

$$0 = \langle\phi| \left[(\widetilde{\mathcal{R}} \circ \mathcal{N} - \mathcal{I})(|\psi\rangle\langle\psi|)\right] |\phi\rangle = \sum_{\mu,i} |\langle\phi| D_\mu K_i |\psi\rangle|^2 - |\langle\phi|\psi\rangle|^2 = \sum_{\mu,i} |\langle\phi| D_\mu K_i |\psi\rangle|^2 . \tag{3.40}$$

Every nonnegative summand must vanish. Hence $D_\mu K_i |\psi\rangle$ is parallel to $|\psi\rangle$ for every codeword $|\psi\rangle$.

The proportionality factor cannot depend on the codeword. If $A = D_\mu K_i$ obeys $A|\psi\rangle = c_\psi |\psi\rangle$ and $A|\chi\rangle = c_\chi |\chi\rangle$ for two linearly independent codewords, then applying the same statement to their sum gives

$$c_\psi |\psi\rangle + c_\chi |\chi\rangle = c_{\psi+\chi}(|\psi\rangle + |\chi\rangle), \tag{3.41}$$

which forces $c_\psi = c_\chi = c_{\psi+\chi}$. This proves Eq. (3.39).

Finally, because $\widetilde{\mathcal{R}}$ is trace preserving, Theorem 1.4 implies that

$$\sum_\mu D_\mu^\dagger D_\mu = \mathbb{I}. \tag{3.42}$$

Using Eqs. (3.39) and (3.42), we obtain

$$PK_i^\dagger K_j P = \sum_\mu PK_i^\dagger D_\mu^\dagger D_\mu K_j P = \sum_\mu \overline{c_{\mu i}} c_{\mu j} P = \alpha_{ij} P, \tag{3.43}$$

where

$$\alpha_{ij} := \sum_{\mu} \overline{c_{\mu i}} c_{\mu j}. \tag{3.44}$$

This is precisely the Knill–Laflamme condition. □

Together with the sufficiency construction, this completes the proof of Theorem 3.1.

3.3.2 Physical interpretation

The necessity proof has a simple interpretation. If there were an operator associated with the error process whose expectation value differed between two orthogonal codewords, then measuring that operator—or allowing the environment to record the same information—would distinguish the logical states. In the limiting case of perfect distinguishability, the logical superposition would collapse as

$$\alpha \left|\psi\right\rangle + \beta \left|\phi\right\rangle \longrightarrow \begin{cases} \left|\psi\right\rangle, & \text{with probability } |\alpha|^2, \\ \left|\phi\right\rangle, & \text{with probability } |\beta|^2. \end{cases} \tag{3.45}$$

No recovery operation can reconstruct the lost relative phase.

Equivalently, in any orthonormal logical basis $\{|m_L\rangle\}$, the Knill–Laflamme condition reads

$$\left\langle m_L\right| E_a^\dagger E_b \left|n_L\right\rangle = \alpha_{ab}\delta_{mn}. \tag{3.46}$$

Thus the error process has identical diagonal matrix elements on every logical state and no off-diagonal matrix elements between distinct logical states: it cannot learn which encoded state was present. We can therefore correct only errors that do not reveal logical information. This is the content of the Knill–Laflamme conditions.

4 1d Ising model as the repetition code

Our goal is to construct codes whose performance improves as the number n of physical bits or qubits grows. A larger code should protect against more errors, and we would also like to understand whether decoding can remain manageable. Analogies with physics provide useful insight into both questions. We begin with a minimal scalable example whose connection to statistical mechanics foreshadows many later ideas.

4.1 The scalable repetition code is the Ising ferromagnet

The length-n classical repetition code encodes one logical bit as

$$0_L \longleftrightarrow 00\cdots 0, \qquad 1_L \longleftrightarrow 11\cdots 1. \tag{4.1}$$

Previously we used $n = 3$, but there is no obstruction to taking n much larger.

Map the binary symbols to Ising spins by

$$0 \longleftrightarrow +1, \qquad 1 \longleftrightarrow -1. \tag{4.2}$$

Then the two codewords become the two ferromagnetic configurations

$$00\cdots 0 \longleftrightarrow (+,+,\ldots,+), \qquad 11\cdots 1 \longleftrightarrow (-,-,\ldots,-). \tag{4.3}$$

Let $z_i \in \{+1,-1\}$ be classical spin variables and consider the open one-dimensional Ising Hamiltonian

$$H_0(z) = -\sum_{i=1}^{n-1} z_i z_{i+1}, \tag{4.4}$$

as illustrated in Figure 4.1. A ground state must obey

$$z_i z_{i+1} = 1 \qquad \text{for every } i = 1,\ldots,n-1. \tag{4.5}$$

Multiplying by z_{i+1} and using $z_{i+1}^2 = 1$ gives

$$z_i = z_{i+1} \qquad \text{for every } i, \tag{4.6}$$

which is precisely the repetition-code constraint.

From the physics perspective, the two ground states are related by the global $\mathbb{Z}_2$ transformation

$$(z_1, z_2, \ldots, z_n) \longmapsto (-z_1, -z_2, \ldots, -z_n). \tag{4.7}$$

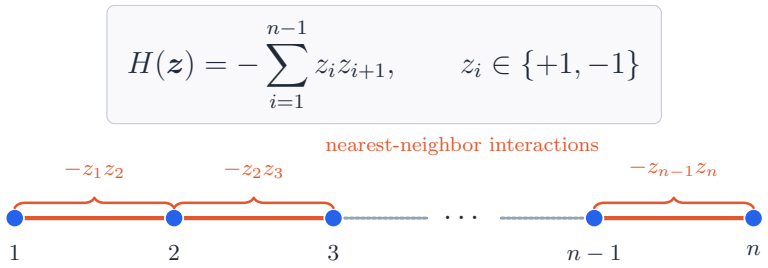

Figure 4.1: The repetition code with n classical bits can be mapped to the Ising model on a 1d chain of length n.

The Hamiltonian is invariant because

$$z_i z_{i+1} \longmapsto (-z_i)(-z_{i+1}) = z_i z_{i+1}. \tag{4.8}$$

Each individual ground state is not invariant under this symmetry: the two states are exchanged. In the thermodynamic language, the ferromagnetic ground states spontaneously break the global $\mathbb{Z}_2$ symmetry.

4.2 Independent bit flips and majority-vote decoding

We now return to the error-correction perspective and ask what is gained by using $n \gg 3$ physical bits. Consider an independent error model in which

$$z_i \longmapsto \eta_i z_i, \tag{4.9}$$

with independent random variables $\eta_i \in \{+1, -1\}$ distributed as

$$\mathbb{P}(\eta_i = +1) = 1 - p, \qquad \mathbb{P}(\eta_i = -1) = p. \tag{4.10}$$

Thus p is the probability that bit i flips. Define the indicator

$$e_i := \frac{1 - \eta_i}{2} = \begin{cases} 0, & \text{no error,} \\ 1, & \text{error.} \end{cases} \tag{4.11}$$

The total number of errors is

$$N := \sum_{i=1}^{n} e_i. \tag{4.12}$$

Its expectation value is

$$\mathbb{E}[N] = \sum_{i=1}^{n} \mathbb{E}[e_i] = n\big[(1-p)\cdot 0 + p \cdot 1\big] = pn. \tag{4.13}$$

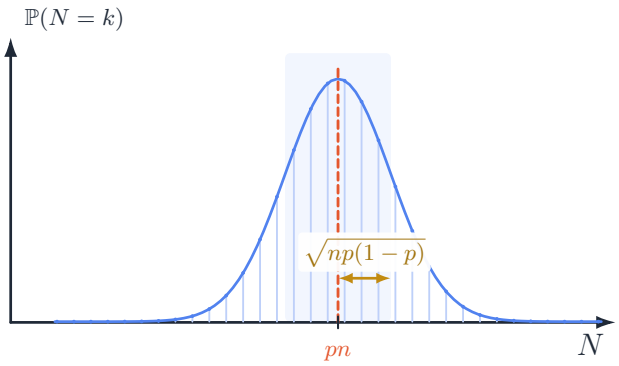

Figure 4.2: The independent errors make N a tightly peaked random variable in the large n limit.

Because $e_i^2 = e_i$ and distinct e_i, e_j are independent,

$$\mathbb{E}[N^2] = \sum_{i,j=1}^{n} \mathbb{E}[e_i e_j] = \sum_{i=1}^{n} \mathbb{E}[e_i] + 2\sum_{i<j} \mathbb{E}[e_i]\mathbb{E}[e_j] = pn + n(n-1)p^2 = (pn)^2 + np(1-p). \quad (4.14)$$

Therefore

$$\mathrm{Var}(N) := \mathbb{E}[N^2] - \mathbb{E}[N]^2 = np(1-p). \quad (4.15)$$

The distribution of N is consequently concentrated near pn, with a characteristic width

$$\sqrt{\mathrm{Var}(N)} = \sqrt{p(1-p)n} \ll pn \qquad (n \longrightarrow \infty) \quad (4.16)$$

for fixed $p > 0$: see Figure 4.2.

Suppose now that

$$0 < p < \frac{1}{2}, \quad (4.17)$$

and decode by majority vote.

Proposition 4.1. For independent bit flips with $p < 1/2$, the probability that majority-vote decoding succeeds approaches one as $n \to \infty$.

Proof. The decoder can fail only if at least half of the bits flip. Hence

$$\mathbb{P}(\text{decoder fails}) = \mathbb{P}\left(N \geq \frac{n}{2}\right) \leq \mathbb{P}\left(|N - pn| \geq \left(\frac{1}{2} - p\right) n\right). \quad (4.18)$$

Chebyshev's inequality and Eq. (4.15) give

$$\mathbb{P}(\text{decoder fails}) \leq \frac{\mathbb{E}[(N-pn)^2]}{\left[(\frac{1}{2}-p)n\right]^2} = \frac{p(1-p)}{n(\frac{1}{2}-p)^2} \longrightarrow 0 \qquad (n \longrightarrow \infty). \tag{4.19}$$

Thus the decoder almost surely works as $n \to \infty$. □

For even n, an exact tie can be declared a failure; the event $N \geq n/2$ used above is therefore a conservative failure event. Taking n odd removes the tie without changing the asymptotic conclusion.

4.3 Faulty measurements of the physical bits

A first apparent problem is that the physical-bit measurements may themselves be faulty. Model the observed spin as

$$z_i \longmapsto \theta_i \eta_i z_i, \tag{4.20}$$

where η_i is the intrinsic bit-flip error and θ_i is a measurement error, with

$$\mathbb{P}(\eta_i = -1) = p_e, \qquad \mathbb{P}(\theta_i = -1) = p_m. \tag{4.21}$$

The observed bit is wrong precisely when exactly one of η_i, θ_i is negative. Its effective error probability is therefore

$$p := \mathbb{P}(\theta_i \eta_i = -1) = p_e(1-p_m) + p_m(1-p_e). \tag{4.22}$$

The same majority-vote argument works whenever $p < 1/2$. In particular, this condition is achieved if

$$p_e < \frac{1}{2}, \qquad p_m < \frac{1}{2}. \tag{4.23}$$

Thus independent measurement faults do not alter Proposition 4.1.

4.4 Noisy syndrome measurements and MAP decoding

4.4.1 Why syndrome noise is more subtle

Quantum error correction measures syndromes rather than the physical qubits. For the repetition code, the available checks are the neighboring products

$$z_i z_{i+1}, \qquad i = 1, \ldots, n-1. \tag{4.24}$$

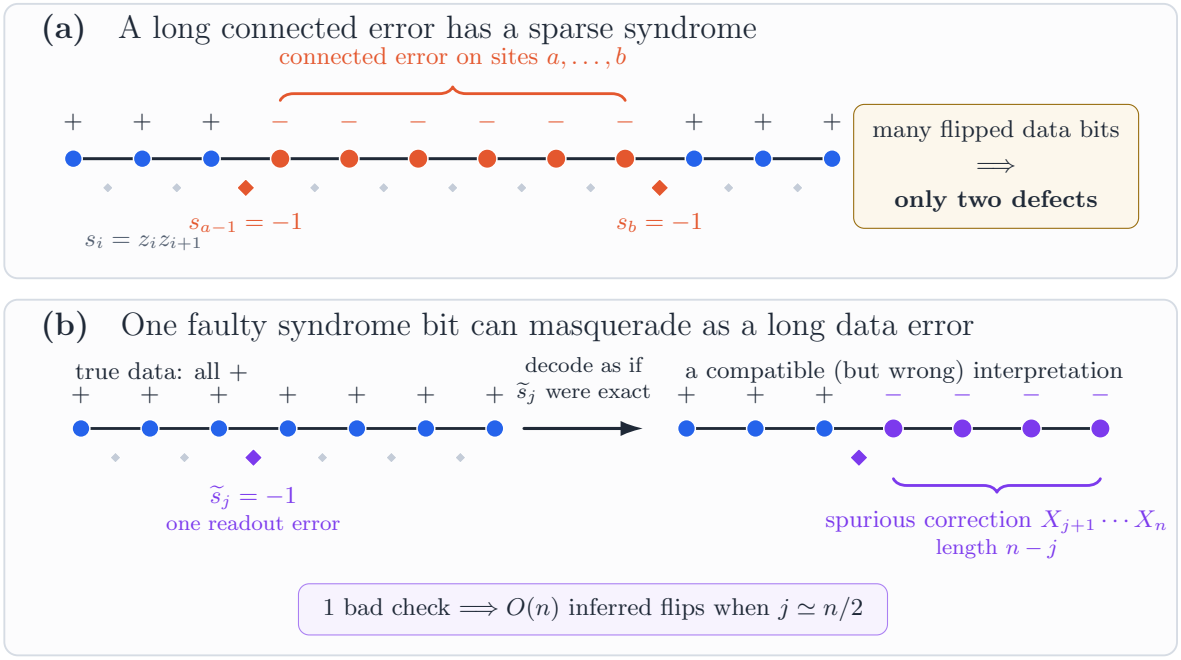

Figure 4.3: Overview of the challenges with decoding the 1d Ising model with noisy syndromes. (a) Large errors can correspond to small syndromes. (b) Not being skeptical of the syndrome can lead to catastrophic decoding errors that introduce very large errors.

A contiguous flipped interval changes only the checks at its endpoints. For example,

$$(z_1, \dots, z_n) = (+, +, \underbrace{-, -, \dots, -}_{\text{flipped interval}}, +) \tag{4.25}$$

has the syndrome pattern

$$(z_1 z_2, z_2 z_3, \dots, z_{n-1} z_n) = (+, -, +, \dots, +, -). \tag{4.26}$$

Only the two domain walls are visible; the syndrome does not directly reveal the length of the flipped interval.

Combined with syndrome-measurement noise, this creates a genuine ambiguity. For example, a measured pattern with one isolated negative check,

$$(s_1, \dots, s_{n-1}) = (+, -, +, \dots, +), \tag{4.27}$$

may be a single faulty syndrome measurement. If it is instead interpreted as a reliable domain wall, the decoder may infer a long boundary-attached error string such as

$$(z_1, \dots, z_n) = (+, +, -, -, \dots, -), \tag{4.28}$$

which is the wrong interpretation of the data: see Figure 4.3.

4.4.2 Bayesian formulation

Given noisy syndrome outcomes

$$s = (s_1, \dots, s_{n-1}), \qquad s_i \in \{+1, -1\}, \tag{4.29}$$

we seek the most likely error configuration

$$z = (z_1, \dots, z_n), \qquad z_i = \begin{cases} +1, & \text{no physical error,} \\ -1, & \text{physical error.} \end{cases} \tag{4.30}$$

Let $t_i \in \{+1, -1\}$ be the syndrome-measurement error, so that

$$s_i = z_i z_{i+1} t_i, \qquad \mathbb{P}(t_i = -1) = p_m. \tag{4.31}$$

The probability of the observed syndrome is

$$\mathbb{P}(s) = \sum_z \mathbb{P}(s \mid z)\mathbb{P}(z). \tag{4.32}$$

Bayes' theorem gives

$$\mathbb{P}(s, z) = \mathbb{P}(s \mid z)\mathbb{P}(z) = \mathbb{P}(z \mid s)\mathbb{P}(s), \tag{4.33}$$

and hence

$$\mathbb{P}(z \mid s) = \frac{\mathbb{P}(s \mid z)\mathbb{P}(z)}{\mathbb{P}(s)}. \tag{4.34}$$

Since $\mathbb{P}(s)$ is fixed by the measured data, the maximum a posteriori (MAP) decoder maximizes $\mathbb{P}(s \mid z)\mathbb{P}(z)$ over z.

4.4.3 Mapping to a random-bond Ising model

Under independent physical errors,

$$\mathbb{P}(z) = \prod_{i=1}^{n} \begin{cases} p_e, & z_i = -1, \\ 1 - p_e, & z_i = +1. \end{cases} \tag{4.35}$$

Introduce h through

$$\mathrm{e}^{-2h} = \frac{p_e}{1 - p_e}, \qquad h = \frac{1}{2} \log \frac{1 - p_e}{p_e}. \tag{4.36}$$

Then, up to a normalization independent of z,

$$\mathbb{P}(z) \propto \exp\left(h \sum_{i=1}^{n} z_i\right). \tag{4.37}$$

Similarly,

$$\mathbb{P}(s \mid z) = \prod_{i=1}^{n-1} \begin{cases} p_m, & z_i z_{i+1} s_i = -1, \\ 1 - p_m, & z_i z_{i+1} s_i = +1. \end{cases} \tag{4.38}$$

Introduce J by

$$\mathrm{e}^{-2J} = \frac{p_m}{1 - p_m}, \qquad J = \frac{1}{2} \log \frac{1 - p_m}{p_m}. \tag{4.39}$$

The likelihood is then

$$\mathbb{P}(s \mid z) \propto \exp\left(J \sum_{i=1}^{n-1} s_i z_i z_{i+1}\right). \tag{4.40}$$

Combining the prior and likelihood gives

$$\mathbb{P}(s \mid z)\mathbb{P}(z) \propto \exp\left[J \sum_{i=1}^{n-1} s_i z_i z_{i+1} + h \sum_{i=1}^{n} z_i\right]. \tag{4.41}$$

Equivalently, define the syndrome-dependent random-bond Ising Hamiltonian

$$H_s(z) := -J \sum_{i=1}^{n-1} s_i z_i z_{i+1} - h \sum_{i=1}^{n} z_i. \tag{4.42}$$

Proposition 4.2. The MAP estimate is a ground state of H_s. This decoder succeeds in the presence of a finite probability for both physical errors and syndrome-measurement errors. It is a homework problem to show that it works whenever $p_{\mathrm{e}} = p_{\mathrm{m}} \leq 1/6$ (not intended to be optimal).

Although the decoder is global, this particular one-dimensional ground-state problem is computationally simple: one may minimize successively over the two possible values of each spin, keeping the best partial cost for each boundary spin. This dynamic-programming procedure takes time proportional to n. But the general mapping of the decoding problem to a statistical mechanics problem (random-bond Ising model, or generalization thereof) is more general, and we will return to it in the context of quantum codes in Section **??**.

The decoding decision uses global information from the syndrome record. It is possible in principle to construct local decoders, but doing so is more subtle.

4.5 Finite temperature and the failure of passive order in one dimension

The coding problem suggests an analogy between finite error probability and finite temperature. In fact, we will see in Section 5.6 that some codes admit very simple decoders based on mimicking the process of thermalizing the system with a bath.

Unfortunately, this analogy to physics does not work for the 1d Ising model. Consider the thermal partition function of Eq. (4.4):

$$Z_n(\beta) := \sum_{z_1,\ldots,z_n} \mathrm{e}^{-\beta H(z)} \tag{4.43}$$

$$= \sum_{z_1,\ldots,z_n} \exp\left(\beta \sum_{i=1}^{n-1} z_i z_{i+1}\right). \tag{4.44}$$

Introduce the transfer matrix

$$T_\beta := \begin{pmatrix} \mathrm{e}^{\beta} & \mathrm{e}^{-\beta} \\ \mathrm{e}^{-\beta} & \mathrm{e}^{\beta} \end{pmatrix}, \tag{4.45}$$

whose row and column indices are the two possible spin values. For open boundary conditions,

$$Z_n(\beta) = \begin{pmatrix} 1 & 1 \end{pmatrix} T_\beta^{n-1} \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \tag{4.46}$$

The vector $(1,1)^{\mathsf{T}}$ is an eigenvector:

$$T_\beta \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 2\cosh(\beta) \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \tag{4.47}$$

Therefore

$$Z_n(\beta) = 2\big(2\cosh\beta\big)^{n-1}. \tag{4.48}$$

The finite-size free energy is

$$F_n(\beta) = -\frac{1}{\beta}\log Z_n(\beta) = -\frac{1}{\beta}\left[\log 2 + (n-1)\log(2\cosh\beta)\right], \tag{4.49}$$

and the thermodynamic free-energy density is

$$f(\beta) := \lim_{n\to\infty} \frac{F_n(\beta)}{n} = -\frac{1}{\beta}\log(2\cosh\beta). \tag{4.50}$$

This expression is nonsingular at every finite β, so the one-dimensional Ising model has no finite-temperature thermodynamic phase transition.

Strictly speaking, this does not necessarily show that there is no dynamical stability of the

ferromagnet! However in the 1d Ising model there is genuinely no passive memory. A more physical picture as to why is that domain walls between + and − easily proliferate.

Indeed, thinking about domain walls gives us another nice way to see why the 1d Ising model at finite temperature wouldn't protect information well. The free-energy cost of a domain wall scales as

$$\Delta F_{\mathrm{dw}} \sim \Delta E_{\mathrm{dw}} - T\Delta S_{\mathrm{dw}}, \tag{4.51}$$

where

$$\Delta E_{\mathrm{dw}} = O(1), \qquad \Delta S_{\mathrm{dw}} \sim \log n. \tag{4.52}$$

Thus, at every fixed $T > 0$, the entropic gain eventually overwhelms the finite energy cost as n grows. Domain walls proliferate and destroy long-range ferromagnetic order.

This distinction is important: active majority-vote decoding succeeds whenever $p < 1/2$, while the bare one-dimensional Ising Hamiltonian does not store the bit passively at any nonzero temperature. A decoder that can succeed and a finite-temperature ordered phase are related ideas, but they are not the same notion of protection.

5 2d Ising model

In Section 4, the classical repetition code was identified with the one-dimensional Ising model. That analogy supplied a physical picture for error correction, but its real utility becomes clearer in two spatial dimensions.

5.1 From the repetition code to the 2d Ising model

Consider an $L \times L$ square lattice, with a classical spin $z_i \in \{+1, -1\}$ at each vertex. The ferromagnetic Ising Hamiltonian is

$$H_L(z) = -\sum_{\langle i,j \rangle} z_i z_j, \tag{5.1}$$

where the sum runs over neighboring vertices. As in one dimension, the two ground states are the two repetition-code words,

$$0_L \longleftrightarrow z_i = +1 \text{ for every } i, \qquad 1_L \longleftrightarrow z_i = -1 \text{ for every } i. \tag{5.2}$$

At inverse temperature β, the thermal Gibbs distribution is

$$\mathbb{P}_\beta(z) := \frac{1}{Z_L(\beta)} \mathrm{e}^{-\beta H_L(z)}, \tag{5.3a}$$

$$Z_L(\beta) := \sum_z \mathrm{e}^{-\beta H_L(z)}. \tag{5.3b}$$

The two-dimensional model has a genuine phase transition: see Figure 5.1. The low-temperature ferromagnetic phase will be useful for error correction.

For every edge $\langle i, j \rangle$, define the bond check

$$s_{ij} := z_i z_j \in \{+1, -1\}. \tag{5.4}$$

A check is violated when $s_{ij} = -1$. Drawing a dual edge across every violated bond produces the domain walls separating regions of positive and negative spins. If these domain walls do not intersect the boundary, they form closed loops, as illustrated in Figure 5.2. We will see that this property leads to a drastic change in the physical behavior of the 2d Ising model vs. 1d Ising model!

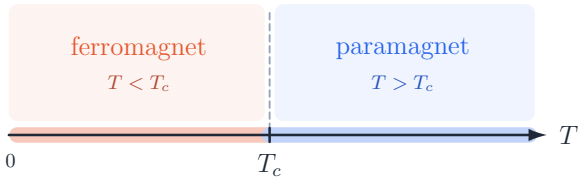

Figure 5.1: The phase diagram of the 2d Ising model.

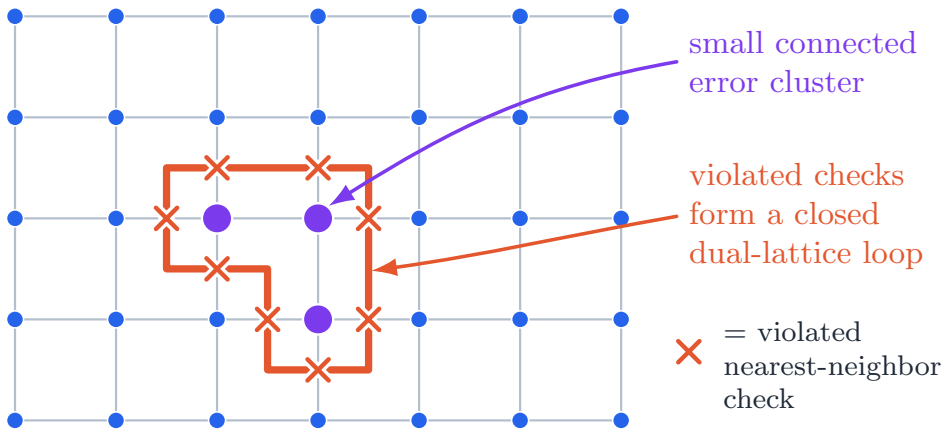

Figure 5.2: Violated parity-checks in the 2d Ising model form closed loops (except at the boundaries). Detecting error clusters is much easier!

Each violated bond changes its contribution to Eq. (5.1) from -1 to $+1$, so a domain wall of length ℓ has energy cost

$$\Delta E_{\mathrm{dw}} = 2\ell. \tag{5.5}$$

Proposition 5.1 (Linear energy barrier)**.** Any sequence of single-spin flips that carries the all-positive codeword to the all-negative codeword must pass through a configuration containing a domain wall whose length is at least of order $L - 1$. Consequently, the maximum energy encountered along the path obeys

$$\Delta E_{\mathrm{barrier}} \gtrsim 2(L - 1). \tag{5.6}$$

Proof sketch. The reason is topological. Initially the positive spins percolate across the sample, whereas finally the negative spins do. At an intermediate stage the percolating region must be cut by a domain wall joining opposite boundaries: see Figure 5.3. Such a wall has length at least of order $L - 1$, and every edge of the wall costs two units of energy. □

5.2 Peierls' argument

The energy of a long domain wall grows with its length, but so does its entropy. A crude Peierls estimate compares these two effects. Once one edge of a domain wall is fixed, there are at most three choices for the next step: the wall cannot immediately retrace the edge from which it arrived.

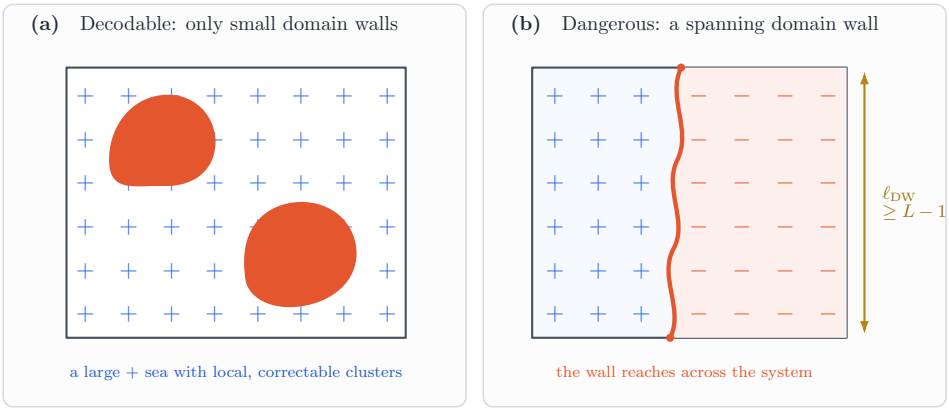

Figure 5.3: Left: small error clusters in the 2d Ising model are easily decodable and do not destroy long-range order. Right: dangerous "non-decodable" configurations can be taken as those where a long domain wall stretches across the entire system.

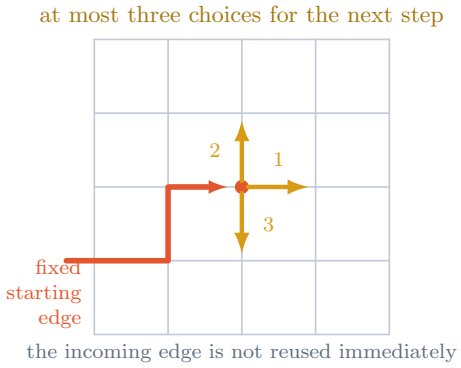

Figure 5.4: Justification for overcounting the number of domain walls in (5.7).

Hence the number of length-ℓ walls passing through a fixed dual edge is bounded by (see Figure 5.4)

$$N_\ell \leq 3^\ell. \tag{5.7}$$

The corresponding entropy is bounded by

$$\Delta S_{\mathrm{dw}} \lesssim \log N_\ell \leq \ell \log 3. \tag{5.8}$$

Combining this estimate with Eq. (5.5) gives the free-energy cost

$$\Delta F_{\mathrm{dw}} = \Delta E_{\mathrm{dw}} - T\Delta S_{\mathrm{dw}} \gtrsim 2\ell - T\log\left(3^\ell\right) = \ell(2 - T\log 3). \tag{5.9}$$

Therefore domain walls are disfavored whenever

$$T < \frac{2}{\log 3}. \tag{5.10}$$

This is a lower bound on the actual critical temperature, not an exact calculation of T_c. It is nevertheless enough to establish a genuine low-temperature ordered phase.

The crude count (5.7) overcounts contours by not restricting to closed, self-avoiding contours, but it correctly captures the exponential competition between contour entropy and Boltzmann suppression.

5.3 Consequences of the low-temperature phase

We will not calculate the exact location of the phase transition. Instead, we summarize the properties of the ferromagnetic phase $T < T_c$ that matter for error correction.

5.3.1 Long-range order

Small error clusters are present at nonzero temperature, but large domain walls remain rare. In the thermodynamic limit, the two-point function obeys

$$\lim_{\mathrm{dist}(i,j)\to\infty} \langle z_i z_j \rangle_\beta > 0, \tag{5.11}$$

where

$$\langle z_i z_j \rangle_\beta := \frac{1}{Z_L(\beta)} \sum_z \mathrm{e}^{-\beta H_L(z)} z_i z_j. \tag{5.12}$$

This is the signature of spontaneous symmetry breaking and long-range ferromagnetic order.

5.3.2 Thermodynamic nonanalyticity

Define the finite-volume free energy and its density by

$$F_L(\beta) := -\frac{1}{\beta} \log Z_L(\beta), \tag{5.13a}$$

$$f(\beta) := \lim_{L\to\infty} \frac{F_L(\beta)}{L^2} = -\lim_{L\to\infty} \frac{1}{\beta L^2} \log Z_L(\beta). \tag{5.13b}$$

The function $f(\beta)$ is nonanalytic at $\beta = \beta_c$, so the phase transition is thermodynamically detectable.

5.3.3 Concentration of the Gibbs measure

For $\beta > \beta_c$, configuration space separates into two large regions A_+ and A_-, concentrated near the all-positive and all-negative codewords. By the global $\mathbb{Z}_2$ symmetry, their probabilities are

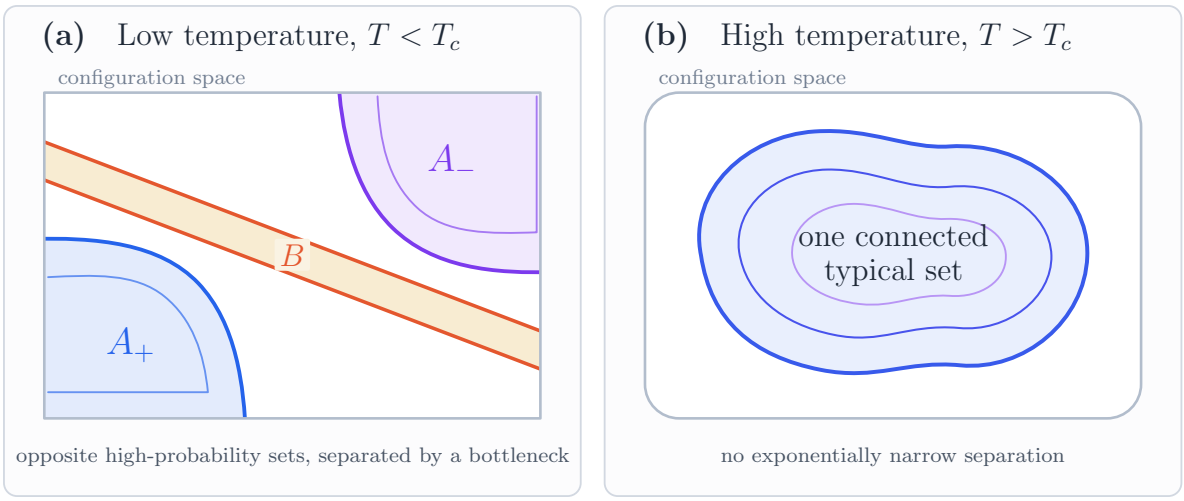

Figure 5.5: A cartoon of where the probability distribution $\mathbb{P}_\beta$ is domianted at: (a) $T < T_c$; (b) $T > T_c$.

approximately equal:

$$\mathbb{P}_\beta(A_+) = \mathbb{P}_\beta(A_-) \simeq \frac{1}{2}. \tag{5.14}$$

The two regions are separated by a bottleneck B consisting of configurations with a system-spanning domain wall. At low temperature,

$$\mathbb{P}_\beta(B) \sim \mathrm{e}^{-\alpha L} \qquad \text{for some } \alpha > 0. \tag{5.15}$$

Thus a thermal configuration is overwhelmingly likely to be found near one of the two codewords, even though it generally contains small error clusters. At high temperature, $\beta < \beta_c$, these two regions merge into one broad cluster in configuration space, and typical states have no long-range order. See Figure 5.5 for illustrations.

5.4 Gibbs samplers and detailed balance

Ferromagnetism suggests a passive memory: a local thermal process should remain within one of the two ordered basins for a very long time. A *Gibbs sampler* is a Markov chain whose stationary distribution is the thermal Gibbs distribution.

Definition 5.2. Consider single-spin Metropolis updates. Let $z^{(i)}$ denote the configuration obtained from z by flipping spin i. Choose a site uniformly and accept the proposed flip with probability

$$a(z \to z^{(i)}) := \min\left\{1, \mathrm{e}^{-\beta[H_L(z^{(i)}) - H_L(z)]}\right\}. \tag{5.16}$$

The transition probabilities may therefore be written as

$$P(z \to z^{(i)}) = \frac{1}{L^2} a(z \to z^{(i)}), \tag{5.17}$$

with

$$P(z \to z') = 0 \qquad \text{if } z \text{ and } z' \text{ differ on two or more spins.} \tag{5.18}$$

The rejection probability is included in the self-transition,

$$P(z \to z) = 1 - \sum_i P(z \to z^{(i)}). \tag{5.19}$$

The Metropolis rule obeys detailed balance:

$$P(z \to z') \mathrm{e}^{-\beta H_L(z)} = P(z' \to z) \mathrm{e}^{-\beta H_L(z')}. \tag{5.20}$$

Proposition 5.3. If the initial configuration is sampled from the thermal distribution, then one Markov step leaves that distribution invariant:

$$\begin{aligned} \mathbb{P}_{\text{after}}(z') &= \sum_z P(z \to z') \frac{\mathrm{e}^{-\beta H_L(z)}}{Z_L(\beta)} = \sum_z P(z' \to z) \frac{\mathrm{e}^{-\beta H_L(z')}}{Z_L(\beta)} \\ &= \frac{\mathrm{e}^{-\beta H_L(z')}}{Z_L(\beta)} \sum_z P(z' \to z) = \mathbb{P}_\beta(z'). \end{aligned} \tag{5.21}$$

Thus the Gibbs state is guaranteed to be stationary.

5.5 Bottlenecks and exponentially long memory times

The separation of the two ordered regions by an exponentially unlikely set implies a long memory time for any local Gibbs sampler.

Theorem 5.4 (Bottleneck estimate)**.** Let A, B, C be regions of configuration space such that every local path from A to C must pass through B, as illustrated in Figure 5.6. For a reversible Markov chain with equilibrium distribution $\mathbb{P}_{\text{eq}}$, the characteristic time τ to cross from one large basin to the other satisfies, up to $\mathrm{O}(1)$ constants,

$$\mathbb{E}[\tau] \gtrsim \frac{\min\{\mathbb{P}_{\text{eq}}(A), \mathbb{P}_{\text{eq}}(C)\}}{\mathbb{P}_{\text{eq}}(B)}. \tag{5.22}$$

Proof sketch. Restrict the dynamics to $A \cup B$, reflecting any attempted transition out of this set. The restricted equilibrium measure assigns probability

$$\mathbb{P}_{AB}(B) = \frac{\mathbb{P}_{\text{eq}}(B)}{\mathbb{P}_{\text{eq}}(A) + \mathbb{P}_{\text{eq}}(B)} \tag{5.23}$$

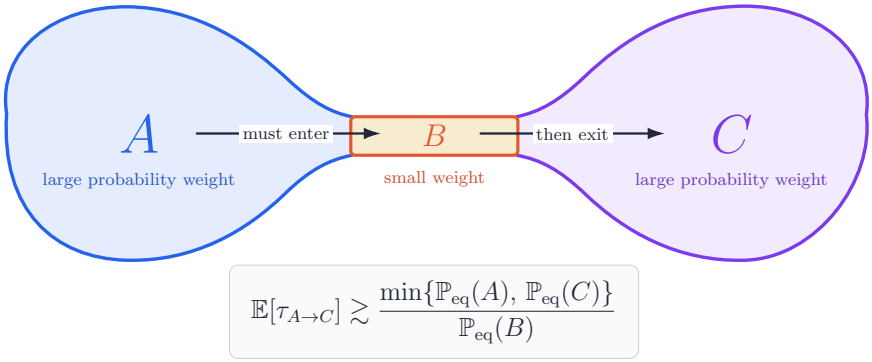

Figure 5.6: A sketch of the sets A, B and C in Theorem 5.4.

to the bottleneck. Until the first visit to B, the restricted dynamics and the full dynamics agree. If the restricted chain starts in its stationary distribution, then

$$\mathbb{P}(\tau_B \le t) \le \sum_{s=0}^{t} \mathbb{P}(X_s \in B) = (t+1)\mathbb{P}_{AB}(B). \tag{5.24}$$

A time of order $1/\pi_{AB}(B)$ is therefore needed before a visit to B is likely. Repeating the argument from the C side gives Eq. (5.22). □

A fully rigorous Markov-chain statement is usually formulated using conductance and specifies the initial distribution and a precise hitting or mixing time. The estimate above is the scaling statement used in this lecture: an exponentially small bottleneck measure produces an exponentially large crossing time.

For the low-temperature two-dimensional Ising model, take B_L to be the set of configurations containing a domain wall of length at least $L-1$, as described in Section 5.3.3:

$$B_L := \{z : \ z \text{ contains a domain wall of length at least } L-1\}. \tag{5.25}$$

There are at most order $L^2 3^\ell$ possible placements and shapes of a wall of length ℓ. As shown in Figure 5.7, removing one selected long wall produces a configuration $z_\star$ of lower energy while leaving the other, smaller walls untouched. For the normalization of Eq. (5.1),

$$\frac{\mathbb{P}_\beta(z)}{\mathbb{P}_\beta(z_\star)} = \mathrm{e}^{-\beta[H_L(z)-H_L(z_\star)]} \le \mathrm{e}^{-2\beta\ell}. \tag{5.26}$$

The Peierls count then gives

$$\mathbb{P}_\beta(B_L) \lesssim \sum_{\ell \ge L-1} L^2 3^\ell \mathrm{e}^{-2\beta\ell} \lesssim \mathrm{e}^{-\alpha L} \qquad \text{for sufficiently large } \beta, \tag{5.27}$$

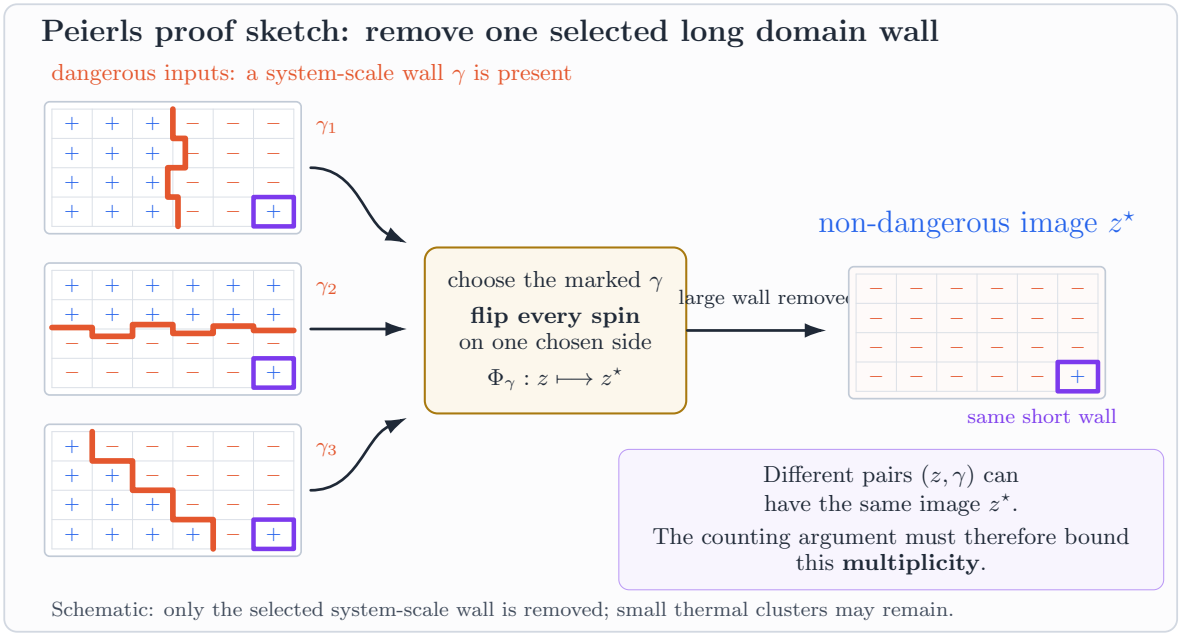

Figure 5.7: Identifying the bottleneck set B in the 2d Ising model and establishing (5.26) proceeds by counting the number of large domain walls that can exist, and showing that the entire probability of all such large-domain configurations is smaller than the probability of the (many fewer) states without these large domain walls.

where $\alpha > 0$; this is (5.15). Combining this with Theorem 5.4 yields

$$\mathbb{E}[\tau] \gtrsim \mathrm{e}^{\alpha L}. \tag{5.28}$$

It is essential that the bottleneck is defined by a *large* domain wall. A typical thermal configuration has a finite density of small domain walls. The dynamics need not remove every error simultaneously; it only needs to prevent those local errors from assembling into a system-spanning wall that changes the stored logical bit.

5.6 The Gibbs sampler as a local, fault-tolerant decoder

Consider a bulk site labeled 1, with its four neighboring sites labeled $2, 3, 4, 5$, as in Figure 5.8. The part of the energy involving z_1 is

$$H_1(z) = -z_1(z_2 + z_3 + z_4 + z_5). \tag{5.29}$$

Flipping z_1 changes the energy by

$$\Delta E_1 := H_L(z^{(1)}) - H_L(z) = 2z_1(z_2 + z_3 + z_4 + z_5) = 2(s_{12} + s_{13} + s_{14} + s_{15}). \tag{5.30}$$

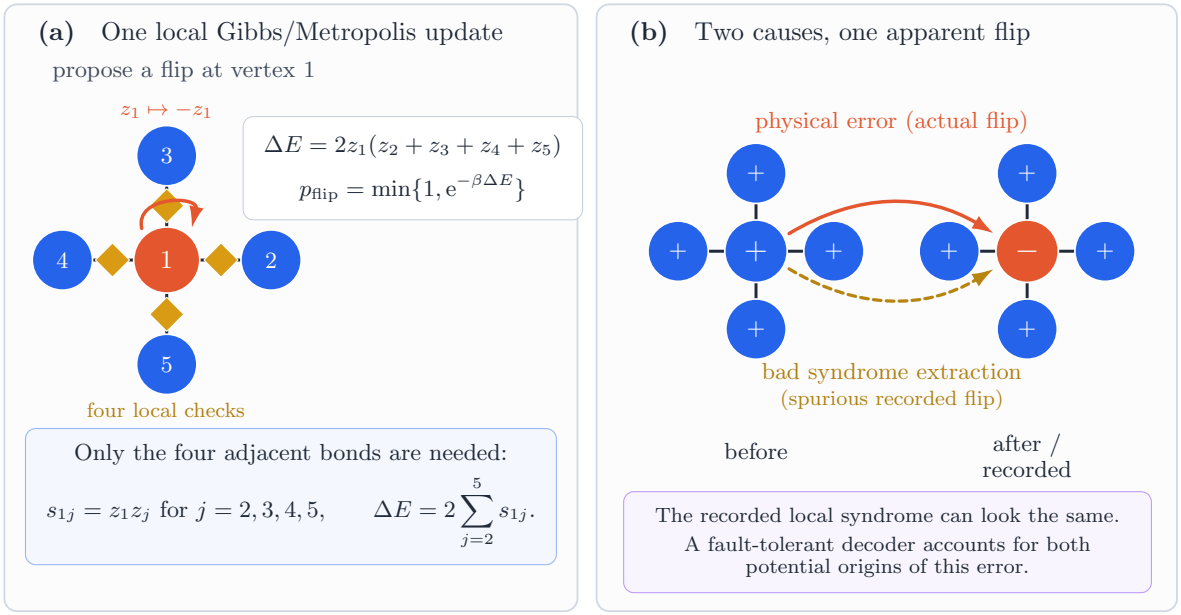

Figure 5.8: A sketch of how the local Gibbs sampler acts on a single vertex in the 2d Ising model by only measuring the local syndromes. As long as we ensure that (5.31) is obeyed, we can in principle convert the local Gibbs sampler into an local and fault-tolerant "passive" decoder.

Thus the Metropolis decision requires only local information about the four neighboring syndrome checks. It does not measure the logical bit. The thermal decoder flips site 1 with probability $\min\left\{1, e^{-\beta\Delta E_1}\right\}$. This is precisely the kind of local update that is safe for error correction.

We say that a decoder has a **threshold** if, when subject to repeated rounds of physical errors and noisy syndrome extraction followed by error correction, the probability for a logical error vanishes as the code increases in size ($n \to \infty$). To see how a nonzero fault-tolerance threshold can arise, consider the worst local energy-increasing move. If all four neighboring checks are initially satisfied, then flipping the central spin creates four violated checks and costs $\Delta E_1 = 8$. Let γ_{phys} be the rate of a physical spin error, $\gamma_{\text{dec}}^{\text{fail}}$ the rate at which a faulty decoder produces the same upward transition, and $\gamma_{\text{dec}}^{\text{succ}}$ the rate of the reverse successful correction. To reproduce detailed balance at inverse temperature β, the local rates should satisfy

$$\gamma_{\text{phys}} + \gamma_{\text{dec}}^{\text{fail}} = e^{-8\beta}\gamma_{\text{dec}}^{\text{succ}}. \tag{5.31}$$

At fixed β, the factor $e^{-8\beta}$ is a nonzero constant independent of the system size. The physical-error and decoder-failure rates may therefore be a finite fraction of the successful-decoding rate while the combined dynamics still behaves like a low-temperature Gibbs sampler. This is the origin of a finite threshold.

For other local patterns, an exact finite-temperature sampler may intentionally accept some

energy-increasing moves. Schematically, the rates obey

$$\gamma_{\text{phys}} + \gamma_{\text{dec}}^{\text{fail}} + \gamma_{\text{add}}^{\text{err}} = \mathrm{e}^{-\mathrm{O}(\beta)} \gamma_{\text{dec}}^{\text{succ}}, \tag{5.32}$$

where $\gamma_{\text{add}}^{\text{err}}$ denotes the deliberate energy-raising moves needed to match the target Gibbs sampler. A thermal decoder does not monotonically decrease the energy; it preserves the desired low-temperature distribution.

Thus a physical phase of matter can inspire a robust, local, fault-tolerant decoder; classical magnetic memory similarly stores a bit in a ferromagnet.

6 Classical parity-check codes

6.1 From the Ising repetition code to linear algebra

The one-dimensional Ising model describes the classical repetition code. Its codewords are $0\,0\,0\,0\cdots$ and $1\,1\,1\,1\cdots$ which look a lot like the two ground states of the ferromagnetic energy

$$E(z) = -\sum_{i\sim j} z_i z_j, \qquad z_i \in \{+1, -1\} \tag{6.1}$$

if we replaced 0 and 1 with spin configurations $+1$ and -1. But let's work with the bits $\{0, 1\}$ instead, and write

$$x_i := \frac{1 - z_i}{2} \in \{0, 1\}. \tag{6.2}$$

Then a bond is satisfied or violated according to

$$z_i z_j = \begin{cases} +1, & x_i + x_j = 0 \pmod 2, \\ -1, & x_i + x_j = 1 \pmod 2. \end{cases} \tag{6.3}$$

Thus the ferromagnetic ground-state condition is a parity-check condition. It is helpful to express it in a more mathematical language: addition and multiplication are henceforth understood modulo two, so that we work over the field $\mathbb{F}_2$. An n-bit string is a vector

$$\mathbf{x} = (x_1, \ldots, x_n)^{\mathsf{T}} \in \mathbb{F}_2^n. \tag{6.4}$$

For the one-dimensional repetition code, the neighboring parity checks are

$$x_1 + x_2 = 0, \qquad x_2 + x_3 = 0, \qquad \ldots, \qquad x_{n-1} + x_n = 0. \tag{6.5}$$

They can be collected into the matrix equation

$$H_{\text{rep}}\mathbf{x} = \mathbf{0}, \qquad H_{\text{rep}} := \begin{pmatrix} 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 1 \end{pmatrix} \in \mathbb{F}_2^{(n-1)\times n}. \tag{6.6}$$

The matrix H is called a *parity-check matrix*. A bit string is a codeword precisely when it is a right null vector of H:

$$\mathbf{x} \text{ is a codeword} \quad \Longleftrightarrow \quad H\mathbf{x} = \mathbf{0} \quad \Longleftrightarrow \quad \mathbf{x} \in \ker(H). \tag{6.7}$$

For the repetition code,

$$\ker(H_{\text{rep}}) = \text{span}_{\mathbb{F}_2} \left\{ \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \right\} = \left\{ \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \right\}. \tag{6.8}$$

A *generator matrix* has rows that generate the codewords. For the repetition code one may take

$$G_{\text{rep}} = \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix}. \tag{6.9}$$

It obeys

$$H_{\text{rep}} G_{\text{rep}}^{\mathsf{T}} = 0, \qquad \ker(H_{\text{rep}}) = \text{im}(G_{\text{rep}}^{\mathsf{T}}). \tag{6.10}$$

This may seem like an overly complicated description of the repetition code, but it has an immediate generalization: other classical parity-check codes are obtained by choosing different matrices H and G.

6.2 Dimension, distance, and syndrome decoding

The first goal is to store more than one logical bit, which requires a larger kernel.

Theorem 6.1 (Rank-nullity over $\mathbb{F}_2$)**.** For a parity-check matrix $H \in \mathbb{F}_2^{m \times n}$, the code $\ker(H)$ is a vector space of dimension

$$k := \dim \ker(H) = n - \text{rank}(H). \tag{6.11}$$

The integer k is the number of logical bits, and the code contains 2^k codewords.

Choosing a basis of $\ker(H)$ as the rows of a matrix $G \in \mathbb{F}_2^{k \times n}$ gives

$$HG^{\mathsf{T}} = 0, \qquad \ker(H) = \text{im}(G^{\mathsf{T}}). \tag{6.12}$$

The second goal is robustness against errors.

Definition 6.2 (Hamming weight and code distance)**.** The Hamming weight of a vector is the number of its nonzero entries:

$$|\mathbf{x}| := \#\{i : x_i = 1\}. \tag{6.13}$$

The distance of a linear code is the smallest weight of a nonzero codeword:

$$d := \min_{\substack{\mathbf{x}\in\ker(H)\\ \mathbf{x}\neq\mathbf{0}}} |\mathbf{x}|. \tag{6.14}$$

It is the minimum number of bit flips needed to change one codeword into a different codeword. Suppose that a codeword $\mathbf{x} \in \ker(H)$ suffers a bit-flip error $\mathbf{e}$, so that the received string is

$$\mathbf{r} = \mathbf{x} + \mathbf{e}. \tag{6.15}$$

A minimum-distance decoder looks for a valid codeword $\mathbf{y} \in \ker(H)$ that minimizes

$$|\mathbf{r} + \mathbf{y}| = |(\mathbf{x} + \mathbf{e}) + \mathbf{y}|. \tag{6.16}$$

Here subtraction and addition are identical because the arithmetic is over $\mathbb{F}_2$.

Proposition 6.3 (Correction and detection from the distance)**.** A minimum-distance decoder corrects every error satisfying

$$|\mathbf{e}| \leq t, \qquad t := \left\lfloor \frac{d-1}{2} \right\rfloor. \tag{6.17}$$

Moreover, every nonzero error of weight at most $d-1$ is detectable.

Proof. For any other codeword $\mathbf{y} \neq \mathbf{x}$, the Hamming triangle inequality gives

$$|\mathbf{r} + \mathbf{y}| = |\mathbf{e} + (\mathbf{x} + \mathbf{y})| \geq |\mathbf{x} + \mathbf{y}| - |\mathbf{e}| \geq d - |\mathbf{e}| > |\mathbf{e}| = |\mathbf{r} + \mathbf{x}|. \tag{6.18}$$

Thus $\mathbf{x}$ is the unique nearest codeword whenever $2|\mathbf{e}| < d$. An error is undetectable only when it is a nonzero codeword, whose weight is at least d. □

In practice, decoding begins by measuring the *syndrome*

$$\mathbf{s} := H\mathbf{r} = H(\mathbf{x} + \mathbf{e}) = H\mathbf{e}. \tag{6.19}$$

The syndrome is invariant if the error is shifted by a codeword $\mathbf{y} \in \ker(H)$:

$$H(\mathbf{e} + \mathbf{y}) = H\mathbf{e}. \tag{6.20}$$

Errors that differ by a codeword therefore have the same syndrome. A nonzero codeword added to the error is an undetectable logical error.

A classical code with n physical bits, k logical bits, and distance d is called an $[n, k, d]$ code. The repetition code has parameters $[n, 1, n]$. We'd ideally like to have both k and d are large; that this is possible will be confirmed soon in Proposition 6.5.

6.3 The Hamming code

Example 6.4 (The seven-bit Hamming code)**.** Consider the parity-check and generator matrices

$$H_{\text{Ham}} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}, \tag{6.21}$$

$$G_{\text{Ham}} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}. \tag{6.22}$$

One checks that

$$H_{\text{Ham}} G_{\text{Ham}}^{\mathsf{T}} = 0. \tag{6.23}$$

The four rows of G_{Ham} are linearly independent, so $k = 4$. The seven columns of H_{Ham} are the seven distinct nonzero vectors in $\mathbb{F}_2^3$. Consequently no weight-one vector has zero syndrome, and no weight-two vector has zero syndrome because two distinct columns cannot sum to zero. The first row of G_{Ham} has weight three, so the distance is exactly $d = 3$. Thus the Hamming code has parameters [7,4,3].

Like in the Ising model, the Hamming codewords can be viewed as the ground states of a classical check Hamiltonian. Translating bits to spins with Eq. (6.2) gives

$$H_0(z) = -z_1 z_3 z_5 z_7 - z_2 z_3 z_6 z_7 - z_4 z_5 z_6 z_7. \tag{6.24}$$

Logical bit flips are symmetries of this Hamiltonian. For example, the first row of G_{Ham} flips z_1, z_2, z_3, under which

$$H_0 \longmapsto -(-z_1)(-z_3) z_5 z_7 - (-z_2)(-z_3) z_6 z_7 - z_4 z_5 z_6 z_7 = H_0. \tag{6.25}$$

Thus the logical operations are again symmetries of the classical check Hamiltonian. Since $d = 3$, the Hamming code corrects any single-bit error, from Proposition 6.3.

The weight-four checks of the Hamming code are useful. To store four logical bits using four independent length-three repetition codes, whose checks all have weight two, one would use

$$H_{\text{rep}}^{(3)} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \qquad H_{4\text{rep}} = \begin{pmatrix} H_{\text{rep}}^{(3)} & 0 & 0 & 0 \\ 0 & H_{\text{rep}}^{(3)} & 0 & 0 \\ 0 & 0 & H_{\text{rep}}^{(3)} & 0 \\ 0 & 0 & 0 & H_{\text{rep}}^{(3)} \end{pmatrix}. \tag{6.26}$$

The resulting parameters would be

$$[4 \cdot 3,\ 4 \cdot 1,\ 3] = [12, 4, 3]. \tag{6.27}$$

Higher-weight checks have therefore improved both k/n and d/n relative to this collection of repetition codes.

6.4 Asymptotic properties of codes as $n \to \infty$

6.4.1 Classical Hamming bound

If the check weight is allowed to increase, how large can k and d be simultaneously?

Proposition 6.5 (Hamming bound)**.** Every binary linear $[n, k, d]$ code obeys

$$\sum_{j=0}^{t} \binom{n}{j} \leq 2^{n-k}, \qquad t = \left\lfloor \frac{d-1}{2} \right\rfloor, \tag{6.28}$$

or equivalently

$$k \leq n - \log_2 \left(\sum_{j=0}^{t} \binom{n}{j} \right). \tag{6.29}$$

Proof. For a fixed syndrome $\mathbf{s}$, the equation

$$H\mathbf{r} = \mathbf{s} \tag{6.30}$$

has 2^k solutions: if $\mathbf{r}_0$ is one solution, then all solutions are $\mathbf{r}_0 + \mathbf{y}$ with $\mathbf{y} \in \ker(H)$. Since $\operatorname{rank}(H) = n - k$, there are at most 2^{n-k} possible syndromes.

The distance condition implies that every error $\mathbf{e}$ with $|\mathbf{e}| \leq t$ has a unique syndrome. Otherwise two distinct such errors would satisfy

$$H\mathbf{e}_1 = H\mathbf{e}_2 \quad \Longrightarrow \quad H(\mathbf{e}_1 + \mathbf{e}_2) = 0, \tag{6.31}$$

so $\mathbf{e}_1 + \mathbf{e}_2 \in \ker(H)$ is a codeword. But that's an issue:

$$|\mathbf{e}_1 + \mathbf{e}_2| \leq |\mathbf{e}_1| + |\mathbf{e}_2| \leq 2t < d, \tag{6.32}$$

since we'd have a codeword of Hamming weight $< d$. That's a contradiction since

$$\#\{\text{errors of weight} \leq t\} \leq \sum_{j=0}^{t} \binom{n}{j}. \tag{6.33}$$

Injecting these errors into the set of syndromes proves Eq. (6.28). □

The $[7, 4, 3]$ Hamming code saturates this bound. Here $t = 1$, and

$$2^{7-4} = 8 = \binom{7}{0} + \binom{7}{1} = 1 + 7. \tag{6.34}$$

For large n and j, the binomial coefficient has the asymptotic form

$$\binom{n}{j} = 2^{nh_2(j/n)+\mathrm{o}(n)}, \tag{6.35}$$

where the binary entropy is

$$h_2(x) := -x \log_2 x - (1-x) \log_2(1-x). \tag{6.36}$$

Thus, we find:

Proposition 6.6. As $n \to \infty$, the Hamming bound gives schematically

$$\frac{k}{n} \leq 1 - h_2\left(\frac{d}{2n}\right) + \mathrm{o}(1). \tag{6.37}$$

In particular, the bound permits both k/n and d/n to remain finite as $n \to \infty$.

6.4.2 LDPC codes

Codes with k/n and d/n positive as $n \to \infty$ might use checks whose weights are of order n. We will see in Section **??** that this is very undesirable – especially for a quantum code! What we'd prefer instead is a low-density parity-check code:

Definition 6.7 (Low-density parity-check code)**.** A family of parity-check matrices is *low-density* when there are constants $\Delta_C, \Delta_B = O(1)$, independent of n, such that each check contains at most Δ_{C} bits, and each bit participates in at most Δ_{B} checks. Equivalently, every row and every column of H has bounded Hamming weight.

Theorem 6.8 (Asymptotically good LDPC codes)**.** There exist families of LDPC codes with block length $n \to \infty$ for which

$$\liminf_{n\to\infty} \frac{k}{n} > 0, \qquad \liminf_{n\to\infty} \frac{d}{n} > 0. \tag{6.38}$$

Such code families are called *asymptotically good*.

Proof. The proof of this is, in fact, a corollary of a much later Theorem **??**. □

6.5 Tanner graphs

A useful way to depict a classical linear code is with a *Tanner graph*. It is a bipartite graph with a circular node for every bit and a square node for every check. Bit i is connected to check a precisely

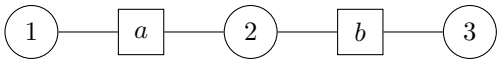

Figure 6.1: The Tanner graph of the three-bit repetition code (Example 1.1). Circles denote bits, squares denote checks, and an edge records a nonzero entry of the parity-check matrix.

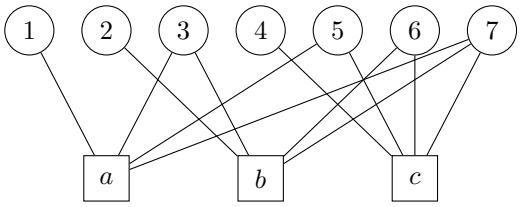

Figure 6.2: A Tanner graph for the seven-bit Hamming code (Example 6.4).

when $H_{ai} = 1$, equivalently annotated $i \in \partial a$.

Example 6.9. The three-bit repetition code (Example 1.1) has parity-check matrix

$$H = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \tag{6.39}$$

and the Tanner graph shown in Figure 6.1. The Tanner graph of the $[7, 4, 3]$ Hamming code is shown in Figure 6.2. Check nodes a, b, c correspond to the three rows of Eq. (6.21).

6.6 Redundant checks

The choice of parity-check matrix is not unique. One may add a redundant check, meaning a row that is linearly dependent on the existing rows. For example, adding the first two rows of H_{Ham} gives

$$\mathbf{h}_{\mathrm{red}} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}, \qquad \widetilde{H}_{\mathrm{Ham}} = \begin{pmatrix} H_{\mathrm{Ham}} \\ \mathbf{h}_{\mathrm{red}} \end{pmatrix}. \tag{6.40}$$

Because the new row is already in the row span,

$$\ker(\widetilde{H}_{\mathrm{Ham}}) = \ker(H_{\mathrm{Ham}}), \qquad \operatorname{rank}(\widetilde{H}_{\mathrm{Ham}}) = \operatorname{rank}(H_{\mathrm{Ham}}). \tag{6.41}$$

The codespace and the number of logical bits are unchanged, but a classical Hamiltonian containing one energy term per listed check has different energetics. Redundant checks can therefore be useful: e.g. compare the physics of the 1d Ising model (Section 4) vs. the 2d Ising model (Section 5).

6.7 Canonical forms for check and generator matrices

A parity-check matrix can be simplified by row reduction, removal of redundant rows, and permutation of columns, where a column permutation merely relabels the physical bits. After these

operations it can be written in the canonical form

$$H_{\text{can}} = \left(\mathbb{I}_{(n-k)\times(n-k)} \;\middle|\; A_{(n-k)\times k} \right), \tag{6.42}$$

where A is arbitrary. This reduction generally destroys the LDPC property, even when the original matrix was sparse.

Proposition 6.10 (Canonical generator matrix)**.** For the check matrix in Eq. (6.42), a canonical generator matrix is

$$G_{\text{can}} = \left(A^{\mathsf{T}}_{k\times(n-k)} \;\middle|\; \mathbb{I}_{k\times k} \right). \tag{6.43}$$

Its rows form a basis of $\ker(H_{\text{can}})$, so

$$\ker(H_{\text{can}}) = \operatorname{im}(G^{\mathsf{T}}_{\text{can}}). \tag{6.44}$$

Proof. First,

$$H_{\text{can}} G^{\mathsf{T}}_{\text{can}} = (\mathbb{I}_{n-k} \mid A) \begin{pmatrix} A \\ \mathbb{I}_k \end{pmatrix} = A + A = 0, \tag{6.45}$$

because the arithmetic is over $\mathbb{F}_2$. The identity blocks also imply

$$\mathbf{s}^{\mathsf{T}} H_{\text{can}} = 0 \quad \Longrightarrow \quad \mathbf{s} = 0, \tag{6.46a}$$

$$G^{\mathsf{T}}_{\text{can}} \mathbf{v} = 0 \quad \Longrightarrow \quad \mathbf{v} = 0. \tag{6.46b}$$

Hence H_{can} has rank $n-k$, while G_{can} has rank k. Since $H_{\text{can}} G^{\mathsf{T}}_{\text{can}} = 0$, one has

$$\operatorname{im}(G^{\mathsf{T}}_{\text{can}}) \subseteq \ker(H_{\text{can}}). \tag{6.47}$$

Both spaces have dimension k, so the inclusion is an equality. □

7 Pauli stabilizer codes

7.1 From classical parity checks to Pauli Hamiltonians

To transition from classical to quantum code design, let us first rewrite a classical linear code in quantum language. Consider the three-bit repetition code with parity-check matrix

$$H = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}. \tag{7.1}$$

The corresponding quantum Hamiltonian is

$$H_0 = -Z_1 Z_2 - Z_2 Z_3. \tag{7.2}$$

The classical codewords are generated by

$$G = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}, \tag{7.3}$$

while the classical codespace is

$$C_{\mathrm{cl}} := \ker_{\mathbb{F}_2}(H) = \mathrm{im}_{\mathbb{F}_2}(G^{\mathsf{T}}). \tag{7.4}$$

They define the Hilbert-space codespace

$$\mathcal{C} := \mathrm{span}\left\{|\mathbf{x}\rangle : \mathbf{x} \in C_{\mathrm{cl}}\right\} = \mathrm{span}\{|000\rangle, |111\rangle\}. \tag{7.5}$$

This is precisely the ground-state space of (7.2). The generator of the classical code becomes the quantum symmetry

$$X_L = X_1 X_2 X_3, \qquad [H_0, X_L] = 0. \tag{7.6}$$

This is the bit-flip repetition code that we have already seen (Example 1.1).

More generally, a parity-check matrix $H \in \mathbb{F}_2^{m \times n}$ determines the commuting Hamiltonian

$$H_0 = -\sum_{c=1}^{m} \prod_{\substack{b=1 \\ H_{cb}=1}}^{n} Z_b. \tag{7.7}$$

If the rows of $G \in \mathbb{F}_2^{k \times n}$ generate $\ker(H)$, then each row gives a symmetry

$$X_{L,\alpha} := \prod_{\substack{b=1 \\ G_{\alpha b}=1}}^{n} X_b. \tag{7.8}$$

It is a symmetry because $[H_0, X_{L,\alpha}] = 0$. Indeed, the parity of the overlap between check c and

generator α is

$$\sum_{b=1}^{n} H_{cb} G_{\alpha b} = (H G^{\mathsf{T}})_{c\alpha} = 0 \qquad \text{in } \mathbb{F}_2, \tag{7.9}$$

so the corresponding Z- and X-type Pauli strings commute.

A natural generalization is to look for quantum codes defined by Hamiltonians

$$H_0 = -\sum_i S_i, \tag{7.10}$$

where each S_i is a Pauli string and they mutually commute: $[S_i, S_j] = 0$. This leads to the theory of Pauli stabilizer codes that we develop in this lecture. The Shor code (Section 2) already fits into this framework; the goal of this lecture is to build a formalism that makes the construction easy to generalize.

7.2 Groups and the Pauli group

Definition 7.1 (Group)**.** A *group* is a set G equipped with a multiplication satisfying

$$\text{closed:} \qquad g_1, g_2 \in G \Longrightarrow g_1 g_2 \in G, \tag{7.11a}$$

$$\text{identity:} \qquad \exists\, 1 \in G \text{ such that } 1g = g1 = g, \tag{7.11b}$$

$$\text{inverses:} \qquad \forall g \in G\ \exists\, g^{-1} \in G \text{ such that } g g^{-1} = g^{-1} g = 1, \tag{7.11c}$$

$$\text{associativity:} \qquad (g_1 g_2) g_3 = g_1 (g_2 g_3). \tag{7.11d}$$

Definition 7.2 (Abelian group)**.** A group is *Abelian* if

$$g_1 g_2 = g_2 g_1 \qquad \text{for every } g_1, g_2 \in G. \tag{7.12}$$

Definition 7.3 (Pauli group)**.** The n-qubit Pauli group is

$$\mathcal{P}_n := \{\omega P_1 \otimes \cdots \otimes P_n : \omega \in \{+1, -1, +\mathrm{i}, -\mathrm{i}\},\ P_b \in \{\mathbb{I}, X, Y, Z\}\}\,. \tag{7.13}$$

The Pauli group is non-Abelian. On one qubit,

$$X^2 = Y^2 = Z^2 = \mathbb{I}, \qquad \mathbb{I}X = X\mathbb{I}, \qquad XZ = -\mathrm{i}Y, \qquad ZX = \mathrm{i}Y, \tag{7.14}$$

and analogous relations hold for the other pairs. On different qubits the operators commute. Consequently, any two Pauli strings either commute or anticommute:

$$PQ = \pm QP \qquad \text{for every } P, Q \in \mathcal{P}_n. \tag{7.15}$$

Definition 7.4 (Pauli stabilizer code)**.** A *Pauli stabilizer code* is a quantum code whose codespace is

$$\mathcal{C} = \left\{ |\psi\rangle \in (\mathbb{C}^2)^{\otimes n} : S\,|\psi\rangle = |\psi\rangle \text{ for every } S \in \mathcal{S} \right\}, \tag{7.16}$$

where $\mathcal{S} \leq \mathcal{P}_n$ is a subgroup called the *stabilizer group*.

A key question will be how to choose $\mathcal{S}$ so that the resulting code has useful error-correcting properties. We begin with two elementary constraints that follow directly from the definition.

Proposition 7.5 (The stabilizer group is Abelian)**.** If $\mathcal{C} \neq \{0\}$, then every pair of elements of $\mathcal{S}$ commutes.

Proof. Take $S_1, S_2 \in \mathcal{S}$ and a nonzero $|\psi\rangle \in \mathcal{C}$. Both operators stabilize the codeword, so

$$S_1 S_2\,|\psi\rangle = |\psi\rangle = S_2 S_1\,|\psi\rangle\,. \tag{7.17}$$

Any two Pauli strings commute or anticommute. If they anticommuted, then Eq. (7.17) would imply

$$|\psi\rangle = S_1 S_2\,|\psi\rangle = -S_2 S_1\,|\psi\rangle = -\,|\psi\rangle\,, \tag{7.18}$$

which is impossible for a nonzero state. Therefore $S_1 S_2 = S_2 S_1$. □

Proposition 7.6 (Nontrivial phases are not stabilizers)**.** If $S \in \mathcal{S}$, then

$$-S \notin \mathcal{S}, \qquad +\mathrm{i}S \notin \mathcal{S}, \qquad -\mathrm{i}S \notin \mathcal{S}. \tag{7.19}$$

Proof. For every $|\psi\rangle \in \mathcal{C}$, one has

$$S\,|\psi\rangle = |\psi\rangle\,, \qquad (-S)\,|\psi\rangle = -\,|\psi\rangle\,, \qquad (\pm\mathrm{i}S)\,|\psi\rangle = \pm\mathrm{i}\,|\psi\rangle\,. \tag{7.20}$$

Only the first operator fixes the codeword with eigenvalue $+1$. □

In particular, every stabilizer element is Hermitian and obeys

$$S^\dagger = S, \qquad S^2 = \mathbb{I}. \tag{7.21}$$

The classical codes at the beginning of the lecture fit neatly into this framework: their Z-type parity checks generate an Abelian Pauli stabilizer group.

7.3 Independent generators and the dimension of the codespace

Let $S_1, \ldots, S_m$ be independent generators of $\mathcal{S}$. Thus every stabilizer has a unique expression as a product of the generators, and no nonempty product of distinct generators equals the identity:

$$\prod_{j \in A} S_j = \mathbb{I} \qquad \Longrightarrow \qquad A = \varnothing. \tag{7.22}$$

Consequently,

$$\mathcal{S} = \left\{ \prod_{j=1}^{m} S_j^{a_j} : a_j \in \{0, 1\} \right\}, \qquad |\mathcal{S}| = 2^m. \tag{7.23}$$

Proposition 7.7 (Dimension of a stabilizer code)**.** The joint $+1$ eigenspace of m independent commuting stabilizer generators has dimension

$$\dim(\mathcal{C}) = 2^{n-m} =: 2^k, \qquad k = n - m. \tag{7.24}$$

The integer k is the number of logical qubits.

Proof. The projector onto the joint $+1$ eigenspace is

$$P_{\mathcal{C}} = \prod_{j=1}^{m} \frac{\mathbb{I} + S_j}{2} = \frac{1}{2^m} \sum_{S \in \mathcal{S}} S. \tag{7.25}$$

Every nonidentity Pauli string has vanishing trace, while

$$\operatorname{tr}(\mathbb{I}) = 2^n. \tag{7.26}$$

Independence guarantees that the identity appears exactly once in the sum in (7.25). Therefore

$$\dim(\mathcal{C}) = \operatorname{tr}(P_{\mathcal{C}}) = \frac{1}{2^m} \sum_{S \in \mathcal{S}} \operatorname{tr}(S) = \frac{2^n}{2^m} = 2^{n-m}, \tag{7.27}$$

which implies (7.24). □

We take $m < n$, so (as we will see) at least one logical qubit is encoded; however, all the formulas we develop here allow $m = n$, which gives a one-dimensional stabilizer-state codespace with $k = 0$. This stores no logical information.

For a syndrome

$$\boldsymbol{\eta} = (\eta_1, \ldots, \eta_m) \in \{+1, -1\}^m, \tag{7.28}$$

define the simultaneous eigenspace and its projector by

$$\mathcal{C}_{\boldsymbol{\eta}} := \{|\psi\rangle : S_j |\psi\rangle = \eta_j |\psi\rangle \text{ for every } j\}, \tag{7.29a}$$

$$P_{\boldsymbol{\eta}} := \prod_{j=1}^{m} \frac{\mathbb{I} + \eta_j S_j}{2}. \tag{7.29b}$$

The projectors are mutually orthogonal and resolve the identity:

$$P_{\boldsymbol{\eta}} P_{\boldsymbol{\eta}'} = \delta_{\boldsymbol{\eta},\boldsymbol{\eta}'} P_{\boldsymbol{\eta}}, \tag{7.30a}$$

$$\sum_{\boldsymbol{\eta}\in\{\pm1\}^m} P_{\boldsymbol{\eta}} = \mathbb{I}. \tag{7.30b}$$

Corollary 7.8 (Dimension of every syndrome sector)**.** Every syndrome sector has dimension

$$\dim(\mathcal{C}_{\boldsymbol{\eta}}) = 2^{n-m} = 2^k. \tag{7.31}$$

Proof. Expanding Eq. (7.29b) gives a signed sum of all elements of $\mathcal{S}$. As in the proof of Proposition 7.7, only the identity has nonzero trace, and its coefficient remains 2^{-m}. Hence

$$\operatorname{tr}(P_{\boldsymbol{\eta}}) = 2^{-m} \operatorname{tr}(\mathbb{I}) = 2^{n-m}. \tag{7.32}$$

Since $\operatorname{tr}(P_{\boldsymbol{\eta}}) = \dim(\mathcal{C}_{\boldsymbol{\eta}})$ this leads to (7.31). □

Proposition 7.9 (Syndrome of a Pauli error)**.** Let $E \in \mathcal{P}_n$, and define signs $\eta_j \in \{+1, -1\}$ by

$$S_j E = \eta_j E S_j. \tag{7.33}$$

If $|\psi\rangle \in \mathcal{C}$, then

$$S_j(E|\psi\rangle) = \eta_j(E|\psi\rangle) \qquad \text{for every } j. \tag{7.34}$$

Thus $E|\psi\rangle \in \mathcal{C}_{\boldsymbol{\eta}}$.

Proof. This is analogous to Proposition 2.2. Since $S_j|\psi\rangle = |\psi\rangle$, $S_j E|\psi\rangle = \eta_j E S_j|\psi\rangle = \eta_j E|\psi\rangle$. □

The preceding corollary shows that every formal syndrome sector is nonzero. For completeness, we now prove an additional fact used below: every such sector is reached from the codespace by an actual Pauli operator.

Lemma 7.10 (Every syndrome has a Pauli representative)**.** For every $\boldsymbol{\eta} \in \{+1, -1\}^m$, there exists a phase-free Pauli string $E_{\boldsymbol{\eta}} \in \{\mathbb{I}, X, Y, Z\}^{\otimes n}$ such that

$$S_j E_{\boldsymbol{\eta}} S_j = \eta_j E_{\boldsymbol{\eta}} \qquad \text{for every } j. \tag{7.35}$$

Moreover,

$$E_{\boldsymbol{\eta}}\mathcal{C} = \mathcal{C}_{\boldsymbol{\eta}}. \tag{7.36}$$

Proof. Choose nonzero states

$$|c\rangle \in \mathcal{C}, \qquad |\boldsymbol{\eta}\rangle \in \mathcal{C}_{\boldsymbol{\eta}}, \tag{7.37}$$

and define the nonzero operator

$$A := |\boldsymbol{\eta}\rangle\langle c| \,. \tag{7.38}$$

Since every stabilizer generator fixes $|c\rangle$ and has eigenvalue η_j on $|\boldsymbol{\eta}\rangle$,

$$S_j A S_j = \eta_j A \qquad \text{for every } j. \tag{7.39}$$

Expand A in the phase-free Pauli operator basis:

$$A = \sum_{P \in \{I,X,Y,Z\}^{\otimes n}} a_P P. \tag{7.40}$$

Conjugation of a Pauli by a Pauli only changes its sign, so for every j and P there is a sign $\chi_j(P) \in \{+1, -1\}$ such that

$$S_j P S_j = \chi_j(P) P. \tag{7.41}$$

Combining Eqs. (7.39) and (7.40), and using linear independence of the Pauli basis, gives

$$a_P \neq 0 \qquad \Longrightarrow \qquad \chi_j(P) = \eta_j \quad \text{for every } j. \tag{7.42}$$

Because $A \neq 0$, at least one coefficient is nonzero. Choose one such Pauli and call it $E_{\boldsymbol{\eta}}$. It obeys Eq. (7.35). Proposition 7.9 then gives

$$E_{\boldsymbol{\eta}}\mathcal{C} \subseteq \mathcal{C}_{\boldsymbol{\eta}}. \tag{7.43}$$

The Pauli operator is unitary, so the space on the left has dimension 2^k. Corollary 7.8 gives the same dimension for the space on the right. The inclusion is therefore an equality. □

7.4 Syndrome measurement and a candidate recovery channel

The commuting observables $S_1, \ldots, S_m$ may be measured in any order. Their outcomes give the syndrome $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_m)$, while preserving all logical information inside the corresponding 2^k-dimensional sector.

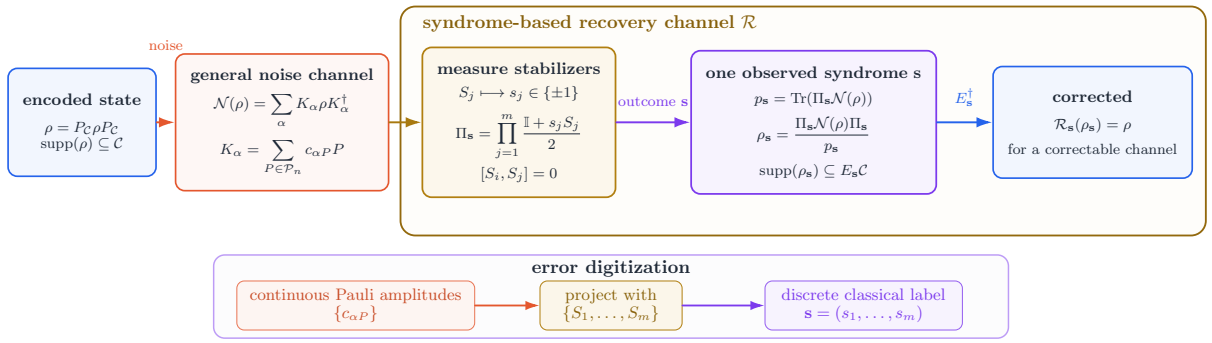

Figure 7.1: As in Section 3, measuring the Pauli stabilizers digitizes the error into a Pauli error, which can be corrected by applying an appropriate choice of $E_{\boldsymbol{\eta}}^\dagger$.

Choose one Pauli representative $E_{\boldsymbol{\eta}}$ for every syndrome. After observing $\boldsymbol{\eta}$, one applies $E_{\boldsymbol{\eta}}^\dagger$. The resulting candidate recovery channel is

$$\mathcal{R}(\rho) := \sum_{\boldsymbol{\eta} \in \{\pm 1\}^m} E_{\boldsymbol{\eta}}^\dagger P_{\boldsymbol{\eta}} \rho P_{\boldsymbol{\eta}} E_{\boldsymbol{\eta}}. \tag{7.44}$$

This placement of the syndrome projectors makes the measurement-and-correction order explicit. Equivalently, since $P_{\boldsymbol{\eta}} = E_{\boldsymbol{\eta}} P_{\mathcal{C}} E_{\boldsymbol{\eta}}^\dagger$, one may write

$$\mathcal{R}(\rho) = \sum_{\boldsymbol{\eta}} P_{\mathcal{C}} E_{\boldsymbol{\eta}}^\dagger \rho E_{\boldsymbol{\eta}} P_{\mathcal{C}}. \tag{7.45}$$

The Kraus operators

$$K_{\boldsymbol{\eta}} := E_{\boldsymbol{\eta}}^\dagger P_{\boldsymbol{\eta}} \tag{7.46}$$

satisfy

$$\sum_{\boldsymbol{\eta}} K_{\boldsymbol{\eta}}^\dagger K_{\boldsymbol{\eta}} = \sum_{\boldsymbol{\eta}} P_{\boldsymbol{\eta}} = \mathbb{I}, \tag{7.47}$$

so $\mathcal{R}$ is a quantum channel. This is illustrated in Figure 7.1.

Suppose an actual Pauli error E has syndrome $\boldsymbol{\eta}$ and $\rho = P_{\mathcal{C}} \rho P_{\mathcal{C}}$. The candidate recovery gives

$$\mathcal{R}(E \rho E^\dagger) = E_{\boldsymbol{\eta}}^\dagger E \rho E^\dagger E_{\boldsymbol{\eta}}. \tag{7.48}$$

Thus syndrome recovery always returns the state to the codespace, but it corrects the logical state only when $E_{\boldsymbol{\eta}}^\dagger E$ acts trivially on the code. We now classify exactly when this happens.

A Pauli error is *detectable by the stabilizer measurement* if at least one syndrome bit is negative:

$$\eta_j = -1 \text{ for at least one } j \qquad \Longleftrightarrow \qquad \{E, S_j\} = 0 \text{ for at least one } j. \tag{7.49}$$

It has zero syndrome if

$$\eta_j = +1 \text{ for every } j \qquad \Longleftrightarrow \qquad [E, S_j] = 0 \text{ for every } j. \tag{7.50}$$

A zero-syndrome operator is undetectable by the stabilizer measurements, but it need not be a harmful logical error. It may instead be a global phase times a stabilizer, in which case its action on the encoded state is trivial. The normalizer formalism below separates these two possibilities.

7.5 The normalizer and logical Pauli operators

Define the Pauli normalizer of the stabilizer group by

$$N(\mathcal{S}) := \{E \in \mathcal{P}_n : [E, S] = 0 \text{ for every } S \in \mathcal{S}\}. \tag{7.51}$$

It is exactly the set of zero-syndrome Pauli operators.

Strictly speaking, Eq. (7.51) defines the centralizer of $\mathcal{S}$ inside $\mathcal{P}_n$. It equals the group-theoretic Pauli normalizer: conjugating one Pauli by another gives $ESE^\dagger = \pm S$, and $-S \notin \mathcal{S}$ by Proposition 7.6. Hence conjugation preserves $\mathcal{S}$ precisely when the operators commute.

Proposition 7.11 (Number of Pauli operators with a fixed syndrome)**.** For every syndrome $\boldsymbol{\eta}$,

$$|\{E \in \mathcal{P}_n : S_j E S_j = \eta_j E \text{ for every } j\}| = 4 \cdot 2^{n+k}. \tag{7.52}$$

In particular,

$$|N(\mathcal{S})| = 4 \cdot 2^{n+k}. \tag{7.53}$$

Proof. There are four choices of global phase and four phase-free Pauli choices on each qubit, so

$$|\mathcal{P}_n| = 4 \cdot 4^n. \tag{7.54}$$

Let $E_{\boldsymbol{\eta}}$ be one representative of syndrome $\boldsymbol{\eta}$. Multiplication by a zero-syndrome Pauli preserves the syndrome, and if F has syndrome $\boldsymbol{\eta}$, then $E_{\boldsymbol{\eta}}^\dagger F$ has zero syndrome. Therefore the full fiber is the coset

$$\{E \in \mathcal{P}_n : E \text{ has syndrome } \boldsymbol{\eta}\} = E_{\boldsymbol{\eta}} N(\mathcal{S}). \tag{7.55}$$

All fibers consequently have cardinality $|N(\mathcal{S})|$. By Lemma 7.10, all $2^m = 2^{n-k}$ syndromes occur. Hence

$$|N(\mathcal{S})| = \frac{|\mathcal{P}_n|}{2^{n-k}} = \frac{4 \cdot 4^n}{2^{n-k}} \tag{7.56}$$

which implies (7.53), and therefore (7.52). □

The count has the useful factorization

$$4 \cdot 2^{n+k} = \underbrace{4}_{\substack{\text{global}\\ \text{phases}}} \times \underbrace{2^{n-k}}_{\text{stabilizers}} \times \underbrace{2^{2k}}_{\substack{\text{logical Pauli}\\ \text{classes}}} . \tag{7.57}$$

Indeed, if $Q \in N(\mathcal{S})$, $S \in \mathcal{S}$, and $|\psi\rangle \in \mathcal{C}$, then

$$QS\,|\psi\rangle = Q\,|\psi\rangle\,. \tag{7.58}$$

Thus multiplying by a stabilizer does not change the encoded action. Global phases are also physically irrelevant because

$$|\psi\rangle \sim e^{i\phi}\,|\psi\rangle\,, \qquad e^{i\phi}\rho e^{-i\phi} = \rho. \tag{7.59}$$

Let

$$\Phi_{\text{ph}} := \{\mathbb{I}, -\mathbb{I}, +i\mathbb{I}, -i\mathbb{I}\} \tag{7.60}$$

be the phase subgroup. The logical Pauli group is the quotient

$$\mathcal{P}_{\text{logical}} := N(\mathcal{S})/(\Phi_{\text{ph}}\mathcal{S}), \tag{7.61}$$

and its cardinality is

$$|\mathcal{P}_{\text{logical}}| = \frac{4 \cdot 2^{n+k}}{4 \cdot 2^{n-k}} = 2^{2k} = 4^k. \tag{7.62}$$

The $4^k = (2^k)^2$ quotient classes are the logical Pauli operators modulo stabilizers and global phases. Choosing one representative of each class gives an operator basis on the 2^k-dimensional codespace. This is a basis of the operator space; of course there are infinitely many linear combinations of such operators that act as valid logical operators.

7.6 Stabilizer equivalence and the Knill–Laflamme condition

Definition 7.12 (Equivalent Pauli errors)**.** Two Pauli errors $E_1, E_2 \in \mathcal{P}_n$ are *equivalent on the code* if they have the same action on every code state up to an overall phase:

$$E_1 P_{\mathcal{C}} = \omega E_2 P_{\mathcal{C}} \qquad \text{for some } \omega \in \{+1, -1, +\mathrm{i}, -\mathrm{i}\} =: \Phi_{\text{ph}}. \tag{7.63}$$

Proposition 7.13 (Stabilizer characterization of equivalent errors)**.** Two Pauli errors are equivalent on the code if and only if for some $S \in \mathcal{S}$ and $\omega \in \Phi_{\text{ph}}$,

$$E_1 = \omega E_2 S. \tag{7.64}$$

Equivalently, $E_2^\dagger E_1 \in \Phi_{\text{ph}}\mathcal{S}$.

Proof. If Eq. (7.64) holds, then

$$E_1 P_{\mathcal{C}} = \omega E_2 S P_{\mathcal{C}} = \omega E_2 P_{\mathcal{C}}, \tag{7.65}$$

so the two errors are equivalent.

Conversely, suppose Eq. (7.63) holds and set

$$Q := \omega^{-1} E_2^\dagger E_1 \in \mathcal{P}_n. \tag{7.66}$$

Then

$$Q P_{\mathcal{C}} = P_{\mathcal{C}}. \tag{7.67}$$

Using the stabilizer expansion of the projector gives

$$Q P_{\mathcal{C}} = \frac{1}{2^{n-k}} \sum_{S \in \mathcal{S}} QS = \frac{1}{2^{n-k}} \sum_{S \in \mathcal{S}} S = P_{\mathcal{C}}. \tag{7.68}$$

The phase-free Pauli matrices are linearly independent. The Pauli terms on the two sides of Eq. (7.68) must therefore match. In particular, $QS_0 = S_1$ for some $S_0, S_1 \in \mathcal{S}$, and hence

$$Q = S_1 S_0^{-1} \in \mathcal{S}. \tag{7.69}$$

Substituting this into Eq. (7.66) proves Eq. (7.64). □

The relation to the Knill–Laflamme condition is especially simple for Pauli errors. Let

$$Q := E_a^\dagger E_b. \tag{7.70}$$

If $Q \notin N(\mathcal{S})$, then it anticommutes with some stabilizer S, and

$$P_{\mathcal{C}} Q P_{\mathcal{C}} = P_{\mathcal{C}} S Q P_{\mathcal{C}} = -P_{\mathcal{C}} Q S P_{\mathcal{C}} = -P_{\mathcal{C}} Q P_{\mathcal{C}}, \tag{7.71}$$

so

$$P_{\mathcal{C}} Q P_{\mathcal{C}} = 0. \tag{7.72}$$

If $Q = \omega S \in \Phi_{\text{ph}}\mathcal{S}$, then

$$P_{\mathcal{C}} Q P_{\mathcal{C}} = \omega P_{\mathcal{C}}. \tag{7.73}$$

The only forbidden case is

$$Q \in N(\mathcal{S}) \setminus (\Phi_{\mathrm{ph}}\mathcal{S}), \tag{7.74}$$

where the two errors have the same syndrome but differ by a nontrivial logical Pauli. Thus a set of Pauli errors is correctable precisely when this third case never occurs for any pair.

Operationally, after observing a syndrome, the decoder must choose a Pauli to apply. Multiplication of that choice by a stabilizer is harmless,

$$E \longmapsto ES, \qquad S \in \mathcal{S}, \tag{7.75}$$

whereas multiplication by a nontrivial logical Pauli is harmful:

$$E \longmapsto EL, \qquad L \in N(\mathcal{S}) \setminus (\Phi_{\mathrm{ph}}\mathcal{S}). \tag{7.76}$$

7.7 Code distance

The weight of a Pauli string is the number of qubits on which it acts nontrivially:

$$\mathrm{wt}(E) := \#\left\{b : E_b \neq I\right\}. \tag{7.77}$$

Definition 7.14 (Distance of a stabilizer code)**.** The code distance is the smallest weight of a nontrivial logical Pauli:

$$d := \min\left\{\mathrm{wt}(L) : L \in N(\mathcal{S}) \setminus (\Phi_{\mathrm{ph}}\mathcal{S})\right\}. \tag{7.78}$$

Proposition 7.15 (Correction radius from the distance)**.** A stabilizer code of distance d corrects every Pauli error of weight at most

$$t := \left\lfloor \frac{d-1}{2} \right\rfloor. \tag{7.79}$$

Proof. Let E_1, E_2 be two Pauli errors with $\mathrm{wt}(E_1), \mathrm{wt}(E_2) \leq t$, and set

$$Q := E_1^\dagger E_2. \tag{7.80}$$

Its weight obeys

$$\mathrm{wt}(Q) \leq \mathrm{wt}(E_1) + \mathrm{wt}(E_2) \leq 2t < d. \tag{7.81}$$

By the definition of d, Q cannot belong to $N(\mathcal{S}) \setminus (\Phi_{\mathrm{ph}}\mathcal{S})$. There are only two possibilities.

If $Q \notin N(\mathcal{S})$, then the errors have different syndromes and Eq. (7.72) gives

$$P_{\mathcal{C}} E_1^\dagger E_2 P_{\mathcal{C}} = 0. \tag{7.82}$$

If $Q \in N(\mathcal{S})$, then the weight bound forces $Q \in \Phi_{\mathrm{ph}}\mathcal{S}$, so

$$P_{\mathcal{C}} E_1^{\dagger} E_2 P_{\mathcal{C}} = \omega_{12} P_{\mathcal{C}} \tag{7.83}$$

for some phase ω_{12}. In either case the Knill–Laflamme condition holds for all errors of weight at most t, proving the claim. □

The first part of this proof is identical to the classical distance argument: two sufficiently small errors cannot differ by a logical operator. The new quantum feature is that errors may differ by a stabilizer and still be perfectly correctable. The decoder only needs to identify an error up to stabilizer equivalence.

7.8 The Shor code revisited

Let us briefly revisit the Shor code to see how the abstract statements fit together. Its stabilizer group is generated by the eight operators

$$\begin{aligned} S_1 &= X_1X_2X_3X_4X_5X_6, & S_2 &= X_4X_5X_6X_7X_8X_9, \\ S_3 &= Z_1Z_2, & S_4 &= Z_2Z_3, \\ S_5 &= Z_4Z_5, & S_6 &= Z_5Z_6, \\ S_7 &= Z_7Z_8, & S_8 &= Z_8Z_9. \end{aligned} \tag{7.84}$$

They are independent, so the codespace encodes $k = 9 - 8 = 1$ logical qubits. Up to global phase, the normalizer is generated by $\mathcal{S}$ together with the logical operators

$$X_L = X_1X_2X_3, \qquad Z_L = Z_1Z_4Z_7. \tag{7.85}$$

Logical representatives are not unique. For example, multiplying X_L by the first X-type stabilizer gives

$$(X_1X_2X_3)(X_1X_2X_3X_4X_5X_6) = X_4X_5X_6, \tag{7.86}$$

so $X_1X_2X_3$ and $X_4X_5X_6$ represent the same logical Pauli. By checking the size of all the stabilizer-equivalent logical Paulis in $N(\mathcal{S}) \setminus \mathcal{S}$, one can show that the Shor code has $d = 3$, and therefore corrects every single-qubit Pauli error, as we claimed in Section 2.

Not all single-qubit errors need to be distinguishable. Since $Z_1Z_2 \in \mathcal{S}$,

$$Z_1(Z_1Z_2) = Z_2, \qquad Z_1 P_{\mathcal{C}} = Z_2 P_{\mathcal{C}}. \tag{7.87}$$

Thus Z_1 and Z_2 have the same syndrome and the same action on every encoded state. The same correction representative works for both errors, which is precisely stabilizer degeneracy in the language developed above.

8 CSS codes

Today we will describe Pauli stabilizer codes using a linear-algebra formalism much like the one used for classical parity-check codes. Recall that a Pauli stabilizer code has codespace

$$\mathcal{C} = \{|\psi\rangle : S|\psi\rangle = |\psi\rangle \text{ for every } S \in \mathcal{S}\}, \qquad \mathcal{S} \leq \mathcal{P}_n, \qquad -\mathbb{I} \notin \mathcal{S}, \tag{8.1}$$

where the stabilizer group is generated by commuting Pauli operators. For error correction, the central question is which Pauli error gives which syndrome, or equivalently which commutation relations it has with the stabilizers. These data admit a simple binary description.

8.1 Binary representation of Pauli operators

Definition 8.1 (Binary Pauli vector)**.** For one qubit, define a map

$$\varphi : \mathcal{P}_1 \longrightarrow \mathbb{F}_2^2 \tag{8.2}$$

by

$$\varphi(\mathbb{I}) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad \varphi(X) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad \varphi(Y) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \qquad \varphi(Z) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{8.3}$$

The map ignores global phases:

$$\varphi(\omega Q) = \varphi(Q) \qquad \text{for } \omega \in \{+1, -1, +\mathrm{i}, -\mathrm{i}\}. \tag{8.4}$$

For n qubits, write

$$\varphi(Q) = \begin{pmatrix} \mathbf{q}^{(x)} \\ \mathbf{q}^{(z)} \end{pmatrix} \in \mathbb{F}_2^{2n}, \qquad \mathbf{q}^{(x)}, \mathbf{q}^{(z)} \in \mathbb{F}_2^n, \tag{8.5}$$

where the first n bits record the locations of X factors and the last n bits record the locations of Z factors. A Y contributes one to both blocks.

For example, on two qubits,

$$\varphi(X_1) = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \qquad \varphi(Y_1 Z_2) = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}. \tag{8.6}$$

Proposition 8.2 (Pauli multiplication becomes binary addition)**.** For all $Q_1, Q_2 \in \mathcal{P}_n$,

$$\varphi(Q_1 Q_2) = \varphi(Q_1) + \varphi(Q_2), \tag{8.7}$$

where the addition on the right is in $\mathbb{F}_2^{2n}$.

The proposition follows by checking the one-qubit Pauli multiplication table. Thus, after quotienting by global phases, φ is a group isomorphism from Pauli multiplication to binary-vector addition.

Define the $2n \times 2n$ matrix

$$J := \begin{pmatrix} 0_{n\times n} & \mathbb{I}_{n\times n} \\ \mathbb{I}_{n\times n} & 0_{n\times n} \end{pmatrix}. \tag{8.8}$$

Proposition 8.3 (Binary commutation test)**.** For $Q_1, Q_2 \in \mathcal{P}_n$,

$$[Q_1, Q_2] = 0 \quad \Longleftrightarrow \quad \varphi(Q_1)^{\mathsf{T}} J \varphi(Q_2) = 0, \tag{8.9a}$$

$$\{Q_1, Q_2\} = 0 \quad \Longleftrightarrow \quad \varphi(Q_1)^{\mathsf{T}} J \varphi(Q_2) = 1. \tag{8.9b}$$

Proof. If

$$\varphi(Q_a) = \begin{pmatrix} \mathbf{q}_a^{(x)} \\ \mathbf{q}_a^{(z)} \end{pmatrix}, \qquad a \in \{1, 2\}, \tag{8.10}$$

then

$$\varphi(Q_1)^{\mathsf{T}} J \varphi(Q_2) = \mathbf{q}_1^{(x)} \cdot \mathbf{q}_2^{(z)} + \mathbf{q}_1^{(z)} \cdot \mathbf{q}_2^{(x)}. \tag{8.11}$$

This expression counts modulo two the qubits on which the two Pauli strings locally anticommute. An odd number of local anticommutations gives an overall minus sign, while an even number gives an overall plus sign. This proves Eqs. (8.9a) and (8.9b). □

8.2 Stabilizer codes as binary linear codes

Choose stabilizer generators $S_1, \ldots, S_m$ and collect their binary vectors as the rows of the parity-check matrix

$$H := \begin{pmatrix} \varphi(S_1)^{\mathsf{T}} \\ \vdots \\ \varphi(S_m)^{\mathsf{T}} \end{pmatrix} \in \mathbb{F}_2^{m\times 2n}. \tag{8.12}$$

Every entry of HJH^{T} is the binary commutation pairing of two stabilizer generators. Therefore the stabilizer commutation condition is

$$HJH^{\mathsf{T}} = 0. \tag{8.13}$$

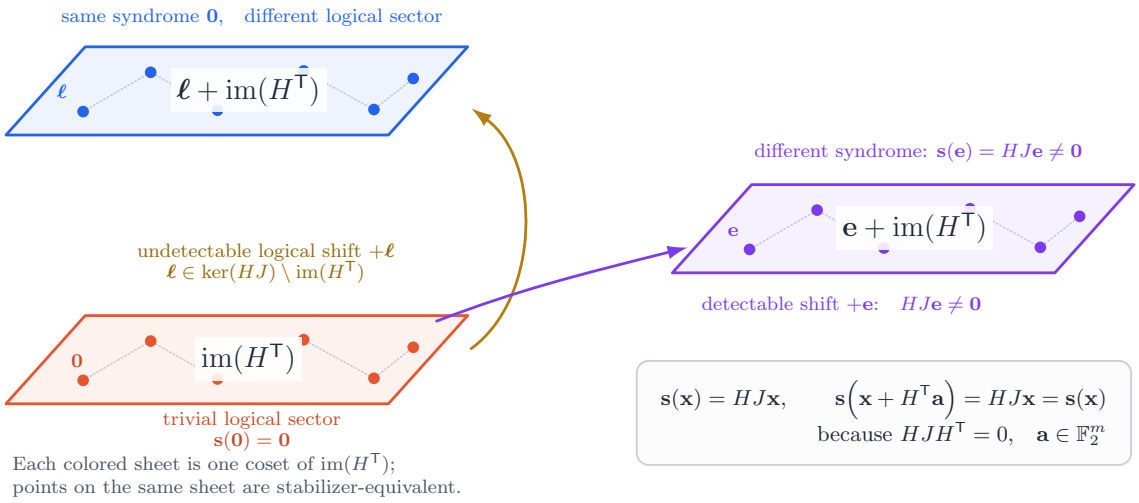

Figure 8.1: Illustration of different syndrome cosets (and logical cosets) in a general Pauli stabilizer code.

The Pauli operators commuting with every stabilizer are represented by

$$\varphi(N(\mathcal{S})) = \ker(HJ) = \left\{\mathbf{v} \in \mathbb{F}_2^{2n} : HJ\mathbf{v} = 0\right\}. \tag{8.14}$$

Meanwhile, products of stabilizer generators are represented by

$$\mathrm{im}(H^\mathsf{T}) = \left\{H^\mathsf{T}\mathbf{a} : \mathbf{a} \in \mathbb{F}_2^m\right\}. \tag{8.15}$$

Consequently, binary logical Pauli operators are the quotient $\ker(HJ)/\mathrm{im}(H^\mathsf{T})$. This is the binary version of identifying Pauli operators that differ by a stabilizer. More generally, all Pauli errors modulo stabilizer equivalence form the error space $\frac{\mathbb{F}_2^{2n}}{\mathrm{im}(H^\mathsf{T})}$. If an error has binary vector $\mathbf{e}$, then its binary syndrome is

$$\mathbf{s} = HJ\mathbf{e}. \tag{8.16}$$

Equation (8.13) implies that the syndrome is independent of the chosen stabilizer-equivalent representative:

$$HJ(\mathbf{e} + H^\mathsf{T}\mathbf{a}) = HJ\mathbf{e} + HJH^\mathsf{T}\mathbf{a} = HJ\mathbf{e}. \tag{8.17}$$

The measured eigenvalue of generator S_a is $(-1)^{s_a}$. Thus the binary syndrome contains exactly the same information as the usual list of stabilizer eigenvalues: see Figure 8.1.

Definition 8.4 (Pauli Hamming weight and reduced weight)**.** For $\mathbf{w} = (\mathbf{w}^{(x)}, \mathbf{w}^{(z)}) \in \mathbb{F}_2^{2n}$, its Pauli

Hamming weight is

$$|\mathbf{w}|_{\mathrm{H}} := \sum_{j=1}^{n} \begin{cases} 1, & (w_j^{(x)}, w_j^{(z)}) \neq (0,0), \\ 0, & (w_j^{(x)}, w_j^{(z)}) = (0,0). \end{cases} \tag{8.18}$$

Its reduced error weight is the smallest weight among all stabilizer-equivalent representatives:

$$|\mathbf{w}|_{\mathrm{red}} := \min_{\mathbf{v} \in \mathrm{im}(H^{\mathsf{T}})} |\mathbf{w} + \mathbf{v}|_{\mathrm{H}}. \tag{8.19}$$

Definition 8.5 (Distance of a stabilizer code)**.** The code distance is

$$d := \min_{\mathbf{w} \in \ker(HJ) \setminus \mathrm{im}(H^{\mathsf{T}})} |\mathbf{w}|_{\mathrm{red}}. \tag{8.20}$$

Thus d is the minimum physical weight of a nontrivial logical Pauli, after optimizing over stabilizer-equivalent representatives.

8.3 The special case of CSS codes

We will mainly study stabilizer codes of a special form.

Definition 8.6 (CSS code)**.** A CSS code has a binary parity-check matrix

$$H = \begin{pmatrix} H_X & 0 \\ 0 & H_Z \end{pmatrix}, \qquad H_X \in \mathbb{F}_2^{m_X \times n}, \qquad H_Z \in \mathbb{F}_2^{m_Z \times n}. \tag{8.21}$$

Thus every stabilizer generator is either purely X-type or purely Z-type.

Example 8.7 (The Shor code)**.** For the nine-qubit Shor code, one may take

$$H_X = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \tag{8.22a}$$

$$H_Z = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}. \tag{8.22b}$$

The first row of H_X, for example, represents $X_1X_2X_3X_4X_5X_6$, while the first row of H_Z represents Z_1Z_2.

8.3.1 Useful properties of CSS codes

Proposition 8.8 (CSS commutation condition)**.** The X- and Z-type stabilizers commute if and only if

$$H_X H_Z^{\mathsf{T}} = 0. \tag{8.23}$$

Proof. For the block form in Eq. (8.21),

$$HJH^{\mathsf{T}} = \begin{pmatrix} H_X & 0 \\ 0 & H_Z \end{pmatrix} \begin{pmatrix} 0 & \mathbb{I} \\ \mathbb{I} & 0 \end{pmatrix} \begin{pmatrix} H_X^{\mathsf{T}} & 0 \\ 0 & H_Z^{\mathsf{T}} \end{pmatrix} = \begin{pmatrix} 0 & H_X H_Z^{\mathsf{T}} \\ H_Z H_X^{\mathsf{T}} & 0 \end{pmatrix}. \tag{8.24}$$

The lower-left block is the transpose of the upper-right block, so Eq. (8.13) is equivalent to Eq. (8.23). □

For a CSS code,

$$HJ = \begin{pmatrix} 0 & H_X \\ H_Z & 0 \end{pmatrix}. \tag{8.25}$$

Therefore the X and Z parts of a logical operator obey separate kernel conditions. After quotienting by stabilizers, the logical-vector spaces are

$$\mathcal{L}_X := \frac{\ker(H_Z)}{\mathrm{im}(H_X^{\mathsf{T}})}, \tag{8.26a}$$

$$\mathcal{L}_Z := \frac{\ker(H_X)}{\mathrm{im}(H_Z^{\mathsf{T}})}. \tag{8.26b}$$

It is convenient to introduce the classical codes

$$C_X := \ker(H_X), \qquad C_Z := \ker(H_Z). \tag{8.27}$$

Since

$$C_X^{\perp} = \mathrm{im}(H_X^{\mathsf{T}}), \qquad C_Z^{\perp} = \mathrm{im}(H_Z^{\mathsf{T}}), \tag{8.28}$$

we may equivalently write

$$\mathcal{L}_X = C_Z / C_X^{\perp}, \qquad \mathcal{L}_Z = C_X / C_Z^{\perp}. \tag{8.29}$$

The commutation condition gives the inclusions $C_X^{\perp} \subseteq C_Z$ and $C_Z^{\perp} \subseteq C_X$ needed for these quotients.

A generator matrix $\widetilde{G}_X$ for C_Z and a generator matrix $\widetilde{G}_Z$ for C_X may be chosen so that

$$\mathrm{im}(\widetilde{G}_X^{\mathsf{T}}) = C_Z, \qquad \mathrm{im}(\widetilde{G}_Z^{\mathsf{T}}) = C_X. \tag{8.30}$$

For the Shor code, one convenient choice is

$$\widetilde{G}_X = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}. \tag{8.31}$$

Only one of these three directions is an independent logical direction. We may take

$$G_X = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{8.32}$$

so that

$$\ker(H_Z) = \mathrm{im}(G_X^{\mathsf{T}}) \oplus \mathrm{im}(H_X^{\mathsf{T}}). \tag{8.33}$$

Here the symbol $\oplus$ means that the vector space $\ker(H_Z)$ is generated by vectors either from $\mathrm{im}(G_X^{\mathsf{T}})$ or $\mathrm{im}(H_X^{\mathsf{T}})$. For example,

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}, \tag{8.34}$$

where the first vector on the right is the chosen logical generator and the second is an X-type stabilizer. Similarly, a Z-logical generator is

$$G_Z = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}. \tag{8.35}$$

We can compare these answers with what we found previously in Section 2; they agree.

8.3.2 Logical operators

Proposition 8.9 (Number of encoded qubits)**.** The CSS codespace has dimension 2^k, where

$$k := \dim \ker(H_X) - \dim \mathrm{im}(H_Z^{\mathsf{T}}) = \dim \ker(H_Z) - \dim \mathrm{im}(H_X^{\mathsf{T}}). \tag{8.36}$$

Equivalently, if

$$m := \mathrm{rank}(H_X) + \mathrm{rank}(H_Z) \tag{8.37}$$

is the number of independent stabilizers, then $k = n - m$.

Proof. Rank–nullity gives

$$2^n = |\ker(H_X)|\,|\mathrm{im}(H_X^{\mathsf{T}})| = |\ker(H_Z)|\,|\mathrm{im}(H_Z^{\mathsf{T}})|. \tag{8.38}$$

Therefore

$$\frac{|\ker(H_X)|}{|\mathrm{im}(H_Z^{\mathsf{T}})|} = \frac{|\ker(H_Z)|}{|\mathrm{im}(H_X^{\mathsf{T}})|} = \frac{2^n}{|\mathrm{im}(H_X^{\mathsf{T}})||\mathrm{im}(H_Z^{\mathsf{T}})|} = 2^{n-m} = 2^k. \tag{8.39}$$

The two quotients in Eqs. (8.26a) and (8.26b) consequently both have dimension k. As in the preceding lecture, m independent stabilizers leave a joint $+1$ eigenspace of dimension $2^{n-m} = 2^k$. □

Proposition 8.10 (Canonical commutation pairing for logical generators)**.** One can choose logical generator matrices

$$G_X, G_Z \in \mathbb{F}_2^{k\times n} \tag{8.40}$$

whose rows represent bases of $\mathcal{L}_X$ and $\mathcal{L}_Z$, respectively, such that

$$G_X G_Z^{\mathsf{T}} = \mathbb{I}_{k\times k}. \tag{8.41}$$

Thus the ath logical X anticommutes with the ath logical Z and commutes with the other logical generators.

Proof. First note that

$$\ker(H_X) = \mathrm{im}(H_X^{\mathsf{T}})^{\perp}, \qquad \ker(H_Z) = \mathrm{im}(H_Z^{\mathsf{T}})^{\perp}. \tag{8.42}$$

Let $\mathbf{v}_Z \in \ker(H_X)$ represent a Z-logical class and let $\mathbf{v}_X \in \ker(H_Z)$ represent an X-logical class. If we replace them by stabilizer-equivalent representatives, then

$$\begin{aligned}&(\mathbf{v}_Z + H_Z^{\mathsf{T}}\mathbf{s}_Z)^{\mathsf{T}}(\mathbf{v}_X + H_X^{\mathsf{T}}\mathbf{s}_X)\\ &\quad = \mathbf{v}_Z^{\mathsf{T}}\mathbf{v}_X + \mathbf{s}_Z^{\mathsf{T}}H_Z\mathbf{v}_X + \mathbf{v}_Z^{\mathsf{T}}H_X^{\mathsf{T}}\mathbf{s}_X + \mathbf{s}_Z^{\mathsf{T}}H_Z H_X^{\mathsf{T}}\mathbf{s}_X = \mathbf{v}_Z^{\mathsf{T}}\mathbf{v}_X.\end{aligned} \tag{8.43}$$

Here the middle terms vanish because $H_Z\mathbf{v}_X = 0$ and $H_X\mathbf{v}_Z = 0$, while the final term vanishes by Eq. (8.23). The dot product therefore defines a pairing

$$\mathcal{L}_X \times \mathcal{L}_Z \longrightarrow \mathbb{F}_2, \qquad ([\mathbf{v}_X], [\mathbf{v}_Z]) \longmapsto \mathbf{v}_X^{\mathsf{T}}\mathbf{v}_Z. \tag{8.44}$$

This pairing is nondegenerate. Indeed, if $\mathbf{v}_Z^{\mathsf{T}}\mathbf{v}_X = 0$ for every $\mathbf{v}_X \in \ker(H_Z)$, then

$$\mathbf{v}_Z \in \ker(H_Z)^{\perp} = \mathrm{im}(H_Z^{\mathsf{T}}), \tag{8.45}$$

so $\mathbf{v}_Z$ represents the trivial Z-logical class. The same argument with X and Z exchanged proves nondegeneracy in the other slot.

Here is the explicit Gaussian-elimination step. Choose arbitrary matrices $\widetilde{G}_X, \widetilde{G}_Z \in \mathbb{F}_2^{k\times n}$ whose

rows represent bases of $\mathcal{L}_X$ and $\mathcal{L}_Z$, and form their pairing matrix

$$M := \widetilde{G}_X \widetilde{G}_Z^{\mathsf{T}} \in \mathbb{F}_2^{k\times k}. \tag{8.46}$$

The matrix M must be invertible. Otherwise there would be a nonzero coefficient vector $\mathbf{a} \in \mathbb{F}_2^k$ with $\mathbf{a}^{\mathsf{T}} M = 0$. The nontrivial logical class represented by $\mathbf{a}^{\mathsf{T}} \widetilde{G}_X$ would then pair to zero with every row of $\widetilde{G}_Z$, and hence with every Z-logical class, contradicting nondegeneracy. Row-reducing the invertible matrix M to the identity is equivalent to replacing the X-logical basis by

$$G_X := M^{-1} \widetilde{G}_X, \qquad G_Z := \widetilde{G}_Z. \tag{8.47}$$

Then

$$G_X G_Z^{\mathsf{T}} = M^{-1} \widetilde{G}_X \widetilde{G}_Z^{\mathsf{T}} = M^{-1} M = \mathbb{I}_{k\times k}, \tag{8.48}$$

which is Eq. (8.41). □

The noncommuting logical X and Z operators are essential: they allow us to manipulate the encoded qubits rather than merely preserve them. For the Shor code, (8.32) and (8.35) give

$$X_L = X_1 X_2 X_3, \qquad Z_L = Z_1 Z_4 Z_7. \tag{8.49}$$

Their binary representatives have dot product one, so they anticommute. Because the pairing is independent of stabilizer representatives, any stabilizer-equivalent representatives also anticommute.

Definition 8.11 (X and Z distances)**.** For a CSS code, define

$$d_X := \min_{\mathbf{v}\in \ker(H_Z)\setminus \mathrm{im}(H_X^{\mathsf{T}})} |\mathbf{v}|, \tag{8.50a}$$

$$d_Z := \min_{\mathbf{v}\in \ker(H_X)\setminus \mathrm{im}(H_Z^{\mathsf{T}})} |\mathbf{v}|. \tag{8.50b}$$

The full code distance is

$$d = \min(d_X, d_Z). \tag{8.51}$$

Finding some logical generator matrices G_X and G_Z is a straightforward linear-algebra problem. Finding the smallest-weight nontrivial logical operator, and hence determining d_X or d_Z, is generally much harder numerically.

For quantum codes, we write a code with n physical qubits, k logical qubits, and code distance d as an $[[n, k, d]]$ code.

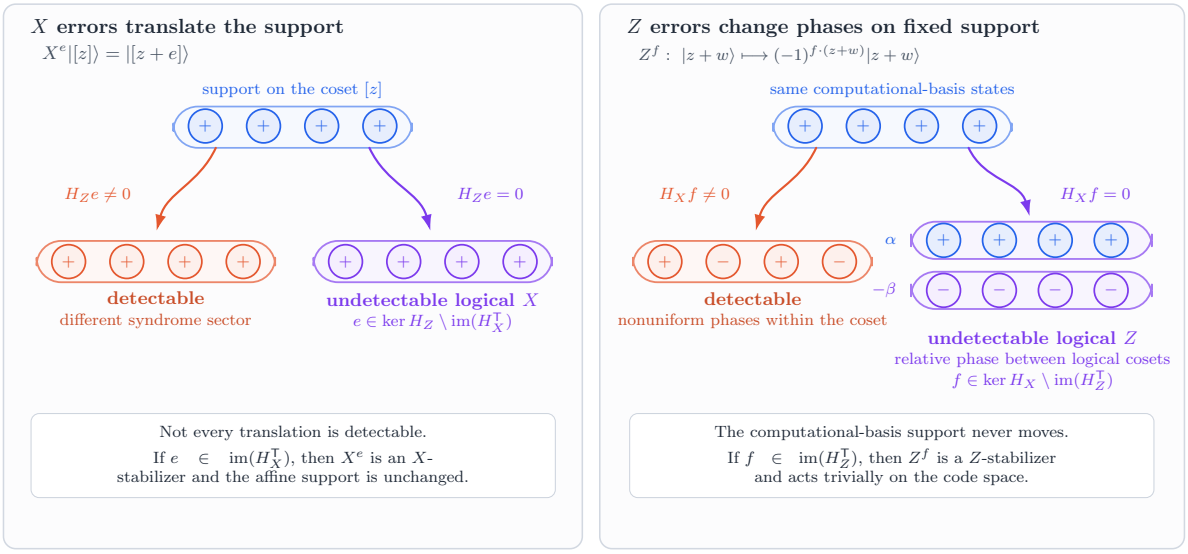

Figure 8.2: Action of X vs. Z errors on (8.53).

8.3.3 Logical codewords as coset states

We have now developed enough formalism to begin studying practical quantum codes. The key will be to use the binary formalism rather than manipulate Pauli strings directly. Nevertheless, it is sometimes useful to think about the wave functions of their codewords.

For $\mathbf{z} \in \mathbb{F}_2^n$, choose the computational basis so that

$$Z_j |\mathbf{z}\rangle = (1 - 2z_j) |\mathbf{z}\rangle = (-1)^{z_j} |\mathbf{z}\rangle . \tag{8.52}$$

For a coset $[\mathbf{z}] \in C_Z/C_X^\perp = \ker(H_Z)/\mathrm{im}(H_X^\mathsf{T})$, define the logical codeword

$$|[\mathbf{z}]\rangle := \frac{1}{\sqrt{|\mathrm{im}(H_X^\mathsf{T})|}} \sum_{\mathbf{w}\in\mathrm{im}(H_X^\mathsf{T})} |\mathbf{z}+\mathbf{w}\rangle . \tag{8.53}$$

There are 2^k such cosets and hence 2^k logical basis states.

The stabilizer conditions are transparent in this representation. An X-type stabilizer translates every summation variable by an element of $\mathrm{im}(H_X^\mathsf{T})$ and therefore only permutes the terms in Eq. (8.53). A Z-type stabilizer labeled by $\mathbf{b} \in \mathrm{im}(H_Z^\mathsf{T})$ contributes the phase

$$(-1)^{\mathbf{b}^\mathsf{T}(\mathbf{z}+\mathbf{w})} = 1, \tag{8.54}$$

because $\mathbf{z} \in \ker(H_Z)$ and $\mathbf{w} \in \mathrm{im}(H_X^\mathsf{T}) \subseteq \ker(H_Z)$. Distinct logical cosets are disjoint, so the states in Eq. (8.53) are orthonormal.

As illustrated in Figure 8.2, an X error translates the entire coset. If its binary vector $\mathbf{e}$ obeys

$H_Z\mathbf{e} \neq 0$, it moves the state into a detectable syndrome sector. If instead

$$\mathbf{e} \in \ker(H_Z) \setminus \mathrm{im}(H_X^{\mathsf{T}}), \tag{8.55}$$

it moves one logical coset to a different logical coset and is an undetectable logical X error.

A Z error leaves the computational-basis labels in the same coset but changes their phases. For a binary vector $\mathbf{f}$,

$$Z^{\mathbf{f}} \left|[\mathbf{z}]\right\rangle = \frac{1}{\sqrt{|\mathrm{im}(H_X^{\mathsf{T}})|}} \sum_{\mathbf{w}\in\mathrm{im}(H_X^{\mathsf{T}})} (-1)^{\mathbf{f}^{\mathsf{T}}(\mathbf{z}+\mathbf{w})} \left|\mathbf{z}+\mathbf{w}\right\rangle. \tag{8.56}$$

If $H_X\mathbf{f} \neq 0$, these phases vary along at least one X-stabilizer direction and the error is detectable. If

$$\mathbf{f} \in \ker(H_X) \setminus \mathrm{im}(H_Z^{\mathsf{T}}), \tag{8.57}$$

the phase is cosetwise constant but logical-label dependent: an undetectable logical Z error.

9 Toric code

9.1 The toric code on a square lattice

We are now ready to introduce a quantum analogue of the one-dimensional Ising model. The toric code will be a workhorse for our study of quantum codes. Consider an $L \times L$ square lattice with periodic boundary conditions, so that the lattice forms a two-dimensional torus. Let V, E, and F denote its sets of vertices, edges, and faces, respectively. We place one physical qubit on every edge $e \in E$: see Figure 9.1.

For every vertex $v \in V$, define an X-type check supported on the four edges incident to v. For every face $f \in F$, define a Z-type check supported on the four edges in the boundary of f:

$$X^{\mathrm{d}v} := \prod_{e \ni v} X_e, \tag{9.1a}$$

$$Z^{\partial f} := \prod_{e \in \partial f} Z_e. \tag{9.1b}$$

The support of $X^{\mathrm{d}v}$ and the support of $Z^{\partial f}$ overlap on either zero or two edges. Since $X_e Z_e = -Z_e X_e$ on a shared edge, the two possible minus signs cancel whenever the supports overlap. Therefore

$$[X^{\mathrm{d}v}, Z^{\partial f}] = 0 \qquad \text{for every } v \in V \text{ and } f \in F. \tag{9.2}$$

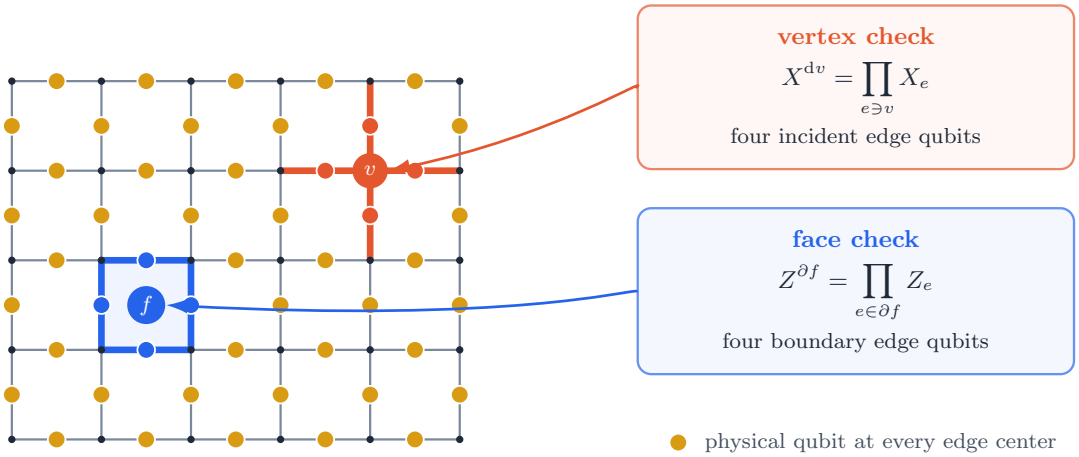

Figure 9.1: Setup of X- and Z-checks in the toric code.

The toric-code codespace is the simultaneous +1 eigenspace of all these commuting checks:

$$\mathcal{C}_{\mathrm{TC}} = \left\{ |\psi\rangle : X^{\mathrm{d}v} |\psi\rangle = |\psi\rangle \ \forall v, \quad Z^{\partial f} |\psi\rangle = |\psi\rangle \ \forall f \right\}. \tag{9.3}$$

Two key advantages of the toric code are its geometric locality in two spatial dimensions and its scalable distance, $d \sim \sqrt{n}$. The checks are local on the torus. A literal periodic identification is not local in a planar device, however, and we will later replace the torus by a planar surface with suitable boundaries.

9.2 Discrete vector calculus and the parity-check matrices

Before studying error correction in two-dimensional codes, we will spend some time understanding where this construction comes from. The basic mathematics is a discrete version of vector calculus.

Choose orientations for the edges and faces. A function G_v on vertices is a discrete 0-form, a vector field u_e on edges is a discrete 1-form, and a dual function ϕ_f on faces is a discrete 2-form. Over the real numbers, the discrete gradient and curl are linear maps

$$\mathrm{d}_0 : \mathbb{R}^V \longrightarrow \mathbb{R}^E, \qquad u = \mathrm{d}_0 G \sim \nabla G, \tag{9.4a}$$

$$\mathrm{d}_1 : \mathbb{R}^E \longrightarrow \mathbb{R}^F, \qquad \phi = \mathrm{d}_1 u \sim \partial_x u_y - \partial_y u_x. \tag{9.4b}$$

The familiar identity that the curl of a gradient vanishes becomes

$$\mathrm{d}_1 \mathrm{d}_0 = 0. \tag{9.5}$$

For one consistent orientation convention, the matrix elements of these maps are

$$\langle e|\mathrm{d}_0|v\rangle = \begin{cases} +1, & e \text{ is oriented away from } v, \\ -1, & e \text{ is oriented toward } v, \\ 0, & e \text{ is not incident to } v, \end{cases} \tag{9.6a}$$

$$\langle f|\mathrm{d}_1|e\rangle = \begin{cases} +1, & e \subset \partial f \text{ and its orientation agrees with } \partial f, \\ -1, & e \subset \partial f \text{ and its orientation opposes } \partial f, \\ 0, & e \not\subset \partial f. \end{cases} \tag{9.6b}$$

See Figure 9.2 for an illustration.

These integer-valued matrices may be reduced modulo two. Over $\mathbb{F}_2$ the signs disappear, and the matrix elements simply record incidence:

$$\langle e|\mathrm{d}_0|v\rangle = \begin{cases} 1, & e \text{ is incident to } v, \\ 0, & \text{otherwise}, \end{cases} \tag{9.7a}$$

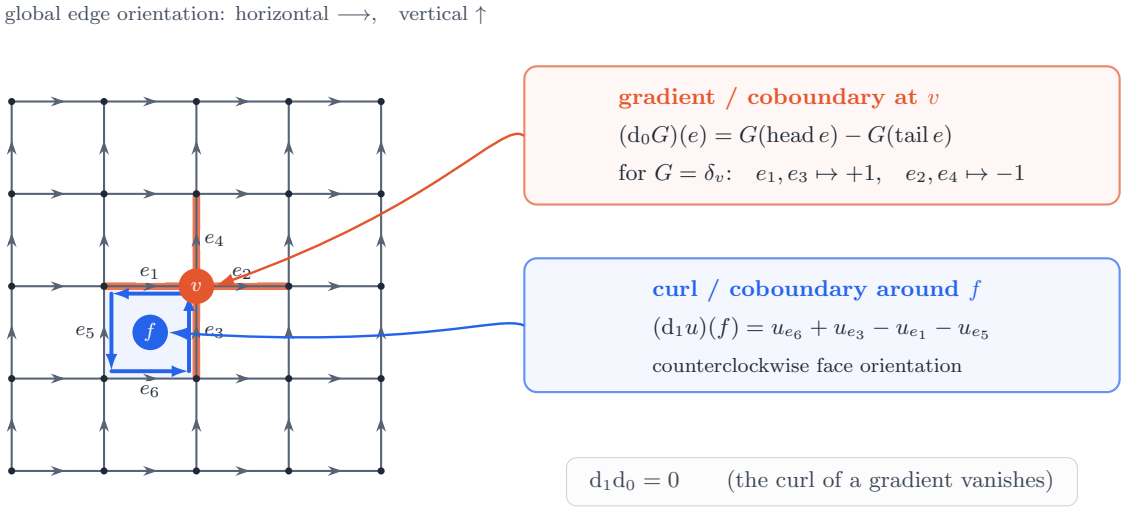

Figure 9.2: Illustration of d_0 and d_1 as discretizations of gradient and curl on a two-dimensional square lattice.

$$\langle f|\mathrm{d}_1|e\rangle = \begin{cases} 1, & e \subset \partial f, \\ 0, & \text{otherwise.} \end{cases} \tag{9.7b}$$

Vertices label X-checks, edges label qubits, and faces label Z-checks. Consequently,

$$\mathrm{d}_0 = H_X^{\mathsf{T}}, \tag{9.8a}$$

$$\mathrm{d}_1 = H_Z, \tag{9.8b}$$

$$H_Z H_X^{\mathsf{T}} = \mathrm{d}_1\mathrm{d}_0 = 0. \tag{9.8c}$$

Thus the elementary vector-calculus identity "curl grad equals zero" is precisely the CSS commutation condition.

9.3 Cohomology and logical X operators

Recall that the X-type logical operators of a CSS code are parameterized by

$$\mathcal{L}_X = \frac{\ker(H_Z)}{\mathrm{im}(H_X^{\mathsf{T}})} = \frac{\ker(\mathrm{d}_1)}{\mathrm{im}(\mathrm{d}_0)}. \tag{9.9}$$

This quotient is a standard object in mathematics.

Definition 9.1 (Cochain complex)**.** A cochain complex over $\mathbb{F}_2$ is a sequence of vector spaces and linear maps

$$V_0 \xrightarrow{\mathrm{d}_0} V_1 \xrightarrow{\mathrm{d}_1} V_2 \tag{9.10}$$

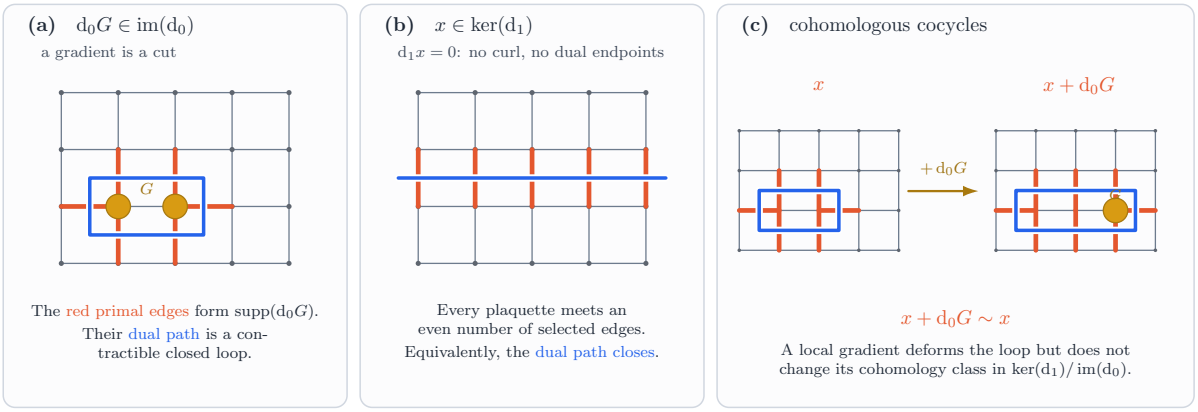

Figure 9.3: Illustration of elements of $\ker(d_1)$ and $\mathrm{im}(d_0)$.

that obey $d_1 d_0 = 0$.

Definition 9.2 (First cohomology group)**.** The first cohomology group of the complex in Eq. (9.10) is

$$H^1 := \frac{\ker(d_1)}{\mathrm{im}(d_0)}. \tag{9.11}$$

A vector $x \in \mathbb{F}_2^E$ may be identified with its support, namely the subset of edges on which $x_e = 1$. Addition of vectors corresponds to symmetric difference of their supports. The condition $x \in \ker(d_1)$ says that every face boundary contains an even number of selected edges. Equivalently, when the selected edges are drawn on the dual lattice, they form a union of closed loops. These are the discrete vector fields with no curl.

An element of $\mathrm{im}(d_0)$ is a discrete gradient. In the code, it is the support of a product of vertex stabilizers. On the dual lattice, such a support is a boundary. Therefore two curl-free configurations x_1 and x_2 represent the same cohomology class exactly when

$$x_1 \sim x_2 \quad \Longleftrightarrow \quad x_1 = x_2 + d_0 G \quad \text{for some } G \in V_0. \tag{9.12}$$

The quotient H^1 is therefore the space of stabilizer-equivalence classes of undetectable X operators: see Figure 9.3.

If $\dim H^1 = k$, then there are 2^k distinct cohomology classes. The dimension, rather than the cardinality of the quotient, is the number of independent logical X generators and hence the number of encoded qubits.

Theorem 9.3 (First cohomology of the torus)**.** For the square lattice with periodic boundary conditions,

$$H^1(T^2; \mathbb{F}_2) \cong \mathbb{F}_2^2. \tag{9.13}$$

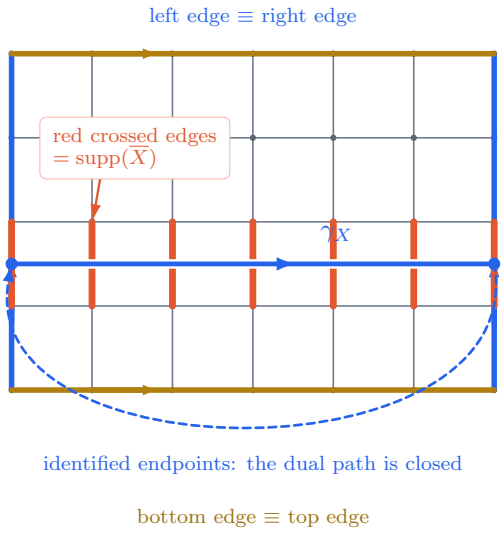

Figure 9.4: An X-logical, i.e. non-trivial element of H^1, is a cocycle that wraps around the periodic boundary conditions on the torus.

In particular, the toric code encodes $k = 2$ logical qubits.

Proof. See Figure 9.4. A curl-free dual-lattice loop can be deformed by adding boundaries, which are precisely the elements of $\mathrm{im}(\mathrm{d}_0)$. Contractible loops can therefore be removed. What remains is characterized by two winding parities: whether the configuration winds around the two independent cycles of the torus. A horizontal winding loop and a vertical winding loop generate these two independent classes; their sum gives the fourth class. □

On an $L \times L$ lattice there are L^2 horizontal edges and L^2 vertical edges, so

$$n = |E| = 2L^2. \tag{9.14}$$

The shortest nontrivial dual-lattice loop has length L. Consequently,

$$d_X = L = \sqrt{\frac{n}{2}}. \tag{9.15}$$

There is also a useful algebraic check on $k = 2$. Every edge appears twice in the product of all vertex checks and twice in the product of all face checks, so

$$\prod_{v \in V} X^{\mathrm{d}v} = \mathbb{I}, \tag{9.16a}$$

$$\prod_{f \in F} Z^{\partial f} = \mathbb{I}. \tag{9.16b}$$

For the connected square torus these are the only relations of their respective types, so $\operatorname{rank} H_X =$

$\operatorname{rank} H_Z = L^2 - 1$ and

$$k = n - \operatorname{rank} H_X - \operatorname{rank} H_Z = 2. \tag{9.17}$$

9.4 Homology and logical Z operators

There is a parallel description of the Z-type logical operators. Reverse the cochain complex and transpose its maps.

Definition 9.4 (Chain complex)**.** The chain complex dual to Eq. (9.10) is

$$V_2 \xrightarrow{\partial_2} V_1 \xrightarrow{\partial_1} V_0, \qquad \partial_{j+1} := \mathrm{d}_j^{\mathsf{T}}. \tag{9.18}$$

It obeys

$$\partial_1 \partial_2 = 0. \tag{9.19}$$

Definition 9.5 (First homology group)**.** The first homology group is

$$\mathrm{H}_1 := \frac{\ker(\partial_1)}{\operatorname{im}(\partial_2)}. \tag{9.20}$$

A vector $z \in V_1 \cong \mathbb{F}_2^E$ is a string on the original lattice. The boundary $\partial_1 z$ is the set of endpoints of that string. In the code, if e_Z is the support vector of a Z error, then its X-syndrome is

$$s_X = H_X e_Z = \partial_1 e_Z. \tag{9.21}$$

Thus an open string has nonzero syndrome at its endpoints, whereas a closed string lies in $\ker(\partial_1)$. The image $\operatorname{im}(\partial_2)$ consists of boundaries of collections of faces; these are products of face stabilizers. Two closed strings are homologous when they differ by such a boundary: see Figure 9.5. It follows that the Z-logical space is

$$\mathcal{L}_Z = \frac{\ker(H_X)}{\operatorname{im}(H_Z^{\mathsf{T}})} = \frac{\ker(\partial_1)}{\operatorname{im}(\partial_2)} = \mathrm{H}_1. \tag{9.22}$$

As proved by linear algebra in the previous lecture, the dual chain and cochain complexes satisfy

$$\dim \mathrm{H}^1 = \dim \mathrm{H}_1. \tag{9.23}$$

Thus there is one independent logical Z generator for every independent logical X generator.

More precisely, logical commutation defines a natural perfect pairing

$$\mathrm{H}^1 \times \mathrm{H}_1 \longrightarrow \mathbb{F}_2, \qquad ([x], [z]) \longmapsto x^{\mathsf{T}} z. \tag{9.24}$$

It is well defined because adding an exact cocycle or a boundary does not change the overlap. Hence

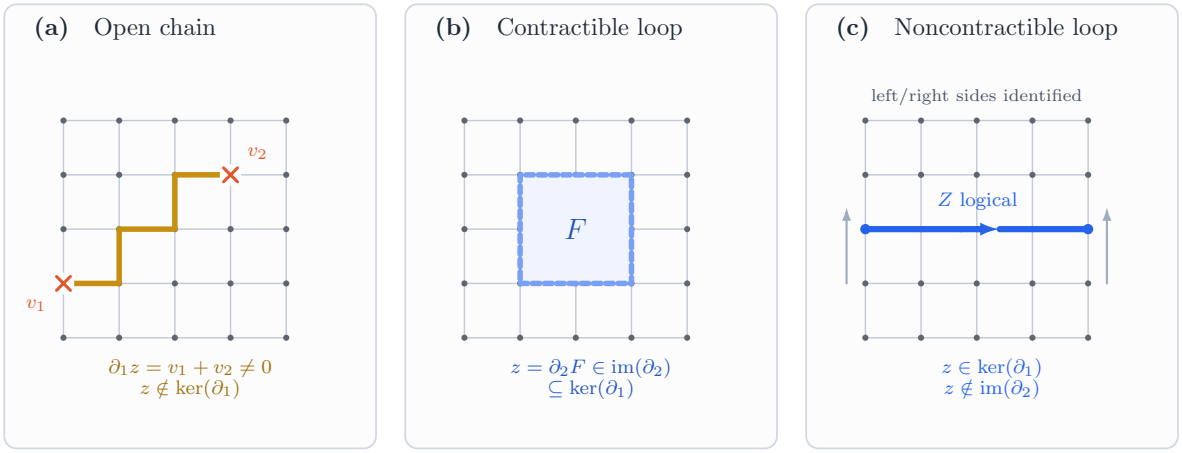

Figure 9.5: Illustration of elements of $\mathrm{im}(\partial_2)$ and $\ker(\partial_1)$.

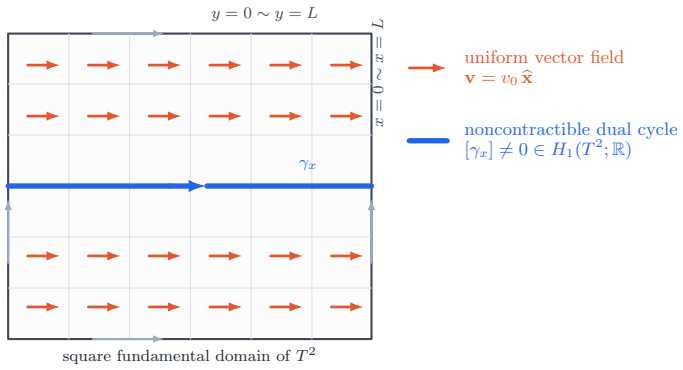

Figure 9.6: An illustration of a natural pairing between homology and cohomology on the torus. Note that this illustration is for real-valued vector fields on a continuum torus, vs. discrete $\mathbb{F}_2$ on the discretized torus.

the canonical statement is $\mathrm{H}^1 \cong \mathrm{H}_1^*$. Choosing dual bases of logical operators gives

$$G_X G_Z^{\mathsf{T}} = \mathbb{I}_{k \times k}, \tag{9.25}$$

which is the binary form of the canonical Pauli commutation relations. This is the topological interpretation of what we did in the proof of Proposition 8.10.

9.5 Topology and code distance

The distinction between closed and exact fields is topological. On a continuum torus, for example, a uniform horizontal real-valued vector field is curl-free but is not the gradient of a globally defined, single-valued function: the would-be coordinate x is not globally defined on the torus: see Figure 9.6. This is a real-valued analogy; the code itself uses coefficients in $\mathbb{F}_2$.

For the CSS code obtained from a cellulation, the cochain and chain complexes are

$$\mathbb{F}_2^{m_X} \xrightarrow{H_X^\mathsf{T}=\mathrm{d}_0} \mathbb{F}_2^n \xrightarrow{H_Z=\mathrm{d}_1} \mathbb{F}_2^{m_Z}, \tag{9.26a}$$

$$\mathbb{F}_2^{m_Z} \xrightarrow{H_Z^\mathsf{T}=\partial_2} \mathbb{F}_2^n \xrightarrow{H_X=\partial_1} \mathbb{F}_2^{m_X}. \tag{9.26b}$$

The logical spaces are therefore

$$\mathcal{L}_X = \mathrm{H}^1, \tag{9.27a}$$

$$\mathcal{L}_Z = \mathrm{H}_1. \tag{9.27b}$$

For the square torus, the shortest homologically nontrivial primal and dual cycles both have length L. Hence

$$d = d_X = d_Z = L = \sqrt{\frac{n}{2}}, \tag{9.28}$$

so the toric code has parameters $[[2L^2, 2, L]]$. For a general cellulation, d_Z is the length of the shortest homologically nontrivial cycle on the primal lattice, while d_X is the corresponding quantity on the dual lattice. They need not be equal when the cellulation is not self-dual.

9.6 Codes on general surfaces

The same construction works for any cellulation of a closed two-dimensional surface: put qubits on edges, X-checks on vertices, and Z-checks on faces. The associated chain complex automatically guarantees $H_Z H_X^\mathsf{T} = 0$. The resulting logical operators are determined by the homology and cohomology of the surface, which is why these are called *topological codes*.

Theorem 9.6 (Homology of a genus-g surface)**.** Let Σ_g be a closed, connected, orientable surface with g handles. Then

$$\mathrm{H}_1(\Sigma_g; \mathbb{F}_2) \cong \mathbb{F}_2^{2g}, \tag{9.29a}$$

$$\mathrm{H}^1(\Sigma_g; \mathbb{F}_2) \cong \mathbb{F}_2^{2g}. \tag{9.29b}$$

Consequently, the corresponding surface code encodes

$$k = 2g \tag{9.30}$$

logical qubits.

Each handle contributes two independent nontrivial cycles, conventionally called an a-cycle and a b-cycle. In experiment it is difficult to realize literal handles or periodic identifications in a planar architecture. We will soon see how suitable boundaries allow one to store logical states in a planar surface code.

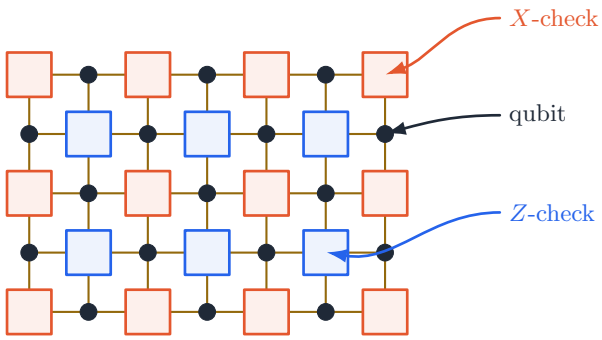

Figure 9.7: Tanner graph for a patch of the toric code.

9.7 Quantum LDPC codes and Tanner graphs

One more important property, especially for the later theory of fault tolerance, is that the toric code is a quantum low-density parity-check code.

Definition 9.7 (Quantum LDPC code)**.** A family of CSS codes is *quantum LDPC* if both H_X and H_Z are LDPC matrices. Equivalently, there are constants Δ_{C} and Δ_{B}, independent of n, such that every stabilizer check acts on at most Δ_{C} qubits and every qubit participates in at most Δ_{B} checks.

For the square toric-code family, $\Delta_{\mathrm{B}} = \Delta_{\mathrm{C}} = 4$ because every vertex or face check has weight four, while every edge belongs to two vertex checks and two face checks.

It is often useful to represent the code by a Tanner graph, as depicted in Figure 9.7. A combined CSS Tanner graph has one class of nodes for qubits, one class for X-checks, and one class for Z-checks; an edge records that a check acts on a qubit. The orthogonality constraint has the graph-theoretic form

$$(H_X H_Z^{\mathsf{T}})_{ab} = \bigl|N(a) \cap N(b)\bigr| \pmod 2 = 0, \tag{9.31}$$

where a is an X-check node, b is a Z-check node, and $N(\cdot)$ denotes the neighboring qubits. Thus every X-check/Z-check pair must share an even number of qubits.

Independently chosen sparse parity-check matrices generally fail this orthogonality condition. The difficult problem is to enforce it together with the LDPC condition and useful code parameters n, k, and d. The topological construction is important because the chain-complex identity $\partial_1 \partial_2 = 0$ enforces commutation automatically, while the topology controls the logical operators and distance.

10 Topologically ordered phases of matter

In Section 9 we introduced the toric code on an $L \times L$ square lattice with periodic boundary conditions. There is one qubit on each edge, an X-check on each vertex, and a Z-check on each face. Recall the notation in (9.1), and consider the Hamiltonian defined by the sum of all of the commuting stabilizers of the toric code:

$$H_{\mathrm{TC}} = -\sum_{v \in V} X^{\mathrm{d}v} - \sum_{f \in F} Z^{\partial f}. \tag{10.1}$$

Today we will explore this code from the condensed-matter perspective: the toric code realizes a topologically ordered phase of matter.

10.1 The toric code as a $\mathbb{Z}_2$ lattice gauge theory

The Hamiltonian in Eq. (10.1) can be interpreted as a $\mathbb{Z}_2$ lattice gauge theory. To motivate this interpretation, first recall the more familiar example of electromagnetism, which is a U(1) gauge theory.

10.1.1 Putting electromagnetism on a lattice

In the absence of matter, Maxwell's equations are

$$\nabla \cdot \mathbf{E} = 0, \tag{10.2a}$$

$$\nabla \cdot \mathbf{B} = 0, \tag{10.2b}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \tag{10.2c}$$

$$\nabla \times \mathbf{B} = \frac{1}{c^2}\frac{\partial \mathbf{E}}{\partial t}. \tag{10.2d}$$

The homogeneous equations $\nabla \cdot \mathbf{B} = 0$ and $\nabla \times \mathbf{E} = -\partial_t \mathbf{B}$ can be solved locally by introducing a scalar potential ϕ and a vector potential $\mathbf{A}$:

$$\mathbf{B} = \nabla \times \mathbf{A}, \tag{10.3a}$$

$$\mathbf{E} = -\nabla\phi - \frac{\partial \mathbf{A}}{\partial t}. \tag{10.3b}$$

Choose the gauge $\phi = 0$ and set $c = 1$. The electromagnetic Lagrangian then becomes

$$\mathcal{L} = \frac{1}{2}\mathbf{E}^2 - \frac{1}{2}\mathbf{B}^2 = \frac{1}{2}\left(\frac{\partial \mathbf{A}}{\partial t}\right)^2 - \frac{1}{2}(\nabla \times \mathbf{A})^2. \tag{10.4}$$

Thus $\mathbf{E}$ and $\mathbf{A}$ are canonically conjugate, with the sign convention

$$[E_i(\mathbf{x}), A_j(\mathbf{y})] = \mathrm{i}\,\delta_{ij}\delta(\mathbf{x} - \mathbf{y}), \tag{10.5}$$

and the Hamiltonian is

$$H_{\mathrm{EM}} = \int \mathrm{d}^3x \left[\frac{1}{2}\mathbf{E}^2 + \frac{1}{2}(\nabla \times \mathbf{A})^2\right]. \tag{10.6}$$

We have written the familiar $3+1$-dimensional vector notation in Eqs. (10.2)–(10.6). Only the canonical gauge-field structure will be used below. In $2+1$ dimensions, $\mathbf{B}$ is replaced by a pseudoscalar magnetic field and the spatial integral is $\int \mathrm{d}^2x$; the lattice construction proceeds in the same way.

To obtain a lattice gauge theory, place A_e and E_e on each oriented edge e. Using the discrete curl d_1 introduced in Section 9.2, the lattice Hamiltonian is

$$H_{\mathrm{lat}} = \frac{1}{2}\sum_{e\in E} E_e^2 + \frac{1}{2}\sum_{f\in F}\left(\sum_{e\in E}\langle f|\mathrm{d}_1|e\rangle A_e\right)^2. \tag{10.7}$$

The remaining Maxwell equation $\nabla\cdot\mathbf{E}=0$ becomes the Gauss-law constraint

$$\sum_{e\in E}\langle e|\mathrm{d}_0|v\rangle E_e = 0 \qquad \text{for every } v\in V. \tag{10.8}$$

One may enforce this constraint softly by adding a penalty:

$$H'_{\mathrm{lat}} = \frac{1}{2}\sum_{e\in E} E_e^2 + \frac{1}{2}\sum_{f\in F}\left(\sum_{e\in E}\langle f|\mathrm{d}_1|e\rangle A_e\right)^2 + g\sum_{v\in V}\left(\sum_{e\in E}\langle e|\mathrm{d}_0|v\rangle E_e\right)^2. \tag{10.9}$$

10.1.2 Making a discrete lattice gauge theory

The first modification is to make the gauge field compact:

$$A_e \sim A_e + 2\pi. \tag{10.10}$$

Just as angular momentum is quantized when its conjugate angle is periodic, Eq. (10.10) makes E_e integer-valued. It is then cleaner to replace the formal commutator $[E_e, A_e] = \mathrm{i}$ by the exponentiated Weyl relation

$$\mathrm{e}^{\mathrm{i}\phi E_e}\mathrm{e}^{\mathrm{i}\ell A_e} = \mathrm{e}^{\mathrm{i}\ell A_e}\mathrm{e}^{\mathrm{i}\phi E_e}\mathrm{e}^{-\mathrm{i}\ell\phi}, \qquad \ell\in\mathbb{Z}. \tag{10.11}$$

For a genuinely compact U(1) lattice gauge theory, the magnetic energy should also be periodic in the lattice flux, for example

$$-K\sum_{f\in F}\cos\left(\sum_{e\in E}\langle f|\mathrm{d}_1|e\rangle A_e\right). \tag{10.12}$$

The quadratic magnetic term in Eq. (10.7) is its small-flux approximation. This distinction will not affect the algebraic reduction to $\mathbb{Z}_2$.

The second modification is to restrict the edge Hilbert space to a binary $\mathbb{Z}_2$ degree of freedom. Heuristically, take

$$A_e \in \{0, \pi\}, \qquad E_e \in \{0, 1\}, \tag{10.13}$$

and define

$$Z_e := \mathrm{e}^{\mathrm{i}A_e}, \tag{10.14a}$$

$$X_e := \mathrm{e}^{\mathrm{i}\pi E_e}. \tag{10.14b}$$

Taking $\phi = \pi$ and $\ell = 1$ in Eq. (10.11) gives

$$X_e Z_e = Z_e X_e \mathrm{e}^{-\mathrm{i}\pi} = -Z_e X_e, \tag{10.15}$$

which is precisely the Pauli algebra.

The finite-dimensional theory should be understood directly through the exponentiated algebra in Eqs. (10.14) and (10.15). Two finite-dimensional Hermitian operators cannot obey an exact canonical commutator, so the notation in Eq. (10.13) is a useful mnemonic rather than a simultaneous diagonalization of A_e and E_e.

The resulting $\mathbb{Z}_2$ lattice gauge Hamiltonian takes the form

$$H_{\mathbb{Z}_2} = -h \sum_{e \in E} X_e - \sum_{f \in F} \prod_{e \in \partial f} Z_e - g \sum_{v \in V} \prod_{e \ni v} X_e. \tag{10.16}$$

The three terms are the $\mathbb{Z}_2$ analogues of the electric-field energy, the magnetic-field energy, and the Gauss-law penalty, respectively. At $h = 0$ and $g = 1$, Eq. (10.16) is the toric code Hamiltonian (10.1). For nonzero h, it is the toric code with an extra few-body perturbation.

10.1.3 Topological order and perturbative stability

The additional electric-field term in Eq. (10.16) does not immediately destroy the toric code. Instead, sufficiently weak few-body perturbations remain in the same phase of matter.

Theorem 10.1 (Stability of the toric-code phase)**.** Let

$$H_0' = H_{\mathrm{TC}} + hV, \tag{10.17}$$

where V is an extensive few-body, or q-local, perturbation, with q independent of L. For sufficiently small $|h|$, independent of L:

1. there exists a quasilocal unitary U that maps the ground-state space of H_{TC} to the four-dimensional low-energy space of H_0';

2. H_{TC} and H_0' are both gapped above their respective low-energy spaces;

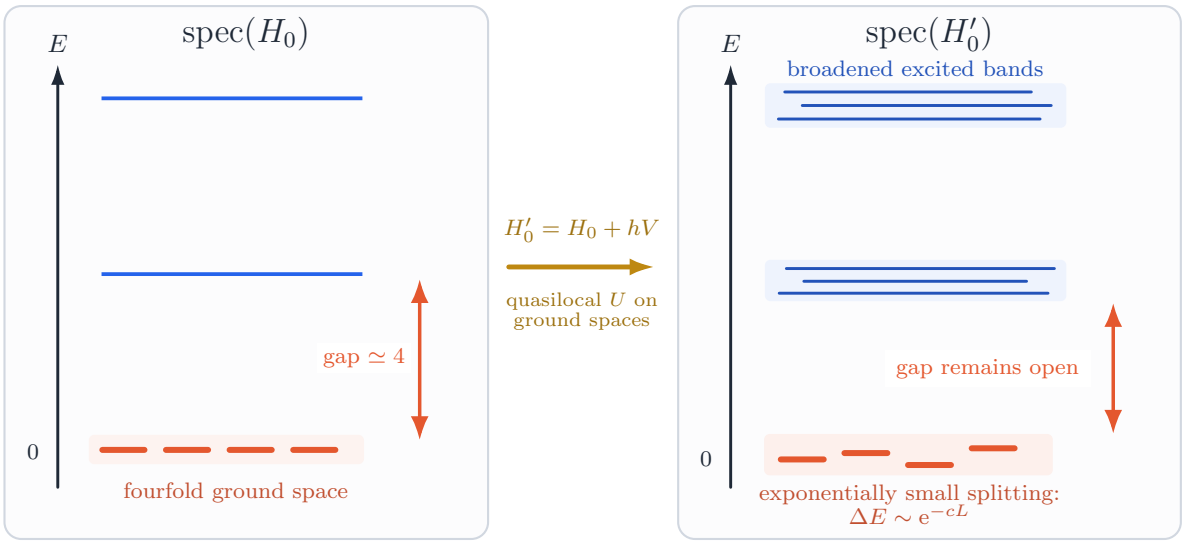

Figure 10.1: The toric code is a robust gapped phase of matter to any perturbation.

3. the splitting of the four low-energy levels obeys

$$\Delta E \lesssim \mathrm{e}^{-fL} \tag{10.18}$$

for some constant $f > 0$.

Here q-local means that V is a sum of terms acting on at most q qubits, with no requirement that those qubits be geometrically nearby. The smallness condition bounds the total interaction strength felt by each qubit; it does not bound the global operator norm $\|V\|$, which is extensive. The word "gap" means the gap above the entire four-dimensional low-energy band. At finite L, a generic perturbation may split the states inside that band by the exponentially small amount in Eq. (10.18). The consequences of Theorem 10.1 are illustrated in Figure 10.1.

The basic idea behind the stability theorem is a local form of the Knill–Laflamme conditions. Let $|\psi_\alpha\rangle$ denote the four toric-code ground states. Any operator O_R supported on a correctable region R obeys

$$\langle\psi_\alpha|O_R|\psi_\beta\rangle = c_{O_R}\delta_{\alpha\beta}. \tag{10.19}$$

In particular, an operator of weight less than the code distance cannot tell the ground states apart: recall Figure 9.3 and Figure 9.5 for an illustration of the operators O_R (correctable vs. uncorrectable errors) which can or cannot tell apart the codewords. Since the toric-code distance is $d = L$, perturbation theory must reach order proportional to L before it can generate a logical operator and lift the degeneracy. For a q-local perturbation, a product of r perturbing terms has weight at most qr. Thus the elementary distance-counting intuition is that one needs $r \geq d/q = \Theta(L)$ before a contribution can act logically. The full stability theorem controls the many perturbative processes

and the gap, using the textbook degenerate perturbation theory of quantum mechanics applied to operators (called Schrieffer-Wolff transformations).

The other required property, satisfied by the toric code, is check soundness.

Definition 10.2 (Check soundness)**.** A stabilizer Hamiltonian has (d_c, F)-check soundness if every stabilizer acting nontrivially on $M < d_c$ qubits can be written as a product of at most $F(M)$ Hamiltonian checks. For the toric code one may take

$$d_c = \Theta(L), \qquad F(M) \lesssim M^2. \tag{10.20}$$

Geometrically, a contractible loop of length M can be filled by a region containing at most order M^2 plaquettes or vertices (recall Figure 9.3 and Figure 9.5). Multiplying the checks in that filling produces the loop operator, which gives the scaling in Eq. (10.20).

For the reasons above, we say that the toric code is therefore a **topologically ordered phase of matter**. Two characteristic properties are:

1. the low-energy degeneracy depends on the topology of the spatial surface;

2. no local order parameter distinguishes the different ground states.

Indeed, products of local stabilizers produce contractible closed loops, whereas an operator that distinguishes or maps between ground states must wind nontrivially around the torus.

10.2 Comparison with a symmetry-breaking phase

It is useful to contrast the toric code with an ordinary phase of matter described by a classical repetition code. Consider

$$H_{\text{rep}} = -\sum_{i\sim j} Z_i Z_j. \tag{10.21}$$

Theorem 10.3 (Stability of the ferromagnetic phase)**.** Let

$$H = H_{\text{rep}} + hV. \tag{10.22}$$

For sufficiently small $|h|$, the system remains in a ferromagnetic phase if the perturbation is locally symmetric: writing $V = \sum_R V_R$, every local term obeys

$$[V_R, X_1 X_2 \cdots X_n] = 0. \tag{10.23}$$

A simple example is the one-dimensional quantum Ising model

$$H_{\text{Ising}} = -\sum_{i=1}^{n-1} Z_i Z_{i+1} - \sum_{i=1}^{n} (hX_i + \epsilon Z_i). \tag{10.24}$$

When $\epsilon = 0$ and h is small, the system is ferromagnetic. The $\epsilon = 0$ transverse-field Ising chain is exactly solvable. By contrast, when $\epsilon \neq 0$, the longitudinal field explicitly breaks the global $\mathbb{Z}_2$ symmetry and selects a unique ground state as $n \to \infty$. The reason is visible directly from the Knill–Laflamme condition:

$$\langle 0 \cdots 0 | \epsilon \sum_{i=1}^{n} Z_i | 0 \cdots 0 \rangle = \epsilon n, \tag{10.25a}$$

$$\langle 1 \cdots 1 | \epsilon \sum_{i=1}^{n} Z_i | 1 \cdots 1 \rangle = -\epsilon n. \tag{10.25b}$$

Thus an arbitrarily weak symmetry-breaking field distinguishes the two codewords at first order, with an extensive energy difference.

As usual for spontaneous symmetry breaking, the degenerate ferromagnetic ground states are most sharply defined in the thermodynamic limit. At finite n, a symmetry-preserving transverse field can produce an exponentially small tunneling splitting between the even and odd combinations of the two classical codewords. This is similar to the exponentially small splitting allowed by Theorem 10.1.

The contrast can be summarized by comparing the phase diagram of (10.24) to a generic perturbed toric code:

$$H'_{\mathrm{TC}} = H_{\mathrm{TC}} - h \sum_{e \in E} X_e - \epsilon \sum_{e \in E} Z_e. \tag{10.26}$$

As depicted in Figure 10.2, for the Ising model, the ferromagnetic phase is protected only along the symmetry-preserving line $\epsilon = 0$; a generic nonzero longitudinal field selects a trivial phase. For the toric code, an open two-dimensional region around $(h, \epsilon) = (0, 0)$ remains topologically ordered. At sufficiently large ϵ, the e particles condense, while at sufficiently large h, the m particles condense. The labels follow from the error strings discussed below: a Z_e term creates or moves e excitations, while an X_e term creates or moves m excitations. The key point is that the stability theorem permits finite generic perturbation strength without imposing a microscopic symmetry. This is the robustness of the $\mathbb{Z}_2$ gauge theory as a phase of matter.

10.3 Excitations in the toric code

Unfortunately, this zero-temperature robustness does not extend to finite temperature in two spatial dimensions.

10.3.1 e and m particles as Z and X syndromes

To understand why, reinterpret an error syndrome as a collection of excitations, as depicted in Figure 10.3. An open Z-string anticommutes with the two vertex checks at its endpoints and therefore creates two e excitations. An open dual-lattice X-string anticommutes with the two face checks at its dual endpoints and creates two m excitations. Thus e and m excitations are created

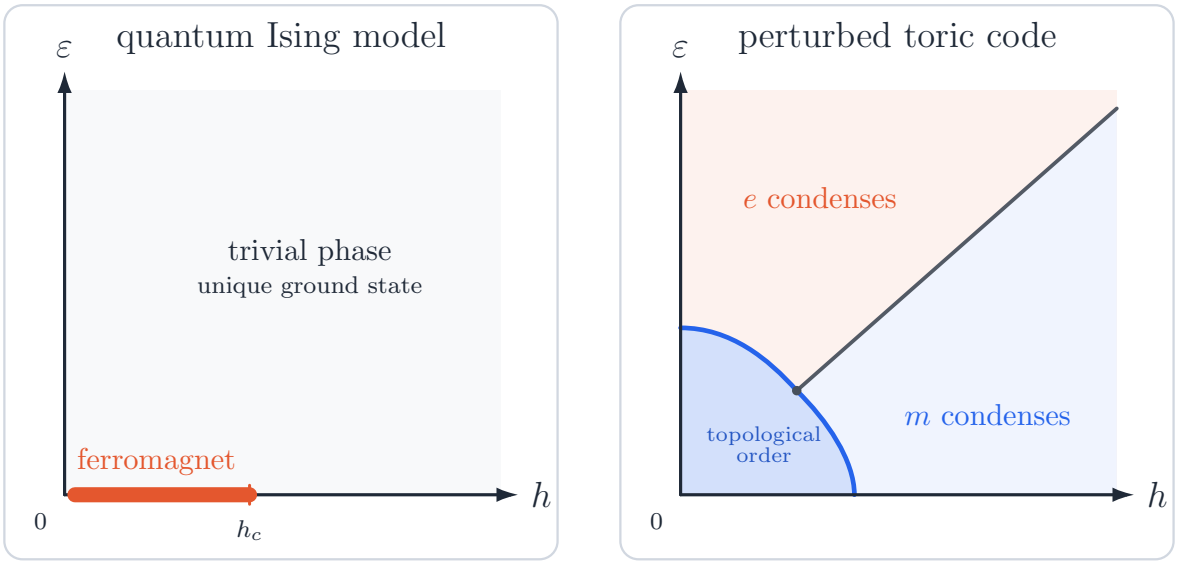

Figure 10.2: Comparing the phase diagrams of the quantum Ising model (10.24) vs. the perturbed toric code (10.26). Symmetry-breaking perturbations destroy ferromagnetism immediately, but no such analogue exists for the toric code.

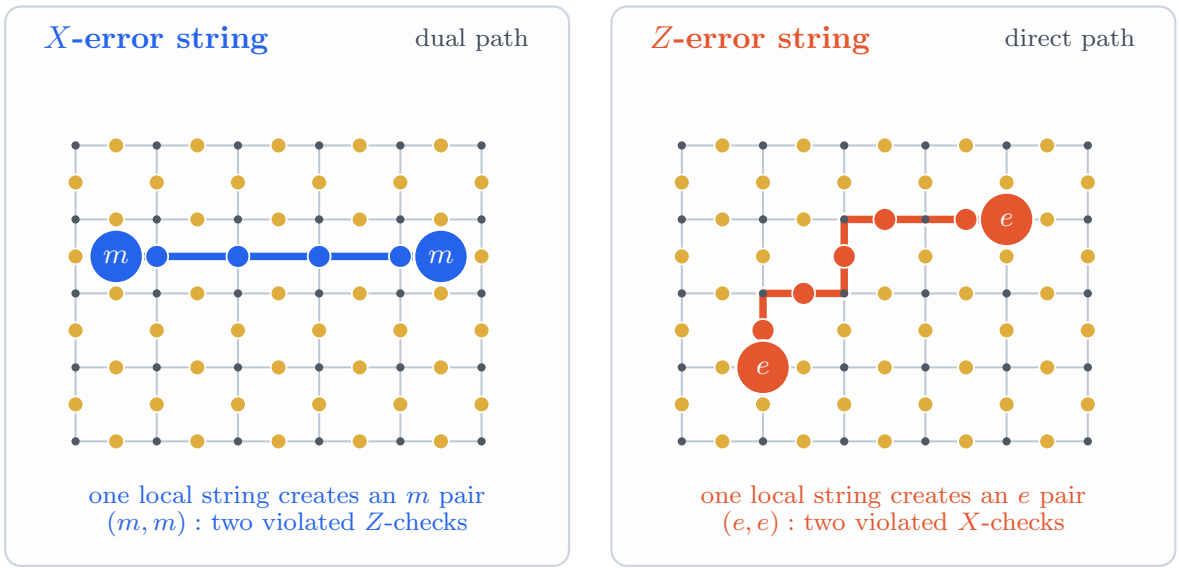

Figure 10.3: e vs. m excitations are created by Z vs. X type errors in the toric code.

in pairs on the torus, much as domain walls occur in pairs in one-dimensional Ising model (except at the boundary, with open boundary conditions).

Definition 10.4 (Fractionalization)**.** The e and m excitations are fractionalized: an isolated e or m cannot be created by a finite-support operator, but after a pair has been created, the two

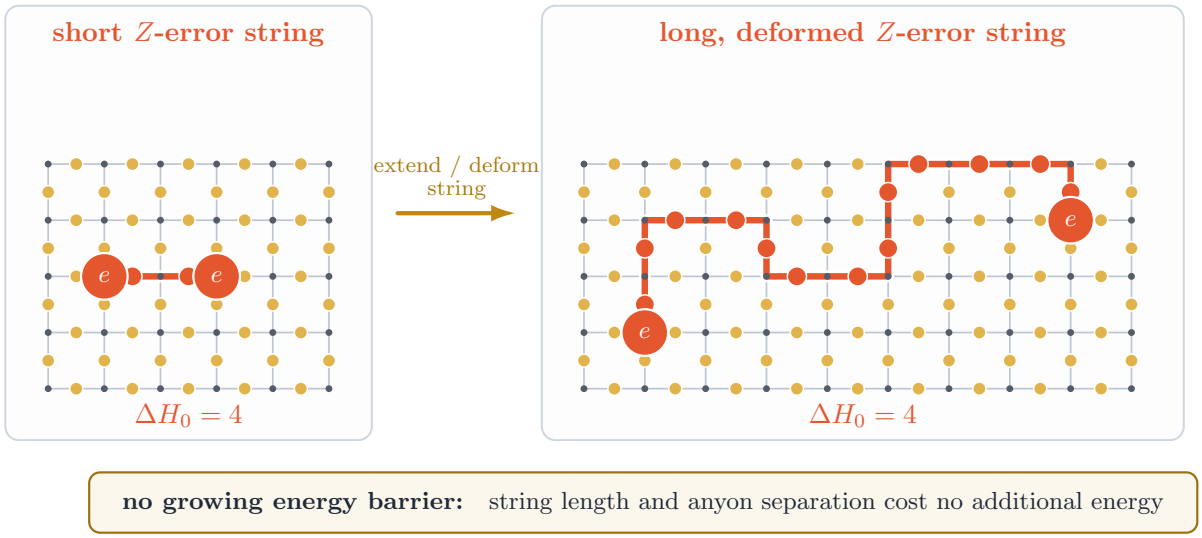

Figure 10.4: There is no energy barrier to causing e or m pairs to widely separate.

excitations can be moved independently.

10.3.2 No energy barrier in the toric code

Once a pair has been created, there is no additional energy penalty for moving its members farther apart: see Figure 10.4. Each violated toric-code check costs energy 2, so a pair has energy

$$\Delta H_0 = 4 \tag{10.27}$$

independently of its separation. The entropy associated with placing the two endpoints on an $L \times L$ torus scales as

$$\Delta S_{e\text{-pair}} \sim \log\left[(L^2)^2\right] = 4 \log L. \tag{10.28}$$

Setting Boltzmann's constant to one, the corresponding free-energy cost is

$$\Delta F_{e\text{-pair}} \sim \Delta H_0 - T\Delta S_{e\text{-pair}} = 4 - T \log\left[(L^2)^2\right] = 4(1 - T \log L). \tag{10.29}$$

For every fixed $T > 0$, this becomes negative at sufficiently large L. Consequently, both e and m excitations proliferate in the thermodynamic limit because their energy barriers are finite. Thus, the two-dimensional toric code has an $O(1)$ energy barrier: a noncontractible logical string can be grown while maintaining only its two endpoint excitations, at energy cost 4. We will see that it is not a self-correcting quantum memory at any fixed $T > 0$ in Section **??**; we require an active decoder to correct errors in the toric code.

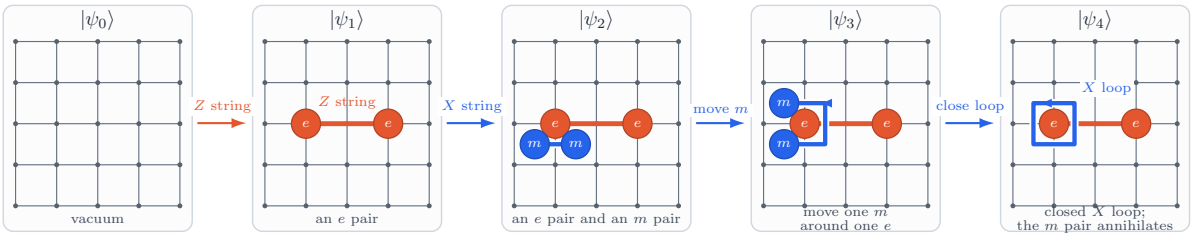

Figure 10.5: Braiding an m around an e leads to an overall phase in the wave function.

10.3.3 Braiding statistics and Abelian anyons

Before discussing the statistics of these excitations, recall that ordinary bosonic and fermionic wavefunctions obey

$$\psi_{\text{boson}}(x_1, x_2) = \psi_{\text{boson}}(x_2, x_1), \tag{10.30a}$$

$$\psi_{\text{fermion}}(x_1, x_2) = -\psi_{\text{fermion}}(x_2, x_1). \tag{10.30b}$$

The e and m excitations have nontrivial braiding statistics and are Abelian anyons. Consider the thought experiment of Figure 10.5. Begin in a toric-code ground state $|\psi_0\rangle$ and create a pair of e excitations with an open Z-string along γ:

$$|\psi_1\rangle = (Z\text{-string})|\psi_0\rangle. \tag{10.31}$$

Next create a pair of m excitations, move one of them around one endpoint of γ, and subsequently annihilate the m pair. The net operation is a contractible X-loop $W_X(\ell^*)$ on the dual lattice which crosses γ once. The final state is therefore

$$|\psi_4\rangle = (X\text{-loop})(Z\text{-string})|\psi_0\rangle = -(Z\text{-string})(X\text{-loop})|\psi_0\rangle = -|\psi_1\rangle \tag{10.32}$$

The minus sign in the second line arises because the two string operators cross once. Recall that the X-loop is a stabilizer of the toric-code ground state. We conclude that braiding an m excitation around an e excitation produces the phase -1.

By contrast, all Z Pauli operators commute with one another, as do all X Pauli operators. Hence braiding two excitations of the same species gives phase $+1$. For these reasons, one says that e and m each have bosonic self-statistics, but they are **mutual semions**: winding one species around the other produces the phase -1. Their bound state $\varepsilon = e \times m$ is a fermion.

11 Surface code

The logical operators of the toric code rely on nontrivial topology. Although it is possible to realize a torus in the laboratory with movable neutral atoms, or cleverly wired superconducting qubits, it is much more natural to ask whether the same basic construction can be implemented on a two-dimensional planar grid. The essential step is to choose suitable boundary conditions. The result is the **surface code**.

11.1 Planar code on an annulus

We first place the toric code checks on a cellulation of an annulus. There is a qubit on every edge. For every face f and vertex v, define

$$Z^{\partial f} := \prod_{e \in \partial f} Z_e, \tag{11.1a}$$

$$X^{\mathrm{d}v} := \prod_{e \ni v} X_e. \tag{11.1b}$$

There is no face, and hence no Z-check, filling the hole in the center of the annulus: see Figure 11.1. Exactly as for the toric code, a vertex and a face share either zero or two edges, and therefore

$$\left[X^{\mathrm{d}v}, Z^{\partial f}\right] = 0. \tag{11.2}$$

Proposition 11.1. The annular code encodes one logical qubit.

Proof. See Figure 11.2. The annulus has one nontrivial first cohomology class,

$$\mathrm{H}^1(\text{annulus}; \mathbb{F}_2) \cong \mathbb{F}_2. \tag{11.3}$$

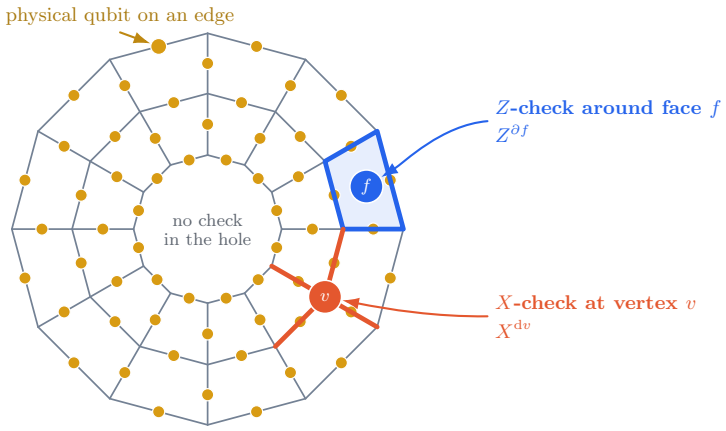

Figure 11.1: Putting toric code checks on the cellulation of an annulus.

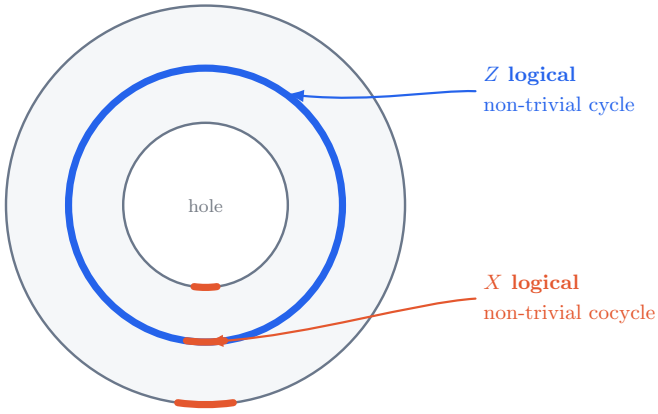

Figure 11.2: Sketch of logical operators (cycles and cocycles) on the annulus code.

A noncontractible primal cycle around the hole supports a Z-logical operator: it commutes with every X-check but cannot be shrunk to a product of face checks. A dual path from the inner boundary to the outer boundary supports the corresponding X-logical operator. The two representatives can be chosen to cross once and therefore anticommute. Hence the code carries one logical qubit. □

Equivalently, the Z-logical is the nonzero class in $\mathrm{H}_1(\text{annulus}; \mathbb{F}_2)$, while the X-logical is its Poincaré-dual cohomology class. This is the same cycle–cocycle pairing that appeared for the toric code.

The annulus is already a substantial improvement over a torus, since it can be embedded in the plane. A still cleaner and more compact construction replaces the hole by mixed boundary conditions.

11.2 A planar surface code from boundary conditions

Begin with a rectangular cellulation of a disk. It is useful to represent two opposite boundaries by two *ghost vertices*. We retain all face Z-checks adjacent to the ghost vertices, but remove the two corresponding vertex X-checks from the stabilizer group, as shown in Figure 11.3. Thus, if the augmented cellulation has V vertices, E edges, and F faces, then the physical qubits live on the E edges, while there are $V - 2$ vertex checks and F face checks.

The following elementary Euler-characteristic identity will be useful.

Lemma 11.2. For a cellulation of a disk,

$$V - E + F = 1, \tag{11.4}$$

where F counts the faces inside the disk and does not include the exterior.

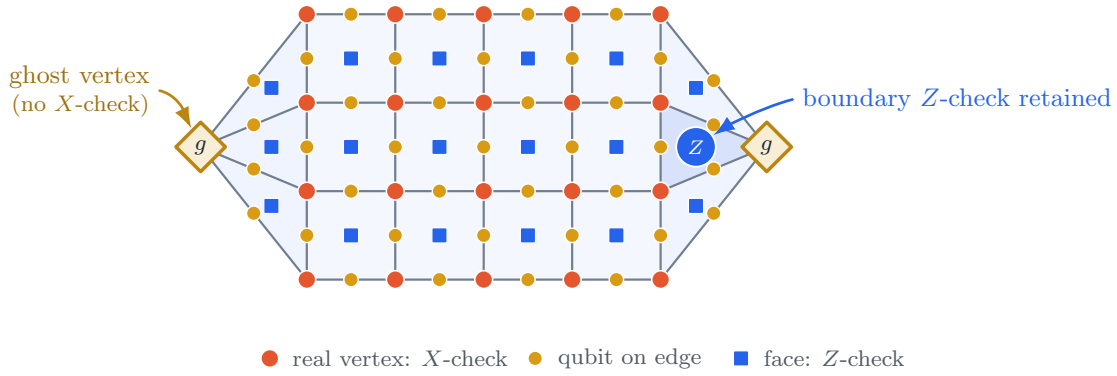

Figure 11.3: A special cellulation of a planar grid with two single ghost vertices at the left and right ends, which will lead us to our first surface code.

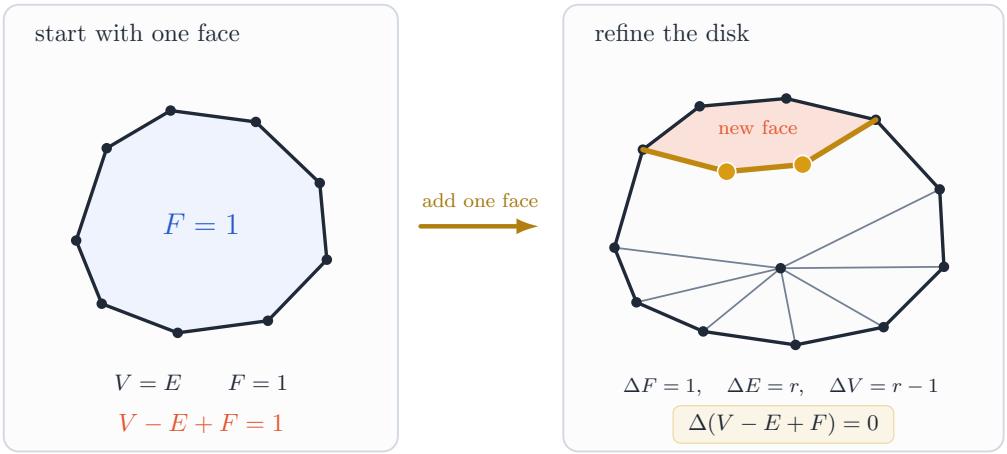

Figure 11.4: Illustrating the proof of Lemma 11.2.

Proof. For a polygon with no internal edges, $V = E$ and $F = 1$, so Eq. (11.4) holds. Add the internal faces one at a time. If a newly added face introduces ΔE edges, then it introduces one face and $\Delta E - 1$ new vertices. Therefore

$$F \longmapsto F + 1, \tag{11.5a}$$

$$E \longmapsto E + \Delta E, \tag{11.5b}$$

$$V \longmapsto V + (\Delta E - 1), \tag{11.5c}$$

which leaves $V - E + F$ unchanged: see Figure 11.4. □

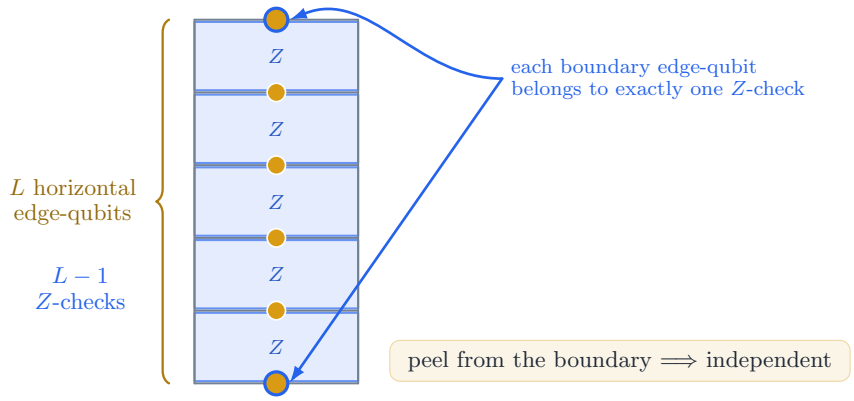

Figure 11.5: The Z-checks in each strip are all linearly independent of each other, and linearly independent from any Z-check in any other strip.

Lemma 11.3. All checks in this planar construction are independent. In particular,

$$\#(X\text{-checks}) = V - 2 = \operatorname{rank}(H_X), \tag{11.6a}$$

$$\#(Z\text{-checks}) = F = \operatorname{rank}(H_Z). \tag{11.6b}$$

Proof. The first equalities follow directly from the construction, so it remains to show independence. Consider the Z-checks in one boundary strip, illustrated in Figure 11.5. The L horizontal qubits in that strip are contained only in the $L-1$ adjacent Z-checks, and the corresponding rows of H_Z are full rank, since:

$$H_Z = \begin{pmatrix} 1 & 1 & 0 & \cdots & 0 & \cdots \\ 0 & 1 & 1 & \ddots & \vdots & \cdots \\ \vdots & \ddots & \ddots & \ddots & 0 & \cdots \\ 0 & \cdots & 0 & 1 & 1 & \cdots \end{pmatrix}, \tag{11.7}$$

The same argument is then applied to the next strip. Iterating proves that all Z-checks are independent. The same peeling argument, applied in the dual direction, proves independence of the X-checks. □

Proposition 11.4. The planar surface code encodes one logical qubit.

Proof. Using Lemmas 11.2 and 11.3,

$$k = n - \operatorname{rank}(H_X) - \operatorname{rank}(H_Z) = E - (V-2) - F = 2 - (V - E + F) = 1, \tag{11.8}$$

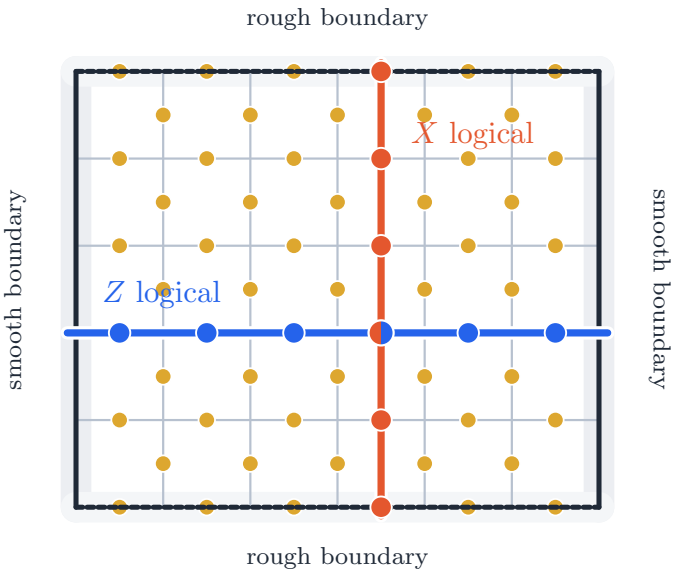

Figure 11.6: Logical operators in the surface code.

which completes the proof. □

A primal Z-string may terminate on the two boundaries where the relevant X-checks have been removed; these are the *rough* boundaries. A dual X-string may terminate on the complementary pair of *smooth* boundaries. Thus a representative Z-logical joins the rough boundaries, while a representative X-logical joins the smooth boundaries: see Figure 11.6.

For the rectangular family drawn in the lecture script, the numbers of qubits, ordinary vertices, and faces are

$$n = L(L-1) + (L-1)(L-2) = 2(L-1)^2, \tag{11.9a}$$

$$\#(X\text{-checks}) = L(L-2), \tag{11.9b}$$

$$\#(Z\text{-checks}) = (L-1)^2. \tag{11.9c}$$

The shortest Z-logical has weight $L-1$, whereas the shortest X-logical has weight L. Hence the parameters are $[\![2(L-1)^2, 1, L-1]\!]$; more precisely,

$$d_Z = L-1, \qquad d_X = L. \tag{11.10}$$

11.3 The rotated surface code

The preceding construction can be made more efficient. Put qubits on the vertices of an $L \times L$ square lattice. As shown in Figure 11.7, color the plaquettes in a checkerboard pattern and place X-checks on one color and Z-checks on the other. Along the boundary, add alternating weight-2

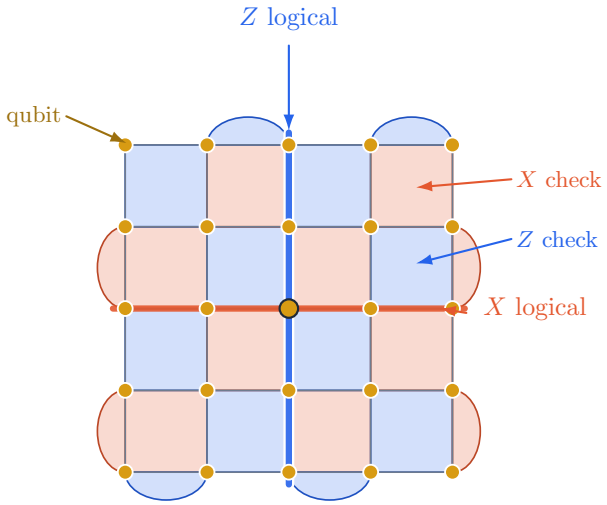

Figure 11.7: The rotated surface code.

checks, so that the boundary "bubbles" skip every other plaquette. The weight-2 Z-checks lie on the rough boundaries, while the weight-2 X-checks lie on the smooth boundaries.

Proposition 11.5. The rotated checks obey

$$H_X H_Z^{\mathsf{T}} = 0. \tag{11.11}$$

Proof. If two plaquettes share an edge, they overlap on two qubits, so an X-check and a Z-check commute. If two plaquettes share only one vertex, then the bipartite coloring of the square lattice assigns them the same check type. The alternating boundary checks are arranged by the same rule. Thus every X-check and Z-check overlap on either zero or two qubits, which proves Eq. (11.11). □

Proposition 11.6. The rotated surface code has parameters $[\![L^2, 1, L]\!]$.

Proof. There are L^2 qubits. The number of bulk plaquette checks plus boundary weight-2 checks is

$$(L-1)^2 + 2(L-1) = L^2 - 1. \tag{11.12}$$

A boundary-peeling argument, analogous to the proof of Lemma 11.3, shows that these checks are independent. Therefore $k = L^2 - (L^2 - 1) = 1$. A Z-string joining the rough boundaries and an X-string joining the smooth boundaries each have weight L: see Figure 11.7. Conversely, a nontrivial logical string must span the patch between the corresponding pair of boundaries, and hence cannot have smaller weight. □

The rotated layout uses roughly half as many qubits as the unrotated layout at fixed distance. It is also the layout most commonly used in experimental work. Henceforth, when we say "surface

code," we will generally mean this rotated version.

11.4 The Bravyi–Poulin–Terhal bound

The surface code constructed above has only $k = 1$ logical qubit. One may ask whether a geometrically local code can have $k \gg 1$ at large n without sacrificing the scaling $d \sim L$.

Theorem 11.7 (Bravyi–Poulin–Terhal bound)**.** For a family of codes in $D \geq 2$ spatial dimensions with geometrically local checks,

$$k\, d^{\frac{2}{D-1}} = \mathrm{O}(n). \tag{11.13}$$

Here geometrically local means that the check diameter and the density of qubits per unit volume are bounded independently of n. The implicit constant in Eq. (11.13) may depend on these microscopic bounds.

We prove the result in two spatial dimensions. The key algebraic input is the cleaning lemma.

Lemma 11.8 (Cleaning lemma)**.** Let R be a set of qubits such that every Pauli error supported in R is correctable. For every logical Pauli class $[L] \in N(\mathcal{S})/\mathcal{S}$, there exists $S \in \mathcal{S}$ such that

$$\mathrm{supp}(LS) \cap R = \varnothing. \tag{11.14}$$

In other words, every logical operator has an equivalent representative with no support in R.

Proof. Work with Pauli operators modulo phases, represented as vectors in $\mathbb{F}_2^{2n}$ with the symplectic inner product. Let H be the Pauli parity-check matrix, so that the stabilizer subspace is

$$\mathcal{S} = \mathrm{im}(H^{\mathsf{T}}). \tag{11.15}$$

Let π_R restrict a Pauli vector to R, and define

$$W := \pi_R(\mathcal{S}). \tag{11.16}$$

For $q \in W^{\perp}$, extend q by zero outside R. The resulting Pauli commutes with every stabilizer. Since it is supported in the correctable region R, it cannot represent a nontrivial logical operator and must itself lie in $\mathcal{S}$.

Now let ℓ represent the logical operator L. For every $q \in W^{\perp}$, the preceding paragraph implies

$$\langle q, \pi_R(\ell) \rangle_{\mathrm{s}} = 0. \tag{11.17}$$

Consequently,

$$\pi_R(\ell) \in (W^{\perp})^{\perp} = W. \tag{11.18}$$

There is therefore a stabilizer vector $s \in \mathcal{S}$ satisfying $\pi_R(s) = \pi_R(\ell)$. In additive Pauli notation,

$$\pi_R(\ell + s) = 0, \tag{11.19}$$

which is precisely Eq. (11.14) in multiplicative operator notation. □

The first geometric consequence is that well-separated correctable regions may be cleaned simultaneously.

Lemma 11.9 (Union lemma)**.** Let $R_1, \ldots, R_m$ be correctable regions. Suppose that no stabilizer generator intersects two distinct regions R_i and R_j. Then the union

$$R := \bigcup_{i=1}^{m} R_i \tag{11.20}$$

is correctable, and every logical operator can be cleaned out of all of R.

Proof. First consider a Pauli Q supported on R that commutes with every stabilizer. Decompose it as $Q = Q_1 \cdots Q_m$, with Q_i supported on R_i. Because no stabilizer generator meets two distinct regions, the commutation of Q with every generator implies that each Q_i separately commutes with every generator. Correctability of R_i then implies $Q_i \in \mathcal{S}$, and hence $Q \in \mathcal{S}$. Thus R supports no nontrivial logical operator and is correctable.

To clean a logical operator from the full union, apply the cleaning lemma successively to the regions R_i. When cleaning R_i, write the required stabilizer as a product of local generators and discard every generator disjoint from R_i; this leaves its restriction to R_i unchanged. Every remaining generator intersects R_i and, by the separation hypothesis, intersects no other R_j. Thus cleaning R_i cannot restore support on a region that has already been cleaned. After all regions have been treated, the logical representative has no support on their union. □

The second consequence is sometimes called a holographic principle for local codes.

Lemma 11.10. For a two-dimensional local stabilizer code of distance d, there is a constant $c > 0$ such that every square of linear size

$$\ell_0 = cd \tag{11.21}$$

is correctable.

Proof. Let $w = \mathrm{O}(1)$ be larger than the diameter of every stabilizer generator. A region containing fewer than d qubits is correctable by the definition of code distance. Begin with a square core of linear size $\mathrm{O}(\sqrt{d})$, chosen so that it contains fewer than d qubits. Surround it by concentric shells of width w, as shown in Figure 11.8. A shell at linear scale ℓ contains $\mathrm{O}(w\ell)$ qubits, and is therefore correctable as long as ℓ is at most a sufficiently small constant times d/w.

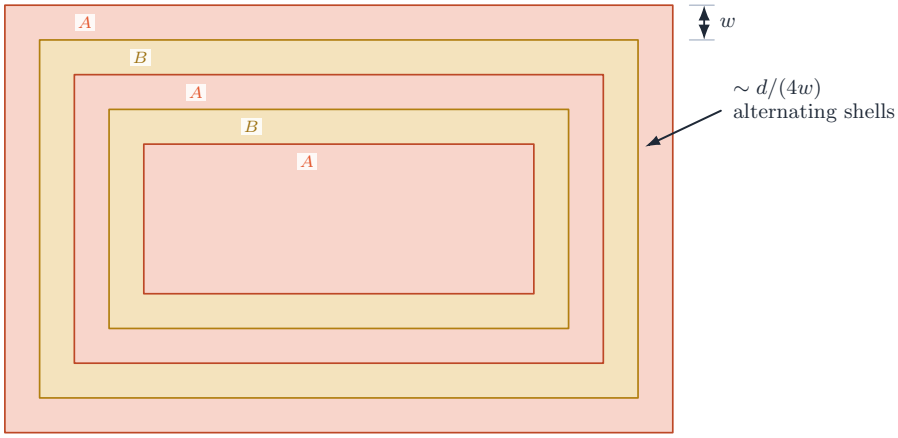

Figure 11.8: Applying the cleaning lemma to a series of concentric shells proves Lemma 11.10.

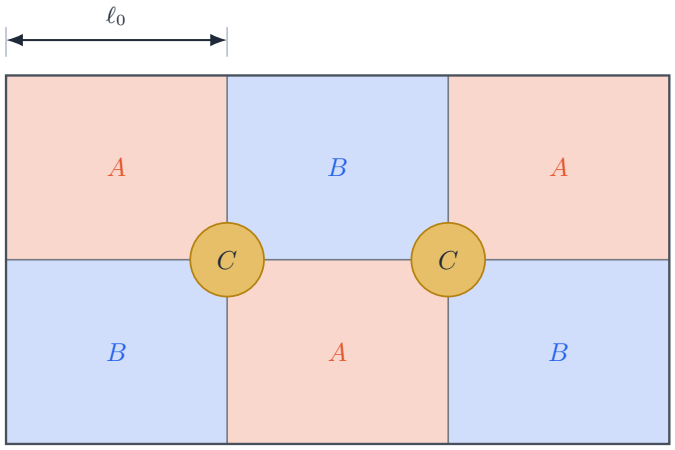

Figure 11.9: The regions A, B and C used for cleaning logicals.

Apply the union lemma separately to all red shells and to all yellow shells. Shells of the same color are separated by more than the check diameter, so each color class is correctable. Choose the outermost shell to be yellow. If a nontrivial logical operator were supported inside the entire square, clean it out of the red shells using only generators that meet those shells. The outermost yellow shell is a buffer of width at least w, so these generators remain inside the square. The resulting logical representative is supported entirely on the yellow region, contradicting its correctability. Thus the full square is correctable up to linear size $\ell_0 \sim d/w \sim d$. □

As shown in Figure 11.9, we now partition the two-dimensional system into square patches of linear size ℓ_0, colored alternately A and B. Around every point where patches of the same color would otherwise meet, remove an $\mathrm{O}(w)$-sized buffer region and place it in a third set C. The buffers are large enough that no stabilizer generator connects two distinct A-patches or two distinct

B-patches.

Each individual patch is correctable by Lemma 11.10, and the union lemma implies that the full regions A and B are correctable. Let $\{\overline{X}_a, \overline{Z}_a\}_{a=1}^k$ be canonical logical Pauli generators. Clean every $\overline{X}_a$ out of A and every $\overline{Z}_a$ out of B. The resulting representatives obey

$$\mathrm{supp}(\overline{X}_a) \subseteq B \cup C, \tag{11.22a}$$

$$\mathrm{supp}(\overline{Z}_a) \subseteq A \cup C. \tag{11.22b}$$

Since A and B are disjoint, the canonical anticommutation pairing is realized entirely inside C:

$$\pi_C(\overline{X}_a) \cdot \pi_C(\overline{Z}_b) = \delta_{ab}. \tag{11.23}$$

The $2k$ restricted logical vectors are therefore linearly independent in the $2|C|$-dimensional Pauli space on C, and hence

$$k \leq |C|. \tag{11.24}$$

There is one $\mathrm{O}(w)$-sized C-region per area ℓ_0^2, so

$$|C| = O\left(\frac{n}{\ell_0^2}\right) = O\left(\frac{n}{d^2}\right). \tag{11.25}$$

Combining Eqs. (11.24) and (11.25) proves (11.13) in two dimensions.

In D dimensions, a constant-width shell around a cube of side ℓ contains $\mathrm{O}(w\ell^{D-1})$ qubits. The same shell argument therefore gives $\ell_0 = \Theta(d^{1/(D-1)})$. The set C is an $\mathrm{O}(w)$-thickened codimension-two skeleton, so it occupies an $\mathrm{O}(\ell_0^{-2})$ fraction of the system. This produces the exponent in Eq. (11.13).

For the rotated surface code, $n = L^2$, $k = 1$, and $d = L$, so

$$kd^2 = n. \tag{11.26}$$

Thus the surface code is "best possible" if one demands both geometric locality and the largest possible distance at fixed n.

11.5 Thermodynamics of the surface code

We have constructed a good two-dimensional code, but this does not yet explain how quantum information should be protected in it. The surface code is not a passive quantum memory at finite temperature: it requires an active decoder, which will be the subject of the next lecture.

A closely related, although not equivalent, observation comes from equilibrium thermodynamics.

Definition 11.11 (Gibbs state)**.** Define the Gibbs state

$$\rho_\beta := \frac{\mathrm{e}^{-\beta H_0}}{\mathrm{tr}(\mathrm{e}^{-\beta H_0})}, \tag{11.27}$$

where the (here, surface code) Hamiltonian is

$$H_0 := -\sum_{a \in \text{checks}} S_a. \tag{11.28}$$

Proposition 11.12. For a surface code on n qubits with one logical qubit and independent checks, the partition function is

$$Z(\beta) := \mathrm{tr}(\mathrm{e}^{-\beta H_0}) = 2(2\cosh\beta)^{n-1}. \tag{11.29}$$

Proof. There are $n-1$ independent stabilizer checks. Consequently every assignment of check eigenvalues $s_a \in \{\pm 1\}$ occurs, and the simultaneous eigenspace for each syndrome has dimension two, corresponding to the one logical qubit. Choosing simultaneous eigenstates of all checks,

$$Z(\beta) = 2 \sum_{s_1,\ldots,s_{n-1}=\pm 1} \exp\left(\beta \sum_{a=1}^{n-1} s_a\right) = 2 \prod_{a=1}^{n-1} \left(\mathrm{e}^{\beta} + \mathrm{e}^{-\beta}\right) = 2(2\cosh\beta)^{n-1}, \tag{11.30}$$

which is (11.29). □

The free-energy density is therefore

$$\begin{aligned} f_n(\beta) &:= -\frac{1}{\beta n} \log Z(\beta) \\ &= -\frac{\log 2}{\beta n} - \frac{n-1}{\beta n} \log(2\cosh\beta) \xrightarrow[n\to\infty]{} -\frac{1}{\beta} \log(2\cosh\beta). \end{aligned} \tag{11.31}$$

This limiting function is analytic for every finite β, so there is no finite-temperature thermodynamic phase transition detectable in the free energy. This static statement is suggestive of, but is not logically equivalent to, a statement about memory lifetime.

The result is consistent with the anyon picture from the toric code. There is only an O(1) energy barrier to nucleate an e- or m-anyon pair, and the two anyons can then be pulled apart without any additional energy cost. At any finite temperature, thermally activated anyons can therefore diffuse over long distances under local dynamics, producing large error strings. An active decoding procedure is needed to infer and remove these strings before they implement a logical operator.