

# AN INTRODUCTION TO QUANTUM ERROR CORRECTION FOR MATHEMATICIANS

EDUARDO DUEÑEZ

**ABSTRACT.** These notes accompany a series of talks in the Quantum Error Correction Seminar at UTSA (Fall 2026). Pure states are points of complex projective spaces; we develop quantum error correction using linear lifts over  $\mathbb{C}$  and binary linear algebra over  $\mathbb{F}_2$ , followed by a problem of classical statistical inference. Surface codes are treated in some detail: we prove that the rotated patch of odd distance  $d$  is a  $[[d^2, 1, d]]$  code and reduce its decoding, under independent noise, to minimum-weight perfect matching. Later sections discuss belief propagation and its dynamics, quantum low-density parity-check codes (hypergraph products and bivariate bicycle codes), algebraic post-processing, repeated measurements, and the statistical comparison of decoders, with notes on open questions. Undergraduate linear algebra and elementary probability are the only prerequisites; the quantum formalism is defined as needed.

## CONTENTS

|                                                                |    |
|----------------------------------------------------------------|----|
| Introduction                                                   | 2  |
| 1. Qubits, measurements, and codes                             | 3  |
| 2. Stabilizer codes                                            | 7  |
| 3. Surface codes                                               | 11 |
| 4. Decoding surface codes                                      | 16 |
| 5. Belief propagation                                          | 21 |
| 6. Quantum low-density parity-check codes                      | 25 |
| 7. Post-processing and other refinements of belief propagation | 28 |
| 8. Faulty measurements and the evaluation of decoders          | 29 |
| Appendix A. Qubits, projective geometry, and the Bloch sphere  | 33 |
| Appendix B. Density operators, channels, and exact recovery    | 36 |
| Appendix C. Solutions to the problems                          | 38 |
| Appendix D. Notation and further reading                       | 42 |
| References                                                     | 43 |

---

*Date:* Version of October 22, 2026 (third revision).

*2020 Mathematics Subject Classification.* Primary 81P73; Secondary 94B05, 94B35, 05C70.

*Key words and phrases.* quantum error correction, stabilizer codes, surface codes, minimum-weight matching, belief propagation.

## INTRODUCTION

A classical error-correcting code protects a message by embedding it in a larger space and recording parities that an error is likely to disturb. A quantum code must do the same for a point of a complex projective space, and two obstacles seem to stand in the way: measuring the encoded state can change it, and the set of possible errors is a continuum rather than a finite set of bit flips. The theory of stabilizer codes resolves both obstacles with linear algebra. Commuting measurements reveal a binary *syndrome* without revealing the encoded state, and correcting a finite set of Pauli errors suffices to correct every error in the linear span of their lifts. What remains is a classical problem: given the syndrome, choose a correction, knowing that many different errors produce the same syndrome and that some of these differ by an undetectable but harmful operator. That choice, called *decoding*, is a problem of statistical inference, and it is where most of the mathematical interest of these notes lies.

We treat one family of codes in detail, the surface codes, because their decoding problem is a graph problem that can be drawn, solved exactly on small instances, and understood geometrically. Most of the general structure of quantum error correction —stabilizers, logical operators, degeneracy, and the gap between consistency with the syndrome and successful recovery— is already visible on a surface patch of nine qubits. The later sections replace the graph by a sparse bipartite graph and the exact optimization by belief propagation, which is the setting of much current work on quantum low-density parity-check codes.

Our standing conventions are the following. Pure states are complex lines in finite-dimensional Hilbert spaces; vectors are representatives, and operators on the Hilbert spaces give linear lifts of transformations of states. The inner products determine orthogonality and transition probabilities on the projective spaces. We work with these lifts when proving statements or making calculations; no step in the main text depends on a physical interpretation, although we occasionally mention one in a remark. Noise is always modeled by an explicitly declared probability distribution on Pauli errors. The stochastic Pauli model is an assumption; the theorem proved in [Section B.3](#) says that a recovery correcting a family of Pauli errors also corrects channels whose Kraus operators lie in their linear span. We distinguish carefully between three properties of a proposed correction (agreement with the actual error, consistency with the observed syndrome, and preservation of the encoded information), since only the last one matters and only the second one can be checked.

**Prerequisites.** We assume undergraduate linear algebra (over  $\mathbb{C}$  and over the two-element field  $\mathbb{F}_2$ ), the elementary theory of tensor products of finite-dimensional spaces, and elementary discrete probability, including conditional probability and Bayes' rule. These facts are used without proof. No previous acquaintance with quantum mechanics or quantum information is needed: [Section 1](#) defines the few notions that we use. [Section A](#) develops the projective geometry further through the Bloch sphere, entanglement, and Clifford transformations, while [Section B](#) supplies density operators and channels. Readers familiar with classical linear codes will recognize the parity-check matrices of [Section 2](#) and the message passing of [Section 5](#); nothing from coding theory is assumed.

**Organization.** The notes are organized as follows. [Section 1](#) defines qubits, measurements, encodings, and Pauli operators, and shows by the example of the repetition code both how a parity measurement can preserve an unknown superposition and why classical repetition alone fails. [Section 2](#) develops stabilizer codes: their dimension, syndromes, logical operators, distance, the binary symplectic description, and the Calderbank–Shor–Steane construction, with Shor's nine-qubit code as the first complete example. [Section 3](#) proves that the rotated surface patch of odd distance  $d$  is a  $[[d^2, 1, d]]$  code and explains the homological meaning of its logical operators. [Section 4](#) formulates decoding as inference, reduces the most-likely-error problem for the surface code to minimum-weight perfect matching, and discusses the threshold phenomenon with a small simulation. [Section 5](#) derives belief propagation on Tanner graphs, discusses the choices (schedule, damping, stopping rule) that determine its behaviour on graphs with cycles, and exhibits its failure modes, including the one caused by degeneracy, on exactly solvable

examples. [Section 6](#) introduces quantum low-density parity-check codes, with the hypergraph product (which contains the surface code) and the bivariate bicycle codes as examples. [Section 7](#) describes ordered-statistics post-processing in detail and surveys the other refinements of belief propagation used for these codes. [Section 8](#) admits faulty measurements, introduces detectors and detector error models, and explains how two decoders should be compared on simulated data. From [Section 3](#) on, each section closes with brief notes on the literature and on open questions, which are meant as an invitation to further reading and research. Each section ends with problems; solutions to all of them are collected in [Section C](#), and [Section D](#) summarizes notation and further reading.

[Sections 1](#) to [4](#) form the core of the seminar and should be read in order. [Sections 5](#) to [8](#) depend on them; [Section 6](#) may be read before [Section 5](#), and [Section 8](#) directly after [Section 4](#), except for its comparison of decoders ([Section 8.4](#)), which uses the post-processing of [Section 7](#).

**Sources and companion material.** For the general stabilizer framework we follow Gottesman [[Got97](#), [Got09](#)]; Roffe’s introductory guide [[Rof19](#)] is a useful companion with many worked examples, and Nielsen and Chuang [[NC10](#)] remain the standard reference for the quantum formalism. Specific sources for later topics are cited where they are used. Every numerical claim about a small example in these notes —ranks, distances, syndromes, posterior probabilities, and belief-propagation messages— has been checked by exhaustive enumeration with a short program that uses only the Python standard library; the simulations of [Sections 4](#) and [8](#) use the open-source packages Stim [[Gid21](#)], PyMatching [[Hig22](#), [HG25](#)], and 1dpc [[RWBC20](#)].<sup>1</sup>

## 1. QUBITS, MEASUREMENTS, AND CODES

This section fixes the minimal quantum formalism used throughout: state spaces, projective measurements, encodings, and Pauli operators. Its single example, the three-qubit repetition code, already shows the mechanism behind every code in these notes (a measurement that reveals an error without revealing the encoded state) together with the reason why classical repetition by itself is insufficient.

**1.1. State spaces.** For a complex Hilbert space  $\mathcal{H}$ , its *projective space* is

$$\mathbb{P}(\mathcal{H}) := (\mathcal{H} \setminus \{0\})/\mathbb{C}^\times.$$

A point  $[u] \in \mathbb{P}(\mathcal{H})$  is the complex line spanned by a nonzero vector  $u$ ; thus  $[u] = [v]$  precisely when  $v = \lambda u$  for some  $\lambda \in \mathbb{C}^\times$ . The inner product supplies additional structure: two points are orthogonal when their representative lines are orthogonal, and their *transition probability* is

$$(1.1) \quad \frac{|\langle u, v \rangle|^2}{\|u\|^2 \|v\|^2}.$$

Both notions are independent of the representatives.

**1.1. Definition.** A *qubit* is a system whose underlying Hilbert space is  $\mathcal{H}_1 := \mathbb{C}^2$  with its standard inner product and a distinguished orthonormal basis, written  $|0\rangle, |1\rangle$  and called the *computational basis*. The underlying Hilbert space of  $n$  qubits is

$$\mathcal{H}_n := (\mathbb{C}^2)^{\otimes n},$$

with orthonormal computational basis  $|b\rangle = |b_1\rangle \otimes \cdots \otimes |b_n\rangle$  indexed by  $b \in \{0, 1\}^n$ ; we abbreviate  $|b\rangle$  as  $|b_1 \cdots b_n\rangle$ . A *pure state* of  $n$  qubits is a point of  $\mathbb{P}(\mathcal{H}_n) \cong \mathbb{CP}^{2^n-1}$ , their *pure-state space*. In particular, the pure-state space of one qubit is the complex projective line  $\mathbb{CP}^1$ .

---

<sup>1</sup>The programs, their outputs, and the sources of these notes will be made available with the notes on the seminar website. Readers are encouraged to modify the examples; in our experience, little builds intuition for decoding as quickly as watching a decoder fail on an instance one has constructed by hand.

We call a nonzero vector  $u$  a *representative* of the state  $[u]$ ; a unit representative is *normalized*. Any two normalized representatives of the same state differ by multiplication by a scalar of modulus one, called a *global phase*. Normalization chooses a length, but does not select a preferred representative. The notation  $|u\rangle$  (“ket  $u$ ”) denotes nothing more than a vector  $u$ , and  $\langle u|$  denotes the linear functional  $v \mapsto \langle u, v \rangle$ , with the inner product linear in its second argument. Thus  $|u\rangle\langle v|$  is the rank-one operator  $w \mapsto \langle v, w \rangle u$ . We use the ket notation only when it makes the tensor-product structure easier to read. Thus  $|b\rangle$  denotes a basis vector and  $[|b\rangle]$  its corresponding computational state. We use representative vectors in calculations and proofs; the pure states themselves are always projective points. [Section A](#) identifies the one-qubit pure-state space with the Bloch sphere and develops its projective geometry further.

Note that a unit vector  $u = \sum_b u_b |b\rangle$  is *not* a probability distribution on  $\{0, 1\}^n$ , although  $(|u_b|^2)_b$  is one. The relative phases of the coefficients  $u_b$  are invisible to a measurement in the computational basis but not to measurements in other bases; for example,  $[|0\rangle + |1\rangle]$  and  $[|0\rangle - |1\rangle]$  are orthogonal states, hence perfectly distinguishable, although normalized representatives have coefficients of equal moduli. Multiplying every coefficient by the same nonzero scalar leaves the state unchanged. A superposition is expressed using chosen representatives and relative coefficients; adding projective points without these choices is not defined.

## 1.2. Measurements.

**1.2. Definition.** A *projective measurement* on a finite-dimensional Hilbert space  $\mathcal{H}$  is a finite family  $(P_a)_{a \in A}$  of orthogonal projections with  $\sum_a P_a = I$ , indexed by a set  $A$  of *outcomes*. Measuring a pure state  $[u]$  produces outcome  $a$  with probability

$$p_a([u]) = \frac{\|P_a u\|^2}{\|u\|^2}.$$

Conditional on an outcome of positive probability, the state after the measurement is  $[P_a u]$ .

The probabilities sum to one because the projections are mutually orthogonal, and both the probabilities and the resulting states are independent of the representative  $u$ . For a unit representative the probability is  $\|P_a u\|^2$ , and  $P_a u / \|P_a u\|$  is a normalized representative of the resulting state. The expression  $[P_a u]$  is undefined when  $P_a u = 0$ , but then the outcome has probability zero. This rule (often called the Born rule) is the only point of contact between the formalism and experiment that we need; we take it as a definition.

The measurements used in these notes are all of one kind. A self-adjoint operator  $g$  with  $g^2 = I$  has spectrum contained in  $\{1, -1\}$ , and its spectral projections are

$$P_+ = \frac{I + g}{2}, \quad P_- = \frac{I - g}{2}.$$

We call  $(P_+, P_-)$  the *measurement of  $g$*  and label its outcomes by the eigenvalues  $\pm 1$ . Two features of this measurement deserve emphasis. First, if  $[u]$  lies in the projectivization of an eigenspace of  $g$ , the outcome is certain and the state is unchanged. Second, the outcome distribution depends on  $[u]$  only through the two numbers  $\|P_\pm u\|^2 / \|u\|^2$ ; a measurement of  $g$  learns nothing else about the state. The sign of the chosen operator labels its two outcomes:  $g$  and  $-g$  induce the same projective transformation but exchange the labels of the eigenspaces.

**1.3. Proposition.** Let  $g_1, \dots, g_r$  be commuting self-adjoint involutions. For  $\sigma \in \{1, -1\}^r$ , the operators  $P_\sigma := \prod_j (I + \sigma_j g_j) / 2$  form a projective measurement whose outcome distribution and post-measurement state coincide with those of measuring  $g_1, \dots, g_r$  successively, in any order.

*Proof.* The factors are commuting orthogonal projections, so each product is an orthogonal projection; expanding  $\prod_j ((I + g_j)/2 + (I - g_j)/2) = I$  shows that the  $P_\sigma$  sum to the identity, and distinct  $\sigma$  give mutually orthogonal products. On input  $[u]$ , successive measurement produces outcome sequence  $\sigma$

with probability  $\|P_\sigma u\|^2/\|u\|^2$  and, when this is positive, leaves the state  $[P_\sigma u]$ , by repeated application of [Definition 1.2](#) and commutativity.  $\square$

We shall therefore speak of *jointly measuring* commuting involutions, and of the resulting sign vector  $\sigma$ . How such a measurement is realized physically is irrelevant for the mathematics; [Section B.4](#) describes one standard realization.

### 1.3. Encodings and codes.

**1.4. Definition.** An *encoding* of  $k$  qubits into  $n$  qubits is a projective embedding

$$\mathbb{P}(\mathcal{H}_k) \longrightarrow \mathbb{P}(\mathcal{H}_n), \quad [u] \longmapsto [Vu],$$

induced by a linear isometry  $V : \mathcal{H}_k \rightarrow \mathcal{H}_n$ , called a *linear lift* of the encoding. Its image  $\mathbb{P}(\mathcal{C})$  is the *code*, where  $\mathcal{C} := V(\mathcal{H}_k)$  is its *underlying code subspace*; elements of  $\mathcal{C}$  are called *code vectors*. The code has projective dimension  $2^k - 1$ ; its underlying subspace has dimension  $2^k$ .

An encoding preserves the transition probabilities of [Equation \(1.1\)](#). More generally, a *unitary transformation* of pure states is a map  $[u] \mapsto [Uu]$  induced by a unitary operator  $U$ ; scalar unitary multiples give the same transformation. General projective linear transformations need not preserve transition probabilities ([Section A.2](#)). We name a code by its underlying subspace  $\mathcal{C}$  when convenient; all dimensions of code subspaces in the parameter notation  $[[n, k, d]]$  are Hilbert-space dimensions. A choice of encoding includes a choice of projective identification with the logical pure-state space, not merely its image.

The goal of error correction is to restore *every* point of  $\mathbb{P}(\mathcal{C})$ , including the states represented by arbitrary superpositions of code basis vectors. Restoring only the computational states would not suffice. The recovery channels considered here then also preserve states entangled with an untouched auxiliary system ([Section B](#)); this is proved using the linear action of channels on density operators. The assignment  $[u] \mapsto [u \otimes u \otimes u]$  is a well-defined nonlinear projective map, but it is not an encoding: no linear isometry, or even linear map nonzero on every input line, induces it ([Problem 1.4](#)). This is the form of the no-cloning obstruction needed here.

### 1.4. Pauli operators.

**1.5. Definition.** The *Pauli matrices* are

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} = iXZ.$$

For  $1 \leq j \leq n$  we write  $X_j, Y_j, Z_j$  for the operator on  $\mathcal{H}_n$  acting as  $X, Y, Z$  on the  $j$ th tensor factor and as the identity on the others. The *projective Pauli group*  $\overline{\mathcal{P}}_n$  consists of the transformations  $[u] \mapsto [(P_1 \otimes \cdots \otimes P_n)u]$  with each  $P_j \in \{I, X, Y, Z\}$ . The *Pauli matrix group*  $\mathcal{P}_n$  is the group of linear lifts  $\lambda P_1 \otimes \cdots \otimes P_n$  with  $\lambda \in \{\pm 1, \pm i\}$ . The *weight*  $|E|$  of such an operator  $E$  is the number of indices  $j$  with  $P_j \neq I$ .

The transformation induced by  $X$  exchanges  $[|0\rangle]$  and  $[|1\rangle]$  and is called a *bit flip*. The transformation induced by  $Z$  is called a *phase flip*: it takes  $[\alpha|0\rangle + \beta|1\rangle]$  to  $[\alpha|0\rangle - \beta|1\rangle]$ , changing a relative sign while fixing both computational states. We specify a Pauli error by a linear lift  $E \in \mathcal{P}_n$  and mean its action  $[u] \mapsto [Eu]$ ; its scalar multiple induces the same error. The relations  $X^2 = Y^2 = Z^2 = I$  and  $XZ = -ZX$  imply that any two elements of  $\mathcal{P}_n$  either commute or anticommute, and that a Pauli operator is self-adjoint precisely when its scalar is  $\pm 1$ , in which case it squares to  $I$ . Operators acting on different tensor factors commute. The quotient  $\mathcal{P}_n/\{\pm I, \pm iI\}$  is precisely  $\overline{\mathcal{P}}_n$ ; chosen signed lifts also specify the labeled check measurements. Finally, the  $4^n$  tensor products  $P_1 \otimes \cdots \otimes P_n$  with  $P_j \in \{I, X, Y, Z\}$  form a basis of the space of all operators on  $\mathcal{H}_n$ , orthogonal for the inner product  $\langle A, B \rangle = \text{Tr}(A^\dagger B)$  (each has trace zero except the identity). This last fact is the reason why Pauli errors play a distinguished role: we return to it in [Section B.3](#).

### 1.5. The repetition code.

1.6. **Example.** The *three-qubit repetition code* is the encoding

$$[\alpha|0\rangle + \beta|1\rangle] \mapsto [\alpha|000\rangle + \beta|111\rangle],$$

whose underlying subspace  $\mathcal{C} = \text{span}\{|000\rangle, |111\rangle\}$  is the common +1 eigenspace of the commuting involutions  $g_1 = Z_1Z_2$  and  $g_2 = Z_2Z_3$ . Indeed,  $Z_iZ_j|b\rangle = (-1)^{b_i+b_j}|b\rangle$ , so  $g_1$  and  $g_2$  are diagonal in the computational basis and measure the parities  $b_1 + b_2$  and  $b_2 + b_3$ . Record the outcome  $(-1)^s$  by the bit  $s$ . For a code state  $[u]$ , choose a normalized representative  $u = \alpha|000\rangle + \beta|111\rangle$ ; we obtain:

| Error | Representative after the error         | Outcome bits $(s_1, s_2)$ |
|-------|----------------------------------------|---------------------------|
| $I$   | $\alpha 000\rangle + \beta 111\rangle$ | $(0, 0)$                  |
| $X_1$ | $\alpha 100\rangle + \beta 011\rangle$ | $(1, 0)$                  |
| $X_2$ | $\alpha 010\rangle + \beta 101\rangle$ | $(1, 1)$                  |
| $X_3$ | $\alpha 001\rangle + \beta 110\rangle$ | $(0, 1)$                  |
| $Z_1$ | $\alpha 000\rangle - \beta 111\rangle$ | $(0, 0)$                  |

Each vector in the middle column is an eigenvector of both  $g_1$  and  $g_2$ , so by [Proposition 1.3](#) the joint measurement returns the displayed bits with certainty and leaves its represented state unchanged.

The table contains the two observations on which the subject rests. First, the outcome bits depend on the error and not on  $(\alpha, \beta)$ ; measuring the checks therefore learns which single bit flip occurred, if one did, and learns nothing about the encoded state, which survives the measurement intact. Applying  $X_j$  again then restores  $[u]$ . By contrast, measuring the first qubit in the computational basis (i.e., measuring  $Z_1$ ) would produce outcome +1 with probability  $|\alpha|^2$  and return the state  $[|000\rangle]$  or  $[|111\rangle]$ , destroying a superposition with both coefficients nonzero. Second, the phase flip  $Z_1$  produces the trivial outcome and nevertheless changes a general encoded state, since  $[\alpha|000\rangle - \beta|111\rangle]$  is the encoding of  $[\alpha|0\rangle - \beta|1\rangle]$ . The repetition code protects against one bit flip and against no phase flip at all; we shall say in [Section 2](#) that its distance is one. To protect a qubit, a code needs checks that detect both kinds of error, and the checks must still commute with one another. The construction of such checks is the subject of the next section.

1.7. *Remark.* The table treats the error as one of finitely many operators. A unitary error such as  $[u] \mapsto [Uu]$ , with  $U = \cos \theta I + i \sin \theta X_1$ , has a representative  $\cos \theta u + i \sin \theta X_1 u$  with components in two different syndrome subspaces. The joint measurement then returns  $(0, 0)$  with probability  $\cos^2 \theta$  and  $(1, 0)$  with probability  $\sin^2 \theta$ , leaving  $[u]$  or  $[X_1 u]$ , respectively; in either case the prescribed correction restores  $[u]$ . The coherent sum specifies linear lifts and their relative coefficients: replacing  $X_1$  by  $-X_1$  in this sum can change the induced transformation, although these two lifts induce the same Pauli error separately. The measurement has, in effect, discretized the error. [Section B.3](#) turns this observation into a theorem.

### 1.6. Problems.

- 1.1. For the repetition code, compute the outcome bits produced by  $X_1X_3$ ,  $Z_2$ , and  $Y_1$  on an arbitrary code vector. Which errors among  $I, X_1, X_2, X_3, X_1X_3$  can be corrected by a single rule that depends only on the outcome bits?
- 1.2. Write the projections  $P_\sigma$  of [Proposition 1.3](#) for  $g_1 = Z_1Z_2$ ,  $g_2 = Z_2Z_3$  as diagonal matrices in the computational basis, and verify that they sum to the identity.
- 1.3. Let  $|\pm\rangle := (|0\rangle \pm |1\rangle)/\sqrt{2}$ . Show that  $X|\pm\rangle = \pm|\pm\rangle$  and  $Z|\pm\rangle = |\mp\rangle$ , and deduce that the projectivization of  $\text{span}\{|++\rangle, |--\rangle\}$ , the common +1 eigenspace of  $X_1X_2$  and  $X_2X_3$ , corrects one phase flip and no bit flip.
- 1.4. Prove that no linear map  $T : \mathbb{C}^2 \rightarrow (\mathbb{C}^2)^{\otimes 3}$ , nonzero on every nonzero input, satisfies  $[Tu] = [u \otimes u \otimes u]$  for every pure state  $[u]$ . Explain why this does not prevent the assignment  $[u] \mapsto [u \otimes u \otimes u]$  from defining a nonlinear map of projective spaces.

## 2. STABILIZER CODES

The repetition code is the projectivization of the common  $+1$  eigenspace of two commuting Pauli lifts. This section takes that description as a definition, following Gottesman [Got97]. We compute the dimension of such codes, define their syndromes and logical operators, prove that a code of distance  $d$  corrects every Pauli error of weight less than  $d/2$ , and translate the whole theory into linear algebra over  $\mathbb{F}_2$ . The section ends with the Calderbank–Shor–Steane construction and Shor’s nine-qubit code.

### 2.1. Stabilizer groups and their codes.

**2.1. Definition.** A *stabilizer group* is an abelian subgroup  $S \leq \mathcal{P}_n$  of linear Pauli lifts with  $-I \notin S$ . Its *stabilizer code* is  $\mathbb{P}(\mathcal{C}(S))$ , where the underlying code subspace is

$$\mathcal{C}(S) := \{u \in \mathcal{H}_n : gu = u \text{ for every } g \in S\}.$$

The condition  $-I \notin S$  is exactly what makes  $\mathcal{C}(S)$  nonzero, as the next theorem shows; it also forces every element of  $S$  to be self-adjoint, since an element with scalar  $\pm i$  squares to  $-I$ . Thus  $S$  consists of commuting involutions, and every element other than  $I$  has order two; hence  $S \cong \mathbb{F}_2^r$  for some  $r$ , and  $S$  is generated by  $r$  elements  $g_1, \dots, g_r$ , none of which is a product of the others. We call such generators *independent* and always assume that a stabilizer group is presented by independent generators. Since  $\mathcal{C}(S)$  is the common  $+1$  eigenspace of  $g_1, \dots, g_r$ , the generators are also called the *checks* of the code. Their signs select the eigenspaces labeled  $+1$ . Their common projective fixed locus records eigenlines rather than chosen eigenvalues; in general it contains other syndrome sectors as well as the code. For example, the transformation induced by  $Z$  fixes both  $[|0\rangle]$  and  $[|1\rangle]$ , while its  $+1$  eigenspace selects only  $[|0\rangle]$ .

**2.2. Theorem.** Let  $S$  be a stabilizer group with  $r$  independent generators. The orthogonal projection onto  $\mathcal{C}(S)$  is

$$(2.1) \quad P = \prod_{j=1}^r \frac{I + g_j}{2} = \frac{1}{2^r} \sum_{g \in S} g,$$

and  $\dim \mathcal{C}(S) = 2^{n-r}$ .

*Proof.* The factors  $(I + g_j)/2$  are commuting orthogonal projections onto the  $+1$  eigenspaces of the  $g_j$ , so their product is the orthogonal projection onto the intersection of these eigenspaces, i.e., onto  $\mathcal{C}(S)$ . Expanding the product gives  $2^{-r}$  times the sum of all products of subsets of the generators, which by independence are exactly the  $2^r$  elements of  $S$ , each appearing once. An element  $g \neq I$  of  $S$  is not a scalar multiple of the identity (because  $-I \notin S$  and the scalars  $\pm i$  have been excluded above), so it is  $\pm$  a nontrivial tensor product of Pauli matrices and  $\text{Tr } g = 0$ . Hence  $\dim \mathcal{C}(S) = \text{Tr } P = 2^{-r} \text{Tr } I = 2^{n-r}$ .  $\square$

We write  $k := n - r$ ; thus  $\mathbb{P}(\mathcal{C}(S)) \cong \mathbb{C}\mathbb{P}^{2^k-1}$  and the code *encodes  $k$  qubits into  $n$* . For the repetition code,  $S = \{I, Z_1 Z_2, Z_2 Z_3, Z_1 Z_3\}$ ,  $r = 2$ , and  $k = 1$ .

**2.2. Syndromes.** Every Pauli operator  $E$  commutes or anticommutes with each generator. Define  $s_j(E) \in \mathbb{F}_2$  by

$$g_j E = (-1)^{s_j(E)} E g_j,$$

and call  $s(E) = (s_1(E), \dots, s_r(E)) \in \mathbb{F}_2^r$  the *syndrome* of  $E$ . If  $u \in \mathcal{C}(S)$ , then

$$(2.2) \quad g_j(Eu) = (-1)^{s_j(E)} E g_j u = (-1)^{s_j(E)} Eu.$$

**2.3. Proposition.** Let  $E \in \mathcal{P}_n$  and let  $[u] \in \mathbb{P}(\mathcal{C}(S))$  be a code state. Jointly measuring  $g_1, \dots, g_r$  on  $[Eu]$  produces the outcomes  $((-1)^{s_j(E)})_j$  with probability one and leaves  $[Eu]$  unchanged. The map  $s : \mathcal{P}_n \rightarrow \mathbb{F}_2^r$  is a group homomorphism whose kernel is the centralizer

$$\mathcal{Z}(S) := \{R \in \mathcal{P}_n : Rg = gR \text{ for every } g \in S\}.$$

*Proof.* By Equation (2.2),  $Eu$  lies in a joint eigenspace of the generators, so the first assertion is the remark following Definition 1.2, applied to the joint measurement of Proposition 1.3. If  $E$  and  $F$  each commute or anticommute with  $g_j$ , then  $g_j EF = (-1)^{s_j(E)+s_j(F)} EF g_j$ , which is the homomorphism property; the kernel consists of the operators commuting with every generator, hence with all of  $S$ .  $\square$

Multiplying  $E$  by a nonzero scalar does not change its commutation signs, so the syndrome depends only on the projective Pauli error. Thus it is a function of the error alone, not of the encoded state, and measuring it leaves that state after the error unchanged. This is the precise sense in which the checks reveal the error without revealing the information. We shall see in Proposition 2.8 that every vector of  $\mathbb{F}_2^r$  is the syndrome of some Pauli operator; consequently  $\mathcal{H}_n$  is the orthogonal direct sum of the  $2^r$  joint eigenspaces of the generators, each of the form  $E\mathcal{C}(S)$  and of dimension  $2^k$  (Problem 2.2).

**2.3. Logical operators and distance.** Two Pauli errors with the same syndrome cannot be distinguished by the checks. Their quotient lies in the kernel of the syndrome map, and the following proposition decides whether that ambiguity is harmful. Write  $\langle iI \rangle S$  for the group generated by  $S$  and the scalar  $iI$ , i.e., the set of operators  $\lambda g$  with  $g \in S$  and  $\lambda \in \{\pm 1, \pm i\}$ .

**2.4. Proposition.** *Every  $R \in \mathcal{Z}(S)$  induces a transformation of  $\mathbb{P}(\mathcal{C}(S))$ ; this transformation is the identity if and only if  $R \in \langle iI \rangle S$ . If  $R = \lambda g$  with  $g \in S$ , then  $R$  acts on  $\mathcal{C}(S)$  as the scalar  $\lambda$ . If  $R \notin \langle iI \rangle S$ , then the restriction of  $R$  to  $\mathcal{C}(S)$  has trace zero; in particular, it is not a scalar multiple of the identity of  $\mathcal{C}(S)$ .*

*Proof.* Since  $R$  commutes with every  $g \in S$ , it commutes with  $P$  and preserves its range. If  $R = \lambda g$  with  $g \in S$ , then  $Ru = \lambda u$  for  $u \in \mathcal{C}(S)$  by the definition of  $\mathcal{C}(S)$ . If  $R \notin \langle iI \rangle S$ , Equation (2.1) gives  $\text{Tr}(PRP) = \text{Tr}(PR) = 2^{-r} \sum_{g \in S} \text{Tr}(gR)$ . Each  $gR$  is a Pauli operator, and it is a scalar only if  $R$  is a scalar multiple of  $g^{-1} = g$ , which would place  $R$  in  $\langle iI \rangle S$ ; hence every term vanishes. A nonzero scalar multiple of the identity of a space of positive dimension has nonzero trace, and  $R$  is invertible, so the restriction is not scalar. Finally, a linear automorphism induces the identity on projective space precisely when it is scalar: apply it to basis vectors and to their pairwise sums to see that the eigenvalues on all basis vectors agree. This proves the projective assertion as well.  $\square$

**2.5. Definition.** A *logical Pauli transformation* is a transformation of  $\mathbb{P}(\mathcal{C}(S))$  induced by a Pauli operator with zero syndrome. A *logical operator* is a Pauli lift inducing a nonidentity logical transformation, i.e., an element of  $\mathcal{Z}(S) \setminus \langle iI \rangle S$ . If  $k \geq 1$ , the *distance*  $d$  of the code is the minimum weight of a logical operator, and we call  $\mathbb{P}(\mathcal{C}(S))$  an  $[[n, k, d]]$  code.

By Proposition 2.4, a logical operator commutes with every check (so it is invisible to the syndrome) and moves some point of  $\mathbb{P}(\mathcal{C}(S))$  (so it corrupts the encoded information). This is the formal meaning of a *logical error*. In the repetition code,  $Z_1$  commutes with  $Z_1Z_2$  and  $Z_2Z_3$  and is not in  $\langle iI \rangle S$ ; it is a logical operator of weight one, so the repetition code is a  $[[3, 1, 1]]$  code.

**2.6. Theorem.** *Let  $\mathbb{P}(\mathcal{C}(S))$  be an  $[[n, k, d]]$  stabilizer code and let  $t$  be an integer with  $2t < d$ . There is a function  $\rho : \mathbb{F}_2^r \rightarrow \mathcal{P}_n$  such that, for every Pauli error  $E \in \mathcal{P}_n$  of weight at most  $t$ ,*

$$[\rho(s(E))Eu] = [u] \quad \text{for every } [u] \in \mathbb{P}(\mathcal{C}(S)).$$

*Equivalently, the lift  $\rho(s(E))E$  restricts to a scalar of modulus one on  $\mathcal{C}(S)$ .*

*Proof.* For each syndrome  $\sigma$  that occurs among the errors of weight at most  $t$ , choose one such error  $E_\sigma$  and set  $\rho(\sigma) = E_\sigma^\dagger$ ; elsewhere define  $\rho$  arbitrarily. If  $E$  has weight at most  $t$  and syndrome  $\sigma$ , then  $R = E_\sigma^\dagger E$  has trivial syndrome, i.e.,  $R \in \mathcal{Z}(S)$  by Proposition 2.3, and  $|R| \leq 2t < d$ . By the definition of distance,  $R$  is not a logical operator, so  $R \in \langle iI \rangle S$  and Proposition 2.4 shows that  $R$  acts on  $\mathcal{C}(S)$  as a scalar of modulus one.  $\square$

The recovery therefore restores every code state, without prescribing which normalized representative is returned. Note that the theorem does not require  $\rho(s(E))$  to undo  $E$  itself: the recovery may differ from the inverse of the actual error by an element of the stabilizer.

**2.7. Definition.** An  $[[n, k, d]]$  stabilizer code is *degenerate* if its stabilizer group contains an element other than  $I$  of weight less than  $d$ .

In a degenerate code, two distinct errors  $E$  and  $gE$  with  $g \in S$  may both have weight less than  $d$ ; they have the same syndrome and the same effect on the code, so it is neither possible nor necessary to tell them apart. (Terminology varies: such codes are also called *impure*, and “degenerate” is sometimes defined relative to a specified set of correctable errors [Got97].) Degeneracy has no counterpart for classical linear codes, where two distinct errors with the same syndrome differ by a nonzero codeword, so that correcting one by the other always fails; it is the source of much of the difficulty of quantum decoding (Section 5). Theorem B.2 in the appendix shows that the same recovery corrects every error operator in the linear span of the Pauli errors of weight at most  $t$ , and indeed every quantum channel with such Kraus operators; that is the sense in which correcting finitely many discrete errors corrects a continuum of them.

**2.4. Binary coordinates.** Projective Pauli errors admit binary coordinates; the scalar commutators of their lifts retain the information needed for syndromes. For  $x, z \in \mathbb{F}_2^n$  define

$$X(x) := \prod_{j=1}^n X_j^{x_j}, \quad Z(z) := \prod_{j=1}^n Z_j^{z_j}.$$

Every element of  $\mathcal{P}_n$  equals  $\lambda X(x)Z(z)$  for a unique pair  $(x, z)$  and a unique scalar  $\lambda$ , and the product of two Pauli operators corresponds, up to a scalar, to the sum of their pairs. In other words,  $(x, z) \mapsto [X(x)Z(z)]$  gives a group isomorphism  $\mathbb{F}_2^{2n} \cong \overline{\mathcal{P}}_n = \mathcal{P}_n / \langle iI \rangle$ . On the  $j$ th qubit,  $(x_j, z_j) = (1, 0), (0, 1), (1, 1)$  correspond to  $X, Z$ , and  $Y$  (up to phase), so the weight of  $X(x)Z(z)$  is the number of indices with  $(x_j, z_j) \neq (0, 0)$ ; a  $Y$  error has weight one, not two.

**2.8. Proposition.** The operators  $X(x)Z(z)$  and  $X(x')Z(z')$  commute if and only if

$$(2.3) \quad \omega((x, z), (x', z')) := x \cdot z' + z \cdot x' = 0 \quad \text{in } \mathbb{F}_2.$$

The form  $\omega$  is bilinear, alternating, and nondegenerate. Consequently, if the generators  $g_j$  of a stabilizer group correspond to the vectors  $v_j = (a_j, b_j) \in \mathbb{F}_2^{2n}$ , then:

- (1) the vectors  $v_1, \dots, v_r$  are linearly independent and pairwise  $\omega$ -orthogonal;
- (2) the syndrome of  $X(x)Z(z)$  is  $(a_j \cdot z + b_j \cdot x)_j$ , which is a surjective linear function of  $(x, z)$ ;
- (3) the centralizer  $\mathcal{Z}(S)$  corresponds to the  $\omega$ -orthogonal complement of  $\text{span}\{v_j\}$ , a subspace of dimension  $2n - r$  containing the  $r$ -dimensional image of  $S$ .

*Proof.* Moving each  $Z_j$  factor of  $Z(z)$  past each  $X_j$  factor of  $X(x')$  on the same qubit produces a sign, and factors on different qubits commute; this gives the sign  $(-1)^{x \cdot z' + z \cdot x'}$  relating the two products. Bilinearity and  $\omega(v, v) = 2x \cdot z = 0$  are clear, and  $\omega((x, z), \cdot)$  is the functional  $(x', z') \mapsto x \cdot z' + z \cdot x'$ , which vanishes identically only if  $x = z = 0$ . A linear relation among the  $v_j$  would make a nonempty product of generators a scalar, hence equal to  $I$  (the only scalar in  $S$ ), contradicting independence; commutativity of  $S$  is pairwise orthogonality. The syndrome formula is Equation (2.3) applied to each generator. It is surjective because  $v \mapsto (\omega(v_j, v))_j$  is a linear map whose transpose is injective (by nondegeneracy and the independence of the  $v_j$ ). The kernel of a surjective linear map onto  $\mathbb{F}_2^r$  has dimension  $2n - r$ .  $\square$

Although  $\overline{\mathcal{P}}_n$  is abelian, the form  $\omega$  need not vanish: it records the scalar commutators of lifts, independently of their scalar choices. Thus the centralizer  $\mathcal{Z}(S)$  above is computed for the chosen linear checks; taking a centralizer inside the abelian projective Pauli group would lose the syndrome condition. The image of  $S$  has dimension  $r$  and that of  $\mathcal{Z}(S)$  has dimension  $2n - r$ , so the induced logical Pauli transformations form a quotient space of dimension  $2n - 2r = 2k$ . One can choose  $2k$  logical operators  $\overline{X}_1, \overline{Z}_1, \dots, \overline{X}_k, \overline{Z}_k$  that satisfy the same commutation relations as  $X_1, Z_1, \dots, X_k, Z_k$  on  $\mathcal{H}_k$ , and these act

on  $\mathcal{C}(S) \cong \mathcal{H}_k$  as the Pauli operators of  $k$  encoded qubits; we only need the case  $k = 1$ , where it suffices to exhibit two anticommuting logical operators ([Problem 2.3](#)).

**2.5. Calderbank–Shor–Steane codes.** In the codes of interest to us, each check is a product of  $X$ ’s only or of  $Z$ ’s only.

**2.9. Definition.** Let  $H_X$  and  $H_Z$  be binary matrices with  $n$  columns satisfying

$$(2.4) \quad H_X H_Z^\top = 0.$$

The CSS code with these matrices is the stabilizer code whose stabilizer group is generated by the  $X$  checks  $X(a)$ , for  $a$  a row of  $H_X$ , and the  $Z$  checks  $Z(b)$ , for  $b$  a row of  $H_Z$ .

The construction is due independently to Calderbank and Shor [[CS96](#)] and to Steane [[Ste96](#)]. By [Equation \(2.3\)](#), an  $X$  check and a  $Z$  check commute exactly when their supports meet in an even number of qubits, so [Equation \(2.4\)](#) is the commutativity of the stabilizer group; checks of the same type always commute. The rows of  $H_X$  and  $H_Z$  need not be independent, and only the ranks matter: by [Theorem 2.2](#),

$$(2.5) \quad k = n - \text{rank } H_X - \text{rank } H_Z.$$

By [Proposition 2.8](#), an error  $X(x)Z(z)$  has  $X$ -check syndrome and  $Z$ -check syndrome

$$(2.6) \quad s_X = H_X z, \quad s_Z = H_Z x.$$

The opposite subscripts are not a misprint:  $X$  checks detect the  $Z$  component of an error and  $Z$  checks detect its  $X$  component. In particular, the two components of an error are detected by different checks, and for many purposes the  $X$  and  $Z$  parts of a CSS code can be treated separately.

**2.10. Theorem.** Suppose an  $X$  error  $X(x)$  is followed by a correction  $X(\hat{x})$  with  $H_Z \hat{x} = H_Z x$ . The composite fixes every point of the projective code if and only if  $x + \hat{x} \in \text{row}(H_X)$ . The possible residuals modulo stabilizers form the quotient space

$$\ker H_Z / \text{row}(H_X),$$

of dimension  $k$ . The same statements hold for  $Z$  errors after exchanging the roles of  $X$  and  $Z$ .

*Proof.* The residual  $X(x + \hat{x})$  has zero  $Z$ -check syndrome and, being an  $X$  operator, zero  $X$ -check syndrome; hence it lies in  $\mathcal{Z}(S)$ . A pure  $X$  operator lies in  $\langle iI \rangle S$  precisely when its vector is a sum of rows of  $H_X$  ([Problem 2.4](#)), so the first assertion is [Proposition 2.4](#). The dimension of the quotient is  $(n - \text{rank } H_Z) - \text{rank } H_X = k$ .  $\square$

Thus the equation  $H_Z \hat{x} = s_Z$  is a constraint on a correction, not a certificate of its success. Its solutions form an affine space, partitioned into  $2^k$  cosets of  $\text{row}(H_X)$ , and only one of these cosets restores the encoded information. The checks cannot tell which one: that choice is the decoding problem of [Section 4](#).

For CSS codes the distance splits accordingly. Let  $d_X$  be the minimum weight of a vector of  $\ker H_Z \setminus \text{row}(H_X)$  and  $d_Z$  the minimum weight of a vector of  $\ker H_X \setminus \text{row}(H_Z)$ . Then  $d = \min(d_X, d_Z)$ . Indeed,  $X(x)Z(z)$  lies in  $\mathcal{Z}(S)$  if and only if  $x \in \ker H_Z$  and  $z \in \ker H_X$ , and it lies in  $\langle iI \rangle S$  if and only if, in addition,  $x \in \text{row}(H_X)$  and  $z \in \text{row}(H_Z)$ ; a logical operator therefore has a pure component that is itself logical, and the weight of  $X(x)Z(z)$  is at least the weight of each component.

**2.11. Remark.** Decoding the two components separately is an algorithmic choice, not a consequence of [Equation \(2.6\)](#). In the *depolarizing* model, each qubit independently suffers  $X$ ,  $Y$ , or  $Z$  with probability  $p/3$  each. Each binary component then has marginal error probability  $2p/3$  per qubit, but the components are correlated, because a  $Y$  error sets both; a decoder that ignores this correlation discards information.

**2.6. Shor's nine-qubit code.** The first code to protect a qubit against an arbitrary single-qubit error was constructed by Shor [Sho95], by repeating the phase-flip code of Problem 1.3 with blocks of the bit-flip code.

**2.12. Example.** Arrange nine qubits in three blocks  $\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}$ . The CSS code with

$$H_Z = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad H_X = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

has Z checks  $Z_1Z_2, Z_2Z_3, \dots, Z_8Z_9$ , two within each block, and X checks  $X_1 \cdots X_6$  and  $X_4 \cdots X_9$ . Each X check meets each Z check in zero or two qubits, so Equation (2.4) holds; the ranks are 6 and 2, so  $k = 1$  by Equation (2.5). The underlying code subspace is spanned by the normalized representatives

$$|\bar{0}\rangle = \frac{1}{2\sqrt{2}}(|000\rangle + |111\rangle)^{\otimes 3}, \quad |\bar{1}\rangle = \frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle)^{\otimes 3}.$$

We compute the distance directly. A vector  $x$  lies in  $\ker H_Z$  precisely when it is constant on each block, and  $\text{row}(H_X)$  consists of the vectors equal to 1 on an even number of blocks and 0 on the others; hence the vectors of  $\ker H_Z \setminus \text{row}(H_X)$  are those equal to 1 on an odd number of blocks, and  $d_X = 3$ , attained by  $X_1X_2X_3$ . Similarly,  $z \in \ker H_X$  precisely when the parities  $\pi_1, \pi_2, \pi_3$  of  $z$  on the three blocks satisfy  $\pi_1 = \pi_2 = \pi_3$ , while  $\text{row}(H_Z)$  consists of the vectors of even parity on every block; hence  $z \in \ker H_X \setminus \text{row}(H_Z)$  has odd parity on every block, and  $d_Z = 3$ , attained by  $Z_1Z_4Z_7$ . Thus Shor's code is a  $[[9, 1, 3]]$  code, and by Theorem 2.6 it corrects every single-qubit Pauli error.

Note which logical operators these are. The operator  $Z_1$  maps  $|000\rangle + |111\rangle$  to  $|000\rangle - |111\rangle$ , so  $Z_1Z_4Z_7$  exchanges  $|\bar{0}\rangle$  and  $|\bar{1}\rangle$ ; it is a logical X, although it is a product of Z's. Likewise  $X_1X_2X_3$  fixes  $|\bar{0}\rangle$  and negates  $|\bar{1}\rangle$ , so it is a logical Z. There is no contradiction: which Pauli operator of the encoded qubit a logical operator implements depends on the chosen encoding, not on the type of its physical factors. Shor's code is also degenerate in the sense of Definition 2.7, since the check  $Z_1Z_2$  has weight 2 < 3:  $Z_1$  and  $Z_2$  have the same syndrome and differ by that check, so correcting either one by the other succeeds.

The surface patch of Section 3 has the same parameters  $[[9, 1, 3]]$ . It differs from Shor's code in that its checks have weight at most four and act on neighbouring qubits of a square grid, and, more importantly, in that it is the first member of a family whose distance grows with the number of qubits while the check weights stay bounded.

## 2.7. Problems.

- 2.1.** Show that if an abelian subgroup  $S \leq \mathcal{P}_n$  contains  $-I$ , then its common +1 eigenspace is zero.
- 2.2.** Let  $E_\sigma$  be a Pauli operator with syndrome  $\sigma$  (which exists by Proposition 2.8). Show that  $E_\sigma \mathcal{C}(S)$  is the joint eigenspace of  $g_1, \dots, g_r$  with eigenvalues  $((-1)^{\sigma_j})_j$ , that it is independent of the choice of  $E_\sigma$ , and that  $\mathcal{H}_n$  is the orthogonal direct sum of these  $2^r$  spaces.
- 2.3.** Let  $\mathcal{C}(S)$  encode one qubit, and let  $\bar{X}, \bar{Z}$  be self-adjoint logical operators that anticommute. Show that there is an orthonormal basis of  $\mathcal{C}(S)$  in which  $\bar{Z}$  and  $\bar{X}$  act by the matrices  $Z$  and  $X$ .
- 2.4.** For a CSS code, show that  $X(x) \in \langle iI \rangle S$  if and only if  $x \in \text{row}(H_X)$ .
- 2.5.** Let  $H$  be the  $3 \times 7$  binary matrix whose  $j$ th column is the binary expansion of  $j$ . Show that  $HH^T = 0$ , so that  $H_X = H_Z = H$  defines a CSS code, and that this code (Steane's code) is a  $[[7, 1, 3]]$  code.
- 2.6.** In Shor's code, compute the syndromes of  $Z_1, Z_4, X_5$ , and  $Y_5$ . Which single-qubit errors share a syndrome, and why does this not prevent their correction?

## 3. SURFACE CODES

Surface codes are CSS codes whose checks are attached to the faces of a square grid. They were introduced by Kitaev [Kit03] on a torus and by Bravyi and Kitaev [BK98] on a planar region with boundary; the variant used here, often called the *rotated* surface code, achieves the same distance with

roughly half as many qubits [BMD07, TS14]. We first describe the smallest nontrivial patch completely and then prove that the patch of odd distance  $d$  is a  $[[d^2, 1, d]]$  code. Fowler et al. [FMMC12] give a detailed account of surface codes oriented toward implementation, and Dennis et al. [DKLP02] remains the basic reference for their decoding.

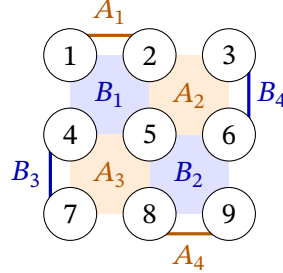


FIGURE 1. The rotated surface patch of distance three. Circles are the nine data qubits, labeled in row order. Orange squares and bars are X checks  $A_j$ , and blue ones are Z checks  $B_j$ ; a square acts on its four corners and a bar on its two endpoints. The labels name checks, not additional qubits.

3.1. **A nine-qubit patch.** The checks of Figure 1 are

$$\begin{aligned} A_1 &= X_1 X_2, & A_2 &= X_2 X_3 X_5 X_6, & A_3 &= X_4 X_5 X_7 X_8, & A_4 &= X_8 X_9, \\ B_1 &= Z_1 Z_2 Z_4 Z_5, & B_2 &= Z_5 Z_6 Z_8 Z_9, & B_3 &= Z_4 Z_7, & B_4 &= Z_3 Z_6, \end{aligned}$$

i.e., the rows of

$$(3.1) \quad H_X = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \quad H_Z = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

Every X check meets every Z check in zero or two qubits (for instance,  $A_2$  meets  $B_1, B_2, B_4$  in  $\{2, 5\}, \{5, 6\}, \{3, 6\}$ ), so Equation (2.4) holds. Both matrices have rank four, so the patch encodes  $k = 9 - 4 - 4 = 1$  qubit. The operators

$$\bar{X} := X_1 X_4 X_7, \quad \bar{Z} := Z_1 Z_2 Z_3$$

commute with every check and anticommute with each other, since their supports meet only in qubit 1. Neither belongs to  $\langle iI \rangle S$ , because an element of  $\langle iI \rangle S$  commutes with every element of  $\mathcal{Z}(S)$ ; hence both are logical operators, and by Problem 2.3 they act on the code as the Pauli matrices of the encoded qubit.

The patch exhibits, on a small scale, every phenomenon that concerns us. A single X error on qubit 1 flips only the check  $B_1$ . The two-qubit operator  $X_1 X_2 = A_1$  flips no check and acts trivially on the code, being a stabilizer. The three-qubit operator  $X_1 X_4 X_7 = \bar{X}$  also flips no check but acts nontrivially. Both are “invisible strings”; one is harmless and the other is a logical error. Exhaustive enumeration of  $\ker H_Z$  and  $\ker H_X$  confirms that no logical operator has weight less than three, so the patch is a  $[[9, 1, 3]]$  code; we now prove this, and its generalization, without enumeration.

### 3.2. The rotated patch of distance $d$ .

3.1. **Definition.** Let  $d \geq 3$  be odd. The *rotated surface patch of distance  $d$*  has  $n = d^2$  data qubits at the points  $(r, c)$ ,  $0 \leq r, c \leq d - 1$ , of a square grid, with row 0 at the top; qubit  $(r, c)$  receives the label  $rd + c + 1$ . Its checks are:

- (1) for  $0 \leq r, c \leq d - 2$ , the *square* on  $(r, c), (r, c + 1), (r + 1, c), (r + 1, c + 1)$ , which is a Z check if  $r + c$  is even and an X check if  $r + c$  is odd;

- (2) the X checks on  $\{(0, c), (0, c + 1)\}$  for even  $c$  (top boundary) and on  $\{(d - 1, c), (d - 1, c + 1)\}$  for odd  $c$  (bottom boundary);
- (3) the Z checks on  $\{(r, 0), (r + 1, 0)\}$  for odd  $r$  (left boundary) and on  $\{(r, d - 1), (r + 1, d - 1)\}$  for even  $r$  (right boundary).

The rule for the boundary checks has a simple description: extend the checkerboard of squares by one row or column beyond each side, and retain the two-qubit halves of those exterior squares whose type is X on the top and bottom sides and Z on the left and right sides. For  $d = 3$  the definition reproduces Figure 1 exactly. Figure 2 shows  $d = 5$ .

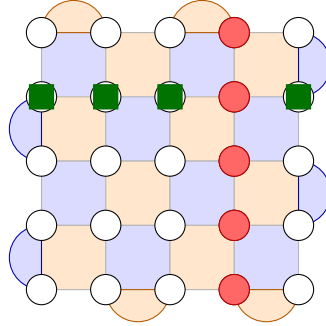


FIGURE 2. The rotated patch of distance five, with X checks orange and Z checks blue. Red circles mark the support of the logical operator  $X_4X_9X_{14}X_{19}X_{24}$  (a column); green squares mark the support of  $Z_6Z_7Z_8Z_9Z_{10}$  (a row). The two supports meet only in qubit 9, which belongs to both and is drawn as a red circle.

There are  $(d - 1)^2$  squares and  $2(d - 1)$  boundary checks ( $(d - 1)/2$  on each side), half of each type in each case; hence the patch has  $d^2 - 1$  checks,  $(d^2 - 1)/2$  of each type. Every check has weight two or four, and every qubit lies in at most four checks.

**3.2. Proposition.** *Every X check of the rotated patch meets every Z check in an even number of qubits.*

*Proof.* Two squares of opposite types are not diagonal neighbours (diagonal neighbours have the same parity of  $r + c$ ), so they share either a whole edge (two qubits) or nothing. A boundary check is the half of an exterior square of its type, so it meets an interior square of the opposite type, if at all, along the shared edge. Finally, two boundary checks of opposite types can meet only at a corner of the grid, and the parity conventions of Definition 3.1 exclude this (Problem 3.1).  $\square$

**3.3. Proposition.** *The X checks of the rotated patch are linearly independent, i.e., the rows of  $H_X$  are linearly independent over  $\mathbb{F}_2$ , and the same holds for the Z checks. Consequently the rotated patch encodes exactly one qubit.*

*Proof.* Order the qubits lexicographically by  $(r, c)$ , which is the order of their labels. The largest qubit of the square with corner  $(r, c)$  is  $(r + 1, c + 1)$ , and the largest qubit of a boundary check  $\{q, q'\}$  with  $q < q'$  is  $q'$ . We claim that distinct X checks have distinct largest qubits. A square and a top boundary check never share their largest qubit, since the latter lies in row 0. The square with corner  $(d - 2, c - 1)$  and the bottom boundary check on  $\{(d - 1, c - 1), (d - 1, c)\}$  share the largest qubit  $(d - 1, c)$ , but the first is an X check only if  $(d - 2) + (c - 1)$  is odd, i.e.,  $d + c$  is even, and the second only if  $c - 1$  is odd, i.e.,  $c$  is even; as  $d$  is odd, these conditions are incompatible. All other coincidences are excluded because the corner of a square is determined by its largest qubit.

Now suppose that a nonempty set  $T$  of X checks has product  $I$ , and let  $q$  be the largest of the largest qubits of the checks in  $T$ . Every check in  $T$  that contains  $q$  has largest qubit at least  $q$ , hence equal to  $q$ , so by the claim exactly one check in  $T$  contains  $q$ . The product of  $T$  therefore acts as  $X$  on  $q$ , a contradiction. The quarter-turn rotation  $(r, c) \mapsto (c, d - 1 - r)$  of the grid permutes the qubits and maps

the set of X checks onto the set of Z checks (Problem 3.2), so the Z checks are independent as well. The ranks are thus  $(d^2 - 1)/2$  each, and Equation (2.5) gives  $k = d^2 - (d^2 - 1) = 1$ .  $\square$

**3.3. Logical operators and distance.** For  $0 \leq i \leq d - 1$  write  $z^{(i)} \in \mathbb{F}_2^n$  for the indicator of row  $i$  and  $x^{(i)}$  for the indicator of column  $i$ .

**3.4. Proposition.** *For every  $i, j$ , the operators  $Z(z^{(i)})$  and  $X(x^{(j)})$  are logical operators, and they anticommute. Any two rows represent the same class in  $\ker H_X / \text{row}(H_Z)$ , and any two columns the same class in  $\ker H_Z / \text{row}(H_X)$ .*

*Proof.* An X square meets a row in zero or two qubits, and the X boundary checks lie inside rows 0 and  $d - 1$ ; hence  $z^{(i)} \in \ker H_X$ . Symmetrically  $x^{(j)} \in \ker H_Z$ . A row and a column meet in exactly one qubit, so  $Z(z^{(i)})$  and  $X(x^{(j)})$  anticommute; in particular neither lies in  $\langle iI \rangle S$ , whose elements commute with all of  $\mathcal{Z}(S)$ . Since  $k = 1$ , each quotient in Theorem 2.10 has exactly one nonzero class, which therefore contains every row (respectively every column).  $\square$

The equivalence of two adjacent rows can also be seen directly:  $z^{(i)} + z^{(i+1)}$  is the sum of the Z checks lying between rows  $i$  and  $i + 1$  (Problem 3.3). Physically minded readers may picture a logical operator as a string that can be deformed by multiplying by checks, but cannot be removed; the following theorem is the precise content of that picture.

**3.5. Lemma.** *For a CSS code, the bilinear map  $(x, z) \mapsto x \cdot z$  on  $\ker H_Z \times \ker H_X$  induces a well-defined pairing*

$$(\ker H_Z / \text{row}(H_X)) \times (\ker H_X / \text{row}(H_Z)) \longrightarrow \mathbb{F}_2.$$

*Proof.* If  $a = H_X^\top y \in \text{row}(H_X)$  and  $z \in \ker H_X$ , then  $a \cdot z = y \cdot H_X z = 0$ ; symmetrically for  $\text{row}(H_Z)$  and  $\ker H_Z$ .  $\square$

**3.6. Theorem.** *The rotated surface patch of odd distance  $d$  is a  $[[d^2, 1, d]]$  code.*

*Proof.* We have  $n = d^2$  and  $k = 1$  by Proposition 3.3, and by the discussion following Theorem 2.10 it suffices to show that  $d_X = d_Z = d$ . Let  $x \in \ker H_Z \setminus \text{row}(H_X)$ . Since  $k = 1$ ,  $x$  lies in the class of  $x^{(0)}$ , so by Lemma 3.5 and Proposition 3.4

$$x \cdot z^{(i)} = x^{(0)} \cdot z^{(i)} = 1 \quad \text{for every } i.$$

Thus  $x$  meets each of the  $d$  pairwise disjoint rows in an odd, hence positive, number of qubits, and  $|x| \geq d$ . A column has weight  $d$ , so  $d_X = d$ . Exchanging rows and columns gives  $d_Z = d$ .  $\square$

By Theorem 2.6, the patch of distance  $d$  corrects every Pauli error of weight at most  $(d - 1)/2$ . The bound is far from the whole story: most errors of much larger weight are also corrected by a good decoder, because a random error of weight comparable to  $d$  is unlikely to resemble a logical operator. Quantifying this requires a noise model, and is the subject of Section 4.6.

**3.4. Strings, boundaries, and homology.** The language of strings has a precise combinatorial counterpart. Consider the graph  $G_X$  whose vertices are the Z checks together with two additional *boundary vertices*, called top and bottom. Each qubit lies in one or two Z checks (Problem 3.4); a qubit in two Z checks is an edge between them, and a qubit in a single Z check (these are exactly the qubits of rows 0 and  $d - 1$ ) is an edge from that check to the top or bottom boundary vertex. A set of qubits  $x$  is then a set of edges, and  $H_Z x$  records the check vertices of odd degree in it. In other words, the syndrome of an X error is the set of *endpoints* of the corresponding edge set, ignoring endpoints on the boundary.

The X checks have a matching description. The four qubits of an X square form a cycle of four edges in  $G_X$  around the square (passing through a boundary vertex if the square touches the top or bottom side), and the two qubits of a top or bottom X check form a cycle of two edges through a boundary vertex.

Thus the X checks are faces of the graph  $G_X$  drawn in the plane, and the binary matrices form a chain complex

$$\mathbb{F}_2^{\{\text{X checks}\}} \xrightarrow{H_X^\top} \mathbb{F}_2^{\{\text{qubits}\}} \xrightarrow{H_Z} \mathbb{F}_2^{\{\text{Z checks}\}},$$

i.e.,  $H_Z H_X^\top = 0$ , which is [Equation \(2.4\)](#) transposed. Its homology at the middle term is exactly  $\ker H_Z / \text{row}(H_X)$ , the space of logical X classes of [Theorem 2.10](#). Geometrically,  $\ker H_Z$  consists of edge sets with no endpoints except on the boundary,  $\text{row}(H_X)$  of boundaries of unions of faces, and the nontrivial class is represented by any path joining the top boundary to the bottom one; this is the homology of the square relative to its top and bottom sides,  $H_1(\text{square}, \text{top} \cup \text{bottom}; \mathbb{F}_2) \cong \mathbb{F}_2$ . The Z errors are described by the dual graph  $G_Z$ , whose vertices are the X checks and whose boundary vertices sit on the left and right; there the nontrivial class is a path from left to right.

We do not use homology to prove anything in these notes; the matrix arguments above are complete. Its value is conceptual: it explains why the number of encoded qubits and the distance are determined by the topology of the surface and the length of its shortest nontrivial cycles, rather than by local details. For instance, on an  $L \times L$  grid on a torus, with qubits on the edges, Z checks on the vertices and X checks on the faces, one obtains Kitaev's *toric code* [[Kit03](#)], which encodes  $\dim H_1(T^2; \mathbb{F}_2) = 2$  qubits with distance  $L$  ([Problem 3.7](#)).

**3.7. Remark.** The literature calls the two kinds of boundary “rough” and “smooth”, but which is which depends on whether one draws the graph  $G_X$  or its dual. To avoid ambiguity we name a boundary by the type of error that can terminate on it without being detected: X errors terminate on the top and bottom boundaries of our patch, and Z errors on the left and right.

**3.5. Notes and open questions.** The construction of [Section 3.4](#) applies to any cellulation of any surface, and to higher-dimensional manifolds, with the parameters governed by homology and by the systoles (lengths of shortest nontrivial cycles) of the cell complex; Breuckmann and Eberhardt [[BE21b](#)] survey these homological codes and their generalizations. In the plane, the bound  $kd^2 = O(n)$  of Bravyi, Poulin and Terhal [[BPT10](#)] ([Section 6](#)) shows that surface codes are optimal up to constants among codes with geometrically local checks, which is the main motivation for the nonlocal codes of [Section 6](#). We have treated surface codes only as memories; computing on encoded qubits (by deforming boundaries, introducing defects, or merging and splitting patches) is a large subject of its own, for which Fowler et al. [[FMMC12](#)] is a starting point.

### 3.6. Problems.

- 3.1.** Complete the proof of [Proposition 3.2](#) by showing that no X boundary check of the rotated patch shares a qubit with a Z boundary check.
- 3.2.** Show that the rotation  $(r, c) \mapsto (c, d - 1 - r)$  maps each X check of the rotated patch onto a Z check and vice versa. (For squares, compute the parity of the new corner; for boundary checks, follow each side of the grid to its image.)
- 3.3.** Show that, for  $0 \leq i \leq d - 2$ , the sum of the indicators of the Z checks contained in rows  $i$  and  $i + 1$  (the Z squares with corner in row  $i$ , and the Z boundary checks between these rows) equals  $z^{(i)} + z^{(i+1)}$ .
- 3.4.** Show that each qubit of the rotated patch lies in exactly one Z check if it is in row 0 or row  $d - 1$ , and in exactly two Z checks otherwise.
- 3.5.** For the nine-qubit patch, compute the Z-check syndrome of each single-qubit X error and the X-check syndrome of each single-qubit Z error. Which pairs of single-qubit errors share a syndrome, and by which check do they differ?
- 3.6.** On the nine-qubit patch, show that  $Y_1 X_4 X_7 Z_2 Z_3$  is a logical operator, identify it (up to phase) as a product of  $\bar{X}$ ,  $\bar{Z}$ , and stabilizers, and compute its weight.

- 3.7. For the toric code on an  $L \times L$  torus, show that the product of all Z checks is the identity, as is the product of all X checks, and that these are the only relations among the checks. Deduce that the code encodes two qubits.

#### 4. DECODING SURFACE CODES

A decoder turns an observed syndrome into a correction. By [Theorem 2.10](#), the checks determine the error only up to an affine space that meets several logical classes, so a decoder must guess, and a principled guess requires a probability model for the errors. This section formulates the decoding problem as statistical inference, shows that for the surface code under independent noise the most likely error can be found by minimum-weight perfect matching, and discusses the asymptotic behaviour of the resulting failure probability.

**4.1. Noise models and decoders.** Throughout this section the checks are measured perfectly and the errors are Pauli operators drawn from a declared distribution; this is the *code-capacity* setting, the simplest of the three settings distinguished in [Section 8.1](#). For a CSS code we further restrict to X errors, since Z errors are handled identically with the roles of the two check matrices exchanged ([Remark 2.11](#) discusses what this restriction ignores).

**4.1. Definition.** In the *independent bit-flip model* with parameters  $p_1, \dots, p_n \in (0, 1/2)$ , the error is  $X(e)$  for a random vector  $e \in \mathbb{F}_2^n$  with independent coordinates and  $\mathbb{P}(e_j = 1) = p_j$ . When all  $p_j$  equal a common value  $p$  we speak of the bit-flip model with parameter  $p$ .

**4.2. Definition.** A *decoder* for the X errors of a CSS code is a function  $\hat{e} : \mathbb{F}_2^{r_Z} \rightarrow \mathbb{F}_2^n$ , where  $r_Z$  is the number of rows of  $H_Z$ , such that  $H_Z \hat{e}(s) = s$  for every syndrome  $s$  in the image of  $H_Z$ . The decoder *fails* on an error  $e$  if  $e + \hat{e}(H_Z e) \notin \text{row}(H_X)$ . Under a noise model, the *logical failure probability* of the decoder is the probability of this event.

By [Theorem 2.10](#), failure means exactly that the residual operator after correction is a logical operator. For the rotated patch,  $e + \hat{e}$  lies in  $\ker H_Z$ , so by [Lemma 3.5](#) and [Proposition 3.4](#) failure is detected by a single parity:

$$(4.1) \quad \text{the decoder fails on } e \iff (e + \hat{e}(H_Z e)) \cdot z^{(0)} = 1,$$

i.e., the residual crosses the top row an odd number of times. A simulation can therefore decide failure without enumerating the stabilizer group.

**4.2. Three decoding objectives.** Under the independent bit-flip model, Bayes' rule gives the posterior distribution of the error given the syndrome  $s$ :

$$(4.2) \quad \mathbb{P}(e \mid s) = \frac{1}{\mathbb{P}(H_Z e = s)} \prod_{j=1}^n p_j^{e_j} (1 - p_j)^{1-e_j} \mathbf{1}\{H_Z e = s\}.$$

A decoding rule can sensibly aim at any of the following.

- (1) *Most likely error:* choose  $\hat{e}(s)$  maximizing  $\mathbb{P}(e \mid s)$ .
- (2) *Bitwise marginals:* compute  $\mathbb{P}(e_j = 1 \mid s)$  for each  $j$  and set  $\hat{e}_j = 1$  when this exceeds  $1/2$ .
- (3) *Most likely class:* fix any  $e_0$  with  $H_Z e_0 = s$ , and choose the coset  $e_0 + \ell + \text{row}(H_X)$ , with  $\ell$  ranging over representatives of  $\ker H_Z / \text{row}(H_X)$ , that maximizes

$$(4.3) \quad \sum_{a \in \text{row}(H_X)} \mathbb{P}(e_0 + \ell + a \mid s);$$

output any element of that coset.

**4.3. Proposition.** *Among all decoders, those of the third kind, and only those, minimize the logical failure probability.*

*Proof.* Conditional on the syndrome  $s$ , a decoder succeeds exactly when  $e$  lies in the coset of  $\text{row}(H_X)$  containing  $\hat{e}(s)$ , an event of conditional probability Equation (4.3) for that coset. Maximizing the success probability for each  $s$  separately maximizes it overall.  $\square$

The three objectives genuinely differ. The second does not even guarantee  $H_Z \hat{e} = s$ , as we shall see in Example 5.3; the first ignores the fact that the many representatives of one class add their probabilities. The following example, too small to protect a qubit but large enough to be instructive, separates the first and third objectives.

**4.4. Example.** Let  $H_X = (1 \ 1 \ 0)$  and  $H_Z = (1 \ 1 \ 1)$ , so that Equation (2.4) holds and  $k = 3 - 1 - 1 = 1$  (the distance is one). For the syndrome  $s = 1$ , the solutions of  $H_Z e = 1$  split into the two cosets  $\{100, 010\}$  and  $\{001, 111\}$  of  $\text{row}(H_X) = \{000, 110\}$ ; here strings list coordinates in order, so that 100 is an error on the first qubit. With  $(p_1, p_2, p_3) = (0.2, 0.2, 0.3)$  the unnormalized posterior masses are:

| Class          | Masses of its elements | Class mass |
|----------------|------------------------|------------|
| $\{100, 010\}$ | 0.112, 0.112           | 0.224      |
| $\{001, 111\}$ | 0.192, 0.012           | 0.204      |

The most likely error is 001, but the most likely class is the other one; after normalization by 0.428 the class probabilities are approximately 0.523 and 0.477. The decoder that outputs the most likely error thus fails with conditional probability 0.523, while the optimal decoder fails with probability 0.477.

Note that the class masses are sums of probabilities of disjoint events, not sums of amplitudes: once the noise model is declared, decoding is entirely classical.

**4.3. Correction on the nine-qubit patch.** Suppose the actual error on the patch of Figure 1 is  $X_1$ , i.e.,  $e = \{1\}$  (we identify a vector with its set of nonzero coordinates). Its syndrome  $H_Z e$  is  $(1, 0, 0, 0)$ , and each of the following corrections has the same syndrome:

| Correction $\hat{e}$ | Residual $e + \hat{e}$ | Effect on the code             |
|----------------------|------------------------|--------------------------------|
| $\{1\}$              | $\emptyset$            | identity                       |
| $\{2\}$              | $\{1, 2\}$             | the stabilizer $A_1$           |
| $\{4, 7\}$           | $\{1, 4, 7\}$          | the logical operator $\bar{X}$ |

The second correction succeeds although it does not identify the actual error; the third restores every check outcome and nevertheless applies a logical error. After the correction, remeasuring the checks cannot distinguish these cases, since all three residuals have zero syndrome. Which correction to prefer depends on the probabilities of the errors  $\{1\}$ ,  $\{2\}$  and  $\{4, 7\}$ , and the next subsection makes that comparison systematic.

**4.4. Minimum-weight matching.** The most likely error is the solution of a shortest-path problem in disguise. Taking logarithms in Equation (4.2), maximizing  $\mathbb{P}(e \mid s)$  over  $\{e : H_Z e = s\}$  is equivalent to minimizing

$$(4.4) \quad \sum_j w_j e_j, \quad w_j := \log \frac{1 - p_j}{p_j} > 0.$$

For equal priors this is the minimum Hamming weight.

Recall the graph  $G_X$  of Section 3.4: its vertices are the  $Z$  checks and two boundary vertices, and each qubit  $j$  is an edge of weight  $w_j$ . Figure 3 draws it for the nine-qubit patch. An error is a set of edges, and its syndrome is the set of check vertices of odd degree. The most-likely-error problem is therefore: find a minimum-weight edge set whose odd-degree check vertices are exactly the given set  $D$  of *defects*, with no constraint at the boundary vertices. In combinatorial optimization this is a minimum-weight  $T$ -join problem, with the twist that the boundary vertices may have any parity.

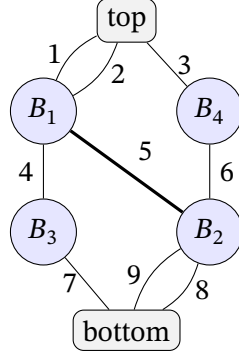


FIGURE 3. The graph  $G_X$  of the nine-qubit patch, redrawn abstractly. Edges are labeled by qubit; parallel edges are distinct qubits with the same endpoints. The four check vertices carry the syndrome, and the top and bottom boundary vertices carry no constraint.

**4.5. Proposition.** Let  $\delta(u, v)$  be the weighted shortest-path distance in  $G_X$ , and let  $\delta(u, \partial)$  be the distance from  $u$  to the nearer boundary vertex. The minimum weight of an edge set with odd-degree check vertices exactly  $D$  equals the minimum, over partitions of  $D$  into pairs  $\{u, v\}$  and singletons  $\{u\}$ , of

$$\sum_{\text{pairs}} \delta(u, v) + \sum_{\text{singletons}} \delta(u, \partial),$$

and a minimizing edge set is obtained as the symmetric difference of the corresponding shortest paths.

*Proof.* The symmetric difference of paths joining the pairs, and joining the singletons to the boundary, has odd degree exactly at  $D$  among check vertices (each defect is an endpoint of one path, and interior vertices of paths have even degree), and its weight is at most the displayed sum. Conversely, let  $F$  be any edge set with odd check vertices  $D$ , and let  $T \supseteq D$  be the set of all vertices of odd degree in  $F$ , boundary vertices included. The edges of  $F$  contain  $|T|/2$  edge-disjoint paths whose endpoints pair up the vertices of  $T$  (by induction on  $|T|$ : every connected component of  $F$  has an even number of odd vertices, so a component containing one vertex of  $T$  contains another; remove a path joining them and repeat). Discard the paths joining two boundary vertices. Each remaining path joins two defects  $u, v$ , and has weight at least  $\delta(u, v)$ , or joins a defect  $u$  to a boundary vertex, and has weight at least  $\delta(u, \partial)$ . Since the weights are positive, the weight of  $F$  is at least the sum for the induced partition.  $\square$

The minimization over partitions is a minimum-weight perfect matching problem on the complete graph whose vertices are the defects together with one boundary copy for each defect, with the edge weights  $\delta(u, v)$  between defects,  $\delta(u, \partial)$  between a defect and its own copy, and zero between boundary copies. Edmonds' blossom algorithm [Edm65] solves such problems in polynomial time. Dennis et al. [DKLP02] proposed this decoder for the surface code, and it remains the standard baseline; PyMatching [Hig22, HG25] is a fast open-source implementation that works directly on the graph rather than on the complete graph of defects. Note that a single boundary vertex with a parity constraint would be wrong: the number of defects joined to the boundary is not determined by the syndrome.

The reduction depends on the fact that every column of  $H_Z$  has at most two nonzero entries, i.e., that each error component flips at most two checks. This property —we shall call such a decoding problem *graphlike*— fails for the Y errors of Remark 2.11 if both sectors are decoded together, for the sparse codes of Section 6, and for many circuit faults (Section 8).

**4.5. A weighted example and its limits.** On the nine-qubit patch, consider the syndrome  $(1, 1, 0, 0)$ , i.e., defects at  $B_1$  and  $B_2$ . The one-edge set  $\{5\}$  and the two-edge set  $\{1, 8\}$  (which joins  $B_1$  to the top and  $B_2$  to the bottom) both have this syndrome. Take  $p_1 = p_8 = 0.1$ ,  $p_5 = 0.01$ , and  $p_j = 0.001$  otherwise.

Then

$$w_5 = \log 99 \simeq 4.595, \quad w_1 + w_8 = 2 \log 9 \simeq 4.394,$$

so the two-edge set is the more likely error despite its larger weight; exhaustive enumeration confirms that it is the most likely error with this syndrome. If the actual error was  $\{5\}$ , the correction  $\{1, 8\}$  leaves the residual  $\{1, 5, 8\}$ , which meets the top row in one qubit and is therefore a logical error by Equation (4.1). An optimal decision can be wrong on an individual instance; optimality is a statement about probabilities.

More importantly, matching optimizes the probability of an individual error, which by Example 4.4 is not the same as optimizing the probability of a logical class. For the surface code under independent noise, the loss is small in practice, but it is not zero, and decoders that approximate the most likely class do exist [DKLP02].

**4.6. Thresholds.** Let  $P_L(d, p)$  denote the logical failure probability of the matching decoder on the rotated patch of distance  $d$  under the bit-flip model with parameter  $p$ . For each fixed  $d$ ,  $P_L(d, p)$  is a polynomial in  $p$  whose lowest-order term has degree exactly  $(d + 1)/2$ : no error of smaller weight causes failure, by Theorem 2.6, while some error of weight  $(d + 1)/2$  does (Problem 4.5); for fixed small  $p$ , larger codes are therefore better. For larger  $p$  the opposite holds, since a longer patch offers more opportunities for a logical string to form. The transition between these regimes is sharp: Dennis et al. [DKLP02] proved that there is a *threshold*  $p_{\text{th}} > 0$  such that  $P_L(d, p) \rightarrow 0$  as  $d \rightarrow \infty$  whenever  $p < p_{\text{th}}$ , and related the optimal decoder's threshold to a phase transition in a random-bond Ising model. The numerical value is approximately 10.3% for matching [WHP03] and approximately 11% for the optimal decoder [DKLP02]. Below threshold the failure probability decays exponentially in  $d$ , roughly like  $(p/p_{\text{th}})^{(d+1)/2}$  up to factors that depend on  $d$  more slowly; we state this only heuristically.

Figure 4 shows the phenomenon in a small simulation. For each  $(d, p)$  we sampled independent bit-flip errors, decoded them with PyMatching using uniform weights, and decided failure by Equation (4.1). Each point is an estimate from finitely many independent samples, and the error bars quantify only that

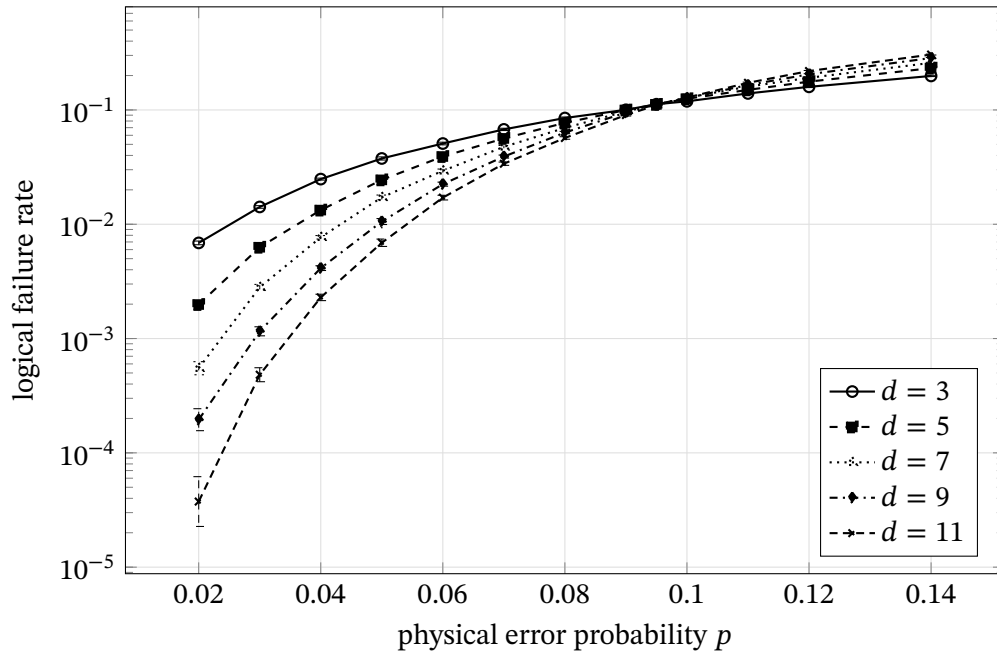


FIGURE 4. Estimated logical failure probability of matching on the rotated patch under the bit-flip model, for several distances. Each point uses  $10^5$  samples ( $4 \times 10^5$  for  $p \leq 0.04$ ); error bars are 95% Wilson intervals.

sampling uncertainty. If  $F$  failures are observed in  $N$  samples, the 95% *Wilson interval* for the failure probability is

$$(4.5) \quad \frac{F + z^2/2}{N + z^2} \pm \frac{z}{N + z^2} \sqrt{\frac{F(N - F)}{N} + \frac{z^2}{4}}, \quad z \simeq 1.96,$$

the set of probabilities not rejected by the usual two-sided normal-approximation test at level 0.05 [Wil27]. Agresti and Coull [AC98] explain why it is preferable to both the naive interval and the “exact” (Clopper–Pearson) interval. Unlike the naive interval  $\hat{p} \pm z\sqrt{\hat{p}(1 - \hat{p})/N}$ , it does not collapse to a point when  $F = 0$ .

In our data the curves cross near  $p \simeq 0.095$ , where all five distances fail with probability about 0.11; below the crossing the order of the curves is reversed, and at  $p = 0.02$  the estimated failure probability falls from about  $7 \times 10^{-3}$  for  $d = 3$  to about  $4 \times 10^{-5}$  for  $d = 11$ . The crossing lies somewhat below the asymptotic value 10.3%, which is typical of small distances (the crossing points of successive pairs of curves drift as  $d$  grows), and a careful estimate of the threshold extrapolates in  $d$  rather than reading off a single crossing. In any case, such a crossing estimates the threshold for this decoder, this code family, and this noise model only. Under the circuit-level models of Section 8 the thresholds of the same codes are smaller by roughly an order of magnitude, and a statement of the form “the threshold is  $p$ ” is meaningless until all three ingredients have been specified.

**4.7. Notes and open questions.** Dennis et al. [DKLP02] prove the existence of a threshold by counting the chains that could cause a logical failure, and relate optimal decoding of the surface code to a two-dimensional random-bond Ising model along its Nishimori line; the value of about 11% for the optimal threshold comes from numerical study of that model, not from a closed-form computation, and we are not aware of a proof of its exact value. Most-likely-class decoding, which matching only approximates, can itself be approximated efficiently for surface codes with tensor-network contractions [BSV14]; this provides a benchmark for measuring how much is lost by optimizing over errors rather than classes (Example 4.4). Below threshold, the exponent  $(d + 1)/2$  in the heuristic  $P_L \approx (p/p_{\text{th}})^{(d+1)/2}$  is forced by Problem 4.5, but the prefactor counts the minimum-weight failing errors and grows with  $d$ ; precise finite-size laws for  $P_L(d, p)$  are a matter of numerical study [WHP03], and the extrapolation from small simulated distances to the large distances of interest is a recurring source of uncertainty.

#### 4.8. Problems.

- 4.1. Verify the table of corrections for the error  $X_1$  on the nine-qubit patch. For the bit-flip model with parameter  $p$ , compute the posterior probabilities of the errors  $\{1\}$ ,  $\{2\}$  and  $\{4, 7\}$  given the syndrome  $(1, 0, 0, 0)$ , up to a common factor.
- 4.2. In the weighted example, find the prior probabilities for which  $\{5\}$  and  $\{1, 8\}$  are equally likely. What does the matching decoder output when all priors are equal?
- 4.3. In Example 4.4, find all priors  $(p, p, q)$  for which the most likely error and the most likely class disagree.
- 4.4. For the nine-qubit patch with uniform weights, list a minimum-weight correction for each of the sixteen Z-check syndromes. Which syndromes have several minimum-weight corrections, and do these always lie in the same logical class?
- 4.5. Let  $d \geq 3$  be odd, and let  $e$  be the X error on the lowest  $(d + 1)/2$  qubits of a column of the rotated patch. Show that  $e$  has a single defect, that in  $G_X$  (with unit weights) this defect is at distance  $(d - 1)/2$  from the top boundary vertex and  $(d + 1)/2$  from the bottom one, and deduce that matching with uniform weights fails on  $e$ , however ties are broken. Then show that for the patch of distance three,  $P_L(3, p) = cp^2 + O(p^3)$  for the matching decoder with uniform weights, compute  $c$ , and show that it does not depend on how ties between minimum-weight corrections are broken.
- 4.6. Derive Equation (4.5) by solving  $|F/N - q| \leq z\sqrt{q(1 - q)/N}$  for  $q$ .

## 5. BELIEF PROPAGATION

Matching solves the most-likely-error problem exactly, but only for graphlike problems. For a general sparse check matrix, the posterior Equation (4.2) is a distribution on the solutions of a linear system over  $\mathbb{F}_2$ , and computing its marginals or its mode is intractable in general (Section 7 gives references). *Belief propagation* (BP) is a message-passing algorithm that computes the marginals exactly when the check matrix has a tree-like structure and serves as a heuristic otherwise. It originated in the decoding of classical low-density parity-check codes, and it is the starting point of most decoders for the quantum codes of Section 6. This section derives BP from conditional probability and exhibits, on exactly solvable examples, its correctness on trees, a failure that no exact marginal computation can avoid, a failure of its dynamics on a graph with cycles, and the obstruction created by degeneracy.

Throughout,  $H$  is an  $m \times n$  binary matrix,  $s \in \mathbb{F}_2^m$  an observed syndrome, and  $e$  a random vector with independent coordinates  $\mathbb{P}(e_j = 1) = p_j$ ; we study  $\mathbb{P}(e \mid He = s)$ , which is Equation (4.2) with  $H$  in place of  $H_Z$ .

## 5.1. Tanner graphs and sum-product messages.

**5.1. Definition.** The *Tanner graph* of  $H$  is the bipartite graph with a *variable node* for each column  $j$ , a *check node* for each row  $a$ , and an edge between  $j$  and  $a$  exactly when  $H_{aj} = 1$ . We write  $N(a)$  and  $N(j)$  for the sets of neighbours.

The posterior factors over the Tanner graph: it is proportional to the product of one prior factor  $q_j(e_j)$  per variable node, where  $q_j(0) = 1 - p_j$  and  $q_j(1) = p_j$ , and one indicator factor  $\mathbf{1}\{\sum_{j \in N(a)} e_j = s_a\}$  per check node. *Sum-product* BP attaches to each directed edge a message, i.e., a probability vector on  $\{0, 1\}$ , and updates the messages by

$$(5.1) \quad m_{j \rightarrow a}(b) \propto q_j(b) \prod_{c \in N(j) \setminus \{a\}} m_{c \rightarrow j}(b),$$

$$(5.2) \quad m_{a \rightarrow j}(b) \propto \sum_{\substack{(b_i)_{i \in N(a) \setminus \{j\}} \\ b + \sum_i b_i = s_a}} \prod_{i \in N(a) \setminus \{j\}} m_{i \rightarrow a}(b_i),$$

starting from  $m_{j \rightarrow a} = q_j$  and normalizing each message to sum to one. The *belief* of variable  $j$  is

$$(5.3) \quad \beta_j(b) \propto q_j(b) \prod_{a \in N(j)} m_{a \rightarrow j}(b).$$

In words, a check tells a variable how likely each of its values is to satisfy the parity, given what the other variables of that check have reported; a variable tells a check what it has learned from its prior and from its other checks. The exclusion of the recipient in each product prevents a node from receiving its own information back immediately.

**5.2. Proposition.** *If the Tanner graph is a tree, then after a number of parallel update rounds equal to its diameter the beliefs Equation (5.3) equal the exact posterior marginals  $\mathbb{P}(e_j = b \mid He = s)$ .*

*Sketch.* Removing the edge  $\{j, a\}$  splits the tree into two components and the factors of the posterior accordingly. By induction on the size of the component containing  $a$ , the message  $m_{a \rightarrow j}$  (once it has stabilized) is proportional to the sum, over all assignments of the variables in that component, of the product of the factors in it, viewed as a function of  $e_j$ ; Equations (5.1) and (5.2) are exactly the recursive form of these partial sums, by the distributive law. Multiplying the contributions of all components around  $j$  gives the unnormalized marginal.  $\square$

On a graph with cycles the components do not separate, the induction fails, and the iteration may oscillate or converge to beliefs that are not the marginals. BP is then a heuristic, often a remarkably good one, with no general guarantee.

**5.2. The binary check update.** For binary variables the check update has a closed form. If  $B_i$  are independent with  $u_i = \mathbb{P}(B_i = 1)$ , then

$$\mathbb{E}(-1)^{\sum_i B_i} = \prod_i (1 - 2u_i),$$

and the probability that  $\sum_i B_i$  has a prescribed parity follows. Writing  $u_i = m_{i \rightarrow a}(1)$ , this gives

$$(5.4) \quad m_{a \rightarrow j}(1) = \frac{1 - (-1)^{s_a} \prod_{i \in N(a) \setminus \{j\}} (1 - 2u_i)}{2},$$

i.e., the probability that the other variables have the parity required when  $e_j = 1$ . In terms of log-likelihood ratios  $L = \log(m(0)/m(1))$ , the updates become

$$L_{j \rightarrow a} = \log \frac{1 - p_j}{p_j} + \sum_{c \in N(j) \setminus \{a\}} L_{c \rightarrow j},$$

$$L_{a \rightarrow j} = 2 \operatorname{artanh} \left( (-1)^{s_a} \prod_{i \in N(a) \setminus \{j\}} \tanh \frac{L_{i \rightarrow a}}{2} \right),$$

since  $1 - 2u = \tanh(L/2)$ . A positive LLR favours the value 0. The *min-sum* variant replaces the second rule by  $(-1)^{s_a} \prod_i \operatorname{sgn} L_{i \rightarrow a} \cdot \min_i |L_{i \rightarrow a}|$ , usually with a scaling factor; it is cheaper and is an approximation of sum-product, not an equivalent form of it. Floating-point implementations clip the arguments of  $\operatorname{artanh}$ , since a check whose other inputs are certain sends an infinite LLR.

### 5.3. Exact marginals can still give an impossible answer.

**5.3. Example.** Let  $H = (1 \ 1 \ 1)$ ,  $s = 1$ , and  $p_1 = p_2 = p_3 = 0.1$ . The Tanner graph is a star, hence a tree, and BP is exact. The solutions are 100, 010, 001, 111, with unnormalized masses 0.081, 0.081, 0.081, 0.001, so

$$\mathbb{P}(e_j = 1 \mid s = 1) = \frac{0.081 + 0.001}{0.244} = \frac{41}{122} \simeq 0.336$$

for each  $j$ . BP reproduces this in one round: Equation (5.4) gives  $m_{a \rightarrow j}(1) = (1 + 0.8^2)/2 = 0.82$ , and Equation (5.3) gives  $0.1 \cdot 0.82 / (0.1 \cdot 0.82 + 0.9 \cdot 0.18) = 41/122$ . Every marginal favours 0, so the bitwise decision is 000, which does not satisfy the check.

The failure has nothing to do with cycles or with quantum mechanics: the three most likely solutions are symmetric, the marginals average over them, and rounding each marginal separately destroys the parity constraint. This is the second objective of Section 4.2 failing in its simplest form. It shows that a BP decoder needs a final step that enforces  $H\hat{e} = s$ , which is the role of the post-processing of Section 7.

**5.4. Example.** Let

$$H = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \quad s = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad (p_1, p_2, p_3) = (0.1, 0.2, 0.3).$$

The Tanner graph is a path, hence a tree. The solutions are 100 and 011, with masses 0.056 and 0.054, so their posterior probabilities are 28/55 and 27/55, and the marginals are (28/55, 27/55, 27/55). BP reaches these values after two rounds; here the bitwise decision 100 is consistent.

**5.4. Schedules, damping, and stopping.** On a tree, the order of the updates and the number of rounds beyond the diameter are immaterial. On a graph with cycles, BP is an iterative map whose behaviour depends on three choices that every implementation must make, and that must be reported with any result obtained with it.

**5.5. Definition.** (1) The *hard decision* after a round is the vector  $\hat{e}$  with  $\hat{e}_j = 1$  exactly when  $\beta_j(1) > \beta_j(0)$ . The *stopping rule* halts the iteration as soon as  $H\hat{e} = s$ , and declares failure after a prescribed maximum number of rounds.

- (2) In the *flooding* (or parallel) schedule, each round recomputes all check-to-variable messages from the current variable-to-check messages, and then all variable-to-check messages from the new check-to-variable messages. In a *serial* schedule, the checks are visited in a fixed order; at each check  $a$ , the messages  $m_{j \rightarrow a}$ ,  $j \in N(a)$ , are first recomputed from the current check-to-variable messages, and then the messages  $m_{a \rightarrow j}$  are recomputed. One sweep through all checks is a round.
- (3) With *damping factor*  $\gamma \in [0, 1)$ , each newly computed variable-to-check LLR  $L^{\text{new}}$  is replaced by  $(1 - \gamma)L^{\text{new}} + \gamma L^{\text{old}}$ .

Conventions vary (damping is sometimes applied to the check-to-variable messages, and serial schedules may visit variables rather than checks), which is one reason why results obtained with “BP” by different authors need not agree. Neither damping nor a serial schedule changes the fixed points of the iteration; they change only its dynamics. For classical LDPC codes, Sharon, Litsyn and Goldberger [SLG07] show that serial schedules typically need about half as many rounds as flooding. With the stopping rule, the output of BP satisfies the syndrome whenever the iteration halts before the maximum; the number of rounds is then a random variable, and its whole distribution, not only its mean, determines the cost of decoding.

The following example, which involves neither degeneracy nor quantum mechanics, shows how flooding alone can prevent BP from ever stopping.

**5.6. Example.** Let

$$H = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad s = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix},$$

with checks  $a, b, c$  in this order and equal priors  $p_j = p < 1/2$ . The matrix is invertible, so 001 is the only solution of  $He = s$ , and any reasonable decoder should find it. We run min-sum (without scaling) with the flooding schedule. The min-sum updates are positively homogeneous of degree one in the LLRs, so all messages are integer multiples of  $\lambda := \log((1 - p)/p)$ , and we write them in units of  $\lambda$ ; in particular, the behaviour below does not depend on  $p$ .

**5.7. Proposition.** In *Example 5.6*, for every  $k \geq 1$ :

- (1) after round  $2k - 1$ , the message from  $c$  to  $e_3$  is  $-k$ , every other check-to-variable message is  $-1$ , and the hard decision is 111;
- (2) after round  $2k$ , the messages from  $a$  to  $e_2$  and from  $b$  to  $e_1$  are  $k$ , every other check-to-variable message is 0, and the hard decision is 000.

*In particular, the stopping rule never halts the iteration, although the beliefs grow without bound.*

*Proof.* Recall that a min-sum check message has absolute value equal to the least absolute value among the other incoming messages, so it vanishes if one of them does. In round 1 all variable-to-check messages equal 1 and  $s = 111$ , so every check-to-variable message is  $-1$ , which is (1) for  $k = 1$ . Suppose (1) holds after round  $2k - 1$ . The new variable-to-check messages are  $e_1 \rightarrow b$ ,  $e_1 \rightarrow c$ ,  $e_2 \rightarrow a$ ,  $e_2 \rightarrow c$  all equal to  $1 - 1 = 0$ ,  $e_3 \rightarrow a$  and  $e_3 \rightarrow b$  equal to  $1 - 1 - k = -k$ , and  $e_3 \rightarrow c$  equal to  $1 - 1 - 1 = -1$ . In round  $2k$ , therefore, every check message with an input from  $e_1$  or  $e_2$  vanishes, while  $a \rightarrow e_2$  and  $b \rightarrow e_1$ , whose only input is  $-k$  from  $e_3$ , equal  $-\text{sgn}(-k)k = k$ ; this is (2). The new variable-to-check messages are now  $e_1 \rightarrow b = e_2 \rightarrow a = 1$ ,  $e_1 \rightarrow c = e_2 \rightarrow c = 1 + k$ , and  $e_3 \rightarrow a = e_3 \rightarrow b = e_3 \rightarrow c = 1$ , so in round  $2k + 1$  every check message equals  $-1$  except  $c \rightarrow e_3 = -\min(1 + k, 1 + k) = -(k + 1)$ ; this is (1) for  $k + 1$ . The beliefs are  $(-1, -1, -(k + 1))$  after round  $2k - 1$  and  $(1 + k, 1 + k, 1)$  after round  $2k$ , whence the hard decisions.  $\square$

The mechanism is easy to describe. In the first round every check is unsatisfied and asks each of its variables to flip; the three variables obey simultaneously, and the hard decision 111 overshoots. In the next round the checks report the overshoot, and the variables flip back. A serial schedule breaks the

simultaneity: one sweep in the order  $a, b, c$  already produces 001, because check  $b$  hears from check  $a$ , through  $e_3$ , before it speaks. Damping slows the response instead: with  $\gamma = 1/2$ , flooding produces 001 in the second round (Problem 5.6). Sum-product oscillates on this example in its first rounds as well, but escapes after a number of rounds that grows as  $p$  decreases (six rounds for  $p = 0.01$ , in our computation); this is one concrete sense in which min-sum is an approximation of sum-product and not an equivalent form of it.

Oscillation, slow convergence, and convergence to beliefs that are not the marginals are the generic failure modes of BP on graphs with cycles; Murphy, Weiss and Jordan [MWJ99] give an early empirical study outside coding theory. For quantum codes, Poulin and Chung [PC08] identify a further obstruction, which no choice of schedule removes; it is the subject of the next subsection.

**5.5. Why quantum codes are harder.** Two features of quantum CSS codes work against BP. First, the commutation condition Equation (2.4) says that every row of  $H_X$  lies in  $\ker H_Z$ , and in a qLDPC code these rows have small weight; low-weight elements of the kernel force short cycles in the Tanner graph of  $H_Z$ .

**5.8. Lemma.** *Let  $a \in \ker H$  be nonzero, and suppose that every coordinate in the support of  $a$  lies in at least two rows of  $H$ . Then the Tanner graph of  $H$  contains a cycle of length at most  $2|a|$ .*

*Proof.* Consider the subgraph spanned by the variable nodes in the support of  $a$  and the check nodes adjacent to them. Each such check meets the support in a nonzero even number of positions, because  $Ha = 0$ , and each variable node has degree at least two by hypothesis; a finite graph with all degrees at least two contains a cycle. A cycle in a bipartite graph alternates between its two sides, and at most  $|a|$  variable nodes are available.  $\square$

In the surface code of distance  $d \geq 5$ , for instance, an X square that does not touch the top or bottom side satisfies the hypothesis, and the bound  $2 \cdot 4 = 8$  is attained (Problem 5.4); on the nine-qubit patch no X check does, and the Tanner graph of  $H_Z$  is a tree. Second, and more fundamentally, degeneracy spreads the posterior over many errors in the same class. When two equally likely errors differ by a stabilizer, BP cannot break the symmetry, and its beliefs fail to single out either one, although choosing either would succeed. The smallest instance has three qubits. Let  $H_Z$  have rows 011 and 111, let  $H_X = (0 \ 1 \ 1)$  (so that Equation (2.4) holds, and  $k = 0$ , which suffices for the point at hand), and let  $s = 11$  with  $p_2 = p_3$ . The solutions of  $H_Z e = s$  are 010 and 001, equally likely and differing by the stabilizer  $X_2 X_3$ , so either is a successful correction. But every schedule that treats  $e_2$  and  $e_3$  alike—flooding, serial, damped or not—gives them equal beliefs, so their hard decisions agree and the first check is never satisfied (Problem 5.7). Poulin and Chung [PC08] analyze both difficulties and propose modifications, and Raveendran and Vasić [RV21] study the small subgraphs (*trapping sets*) on which iterative decoders of qLDPC codes fail, including symmetric configurations forced by degeneracy. Example 5.3 isolates a simpler mechanism of the same kind (a symmetry among solutions) without any degeneracy, and Example 5.6 a failure of the dynamics without any symmetry; neither should be mistaken for a demonstration of the quantum effect.

The remedies used in practice fall into a few families: modifying the iteration (damping, serial schedules, memory terms, adjusted priors), breaking symmetries by fixing or removing some variables and rerunning BP (decimation, inactivation), or passing BP’s reliabilities to an algebraic procedure that enforces the syndrome exactly. They are the subject of Section 7, after we describe in Section 6 the codes on which they are used.

**5.6. Notes and open questions.** The origins of belief propagation are in coding theory, with Gallager’s decoding of LDPC codes [Gal62], and in artificial intelligence, with Pearl’s work on inference in Bayesian networks [Pea88]; the factor-graph formulation of Kschischang, Frey and Loeliger [KFL01] unifies the two, and MacKay [Mac03] and Richardson and Urbanke [RU08] are standard textbook references. For classical LDPC codes, *density evolution* predicts the performance of BP on long random codes, whose Tanner graphs look like trees up to any fixed radius [RU08]. Lemma 5.8 shows that the Tanner graphs of

qLDPC codes cannot look like trees beyond a radius comparable to the weight of the stabilizers, and degeneracy produces exact symmetries of the posterior; we are not aware of a theory that predicts the performance of BP-based decoders on qLDPC codes with an accuracy comparable to density evolution. Natural questions include when BP converges on Tanner graphs with the cycles forced by commutation, how its fixed points on such graphs relate to the logical classes, and which modifications of the iteration provably break the symmetries of degenerate posteriors.

### 5.7. Problems.

- 5.1. In [Example 5.3](#), compute the exact marginal of  $e_1$  by enumeration and by one BP check message, and confirm that they agree.
- 5.2. Carry out the BP iteration for [Example 5.4](#) explicitly, starting from the priors, and verify that the beliefs after two parallel rounds are the exact marginals.
- 5.3. Prove the identity  $\mathbb{E}(-1)^{\sum_i B_i} = \prod_i (1 - 2u_i)$  and deduce [Equation \(5.4\)](#).
- 5.4. Show that the Tanner graph of  $H_Z$  for the nine-qubit patch is a tree, so that BP computes the exact marginals of the X sector there, while for the patch of distance five it contains cycles of length eight but none of length four.
- 5.5. On the nine-qubit patch with all priors equal to  $p$ , consider the syndrome  $(1, 0, 0, 0)$ , produced by  $X_1$  or by  $X_2$ . Show that the exact marginals of  $e_1$  and  $e_2$  are equal. What does this say about a decoder that rounds marginals?
- 5.6. In [Example 5.6](#), show that one serial min-sum sweep in the order  $a, b, c$  produces the beliefs  $(1, 1, -1)$  (in units of  $\lambda$ ) and the hard decision 001, and that flooding min-sum with damping factor  $\gamma = 1/2$  produces the hard decisions 111 and then 001.
- 5.7. In the three-qubit example of [Section 5.5](#), show that the transposition of  $e_2$  and  $e_3$  maps the Tanner graph, the priors, and the syndrome to themselves. Deduce that, for any schedule invariant under this transposition, the messages and beliefs of  $e_2$  and  $e_3$  coincide in every round, so that no hard decision satisfies the syndrome.

## 6. QUANTUM LOW-DENSITY PARITY-CHECK CODES

However large the patch, the surface code of [Section 3](#) protects a single qubit. This section describes the class of codes for which the decoders of [Sections 5](#) and [7](#) are designed: CSS codes whose checks are sparse but not geometrically local in the plane. We present two constructions. The hypergraph product of Tillich and Zémor [[TZ14](#)] contains the surface code as a special case and is the homological picture of [Section 3.4](#) applied to arbitrary classical codes; the bivariate bicycle codes of Bravyi et al. [[BCG<sup>+</sup>24](#)] are, at the time of writing, the most intensively studied qLDPC codes in the decoding literature. Breuckmann and Eberhardt [[BE21b](#)] survey the field.

### 6.1. Sparse checks.

**6.1. Definition.** A family of CSS codes is a family of *quantum low-density parity-check* (qLDPC) codes if the numbers of nonzero entries in the rows and in the columns of its check matrices are bounded uniformly over the family.

The rotated surface patches form a qLDPC family, with rows and columns of weight at most four. Their drawback is that they encode a single qubit, whatever their size, and the distance grows only like the square root of the number of qubits. Codes whose checks are local in a planar layout cannot do much better: Bravyi, Poulin and Terhal [[BPT10](#)] showed that such codes satisfy  $kd^2 = O(n)$ , so substantially better parameters require checks that are not geometrically local in the plane. Sparsity alone, however, is compatible with much better parameters: qLDPC families exist with  $k$  and  $d$  both proportional to  $n$  [[PK22a](#)]. The price is that the decoding problem is no longer graphlike.

**6.2. Hypergraph products.** Let  $H_1$  and  $H_2$  be binary matrices of sizes  $m_1 \times n_1$  and  $m_2 \times n_2$ , the check matrices of the classical codes  $\ker H_1$  and  $\ker H_2$ . Write  $k_i = \dim \ker H_i$  and  $k_i^\top = \dim \ker H_i^\top$ , and let  $d_i$  and  $d_i^\top$  be the minimum weights of the nonzero vectors of  $\ker H_i$  and  $\ker H_i^\top$ , with the convention that the minimum over the empty set is  $\infty$ .

**6.2. Definition.** The *hypergraph product* of  $H_1$  and  $H_2$  is the CSS code on  $n = n_1 n_2 + m_1 m_2$  qubits with check matrices

$$(6.1) \quad H_X = [H_1 \otimes I_{n_2} \mid I_{m_1} \otimes H_2^\top], \quad H_Z = [I_{n_1} \otimes H_2 \mid H_1^\top \otimes I_{m_2}].$$

Thus the qubits are indexed by the pairs (variable of  $H_1$ , variable of  $H_2$ ) and (check of  $H_1$ , check of  $H_2$ ), the X checks by the pairs (check, variable), and the Z checks by the pairs (variable, check). By the mixed-product rule  $(A \otimes B)(C \otimes D) = AC \otimes BD$ ,

$$H_X H_Z^\top = H_1 \otimes H_2^\top + H_1^\top \otimes H_2 = 0,$$

so Equation (2.4) holds, and the weight of each row or column of Equation (6.1) is the sum of a row weight and a column weight of  $H_1$  and  $H_2$  (Problem 6.1). In particular, hypergraph products of classical LDPC codes form qLDPC families.

**6.3. Theorem** (Tillich–Zémor [TZ14]). *The hypergraph product of  $H_1$  and  $H_2$  encodes*

$$k = k_1 k_2 + k_1^\top k_2^\top$$

*qubits, and if  $k > 0$  its distance satisfies*

$$d \geq \min(d_1, d_2, d_1^\top, d_2^\top).$$

*If both  $H_1$  and  $H_2$  have full row rank and  $k_1, k_2 > 0$ , then  $d = \min(d_1, d_2)$ .*

*Proof of the formula for  $k$ .* Regard  $A : \mathbb{F}_2^{n_1} \xrightarrow{H_1} \mathbb{F}_2^{m_1}$  and  $B : \mathbb{F}_2^{n_2} \xrightarrow{H_2} \mathbb{F}_2^{m_2}$  as chain complexes concentrated in degrees 1 and 0. Their tensor product has  $(A \otimes B)_1 = \mathbb{F}_2^{n_1 n_2} \oplus \mathbb{F}_2^{m_1 m_2} = \mathbb{F}_2^n$ , and by the Leibniz rule  $\partial(a \otimes b) = \partial a \otimes b + a \otimes \partial b$  (there are no signs in characteristic two) its boundary maps are  $\partial_1 = H_X$  and  $\partial_2 = H_Z^\top$ . Hence  $k = n - \text{rank } H_X - \text{rank } H_Z = \dim \ker \partial_1 - \dim \text{im } \partial_2 = \dim H_1(A \otimes B)$ . By the Künneth formula for complexes of vector spaces,  $H_1(A \otimes B) \cong H_1(A) \otimes H_0(B) \oplus H_0(A) \otimes H_1(B)$ , and  $H_1(A) = \ker H_1$ ,  $H_0(B) = \text{coker } H_2^\top$ ,  $H_0(A) = \text{coker } H_1$ ,  $H_1(B) = \ker H_2^\top$  have dimensions  $k_1, k_2, k_1^\top, k_2^\top$ .  $\square$

For the distance bounds we refer to [TZ14], Theorem 15; the logical operators of minimum weight are, roughly speaking, copies of minimum-weight codewords of one factor placed along a single “row” or “column” of the product. The dimension formula and the distance bounds have been checked on small examples by our companion verification. Equality with the minimum of all four classical distances need not hold when a transpose code is nonzero but its partner in the corresponding homology summand is zero. For instance, adjoining a zero row to  $R_3$  of Example 6.4 gives  $d_1^\top = 1$ , but its product with the full-row-rank matrix  $R_3$  still has distance three; our verifier checks this case as well.

**6.4. Example.** Let  $R_L$  be the  $(L-1) \times L$  check matrix of the repetition code, whose  $i$ th row is  $e_i + e_{i+1}$ . Then  $\ker R_L = \{0, (1, \dots, 1)\}$  and  $R_L$  has full row rank, so  $k_1 = k_2 = 1$ ,  $k_1^\top = k_2^\top = 0$ ,  $d_1 = d_2 = L$ , and  $d_1^\top = d_2^\top = \infty$ . The hypergraph product of  $R_L$  with itself is therefore an  $[[L^2 + (L-1)^2, 1, L]]$  code, with checks of weight at most four; it is the planar surface code of Bravyi and Kitaev [BK98] (Problem 6.2). The rotated patch of Section 3 achieves the same distance with  $L^2$  qubits.

The example explains the name “product”: the Tanner graph of  $R_L$  is a path, the product of two paths is a square grid, and the chain complex of the proof is the cellular chain complex of that grid (compare Section 3.4). If instead  $H_1 = H_2$  is the check matrix of a classical LDPC code with full row rank and with dimension and distance proportional to its length (such codes exist, e.g., by Gallager’s random construction [Gal62]), then  $k$  is proportional to  $n$  and  $d$  to  $\sqrt{n}$ ; this was the first qLDPC family with both

properties. The square-root barrier for the distance was broken only around 2020, by the fiber bundle codes of Hastings, Haah and O'Donnell [HHO21], the lifted products of Panteleev and Kalachev [PK22b], and the balanced products of Breuckmann and Eberhardt [BE21a]; the asymptotically good codes of Panteleev and Kalachev [PK22a] completed this line, with  $k$  and  $d$  both proportional to  $n$ .

**6.3. Bivariate bicycle codes.** A convenient way to satisfy Equation (2.4) is to start from two commuting square matrices  $A, B$  of the same size over  $\mathbb{F}_2$  and set

$$(6.2) \quad H_X = [A \mid B], \quad H_Z = [B^\top \mid A^\top];$$

then  $H_X H_Z^\top = AB + BA = 0$  in characteristic two. For the *bivariate bicycle* codes of Bravyi et al. [BCG<sup>+</sup>24], let  $S_\ell$  be the  $\ell \times \ell$  cyclic shift matrix, put  $x = S_\ell \otimes I_m$  and  $y = I_\ell \otimes S_m$  (two commuting permutation matrices of size  $\ell m$ ), and take  $A$  and  $B$  to be sums of three monomials in  $x, y$ . The code has  $n = 2\ell m$  qubits and checks of weight six. For example,  $\ell = m = 6$  with

$$A = x^3 + y + y^2, \quad B = y^3 + x + x^2$$

gives a code on 72 qubits; a direct computation shows that both check matrices have rank 30, hence  $k = 72 - 60 = 12$ , and Bravyi et al. report the distance 6, so this is a  $[[72, 12, 6]]$  code. With the same  $A$  and  $B$  but  $\ell = 12, m = 6$ , one obtains the  $[[144, 12, 12]]$  code that Bravyi et al. call the *gross code* (a gross being twelve dozen); here both ranks are 66. (The rank computations are part of our companion verification; the distances are quoted, not recomputed.) For comparison, twelve rotated surface patches, each protecting one qubit, use  $12d^2$  data qubits, i.e., 300 for  $d = 5$ , 588 for  $d = 7$ , and 1728 for  $d = 13$  (rotated patches have odd distance). Counting the auxiliary qubits used to measure the checks, Bravyi et al. [BCG<sup>+</sup>24] find that the gross code uses 288 physical qubits in total, and estimate that surface codes would need nearly 3000 to achieve the same suppression of logical errors at a physical error rate of 0.1%; the comparison depends on the measurement circuits of both codes and on the noise model, but it indicates why such codes attract attention.

Each column of  $H_Z$  in Equation (6.2) has three nonzero entries, so a single X error flips three checks: the decoding problem is not graphlike, matching does not apply directly, and BP with post-processing is the natural first choice. The algebraic structure —  $A$  and  $B$  are elements of the group algebra  $\mathbb{F}_2[\mathbb{Z}_\ell \times \mathbb{Z}_m]$ , which is commutative — also makes these codes amenable to the methods of commutative algebra, which compute  $k$  as the dimension of a quotient module; we do not pursue this here.

**6.4. Notes and open questions.** The constructions of this section are the simplest of a large family: lifted and balanced products replace the tensor product of complexes by twisted or quotient versions, and the asymptotically good codes of Panteleev and Kalachev [PK22a] are lifted products; the survey [BE21b] was written shortly before that construction appeared. Several basic questions about the codes actually considered for implementation remain open or only numerically answered. The distances of bivariate bicycle codes are, as far as we know, determined by computer search [BCG<sup>+</sup>24], and we are not aware of a structural description of their minimum-weight logical operators; since computing the minimum distance of a classical linear code is already NP-hard [Var97], such descriptions would have to exploit the algebraic structure. On the decoding side, the existence of thresholds for qLDPC families is known under optimal decoding when the distance grows as a power of  $n$  [KP13], and for hypergraph products of expander codes under a specific combinatorial decoder [FGL18]; we are not aware of comparable guarantees for the BP-based decoders of Sections 5 and 7, which are the ones used in practice.

### 6.5. Problems.

- 6.1.** Show that in Equation (6.1) the row of  $H_X$  indexed by (check  $a$  of  $H_1$ , variable  $j$  of  $H_2$ ) has weight equal to the weight of row  $a$  of  $H_1$  plus the weight of column  $j$  of  $H_2$ , and describe the column weights.
- 6.2.** Place the qubits of the hypergraph product of  $R_3$  with itself at the points  $(2j_1, 2j_2)$  and  $(2a_1 + 1, 2a_2 + 1)$  of a  $5 \times 5$  grid, the X check indexed by  $(a_1, j_2)$  at  $(2a_1 + 1, 2j_2)$ , and the Z check indexed by  $(j_1, a_2)$

- at  $(2j_1, 2a_2 + 1)$ . Show that every check acts on the qubits at the grid points adjacent to it, and that the code is a  $[[13, 1, 3]]$  code.
- 6.3. Verify that  $x$  and  $y$  commute, and that the code Equation (6.2) with  $\ell = m = 6$  has checks of weight six and columns of weight three in each check matrix.
- 6.4. Show that the distance of the rotated patch is  $\sqrt{n}$  and that  $k = 1$  for every  $d$ , so that  $kd^2 = n$  for this family.

## 7. POST-PROCESSING AND OTHER REFINEMENTS OF BELIEF PROPAGATION

Example 5.3 shows that even exact marginals may fail to produce a correction consistent with the syndrome, and Section 5.4 that BP may fail to stop at all. This section describes the most widely used remedy, ordered-statistics decoding, in detail, and then indicates the main lines of current work on decoders for the qLDPC codes of Section 6. Throughout,  $H$ ,  $s$  and the priors are as in Section 5.

**7.1. Ordered-statistics decoding.** The method is due to Fossorier and Lin [FL95], for classical codes; BP followed by OSD (BP+OSD) was adapted to quantum codes by Panteleev and Kalachev [PK21] and studied extensively by Roffe et al. [RWBC20]. In the usual arrangement, BP runs first with the stopping rule of Section 5.4, and OSD is invoked only on the instances where BP fails to produce a consistent hard decision. Its algebraic core is simple: use the soft information produced by BP to decide which coordinates to trust, and solve the check equations exactly for the others.

Let  $H$  have rank  $\rho$  (discard dependent rows first, retaining a consistent syndrome). Choose a set  $B$  of  $\rho$  columns such that the square submatrix  $H_B$  is invertible, and let  $F$  be the complementary set of columns. Every solution of  $He = s$  is then determined by its restriction  $u = e_F$ :

$$(7.1) \quad e_F = u, \quad e_B = H_B^{-1}(s + H_F u).$$

This is an affine parametrization of the  $2^{n-\rho}$  solutions by  $\mathbb{F}_2^{n-\rho}$ ; in practice  $e_B$  is computed by Gaussian elimination, not by forming an inverse.

**7.1. Definition.** Given BP log-likelihood ratios  $L_j$ , OSD-0 orders the columns by increasing reliability  $|L_j|$ , chooses  $B$  greedily as the first  $\rho$  linearly independent columns in this order, sets  $u$  equal to the BP hard decisions on  $F$ , and outputs Equation (7.1).

Thus the least reliable coordinates are recomputed and the most reliable ones are kept. Implementations differ in details. The `ldpc` package used in Section 8.4, for instance, orders the columns by increasing signed LLR (most likely flipped first) and sets the free coordinates to 0 rather than to the hard decisions; the two rules agree on  $u$  whenever every coordinate with negative LLR lands in  $B$ , but they may choose different bases. The output always satisfies the syndrome; it need not be the most likely error, let alone a representative of the most likely class. Note the division of labour: feasibility is guaranteed by the linear algebra, and the quality of the answer depends on the reliability ordering.

For Example 5.3, all reliabilities tie; breaking ties by index gives  $B = \{1\}$ ,  $u = (e_2, e_3) = (0, 0)$ , and the solve gives  $e_1 = 1$ , repairing the inconsistent decision 000. For Example 5.4, taking the basis columns in the order 2, 1 and freezing  $e_3 = 0$  gives  $e_2 = 0$  from the second check and then  $e_1 = 1$ , i.e.,  $e = 100$ . These examples illustrate the algebra only; they say nothing about performance on a code family.

*Higher-order* OSD searches over small changes of  $u$ : for each of a prescribed family of subsets of  $F$ , flip those coordinates of  $u$ , solve Equation (7.1), and keep the candidate of highest prior probability. Enumerating all subsets of size at most  $w$  costs up to  $\sum_{i \leq w} \binom{n-\rho}{i}$  solves, so practical variants restrict the family, for instance to subsets of the least reliable coordinates of  $F$  (the *combination sweep*).

**7.2. A map of BP-based decoders.** The literature on decoders for qLDPC codes is large and changes quickly; we only indicate its main lines, organized by two questions that every such decoder must answer: what guarantees that the output satisfies the syndrome, and how the symmetries created by degeneracy (Section 5.5) are broken.

- *Modified message passing.* Memory effects and adjusted priors change the BP iteration itself. Kuo and Lai [KL22] add memory terms (including inhibitory ones) to the variable updates; Relay-BP [MAB<sup>+</sup>25] chains several such runs with disordered memory strengths, some of them negative, so as to damp oscillations and break symmetries, and keeps the best of the consistent solutions it encounters. These decoders remain parallel message-passing algorithms, attractive for hardware; consistency is reached only when the stopping rule fires.
- *Decimation and inactivation.* Guided decimation [YALH<sup>+</sup>24] repeatedly fixes the most reliable variable and reruns BP, without any linear algebra; a wrongly fixed variable can exclude the correct class. Stabilizer inactivation [dCMS22] removes from the iteration the qubits supporting a check of the other type, i.e., a stabilizer of exactly the kind that creates degeneracy, lets BP converge on the rest, and determines the inactivated qubits by solving a small linear system.
- *Algebraic post-processing.* Besides OSD and its variants, localized statistics decoding [HBQ<sup>+</sup>25] performs the reliability-guided solve of Equation (7.1) only on local regions of the Tanner graph grown around the unsatisfied checks, in parallel, and ambiguity clustering [WB24] divides the problem, after BP, into clusters that can be decoded independently.
- *Matching with corrections.* For problems that are nearly graphlike, such as surface codes under circuit-level noise, matching can be adapted to use the correlations that the graphlike decomposition discards [Fow13].

These methods distribute computational effort differently —BP iterations, Gaussian elimination, repeated BP runs, cluster growth— so counts of iterations or of solves are not, by themselves, comparable measures of cost. Their published comparisons also differ in noise model, failure event, and implementation, which is the subject of Section 8.4.

**7.3. Notes and open questions.** Exact decoding is hard in general: finding a most likely error is NP-hard already for classical linear codes [BMvT78] and for quantum stabilizer codes [HLG11], and optimal decoding in the sense of Proposition 4.3, which takes degeneracy into account, is #P-complete [IP15]. Every practical decoder for qLDPC codes is therefore a heuristic, and its quality is established by simulation on specific code families and noise models. For surface codes, the most likely class can be approximated efficiently with tensor-network methods [BSV14], which gives a benchmark against which matching can be measured; for general qLDPC codes no comparably accurate and tractable reference decoder is known to us, and the gap between practical decoders and optimal decoding is correspondingly hard to quantify. The cost of OSD is dominated by Gaussian elimination, cubic in the size of the matrix in the worst case, which motivates the localized and clustered variants above; whether accuracy comparable to BP+OSD can be achieved within the latency budget of real hardware for the codes of Section 6 is an active question.

#### 7.4. Problems.

- 7.1. For  $H$  and  $s$  as in Example 5.4, solve Equation (7.1) with  $B = \{2, 1\}$  for both values of the free bit  $e_3$ , and identify the more probable solution under the priors of that example.
- 7.2. Show that every solution of  $He = s$  is obtained from Equation (7.1) for exactly one  $u$ , and count the solves performed by an exhaustive higher-order OSD of order  $w$ .
- 7.3. For the symmetric example of Problem 5.7, show that OSD-0 returns a correction consistent with the syndrome whichever way ties among reliabilities are broken, and that the possible outputs differ by  $X_2X_3$ . Why does this not matter if  $X_2X_3$  is a stabilizer?

## 8. FAULTY MEASUREMENTS AND THE EVALUATION OF DECODERS

Until now the checks were measured perfectly, once, after a single realization of the error. In any implementation the checks are measured repeatedly by circuits that are themselves faulty, and a recorded outcome can be wrong. This section explains how the decoding problem changes —it becomes inference

on a larger sparse matrix whose columns are fault mechanisms rather than qubits— and then how the performance of two decoders on such problems should be measured and compared.

**8.1. Three settings.** It is essential to distinguish three levels of modeling.

- (1) *Code capacity*: errors on the data qubits only, with perfect check measurements (Section 4).
- (2) *Phenomenological*: errors on the data qubits between rounds of measurement, and independent errors on the recorded check outcomes, specified directly without reference to a circuit.
- (3) *Circuit level*: faults at specified locations of an explicit measurement circuit (preparations, gates, idle steps, and measurements), propagated through the circuit.

The same symbol  $p$  denotes different things in the three settings, and thresholds or failure rates for one cannot be compared with those for another. In the circuit-level setting a single fault on an auxiliary qubit can spread to several data qubits; the order in which the gates of a measurement circuit are applied (the *schedule*) therefore matters, and it is part of the definition of the experiment, not a detail of its implementation. Even within the circuit-level setting, the noise model is a choice. The *uniform depolarizing* model used in our example below assigns the same parameter  $p$  to every operation; the model often called SI1000 [GNFB21], inspired by superconducting hardware, assigns to different operations different multiples of a single parameter (measurements, for instance, are much noisier than gates). Results obtained under one model do not transfer automatically to another.

**8.2. Detectors.** Consider a single check, measured in rounds  $t = 1, \dots, T$  with recorded outcome bits  $m_t$ , and suppose that in the absence of faults all outcomes would be 0 (with  $m_0 := 0$  known from the preparation). A persistent data error that anticommutes with the check changes all subsequent outcomes, while an error in recording one outcome changes only that outcome. The differences

$$\delta_t := m_t + m_{t-1} \quad (1 \leq t \leq T)$$

separate these cases:

| Cause                                 | Outcomes $m_1 \cdots m_5$ | Differences $\delta_1 \cdots \delta_5$ |
|---------------------------------------|---------------------------|----------------------------------------|
| data error before round 2, persisting | 01111                     | 01000                                  |
| recording error in round 2 only       | 01000                     | 01100                                  |

A data error produces one nonzero difference and a recording error produces two, adjacent in time. In the language of Section 3.4, the decoding graph acquires a time direction: a data error is an edge between two checks in the same round, and a recording error is an edge between the same check in consecutive rounds.

**8.1. Definition.** A *detector* is a parity of recorded measurement outcomes whose value is deterministic (we normalize it to 0) in the absence of faults. A *logical observable* is a parity of measurement outcomes whose value is the quantity the experiment is meant to preserve.

The differences  $\delta_t$  are detectors, as are the final comparisons between the last round of checks and the readout of the data qubits; the preparation and the final readout determine the detectors at the two time boundaries, and these cannot be omitted when comparing experiments. In a *memory experiment* for the rotated patch, for example, one prepares the encoded state  $[\bar{0}]$ , measures the checks for  $T$  rounds, reads out all data qubits in the computational basis, and takes as logical observable the parity of the readout along the top row, i.e., the value of  $\bar{Z}$ ; the experiment succeeds if the decoder predicts that parity correctly.

**8.3. Detector error models.** Let  $f \in \mathbb{F}_2^M$  indicate which of  $M$  elementary fault mechanisms occurred. If faults act linearly on detectors and observables (which holds for Pauli faults in circuits of Clifford gates), there are binary matrices  $D$  and  $O$  such that

$$(8.1) \quad \delta = Df, \quad o = Of,$$

where  $\delta$  is the vector of detector values and  $o$  the vector of observable flips. Together with independent prior probabilities for the columns, this is a *detector error model*. The decoder observes  $\delta$  and must predict  $o$ ; it succeeds if its prediction is correct, and consistency of an intermediate estimate  $\hat{f}$  (i.e.,  $D\hat{f} = \delta$ ) is again necessary but not sufficient. The inference problem has exactly the form Equation (4.2), with  $D$  in place of  $H_Z$ , so everything in Sections 4, 5 and 7 applies verbatim. What changes is the matrix: its columns are fault mechanisms, not qubits, and a single column can touch several detectors and observables.

Two approximations are hidden in this description and must be declared when used. Mutually exclusive faults at one circuit location (an  $X$ ,  $Y$  or  $Z$  after a gate, say) are modeled as independent variables, and a fault that flips more than two detectors may be split into several graphlike edges so that matching can be applied, losing the correlation between them. Stim [Gid21] computes detector error models from circuits exactly and records which faults are graphlike; its `file format` is a precise reference for the notions of this subsection.

**8.4. Comparing two decoders.** Decoders are compared by simulation: sample faults from the model, compute the detectors and observables, decode, and count failures. We illustrate the necessary care with the rotated patch of distance five under a standard circuit-level model (Table 1). The circuit, generated by Stim, performs five rounds of check measurements in a memory experiment. Its noise has four components, all with the same parameter  $p = 0.003$ : each Clifford gate is followed by a depolarizing fault on the qubits it acts on, each data qubit is depolarized at the start of each round, and each reset and each measurement is followed (respectively preceded) by a bit flip; there is no noise on idle qubits. We decoded the same sampled shots with matching (PyMatching, on Stim’s graphlike decomposition of the detector error model) and with BP+OSD (the 1dpc package, on the undecomposed model, with a combination sweep of order five). The comparison is deliberately simple and should not be read as a ranking of the best available decoders for this code: matching on the graphlike decomposition discards the correlations between the  $X$  and  $Z$  components of  $Y$ -type faults (Remark 2.11), which BP+OSD on the undecomposed model can use, and variants of matching that recover some of this information exist [Fow13].

| Decoder | Failures | Shots   | LER (%) | Wilson 95% (%) | $\mu\text{s}/\text{shot}$ | batched $\mu\text{s}/\text{shot}$ |
|---------|----------|---------|---------|----------------|---------------------------|-----------------------------------|
| MWPM    | 678      | 200,000 | 0.339   | [0.314, 0.365] | 2.9                       | 0.9                               |
| BP+OSD  | 508      | 200,000 | 0.254   | [0.233, 0.277] | 1154.9                    | –                                 |

| $b$ | $c$ | both fail | $z = (b - c)/\sqrt{b + c}$ | exact two-sided $p$   |
|-----|-----|-----------|----------------------------|-----------------------|
| 401 | 231 | 277       | 6.76                       | $1.4 \times 10^{-11}$ |

TABLE 1. Two decoders on the same simulated shots of a distance-five memory experiment under circuit-level noise with  $p = 0.003$ . The logical failure rate is per shot (five rounds), with 95% Wilson intervals. Times are illustrative single-threaded measurements on one machine; see the text.

Several points deserve emphasis, and they apply to any reported comparison.

*The failure event must be stated.* Here a shot fails if the single logical observable is predicted incorrectly at the end of five rounds. Some authors report a rate per round, obtained by solving  $1 - P_{\text{shot}} = (1 - P_{\text{round}})^T$ , or, if a shot is deemed to fail when the observable is flipped in an odd number of rounds,  $1 - 2P_{\text{shot}} = (1 - 2P_{\text{round}})^T$  (the convention of Stim’s companion tool `sinter`); the two agree to first order in  $P_{\text{round}}$ . For codes with several logical qubits one must also say whether a failure means an error on any of them (the usual convention in the qLDPC literature) or on a particular one. Rates defined differently are not comparable.

*Paired data should be analyzed as such.* Since both decoders saw the same shots, the relevant information is in the shots on which they disagree. Let  $b$  be the number of shots on which the first decoder fails and the second succeeds, and  $c$  the number with the opposite outcome; the difference of failure counts is  $b - c$ , but its uncertainty depends on  $b + c$ , which the totals do not determine. Under the hypothesis that the two decoders have the same failure probability, conditional on  $b + c$  the count  $b$  is binomial with parameters  $b + c$  and  $1/2$ ; this is McNemar’s test [McN47], and for large  $b + c$  the statistic  $(b - c)/\sqrt{b + c}$  is approximately standard normal. Dietterich [Die98] discusses this and related tests for comparing two classifiers on the same data, a problem of exactly the same form. Paired comparisons are typically much more sensitive than a comparison of two separate confidence intervals, and overlapping intervals do not settle the question.

*Accuracy and cost form a partial order.* A decoder must usually finish within a time budget set by the measurement rate of the hardware, so accuracy alone does not rank decoders. The pair (failure rate, cost) is only partially ordered: one decoder dominates another if it is no worse in both coordinates and strictly better in one. Measured times depend on the implementation, the language, batching, the hardware, and the distribution of difficult shots, whose tail matters more than the mean for a real-time decoder; proxies such as iteration counts are not measurements of time. In Table 1 the two decoders are timed under the same per-shot calling convention, but they remain different programs in different stages of optimization, and the times should be read only as orders of magnitude.

*Selection biases evaluation.* If a decoder or its parameters were chosen because they performed well on some data, their performance on those data overestimates their performance on fresh data. Tuning and evaluation should use independent samples, and comparisons among many candidates require a correction for multiple testing; the simplest is Bonferroni’s, which tests each of  $N$  comparisons at level  $\alpha/N$  so that the probability of any false rejection is at most  $\alpha$ .

**8.5. Notes and open questions.** Detector error models reduce fault-tolerant decoding to the inference problem of Sections 4, 5 and 7, but with matrices that are larger, less regular, and more strongly correlated than the check matrices of the underlying codes; the behaviour of a decoder under code-capacity noise need not predict its behaviour under circuit-level noise, and claims should be checked in the setting in which they are made. The methodology of decoder comparison is itself unsettled. Failure rates of interest are small, so that estimating them precisely requires many samples (or importance sampling, which brings its own modeling assumptions); costs depend on implementations that change faster than the algorithms; and the space of decoders and parameters is large enough that selection effects are a real concern. Real-time decoding adds a further constraint, since the decoder must keep pace with the measurement rate of the hardware, and the tail of the distribution of decoding times matters more than its mean.

## 8.6. Problems.

- 8.1.** With  $m_0 = 0$ , compute the differences  $\delta_t$  for the outcome sequences 01111, 01000 and 00110, and describe for each the simplest fault history consistent with it.
- 8.2.** Convert the per-shot failure rates of Table 1 into per-round rates, and explain why the conversion assumes that failures in different rounds are independent and rare.
- 8.3.** Using the paired counts of Table 1, compute the statistic  $(b - c)/\sqrt{b + c}$ . Then determine the range of values that this statistic could take over all paired outcomes consistent with the two failure totals alone, and compare the extreme cases.
- 8.4.** In a memory experiment on the nine-qubit patch with  $T$  rounds, count the detectors, including those formed by comparing the final readout of the data qubits with the last round of  $Z$  checks.

## APPENDIX A. QUBITS, PROJECTIVE GEOMETRY, AND THE BLOCH SPHERE

The pure states of one qubit form a familiar mathematical space: the complex projective line, or Riemann sphere. This appendix makes that identification explicit and interprets Pauli and Clifford operators as projective transformations. We then distinguish the projective space of a tensor product from the product of the projective spaces; this distinction accounts for entangled pure states. The projective definition of pure states is used throughout the main text; the sphere picture and the additional group-theoretic descriptions here are optional. For the quantum formalism see Nielsen and Chuang [NC10], Sections 1.2 and 2.2; for Clifford operators see Gottesman [Got09], Section 3.5.

**A.1. Pure states as projective points.** Recall from Definition 1.1 that  $\mathbb{P}(\mathcal{H}_n)$  is the pure-state space, whose points  $[u]$  are the one-dimensional complex subspaces of  $\mathcal{H}_n$ . For one qubit it is the *complex projective line*  $\mathbb{P}(\mathbb{C}^2) = \mathbb{CP}^1$ . Its complex dimension is one, although its points are lines in a two-dimensional vector space. The inner product equips this projective line with the transition probabilities of Equation (1.1); in the sphere picture below these are determined by the round metric (Problem A.1).

We write  $[z : w]$  for the line through  $(z, w) \neq (0, 0)$ . The map

$$(A.1) \quad [z : w] \mapsto \beta := z/w \quad (w \neq 0), \quad [1 : 0] \mapsto \infty$$

identifies  $\mathbb{CP}^1$  with the *extended complex plane*  $\hat{\mathbb{C}} := \mathbb{C} \cup \{\infty\}$ . Near infinity we use the coordinate  $w/z = 1/\beta$ ; these two charts give  $\hat{\mathbb{C}}$  its structure as the *Riemann sphere*. The computational states  $[|0\rangle]$  and  $[|1\rangle]$  correspond to  $\infty$  and 0, respectively.

**A.2. An oriented sphere and the Pauli action.** Let  $B := \{b = (b_1, b_2, b_3) \in \mathbb{R}^3 : \|b\| = 1\}$  have its outward orientation. Identify the complex plane with  $\{b_3 = 0\}$  by  $(x, y, 0) \leftrightarrow x + iy$ . Stereographic projection from the south pole  $s = (0, 0, -1)$  is

$$\beta = \frac{b_1 + ib_2}{1 + b_3}, \quad s \mapsto \infty.$$

Its inverse, composed with Equation (A.1), is

$$(A.2) \quad b([z : w]) = \frac{(2 \operatorname{Re}(z\bar{w}), 2 \operatorname{Im}(z\bar{w}), |w|^2 - |z|^2)}{|z|^2 + |w|^2}, \quad b(\beta) = \frac{(2 \operatorname{Re} \beta, 2 \operatorname{Im} \beta, 1 - |\beta|^2)}{1 + |\beta|^2}.$$

The first expression is unchanged by multiplying  $(z, w)$  by a nonzero complex scalar, and its norm is one. The inverse pulls the round metric back to  $4|d\beta|^2/(1 + |\beta|^2)^2$ , so it is conformal. At  $\beta = 0$  its two coordinate tangent vectors have cross product  $(0, 0, 4)$ , pointing outward at the north pole; hence the identification preserves orientation. This realization of the pure-state space is called the *Bloch sphere*. Mixed one-qubit states, in the density-operator description of Section B, fill the closed ball bounded by this sphere.

**A.1. Remark.** The pole and the complex coordinate must be specified together. With the usual coordinate order on  $\mathbb{R}^3$ , the south pole is  $(0, 0, -1)$  and the projection plane is the  $xy$ -plane. Our convention keeps  $\beta = z/w$  and south-pole projection simultaneously. For a unit vector  $u = z|0\rangle + w|1\rangle$ , the often-used Pauli expectation coordinates are

$$r = (\langle u, Xu \rangle, \langle u, Yu \rangle, \langle u, Zu \rangle) = (b_1, -b_2, -b_3).$$

Thus the two sphere pictures differ by a rotation through  $\pi$  about the first axis. In our picture  $[|0\rangle]$  is the south pole and  $[|1\rangle]$  the north pole.

An invertible matrix  $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$  acts on lines by  $[u] \mapsto [Au]$ . Scalar multiples of  $A$  act identically, and in the affine coordinate the action is

$$(A.3) \quad \beta \mapsto \frac{a\beta + b}{c\beta + d}.$$

These are the *Möbius transformations*, and their group is

$$\mathrm{PGL}(2, \mathbb{C}) := \mathrm{GL}(2, \mathbb{C}) / \mathbb{C}^\times I.$$

It is exactly the group of holomorphic (equivalently, conformal and orientation-preserving) automorphisms of the Riemann sphere. Indeed, a meromorphic function on the Riemann sphere is rational, and a bijective rational map has degree one, hence has the form [Equation \(A.3\)](#). The rationality assertion follows by subtracting the principal parts at its finitely many poles and applying Liouville's theorem to the remainder (including the pole at infinity).

For the matrices of [Definition 1.5](#), direct multiplication gives

$$X[z : w] = [w : z], \quad Y[z : w] = [-iw : iz] = [-w : z], \quad Z[z : w] = [z : -w].$$

Consequently their actions in the two pictures are

$$(A.4) \quad \begin{array}{c|cc} & \hat{\mathbb{C}} & B \\ \hline X & \beta \mapsto 1/\beta & (b_1, b_2, b_3) \mapsto (b_1, -b_2, -b_3) \\ Y & \beta \mapsto -1/\beta & (b_1, b_2, b_3) \mapsto (-b_1, b_2, -b_3) \\ Z & \beta \mapsto -\beta & (b_1, b_2, b_3) \mapsto (-b_1, -b_2, b_3). \end{array}$$

Here  $1/0 = \infty$ ,  $1/\infty = 0$ , and  $-\infty = \infty$ . Thus each Pauli matrix induces a half-turn about its respective coordinate axis. Abusing notation, we have used  $X, Y, Z$  also for their induced transformations.

The projective image  $\overline{\mathcal{P}}_1$  of the Pauli matrix group is the Klein four-group  $\{[I], [X], [Y], [Z]\}$ . The matrix group  $\mathcal{P}_1$  itself has sixteen elements,

$$\{\lambda P : \lambda \in \{1, -1, i, -i\}, P \in \{I, X, Y, Z\}\};$$

each projective transformation has four representatives in this group. For instance,  $XZ = -ZX$  as matrices but  $[X][Z] = [Z][X]$  as projective transformations.

**A.2. Remark.** A global scalar is immaterial to the action of an operator on pure states, but the scalar relations between chosen lifts retain useful information. The commutator  $XZX^{-1}Z^{-1} = -I$  records the anticommutation that defines the syndrome in [Proposition 2.3](#). Moreover,  $Z$  and  $-Z$  induce the same projective map but have different  $+1$  eigenspaces. Hence a projective Pauli subgroup alone does not specify the stabilizer code: the signed lifts of its checks are needed. The main text uses signed lifts to present this measurement data, while the state transformations belong to  $\overline{\mathcal{P}}_n$ .

Möbius transformations need not preserve the round metric or measurement probabilities. The subgroup represented by unitary matrices is

$$\mathrm{PU}(2) = \mathrm{U}(2) / \mathrm{U}(1) \cong \mathrm{SU}(2) / \{\pm I\} \cong \mathrm{SO}(3),$$

and acts by rotations of  $B$ . To see the last identification, conjugate the traceless Hermitian matrices  $r_1X + r_2Y + r_3Z$  by  $U \in \mathrm{SU}(2)$ . Conjugation preserves the inner product  $\mathrm{Tr}(AB)/2$  and has kernel  $\{\pm I\}$ ; its image consists of rotations, since  $\mathrm{SU}(2)$  is connected. Every rotation is obtained: for a unit vector  $a \in \mathbb{R}^3$ , conjugation by  $\exp(-i\theta(a_1X + a_2Y + a_3Z)/2)$  rotates the expectation coordinates through angle  $\theta$  about  $a$ , as follows by expanding the exponential using  $(a_1X + a_2Y + a_3Z)^2 = I$ . The rotated coordinate convention of [Remark A.1](#) gives the same rotation group on  $B$ . For example,  $\mathrm{diag}(2, 1)$  induces the Möbius transformation  $\beta \mapsto 2\beta$ , which is not a rotation.

**A.3. Tensor products and entanglement.** The pure states of  $n$  qubits form

$$\mathbb{P}((\mathbb{C}^2)^{\otimes n}) \cong \mathbb{CP}^{2^n-1},$$

where the standard tensor basis identifies  $(\mathbb{C}^2)^{\otimes n}$  with  $\mathbb{C}^{2^n}$ . We take the tensor product of vector spaces first and projectivize the resulting space once. There is a natural map, called the *Segre embedding*,

$$(A.5) \quad (\mathbb{CP}^1)^n \longrightarrow \mathbb{CP}^{2^n-1}, \quad ([u_1], \dots, [u_n]) \longmapsto [u_1 \otimes \dots \otimes u_n].$$

It is well-defined because rescaling any factor only rescales the tensor, and is injective because the line of a nonzero decomposable tensor determines each factor line. Its image consists of the *product pure states*; pure states outside it are called *entangled* (relative to these tensor factors). Thus entanglement is the failure of this embedding to be surjective, rather than a further identification of product states.

For two qubits, the coefficients of  $a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$  form the matrix

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}.$$

The vector is a product precisely when this nonzero matrix has rank one, i.e.,  $ad - bc = 0$ . The image of Equation (A.5) is therefore a quadric in  $\mathbb{CP}^3$ . The state  $[|00\rangle + |11\rangle]$  lies outside this quadric and is entangled; its unit representatives include  $(|00\rangle + |11\rangle)/\sqrt{2}$ . No tuple of points of  $B$  describes such a state.

**A.4. The projective Clifford group.** For  $n \geq 1$ , let  $\overline{\mathcal{P}}_n$  denote the image of  $\mathcal{P}_n$  in  $\text{PGL}(\mathcal{H}_n) = \text{PGL}(2^n, \mathbb{C})$ . Then

$$(A.6) \quad 1 \longrightarrow \{\pm I, \pm iI\} \longrightarrow \mathcal{P}_n \longrightarrow \overline{\mathcal{P}}_n \longrightarrow 1, \quad \overline{\mathcal{P}}_n \cong \mathbb{F}_2^{2n}.$$

This is the quotient already used in the binary coordinates of Proposition 2.8.

**A.3. Definition.** The *projective Clifford group* is the normalizer

$$\text{Cl}_n^{\text{proj}} := N_{\text{PGL}(\mathcal{H}_n)}(\overline{\mathcal{P}}_n) = \{[A] : [A]\overline{\mathcal{P}}_n[A]^{-1} = \overline{\mathcal{P}}_n\}.$$

Using the full projective linear group in this definition introduces no additional nonunitary elements.

**A.4. Proposition.** Every element of  $\text{Cl}_n^{\text{proj}}$  has a unitary representative, and

$$\text{Cl}_n^{\text{proj}} \cong N_{\text{U}(\mathcal{H}_n)}(\mathcal{P}_n)/\text{U}(1).$$

*Proof.* Let  $[A]$  normalize  $\overline{\mathcal{P}}_n$ , and let  $P$  be a tensor product of  $I, X, Y, Z$ . We have  $APA^{-1} = cQ$  for some such tensor product  $Q$  and  $c \in \mathbb{C}^\times$ . Squaring gives  $c^2 = 1$ , so  $APA^{-1} = \pm Q$ . Taking adjoints and using that  $P, Q$  are self-adjoint gives

$$APA^{-1} = (A^{-1})^\dagger P A^\dagger, \quad A^\dagger A P = P A^\dagger A.$$

These  $P$  span the full matrix algebra, so  $A^\dagger A = tI$  for some  $t > 0$ . Hence  $t^{-1/2}A$  is unitary and normalizes the matrix group  $\mathcal{P}_n$ . Conversely, any unitary normalizer induces a projective normalizer, and the kernel is exactly the scalar unitaries.  $\square$

The following gates, acting on any indicated tensor factors, generate  $\text{Cl}_n^{\text{proj}}$ :

$$(A.7) \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \quad \text{CNOT}_{c \rightarrow t} |a\rangle_c |b\rangle_t = |a\rangle_c |a + b\rangle_t,$$

where  $a, b \in \mathbb{F}_2$ . Here  $S$  denotes the *phase gate*, rather than a stabilizer subgroup. Their normalizing property follows from

$$\begin{aligned} HXH^{-1} &= Z, & HZH^{-1} &= X, & SXS^{-1} &= Y, & SZS^{-1} &= Z, \\ \text{CNOT}_{c \rightarrow t} : & X_c \mapsto X_c X_t, & X_t &\mapsto X_t, & Z_c &\mapsto Z_c, & Z_t &\mapsto Z_c Z_t, \end{aligned}$$

where the arrows on the second line also mean conjugation. Generation is a further theorem, not a consequence of these identities alone; see Gottesman [Got97], Section 5.8. For one qubit the affine formulas are  $H(\beta) = (\beta + 1)/(\beta - 1)$  and  $S(\beta) = -i\beta$ . For two qubits, CNOT takes the product state  $[|+\rangle \otimes |0\rangle]$  to  $[|00\rangle + |11\rangle]$ , so Clifford transformations can create entanglement.

Conjugation preserves matrix commutators, hence the form  $\omega$  of Equation (2.3). More precisely, there is an exact sequence

$$(A.8) \quad 1 \longrightarrow \overline{\mathcal{P}}_n \longrightarrow \text{Cl}_n^{\text{proj}} \longrightarrow \text{Sp}(2n, \mathbb{F}_2) \longrightarrow 1.$$

Here  $\text{Sp}(2n, \mathbb{F}_2)$  is the group of invertible linear maps preserving  $\omega$ ; surjectivity is proved in Gottesman [Got09], Section 3.5, and the kernel is computed in Problem A.2. In particular, the projective Clifford group is finite.

**A.5. Remark.** The scalar kernel depends on the chosen matrix group above the projective group. The full unitary normalizer in Proposition A.4 contains every scalar  $e^{i\theta}I$ , so its kernel is  $U(1)$ . Alternatively, take the matrix subgroup  $\tilde{\text{Cl}}_n$  generated by the specific gates in Equation (A.7). Its scalar subgroup is  $\mu_8 I$ , where  $\mu_8 := \{e^{\pi i k/4} : 0 \leq k < 8\}$ , and

$$\text{Cl}_n^{\text{proj}} \cong \tilde{\text{Cl}}_n / \mu_8 I.$$

Indeed,  $(HS)^3 = e^{\pi i/4}I$ , so all eight scalars occur. Conversely, every gate has entries in  $\mathbb{Q}(e^{\pi i/4})$  and determinant a root of unity; a scalar product of gates is therefore a root of unity in this field, whose roots of unity are precisely  $\mu_8$ . This last elementary cyclotomic fact can also be seen from  $\mathbb{Q}(e^{\pi i/4}) = \mathbb{Q}(i, \sqrt{2})$ : a root of unity of order  $m$  in this degree-four field has  $\varphi(m) \mid 4$ , so  $m \in \{1, 2, 3, 4, 5, 6, 8, 10, 12\}$ ; the orders divisible by 3 or by 5 are excluded because they would force the quadratic subfield  $\mathbb{Q}(\sqrt{-3})$  or  $\mathbb{Q}(\sqrt{5})$ , whereas the only quadratic subfields of  $\mathbb{Q}(i, \sqrt{2})$  are  $\mathbb{Q}(i)$ ,  $\mathbb{Q}(\sqrt{2})$  and  $\mathbb{Q}(\sqrt{-2})$ . Thus this finite Clifford lift has eight scalar representatives of each projective element, whereas the Pauli matrix group has four.

#### A.5. Problems.

- A.1.** Let  $u, v \in \mathbb{C}^2$  be unit vectors, with sphere points  $b, b'$  as in Equation (A.2). Show that  $|\langle u, v \rangle|^2 = (1 + b \cdot b')/2$ . Deduce that orthogonal pure states correspond to antipodal points, and that the affine coordinate of the state orthogonal to  $\beta$  is  $-1/\beta$ . Interpret a rank-one projective measurement in terms of this sphere geometry.
- A.2.** Prove that the centralizer of  $\overline{\mathcal{P}}_n$  in  $\text{PGL}(\mathcal{H}_n)$  is  $\overline{\mathcal{P}}_n$  itself, and deduce the kernel assertion in Equation (A.8). (Hint: for a representative  $A$  of an element of the centralizer, write  $APA^{-1} = \chi(P)P$  and use the nondegeneracy of  $\omega$  to realize the signs  $\chi(P)$  by conjugation with a Pauli operator.)

### APPENDIX B. DENSITY OPERATORS, CHANNELS, AND EXACT RECOVERY

The main text treats pure states as projective points, with projective measurements and Pauli errors described by linear lifts. This appendix embeds those pure states in the convex space of density operators, supplies general noise and recovery channels, and proves the statement used repeatedly there: correcting a set of Pauli errors corrects every channel whose Kraus operators lie in the linear span of their lifts. For background see Nielsen and Chuang [NC10], Chapters 2, 8 and 10.

**B.1. Density operators and channels.** A *density operator* on a finite-dimensional Hilbert space  $\mathcal{H}$  is a positive semidefinite operator  $\rho$  with  $\text{Tr } \rho = 1$ . A pure state  $[u]$  determines the rank-one density operator

$$(B.1) \quad \rho_{[u]} := \frac{|u\rangle\langle u|}{\langle u, u \rangle},$$

independently of the nonzero representative  $u$ . This identifies  $\mathbb{P}(\mathcal{H})$  with the rank-one density operators; for a unit representative,  $\rho_{[u]} = |u\rangle\langle u|$ . A convex combination  $\sum_j p_j \rho_{[u_j]}$  describes classical uncertainty among pure states, although different ensembles may give the same operator. Density operators of rank greater than one are called *mixed states*; they are not individual projective points. For a projective measurement  $(P_a)$ , outcome  $a$  has probability  $\text{Tr}(P_a \rho)$  and leaves the state  $P_a \rho P_a / \text{Tr}(P_a \rho)$ ; for  $\rho = \rho_{[u]}$  these give exactly the probability and output ray of Definition 1.2.

A *channel* is a completely positive, trace-preserving linear map  $\mathcal{N}$  on the operators of  $\mathcal{H}$ ; complete positivity means that  $\mathcal{N} \otimes \text{id}_{\mathcal{K}}$  is positive for every auxiliary space  $\mathcal{K}$ . In finite dimensions every channel

has a *Kraus representation*

$$\mathcal{N}(\rho) = \sum_a E_a \rho E_a^\dagger, \quad \sum_a E_a^\dagger E_a = I,$$

which is not unique: another family  $F_\mu = \sum_a u_{\mu a} E_a$ , with  $(u_{\mu a})$  an isometry, represents the same channel. These are standard facts of finite-dimensional operator theory, and we use them without proof. A stochastic Pauli model, in which  $E$  occurs with probability  $\pi(E)$ , is the channel with Kraus operators  $\sqrt{\pi(E)} E$ . General channels can send pure inputs to mixed outputs, so they are not, in general, maps of the pure-state projective space to itself.

**B.2. The Knill–Laflamme condition.** Let  $P$  be the orthogonal projection onto the underlying subspace  $\mathcal{C}$  of a projective code  $\mathbb{P}(\mathcal{C})$ . A channel  $\mathcal{R}$  *recovers* from  $\mathcal{N}$  if  $\mathcal{R}\mathcal{N}(\rho) = \rho$  for every density operator  $\rho$  supported on  $\mathcal{C}$ . It is enough to require this for every pure-state density operator  $\rho_{[u]}$  with  $[u] \in \mathbb{P}(\mathcal{C})$ , since every density operator on  $\mathcal{C}$  is a convex combination of these. In fact, rank-one orthogonal projections span the full complex algebra of operators on  $\mathcal{C}$ : projections onto basis lines and onto the lines of  $e_i + e_j$  and  $e_i + ie_j$  span the matrix units. Hence  $\mathcal{R}\mathcal{N}$  is the identity on that algebra and  $(\mathcal{R}\mathcal{N} \otimes \text{id})(\sigma) = \sigma$  for every density operator  $\sigma$  on  $\mathcal{C} \otimes \mathcal{K}$ . Thus recovery preserves entanglement with an untouched reference system as well.

**B.1. Theorem** (Knill–Laflamme [KL97]). *A recovery channel for  $\mathcal{N}$  exists if and only if there are complex numbers  $c_{ab}$  such that*

$$(B.2) \quad P E_a^\dagger E_b P = c_{ab} P \quad \text{for all } a, b.$$

*Proof of sufficiency.* The matrix  $C = (c_{ab})$  is positive semidefinite, since for coefficients  $v_a$  the operator  $P(\sum_a v_a E_a)^\dagger (\sum_b v_b E_b)P = (\bar{v}^\top C v)P$  is positive. A unitary change of Kraus operators diagonalizes  $C$ , so we may assume  $P F_\mu^\dagger F_\nu P = \lambda_\mu \delta_{\mu\nu} P$  with  $\lambda_\mu \geq 0$ , and  $\sum_\mu \lambda_\mu = 1$  by trace preservation. For  $\lambda_\mu > 0$ , the operators  $V_\mu = F_\mu P / \sqrt{\lambda_\mu}$  restrict to isometries on  $\mathcal{C}$  with pairwise orthogonal ranges, while  $F_\mu P = 0$  if  $\lambda_\mu = 0$ . Measure the projections onto these ranges (completed by the projection onto the orthogonal complement of their sum) and, on outcome  $\mu$ , apply  $V_\mu^\dagger$ ; on the complement apply any channel. Each branch returns the input state, so the composite is the identity on states supported on  $\mathcal{C}$ .  $\square$

Necessity, which we do not need, is proved in [KL97, Got09].

### B.3. Why a Pauli basis suffices.

**B.2. Theorem.** *If the operators  $E_a$  satisfy Equation (B.2), then so does every finite family of operators in their linear span. In particular, for a stabilizer code of distance  $d$  and  $2t < d$ , every channel whose Kraus operators are linear combinations of Pauli operators of weight at most  $t$  admits a recovery, and the recovery of Theorem 2.6 (measure the syndrome, then apply  $\rho(s)$ ) is one.*

*Proof.* If  $F_\mu = \sum_a u_{\mu a} E_a$ , then

$$P F_\mu^\dagger F_\nu P = \sum_{a,b} \overline{u_{\mu a}} u_{\nu b} c_{ab} P,$$

which is of the required form. For the second statement, let  $E, F$  be Pauli operators of weight at most  $t$ . If  $E^\dagger F$  has nonzero syndrome, it maps  $\mathcal{C}$  into an orthogonal eigenspace of some check (Equation (2.2)), so  $P E^\dagger F P = 0$ ; otherwise  $E^\dagger F \in \mathcal{Z}(S)$  has weight less than  $d$ , hence lies in  $\langle iI \rangle S$ , and  $P E^\dagger F P$  is a scalar multiple of  $P$  by Proposition 2.4. This verifies Equation (B.2) for the Pauli operators of weight at most  $t$ , and the first statement extends it to their span. That the syndrome recovery works is checked directly: a Kraus operator  $\sum_E u_E E$  applied to  $u \in \mathcal{C}$  gives  $\sum_\sigma (\sum_{s(E)=\sigma} u_E E)u$ , the syndrome measurement selects one value of  $\sigma$ , and on that branch every  $E$  with  $s(E) = \sigma$  satisfies  $\rho(\sigma)E|_{\mathcal{C}} = \lambda_E I_{\mathcal{C}}$  for a phase  $\lambda_E$  independent of  $u$  (because these errors differ by elements of  $\langle iI \rangle S$ ). The phases may depend on  $E$ , but the branch returns  $(\sum_{s(E)=\sigma} u_E \lambda_E)u$ , hence a multiple of  $u$ .  $\square$

This theorem does not say that every channel is a stochastic Pauli channel. For the coherent rotation  $U = \cos \theta I + i \sin \theta X_1$  of [Remark 1.7](#), the state  $U\rho U^\dagger$  contains cross terms  $\cos \theta \sin \theta (iX_1\rho - i\rho X_1)$  absent from the mixture  $\cos^2 \theta \rho + \sin^2 \theta X_1\rho X_1$ ; the syndrome measurement removes these cross terms only because the two components have different syndromes. Components with the same syndrome are not decohered by the measurement. The stochastic Pauli models of the main text are therefore explicit modeling assumptions, made to formulate decoding as classical inference; they are not consequences of the theorem.

**B.4. A circuit that measures a check.** For readers who want a realization of the joint measurement of [Proposition 1.3](#), consider a Z check  $Z(b)$  with support  $J$ . Adjoin an auxiliary qubit in the state  $|0\rangle$  and, for each  $j \in J$ , apply the controlled-NOT gate from qubit  $j$  to the auxiliary qubit, with linear lift  $|c\rangle|a\rangle \mapsto |c\rangle|a+c\rangle$  on that pair. On computational basis vectors the composite acts by

$$|x\rangle|0\rangle \mapsto |x\rangle \left| \sum_{j \in J} x_j \right\rangle.$$

Thus a state  $[u]$ , with representative  $u = u_+ + u_-$  decomposed into the two eigenspaces of  $Z(b)$ , is taken to  $[u_+|0\rangle + u_-|1\rangle]$ . Measuring the auxiliary qubit in the computational basis therefore realizes the measurement of  $Z(b)$ , including its post-measurement state. Conjugating the data qubits by the Hadamard matrix  $2^{-1/2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ , which exchanges  $X$  and  $Z$ , gives a circuit for an X check.

This ideal circuit says nothing about fault tolerance. A fault on the auxiliary qubit in the middle of the circuit can propagate, through the remaining gates, to several data qubits; that is why the circuit-level models of [Section 8](#) depend on the schedule of the gates and not only on the check matrices.

## APPENDIX C. SOLUTIONS TO THE PROBLEMS

Numerical answers about the small examples agree with the companion enumeration.

### Section 1.

**Problem 1.1.:**  $X_1X_3$  gives  $(1, 1)$ ,  $Z_2$  gives  $(0, 0)$ , and  $Y_1 = iX_1Z_1$  gives  $(1, 0)$ , the outcome of  $X_1$ . The rule “apply  $X_1, X_2, X_3$  after  $(1, 0), (1, 1), (0, 1)$ ” corrects  $I, X_1, X_2, X_3$ ; since  $X_1X_3$  and  $X_2$  share the outcome  $(1, 1)$  and differ by  $X_1X_2X_3$ , which exchanges  $|000\rangle$  and  $|111\rangle$ , no rule corrects both.

**Problem 1.2.:** In the computational basis,  $P_{++}$  is the diagonal projection onto  $\text{span}\{|000\rangle, |111\rangle\}$ ,  $P_{-+}$  onto  $\text{span}\{|100\rangle, |011\rangle\}$ ,  $P_{--}$  onto  $\text{span}\{|010\rangle, |101\rangle\}$ , and  $P_{+-}$  onto  $\text{span}\{|001\rangle, |110\rangle\}$ , where the first sign refers to  $g_1$ . The four sets partition the eight basis vectors, so the projections sum to  $I$ .

**Problem 1.3.:** Direct computation gives  $X|\pm\rangle = \pm|\pm\rangle$  and  $Z|\pm\rangle = |\mp\rangle$ . In the basis  $|\pm\rangle^{\otimes 3}$  the roles of  $X$  and  $Z$  are exchanged, so the projectivization of  $\text{span}\{|+++\rangle, |---\rangle\}$  with checks  $X_1X_2, X_2X_3$  is the repetition code written in that basis: each  $Z_j$  produces a distinct nonzero outcome, while  $X_1$  has trivial outcome and maps  $\alpha|+++\rangle + \beta|---\rangle$  to  $\alpha|+++\rangle - \beta|---\rangle$ .

**Problem 1.4.:** The condition on the two computational states forces  $T|0\rangle = a|000\rangle$  and  $T|1\rangle = b|111\rangle$  with  $a, b \neq 0$ . By linearity,  $T(|0\rangle + |1\rangle) = a|000\rangle + b|111\rangle$ ; this cannot represent the same projective point as  $(|0\rangle + |1\rangle)^{\otimes 3} = \sum_{c \in \{0,1\}^3} |c\rangle$ , which has eight nonzero coefficients. Nevertheless, rescaling  $u$  by  $\lambda \neq 0$  rescales its tensor cube by  $\lambda^3$ , so  $[u] \mapsto [u^{\otimes 3}]$  is a well-defined projective map. It is nonlinear and has no linear lift of the required kind.

### Section 2.

**Problem 2.1.:** If  $-I \in S$  and  $u$  is fixed by every element of  $S$ , then  $u = -u$ , so  $u = 0$ .

**Problem 2.2.:** Let  $W_\sigma$  be the joint eigenspace with eigenvalues  $((-1)^{\sigma_j})_j$ . By [Equation \(2.2\)](#),  $E_\sigma \mathcal{C}(S) \subseteq W_\sigma$ , and  $\dim E_\sigma \mathcal{C}(S) = 2^k$  since  $E_\sigma$  is unitary. The  $2^r$  spaces  $W_\sigma$  are the ranges of the projections of [Proposition 1.3](#), hence mutually orthogonal with sum  $\mathcal{H}_n$ ; as  $2^r \cdot 2^k = 2^n$ , all inclusions are equalities. In particular  $E_\sigma \mathcal{C}(S) = W_\sigma$  does not depend on the choice of  $E_\sigma$ .

**Problem 2.3.:** By Proposition 2.4,  $\bar{Z}$  restricted to the two-dimensional code has trace zero, and it is a self-adjoint involution, so it has eigenvalues  $+1$  and  $-1$  with one-dimensional eigenspaces. Let  $v_0$  be a unit  $+1$  eigenvector and put  $v_1 = \bar{X}v_0$ . Anticommutation gives  $\bar{Z}v_1 = -v_1$ , and  $\bar{X}v_1 = \bar{X}^2 v_0 = v_0$ . In the basis  $v_0, v_1$  the two operators are  $Z$  and  $X$ .

**Problem 2.4.:** The generators are pure  $X$  or pure  $Z$  operators without phases, so every element of  $S$  has the form  $\pm X(a)Z(b)$  with  $a \in \text{row}(H_X)$  and  $b \in \text{row}(H_Z)$ . Thus  $X(x) \in \langle iI \rangle S$  forces  $(x, 0) = (a, b)$ , i.e.,  $x \in \text{row}(H_X)$ . Conversely, a sum of rows of  $H_X$  corresponds to the product of the corresponding  $X$  checks, which is exactly  $X(x)$  because  $X$  operators commute.

**Problem 2.5.:** Each row of  $H$  has weight four, and two distinct rows overlap in two columns, so  $HH^T = 0$ . The rank is three, so  $k = 7 - 3 - 3 = 1$ . The kernel of  $H$  is the Hamming code, whose nonzero vectors have weight at least three, while  $\text{row}(H)$  consists of 0 and seven vectors of weight four; a Hamming codeword of weight three is therefore in  $\ker H \setminus \text{row}(H)$ , and  $d_X = d_Z = 3$ .

**Problem 2.6.:** Order the  $X$  checks as  $X_1 \cdots X_6, X_4 \cdots X_9$  and the  $Z$  checks as  $Z_1 Z_2, Z_2 Z_3, \dots, Z_8 Z_9$ . Then  $Z_1$  has  $X$ -check syndrome  $(1, 0)$  and  $Z_4$  has  $(1, 1)$ ;  $X_5$  has  $Z$ -check syndrome  $(0, 0, 1, 1, 0, 0)$ , and  $Y_5$  has both of the last two syndromes. The  $Z$  errors within one block share a syndrome  $((1, 0), (1, 1), (0, 1))$  for the three blocks), and any two of them differ by a  $Z$  check; correcting one by another leaves a stabilizer, so the sharing is harmless. The nine single-qubit  $X$  errors have distinct syndromes.

### Section 3.

**Problem 3.1.:**  $X$  boundary checks lie in rows 0 and  $d - 1$ ,  $Z$  boundary checks in columns 0 and  $d - 1$ , so only the four corner qubits are at issue. The corner  $(0, 0)$  lies in the top check with  $c = 0$  but in no left check, since those have odd  $r \geq 1$ ;  $(0, d - 1)$  lies in the right check with  $r = 0$  but in no top check, since  $c = d - 2$  is odd;  $(d - 1, 0)$  lies in the left check with  $r = d - 2$  (odd) but in no bottom check, since  $c = 0$  is even; and  $(d - 1, d - 1)$  lies in the bottom check with  $c = d - 2$  (odd) but in no right check, since  $r = d - 2$  is odd.

**Problem 3.2.:** The rotation maps the square with corner  $(r, c)$  to the square with corner  $(c, d - 2 - r)$ , whose parity  $c + d - 2 - r \equiv r + c + 1 \pmod{2}$  is opposite because  $d - 2$  is odd. It maps the top side to the right side, the right side to the bottom, the bottom to the left, and the left to the top; the top check on columns  $c, c + 1$  goes to the right check on rows  $c, c + 1$ , and  $c$  even is the condition for both, and similarly for the other three sides.

**Problem 3.3.:** A qubit  $(i, c)$  or  $(i + 1, c)$  of the two rows lies in exactly one of the squares with corners  $(i, c - 1)$  and  $(i, c)$  of type  $Z$  when  $0 < c < d - 1$ , by parity; at  $c = 0$  it lies in the square  $(i, 0)$  if  $i$  is even and in the left boundary check if  $i$  is odd, and at  $c = d - 1$  similarly with the square  $(i, d - 2)$  and the right boundary check. Hence each qubit of the two rows is covered exactly once, and no other qubit is covered.

**Problem 3.4.:** An interior qubit is a corner of four squares, two of each type. A qubit  $(r, 0)$  with  $0 < r < d - 1$  is a corner of two squares of opposite types and lies in exactly one left boundary check (the one with odd index among  $r - 1, r$ ); similarly on the right. A qubit  $(0, c)$  with  $0 < c < d - 1$  is a corner of two squares of opposite types and of no  $Z$  boundary check; the corners  $(0, 0)$  and  $(0, d - 1)$  lie in the  $Z$  square  $(0, 0)$  and in the right  $Z$  check, respectively; the bottom row is analogous.

**Problem 3.5.:** In the order  $(B_1, B_2, B_3, B_4)$ , the  $X$  errors on qubits 1, ..., 9 have syndromes 1000, 1000, 0001, 1010, 1100, 0101, 0010, 0100, 0100. In the order  $(A_1, A_2, A_3, A_4)$ , the  $Z$  errors have syndromes 1000, 1100, 0100, 0010, 0110, 0100, 0010, 0011, 0001. The coincident pairs are  $X_1, X_2$  (differing by  $A_1$ ),  $X_8, X_9$  (by  $A_4$ ),  $Z_3, Z_6$  (by  $B_4$ ), and  $Z_4, Z_7$  (by  $B_3$ ).

**Problem 3.6.:** Since  $Y_1 = iX_1Z_1$ , the operator equals  $iX_1X_4X_7Z_1Z_2Z_3 = i\bar{X}\bar{Z}$ , a logical operator (it anticommutes with both  $\bar{X}$  and  $\bar{Z}$ ) of weight five.

**Problem 3.7.:** Each edge has two endpoints and lies on two faces, so the product of all vertex checks and the product of all face checks are  $I$ . Conversely, if a nonempty set  $T$  of vertices has product  $I$ , every edge has both or neither endpoints in  $T$ , and connectivity of the grid forces  $T$  to contain all vertices; the same argument applies to faces through the dual grid. Hence both check matrices have rank  $L^2 - 1$ , and  $k = 2L^2 - 2(L^2 - 1) = 2$ .

## Section 4.

**Problem 4.1.:** The residuals are computed in the text. Up to the common factor  $\mathbb{P}(H_Z e = (1, 0, 0, 0))^{-1}$ , the posterior probabilities are  $p(1 - p)^8$  for  $\{1\}$  and for  $\{2\}$ , and  $p^2(1 - p)^7$  for  $\{4, 7\}$ .

**Problem 4.2.:** Equality  $w_5 = w_1 + w_8$  means  $(1 - p_5)/p_5 = ((1 - p_1)/p_1)((1 - p_8)/p_8)$ ; with  $p_1 = p_8 = 0.1$  this gives  $p_5 = 1/82 \simeq 0.0122$ . With equal priors every edge has the same weight and matching outputs the single edge  $\{5\}$ .

**Problem 4.3.:** The candidate errors 100 (or 010), 001, 111 have masses  $p(1 - p)(1 - q)$ ,  $(1 - p)^2 q$ ,  $p^2 q$ , so 001 is the most likely error exactly when  $q > p$ . The class  $\{001, 111\}$  is more likely exactly when  $q((1 - p)^2 + p^2) > 2p(1 - p)(1 - q)$ , i.e.,  $q > 2p(1 - p)$ . The two decisions disagree precisely when  $p < q < 2p(1 - p)$ , e.g., for  $(p, q) = (0.2, 0.3)$ .

**Problem 4.4.:** The syndromes with a unique minimum-weight correction are  $0000 \mapsto \emptyset$ ,  $0001 \mapsto \{3\}$ ,  $0010 \mapsto \{7\}$ ,  $0011 \mapsto \{3, 7\}$ ,  $0101 \mapsto \{6\}$ ,  $0111 \mapsto \{6, 7\}$ ,  $1010 \mapsto \{4\}$ ,  $1011 \mapsto \{3, 4\}$ ,  $1100 \mapsto \{5\}$ ,  $1111 \mapsto \{4, 6\}$ . The others have several:  $0100 (\{8\}, \{9\})$ ,  $1000 (\{1\}, \{2\})$ ,  $0110 (\{4, 5\}, \{7, 8\}, \{7, 9\})$ ,  $1001 (\{1, 3\}, \{2, 3\}, \{5, 6\})$ ,  $1101 (\{1, 6\}, \{2, 6\}, \{3, 5\})$ , and  $1110 (\{4, 8\}, \{4, 9\}, \{5, 7\})$ . In every case the alternatives meet the top row with the same parity, so they lie in the same logical class; for this small patch, tie-breaking never affects success.

**Problem 4.5.:** Call the pair of rows  $\{b, b + 1\}$  the *band*  $b$ . Qubit  $(r, c)$  lies in exactly one Z check of band  $r - 1$  (if  $r \geq 1$ ) and one of band  $r$  (if  $r \leq d - 2$ ), and the qubits  $(r, c)$ ,  $(r + 1, c)$  of a column lie in a common Z check of band  $r$  (a square, or a left or right boundary check at  $c = 0$  or  $c = d - 1$ ). Hence every edge of  $G_X$  between check vertices changes the band by one, the top (bottom) boundary vertex is reached only through a qubit of row 0 (row  $d - 1$ ), and the lowest  $(d + 1)/2$  qubits of a column form a path from the bottom boundary vertex to a single defect in band  $(d - 3)/2$ . That defect is at distance  $(d - 3)/2 + 1 = (d - 1)/2$  from the top boundary vertex and  $(d - 2) - (d - 3)/2 + 1 = (d + 1)/2$  from the bottom one, so matching joins it to the top by a path of length  $(d - 1)/2$ , which meets row 0 exactly once (in its last edge). The error does not meet row 0, so the residual meets the top row once and matching fails by Equation (4.1).

For  $d = 3$ , no error of weight at most one causes failure, by Theorem 2.6. Enumerating the 36 errors of weight two shows that exactly 18 cause failure, independently of tie-breaking by the previous solution. Hence  $P_L(3, p) = 18p^2(1 - p)^7 + O(p^3) = 18p^2 + O(p^3)$ .

**Problem 4.6.:** Squaring  $|F/N - q| \leq z\sqrt{q(1 - q)/N}$  gives the quadratic inequality  $(1 + z^2/N)q^2 - (2F/N + z^2/N)q + (F/N)^2 \leq 0$ , whose roots are the endpoints in Equation (4.5).

## Section 5.

**Problem 5.1.:** This is the computation in Example 5.3: both methods give  $41/122$ .

**Problem 5.2.:** Let  $a$  and  $b$  be the checks  $e_1 + e_2 = 1$  and  $e_2 + e_3 = 0$ . In the first round, Equation (5.4) gives  $m_{a \rightarrow 1}(1) = 0.8$ ,  $m_{a \rightarrow 2}(1) = 0.9$ ,  $m_{b \rightarrow 2}(1) = 0.3$ ,  $m_{b \rightarrow 3}(1) = 0.2$ , and the belief of  $e_2$  is already  $0.2 \cdot 0.9 \cdot 0.3 / (0.2 \cdot 0.9 \cdot 0.3 + 0.8 \cdot 0.1 \cdot 0.7) = 27/55$ . In the second round  $m_{2 \rightarrow a}(1) = 3/31$  and  $m_{2 \rightarrow b}(1) = 9/13$ , hence  $m_{a \rightarrow 1}(1) = 28/31$  and  $m_{b \rightarrow 3}(1) = 9/13$ , and the beliefs of  $e_1$  and  $e_3$  are  $28/55$  and  $27/55$ .

**Problem 5.3.:** By independence,  $\mathbb{E} \prod_i (-1)^{B_i} = \prod_i \mathbb{E} (-1)^{B_i} = \prod_i (1 - 2u_i)$ , and  $\mathbb{P}(\sum_i B_i \text{ odd}) = (1 - \mathbb{E}(-1)^{\sum_i B_i})/2$ . When  $e_j = 1$  the other variables must have parity  $s_a + 1$ , which gives Equation (5.4) in both cases  $s_a = 0, 1$ .

**Problem 5.4.:** For  $d = 3$ , the Tanner graph has  $4 + 9 = 13$  nodes and 12 edges (six columns of weight one and three of weight two) and is connected, hence a tree. For  $d = 5$ , the X square with qubits 6, 7, 11, 12 gives the cycle through the Z checks on  $\{1, 2, 6, 7\}$ ,  $\{7, 8, 12, 13\}$ ,  $\{11, 12, 16, 17\}$ ,  $\{6, 11\}$  and the qubits 7, 12, 11, 6, of length eight. A cycle of length four would require two Z checks sharing two qubits, i.e., two Z checks sharing an edge, and these do not exist.

**Problem 5.5.:** The columns 1 and 2 of  $H_Z$  coincide, so exchanging  $e_1$  and  $e_2$  preserves the posterior, and their marginals are equal (approximately 0.463 for  $p = 0.05$ ). Rounding gives  $e_1 = e_2 = 0$  together with 0 on the other coordinates, which is inconsistent with the syndrome; the symmetry that makes either choice correct is exactly what prevents the marginals from choosing.

**Problem 5.6.:** In units of  $\lambda$ , the serial sweep computes at check  $a$  the messages  $e_2 \rightarrow a = e_3 \rightarrow a = 1$  and  $a \rightarrow e_2 = a \rightarrow e_3 = -1$ ; at check  $b$ ,  $e_1 \rightarrow b = 1$  and  $e_3 \rightarrow b = 1 + (-1) + 0 = 0$ , hence  $b \rightarrow e_1 = 0$  and  $b \rightarrow e_3 = -1$ ; at check  $c$ ,  $e_1 \rightarrow c = 1$ ,  $e_2 \rightarrow c = 0$ ,  $e_3 \rightarrow c = -1$ , hence  $c \rightarrow e_1 = c \rightarrow e_3 = 0$  and  $c \rightarrow e_2 = +1$ . The beliefs are  $1 + 0 + 0 = 1$ ,  $1 - 1 + 1 = 1$  and  $1 - 1 - 1 + 0 = -1$ , so the hard decision is 001. With damping  $\gamma = 1/2$ , round 1 is as in Proposition 5.7; the damped variable-to-check messages are then  $1/2$  from  $e_1$  and  $e_2$  and 0 from  $e_3$ , so in round 2 the only nonzero check messages are  $a \rightarrow e_3 = b \rightarrow e_3 = c \rightarrow e_3 = -1/2$ , and the beliefs  $(1, 1, -1/2)$  give 001.

**Problem 5.7.:** The transposition  $\sigma$  fixes both rows 011 and 111, the priors, and the syndrome, so it is an automorphism of the Tanner graph that fixes every check node. The update rules are equivariant under such automorphisms, and the initial messages are invariant; by induction on the number of updates, every message from or to  $e_2$  equals the corresponding message from or to  $e_3$ , for any schedule that visits the checks (or variables) in a  $\sigma$ -invariant manner. Hence  $e_2$  and  $e_3$  have equal beliefs and equal hard decisions, and the first check,  $e_2 + e_3 = 1$ , is never satisfied.

## Section 6.

**Problem 6.1.:** In the row of  $H_X$  indexed by  $(a, j)$ , the block  $H_1 \otimes I_{n_2}$  has entry  $(H_1)_{a,j_1} \delta_{jj_2}$  in the column  $(j_1, j_2)$ , and the block  $I_{m_1} \otimes H_2^T$  has entry  $\delta_{aa_1} (H_2)_{a_2,j}$  in the column  $(a_1, a_2)$ ; the weight is therefore the weight of row  $a$  of  $H_1$  plus the weight of column  $j$  of  $H_2$ . Likewise, the column of  $H_X$  indexed by  $(j_1, j_2)$  has the weight of column  $j_1$  of  $H_1$ , and that indexed by  $(a_1, a_2)$  the weight of row  $a_2$  of  $H_2$ ; the rows and columns of  $H_Z$  are described symmetrically.

**Problem 6.2.:** The X check  $(a_1, j_2)$  acts on the qubits  $(j_1, j_2)$  with  $j_1 \in \{a_1, a_1 + 1\}$ , at the points  $(2a_1, 2j_2)$  and  $(2a_1 + 2, 2j_2)$  just above and below  $(2a_1 + 1, 2j_2)$ , and on the qubits  $(a_1, a_2)$  with  $a_2 \in \{j_2 - 1, j_2\}$ , at the points  $(2a_1 + 1, 2j_2 \pm 1)$  to its left and right that lie in the grid. The Z checks are described in the same way, with the roles of the coordinates exchanged. The code has  $9 + 4 = 13$  qubits, and  $k = 1$ ,  $d = 3$  by Theorem 6.3 and Example 6.4.

**Problem 6.3.:**  $xy = S_\ell \otimes S_m = yx$  by the mixed-product rule. The monomials in  $A$  are distinct elements of the group  $\langle x, y \rangle \cong \mathbb{Z}_6 \times \mathbb{Z}_6$ , so they are permutation matrices with disjoint supports, and  $A$  has three ones in each row and column; the same holds for  $B$ . Each row of  $[A \mid B]$  and of  $[B^T \mid A^T]$  therefore has weight six, and each column weight three.

**Problem 6.4.:** By Theorem 3.6,  $n = d^2$  and  $k = 1$ , so  $d = \sqrt{n}$  and  $kd^2 = n$ .

## Section 7.

**Problem 7.1.:** With  $e_3 = 0$  one obtains  $e = 100$ ; with  $e_3 = 1$ ,  $e = 011$ . Their masses are 0.056 and 0.054, so 100 is the more probable.

**Problem 7.2.:** Given  $u$ , Equation (7.1) determines  $e_B$ , and every solution arises from its own restriction to  $F$ ; this is a bijection between  $F_2^{n-\rho}$  and the solution set. An exhaustive search of order  $w$  performs  $\sum_{i=0}^w \binom{n-\rho}{i}$  solves.

**Problem 7.3.:** The columns of  $H_Z$  are  $(0, 1)$ ,  $(1, 1)$ ,  $(1, 1)$ , so any basis  $B$  contains column 1 and exactly one of the equal columns 2 and 3, whatever the reliability order. If  $B = \{1, 2\}$  and  $u = e_3$ , the solve gives

$e_2 = 1 + u$  and  $e_1 = 0$ , i.e., 010 or 001; the case  $B = \{1, 3\}$  is symmetric. Every output is consistent, and any two outputs differ by 011, i.e., by the operator  $X_2X_3$ , which acts trivially on the code if it is a stabilizer.

### Section 8.

**Problem 8.1.:** The differences are 01000 (one data error before round 2), 01100 (one recording error in round 2), and 00101, which requires two faults: either a data error before round 3 and another undoing it before round 5, or recording errors in rounds 3 and 4.

**Problem 8.2.:** Solving  $1 - P_{\text{shot}} = (1 - P_{\text{round}})^5$  gives  $P_{\text{round}} = 1 - (1 - P_{\text{shot}})^{1/5} \approx P_{\text{shot}}/5$ , i.e., approximately 0.068% for matching and 0.051% for BP+OSD. The formula treats the rounds as independent opportunities for one logical failure; it ignores both the correlations between consecutive rounds and the time boundaries, and it is meaningful only when the rates are small.

**Problem 8.3.:** With  $b = 401$  and  $c = 231$ , the statistic is  $170/\sqrt{632} \simeq 6.76$ . The totals 678 and 508 fix  $b - c = 170$  and force  $0 \leq c \leq 508$ , so  $b + c$  ranges from 170 to 1186 and the statistic from  $170/\sqrt{1186} \simeq 4.94$  to  $170/\sqrt{170} \simeq 13.0$ . In this example every consistent outcome favours the same decoder strongly, but the strength of the evidence depends on information that the totals do not contain.

**Problem 8.4.:** The four Z checks have deterministic outcomes from the first round, since the data qubits are prepared in the computational state  $|0\rangle$ ; each contributes  $T$  differences and one final comparison with the readout, i.e.,  $4(T + 1)$  detectors. The four X checks have random first outcomes and contribute  $T - 1$  differences each, so there are  $4(T + 1) + 4(T - 1) = 8T$  detectors in all.

### Section A.

**Problem A.1.:** Write  $r = (b_1, -b_2, -b_3)$  and  $r' = (b'_1, -b'_2, -b'_3)$ . Direct expansion gives  $|u\rangle\langle u| = (I + r_1X + r_2Y + r_3Z)/2$ , and similarly for  $v$ . The trace identities  $\text{Tr } P = 0$  and  $\text{Tr}(PQ) = 2\mathbf{1}\{P = Q\}$  for  $P, Q \in \{X, Y, Z\}$  give

$$|\langle u, v \rangle|^2 = \text{Tr}(|u\rangle\langle u||v\rangle\langle v|) = (1 + r \cdot r')/2 = (1 + b \cdot b')/2.$$

This vanishes precisely when  $b' = -b$ . A vector orthogonal to  $(z, w)$  is  $(-\bar{w}, \bar{z})$ , giving the coordinate  $-1/\bar{\beta}$ , including the exchange of 0 and  $\infty$ . Measuring the projections onto  $|v\rangle$  and its orthogonal line returns the antipodal points  $b'$  and  $-b'$  with probabilities  $(1 + b \cdot b')/2$  and  $(1 - b \cdot b')/2$ ; the post-measurement state is the corresponding endpoint.

**Problem A.2.:** Squaring a Hermitian Pauli representative shows that  $\chi(P) \in \{\pm 1\}$ , and conjugating a product shows that these signs define a character of  $\bar{\mathcal{P}}_n \cong \mathbb{F}_2^{2n}$ . By nondegeneracy of  $\omega$ , there is a Pauli operator  $R$  with  $\chi(P) = (-1)^{\omega([R], [P])}$  for every  $P$ , where the brackets are identified with binary coordinates. Thus  $R^{-1}A$  commutes with every Pauli matrix and hence with the full matrix algebra, so it is scalar and  $[A] = [R]$ . Conversely, all projective Paulis commute with one another. The kernel in Equation (A.8) is exactly this projective centralizer.

### APPENDIX D. NOTATION AND FURTHER READING

| Symbol or term                                   | Meaning                                                                                   |
|--------------------------------------------------|-------------------------------------------------------------------------------------------|
| $\mathcal{H}_n$                                  | The underlying Hilbert space $(\mathbb{C}^2)^{\otimes n}$ of $n$ qubits (Definition 1.1). |
| $\mathcal{P}_n,  E $                             | The Pauli matrix group of linear lifts and their weight (Definition 1.5).                 |
| $\mathbb{P}(\mathcal{H}_n), [u]$                 | The pure-state space and the state represented by a nonzero vector $u$ (Definition 1.1).  |
| $\beta, B$                                       | The affine coordinate $z/w$ and the Bloch sphere (Section A).                             |
| $\bar{\mathcal{P}}_n, \text{Cl}_n^{\text{proj}}$ | The projective Pauli group and its normalizer (Section A.4).                              |

|                                            |                                                                                                                                           |
|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| $S, g_j, \mathcal{C}(S), P$                | Signed stabilizer lifts, independent checks, the underlying code subspace, and its orthogonal projection ( <a href="#">Theorem 2.2</a> ). |
| $\mathbb{P}(\mathcal{C})$                  | The projective code with underlying subspace $\mathcal{C}$ ( <a href="#">Definition 1.4</a> ).                                            |
| $s(E)$                                     | The syndrome of a Pauli operator ( <a href="#">Proposition 2.3</a> ).                                                                     |
| $\mathcal{Z}(S), \langle iI \rangle S$     | The centralizer of $S$ in $\mathcal{P}_n$ , and $S$ with scalars adjoined ( <a href="#">Proposition 2.4</a> ).                            |
| $[[n, k, d]]$                              | Number of data qubits, number of encoded qubits, and distance ( <a href="#">Definition 2.5</a> ); auxiliary qubits are not counted.       |
| $X(x), Z(z), \omega$                       | Binary coordinates of Pauli operators and the symplectic form ( <a href="#">Proposition 2.8</a> ); $Y$ has both coordinates 1.            |
| $H_X, H_Z$                                 | Check matrices of a CSS code: rows are supports of $X$ and $Z$ checks ( <a href="#">Definition 2.9</a> ).                                 |
| $s_X, s_Z$                                 | Outcomes of the $X$ and $Z$ checks: $s_X = H_X z, s_Z = H_Z x$ .                                                                          |
| $\bar{X}, \bar{Z}$                         | Logical Pauli operators of an encoded qubit.                                                                                              |
| $z^{(i)}, x^{(j)}$                         | Indicators of row $i$ and column $j$ of the rotated patch ( <a href="#">Proposition 3.4</a> ).                                            |
| $G_X$                                      | The decoding graph of $X$ errors on the rotated patch ( <a href="#">Section 3.4</a> ).                                                    |
| $\hat{e}$                                  | A decoder's correction ( <a href="#">Definition 4.2</a> ).                                                                                |
| $w_j$                                      | The log-odds weight $\log((1 - p_j)/p_j)$ ( <a href="#">Equation 4.4</a> ).                                                               |
| $P_L(d, p)$                                | Logical failure probability of a specified decoder on the patch of distance $d$ ( <a href="#">Section 4.6</a> ).                          |
| $m_{j \rightarrow a}, L_{j \rightarrow a}$ | BP messages and their log-likelihood ratios; a positive LLR favours 0 ( <a href="#">Section 5</a> ).                                      |
| $\gamma$                                   | Damping factor of BP ( <a href="#">Definition 5.5</a> ).                                                                                  |
| $k_i^\top, d_i^\top$                       | Dimension and distance of $\ker H_i^\top$ in a hypergraph product ( <a href="#">Section 6.2</a> ).                                        |
| $D, O, f$                                  | Detector and observable matrices of a detector error model, and the fault vector ( <a href="#">Equation 8.1</a> ).                        |
| Shot                                       | One complete simulated experiment, possibly spanning many rounds.                                                                         |

**Further reading.** Gottesman's survey [[Got09](#)] and draft textbook [[Got26](#)] are the most complete mathematical accounts of stabilizer codes, and his thesis [[Got97](#)] is still worth reading for the original formulation; Roffe [[Rof19](#)], Sections 3–5, is a gentle companion with many examples. For surface codes, the original sources are Kitaev [[Kit03](#)] and Bravyi and Kitaev [[BK98](#)]; Dennis et al. [[DKLP02](#)] develop decoding, matching, and the connection with statistical mechanics, and Fowler et al. [[FMMC12](#)] treat implementation. For belief propagation in general, see Kschischang, Frey and Loeliger [[KFL01](#)] and, for its use on classical LDPC codes, MacKay [[Mac03](#)] (freely available online) and Richardson and Urbanke [[RU08](#)]. For belief propagation on quantum codes and its difficulties, see Poulin and Chung [[PC08](#)], Sections III–V, and Raveendran and Vasić [[RV21](#)]; for ordered-statistics post-processing, Panteleev and Kalachev [[PK21](#)], Section 3, and Roffe et al. [[RWBC20](#)]. Breuckmann and Eberhardt [[BE21b](#)] survey quantum LDPC codes, including hypergraph products [[TZ14](#)]; for bivariate bicycle codes and their measurement circuits, see Bravyi et al. [[BCG<sup>+</sup>24](#)], Sections 4–6. [Section 7.2](#) lists the primary sources for the main families of BP-based decoders. For the statistics of [Sections 4.6](#) and [8.4](#), Agresti and Coull [[AC98](#)] on binomial intervals and Dietterich [[Die98](#)] on paired comparisons are short and readable. The documentation of Stim [[Gid21](#)] and PyMatching [[Hig22](#)] is the best introduction to simulating these experiments in practice.

## REFERENCES

- [AC98] Alan Agresti and Brent A. Coull, *Approximate is better than “exact” for interval estimation of binomial proportions*, Amer. Statist. **52** (1998), no. 2, 119–126.
- [BCG<sup>+</sup>24] Sergey Bravyi, Andrew W. Cross, Jay M. Gambetta, Dmitri Maslov, Patrick Rall, and Theodore J. Yoder, *High-threshold and low-overhead fault-tolerant quantum memory*, Nature **627** (2024), 778–782.

- [BE21a] Nikolas P. Breuckmann and Jens Niklas Eberhardt, *Balanced product quantum codes*, IEEE Trans. Inform. Theory **67** (2021), no. 10, 6653–6674.
- [BE21b] ———, *Quantum low-density parity-check codes*, PRX Quantum **2** (2021), no. 4, 040101.
- [BK98] S. B. Bravyi and A. Yu. Kitaev, *Quantum codes on a lattice with boundary*, 1998, arXiv:quant-ph/9811052.
- [BMD07] H. Bombin and M. A. Martin-Delgado, *Optimal resources for topological two-dimensional stabilizer codes: Comparative study*, Phys. Rev. A **76** (2007), no. 1, 012305.
- [BMvT78] Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. van Tilborg, *On the inherent intractability of certain coding problems*, IEEE Trans. Inform. Theory **24** (1978), no. 3, 384–386.
- [BPT10] Sergey Bravyi, David Poulin, and Barbara Terhal, *Tradeoffs for reliable quantum information storage in 2D systems*, Phys. Rev. Lett. **104** (2010), no. 5, 050503.
- [BSV14] Sergey Bravyi, Martin Suchara, and Alexander Vargo, *Efficient algorithms for maximum likelihood decoding in the surface code*, Phys. Rev. A **90** (2014), no. 3, 032326.
- [CS96] A. R. Calderbank and Peter W. Shor, *Good quantum error-correcting codes exist*, Phys. Rev. A **54** (1996), no. 2, 1098–1105.
- [dCMS22] Julien du Crest, Mehdi Mhalla, and Valentin Savin, *Stabilizer inactivation for message-passing decoding of quantum LDPC codes*, arXiv:2205.06125, 2022.
- [Die98] Thomas G. Dietterich, *Approximate statistical tests for comparing supervised classification learning algorithms*, Neural Comput. **10** (1998), no. 7, 1895–1923.
- [DKLP02] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill, *Topological quantum memory*, J. Math. Phys. **43** (2002), no. 9, 4452–4505.
- [Edm65] Jack Edmonds, *Paths, trees, and flowers*, Canad. J. Math. **17** (1965), 449–467.
- [FGL18] Omar Fawzi, Antoine Grospellier, and Anthony Leverrier, *Constant overhead quantum fault-tolerance with quantum expander codes*, 2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS), 2018, pp. 743–754.
- [FL95] Marc P. C. Fossorier and Shu Lin, *Soft-decision decoding of linear block codes based on ordered statistics*, IEEE Trans. Inform. Theory **41** (1995), no. 5, 1379–1396.
- [FMMC12] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland, *Surface codes: Towards practical large-scale quantum computation*, Phys. Rev. A **86** (2012), no. 3, 032324.
- [Fow13] Austin G. Fowler, *Optimal complexity correction of correlated errors in the surface code*, arXiv:1310.0863, 2013.
- [Gal62] Robert G. Gallager, *Low-density parity-check codes*, IRE Trans. Inform. Theory **8** (1962), no. 1, 21–28.
- [Gid21] Craig Gidney, *Stim: a fast stabilizer circuit simulator*, Quantum **5** (2021), 497.
- [GNFB21] Craig Gidney, Michael Newman, Austin Fowler, and Michael Broughton, *A fault-tolerant honeycomb memory*, Quantum **5** (2021), 605.
- [Got97] D. Gottesman, *Stabilizer codes and quantum error correction*, Ph.D. thesis, California Institute of Technology, 1997, arXiv:quant-ph/9705052.
- [Got09] Daniel Gottesman, *An introduction to quantum error correction and fault-tolerant quantum computation*, 2009, arXiv:0904.2557.
- [Got26] ———, *Surviving as a quantum computer in a classical world*, 2026, Draft textbook (2026 draft).
- [HBQ<sup>+</sup>25] Timo Hillmann, Lucas Berent, Armanda O. Quintavalle, Jens Eisert, Robert Wille, and Joschka Roffe, *Localized statistics decoding for quantum low-density parity-check codes*, Nat. Commun. **16** (2025), 8214.
- [HG25] Oscar Higgott and Craig Gidney, *Sparse Blossom: correcting a million errors per core second with minimum-weight matching*, Quantum **9** (2025), 1600.
- [HHO21] Matthew B. Hastings, Jeongwan Haah, and Ryan O’Donnell, *Fiber bundle codes: breaking the  $n^{1/2}$  polylog( $n$ ) barrier for quantum LDPC codes*, STOC ’21: Proc. 53rd Annual ACM SIGACT Symposium on Theory of Computing (New York), ACM, 2021, pp. 1276–1288.
- [Hig22] Oscar Higgott, *PyMatching: A Python package for decoding quantum codes with minimum-weight perfect matching*, ACM Trans. Quantum Comput. **3** (2022), no. 3, 16:1–16:16.
- [HLG11] Min-Hsiu Hsieh and François Le Gall, *NP-hardness of decoding quantum error-correction codes*, Phys. Rev. A **83** (2011), no. 5, 052331.
- [IP15] Pavithran Iyer and David Poulin, *Hardness of decoding quantum stabilizer codes*, IEEE Trans. Inform. Theory **61** (2015), no. 9, 5209–5223.
- [KFL01] Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger, *Factor graphs and the sum-product algorithm*, IEEE Trans. Inform. Theory **47** (2001), no. 2, 498–519.
- [Kit03] A. Yu. Kitaev, *Fault-tolerant quantum computation by anyons*, Ann. Physics **303** (2003), no. 1, 2–30.
- [KL97] Emanuel Knill and Raymond Laflamme, *Theory of quantum error-correcting codes*, Phys. Rev. A **55** (1997), no. 2, 900–911.
- [KL22] Kao-Yueh Kuo and Ching-Yi Lai, *Exploiting degeneracy in belief propagation decoding of quantum codes*, npj Quantum Inf. **8** (2022), 111.

- [KP13] Alexey A. Kovalev and Leonid P. Pryadko, *Fault tolerance of quantum low-density parity check codes with sublinear distance scaling*, Phys. Rev. A **87** (2013), no. 2, 020304.
- [MAB<sup>+</sup>25] Tristan Müller, Thomas Alexander, Michael E. Beverland, Markus Böhler, Blake R. Johnson, Thilo Maurer, and Drew Vandeth, *Improved belief propagation is sufficient for real-time decoding of quantum memory*, arXiv:2506.01779, 2025.
- [Mac03] David J. C. MacKay, *Information theory, inference, and learning algorithms*, Cambridge University Press, Cambridge, 2003.
- [McN47] Quinn McNemar, *Note on the sampling error of the difference between correlated proportions or percentages*, Psychometrika **12** (1947), no. 2, 153–157.
- [MWJ99] Kevin P. Murphy, Yair Weiss, and Michael I. Jordan, *Loopy belief propagation for approximate inference: an empirical study*, Proc. 15th Conference on Uncertainty in Artificial Intelligence (UAI 1999); arXiv:1301.6725, 1999.
- [NC10] Michael A. Nielsen and Isaac L. Chuang, *Quantum computation and quantum information*, 10th anniversary ed., Cambridge University Press, Cambridge, 2010.
- [PC08] David Poulin and Yeojin Chung, *On the iterative decoding of sparse quantum codes*, Quantum Information and Computation **8** (2008), no. 10, 987–1000.
- [Pea88] Judea Pearl, *Probabilistic reasoning in intelligent systems: Networks of plausible inference*, Morgan Kaufmann, San Mateo, CA, 1988.
- [PK21] Pavel Panteleev and Gleb Kalachev, *Degenerate quantum LDPC codes with good finite length performance*, Quantum **5** (2021), 585.
- [PK22a] ———, *Asymptotically good quantum and locally testable classical LDPC codes*, STOC '22: Proc. 54th Annual ACM Symposium on Theory of Computing (New York), ACM, 2022, pp. 375–388.
- [PK22b] ———, *Quantum LDPC codes with almost linear minimum distance*, IEEE Trans. Inform. Theory **68** (2022), no. 1, 213–229.
- [Rof19] Joschka Roffe, *Quantum error correction: an introductory guide*, Contemporary Physics **60** (2019), no. 3, 226–245.
- [RU08] Tom Richardson and Rüdiger Urbanke, *Modern coding theory*, Cambridge University Press, Cambridge, 2008.
- [RV21] Nithin Raveendran and Bane Vasić, *Trapping sets of quantum LDPC codes*, Quantum **5** (2021), 562.
- [RWBC20] Joschka Roffe, David R. White, Simon Burton, and Earl Campbell, *Decoding across the quantum low-density parity-check code landscape*, Physical Review Research **2** (2020), 043423.
- [Sho95] Peter W. Shor, *Scheme for reducing decoherence in quantum computer memory*, Phys. Rev. A **52** (1995), no. 4, R2493–R2496.
- [SLG07] Eran Sharon, Simon Litsyn, and Jacob Goldberger, *Efficient serial message-passing schedules for LDPC decoding*, IEEE Trans. Inform. Theory **53** (2007), no. 11, 4076–4091.
- [Ste96] Andrew M. Steane, *Multiple-particle interference and quantum error correction*, Proc. Roy. Soc. London Ser. A **452** (1996), no. 1954, 2551–2577.
- [TS14] Yu Tomita and Krysta M. Svore, *Low-distance surface codes under realistic quantum noise*, Phys. Rev. A **90** (2014), no. 6, 062320.
- [TZ14] Jean-Pierre Tillich and Gilles Zémor, *Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength*, IEEE Trans. Inform. Theory **60** (2014), no. 2, 1193–1202.
- [Var97] Alexander Vardy, *The intractability of computing the minimum distance of a code*, IEEE Trans. Inform. Theory **43** (1997), no. 6, 1757–1766.
- [WB24] Stasiu Wolanski and Ben Barber, *Ambiguity clustering: an accurate and efficient decoder for qLDPC codes*, arXiv:2406.14527; abstract in Proc. IEEE QCE 2024, 402–403, 2024.
- [WHP03] Chenyang Wang, Jim Harrington, and John Preskill, *Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory*, Ann. Physics **303** (2003), no. 1, 31–58.
- [Wil27] Edwin B. Wilson, *Probable inference, the law of succession, and statistical inference*, J. Amer. Statist. Assoc. **22** (1927), no. 158, 209–212.
- [YALH<sup>+</sup>24] Hanwen Yao, Waleed Abu Laban, Christian Häger, Alexandre Graell i Amat, and Henry D. Pfister, *Belief propagation decoding of quantum LDPC codes with guided decimation*, 2024 IEEE International Symposium on Information Theory (ISIT), 2024, pp. 2478–2483.