Fault-Tolerant Quantum Computing
by
Code Deformation

Sergey Bravyi

IBM Watson Center

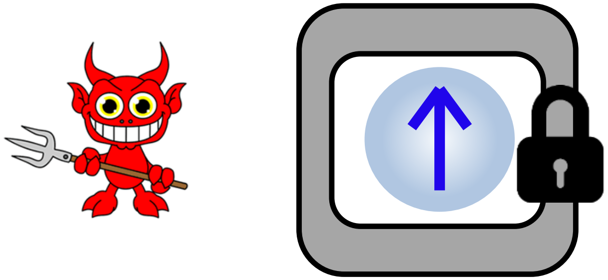
QIP Tutorial
2016

Fault-tolerant quantum computing

1. How to implement a reliable quantum memory.
   Need decoherence time larger than the runtime of a quantum algorithm.

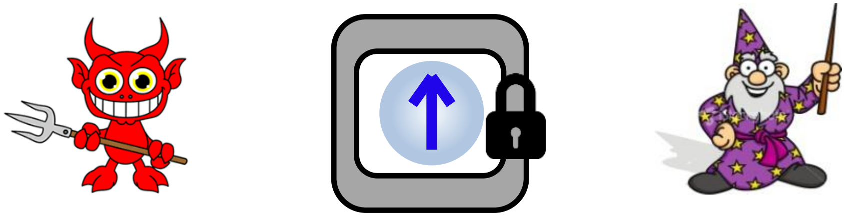
2. How to implement reliable logical gates.
   Need precision smaller than the inverse number of gates in a quantum algorithm.

Solution: quantum error correction



Fault-tolerant quantum computing

How to implement operations on the logical qubits encoded by a quantum code without exposing them to the environment ?

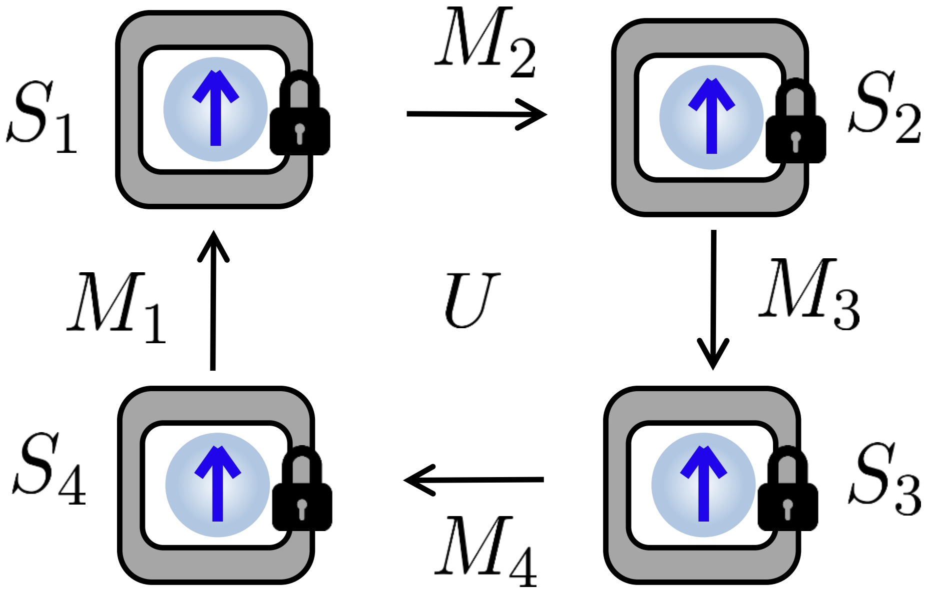


Exploit code symmetries. Certain logical operations can be implemented transversally by acting independently on each physical qubit.

Bad news: transversal logical gates cannot be universal [Eastin and Knill 2009]

Solution 1: Code Deformation

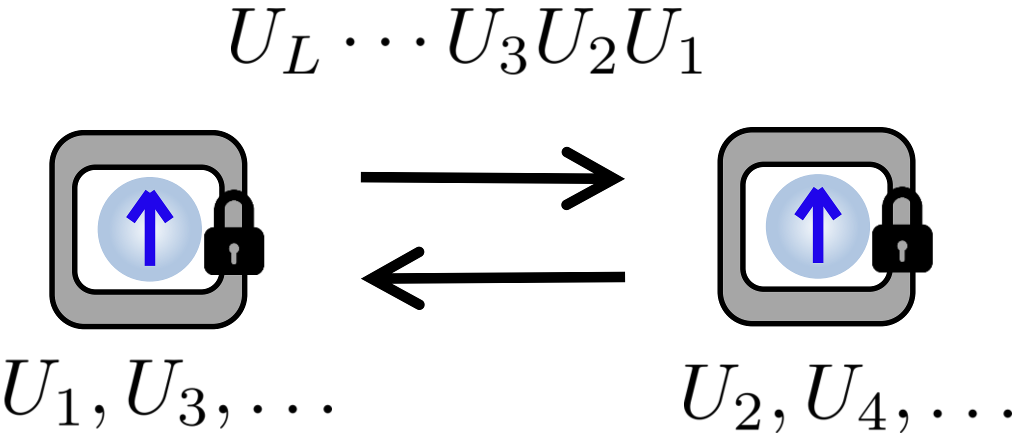
Logical gates are implemented by traversing a closed loop in the space of codes. The codes share same physical qubits but may have different parity checks.



Kitaev (1997); Raussendorf, Harrington, Goyal (2007)
Fowler, Stephens, Groszkowski (2008)
Horsman et al (2011); Gottesman and Zhang (2013);
Landahl and Ryan-Anderson (2014)

Solution 2: Gauge Fixing Method

Transfer logical qubits between two or more codes. Universality is achieved by combing transversal logical gates of different codes.



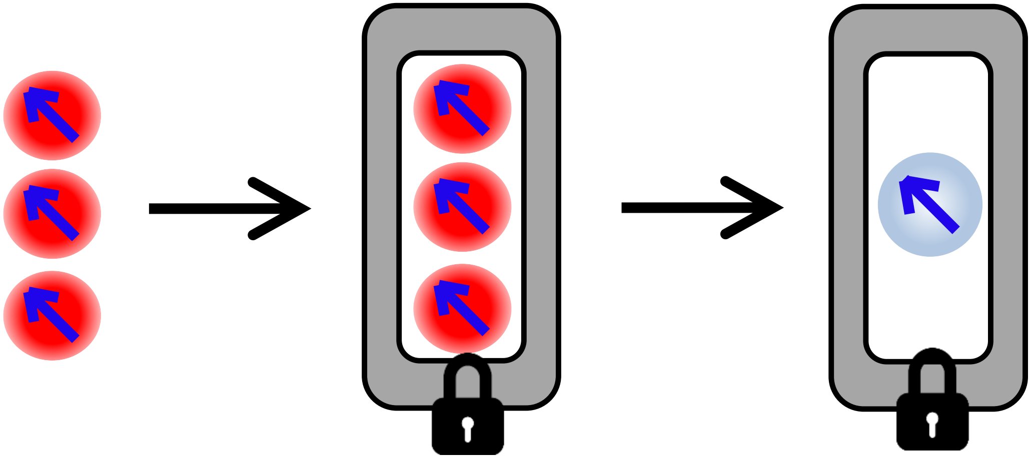
$$U_L \cdots U_3 U_2 U_1$$

$$U_1, U_3, \ldots \qquad U_2, U_4, \ldots$$

Paetznick and Reichardt (2013);
Jochym-O'Connor and Laflamme (2013); Bombin (2013-2014)
Anderson, Duclos-Cianci, Poulin (2014); SB and Cross (2015);
Jochym-O'Connor and Bartlett (2015);
Jones, Brooks, Harrington (2015);

Solution 3: Magic state distillation*

Prepare unprotected ancillary qubits in a desired state.
Inject the noisy ancillas into the codespace.
Use protected logical gates and measurements to distill a few noiseless ancillas from many noisy ones.



SB and Kitaev (2004); Reichardt (2004);
Meier, Eastin, and Knill (2012); SB and Haah (2012)
Jones (2012); Fowler, Devitt, and Jones (2012)

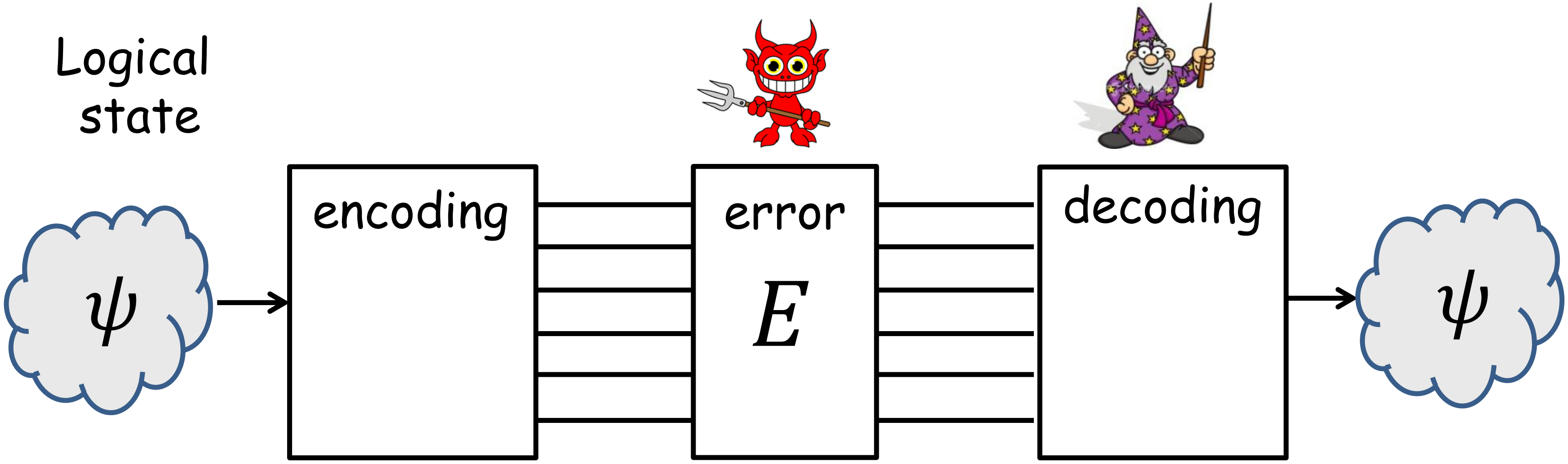
---

* Very large overhead.

Outline

- Stabilizer codes
- The decoding problem and code distance
- Fault-tolerant code deformation
- Example 1: Shor's 4-qubit code
- Example 2: lattice surgery
- Maximum likelihood decoding

Quantum Error Correction



Logical state

ψ

encoding

error

E

decoding

ψ

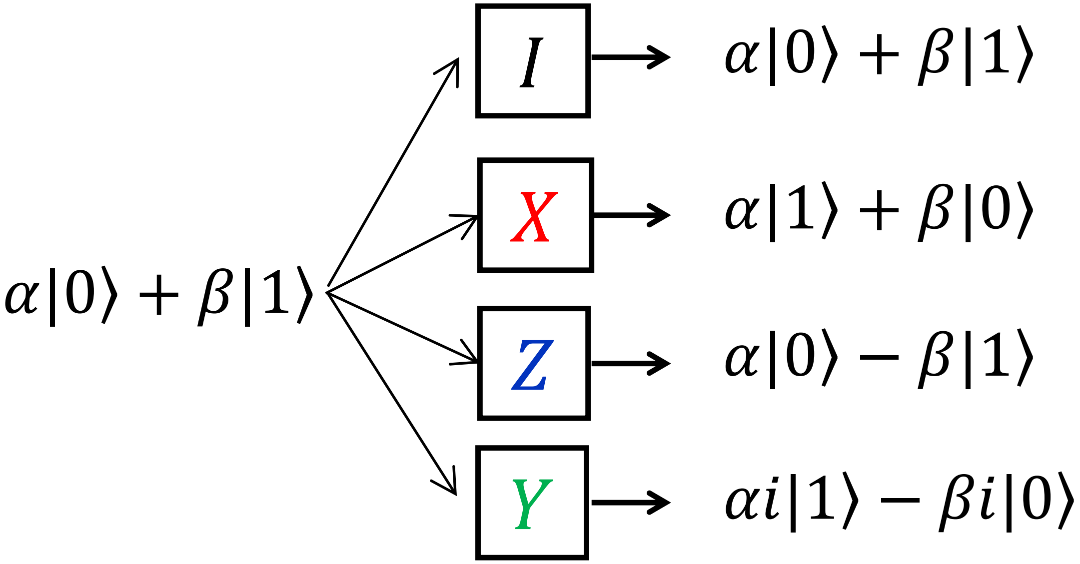
Pauli errors:

Perfect transmission I

Bit flip X

Phase flip Z

Bit and phase flip Y



$\alpha|0\rangle + \beta|1\rangle$

$\boxed{I} \rightarrow \alpha|0\rangle + \beta|1\rangle$

$\boxed{X} \rightarrow \alpha|1\rangle + \beta|0\rangle$

$\boxed{Z} \rightarrow \alpha|0\rangle - \beta|1\rangle$

$\boxed{Y} \rightarrow \alpha i|1\rangle - \beta i|0\rangle$

More on Pauli errors

1. Pauli operators either commute or anti-commute

$$\begin{array}{cccccccccc} P = & X & \otimes & X & \otimes & Z & \otimes & Z & \otimes & I \\ Q = & Z & \otimes & X & \otimes & I & \otimes & X & \otimes & I \\ & -1 & & & & & & -1 & & \end{array}$$

$$PQ = (-1)^2 QP = QP$$

2. Pauli operators have eigenvalues ± 1.
   Applying the same Pauli twice does nothing.

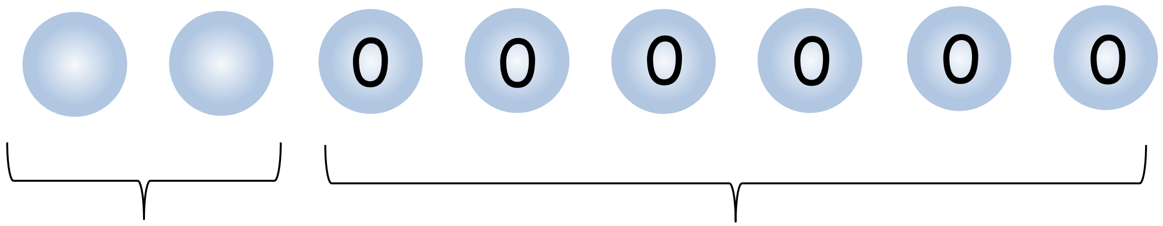
3. Pauli operators on n qubits form a group

$$\mathrm{Pauli}(n) = \mathbb{Z}_4 \rtimes \mathbb{Z}_2^{2n}$$

$$X \leftrightarrow (01), \quad Z \leftrightarrow (10), \quad Y \leftrightarrow (11)$$

4. Weight of a Pauli operator: $|P| = \#\{a \,:\, P_a \neq I\}$

Dummy [n,k] stabilizer code



k logical qubits

$n-k$ syndrome qubits

Logical-Z operators $Z_1, \ldots, Z_k$	**Stabilizers** $Z_{k+1}, \ldots, Z_n$
Logical-X operators $X_1, \ldots, X_k$	**Destabilizers** $X_{k+1}, \ldots, X_n$

Dummy [n,k] stabilizer code

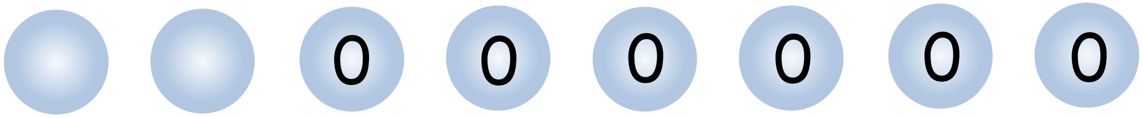
Codespace: a linear subspace spanned by n-qubit states that are +1 eigenvectors of any stabilizer

$$\mathcal{Q} = \underbrace{\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2}_{k} \otimes |0^{n-k}\rangle$$

Logical-Z operators $Z_1, \ldots, Z_k$	Stabilizers $Z_{k+1}, \ldots, Z_n$
Logical-X operators $X_1, \ldots, X_k$	Destabilizers $X_{k+1}, \ldots, X_n$

General [n,k] stabilizer code

$U \cdot$



$$UU^\dagger = I$$

Restriction: U must map Pauli to Pauli.

$$\overline{Z}_a = U Z_a U^\dagger \qquad \overline{X}_a = U X_a U^\dagger$$

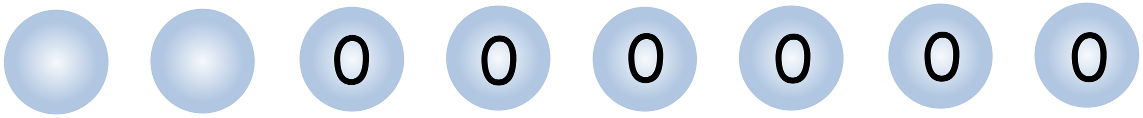
$$\overline{X}_a, \overline{Z}_a \in \mathrm{Pauli}(n)$$

Logical-Z operators $\overline{Z}_1, \dots, \overline{Z}_k$	Stabilizers $\overline{Z}_{k+1}, \dots, \overline{Z}_n$
Logical-X operators $\overline{X}_1, \dots, \overline{X}_k$	Destabilizers $\overline{X}_{k+1}, \dots, \overline{X}_n$

The dummy code in a rotated basis

General [n,k] stabilizer code

$U \cdot$



$UU^\dagger = I$

Restriction: U must map Pauli to Pauli.

$$\overline{Z}_a = U Z_a U^\dagger \qquad \overline{X}_a = U X_a U^\dagger$$

$$\overline{X}_a, \overline{Z}_a \in \mathrm{Pauli}(n)$$

Logical-Z operators $\overline{Z}_1, \ldots, \overline{Z}_k$	Stabilizers $S_1, \ldots, S_{n-k}$
Logical-X operators $\overline{X}_1, \ldots, \overline{X}_k$	Destabilizers $D_1, \ldots, D_{n-k}$

The dummy code in a rotated basis

General [n,k] stabilizer code

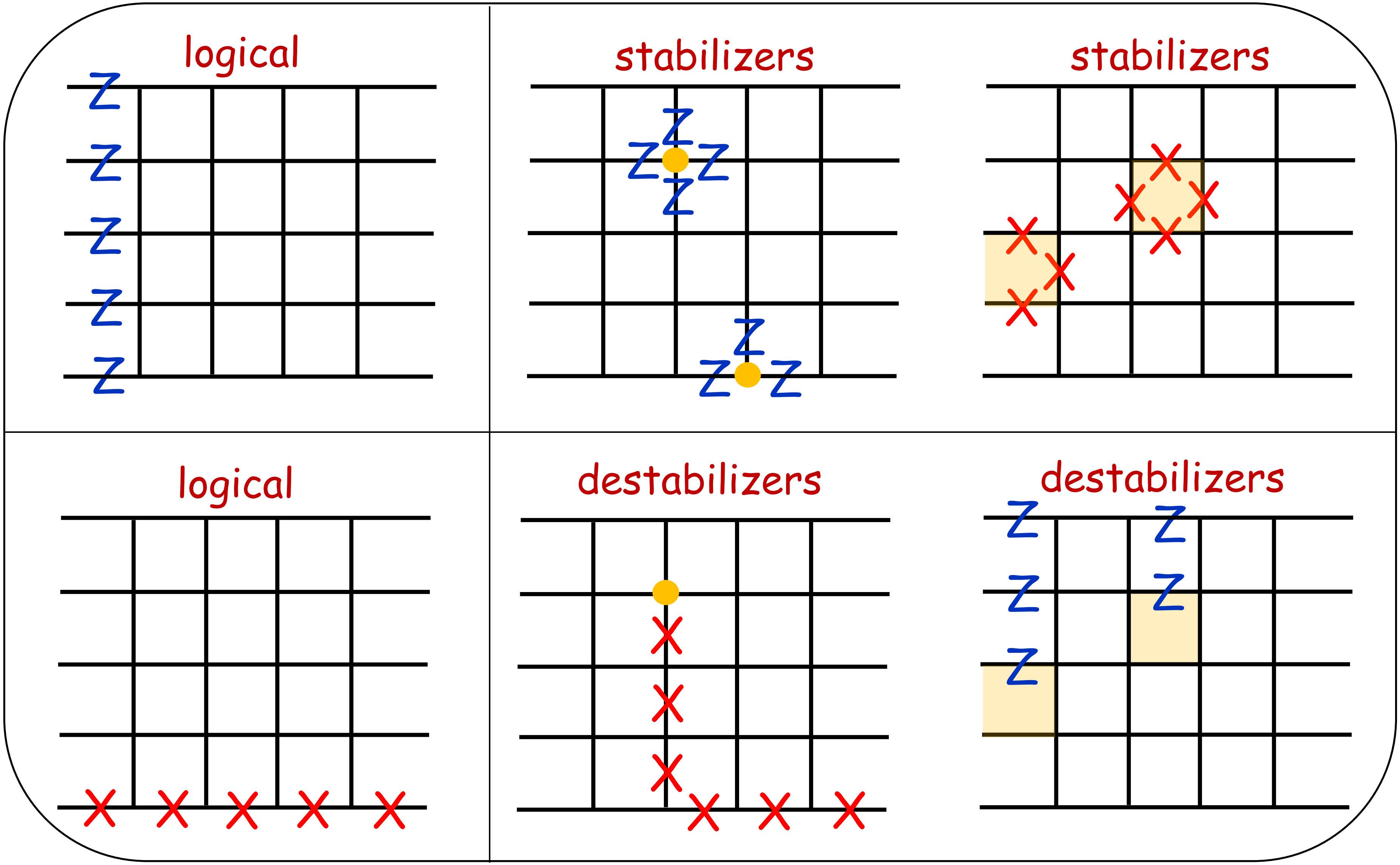
Codespace: $\mathcal{Q} = \{\psi \in (\mathbb{C}^2)^{\otimes n} \; : \; S_a \psi = \psi \quad \forall a\}$

$$\dim(\mathcal{Q}) = 2^k$$

Logical-Z operators $\overline{Z}_1, \ldots, \overline{Z}_k$	Stabilizers $S_1, \ldots, S_{n-k}$
Logical-X operators $\overline{X}_1, \ldots, \overline{X}_k$	Destabilizers $D_1, \ldots, D_{n-k}$

The dummy code in a rotated basis

Example: surface codes



Kitaev (1997), SB and Kitaev (1998), Freedman and Meyer (1998)

Stabilizer group: $\mathcal{S} = \langle S_1, \ldots, S_{n-k} \rangle$

Any Pauli error E admits a unique decomposition

$$E = \underset{\text{destabilizer}}{D} \cdot \underset{\text{logical}}{L} \cdot \underset{\text{stabilizer}}{S}$$

Logical-Z operators $\overline{Z}_1, \ldots, \overline{Z}_k$	Stabilizers $S_1, \ldots, S_{n-k}$
Logical-X operators $\overline{X}_1, \ldots, \overline{X}_k$	Destabilizers $D_1, \ldots, D_{n-k}$

The dummy code in a rotated basis

Error model

Random Pauli errors:

$$\rho \rightarrow \sum_E \Pr(E) E \rho E^\dagger$$

Standard toy model: depolarizing i.i.d. noise:

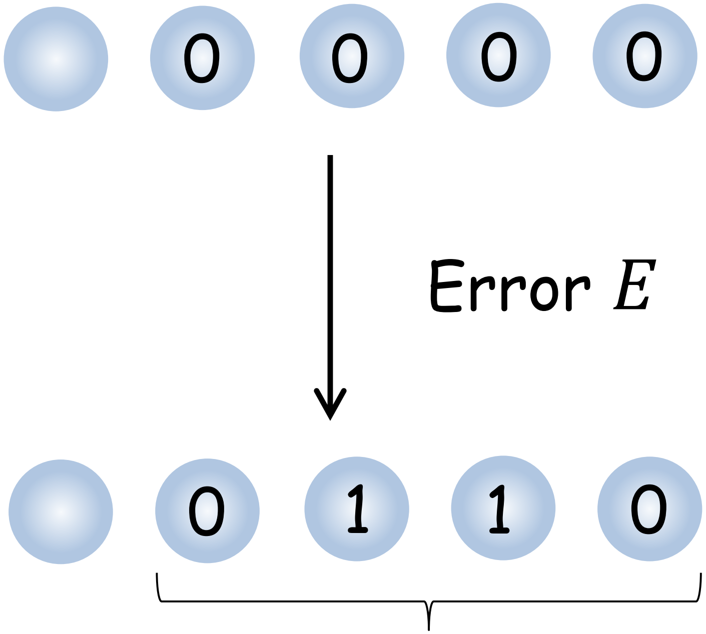
$$\Pr(X) = \Pr(Y) = \Pr(Z) = \epsilon/3$$

$$\Pr(I) = 1 - \epsilon$$

ϵ - error rate

Errors on different qubits are independent

Syndrome measurement (dummy code)



Encoded state $\psi \in \mathcal{Q}$

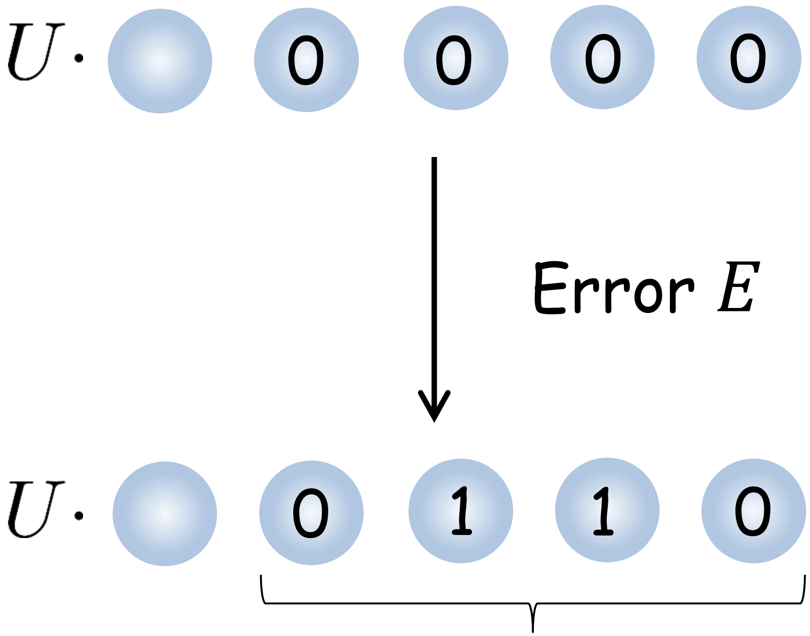
Error E $I \otimes I \otimes X \otimes Y \otimes Z$

Corrupted state $E\psi$

Error syndrome

- Eigenvalue measurement of stabilizers $Z_{k+1}, \ldots, Z_n$ reveals the state of syndrome qubits (error syndrome)
- Error syndrome determines whether the error commutes or anti-commutes with each stabilizer

Syndrome measurement (general code)



Encoded state $\psi \in \mathcal{Q}$

Error E $I \otimes I \otimes X \otimes Y \otimes Z$

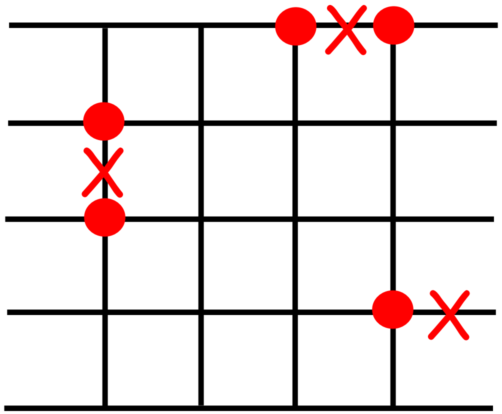
Corrupted state $E\psi$

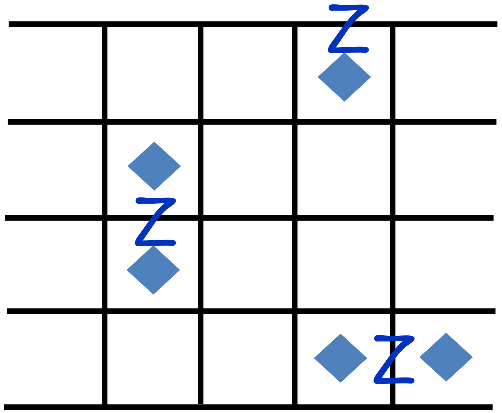
Error syndrome

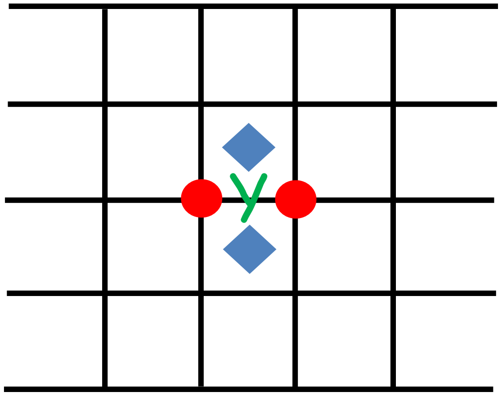
- Eigenvalue measurement of stabilizers $S_1, \ldots, S_{n-k}$ reveals the state of syndrome qubits (error syndrome)

- Error syndrome determines whether the error commutes or anti-commutes with each stabilizer

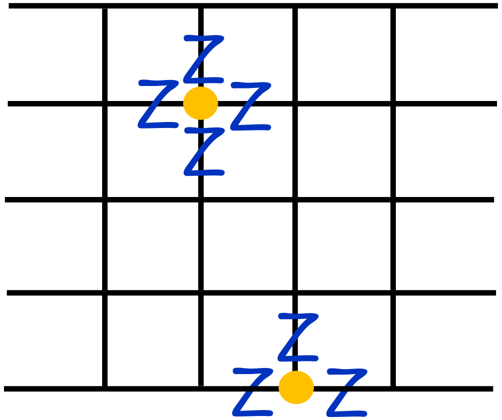
Surface code: errors vs syndromes



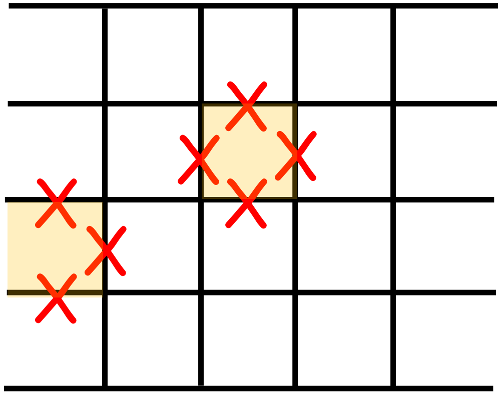




stabilizers



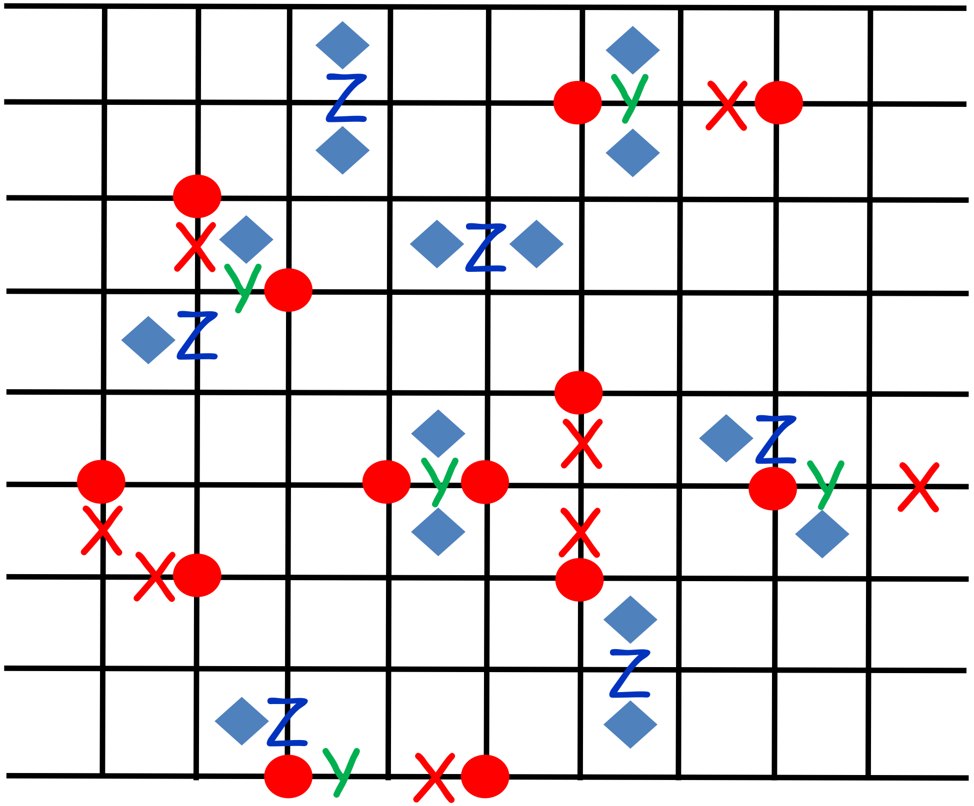
stabilizers



Outline

- Stabilizer codes
- The decoding problem and code distance
- Fault-tolerant code deformation
- Example 1: Shor's 4-qubit code
- Example 2: lattice surgery
- Maximum likelihood decoding

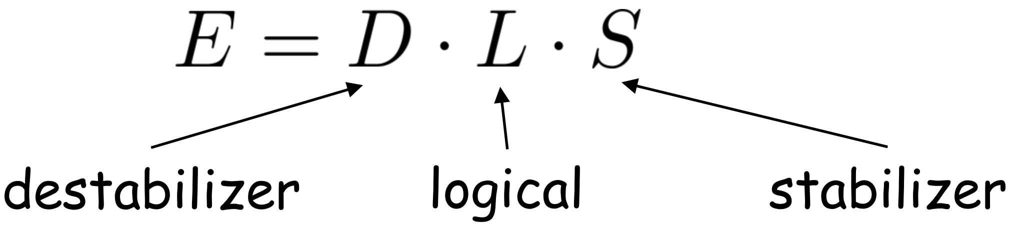
$\epsilon = 10\%$



Decoding problem

Given the error rate and the error syndrome.
Correct the error.

Reminder: Pauli errors E admit a unique decomposition



$$E = D \cdot L \cdot S$$

destabilizer

logical

stabilizer

Syndrome of E determines the destabilizer part D

Is this enough information to correct the error ?

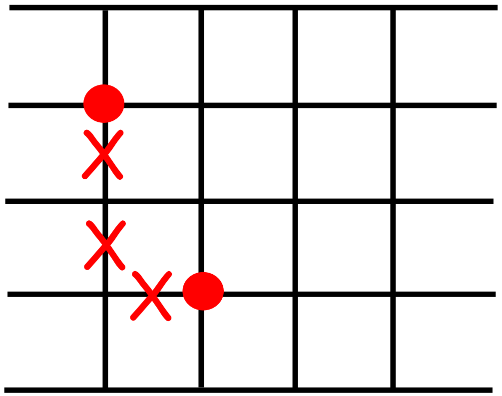
$$E = D \cdot L \cdot S \qquad\qquad E' = D \cdot L' \cdot S'$$

$$\boxed{E\psi = E'\psi \ \ \forall\, \psi \in Q \quad \text{iff} \quad L = L'}$$

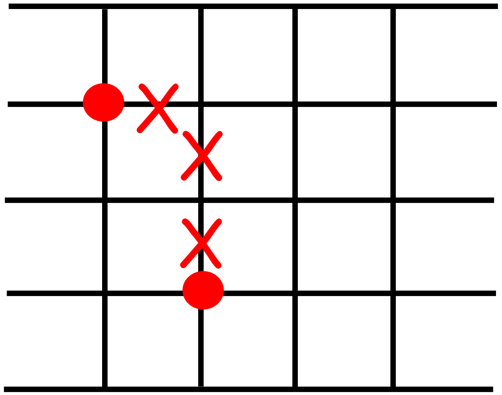
Decoder has to guess the logical part of the error

The stabilizer part does not matter

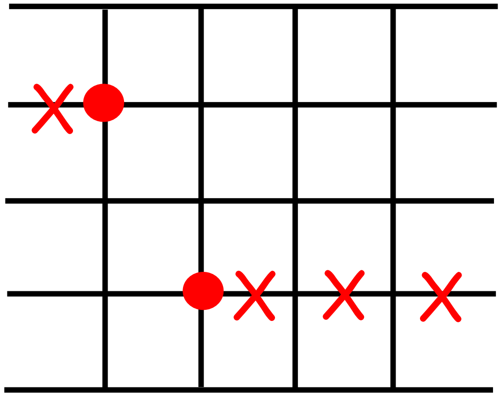
Two errors have the same logical part iff they have same commutation rules with the logical operators:



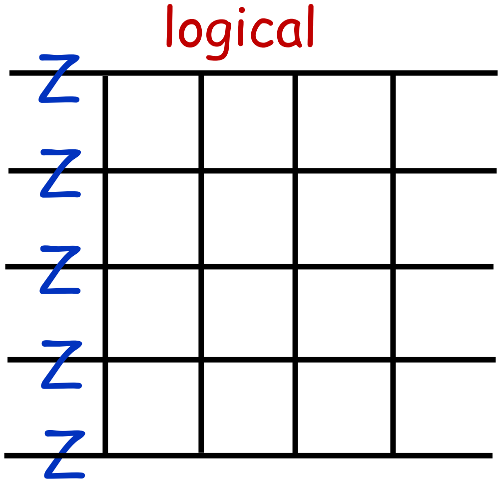
E



E'

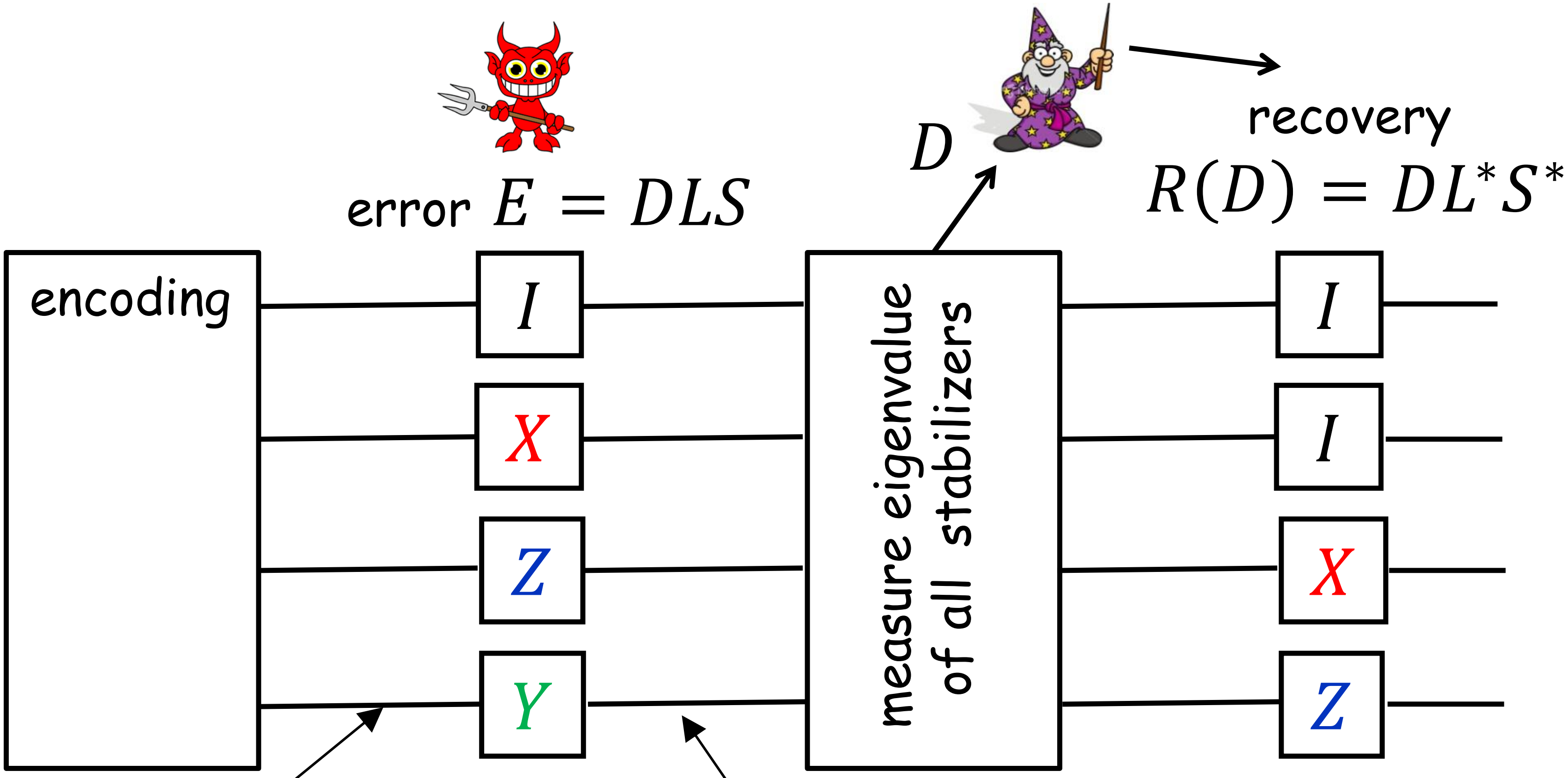


E''



$$L = L' \neq L''$$

Decoding: syndrome measurement + recovery



error $E = DLS$

D

recovery

$R(D) = DL^*S^*$

encoding

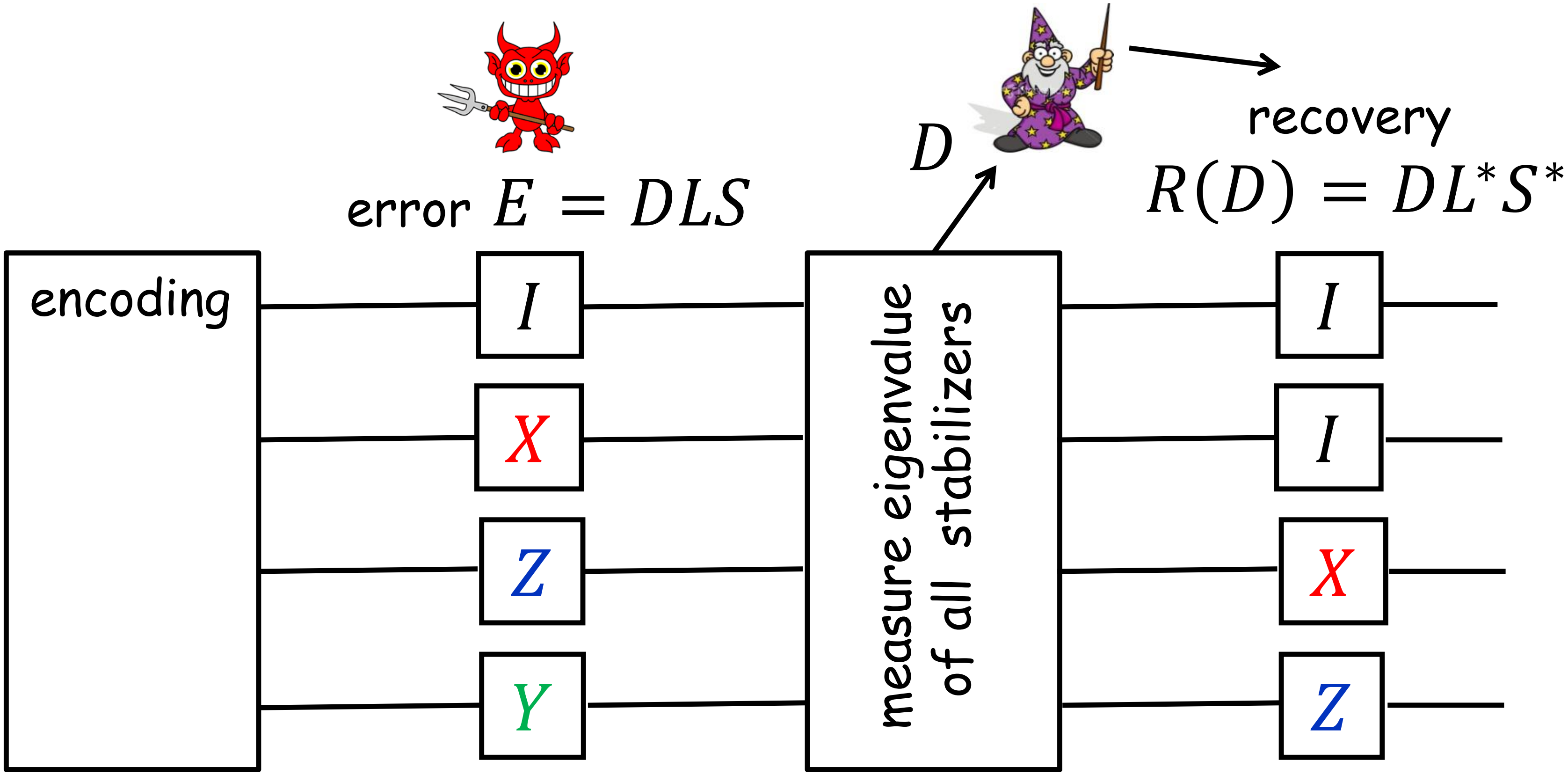
measure eigenvalue of all stabilizers

$S_a = 1$ for all a

syndrome determines D

$$S_a = 1 \quad \text{if} \quad S_aE = ES_a$$
$$S_a = -1 \quad \text{if} \quad S_aE = -ES_a$$

Decoding: syndrome measurement + recovery



Decoding succeeds if the error E and the recovery R differ by a product of stabilizers:

$$\boxed{E \cdot R \in S} \quad \Leftrightarrow \quad \boxed{L = L^*}$$

How to choose the recovery ?

Minimum Weight Decoder (MWD): pick an error with the minimum weight consistent with the syndrome

$$(L^*, S^*) = \arg\min_{L,S} |D \cdot L \cdot S|$$

Recovery: $R(D) = D \cdot L^* \cdot S^*$

Depolarizing i.i.d. errors: min weight = max probability

$$\Pr(E) = (1-\epsilon)^{n-|E|}(\epsilon/3)^{|E|} \sim \left(\frac{\epsilon}{3(1-\epsilon)}\right)^{|E|}$$

How to choose the recovery ?

Minimum Weight Decoder (MWD): pick an error with the minimum weight consistent with the syndrome

$$(L^*, S^*) = \arg\min_{L,S} |D \cdot L \cdot S|$$

Recovery: $R(D) = D \cdot L^* \cdot S^*$

Computing a min-weight recovery is NP-hard problem for classical linear codes.

Berlekamp, McEliece, van Tilborg (1978)

The same is true for quantum stabilizer codes since any classical linear code is a stabilizer code.

Code distance

Minimum weight of a Pauli error which has zero syndrome and has non-trivial logical part:

$$\boxed{d = \min_{S} \min_{L \neq I} |L \cdot S|}$$

$$[[n, k, d]]$$

Half-distance: $t = (d-1)/2$

Min-Weight decoder corrects any error of weight $\le t$

error: $E = D \cdot L \cdot S$ $\quad |E| \le t$

recovery: $R = D \cdot L^* \cdot S^*$ $\quad |R| \le |E| \le t$

$$|R \cdot E| = |LL^* \cdot SS^*| \le 2t < d \quad \Rightarrow \quad L = L^*$$

Code distance

Minimum weight of a Pauli error which has zero syndrome and has non-trivial logical part:

$$\boxed{d = \min_{S} \min_{L \neq I} |L \cdot S|}$$

$$[[n, k, d]]$$

Computing the distance of a given stabilizer code is NP-hard problem

Berlekamp et al (1978); Vardy (1997)
Tillich and Zemor (2009)

Idea: map a classical linear code to a quantum stabilizer code (hypergraph product construction)

$$[n, k, d] \quad \rightarrow \quad [[n^2 + (n-k)^2, k^2, d]]$$

Code distance

Minimum weight of a Pauli error which has zero syndrome and has non-trivial logical part:

$$\boxed{d = \min_{S} \min_{L \neq I} |L \cdot S|}$$

$$[[n, k, d]]$$

Exercise: Given a stabilizer code on n qubits and integer $t = O(1)$. One needs to decide whether

$$d \geq 2t + 1$$

Construct an algorithm for this task with a runtime

$$O(n^{t+1} \log{(n)})$$

Important special case: “homological CSS codes”

1. Each stabilizer is composed of X only or Z only

$X \otimes X \otimes I \otimes X \otimes I$

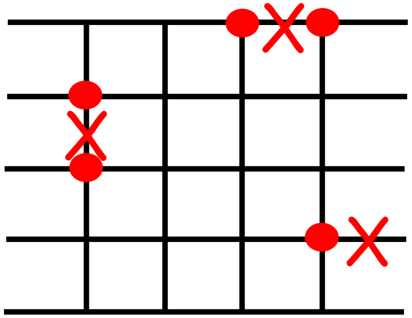
$Z \otimes Z \otimes I \otimes Z \otimes I$

OK

$X \otimes Z \otimes I \otimes X \otimes I$

forbidden

2. Any single-qubit X or Z error creates at most two non-zero syndromes



Important special case: "homological CSS codes"

1. Each stabilizer is composed of X only or Z only

$X \otimes X \otimes I \otimes X \otimes I$ $\qquad$ $X \otimes Z \otimes I \otimes X \otimes I$

$Z \otimes Z \otimes I \otimes Z \otimes I$ $\qquad$ forbidden

OK

2. Any single-qubit X or Z error creates at most two non-zero syndromes

Lemma. The distance of any homological CSS code on n qubits can be computed in time

$$O(kn^2 + n^2 \log(n))$$

Sketch of the proof

Claim 1: it suffices to consider errors composed of X only or Z only.

$$d = \min\{d^X, d^Z\}$$

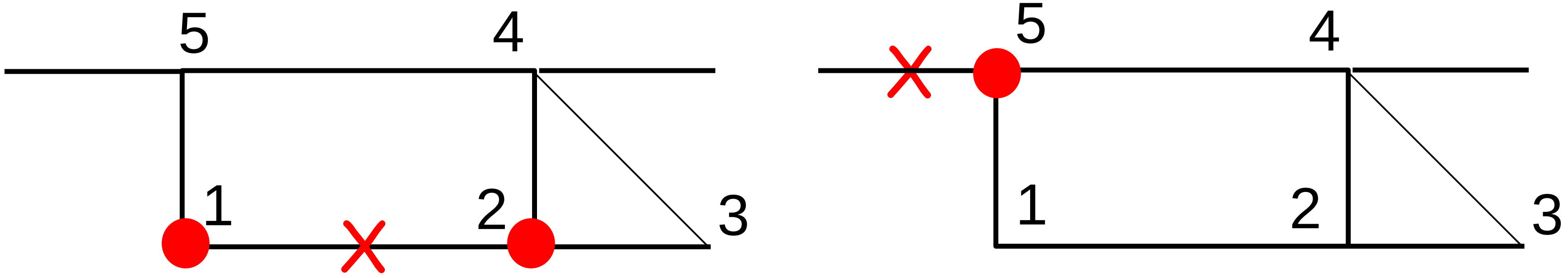
$$d^X = \min |E| \qquad \text{subject to}$$

$$\begin{cases} E \text{ is X-type error} \\ E \text{ commutes with all Z-stabilizers} \\ E \text{ anti-commutes with some logical-Z operator} \end{cases}$$

Sketch of the proof

Define a syndrome graph: vertices = Z-stabilizers
edges = qubits

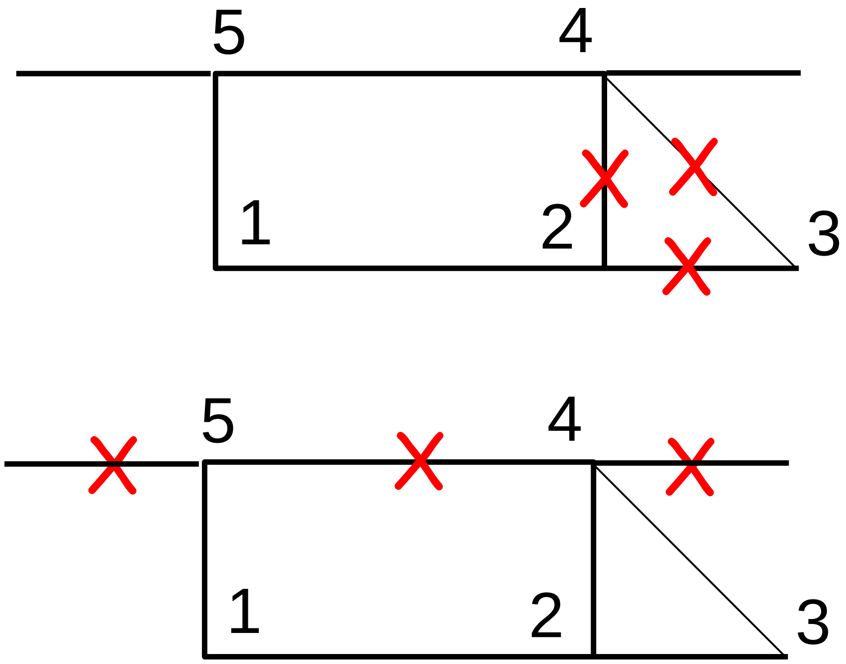
A single-qubit X error on edge (u, v) creates syndromes at u, v



X-errors = subsets of edges in the syndrome graph

Sketch of the proof

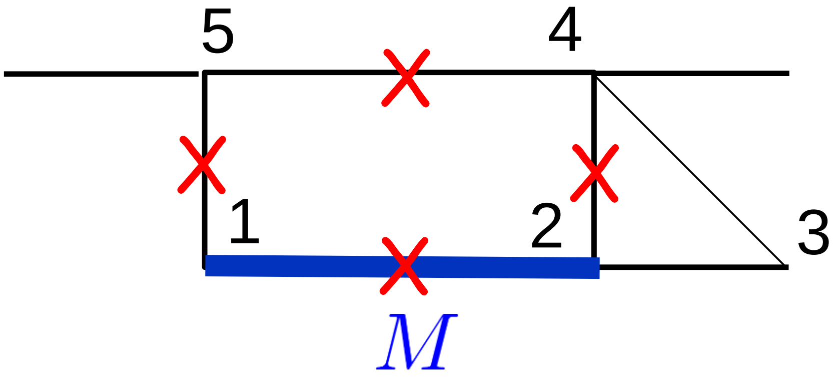
Claim 2: an X-error has zero syndrome iff it is a
cycle in the syndrome graph



Reminder: a subset of edges L is a cycle if any vertex has even number of incident edges from L

Sketch of the proof

We need to find a shortest cycle in the syndrome graph that has odd overlap with a given subset of edges M



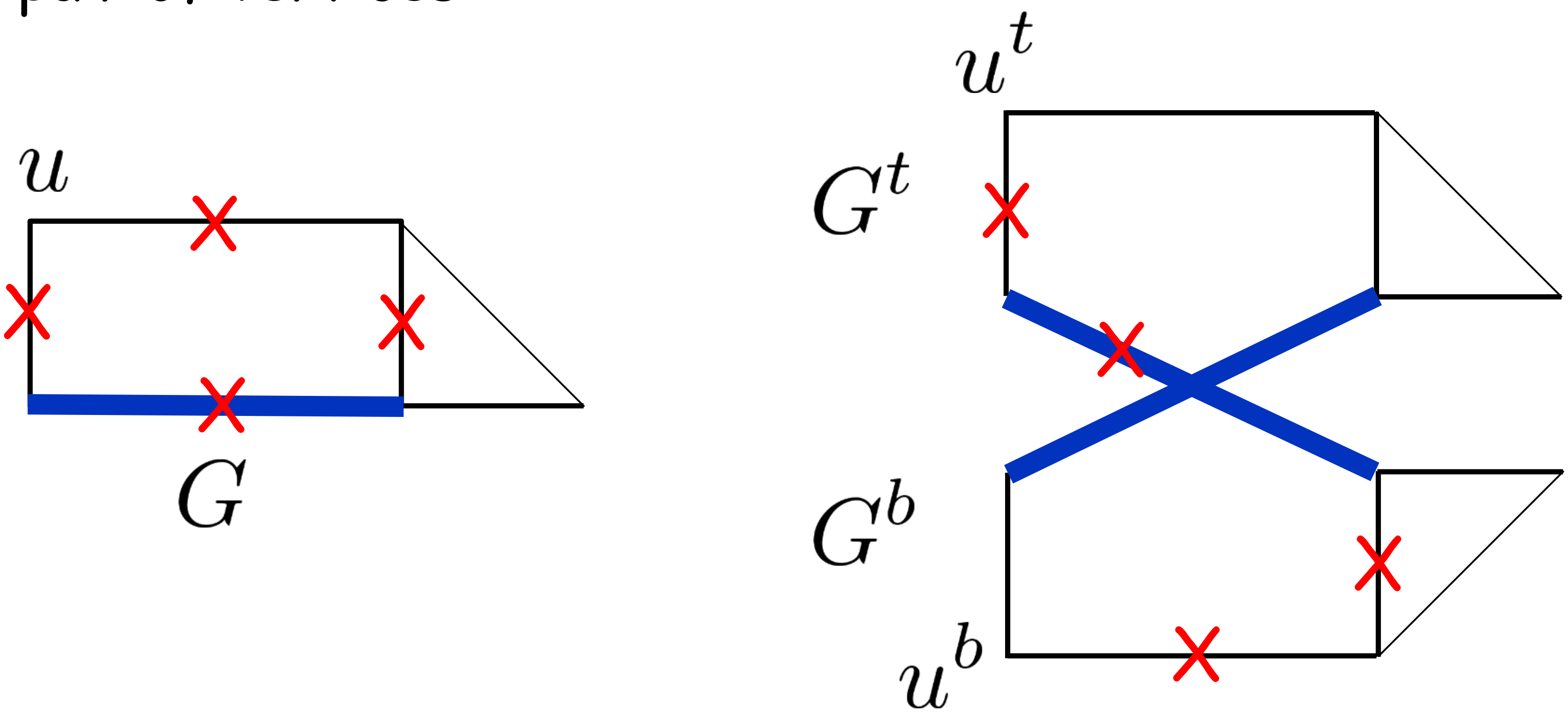
$$M = \overline{Z}_1, \ldots, \overline{Z}_k$$

Try all M's and pick the shortest cycle

Sketch of the proof

We need to find a shortest cycle in the syndrome graph that has odd overlap with a given subset of edges M

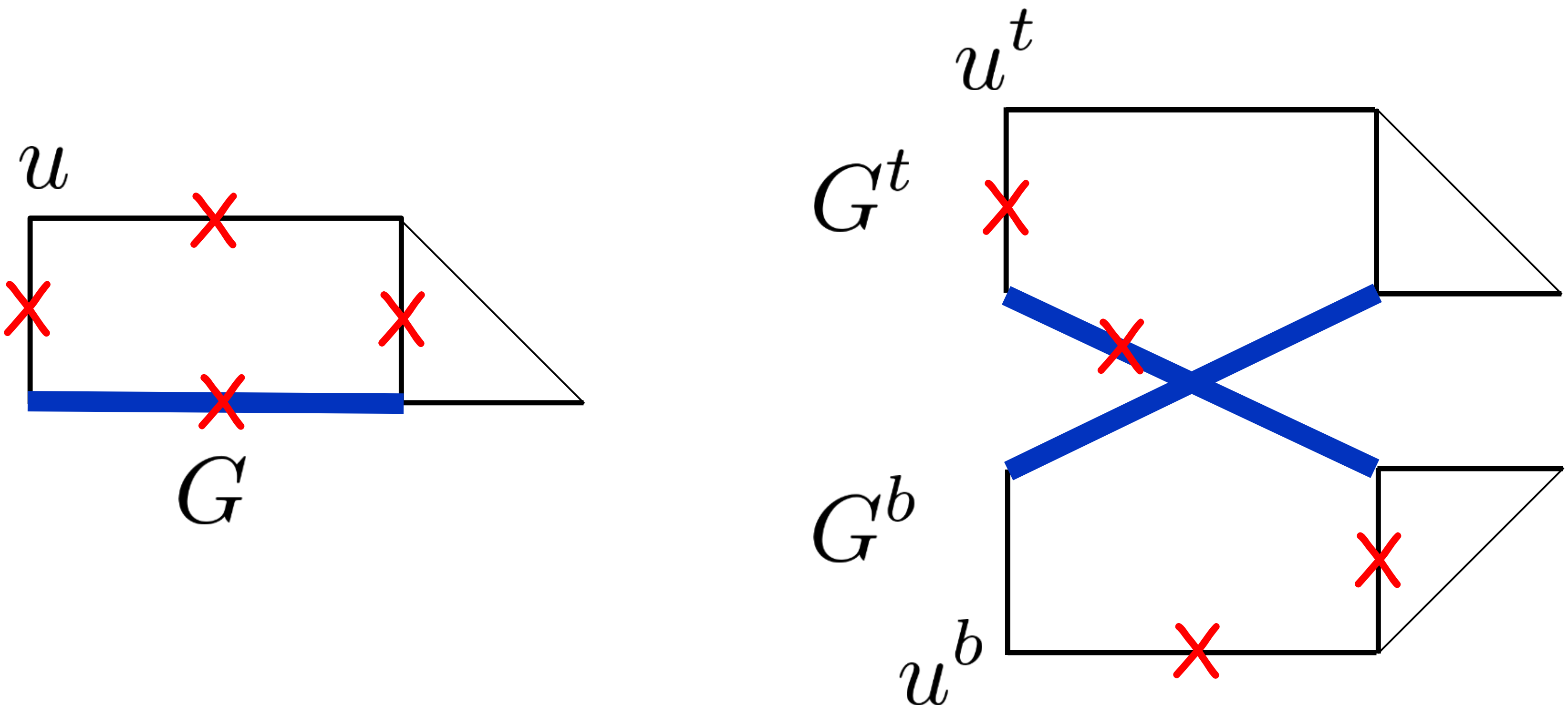
Reduction to finding a shortest path between a given pair of vertices:



Sketch of the proof

Claim 3: a shortest cycle that has odd overlap with M and contains a vertex u is mapped to a shortest path in the doubled graph connecting u^t and u^b

(or connecting some dangling nodes u^t and v^b)



Sketch of the proof

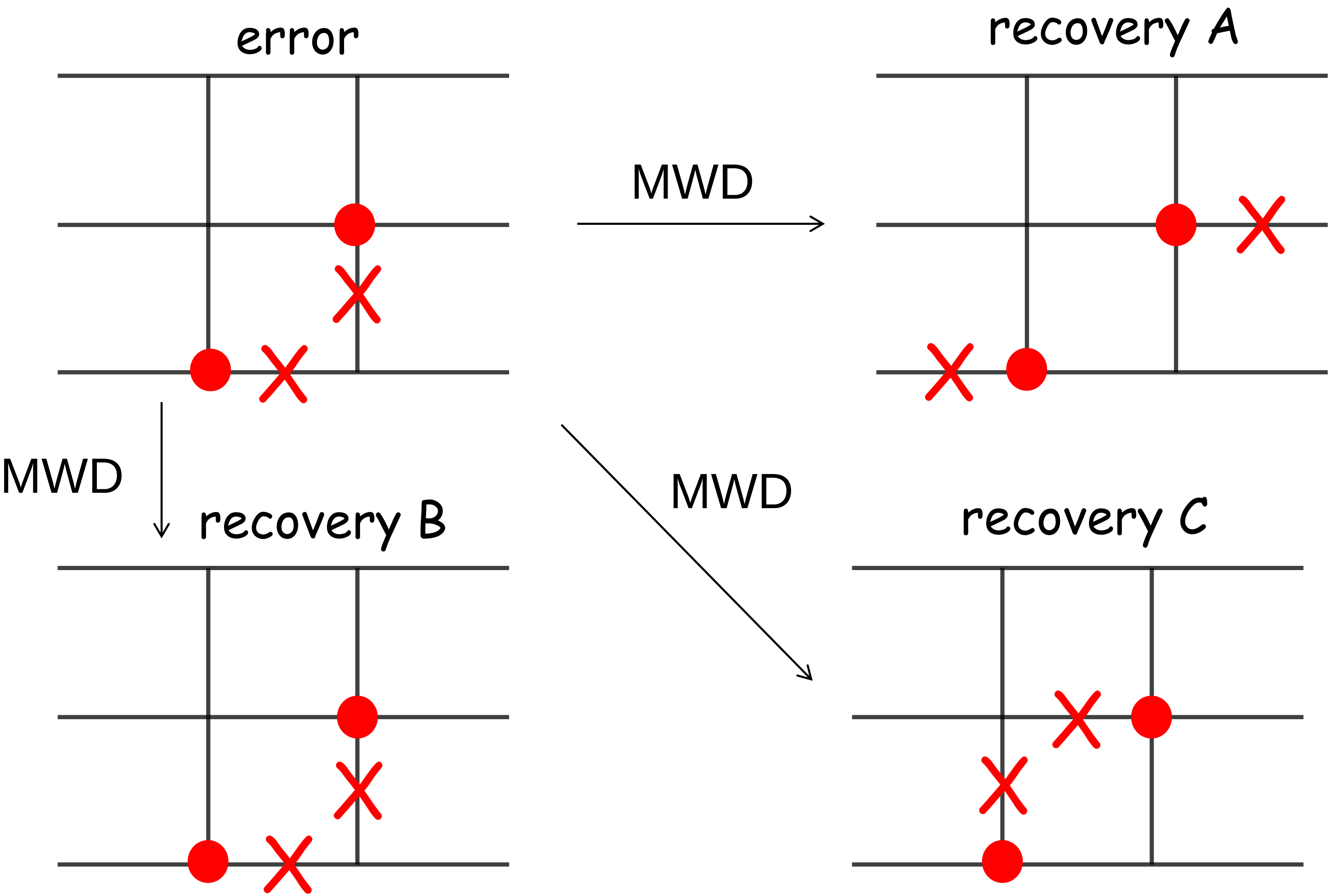
One can precompute shortest paths in the doubled graph using Dijkstra's algorithm in time

$$|V| \cdot O(|E| + |V| \log |V|) = O(n^2 \log(n))$$

Trying all pairs of vertices u^t and v^b and all logical operators takes time

$$O(kn^2 + n^2 \log(n))$$

Minimum-Weight decoder is not always optimal



error

MWD

recovery A

MWD

recovery B

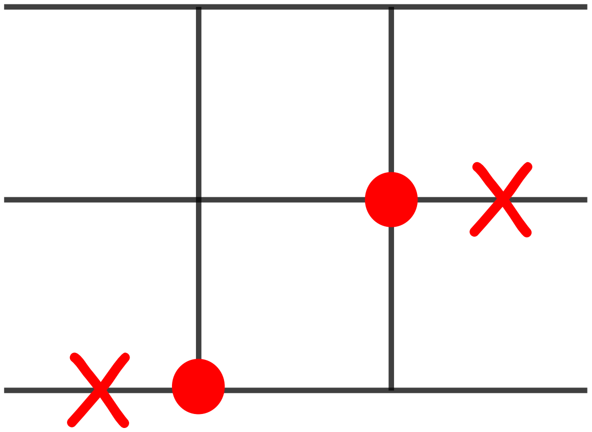
MWD

recovery C

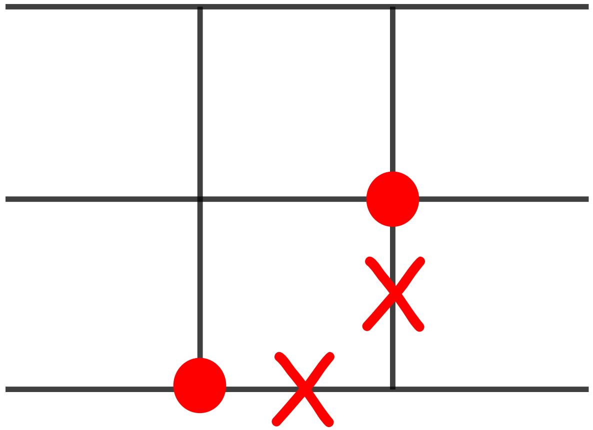
Minimum-Weight decoder is not always optimal

B and C differ by a stabilizer

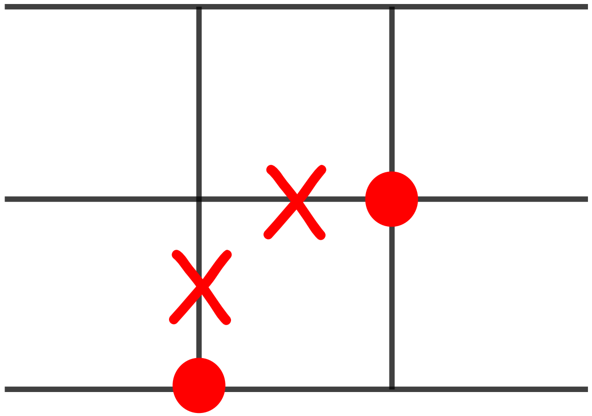
It does not matter whether the decoder picks B or C



recovery A



recovery B



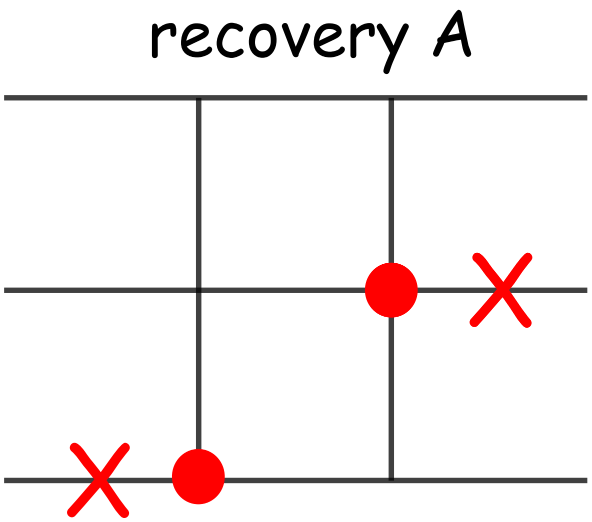
recovery C

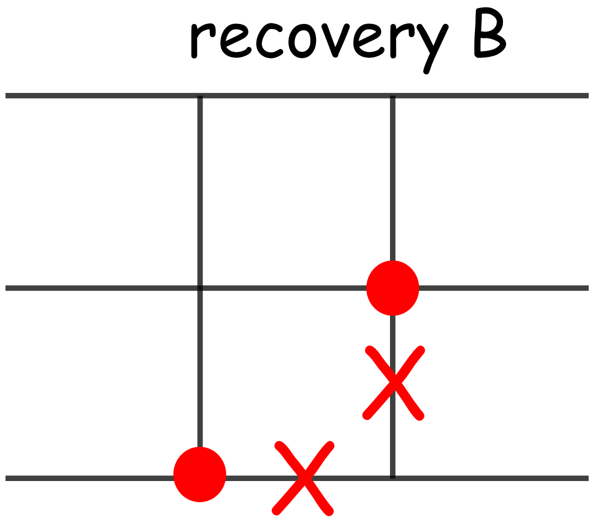
Minimum-Weight decoder is not always optimal

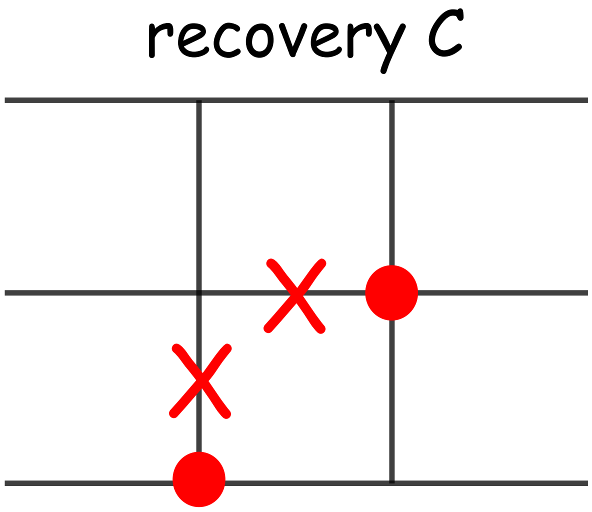
B and C differ by a stabilizer

$$\Pr(A) = \Pr(B) = \Pr(C)$$

Picking B or C is twice as likely to correct the error than picking A.







Max-Likelihood Decoder: pick the most likely equivalence class of errors consistent with the observed syndrome

$$L^* = \arg\max_L \sum_S \Pr(D \cdot L \cdot S)$$

Recovery: $R(D) = D \cdot L^*$

Optimal decoder for a given error model.

Computing the ML recovery is #P-hard problem

Iyer and Poulin (2013)

Min-Weight decoders*

Dennis, Kitaev, Landahl, Preskill (2001):

Reduction to Minimum Matching.
Use Edmond's blossom algorithm.
Worst-case runtime $O(n^3)$
Ignores correlations between
X and Z errors

Fowler, Wang, Hollenberg (2010):

More efficient implementation.
Average case runtime $O(n)$.
The fastest decoder for small
error rates.

Fowler (2013):

Account for correlations between
X and Z errors.

Max-Likelihood decoders*

Duclos-Cianci and Poulin (2010):

RG decoder: approximate
surface code by a concatenated
code; use belief propagation.
Worst-case runtime: $O(n\log(n))$

Hutter, Wootton and Loss (2014):

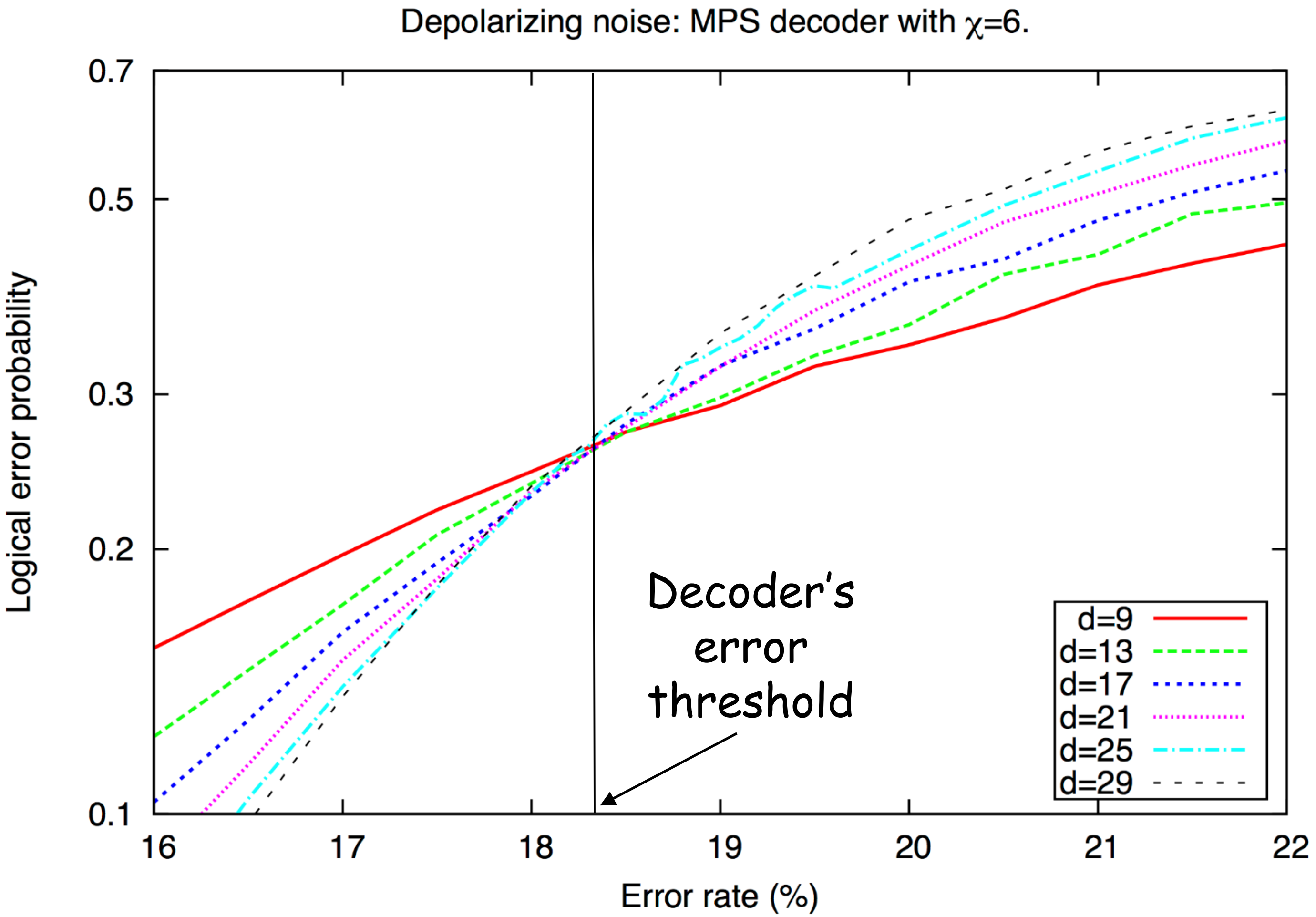
Markov chain decoder: sample
errors with a given syndrome.
Average-case runtime: $O(n^2)$

SB, Suchara, Vargo (2014):

Approximate probability of each
equivalence class of errors.
Matrix Product States algorithm
Worst-case runtime $O(n)$

---

* Surface code; Approximate implementations



Depolarizing noise: MPS decoder with χ=6.

Decoder's error threshold

Min-matching threshold: 15%

Markov chain decoder: 16% Hutter et al, PRA 89 022326 (2014)

Theoretical maximum: 18.9% Bombin et al, PRX 2 021004 (2012)

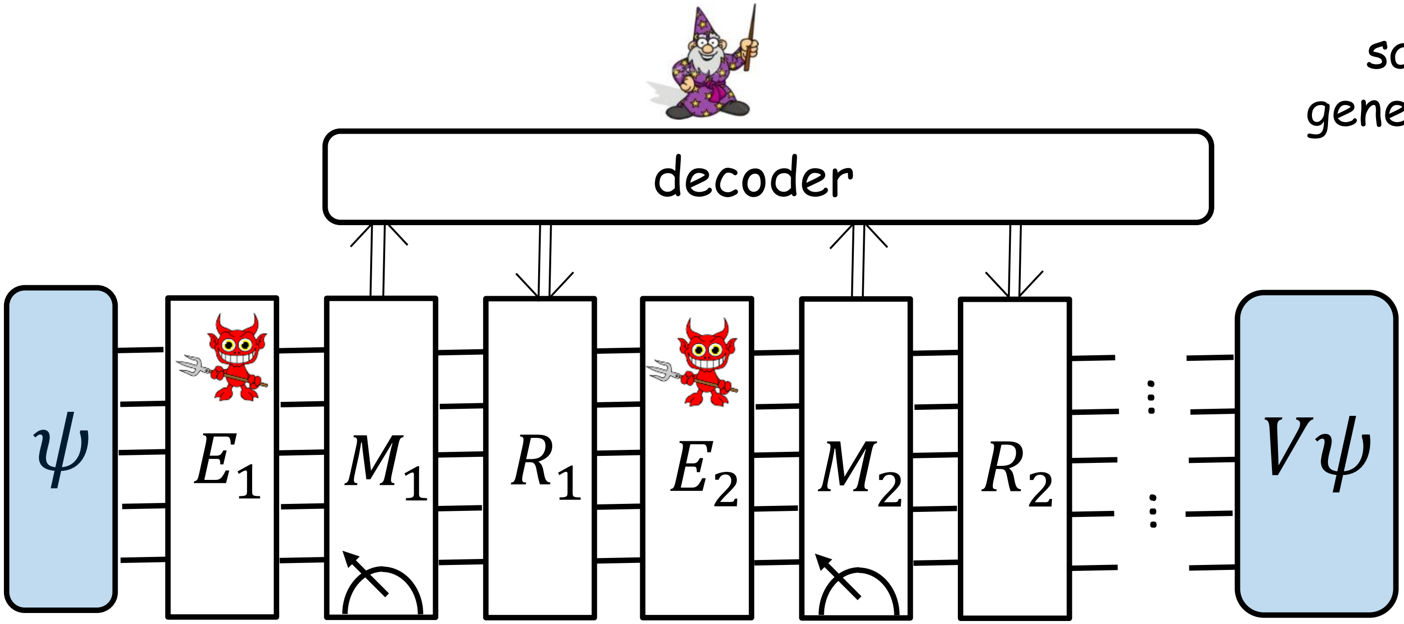
Outline

- Stabilizer codes
- The decoding problem and code distance
- Fault-tolerant code deformation
- Example 1: Shor's 4-qubit code
- Example 2: lattice surgery
- Maximum likelihood decoding

Code Deformation

Sequence of stabilizer codes $C_1, \ldots, C_L$ with $C_1 = C_L$

same generators



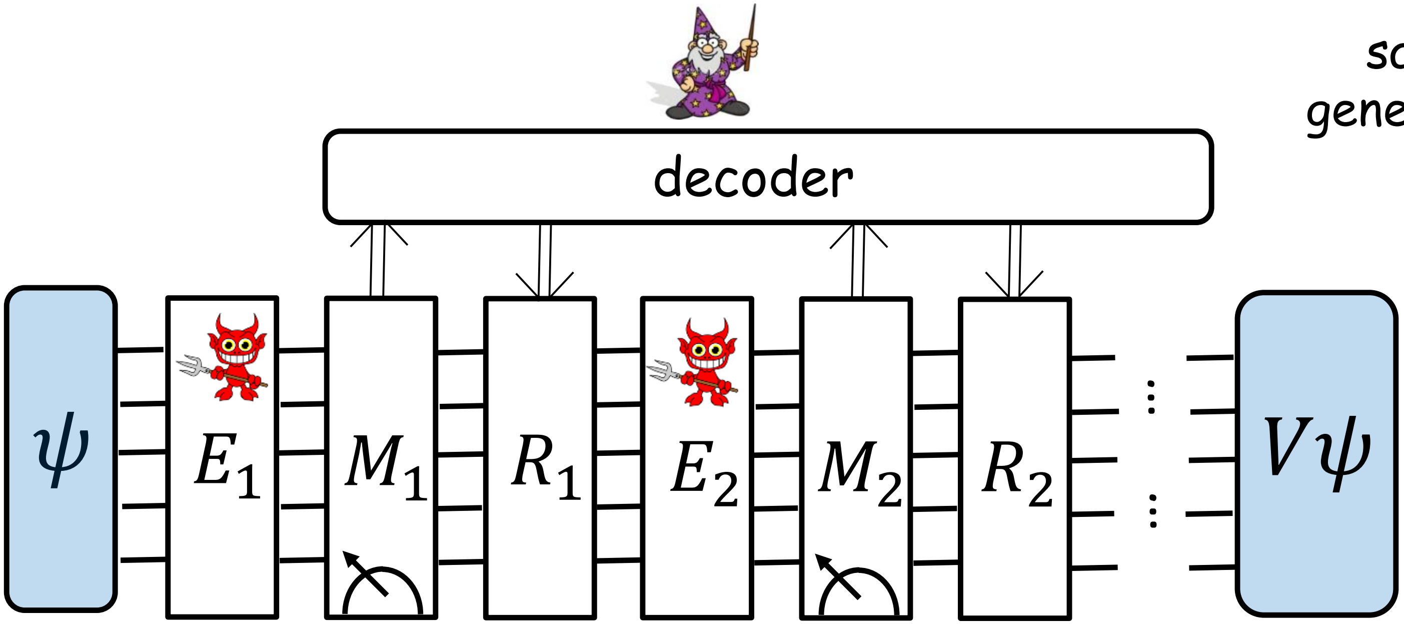
M_j : syndrome measurement for the code C_j

Goal: implement a target logical operation V using only syndrome measurements, error correction, and transversal gates

Code Deformation

Sequence of stabilizer codes $C_1, \ldots, C_L$ with $C_1 = C_L$

same generators



Fault-tolerance: must implement the target operation V for any pattern of low-weight errors:

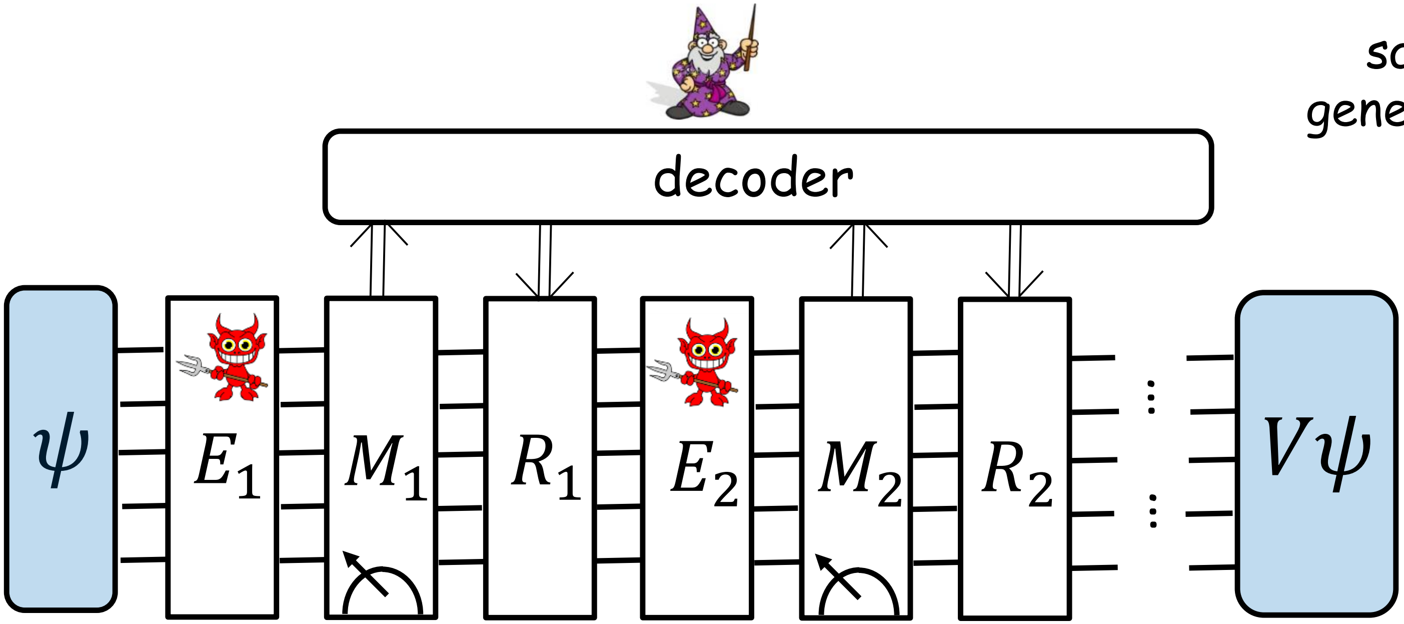
$$|E_j| \leq t \quad \forall j$$

Simplification: ideal syndrome measurements

Code Deformation

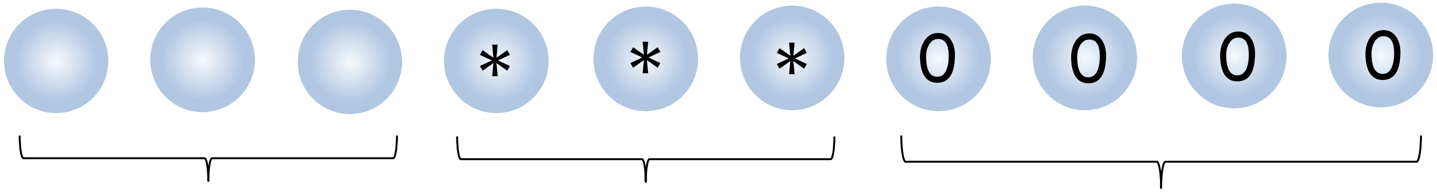
Sequence of stabilizer codes $C_1, \ldots, C_L$ with $C_1 = C_L$

same generators



New feature: newly added stabilizers do not commute with the existing stabilizers. Non-commuting stabilizers should not be used to diagnose errors. To describe this we need subsystem quantum codes.

Dummy subsystem code



Logical subsystem L

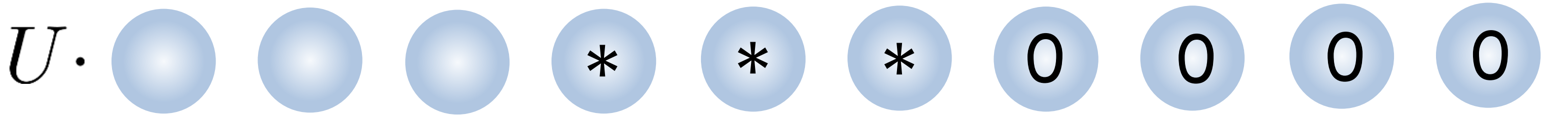
Gauge subsystem G

Syndrome subsystem S

L	G	S
Logical-Z Z_a	Gauge operators Z_a	Stabilizers Z_a
Logical-X X_a	Gauge operators X_a	Destabilizers X_a

State of the gauge qubits can be arbitrary

General subsystem code



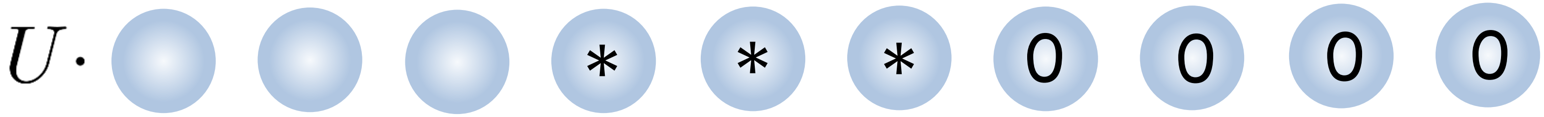
$$\overline{Z}_a = U Z_a U^\dagger \qquad \overline{X}_a = U X_a U^\dagger$$

$$\overline{X}_a, \overline{Z}_a \in \mathrm{Pauli}(n)$$

L	G	S
Logical-Z $\overline{Z}_a$	Gauge operators $\overline{Z}_a$	Stabilizers $\overline{Z}_a$
Logical-X $\overline{X}_a$	Gauge operators $\overline{X}_a$	Destabilizers $\overline{X}_a$

The dummy subsystem code in a rotated basis

General subsystem code



$$\overline{Z}_a = U Z_a U^\dagger \qquad \overline{X}_a = U X_a U^\dagger$$

$$\overline{X}_a, \overline{Z}_a \in \mathrm{Pauli}(n)$$

L	G	S
Logical-Z $\overline{Z}_a$	Gauge operators G_a^Z	Stabilizers S_a
Logical-X $\overline{X}_a$	Gauge operators G_a^X	Destabilizers D_a

The dummy subsystem code in a rotated basis

General subsystem code

Codespace: $\mathcal{Q} = \{\psi \in (\mathbb{C}^2)^{\otimes n} \; : \; S_a \psi = \psi \quad \forall a\}$

$$\mathcal{Q} = \mathcal{Q}_L \otimes \mathcal{Q}_G$$

A logical state η is encoded by

$$\rho(\eta) = \eta \otimes \frac{I}{\dim\left(\mathcal{Q}_G\right)}$$

The gauge subsystem can be made maximally mixed by applying a random gauge operator

Any Pauli error E admits a unique decomposition

$$E = \underset{\text{destabilizer}}{D} \cdot \underset{\text{logical operator}}{L} \cdot \underset{\text{gauge operator}}{G} \cdot \underset{\text{stabilizer}}{S}$$

Syndrome of E determines the destabilizer part D

Is this enough information to correct the error ?

$$E = D \cdot L \cdot G \cdot S \qquad\qquad E' = D \cdot L' \cdot G' \cdot S'$$

$$\boxed{E\rho(\eta)E^\dagger = E'\rho(\eta)(E')^\dagger \quad \forall\, \eta \quad \text{iff} \quad L = L'}$$

$$\rho(\eta) = \eta \otimes \frac{I}{\dim\left(\mathcal{Q}_G\right)}$$

Any Pauli error E admits a unique decomposition

$$E = D \cdot L \cdot G \cdot S$$

destabilizer logical operator gauge operator stabilizer

Syndrome of E determines the destabilizer part D

Is this enough information to correct the error ?

$$E = D \cdot L \cdot G \cdot S \qquad E' = D \cdot L' \cdot G' \cdot S'$$

$$\boxed{E\rho(\eta)E^\dagger = E'\rho(\eta)(E')^\dagger \quad \forall\, \eta \quad \text{iff} \quad L = L'}$$

Decoder has to guess the logical part of the error

The stabilizer and the gauge parts do not matter

Code distance

Minimum weight of a Pauli error which has zero syndrome and has non-trivial logical part:

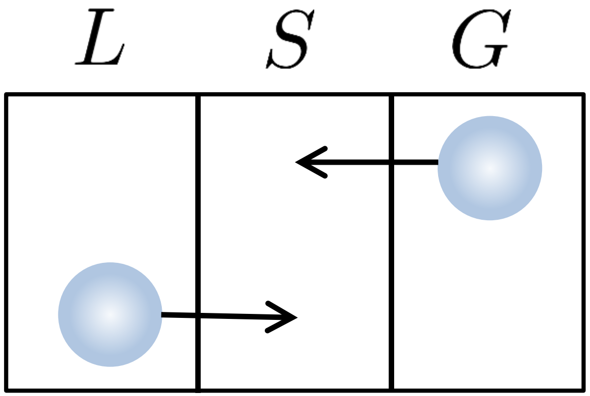
$$\boxed{d = \min_{G,S} \min_{L \neq I} |L \cdot G \cdot S|}$$

$$[[n, k, d]]$$

Half-distance: $t = (d-1)/2$

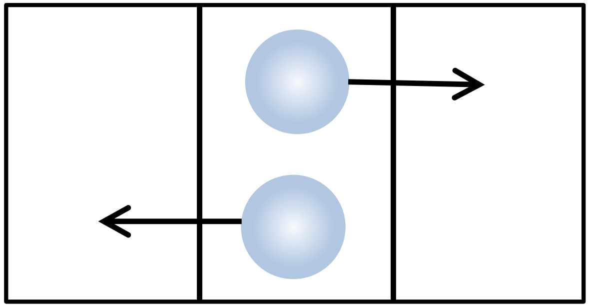
Min-Weight decoder corrects any error of weight $\leq t$

Elementary code deformations



start measuring new stabilizers

measure some logical qubits

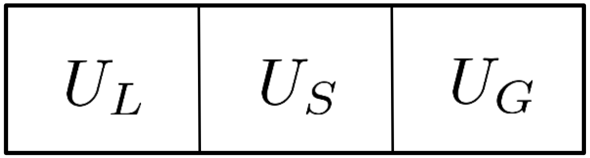


stop measuring some stabilizers

initialize some logical qubits

transversal logical gates

Choose new generators of L, S, G (leave the code unchanged):



$$\overline{X}_a \leftarrow \overline{X}_a S_b \qquad G_a^X \leftarrow G_a^X S_b$$

$$\overline{Z}_a \leftarrow \overline{Z}_a S_b \qquad G_a^Z \leftarrow G_a^Z S_b$$

Fault-tolerant code deformation

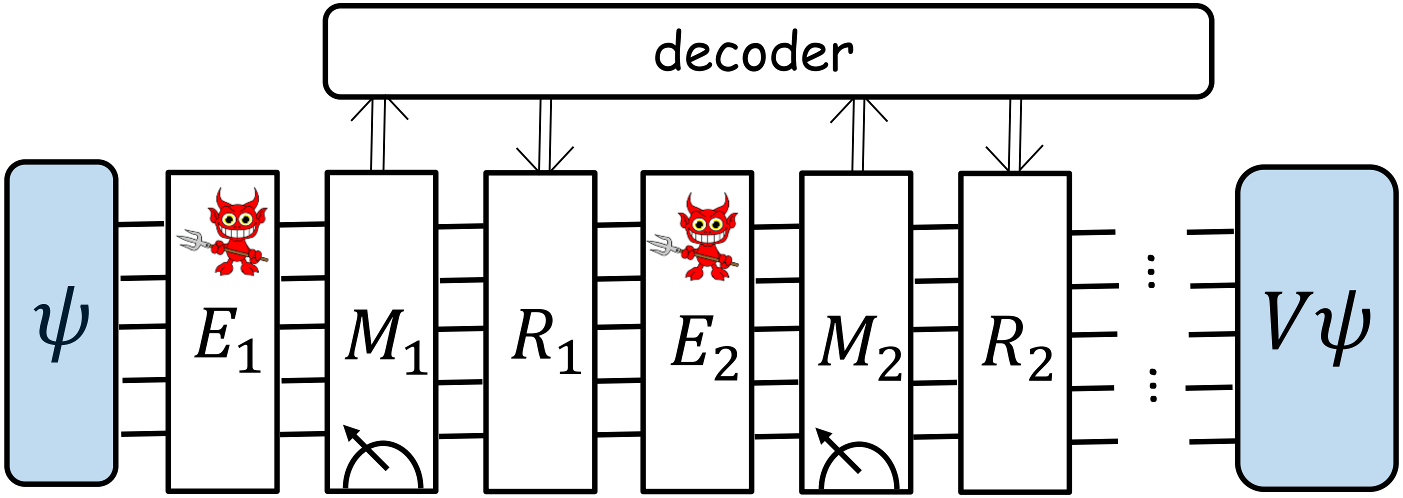
Sequence of subsystem codes $C_1, \ldots, C_L$ with $C_1 = C_L$

Lemma.

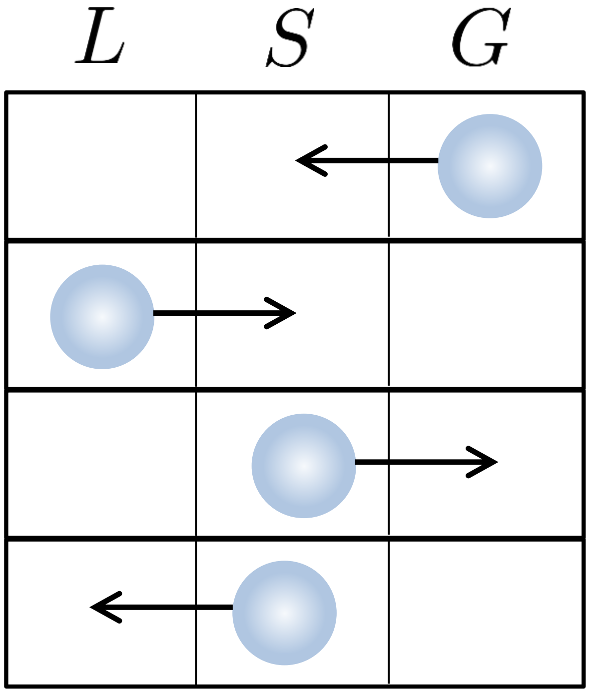
Assume C_j has distance $d_j \geq 2t + 1$.

Assume C_{j+1} is elementary deformation of C_j

Then any error pattern with $|E_j| \leq t$ can be corrected.



Fault-tolerance conditions



$$d_{j-1} \geq 2t+1$$

$$d^*_{j-1} \geq 2t+1$$

$$d_j \geq 2t+1$$

$$d^*_j \geq 2t+1$$

$$C_{j-1} \to C_j$$

d^* : ignore logical operators that were stabilizers at the previous step. Ignore logical operators that will become stabilizers at the next step.