LECTURE 1 - 3

Introduction to quantum error correction

Tobias Haug · Technology Innovation Institute

3rd Quantum Computing School · ICTP-SAIFR · November 2026

Daniel Gottesman, "Surviving as a Quantum Computer in a Classical World" (2024 draft)

D.J. Spencer, S.P. Jain, A. Tanggara, Z. Sun, T. Haug, D. Khu, K. Bharti, “Quantum error correction and fault tolerance: A comprehensive tutorial”l, arXiv:2605.29137 (2026)

Asia Pacific QEC Seminars https://www.youtube.com/playlist?list=PLKl95OY6CIkJWDk1y4xgN0ftb7i0slc2a

About me

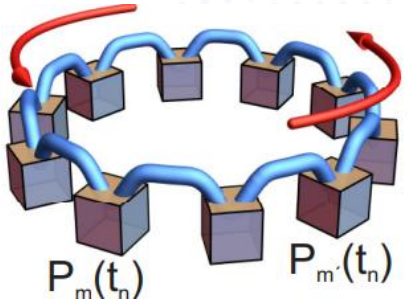
Tobias Haug – Lead researcher at Technology Innovation Institute (TII)

Employment

CQT → Imperial College London → TII Technology Innovation Institute

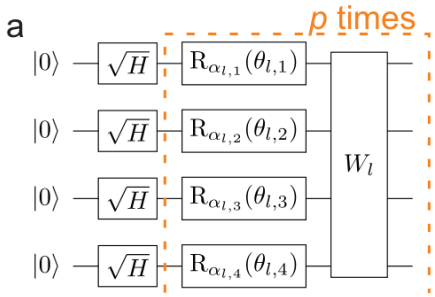
Research

2016 – Cold atoms and machine learning



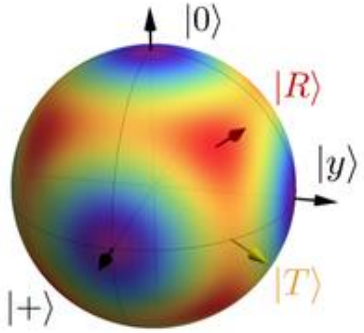
- Cold atom quantum transport
- Neural network optimisation

2020 – Quantum machine learning



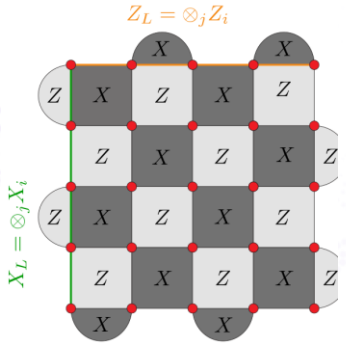
- Circuit capacity
- NISQ algorithms

2022 – Quantum many-body magic



- "resource theory of fault-tolerant quantum computing"
- Complexity of many-body systems

2024 – Quantum error correction



- New types of decoders
- Improved scheduling of QEC
- Limits of fault-tolerance
- Magic gates
- Tutorial on QEC

Technology Innovation Institute



- Research institute in Abu Dhabi, UAE
- ~100 researchers experiment&theory, on quantum error correction quantum computing, physics, superconducting qubits, quantum communication,…
- We are hiring!







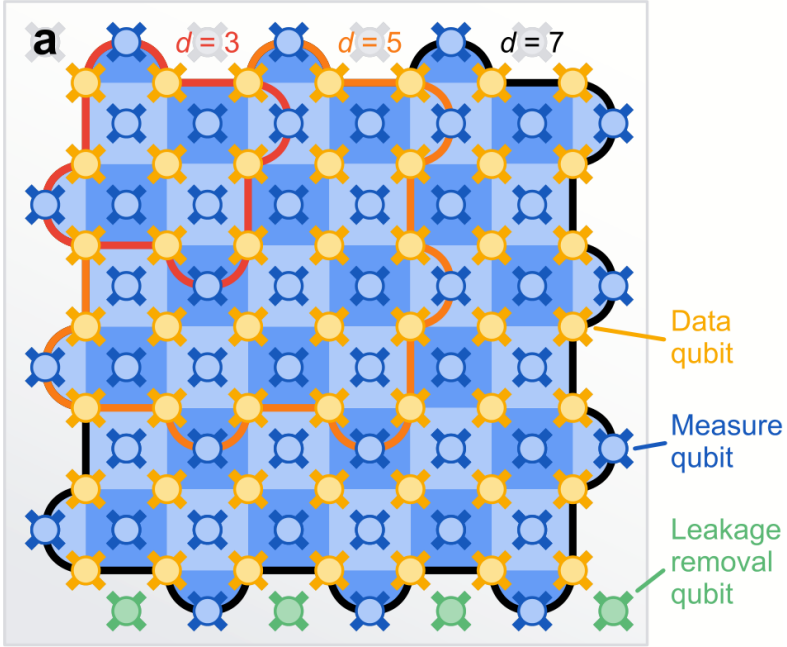


Why quantum computers need error correction

The goal: reliable quantum advantage

Quantum algorithms offer to solve problems intractable for classical computers.

However, bare quantum systems are too noisy
→Quantum error correction



Many physical qubits
One protected logical qubit

What would useful quantum advantage mean?

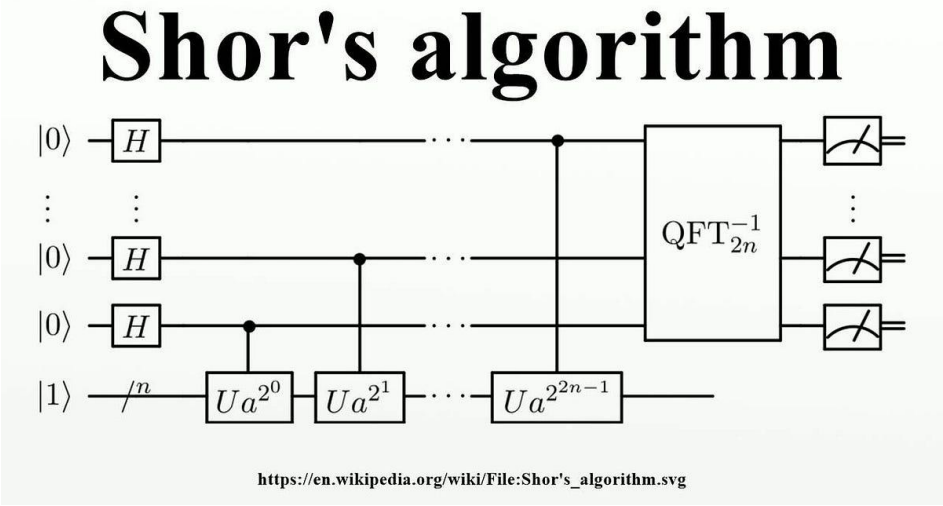
“Deep” quantum algorithms

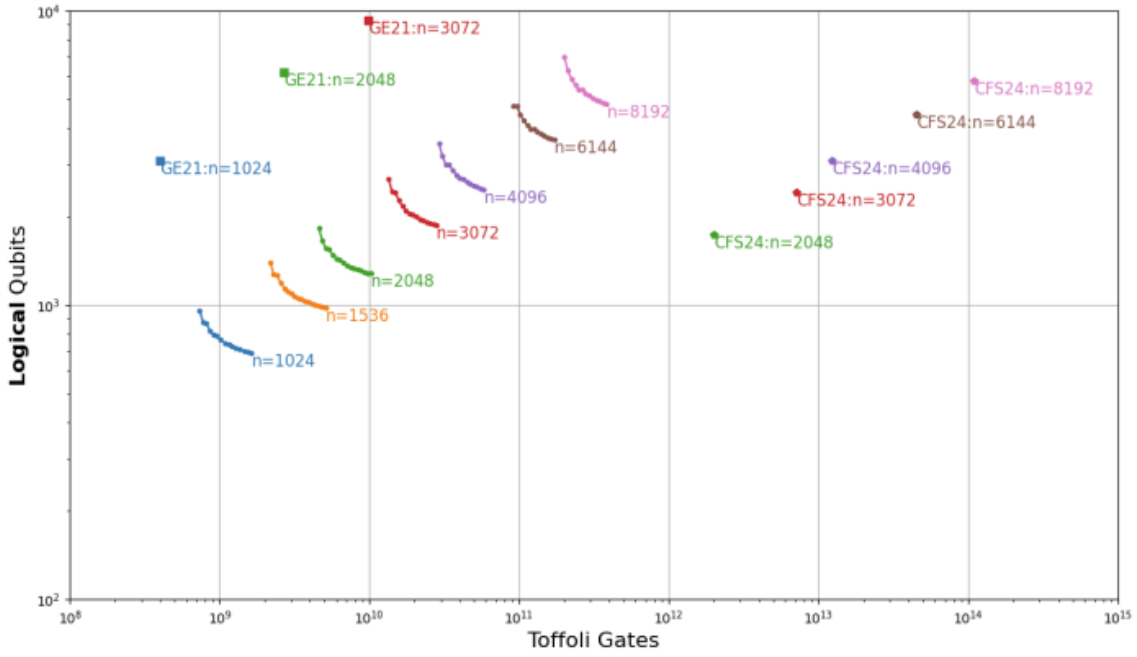
Quantum algorithms that are believed to outperform any classical method

Examples:

- Shor’s algorithms for factoring
- Simulation of quantum systems such as molecules (e.g. for chemistry, medicine,…)

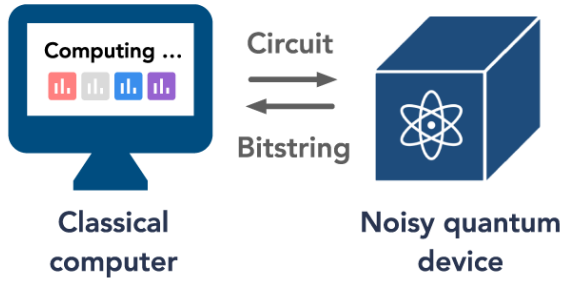
Require very large quantum circuits, how to implement?





Lloyd, Science 273, 1073 (1996); Shor, SIAM J. Comput. 26, 1484 (1997), C. Gidney arXiv:2505.15917

NISQ: experiments within a noise budget

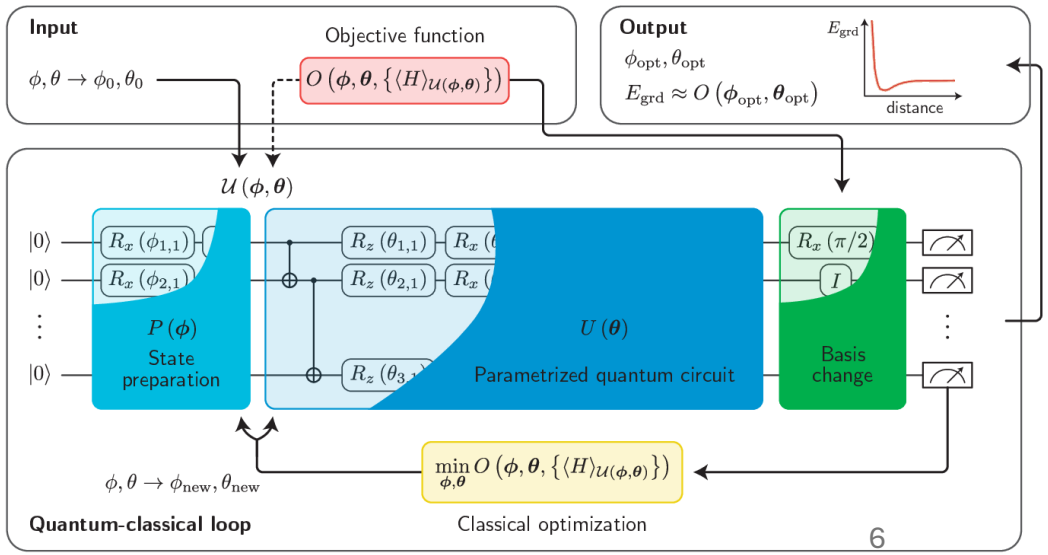


With noisy hardware, cannot run “deep” quantum algorithms (e.g. Shor needs noise per gate 10^{-15})

Instead: Noisy intermediate-scale quantum algorithms
“Combine classical optimization + shallow variational ansatz on quantum computer”

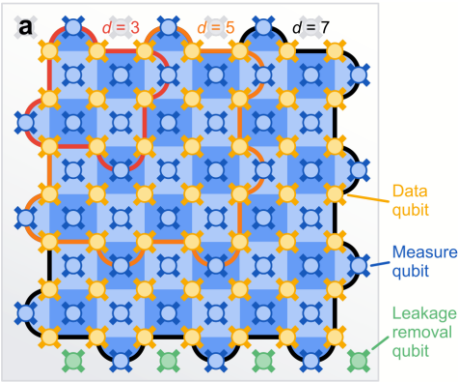
Many difficulties:

- Need to deal with noise (error mitigation is limited)
- Hard to prove speedup for NISQ
- Classical part of NISQ is often NP-hard



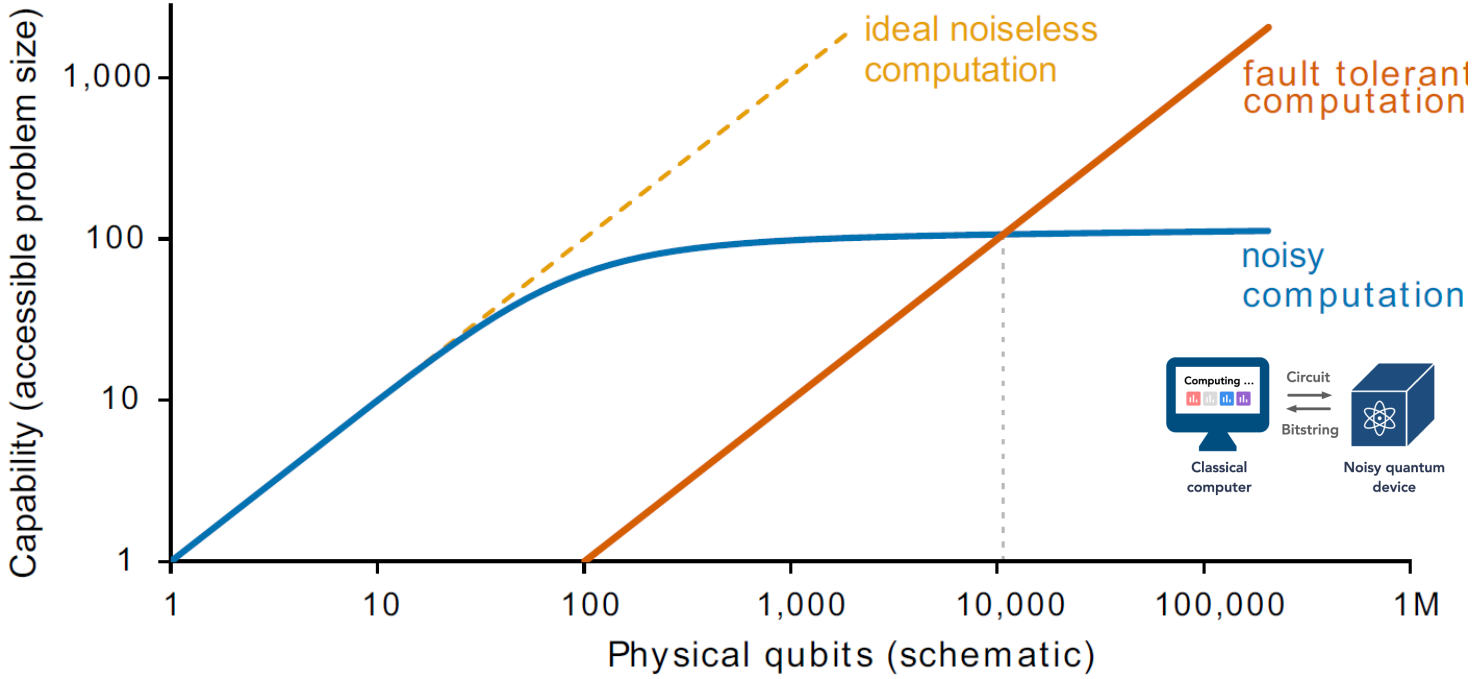
Preskill (2018); Takagi, Tajima & Gu, PRL 131, 210602 (2023)

NISQ vs QEC



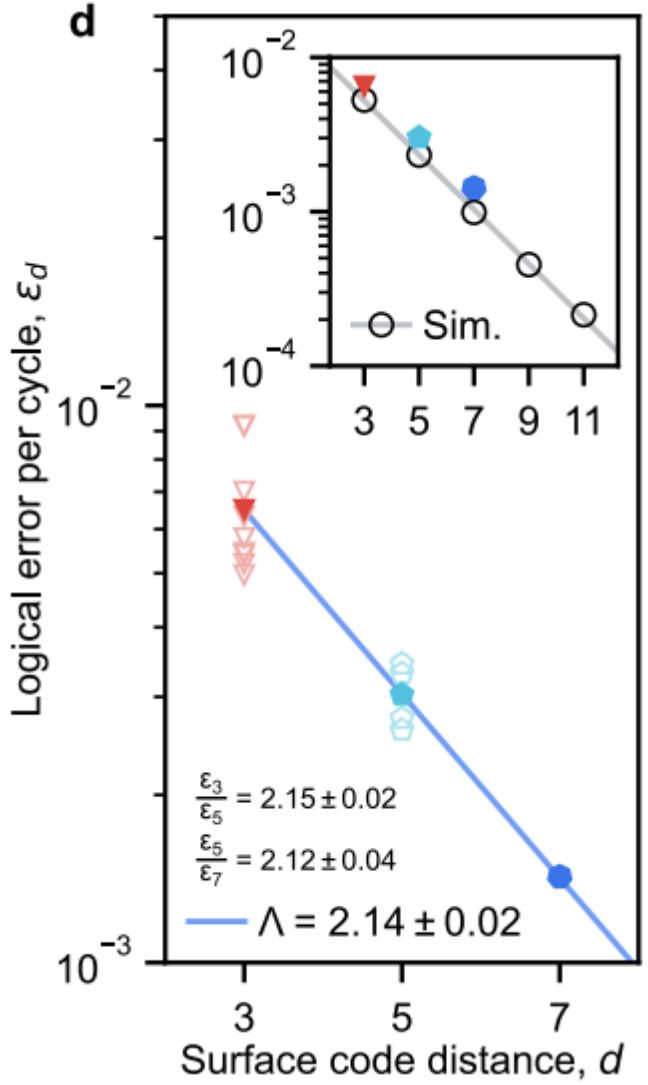
- NISQ hard to scale due to noise, reaches natural limit given by noise of quantum devices (more qubits do not help!)
- QEC has overhead (e.g. 1 logical qubit per 100 physical qubit), but error can be arbitrarily supressed with more qubits

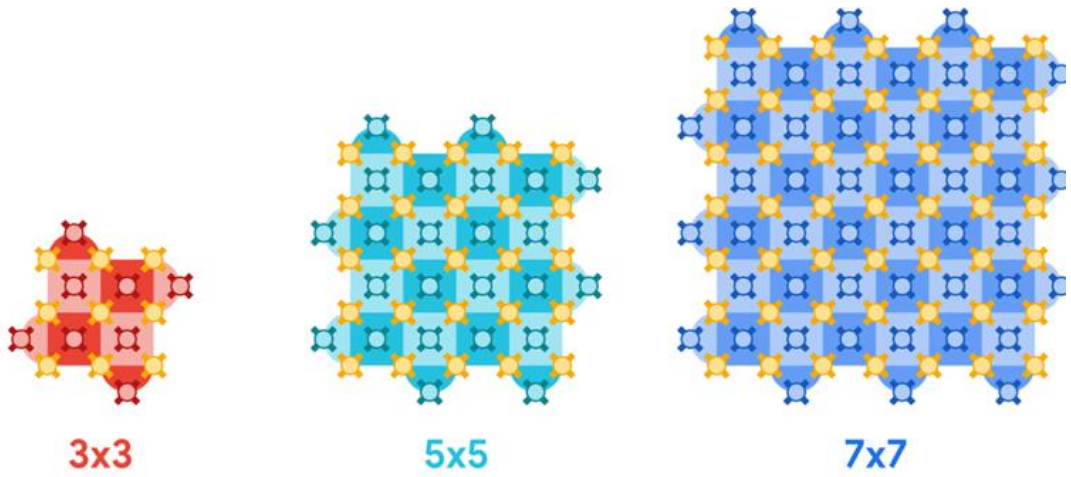
→QEC can run any problem sizes with enough physical qubits

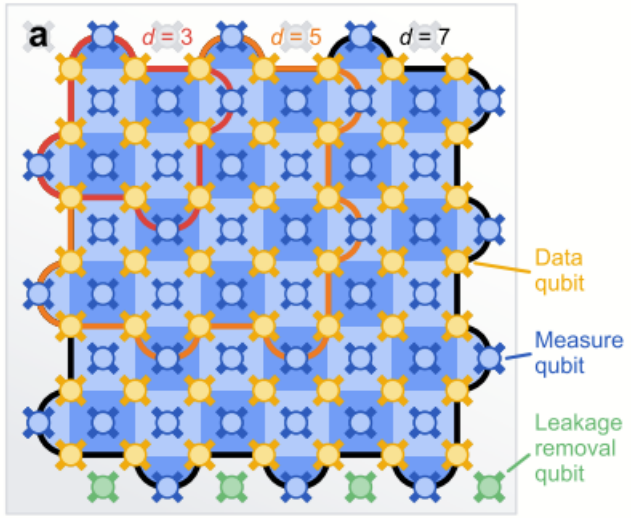


Preskill (2018); Takagi, Tajima & Gu, PRL 131, 210602 (2023)

Experiment: exponential error reduction with code distance







One logical qubit in many physical qubits

More physical qubits provide more protection.

About a factor of two less logical errors at each step.

Google Quantum AI (2024/2025). Redrawn from Table S5, arXiv:2408.13687. Ancillas are additional.

A lot of progress recently

2023 — **Neutral atoms**

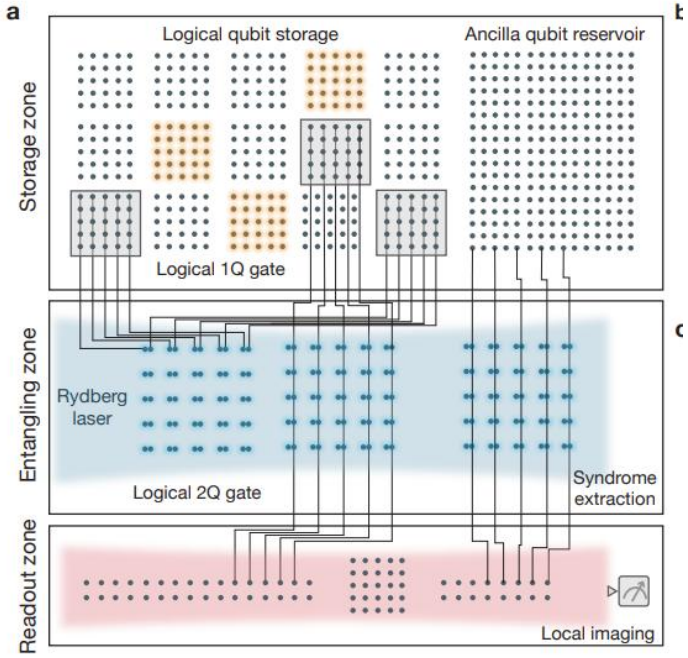
Up to 48 logical qubits in sampling circuits,
with error detection improving performance.

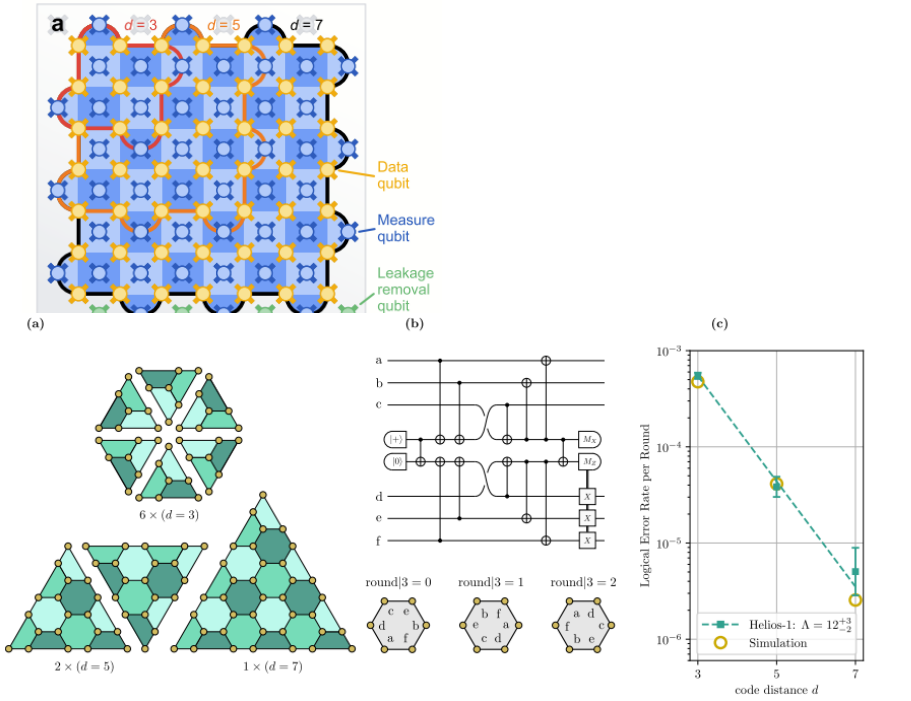
2024 — **Superconducting qubits**

Larger surface-code memories suppress errors.
A logical memory outlives its best physical qubit.

Oct 2026 — **Trapped ions**

Below-threshold color-code memories and
scaling improvements in logical Clifford circuits.





Bluvstein et al. (2023); Google Quantum AI (2024/2025); Lavasani et al., arXiv:2610.24077 (2026 preprint)

GOAL OF ERROR CORRECTION

- To preserve messages sent through a noisy transmission channel by encoding the messages in an **error-correcting code**
  - *More precisely:* to make sure the rate of corruption of encoded (i.e., **logical**) information is lower than that of the same information sent without the extra encoding step.

message

$$\rho \rightarrow \boxed{\mathcal{E}\text{ncoder} \quad \mathcal{N}\text{oise} \quad \mathcal{D}\text{ecoder}} \rightarrow \mathcal{D}\mathcal{N}\mathcal{E}(\rho) \approx \rho$$

space- or time-like



QUANTUM THREE QUBIT REPETITION CODE

(B) Quantum three qubit repetition code

(1) Code words

$$|0\rangle_L = |000\rangle \qquad |1\rangle_L = |111\rangle \qquad |\psi\rangle = \alpha|0\rangle + \beta|1\rangle \rightarrow |\psi\rangle_L = \alpha|000\rangle + \beta|111\rangle$$

Codespace $\quad \mathrm{span}(|0\rangle_L, |1\rangle_L)$

Error free encoding of a quantum state lies in this subspace

The states outside the codespace are the Errorspace (e.g. |010>

Code properties are written as [[n,k,d]] (here [[3,1,1]])

QUANTUM THREE QUBIT REPETITION CODE

(2) Errors

(a) Bit flip error

$$X|0\rangle = |1\rangle \qquad X|1\rangle = |0\rangle$$

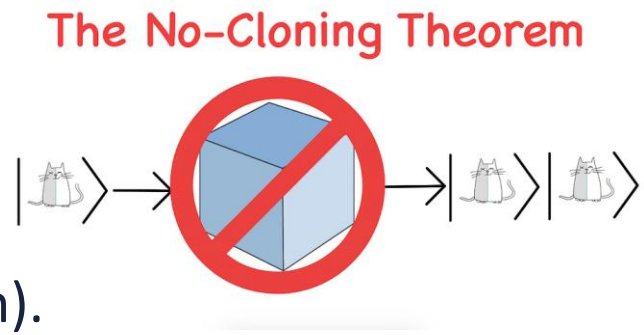
(b) Phase flip error

$$Z|0\rangle = |0\rangle \qquad Z|1\rangle = -|1\rangle$$

Remark: If both X and Z errors can be corrected, more general errors can be corrected as well.

How to go from classical to quantum error correction

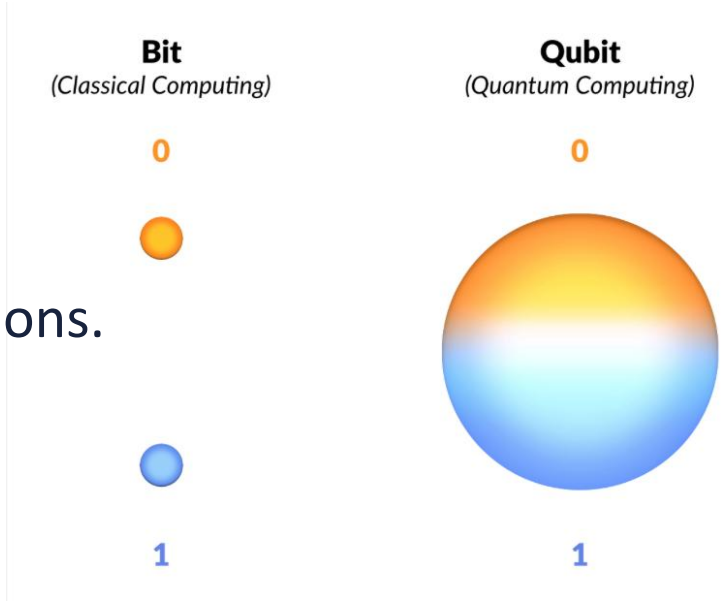
Classical error correction is simpler than quantum:
<u>Key obstacles:</u>



An unknown quantum state cannot be copied (no cloning theorem).

Measuring the state generally disturbs the information.

Noise model more complicated: phase errors and continuous rotations.



G §2.1; T §3.2

QUANTUM THREE QUBIT REPETITION CODE

(3) Encoded state after a single bit flip error

$$|\psi\rangle_L = \alpha|000\rangle + \beta|111\rangle$$

$$(I \otimes X \otimes I)|\psi\rangle_L = \alpha|010\rangle + \beta|101\rangle$$

(4) Error detection

$$\alpha\,|010\rangle + \beta\,|101\rangle \longrightarrow |010\rangle \quad \text{or} \quad |101\rangle$$



- Do not measure individual qubits in the computational basis
- it would detect the error, it would also result in the collapse of the encoded qubit.



- Measure symmetries of the state rather than individual bits
- Example: parity of any pair of bits

Even parity: 00,11
Odd parity: 01, 10

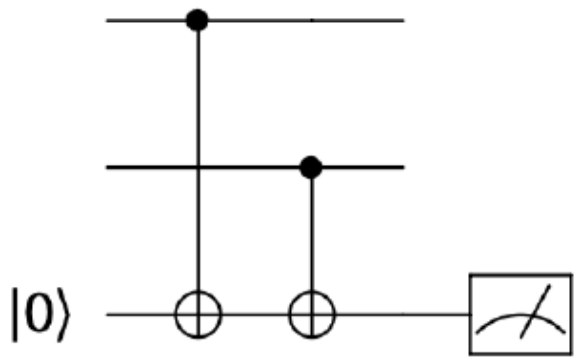
QUANTUM THREE QUBIT REPETITION CODE

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Quantum circuit to measure parity



$$|x_1 x_2\rangle\,|0\rangle \longmapsto |x_1 x_2\rangle\,|x_1 \oplus x_2\rangle$$

$$ZZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Equivalent to measuring $Z \otimes Z$ on first two qubits

Do not affect code space, i.e. stabilize the code space (+1 eigenvalue)!

$$|0\rangle_L = |000\rangle \qquad |1\rangle_L = |111\rangle$$

$$Z \otimes Z|00\rangle = |00\rangle$$

$$Z \otimes Z|11\rangle = |11\rangle$$

$$Z \otimes Z|01\rangle = -|01\rangle$$

$$Z \otimes Z|10\rangle = -|10\rangle$$

Error	$Z \otimes Z \otimes I$	$Z \otimes I \otimes Z$	$I \otimes Z \otimes Z$
$X \otimes I \otimes I$	-1	-1	+1
$I \otimes X \otimes I$	-1	+1	-1
$I \otimes I \otimes X$	+1	-1	-1

Apply the inverse of the error operator to correct the error

Error	Correction operator
XII	XII

QUANTUM THREE QUBIT REPETITION CODE

(6) Logical operations on encoded states

Logical operators: Operators that preserve code-space, but are not stabilizers

We want $\bar{X}$ and $\bar{Z}$ such that

$$\bar{X}|0\rangle_L = |1\rangle_L \qquad \bar{X}|1\rangle_L = |0\rangle_L$$

$$\bar{Z}|0\rangle_L = |0\rangle_L \qquad \bar{Z}|1\rangle_L = -|1\rangle_L$$

One possible choice of $\bar{X}$ and $\bar{Z}$

$$\bar{X} = XXX$$

$$\bar{Z} = ZII$$

Code distance *d*: Logical operator (over all possible choices) with lowest Hamming weight → d_Z=1, d_X=3

[[3,1,1]] code!

How to decode: 3-qubit Repetition code

- Codestate $|0_\mathrm{L}\rangle \sim |000\rangle$ $|1_\mathrm{L}\rangle \sim |111\rangle$ $\mathrm{span}(|0\rangle_L, |1\rangle_L)$
- Decode *X* errors using Z stabilizers (leave logical information invariant) [3, 1, 3]

$$S_1 = Z \otimes Z \otimes I \qquad S_2 = I \otimes Z \otimes Z$$

- Assume *X* error on first qubit: $X_1 |\psi_\mathrm{L}\rangle$

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

n qubits; *m* stabilizers

Error vector $e = (1, 0, 0)$

$$\boxed{X_1 |0_\mathrm{L}\rangle = |100\rangle}$$

- We cannot measure *X* directly (destroys logical information!)

→Instead: Measure syndrome: $s = (1, 0)$ (1: syndrome has flipped) $\boldsymbol{s = H_Z e}$

- Next: Use syndrome information to figure out error

Example: 3-qubit Repetition code

- Decoding: Find X error vector *e* that matches syndrome $s = (1, 0)$ $X_1 |\psi_L\rangle$

$$S_1 = Z \otimes Z \otimes I \qquad S_2 = I \otimes Z \otimes Z$$

- Not unique! Two possible choices

$$e = (1, 0, 0)$$
$$X_1 |0_L\rangle = |100\rangle$$

or

$$e = (0, 1, 1)$$
$$X_2 X_3 |0_L\rangle = |011\rangle$$

- How to choose?

→Make best guess, i.e. most likely error

- X error appear with low probability *p*

→ Most likely $e = (1, 0, 0)$

Correction

- After decoding, what can happen?

$s = (1, 0)$ Most likely error $e = (1, 0, 0)$

1. Lets say error occurred is what decoder found:

$$e = (1, 0, 0)$$

$$X_1 |0_\mathrm{L}\rangle = |100\rangle$$

→Correction operation: Apply inverse error

$$X_1 X_1 |0_\mathrm{L}\rangle = |0_\mathrm{L}\rangle$$



2. But if error was actually

$$e = (0, 1, 1)$$

$$X_2 X_3 |0_\mathrm{L}\rangle = |011\rangle$$

→Logical error $X_1 X_2 X_3 |0_\mathrm{L}\rangle = |1_\mathrm{L}\rangle$



Error suppression

- Decoding picks most likely error

→There is a chance we make a mistake!

- Without code: Errors appear with probability *p*

- Now: assume error $e = (0, 1, 1)$ $X_2 X_3 |0_\mathrm{L}\rangle = |011\rangle$ $s = (1, 0)$

  – Appears with probability ~p^2

$$S_1 = Z \otimes Z \otimes I$$

$$S_2 = I \otimes Z \otimes Z$$

- Decoding chooses (wrongly) the more likely error $e = (1, 0, 0)$ (probability *p*)

→We make mistake with probability ~p^2 (instead of *p* without QEC)!

Error suppression with distance

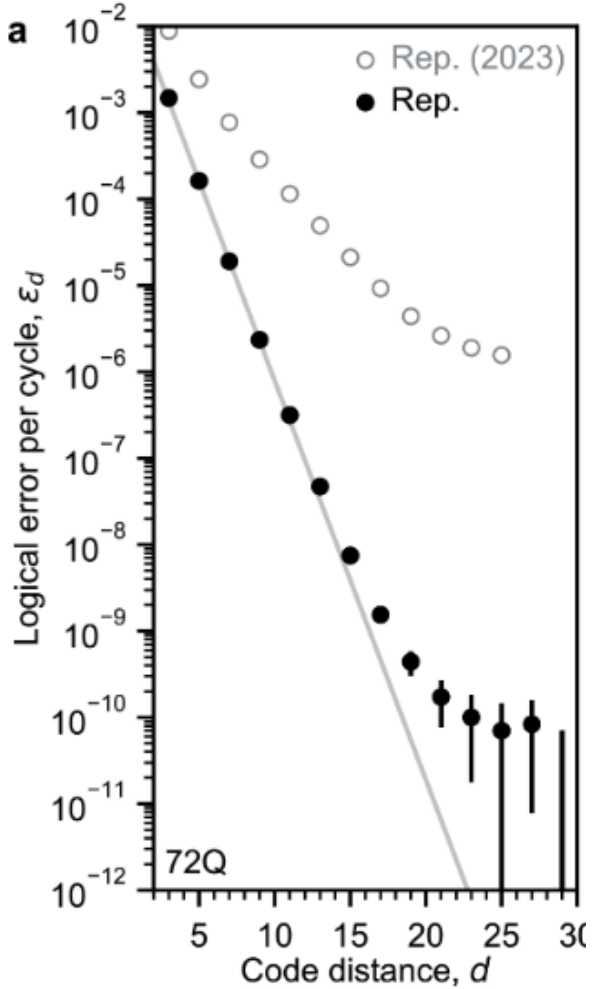
- Below a threshold physical error, logical errors are usually supressed exponentially with *d*

$$p_{\text{logical}} \propto (p/p_{\text{th}})^{\lfloor (d+1)/2 \rfloor}$$

- *Example: Google Willow QEC experiment for repetition code*

→ Scaling QEC allows execution of deep circuits!

- In practice, non-ideal noise sources can prohibit this scaling

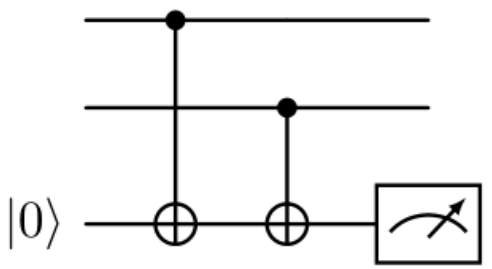


Google Quantum AI and Collaborators Nature 638, 920–926 (2025)

Advanced questions

- What if both data-qubit and syndrome measurement is noisy?

$|0\rangle_L = |000\rangle \qquad |1\rangle_L = |111\rangle \qquad |\psi\rangle = \alpha|0\rangle + \beta|1\rangle \rightarrow |\psi\rangle_L = \alpha|000\rangle + \beta|111\rangle$



$$S_1 = Z \otimes Z \otimes I$$

$$S_2 = I \otimes Z \otimes Z$$

- How to deal with unequal noise probabilities?

How to protect against phase errors

We can protect X errors, but what about phase?

In repetition code, Z errors commute with both parity checks and cannot be corrected!

$$Z_1 |\psi_L\rangle = \alpha |000\rangle - \beta |111\rangle$$

Code parameters of repetition code: [[n,k,d]]:
n: physical qubits
k: logical qubits
d: distance

$$[\![3, 1, 1]\!]$$

G §2.6; T §3.2

Repetition code for phase errors

Encode into $\alpha|+++\rangle + \beta|---\rangle$.

Use X_1X_2 and X_2X_3 as parity checks.

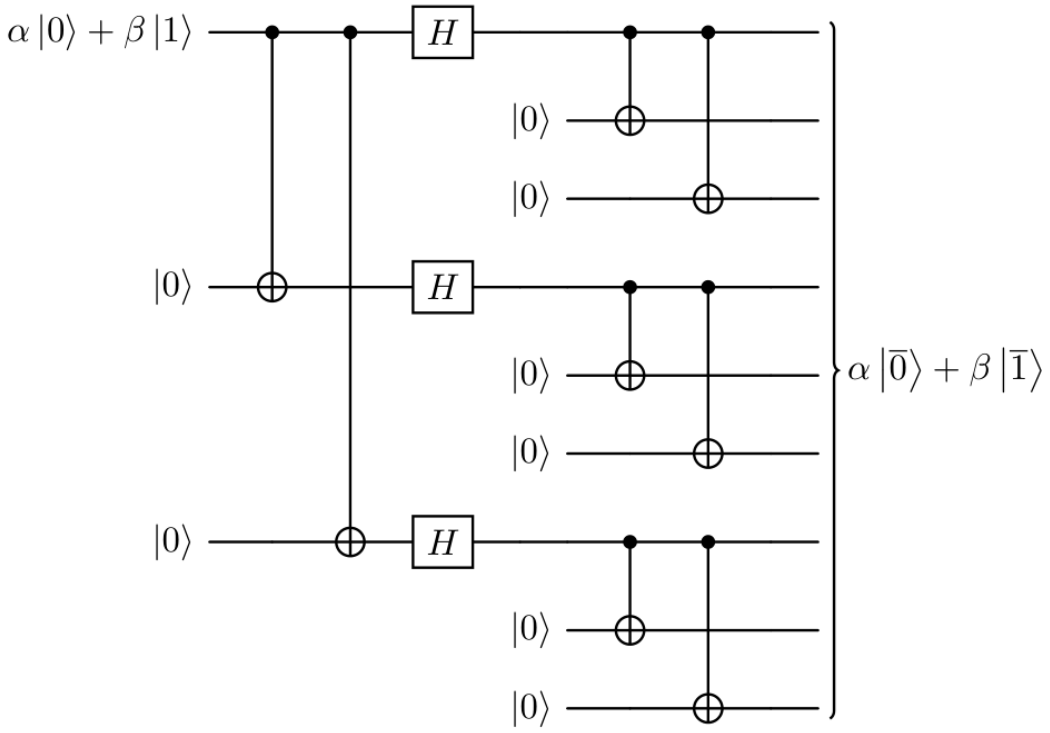
This protects phase flips, but leaves bit flips unprotected.

$$HZH = X$$

H: Hadamard gate

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Concatenate two codes: Shor code



$$|0_L\rangle = \frac{1}{2\sqrt{2}}(|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle) \otimes (|000\rangle + |111\rangle)$$

and

$$|1_L\rangle = \frac{1}{2\sqrt{2}}(|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle) \otimes (|000\rangle - |111\rangle).$$

Shor code [[9,1,3]]:
Concatenation of regular and phase repetition code

Within each group: protection against bit flips.

Across the groups: protection against phase flips.

→ Protection against arbitrary single-qubit errors

$Z_1Z_2, \qquad X_1X_2X_3X_4X_5X_6$

Why Pauli errors are enough

We know how to protect against X and Z errors. What about arbitrary single-qubit channels?
Protection against X and Z implies protection against Y errors

$Y = i\,X\,Z$

Every single-qubit operator is a linear combination of I, X, Y, and Z.

A recovery that corrects a set of operators also corrects its linear span.

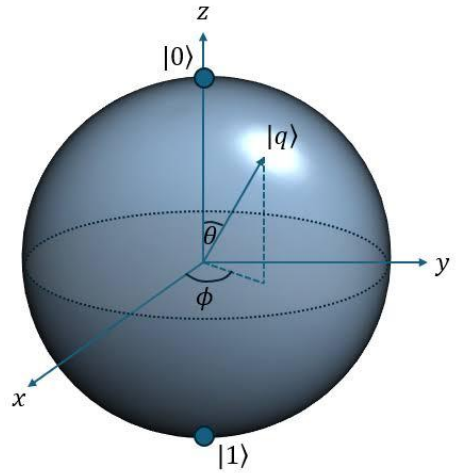
$$A = c_0 I + c_1 X + c_2 Y + c_3 Z$$

For example: Partial trace (loss of qubit), depolarizing channel, …

$$\rho_2 = \mathrm{Tr}_1(|\psi\rangle\langle\psi|)$$

Question: How to recover after qubit loss (note that after that we have a mixed state and lost some qubits!)?

Continuous noise can be corrected



Consider an unknown rotation on the middle qubit.

$$U_2(\theta) = \cos\left(\frac{\theta}{2}\right) I - i\sin\left(\frac{\theta}{2}\right) X_2$$

$$U_2(\theta)\,|\psi_L\rangle = c\,|\psi_L\rangle - isX_2\,|\psi_L\rangle$$

Here c = cos(θ/2) and s = sin(θ/2).

Parity check



Error is a continuous angle θ, but syndromes are discrete, is this problem?
No problem: Parity checks project this state onto either I or X Pauli error

QEC “discretizes” errors, which then can be corrected via Pauli

CSS checks in the Shor code

For general code, we need both X and Z checks.

A CSS code has stabilizer generators made entirely of X operators or entirely of Z operators. (i.e. no mixed operators like XZZX allowed)

Shor's code is an example.

$$Z_1Z_2, \qquad X_1X_2X_3X_4X_5X_6$$

Reading a CSS check matrix

A row lists the qubits touched by one check. Count overlaps modulo two.

Operator on 6 qubits	Binary support	Overlap with the other row
Z_1Z_2	1 1 0 0 0 0	Two shared qubits
$X_1X_2X_3X_4X_5X_6$	1 1 1 1 1 1	Binary dot product = 0

CSS commutation condition for stabilizers:

$$H_X H_Z^{\mathsf{T}} = 0 \pmod 2$$

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

What would happen if an X check overlapped a Z check on just one qubit?

How QEC protects quantum information

Encoding stores a qubit in correlations across a codespace.

Noise perturbs code state

Parity checks give error syndrome without revealing the logical information.

Decoder recovers by using syndrome to find&apply correction operator



G Ch. 2–3; T §3–4

Formally: QEC Decoding for CSS codes

CSS code $[\![n, k, d]\!]$ characterized by parity check matrix

$$H = \begin{pmatrix} 0 & H_X \\ H_Z & 0 \end{pmatrix} \qquad H_X \cdot H_Z^{\mathrm{T}} = 0$$

Lets look at *Z* syndromes (i.e. Pauli Z measurements) which detect *X*-type errors (X syndrome analogously)

$$H_Z = \begin{pmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ h_{m1} & h_{m2} & \cdots & h_{mn} \end{pmatrix}$$

m syndrome measurements

n qubits

$h_{ij} \in \{0, 1\}$

each row corresponds to a syndrome measurement (i.e. stabilizers)

$$S_i = \bigotimes_{j=1}^{n} Z^{h_{ij}}$$

Number of logical qubits $k = n - rank(H_x) - rank(H_z)$

How to decode CSS codes

$$H_Z = \begin{pmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ h_{m1} & h_{m2} & \cdots & h_{mn} \end{pmatrix}$$

- Z syndrome measurement to find X errors
- Syndrome via product of check matrix with actual error (mod 2)

$$\boldsymbol{s} = H_Z \boldsymbol{e}$$

- Maximum likely-hood decoding (find most likely X error $\boldsymbol{e}$ given syndrome $\boldsymbol{s}$)

Find X error $\boldsymbol{e} = \{e_1, \ldots, e_n\}$ that
1. Matches syndrome $\boldsymbol{s} = H_Z \boldsymbol{e}$
2. Most likely $\boldsymbol{e}$ (given prior)

- Syndrome condition (1.) does not uniquely specify error

→ Optimization needed to find most likely error (given syndrome constraint)

Maximum Likelihood Decoding

Find X error $\boldsymbol{e} = \{e_1, \ldots, e_n\}$ that

1. Matches syndrome $\boldsymbol{s} = H_Z \boldsymbol{e}$
2. Most likely $\boldsymbol{e}$ (given prior)

- *X* error occurs on *j*th qubit with probability p_j
- Total probability for error configuration $\boldsymbol{e}$

$$P(\boldsymbol{e}) = \prod_{j=1}^{n} (1-p_j)^{1-e_j} p_j^{e_j} = \left[\prod_{j=1}^{n}(1-p_j)\right] \prod_{j=1}^{n} \left(\frac{p_j}{1-p_j}\right)^{e_j}$$

- Optimization problem can be formulated as a linear program (in general NP-hard)

$$\boldsymbol{e}^{\mathrm{ml}} = \arg\max_{\boldsymbol{e}} \{\ln(P(\boldsymbol{e}))\} = \arg\max_{\boldsymbol{e}} \left(C - \sum_j w_j e_j\right)$$

$$\text{s.t. } \boldsymbol{s} = H_Z \boldsymbol{e}\,,$$

$$w_j = \ln((1-p_j)/p_j)$$

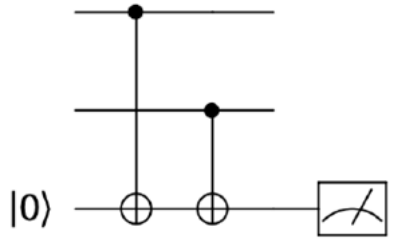
Implication: 1. QEC decoding needs (error corrected) classical computers
2. Decoding needs characterization of error rates

Fast vs accurate decoders

- need to solve optimization problem to infer most likely error that has happened

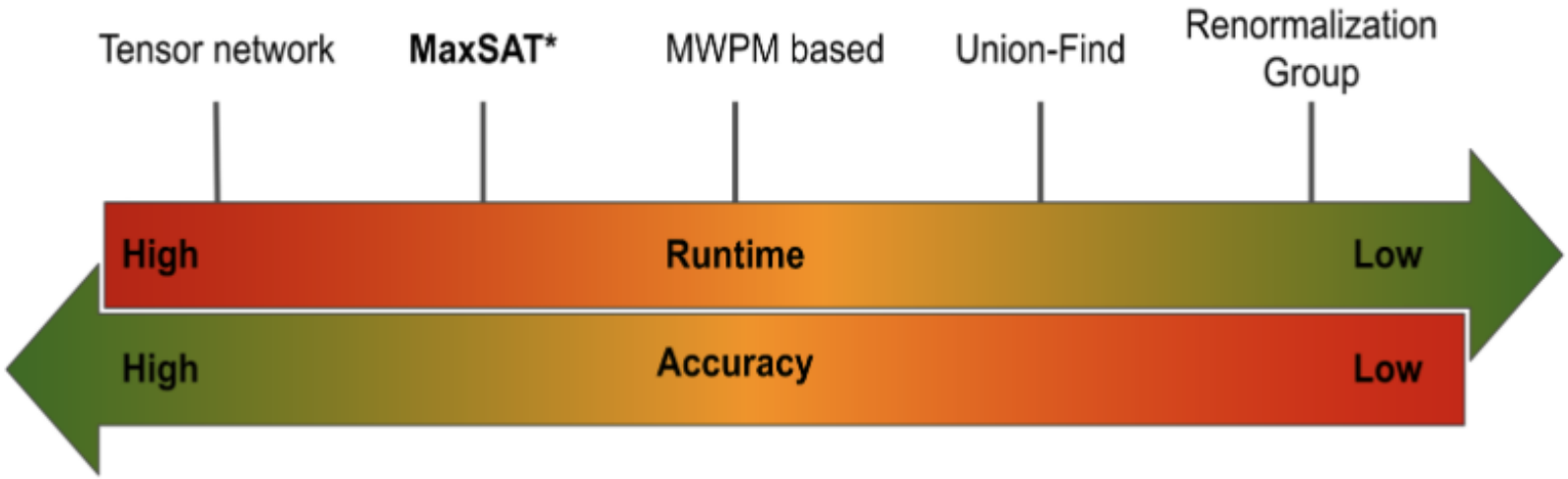
→“Decoding”, NP-hard problem! (In fact, optimal decoding is #P complete)

- Intrinsic tradeoff speed vs accuracy



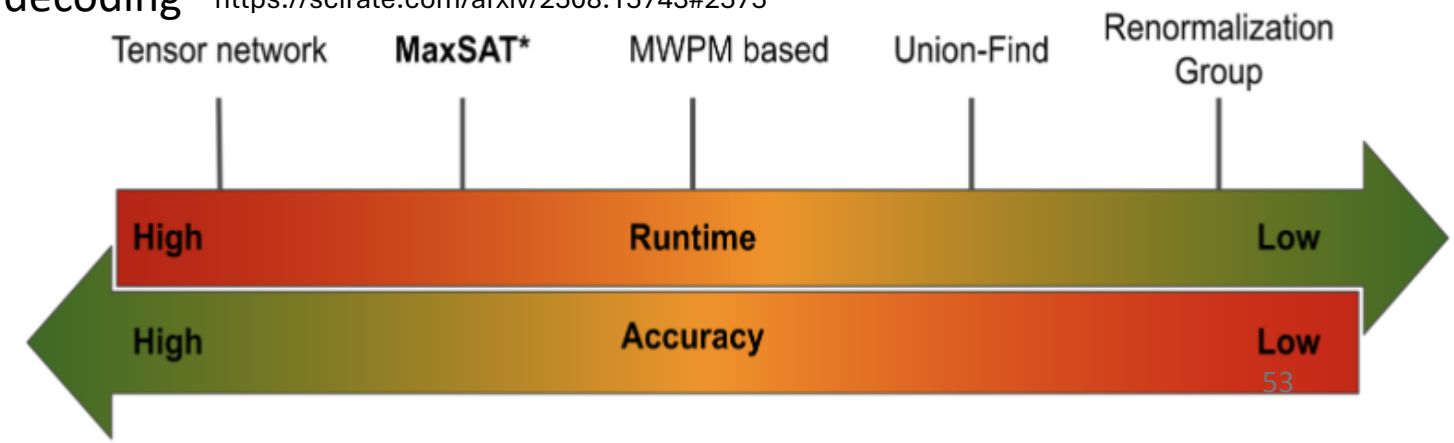
$$Z \otimes Z|00\rangle = |00\rangle$$

$$Z \otimes Z|11\rangle = |11\rangle$$

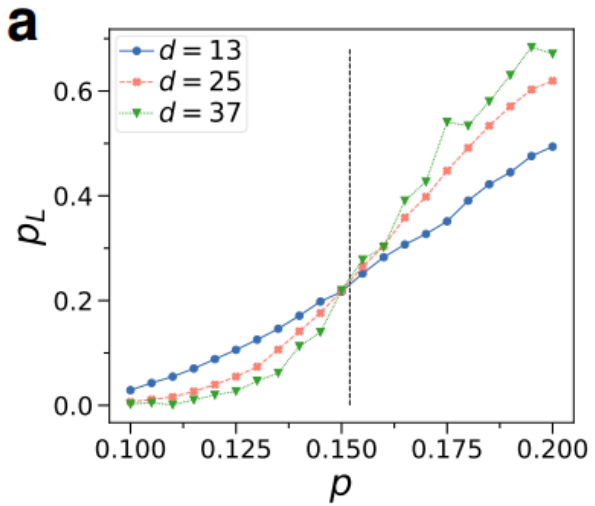


Fast vs accurate decoders

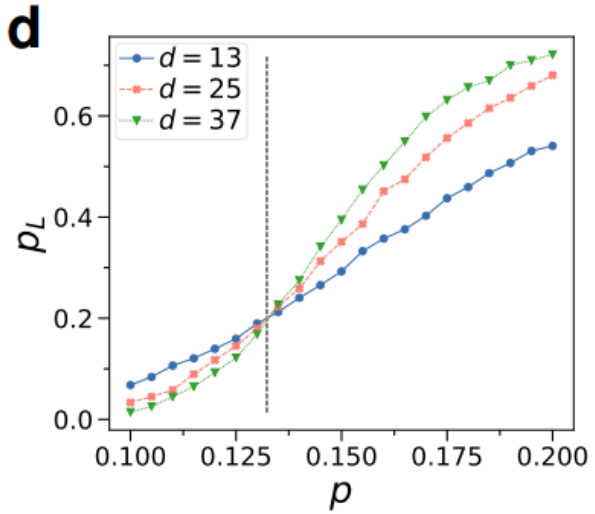
- Intrinsic tradeoff speed vs accuracy
- Also to consider: should run on specialized fast hardware (ASIC, FPGAs)
- For superconducting qubits, decoder needs to run on <µs scale
- Most popular choices:
  - minimum weight perfect matching (MWPM) decoder (graph problem, only works for surface code, not runnable on FGPAs?)
  - Belief propagation (BP): O(n) speed, but often has convergence problems (recent advances on that)
  - Neural network decoders: Fast&accurate, but need to be trained (scaling???)
  - Recent: Tessaract decoder vs BP-Vibe decoding https://scirate.com/arxiv/2508.15743#2575



Decoder can make a difference

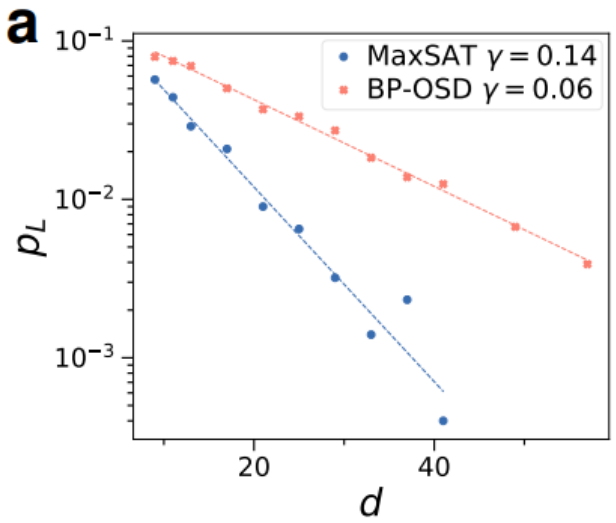


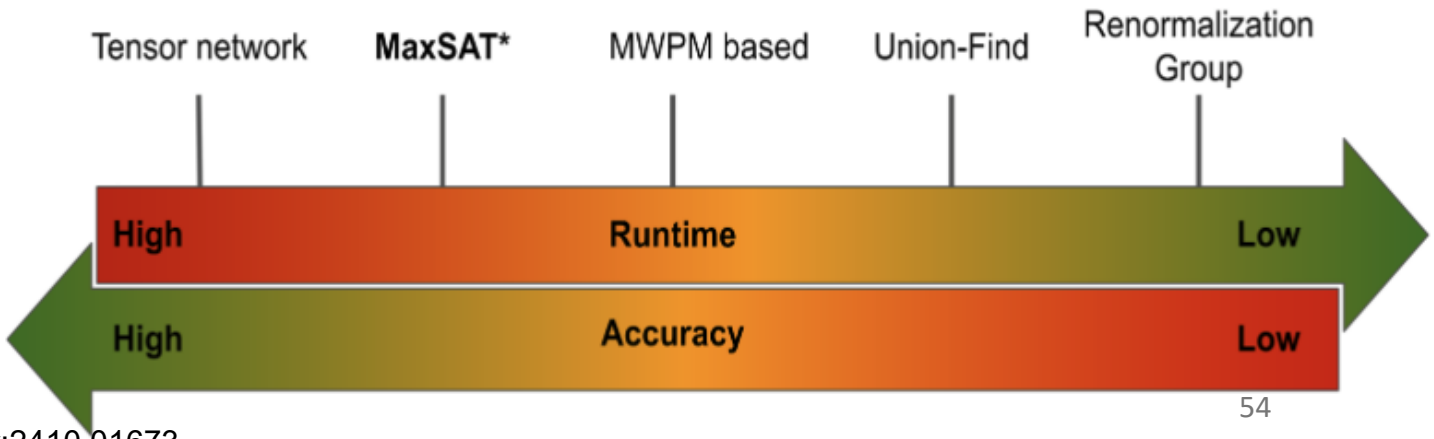
Max-SAT (slow but accurate)



BP-OSD (faster but less accurate)

- Logical error against physical error for different distances d
- Threshold: phase transition in logical error (below p_{th} logical error decreases with *d*)
- Choice of decoders makes a difference!





M Noormandipour, T Haug "MaxSAT decoders for arbitrary CSS codes" arXiv:2410.01673

Real-time decoding

- Decoders need to decode+correct while the quantum computer is running!
- Gate time for superconducting qubits <1µs

→Decoding needs to be extremely fast, including

  - read-out
  - Transfer of data (→finite speed of signals, e.g. 1km cable at speed of light → 3.3µs)
  - Decoding algorithm
  - Application of correction operator

- Future quantum computers have 100 000s of qubits → decoding problem must scale

  (even O(n) may be too slow)

→Parallelise, specialized hardware (FPGA, ASIC)

- To consider: Time to decode can increase with more errors

  negative feedback loop : More errors →more time to decode → More errors built-up while waiting for decoder → Even more errors

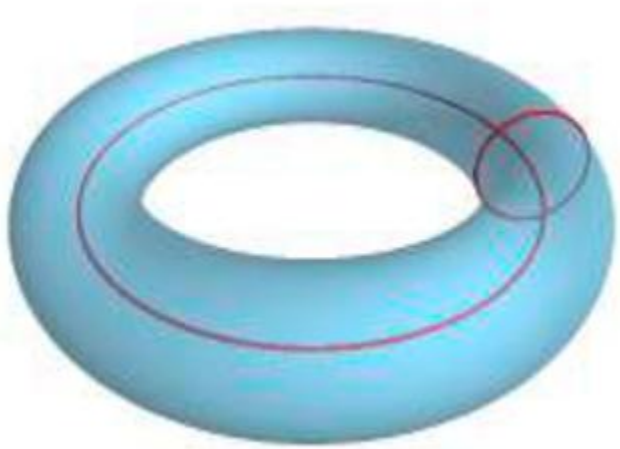
deMarti iOlius A, Fuentes P, Orús R, Crespo PM, Martinez JE. Decoding algorithms for surface codes. Quantum. 2024

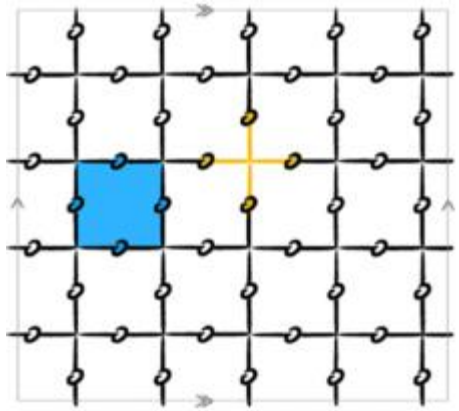
Skoric, Luka, et al. "Parallel window decoding enables scalable fault tolerant quantum computation." *Nature Communications* 14.1 (2023): 7040.

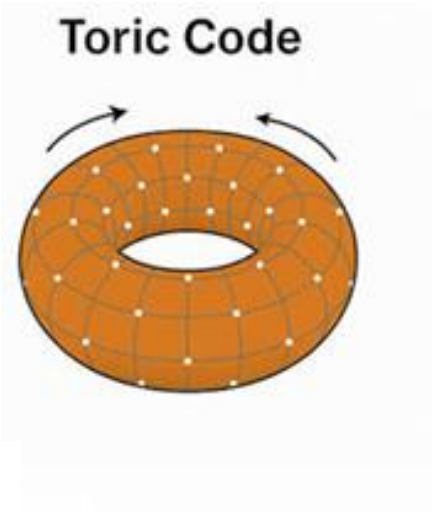
Next: How to protect in practice

The most important CSS code: Toric code

It is a square lattice with periodic boundaries







Identify the left and right edges, and the top and bottom edges.

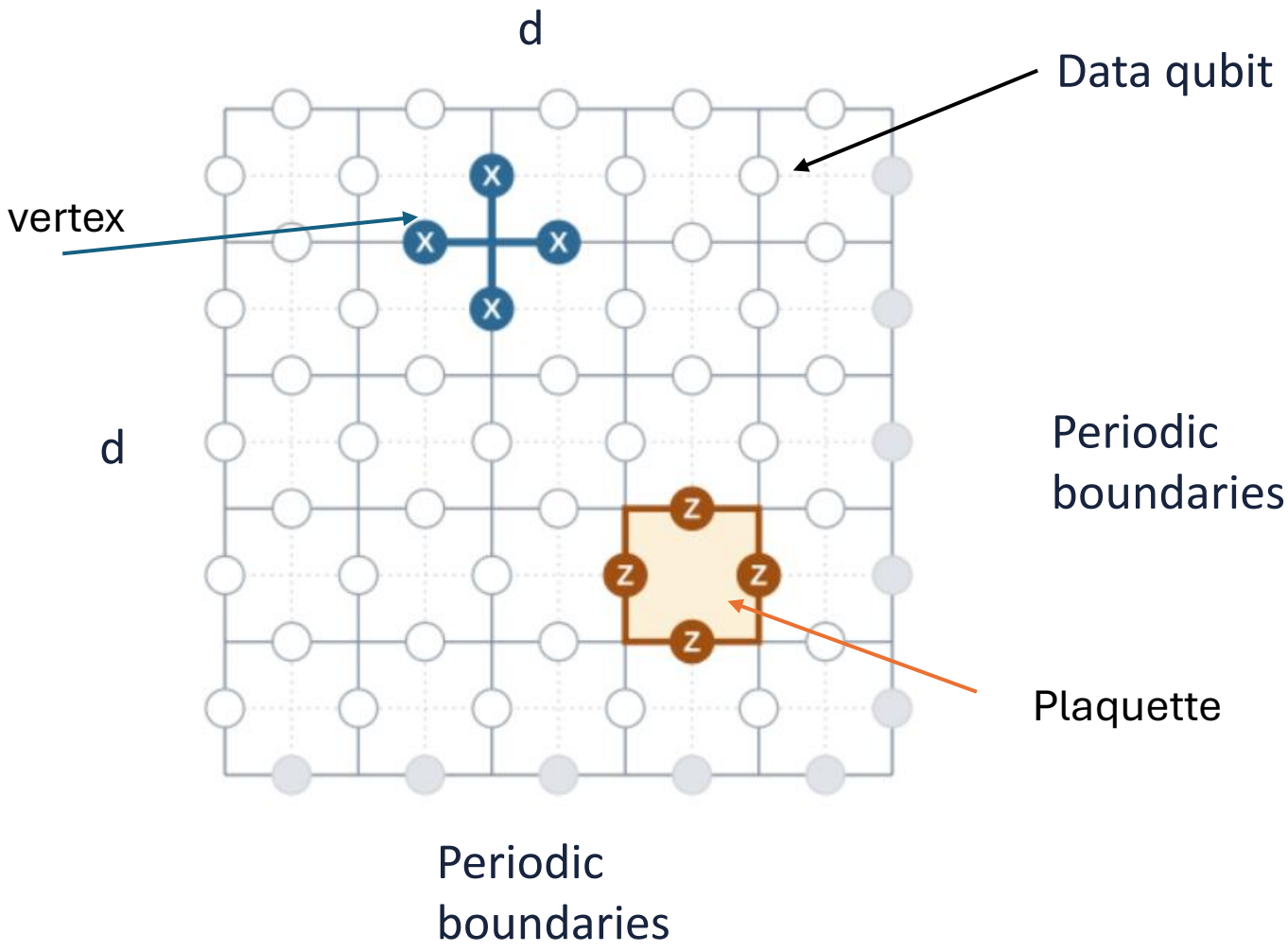
This is the topology of a torus.

Some closed loops can shrink to a point; others wind around the torus.

Can either loop around a handle shrink to a point?

Figures: K. Bharti, Asia-Pacific QEC tutorial (2024)

Toric code



An d × d periodic lattice has d^2 horizontal edges and d^2 vertical edges →$2L^2$ data qubits

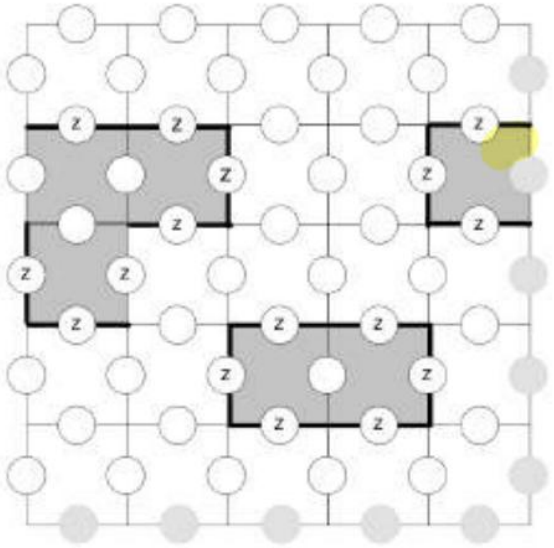
Stabilizer checks involve 4 neighbouring data qubits

X: vertex
Z: plaquette

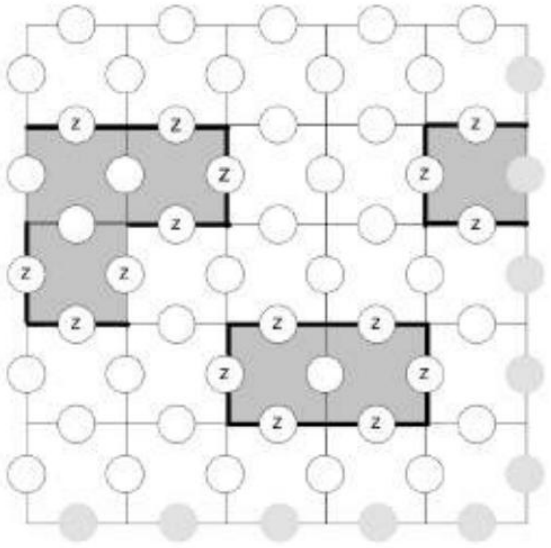
$[[2d^2,2,d]]$

Multiplying checks cancels the shared edge

Local face operators



Their product



Every shared edge receives Z twice.

Those operators cancel.

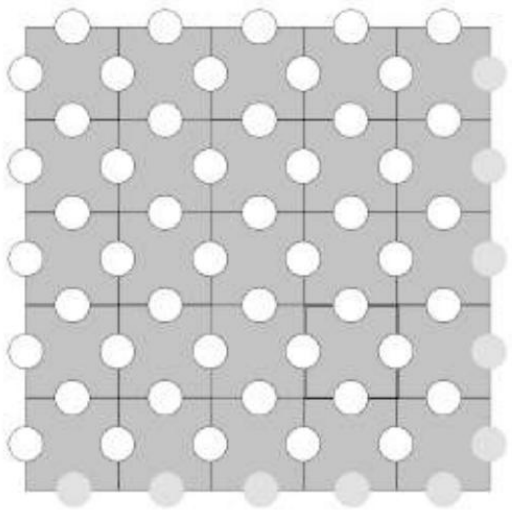
Only the boundary of the combined region remains.

$$Z_e^2 = I, \qquad B_{p_1} B_{p_2} = \prod_{e \in \partial(p_1 \cup p_2)} Z_e$$

What would survive for a larger connected region?

Figures: K. Bharti, Asia-Pacific QEC tutorial (2024)

Two global relations leave two logical qubits



$$n = 2L^2, \quad r = 2L^2 - 2, \quad k = n - r = 2$$

The product of all face checks is identity. The same holds for all vertex checks.

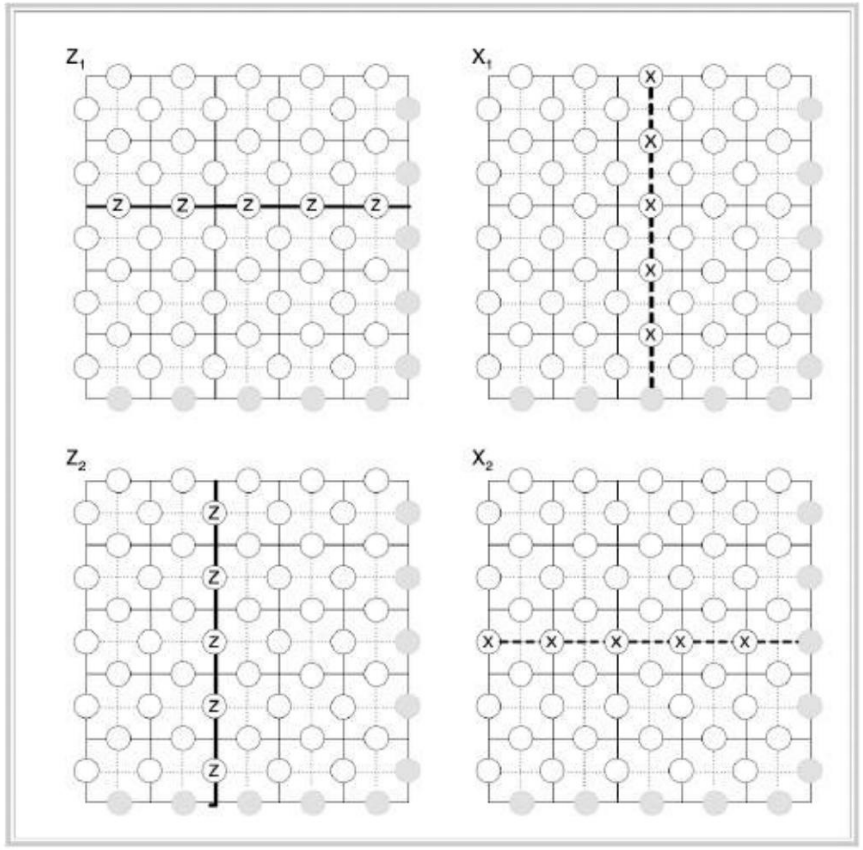
There are $2L^2 - 2$ independent checks.

Two logical qubits remain.

Why does the product of all face checks act trivially?

Figures: K. Bharti, Asia-Pacific QEC tutorial (2024)

Logical qubits of toric code



Dennis et al., Fig. 3

- Closed strings have no syndrome endpoints.
- They are not stabilizers

→A loop winding around the torus is a logical operator.

A winding Z/X string commutes with all checks.

Share odd number of qubits:

$$[\bar{X}_1, \bar{Z}_1] = -2i\bar{Y}_1$$

$$[\bar{X}_2, \bar{Z}_2] = -2i\bar{Y}_2$$

Share even number of qubits:

$$[\bar{X}_1, \bar{Z}_2] = 0$$

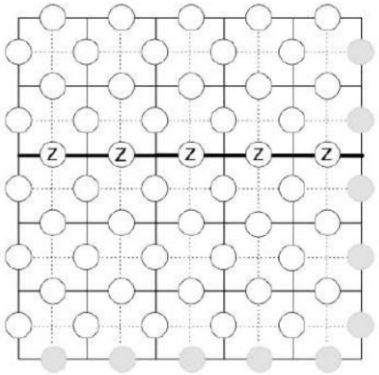
$$[\bar{X}_2, \bar{Z}_1] = 0$$

→ Commutation relationship of two qubits

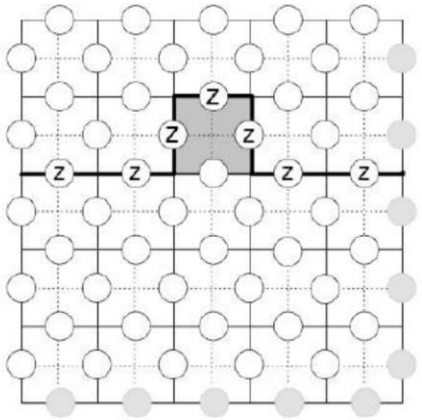
- Distance d is shortest logical operator, i.e. width of lattice (here d=5)

Multiplying a stabilizer deforms a logical string

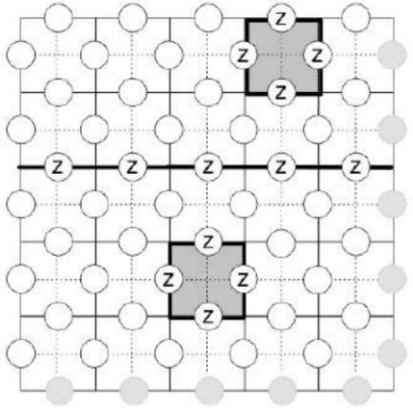
One winding string



Multiply a local check



A different representative



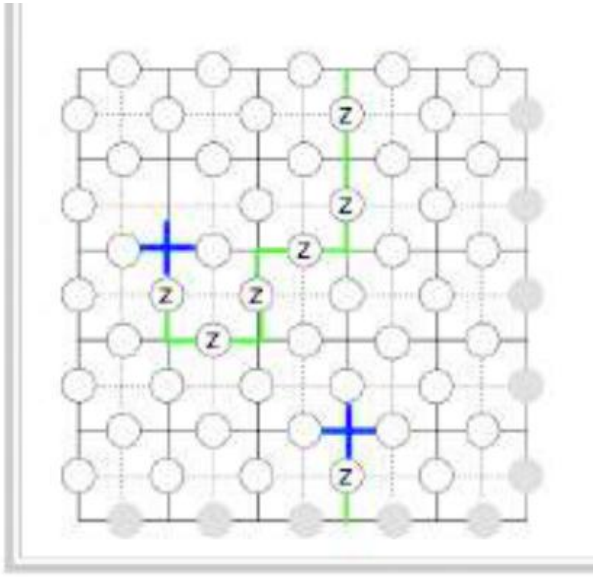
The geometric path changes.

Its action on the logical state stays the same.

Can a sequence of stabilizers remove the winding?

Figures: K. Bharti, Asia-Pacific QEC tutorial (2024)

Decoding surface-code defects



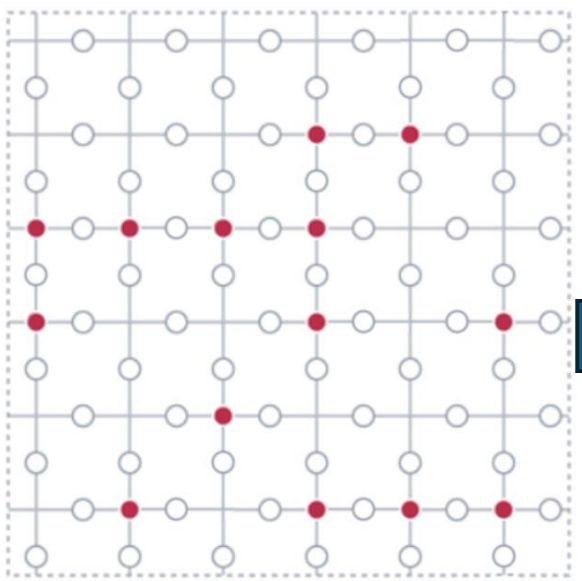
Errors (green) generate a flipped syndrome at endpoints: The blue vertices.
Note that only the syndromes (blue) are observed, but not the error path in green.

Given syndrome, there are many possible green paths.

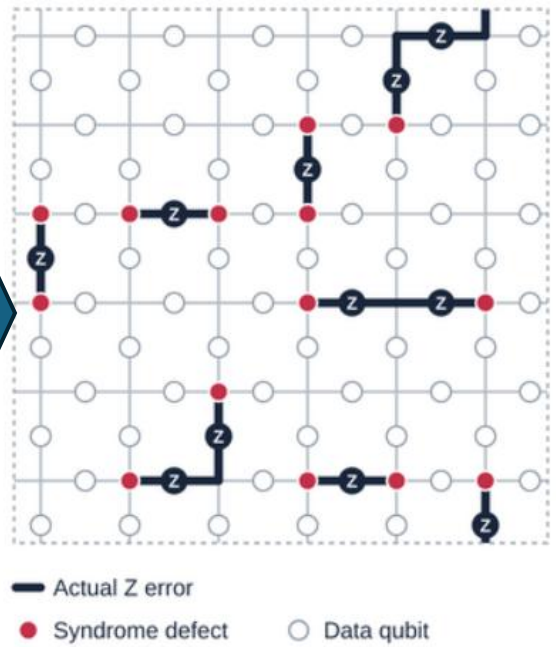
Decoder needs to find the most-likely error, which is the shortest path connecting the vertices.

Dennis et al., Fig. 6

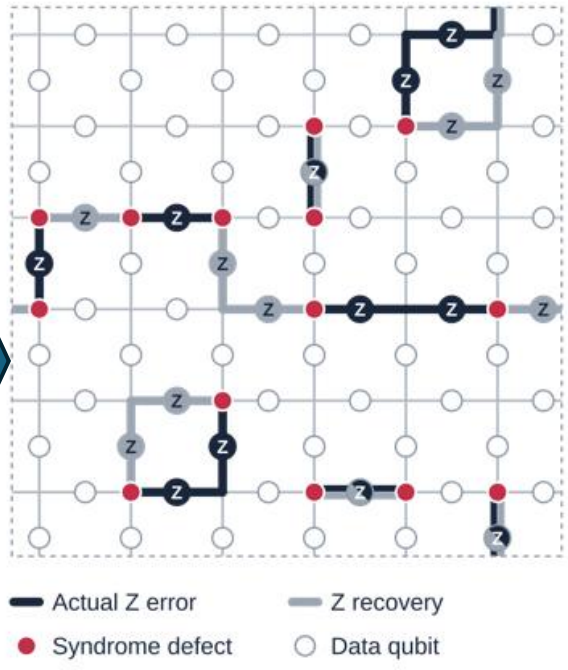
A decoder can close the wrong loop



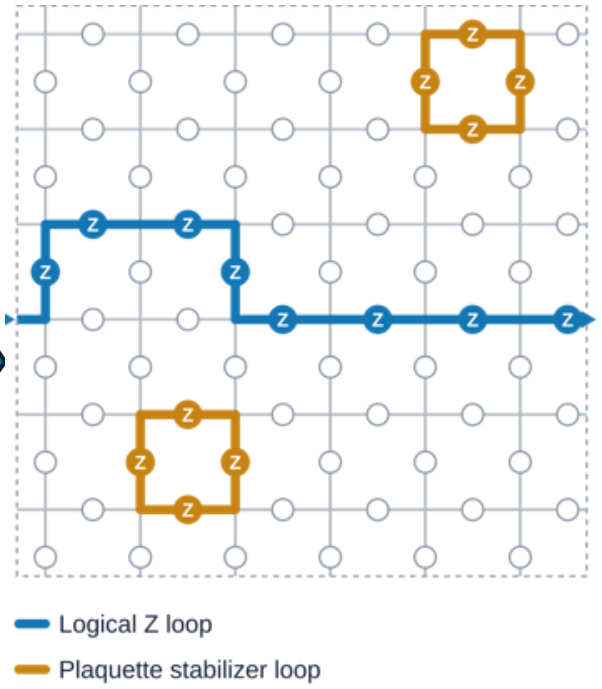
Decoder sees only syndrome (red dots)



Black: Actual errors



Grey: our guess



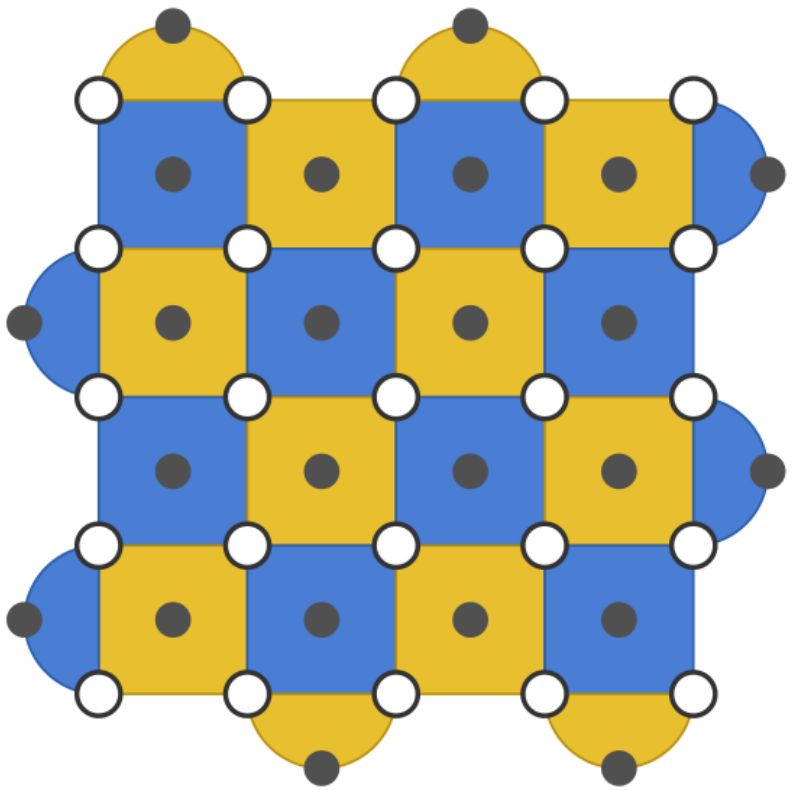
After recovery

- Dark edges show the actual errors (invisible in exp.)
- We only have syndromes (red dots), from which we must choose best recovery.
- Light edges show an example recovery operation (i.e. Pauli operators).

- Add (mod 2) the actual error and recovery

Would another ideal round of check measurements expose the error? No.

The rotated surface code



White circles: data qubits
Gray circles: measurement ancillas

Blue regions measure Z checks.
Yellow regions measure X checks.

Each check touches 4 data qubits in the interior, or 2 at the edge.

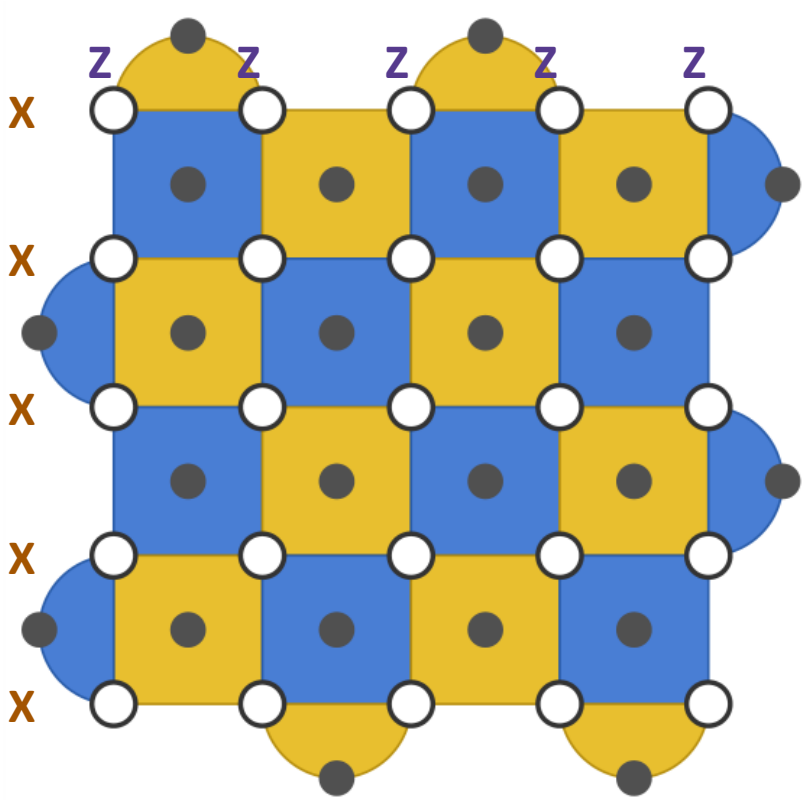
$$[[n, k, d]] = [[d^2, 1, d]]$$

Here d = 5: 25 data qubits, 24 independent checks, one logical qubit.

Which qubits store the state, and which only measure checks?

Figure: Spencer et al., arXiv:2605.29137, Fig. 10b

Logical operators cross the patch



Logical Z: apply Z along the top row.

Logical X: apply X down the left column.

Each string overlaps every opposite-type check on 0 or 2 data qubits.

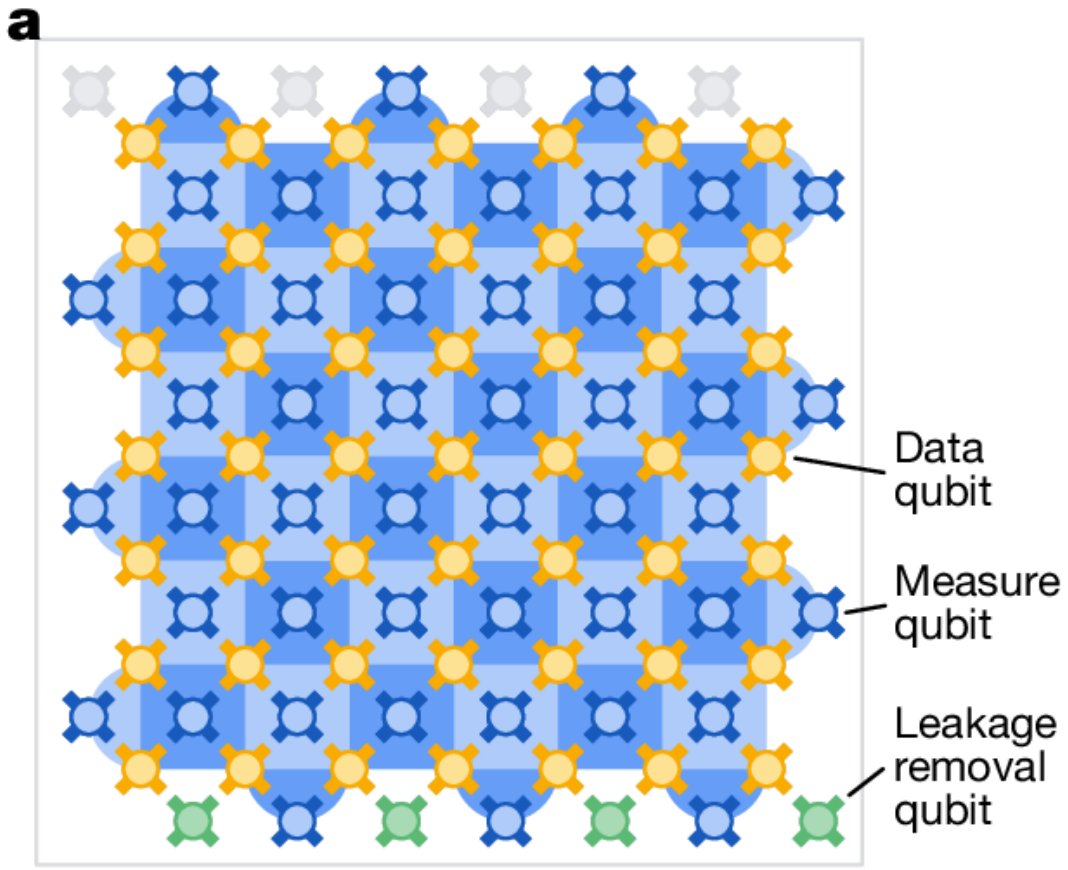
The two strings share the top-left qubit.

$$\overline{X}\,\overline{Z} = -\overline{Z}\,\overline{X}$$

Both shortest logical strings have weight 5, so the distance is 5.

Same patch as previous slide. Figure: Spencer et al., Fig. 10b

Google’s Willow surface-code memory



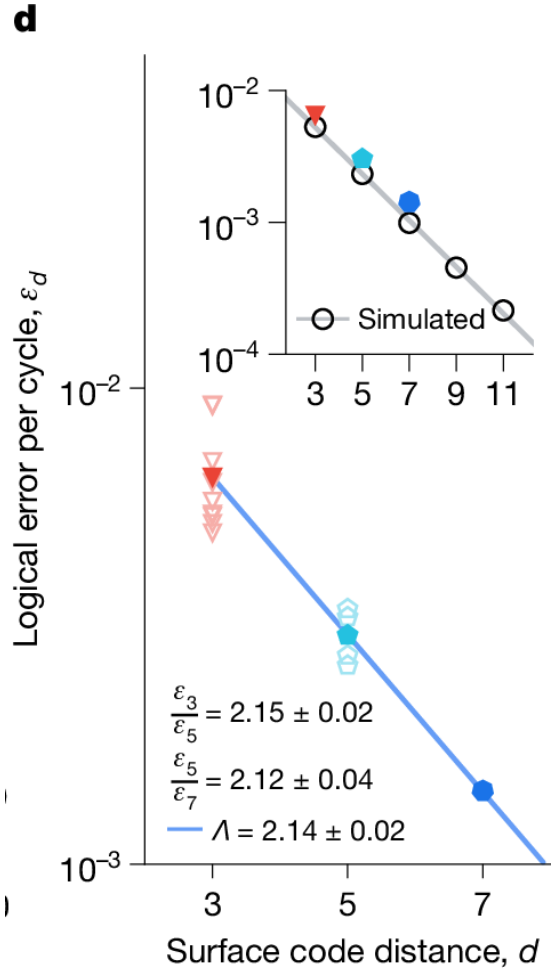
Distance 7 on a superconducting processor

49 data qubits
48 measurement ancillas
4 extra leakage-removal qubits

101 qubits participate in the memory experiment.

Google Quantum AI, Nature 638, 920–926 (2025), Fig. 1a. Announced December 2024.

Below threshold: larger codes make fewer logical errors



Horizontal axis: code distance
Vertical axis: logical error per cycle (logarithmic scale)

Distance 3, 5, 7
Each step roughly halves the error rate.

$$\Lambda = \varepsilon_d / \varepsilon_{d+2} = 2.14 \pm 0.02$$

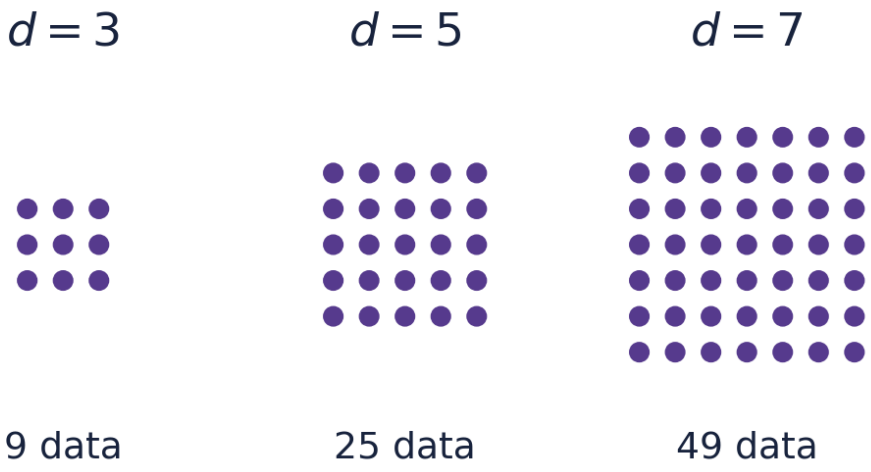
At d = 7: 0.143% error per cycle.
Logical lifetime: 2.4× the best physical qubit.

Why does adding qubits help only when physical noise is low enough?

Google Quantum AI, Nature 638, 920–926 (2025), Fig. 1d

Next: Achieving fault-tolerance

How many qubits does a protected memory need?



Each patch stores one logical qubit.

A rotated surface-code patch uses d^2 data qubits to encode one logical qubit.

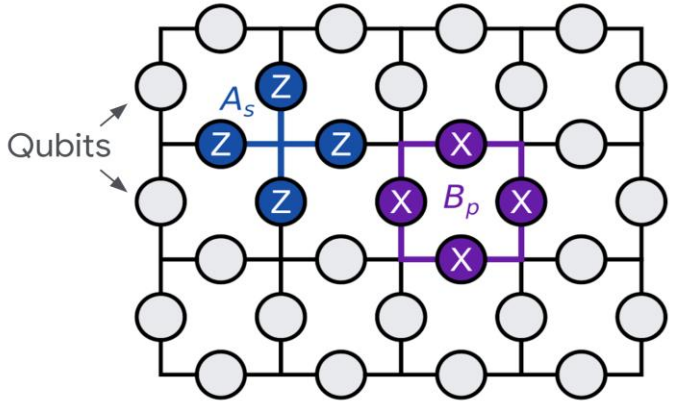
At distance 7, storing 100 logical qubits takes 4,900 data qubits, plus qubits to measure the checks.

Could a single block share that redundancy among many logical qubits?

The encoding rate k/n counts logical qubits per data qubit.

Challenge: Easy to implement vs number of qubits

- Standard code choice: **Surface code**

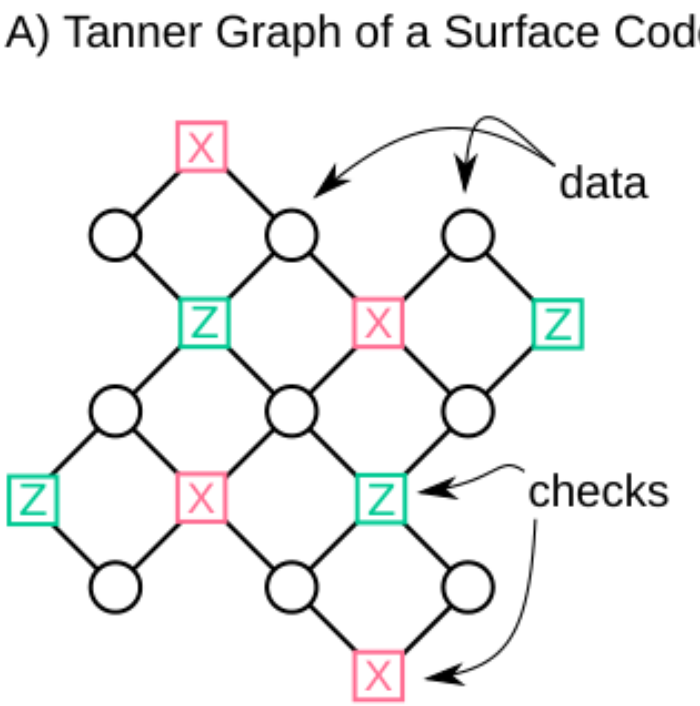


- **Why**? 2D local implementation, know how to implement Clifford gates, good&fast decoders, …
- **Downside**: Low encoding rate: ~1000+ qubits per logical qubit needed to run algorithms…

- Alternatives with better encoding rate, yet still practical?

→Quantum low-weight density parity check (QLDPC) codes

What makes a quantum code LDPC?



Bravyi et al., Fig. 1A

A Tanner graph connects a check to each data qubit it touches.

LDPC:

- each check touches at most a constant number of qubits.
- Each qubit participates in at most a constant number of checks.

The surface code is a LDPC code

More general LDPC codes can have better code properties, but there is a price to pay!

Beyond surface codes

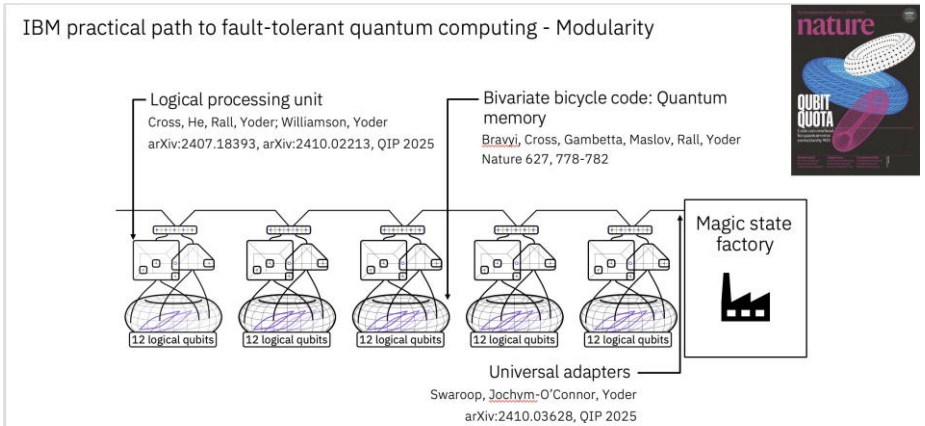
Article

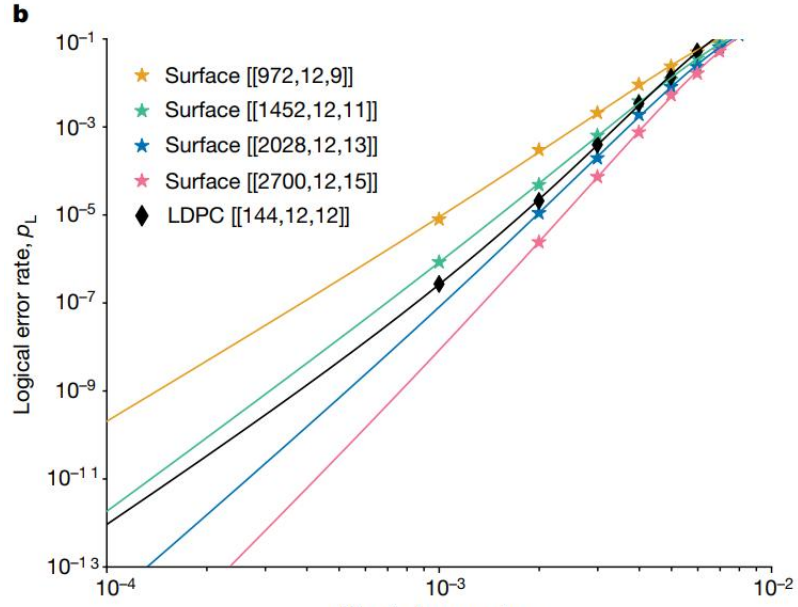
High-threshold and low-overhead fault-tolerant quantum memory

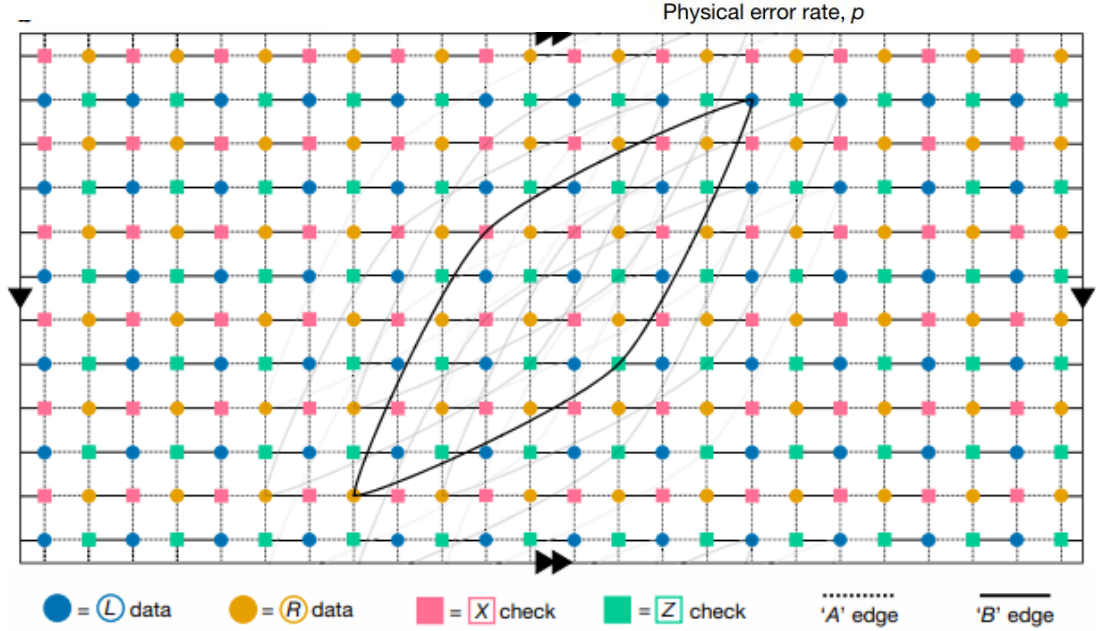
https://doi.org/10.1038/s41586-024-07107-7
Received: 25 August 2023

Sergey Bravyi[1], Andrew W. Cross[1], Jay M. Gambetta[1], Dmitri Maslov[1,2], Patrick Rall[3] & Theodore J. Yoder[1]

- Quantum low-weight density parity check (QLDPC) codes: Bivariate bicycle (BB) codes
- Promise much higher encoding rates than surface codes (e.g. Gross code[[144,12,12]] by Bravyi et al.)
- Prescriptions to implement Clifford gates
- Roadmap to large-scale hardware implementation







Yoder, Theodore J., et al. "Tour de gross: A modular quantum computer based on bivariate bicycle codes." *arXiv preprint arXiv:2506.03094* (2025).

Eberhardt, Jens Niklas, and Vincent Steffan. "Logical operators and fold-transversal gates of bivariate bicycle codes." *IEEE Transactions on Information Theory* (2024).

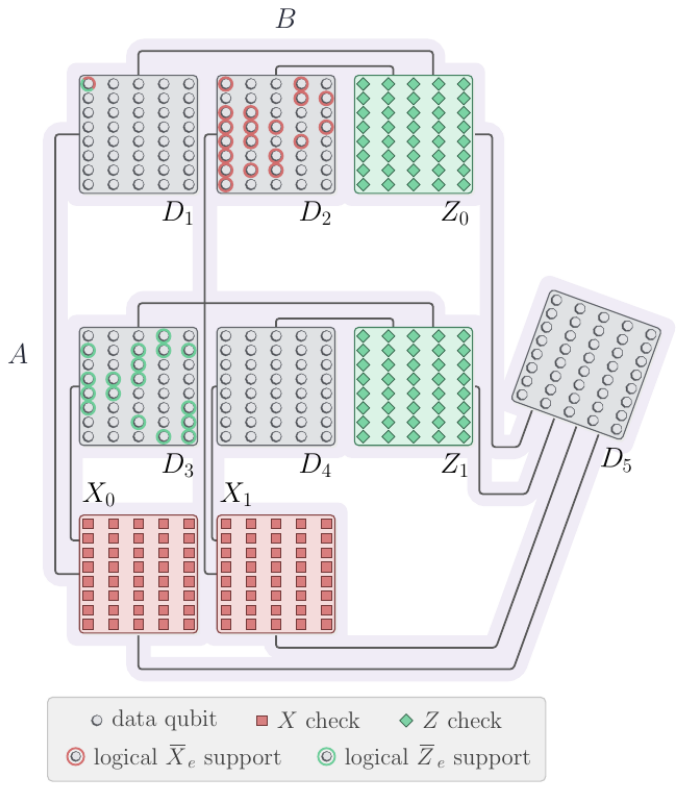
A. A. Kovalev and L. P. Pryadko, Quantum kronecker sum-product low-density parity-check codes with finite rate, Physical Review A 88, 012311 (2013)

S. Bravyi, A. W. Cross, J. M. Gambetta, D. Maslov, P. Rall, and T. J. Yoder, High-threshold and low-overhead faulttolerant quantum memory, Nature 627, 778 (2024).

Why look for other qLDPC codes?

Code family	Logical rate k/n	Geometric requirement
Surface code	k=1, k/n= $1/d^2$	Checks local
Asymptotically good qLDPC codes	k/n ~const, d/n~const	Non-local connectivity, but usually high-weight checks 20+
Recently: High-rate qLDPC codes	k/n ~const (e.g. ½, 1/5), d~20+	• Low check weight with geometry adapted to hardware constraints • Highly suited for Rydberg atoms/superconducting • d sufficiently high for all practical purposes

The Mitten code stores 40 logical qubits in one block



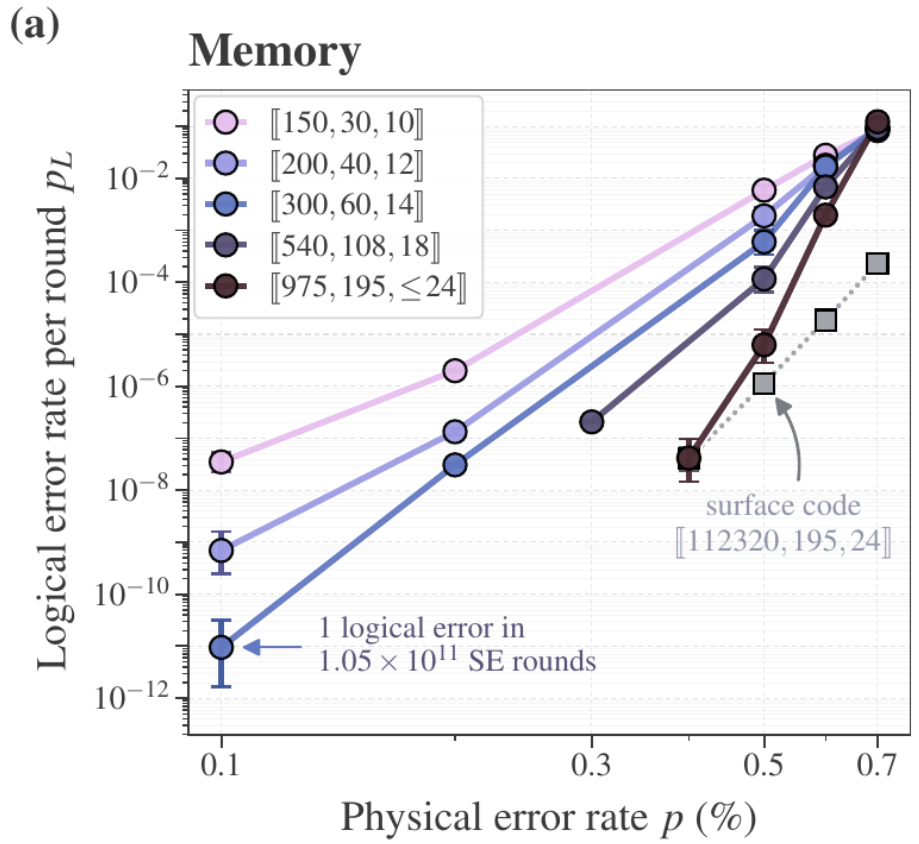
Bhardwaj et al., arXiv:2607.28795, Fig. 1a

The [[200,40,12]] example has five groups of 40 data qubits each. Its encoding rate is 40/200 = 20%.

Each X or Z check acts on nine qubits. The five groups appear as four similar “fingers” and one “thumb” in the check matrices.

$[\![n, k, d]\!]$	group G	$\mathrm{wt}(Lx)/\,\mathrm{wt}(Lz)$	$\bar{X}\bar{X}$-gadget (n_{anc}, w_{merge})	Neutral-Atom SE cycle (ms) (4-AOD, 2-AOD)	Superconducting Complexity C_{hw}
$[\![150, 30, 10]\!]$	$C_5 \times S_3$	18/10	(78, 11)	(5.49, 8.83)	2.02
$[\![200, 40, 12]\!]$	$C_4 \times D_{10}$	20/18	(88, 11)	(7.41, 10.79)	2.09
$[\![300, 60, 14]\!]$	$C_{10} \times S_3$	22/22	(100, 12)	(7.22, 10.98)	2.28
$[\![500, 100, 16]\!]$	$C_5 \rtimes C_{20}$	28/24	(156, 12)	(10.61, 16.15)	2.60
$[\![540, 108, 18]\!]$	$C_9 \rtimes C_{12}$	22/28	(123, 10)	(11.74, 18.41)	2.72
$[\![630, 126, \leq 20]\!]$	$C_7 \rtimes C_{18}$	28/44	(134, 10)	(11.47, 19.00)	2.98
$[\![780, 156, \leq 22]\!]$	$C_{13} \times A_4$	74/84	(435, 12)	(14.90, 23.24)	3.08
$[\![975, 195, \leq 24]\!]$	$C_{13} \rtimes C_{15}$	102/92	(622, 12)	(14.60, 22.70)	3.40

How well does a Mitten block protect its memory?



Bhardwaj et al., arXiv:2607.28795, Fig. 2a. Circuit simulation.

Each curve shows the simulated probability of a logical error anywhere in one block per check round.

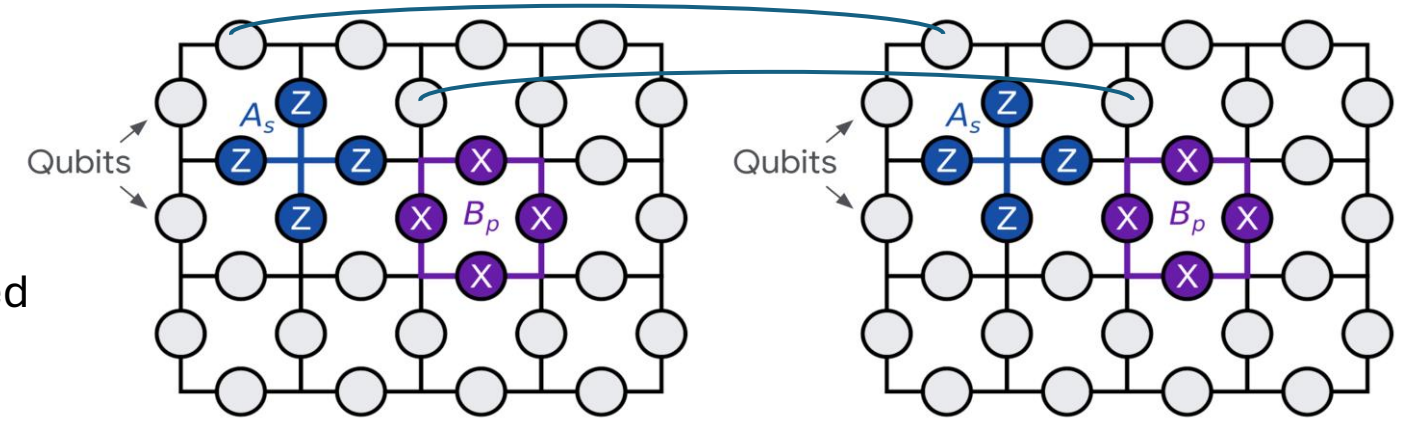
For [[300,60,14]], the authors report about 10^{-11} per round at 0.1% physical error.

The simulation uses circuit-level depolarizing noise and assumes ideal idle qubits.

What´s the deal with magic?

- So far, we talked about quantum memory experiments: Preserve quantum state from errors
- We need Fault tolerance: Implement logical gates protected against errors

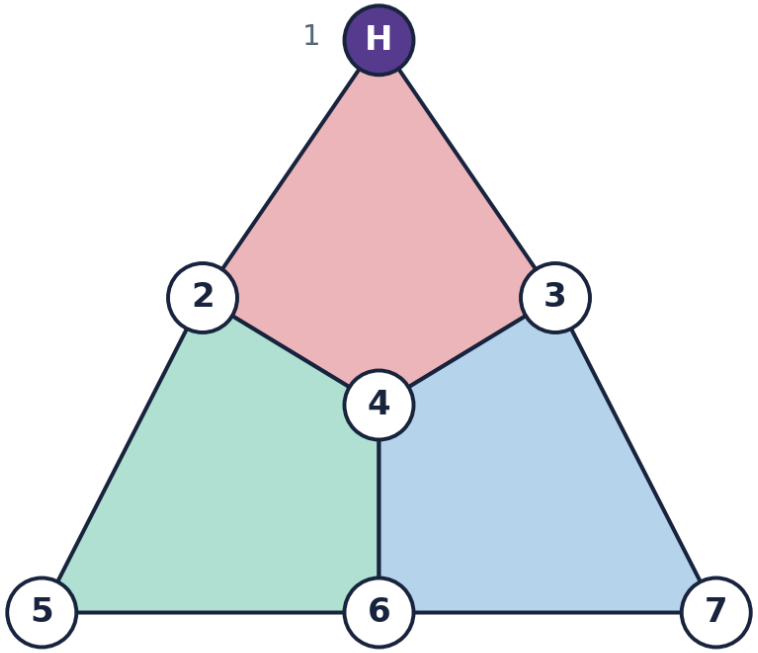
- Transversal: Gates that can be implemented with single-qubit operations on each block
- Examples of transversal gates for 2D CSS codes:
  - CNOT: Implement CNOT gate between each pair of physical qubits of two surface codes
  - Hadamard: apply physical Hadamard on each qubit (+relabeling of qubits)



However: Transversal universal gate set not allowed
E.g. T-gates in 2D codes
(Eastin and Knill (2008) arXiv:0811.4262)

How NOT to implement Hadamard gate

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

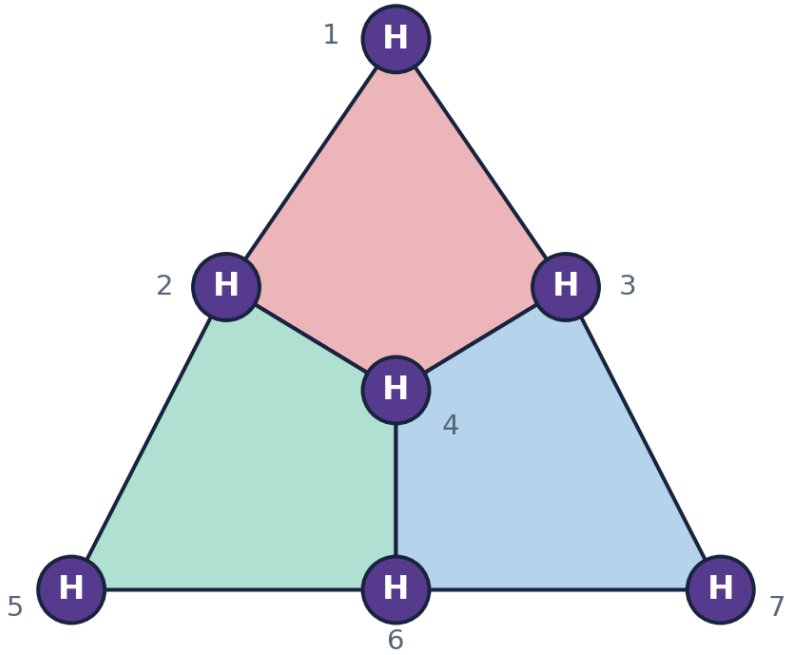


The Steane code encodes one logical qubit in seven physical ones. The color indicates the checks; e.g. the top face has an X check and a Z check on qubits 1, 2, 3 and 4.

On qubit 1, H = (X + Z)/√2. The X term flips the Z check.

The Z term flips the X check→ Error

How to actually implement Hadamard gate



Seven-qubit triangular color code, equivalent to the Steane [[7,1,3]] code.

How to do logical Hadamard: Apply H to all seven qubits in parallel.
This is a transversal gate as it touches only one qubit at a time

On the top face, HXH = Z turns the four-qubit X check into its Z check. Likewise, HZH = X turns the Z check into the X check.

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Seven physical Hadamards implement one logical Hadamard

$$U = H^{\otimes 7}$$

$$U\overline{X}U^{\dagger} = \overline{Z}$$

$$U\overline{Z}U^{\dagger} = \overline{X}$$

$$U|0_L\rangle = |+_L\rangle$$

Choose logical X to be X on all seven qubits, and logical Z to be Z on all seven. They commute with every face check and anticommute with each other.

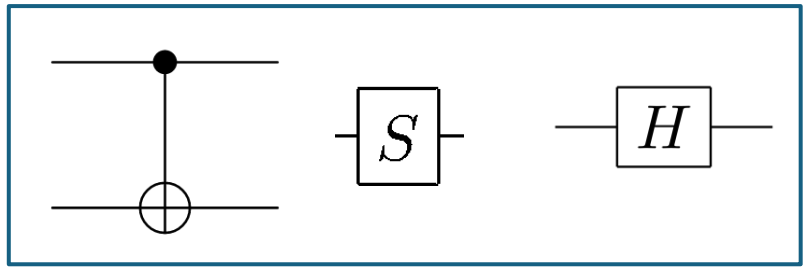
The physical Hadamards exchange these two logical Paulis. On the encoded qubit, that action is a Hadamard up to a global phase.

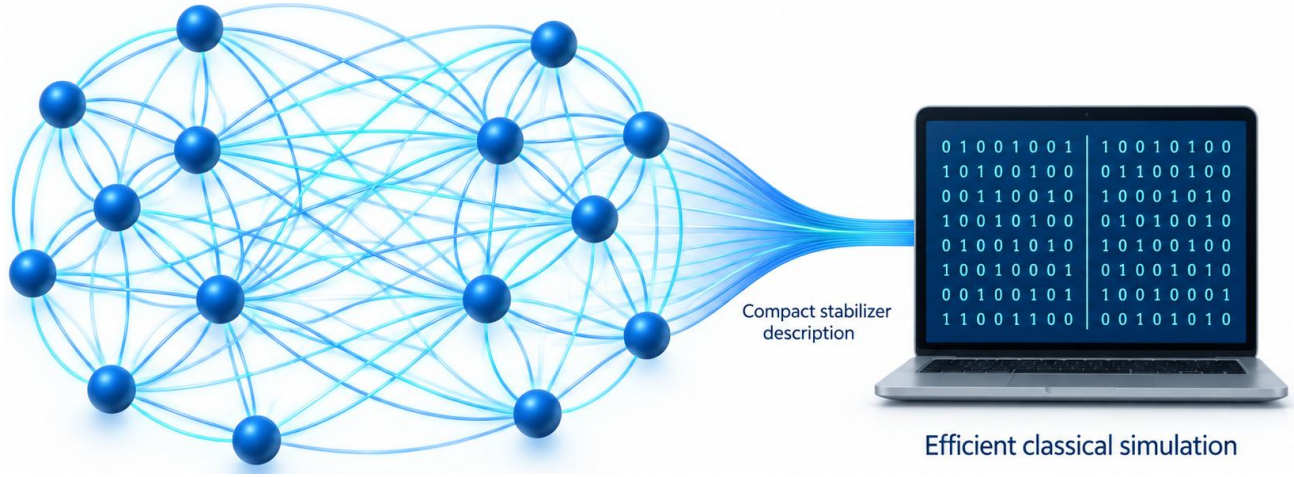
$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

The +1 eigenstate of logical Z becomes the +1 eigenstate of logical X.

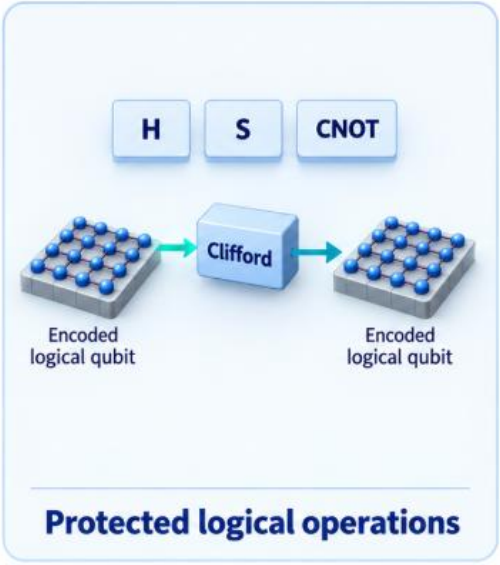
What characterizes complexity of quantum computing?

Clifford gates (non-universal):





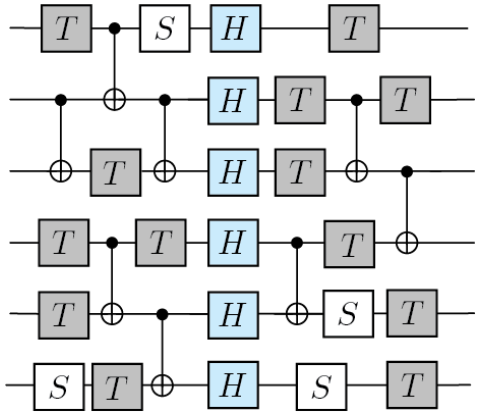
- Generate high entanglement
- But:
  - Efficient classical simulation
  - Transversal implementation on fault-tolerant quantum computers

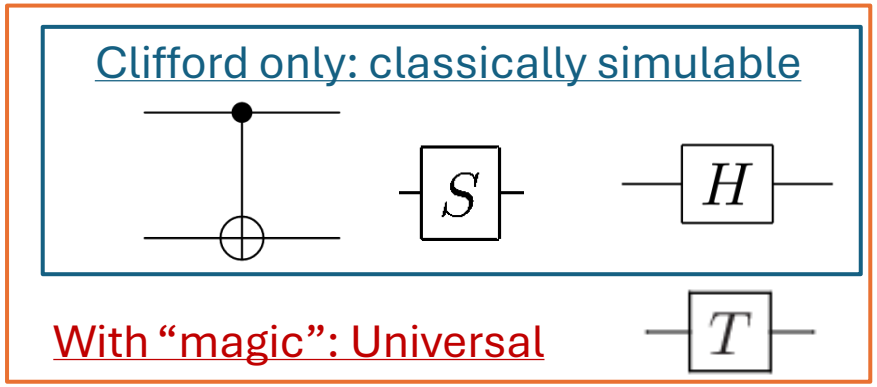


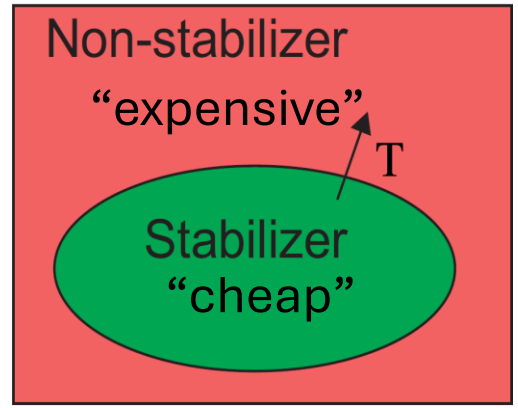
D. Gottesman *"The Heisenberg representation of quantum computers"* arXiv:quant-ph/9807006 (1998)

What makes quantum computing “magical”

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{-i\pi/4} \end{pmatrix}$$







- Clifford (CNOT, Hadamard, S-gate)
  - Fault-tolerant quantum implementation
  - Efficient classical simulation

- “Non-Clifford (“Magic”) (e.g. T-gates) 
  - Costly on fault-tolerant quantum computers (Eastin-Knill)
  - Hard to classically simulate via Clifford+T algorithms



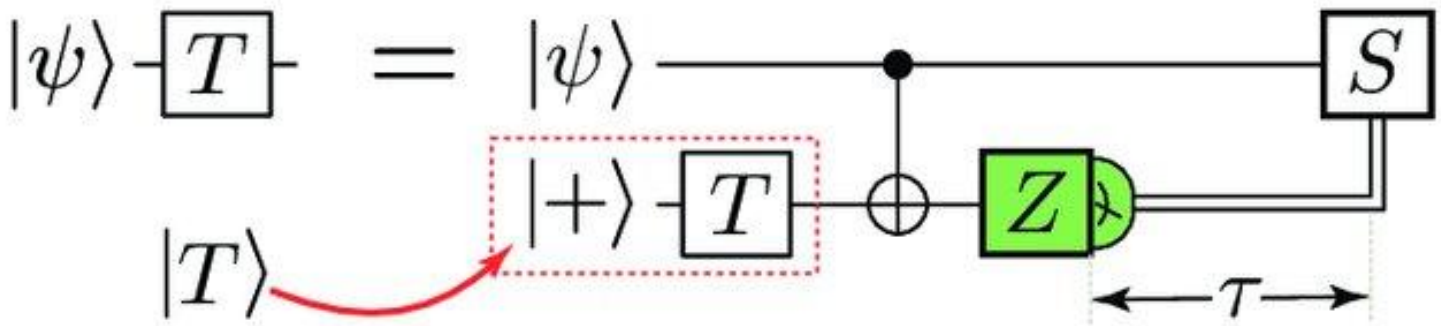
S. Bravyi, A. Kitaev. *Physical Review A* 71.2 (2005)
Howard, Mark, et al. *Nature* 510.7505 (2014): 351-355.

Turn a magic state into T-gate

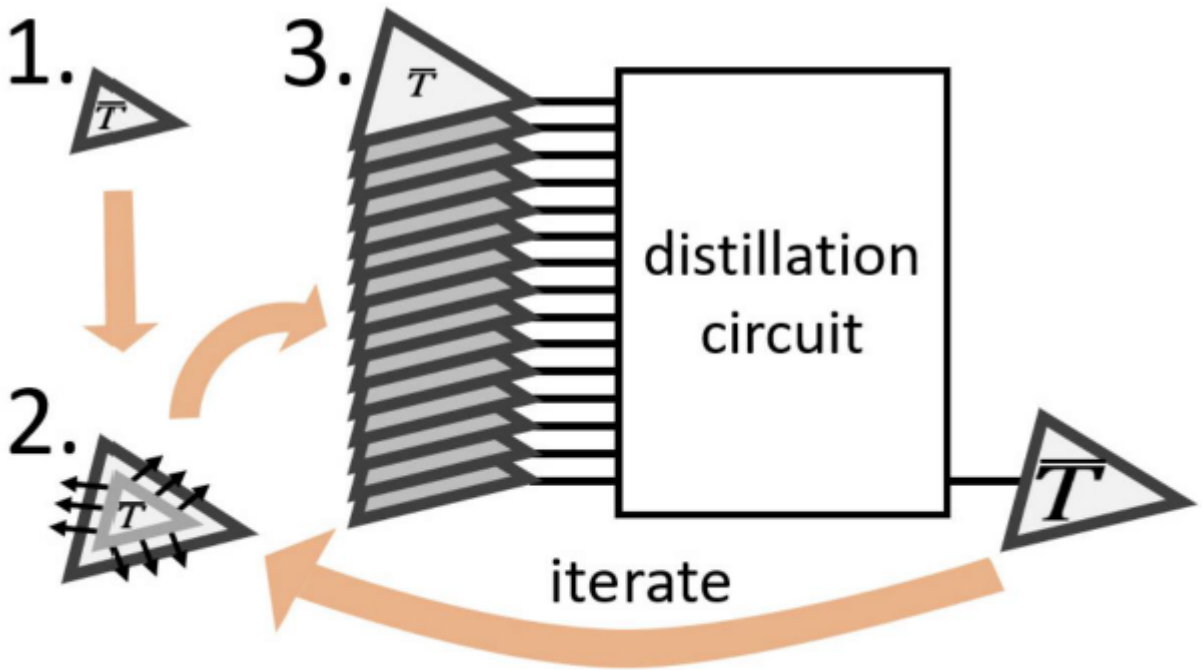
$$T = \mathrm{diag}(1, e^{i\pi/4}), \qquad |T\rangle = T|+\rangle$$

Prepare T-state $|T\rangle = |0\rangle + \exp(i\,\pi/4)|1\rangle$.
Then, T-gate (unitary!) can be implemented via an adaptive Clifford teleportation protocol

Measure the resource after CNOT	Data state, up to phase	Clifford feed-forward
0	$T\,\|\psi\rangle$	Keep the data
1	$T^\dagger\,\|\psi\rangle$	Apply S, giving $T\,\|\psi\rangle$



Distillation



Start with noisy magic states, maybe in a single physical qubit.

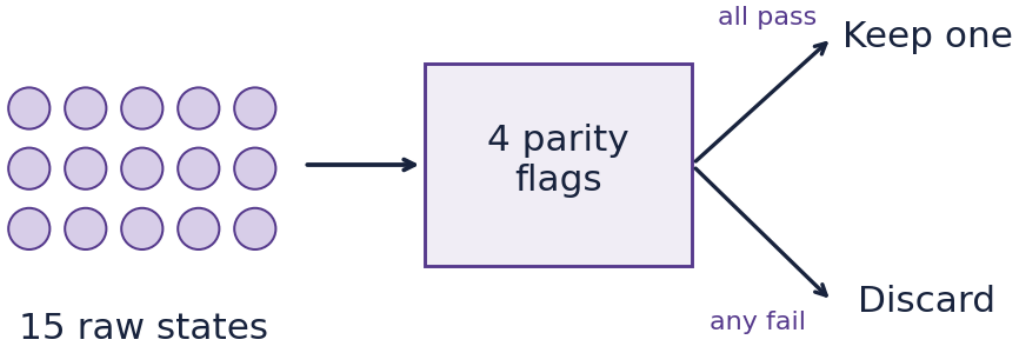
Encode them into a small code, e.g. 2D color code.

Run a Clifford circuit, which uses post-selection, to output fewer *less-noisy* magic states.

Repeat if necessary.

- To achieve 1e-15 logical error rate needed for large algorithms requires 3+ rounds of the simplest distillation protocols. **Expensive!**
- Many practical improvements have reduced the cost of distillation (Litinski) to $10\times$other gates.
- Shockingly, recent constant-rate LDPC code breakthroughs achieve constant ratio of input noisy to output #T gates of any target error rate. Impractically large.

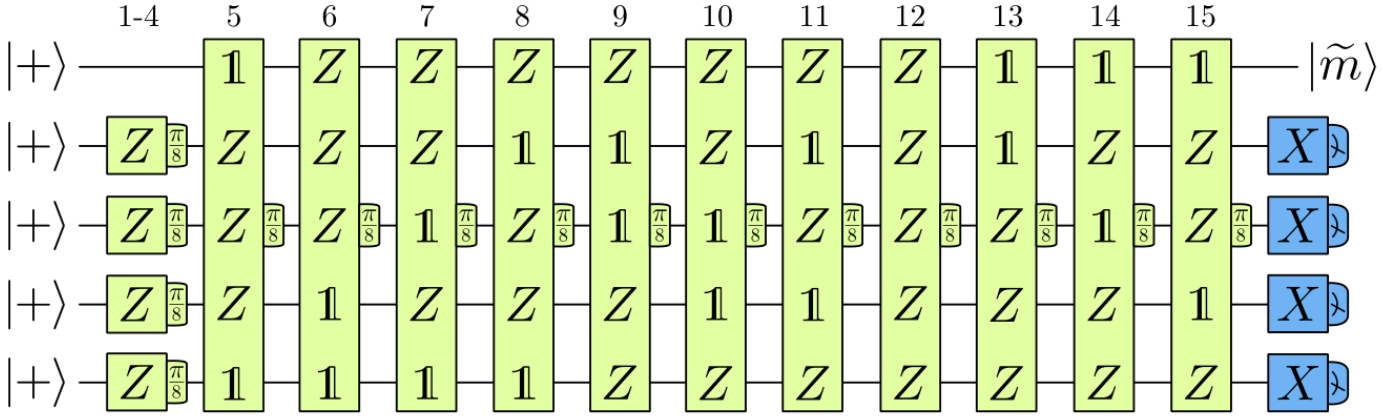
Distillation trades many noisy inputs for one cleaner state



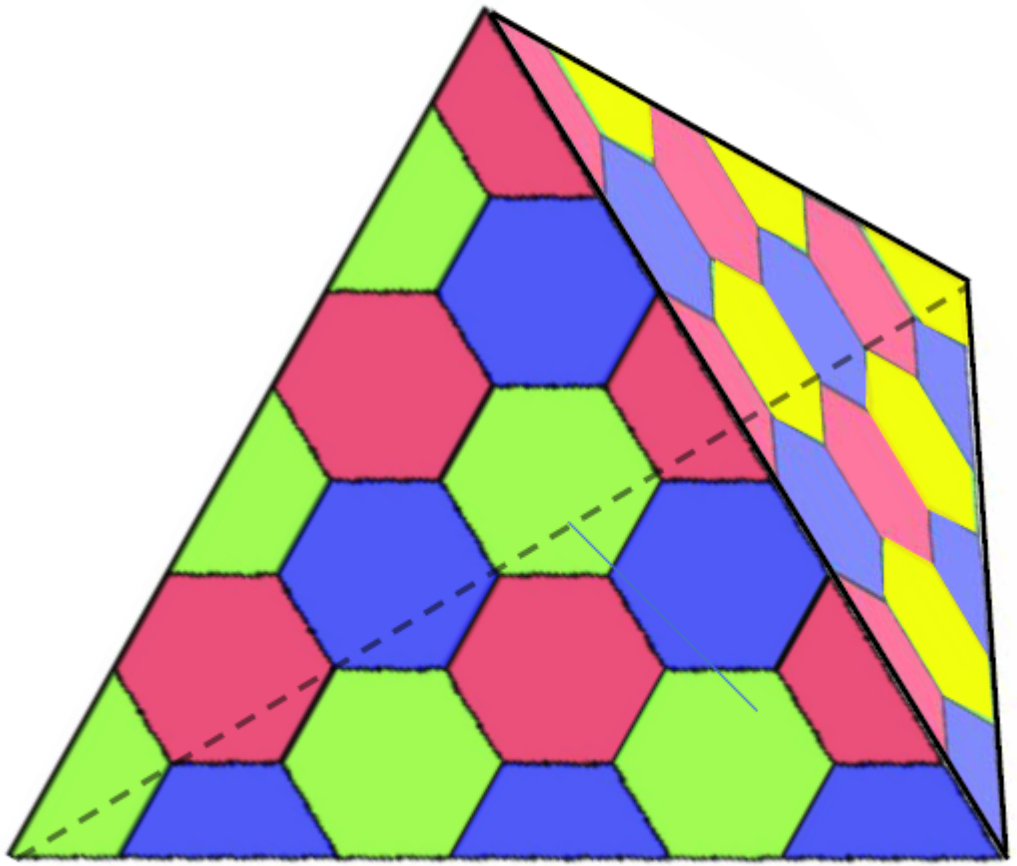
Start with 15 independently noisy magic states. The protected Clifford circuit measures four error flags.

If any flag fires, discard the batch. If all four pass, keep one state and consume it later for a logical T gate.

The accepted state has a lower error rate when input errors are sufficiently rare.



Code switching



It is sometimes possible to fault-tolerantly transfer information back and forth between codes.

e.g. possible between 2D and 3D color codes.

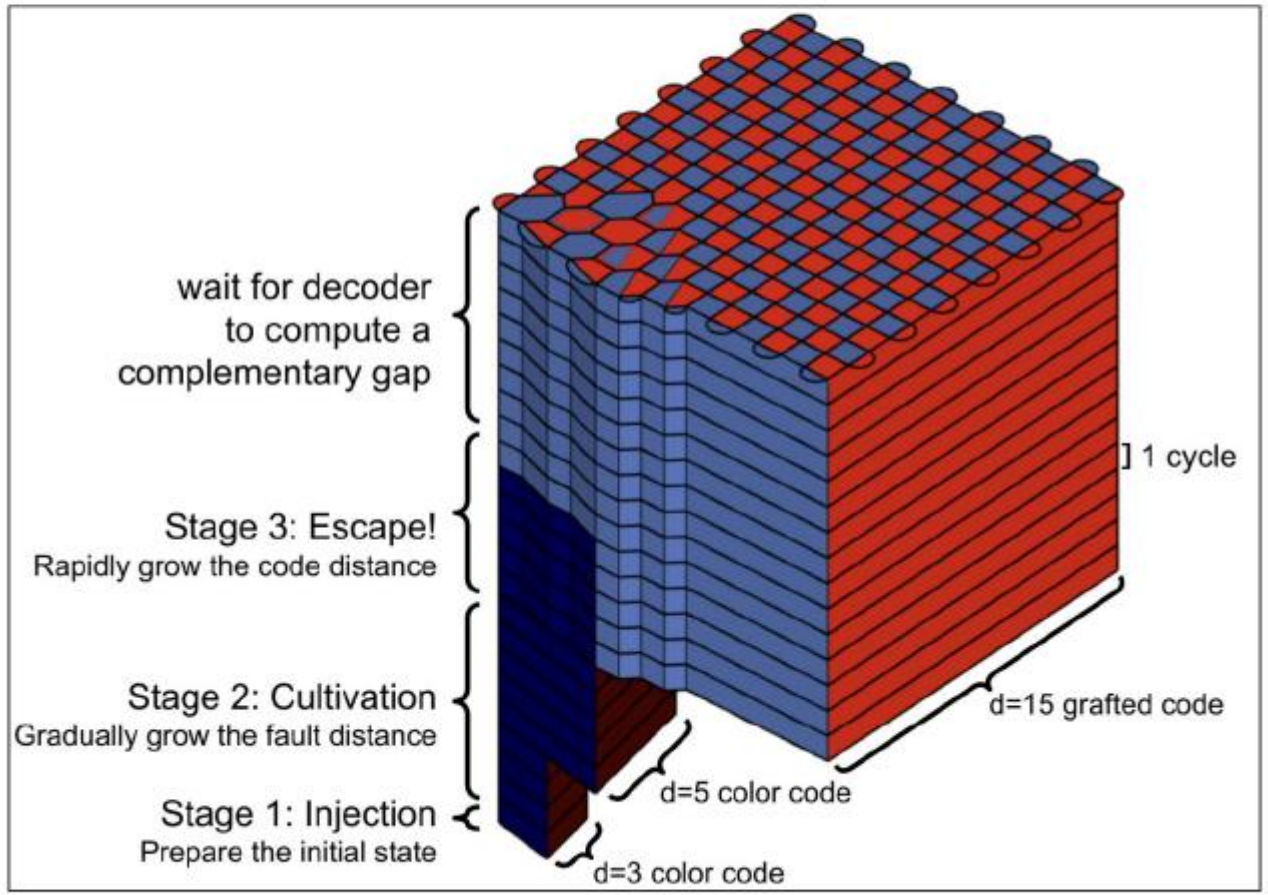
Can do Cliffords in 2D, and switch to 3D each time a T is needed.

Requires 3D rather than just 2D connectivity, and $\sim d^3$ qubits rather than $\sim d^2$, and switching has $\sim d$ time overhead.

In Beverland, Kubica and Svore (2021), we compared with distillation and unfortunately found it to be worse.

Have heard a few groups are revisiting codeswitching with some new ideas…

Cultivation



Gidney, Shutty, and Jones (2024)

A recent approach blurs the lines and uses a slowly evolving code with a more complex circuit.

Numerics-first, rather than provable. Can be done in 2D.

In practice, much much lower cost than everything else.

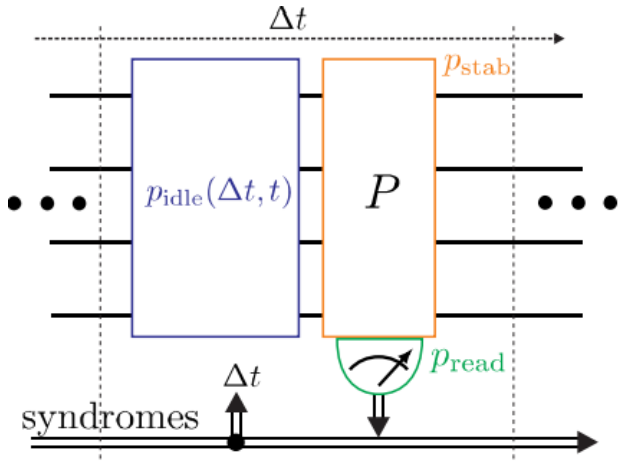
Main catch is that currently not enough – caps out at around logical error rate of 1e-8. Expect to improve…

Cultivation resource cost of magic is comparable to lattice surgery (for logical entangling gates in 2D).

Cultivation resource cost still much worse than all logical gates if you have 3D (or greater) connectivity.

Exponential logical-error reduction in quantum memories via optimal syndrome-measurement timing

Haug, Bharti and Aolita arXiv:2608.06242



QEC measures syndromes at regular intervals *Δt*

- Too seldom: Faults pile up
- Too often: Syndrome measurement introduce too many faults

Error per unit of physical time: $R = \frac{1}{\Delta t}\frac{A}{d^{\beta}}\left(\frac{p}{p_{\text{th}}}\right)^{(d+1)/2}(1+\lambda\Delta t)^{g(d+1)/2}$

$$R = \frac{p_{\text{L}}}{T}$$

→Optimal timing: $\Delta t \propto \frac{1}{d}$

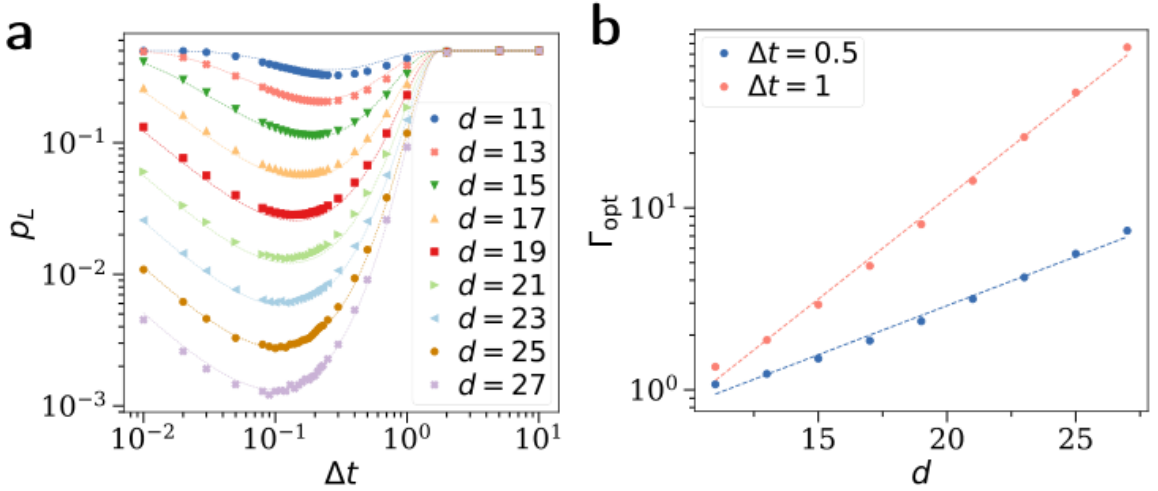
→Gives exponentially lower errors in distance *d*
(compared to constant *Δt)*

$$\Gamma_{\text{opt}} = \frac{R(\Delta t)}{R(\Delta t^{\star})} \sim \frac{2e^{-1}}{\lambda \Delta t g d}(1+\lambda\Delta t)^{gd/2}$$

For Google Willow experiment, we predict up to 40% error reduction

Google Quantum AI and Collaborators, Nature 638, 920 (2025).

For time-dependent noise, we adapt *Δt* as function of time, which we prove gives advantage over any static protocol



Rotated surface code simulations

Reading for Lectures

[G] Daniel Gottesman, Surviving as a Quantum Computer in a Classical World (2024 draft). §§2.1–2.5; §3.1.

[T] Spencer et al., Quantum error correction and fault tolerance: A comprehensive tutorial, arXiv:2605.29137 (2026). §§3.1–3.4.