Beyond NISQ: Fault Tolerant Quantum Computing

10/26/2026

I-Chi Chen

Outline

- Introduction
  1. Quantum Error correction 101

  2. Fault tolerance quantum computation

- Future Challenges

Quantum Simulation via Quantum Computer

Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem because it doesn't look so easy - Richard Feynman

Quantum Computer

Richard Feynman, International Journal of Theoretical Physics, Vol. 21, Nos. 6/7, 1982

The Quantum Noise Is A Challenge

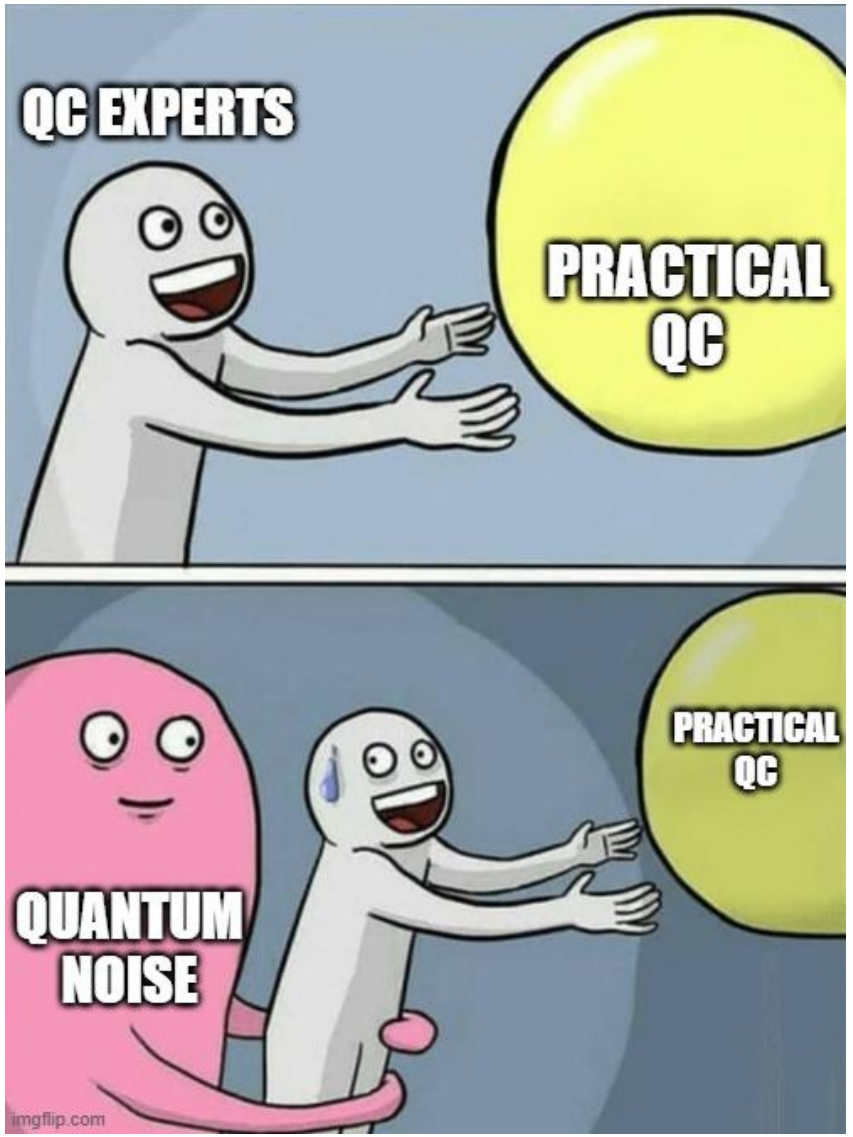
Noise Sources:

- Imperfect gates
- Quantum decoherence (interaction with environment)

Causes errors:

Bitflip: $|0\rangle \rightarrow |1\rangle$ Phaseflip : $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

Depolarization: make state maximally mixed state.



Solution:

- Quantum Error Mitigation

  Can work for noisy intermediate scale quantum (NISQ) devices (Not Scalable)
- Quantum Error Correction

  Possible if error rate is below a threshold.
  Used for fault tolerant quantum computer (FTQC)

Quantum Error Correction 101: Classical Code

Encode logical 1 bit into multi bit

→: encode
→: Error
→: correct

1. Code word:
   e.g. Reptition code $\bar{0} \rightarrow 000$ and $\bar{1} \rightarrow 111$
2. Error:
   bit flip error: $0 \rightarrow 1$ and $1 \rightarrow 0$
3. Detection:
   The result is not 000 or 111. One can discard the results. E. g. 001, 100, 110, and so on.

4. Correction:
   Majority vote $001 \rightarrow 000$, $100 \rightarrow 000$, $110 \rightarrow 111$. However, majority vote is not for other codes.

5. Parity check:
   $P = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$
   Check whether # of 1s in parity check is even or odd.
   Odd: errors occurred ☹ Even: no error ☺

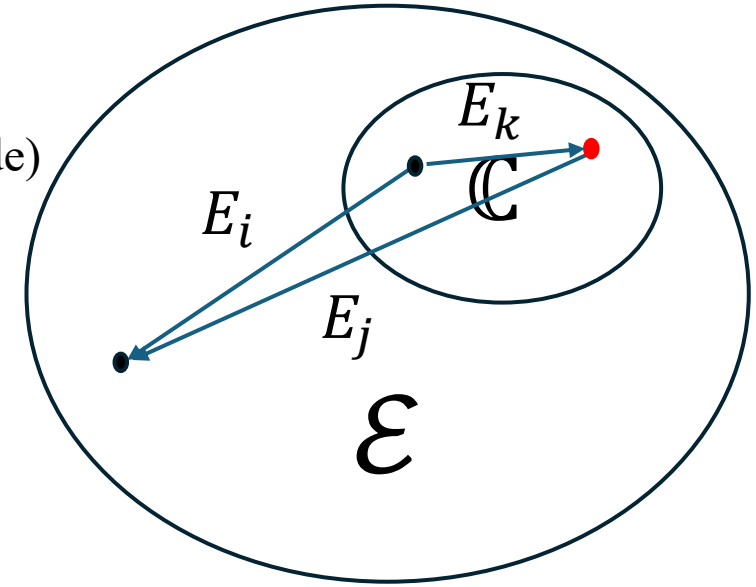
6. Code distance (d):
   At least # of errors happen to make one of code words become another code word.
7. Notation:
   [n,k,d] n: the number of physical bits is encoded from k logical bits.

Quantum Error Correction 101: Quantum Code

Encode logical state $\{|\bar{0}\rangle, |\bar{1}\rangle\}$ into multi physical qubit state

e.g. Redundantly encode logical qubit into 3 physical qubits (Quantum repetition code)

The code word $|\bar{0}\rangle = |000\rangle$ $|\bar{1}\rangle = |111\rangle$



Code space: $\mathbb{C}$

$$|\bar{\psi}\rangle = \alpha|\bar{0}\rangle + \beta|\bar{1}\rangle$$

Error space: $\mathcal{E}$

Outside of code space

Used for quantum error detection.

To check whether the error, E_i, can be detectable $\langle\overline{w_\mu}|E_i|\overline{w_\upsilon}\rangle = a_i\delta_{\mu\upsilon}$

where $|\overline{w_\upsilon}\rangle$ are orthogonal logical basis states

To check whether the set $\{E_i\}$ can be correctable

$$\langle\overline{w_\mu}|E_i^\dagger E_j|\overline{w_\upsilon}\rangle = b_{ij}\delta_{\mu\upsilon}$$

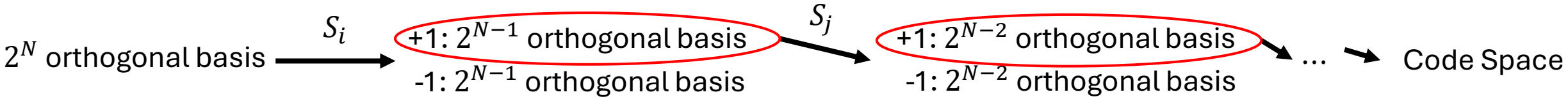
Knill-Laflamme theorem

Quantum Error Correction 101: Stabilizer

How to measure whether the state is outside of the code space?

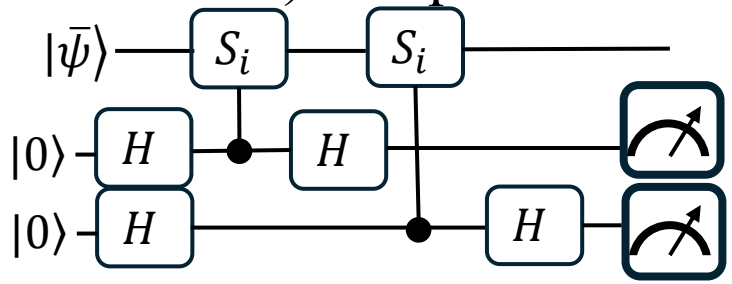
Most of quantum code's code space can characterized by Stabilizers: S_i $\quad S_i|\bar{\psi}\rangle = +1|\bar{\psi}\rangle$

Note: If we encode k logical qubits into N physical qubits, N-k stabilizers are required.



We can do non-destructive measurement of stabilizers (syndrome measurements) for quantum circuits.

The focused quantum code's stabilizers are Pauli string operators (easier for syndrome measurements and correction)



For instance, repetition code has Z type stabilizers, which can detect X-type error

$$|\bar{\psi}\rangle = \alpha|000\rangle + \beta|111\rangle \qquad S_{Z,1} = Z_1Z_2 \qquad S_{Z,2} = Z_2Z_3$$

It has corresponding Z check matrix

$$G_Z = \begin{pmatrix} 1\,1\,0 \\ 0\,1\,1 \end{pmatrix}$$

but it doesn't have X type stabilizers (X checks).

Quantum Error Correction 101: logical operation and notation

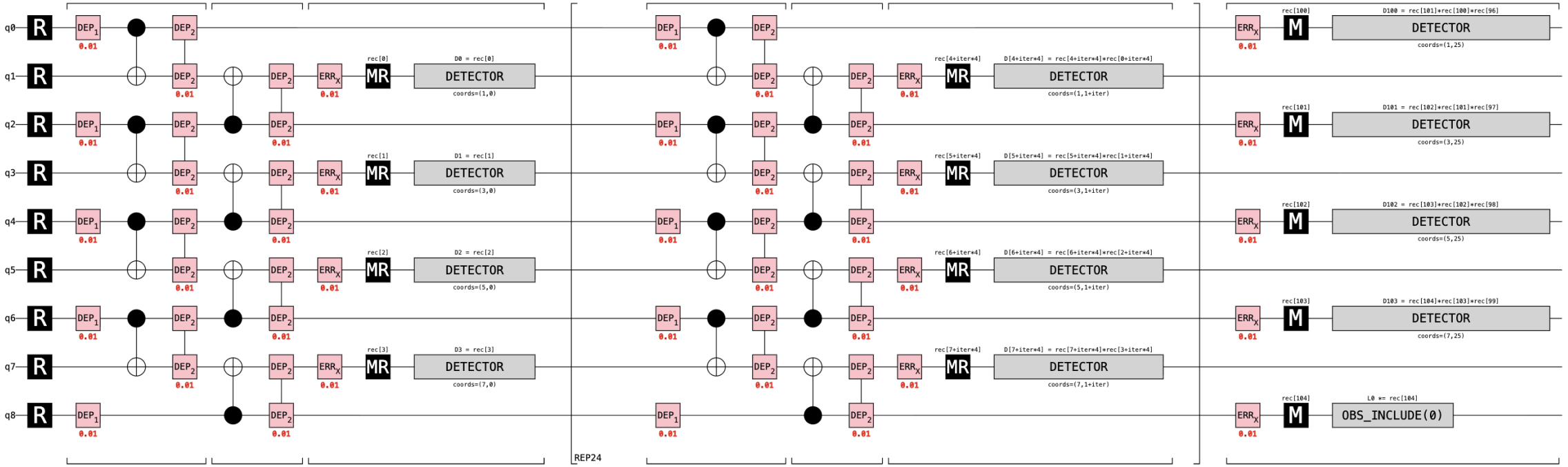
- Stabilizer group: Stabilizers can form a group G. The set of stabilizer operators (stabilizer elements) must commute with each other.
- Stabilizer generators: the group G can be specified by a set of generators $\{g_j\}_{j=1}^{m}$. For Pauli-type stabilizer group, which has self inverse operators, any elements g belonging to the group can written as $g = \prod_m g_j^{\alpha_j}$ where $\alpha_j \in \{0,1\}$. Totally, 2^melements.
- Pauli Logical operations: the operators commute all of operators from stabilizer group, but the operators themselves is not in the stabilizer group.
- Code distance d: lowest weight of Pauli logical operators.
- Quantum code notation: [[n,k,d]]

  n: the number of physical qubits is encoded from k logical qubits. At least d errors happen to form a logical error.

Based on Knill-Laflamme theorem, any quantum code with code distance d can detect all d-1 errors, and correct all $\lfloor (d-1)/2 \rfloor$ errors.

Quantum Error Correction 101: Decoding

What if the stabilizers measurement are noisy? We can do multi-rounds of stabilizer measurements and find mostly likely happened errors.



(d=5 repetition code)

Generated from stim code

(We don't do any correction after each round but record every syndrome measurement result and send it to decoder. Let decoder calculate which error most likely happen)

Sadly, the optimal decoder (maximal likelihood decoder) is # P complete ☹

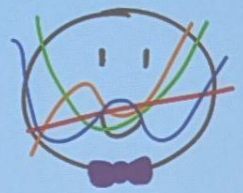
Luckily, there are polynomial time decoders which still can help QC reach fault tolerance ☺

Codes

Classical:

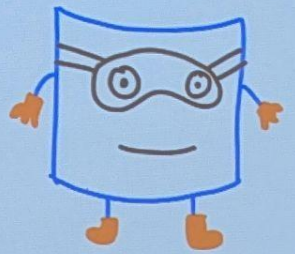


Repetition

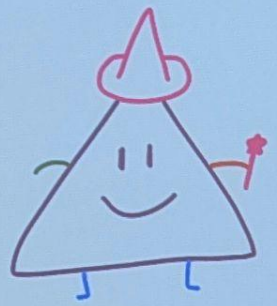


Reed-Solomon

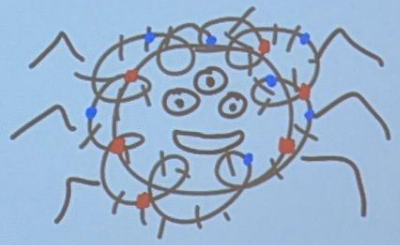
Quantum:



Surface



Color



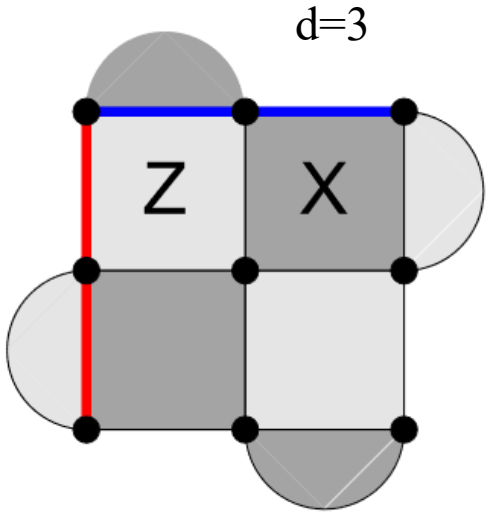
qLDPC

Quantum AI

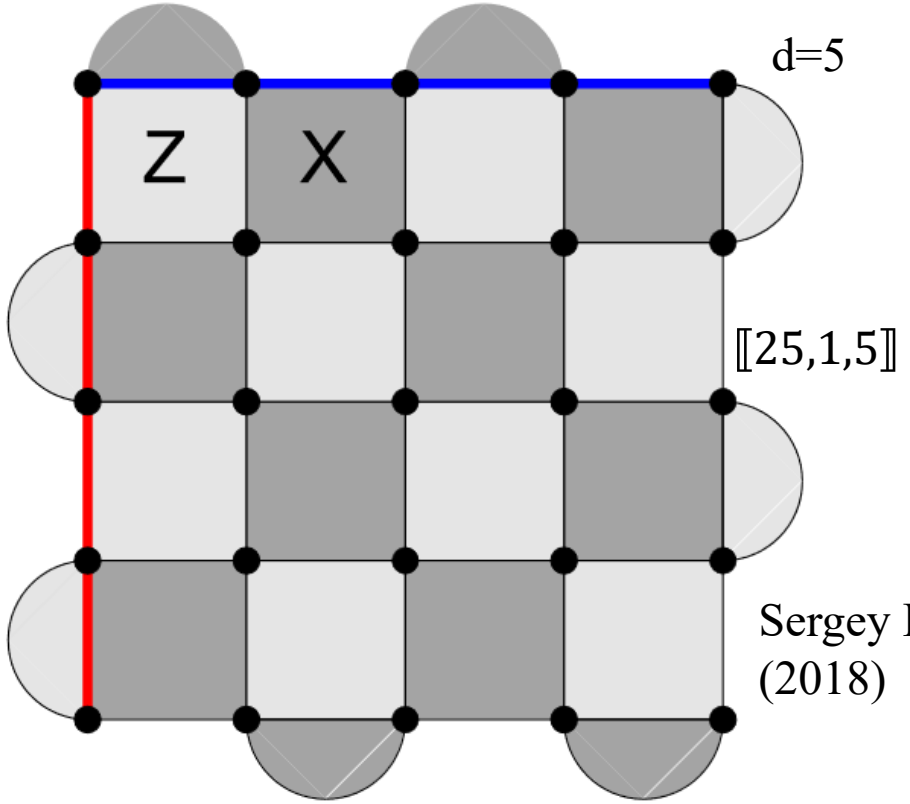
27

From QEC 25 Google's talk

Rotated surface code:



⟦9,1,3⟧



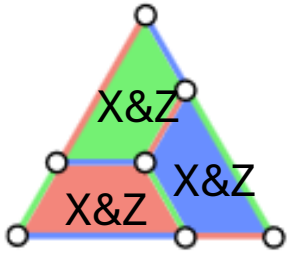
⟦25,1,5⟧

Arbitrary d

→ $[\![d^2, 1, d]\!]$

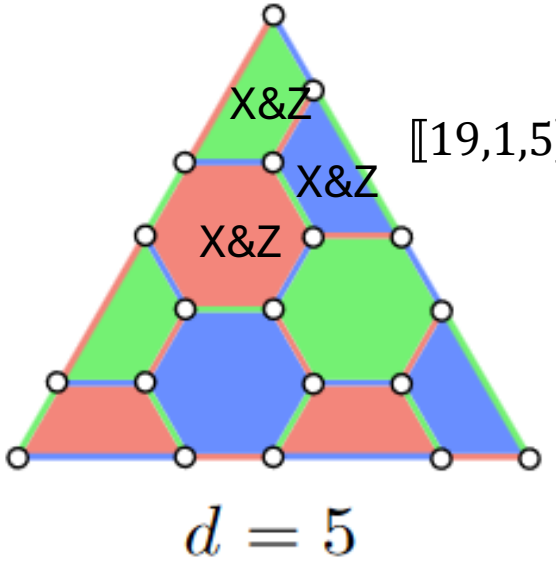
Sergey Bravyi et al, npj Quantum Information **4**, 55 (2018)

Color code:



$d = 3$

⟦7,1,3⟧



$d = 5$

⟦19,1,5⟧

Arbitrary d

→ $[\![(3d^2 + 1)/4, 1, d]\!]$

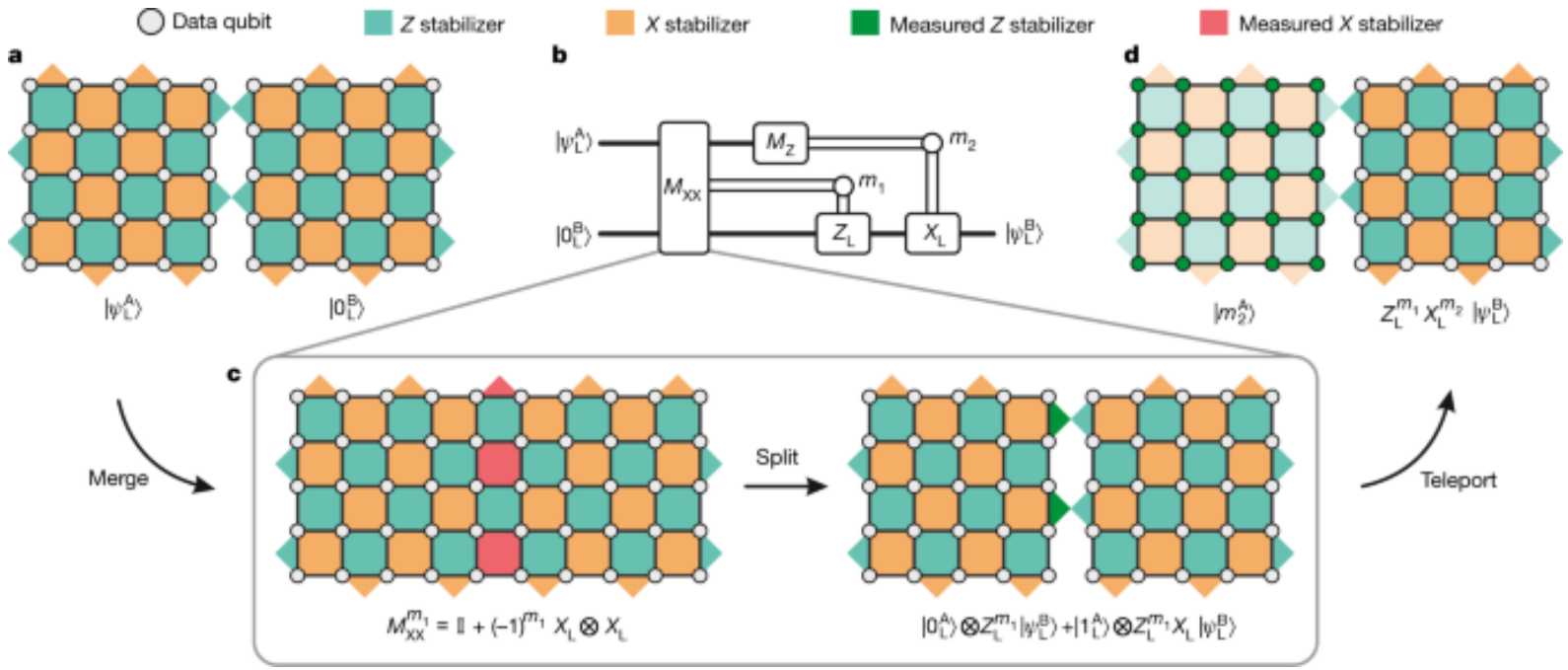
Yugo Takada et al, PRX Quantum **5**, 030352 (2024)

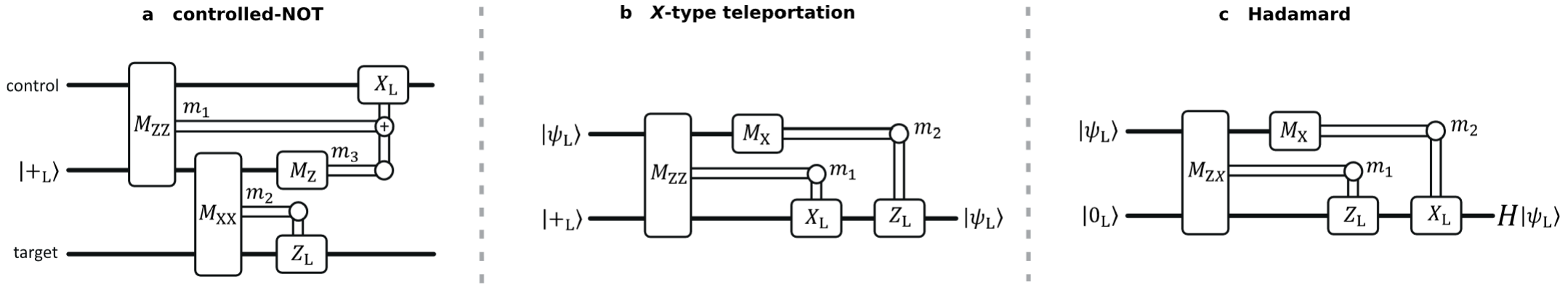
Fault Tolerance Quantum Computation

Quantum computation

Clifford Gate implementation:

1. Lattice Surgery



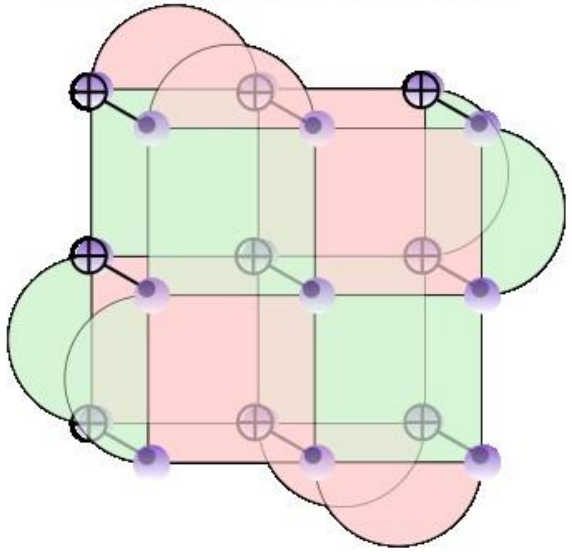


Requires $\mathcal{O}(d)$ rounds of syndrome measurements after each logical gate

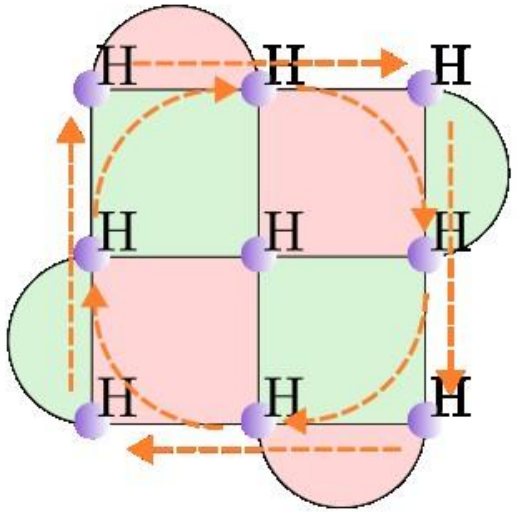
Alexander Erhard et al, Nature volume 589, pages 220–224 (2021)

2. Transversal gates

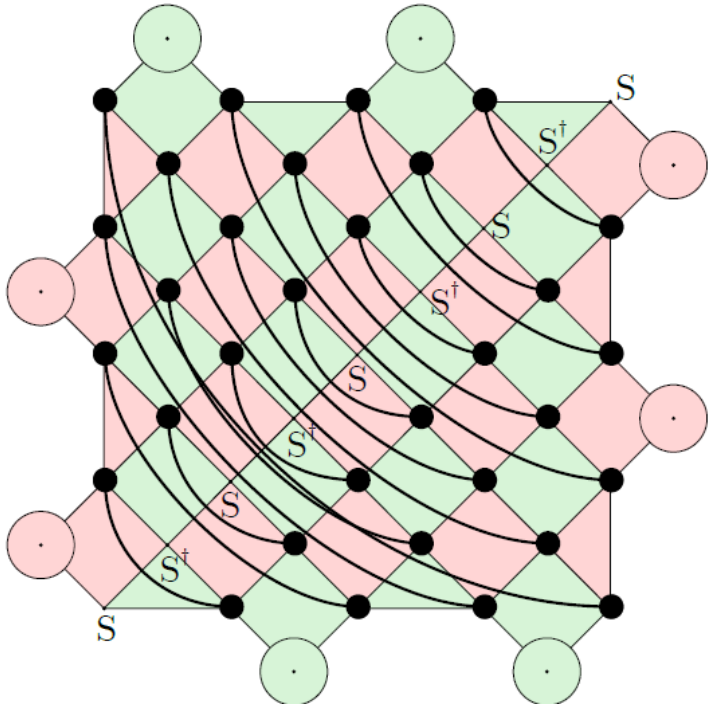
transversal CNOT



fold-transversal H



Logical S gate
(In the middle
syndrome extraction)



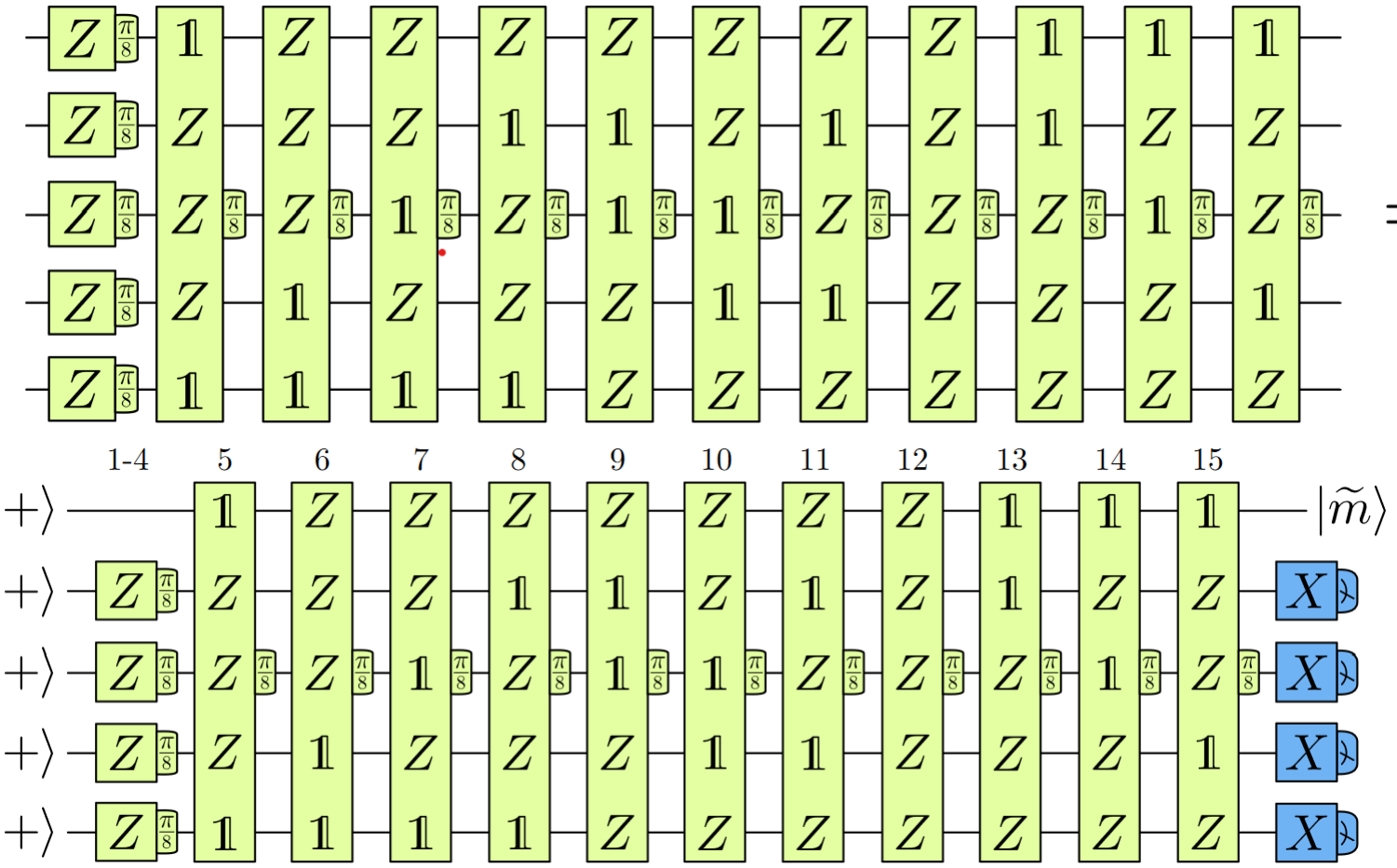
Harvard team claim it just requires $\mathcal{O}(1)$ rounds of syndrome measurements after each logical gate.

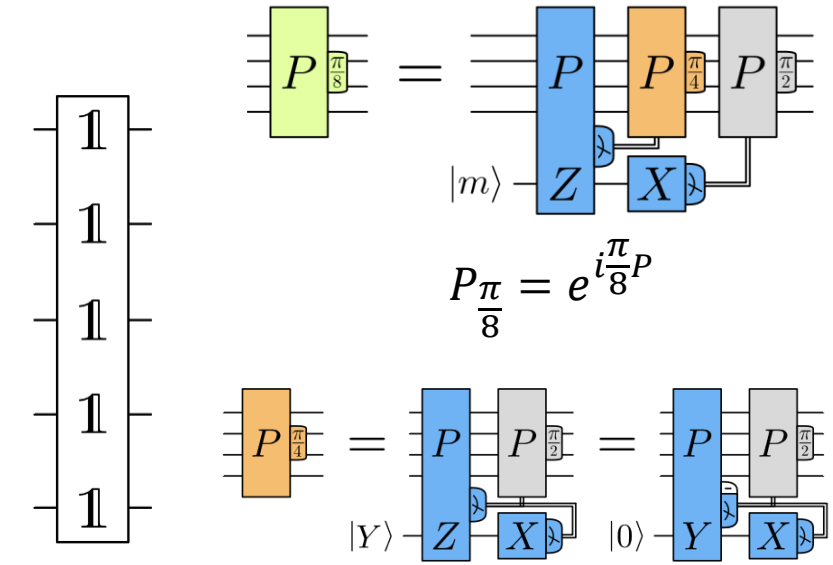
Madelyn Cain et al, arXiv:2403.03272

How about non-Clifford gate?

1. Magic state distillation?

15-1 protocol,





$P_{\frac{\pi}{8}} = e^{i\frac{\pi}{8}P}$

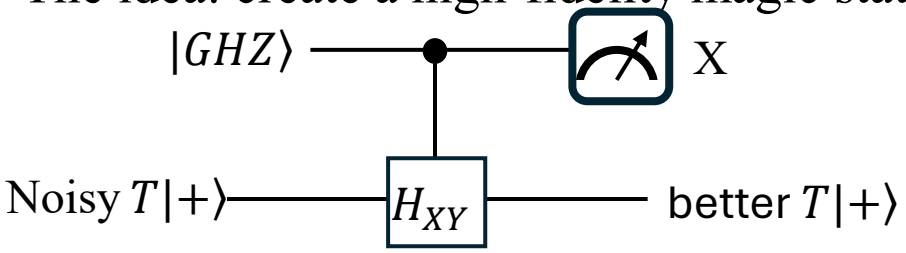
Where $|\widetilde{m}\rangle = T|+\rangle$

Too much spacetime overhead `☹.

D Litinski., Quantum 3, 128 (2019) D Litinski, arXiv:1905.06903

2. Magic state cultivation (Cheap as lattice surgery CNOT) Craig Gidney, et al, arXiv:2409.17595.

The idea: create a high-fidelity magic state in small code, and rapidly grow small code into bigger code.



where $H_{XY} = (X+Y)/\sqrt{2}$ which are transversal operator for 2D color code.

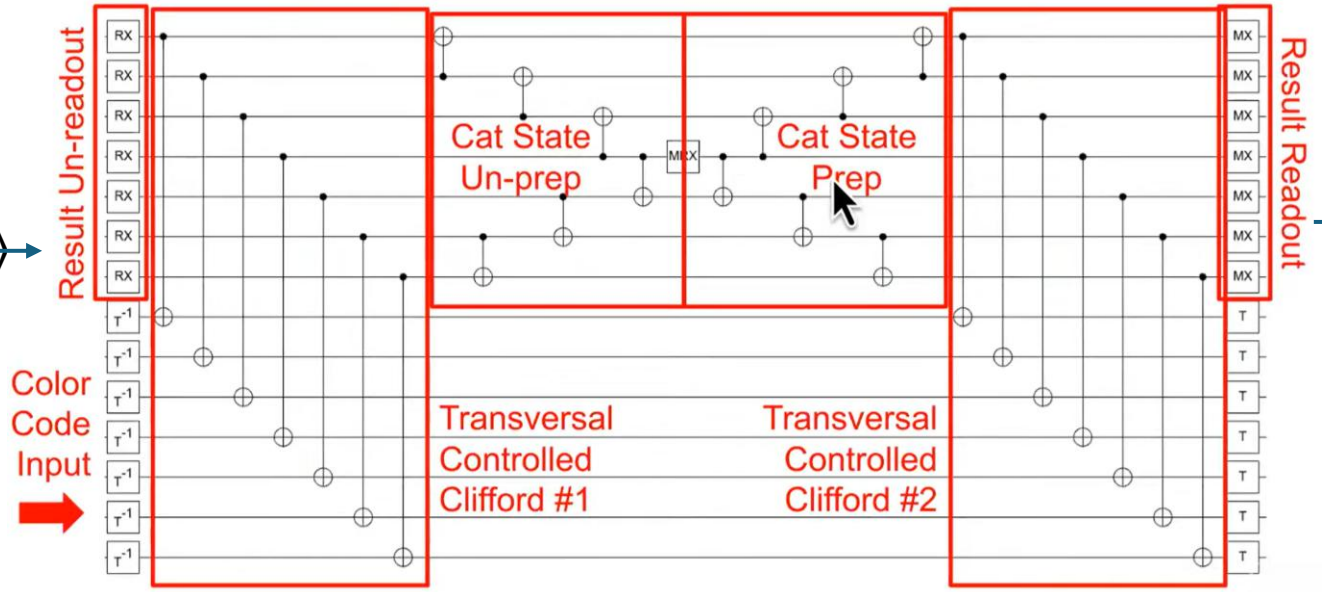
Christopher Chamberland et al, npj Quantum Information **6**, 91 (2020)

Google changes the protocol:

Cultivation: Double checks

Growing

Injection: prepare a noisy $T|+\rangle$



Rapidly grow small code into bigger code.

But, still



Sergey Bravyi and Alexei Kitaev. PRA, 71(2):022316, 2005.

Future Challenges

- How to implement Quantum Error Correction to real hardware.

- How noise beyond Pauli noise affect the Quantum Error Correction performance.