Doubled Color Codes

Sergey Bravyi and Andrew Cross

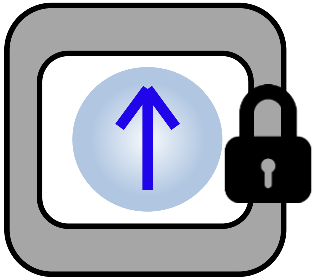
IBM Watson Research Center

Based on
arXiv:1509.03239

QIP
January 12, 2016

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

Too much protection means too little control



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

Transversal Logical Gates (TLG): bitwise application of a physical gate implements a logical gate.

codespace

$$V^{\otimes n} = \begin{array}{|c|c|} \hline U & 0 \\ \hline 0 & * \\ \hline \end{array}$$

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

Transversal Logical Gates (TLG): bitwise application of a physical gate implements a logical gate.

TLGs are highly desirable.

- TLG's do not spread pre-existing errors
- noisy TLG's introduce uncorrelated errors
- no need for ancillary qubits, no time overhead

TLGs are not universal for any error detecting code
Eastin and Knill (2009)

2D stabilizer codes have only Clifford TLGs
SB and Koenig (2012), Pastawski and Yoshida (2014)

Recent breakthrough: gauge fixing method

[Paetznick and Reichardt 2013]

PRL **111,** 090505 (2013) PHYSICAL REVIEW LETTERS week ending 30 AUGUST 2013

Universal Fault-Tolerant Quantum Computation with Only Transversal Gates and Error Correction

Adam Paetznick[1] and Ben W. Reichardt[2]

[1]*David R. Cheriton School of Computer Science and Institute for Quantum Computing, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada*

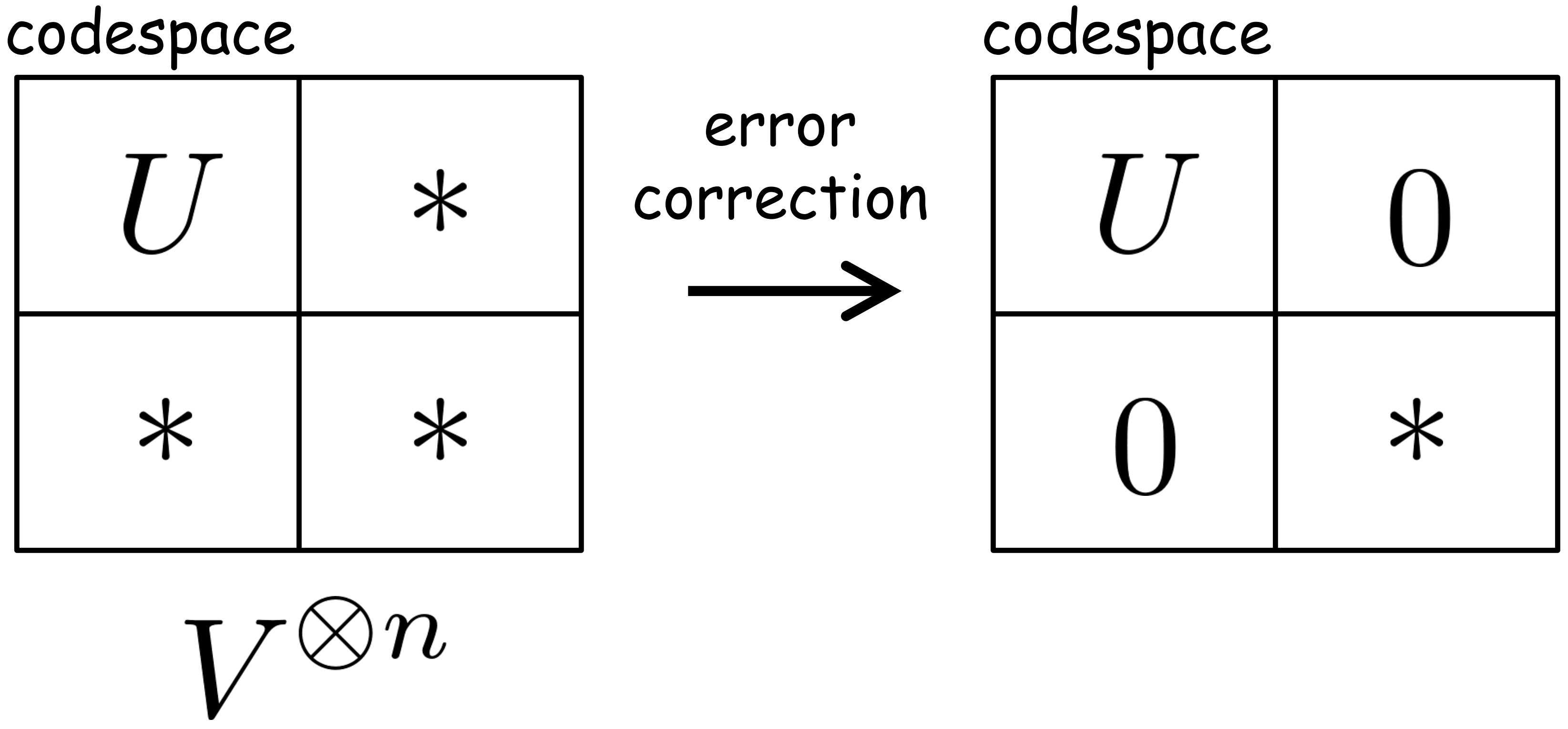
[2]*Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, California 90089, USA*

(Received 13 April 2013; published 29 August 2013)

Recent breakthrough: gauge fixing method

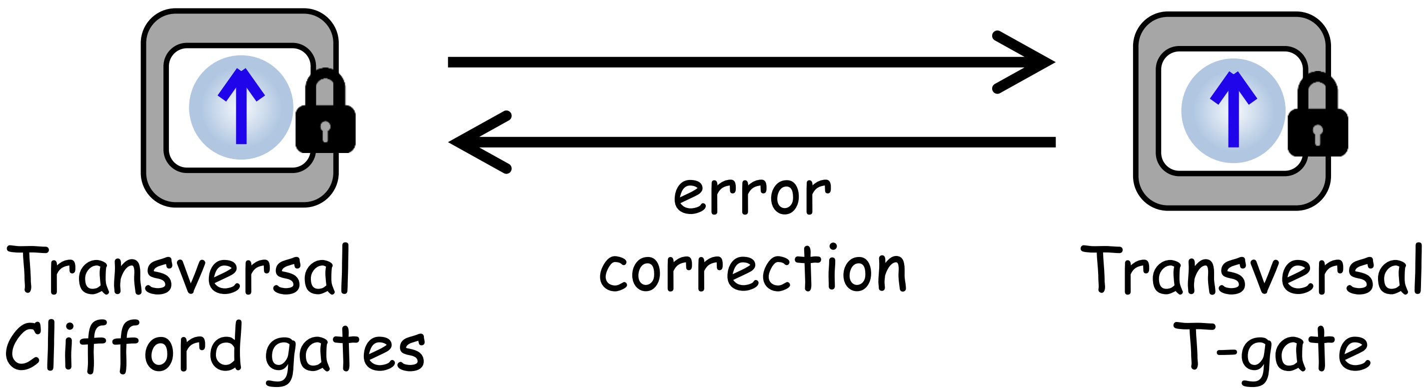
[Paetznick and Reichardt 2013]

Apply a bitwise physical gate. Error-correct the system back to the codespace, if needed.



Recent breakthrough: gauge fixing method
[Paetznick and Reichardt 2013]

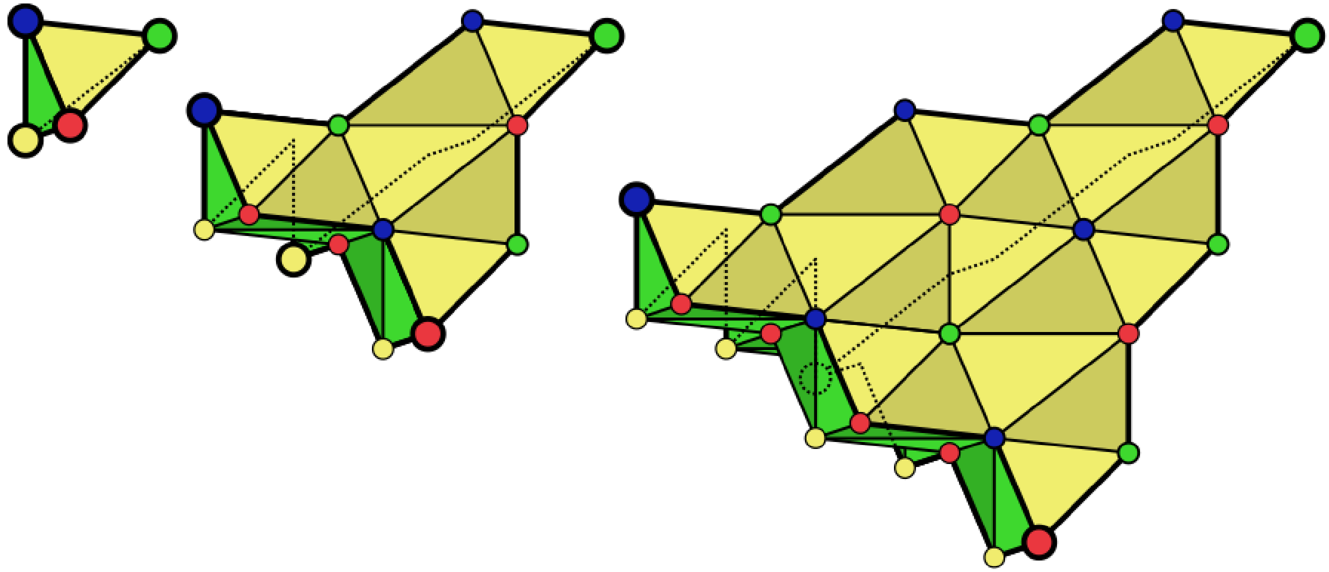
Similar to code conversion/code deformation:



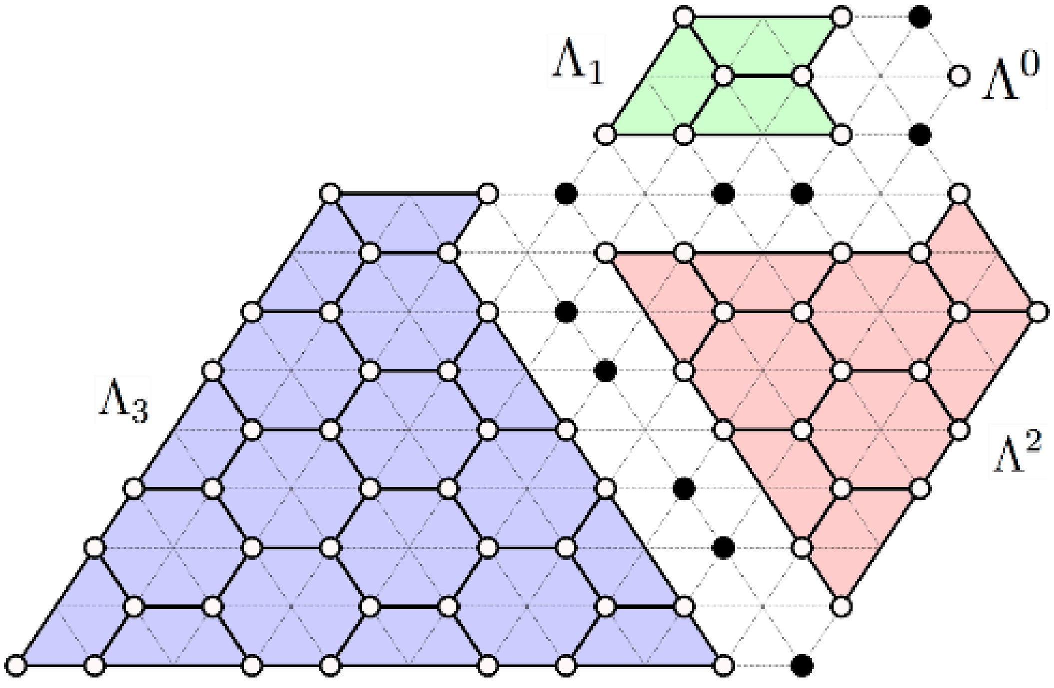
Two codes are defined on the same set of qubits.
Combine TLGs of the two codes to achieve universality.

Can we realize the conversion by local measurements if the physical qubits are laid out on 2D or 3D grid ?

Realization of the gauge fixing method with
3D subsystem color codes [Bombin 2014, 2015]
Transversal implementation of the Clifford+T gate set.
Quantum fault-tolerance with a constant time overhead.



Our result: transversal implementation of the Clifford+T gate set by the gauge fixing method in the 2D architecture.



OUTLINE

- Triply-even CSS codes
- Doubling transformation
- Doubled color codes: overview of the construction
- Logical Clifford+T circuits: numerical simulation

Reminder: Calderbank-Shor-Steane codes

$$\mathrm{CSS}(\mathcal{A})$$

Logical states:

$$|0_L\rangle = \frac{1}{|\mathcal{A}|^{1/2}} \sum_{f \in \mathcal{A}} |f\rangle$$

$$|1_L\rangle = \frac{1}{|\mathcal{A}|^{1/2}} \sum_{f \in \mathcal{A}} |f \oplus \overline{1}\rangle$$

$\mathcal{A} \subseteq \mathcal{A}^{\perp} \subseteq \mathbb{F}_2^n$ self-orthogonal linear subspace with odd n

Stabilizers:

$$\begin{cases} X(f) \, : \, f \in \mathcal{A} \\ Z(f) \, : \, f \in \mathcal{A}^{\perp} \cap Even \end{cases}$$

Reminder: Calderbank-Shor-Steane codes

$$\mathrm{CSS}(\mathcal{A})$$

Logical states:

$$|0_L\rangle = \frac{1}{|\mathcal{A}|^{1/2}} \sum_{f \in \mathcal{A}} |f\rangle$$

$$|1_L\rangle = \frac{1}{|\mathcal{A}|^{1/2}} \sum_{f \in \mathcal{A}} |f \oplus \overline{1}\rangle$$

$\mathcal{A} \subseteq \mathcal{A}^{\perp} \subseteq \mathbb{F}_2^n$ self-orthogonal linear subspace with odd n

Code distance:

$$d(\mathcal{A}) = \min\{|f| \;:\; f \in \mathcal{A}^{\perp} \cap Odd\}$$

Any CSS code has transversal logical Paulis and CNOT.

Other TLGs require a special symmetry.

Self-dual code:

$$\mathcal{A}^{\perp} = \mathcal{A} + \langle \overline{1} \rangle$$

Transversal gate: Hadamard

Doubly-even code:

$$|f| = 0 \pmod 4 \quad \forall\, f \in \mathcal{A}$$

Transversal gate: $S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$

Any CSS code has transversal logical Paulis and CNOT.

Other TLGs require a special symmetry.

Self-dual code:

$$\mathcal{A}^{\perp} = \mathcal{A} + \langle \overline{1} \rangle$$

Transversal gate: Hadamard

Triply-even code:

$$|f| = 0 \pmod 8 \quad \forall\, f \in \mathcal{A}$$

Transversal gate: $T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$

Need a family of triply-even CSS codes with a divergent code distance.

Candidates:

Concatenated Reed-Muller [[15,1,3]]	hard to embed in 2D
3D color codes	

$$
G_0 = \begin{bmatrix}
1111111111111110101010101010101010101010101010101 \\
0000000000000000000111100110011000011001100110011 \\
0000000000000001100000011001100110000000000000000 \\
0000000000000000000000000000000001111000000001111 \\
0000000000000000011110000000000000000111100000000 \\
0000000000000000000001111000011110000000000000000 \\
0000000000000000000000000111111110000000000000000 \\
0000000000000000000000000000000001111111100000000 \\
0000000000000000000000000000000000000000011111111 \\
1010101010101010000000000000000000000000000000000 \\
0110011001100110000000000000000000000000000000000 \\
0001111000011110000000000000000000000000000000000 \\
0000000111111110000000000000000000000000000000000
\end{bmatrix}
$$

[[49,1,5]], triply-even

found by a numerical search [SB and Haah 2012]

smallest distance-5 triply even code

Does it have any structure ?

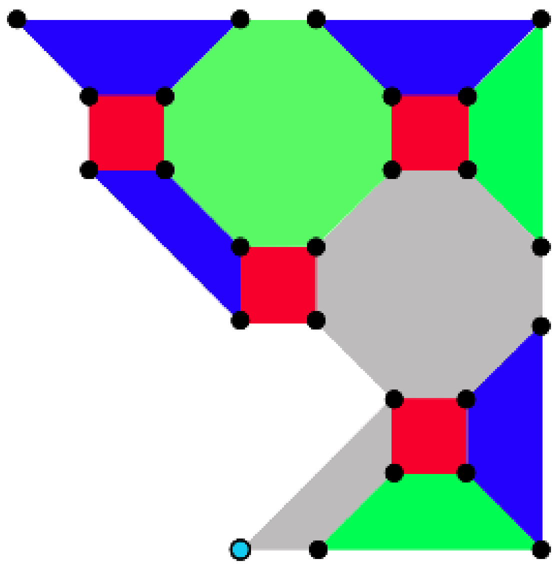
Need a family of triply-even CSS codes with a divergent code distance.

Candidates:

Concatenated Reed-Muller [[15,1,3]]

3D color codes

} hard to embed in 2D



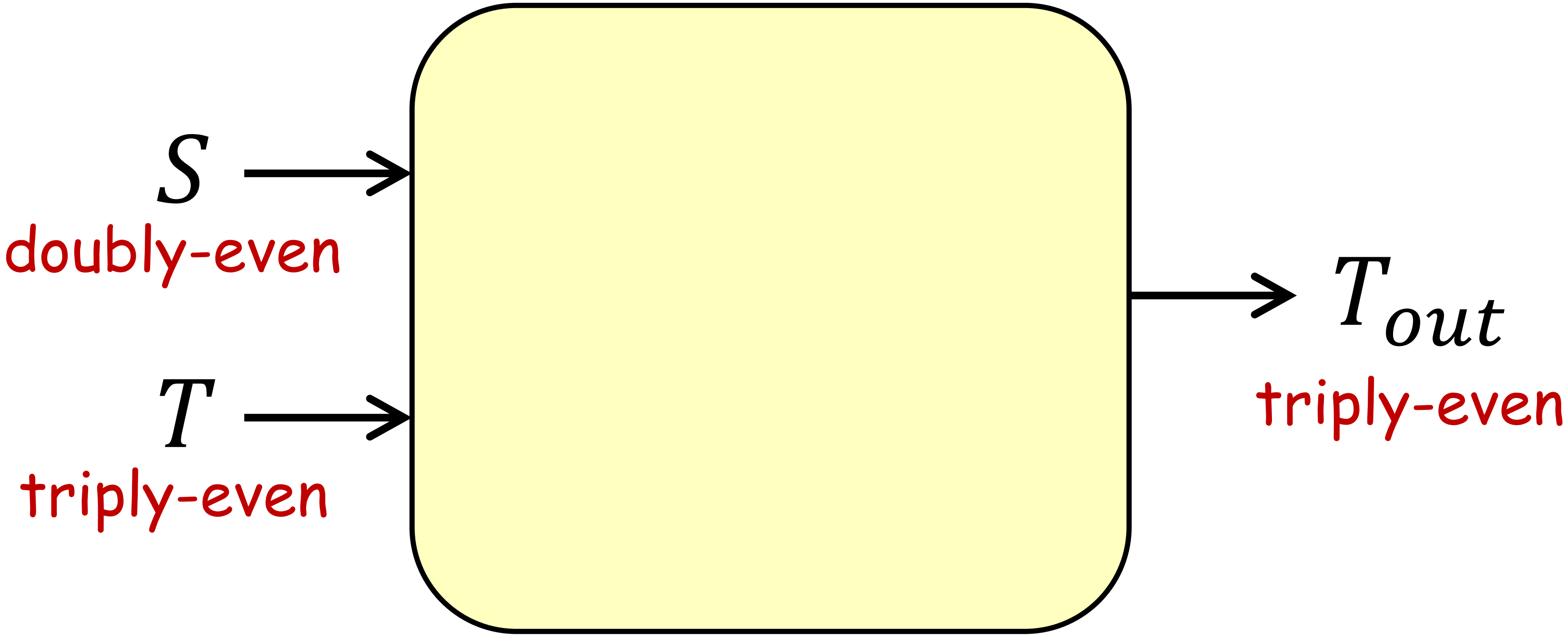
[[49,1,5]], triply-even

found by a numerical search [SB and Haah 2012]

smallest distance-5 triply even code

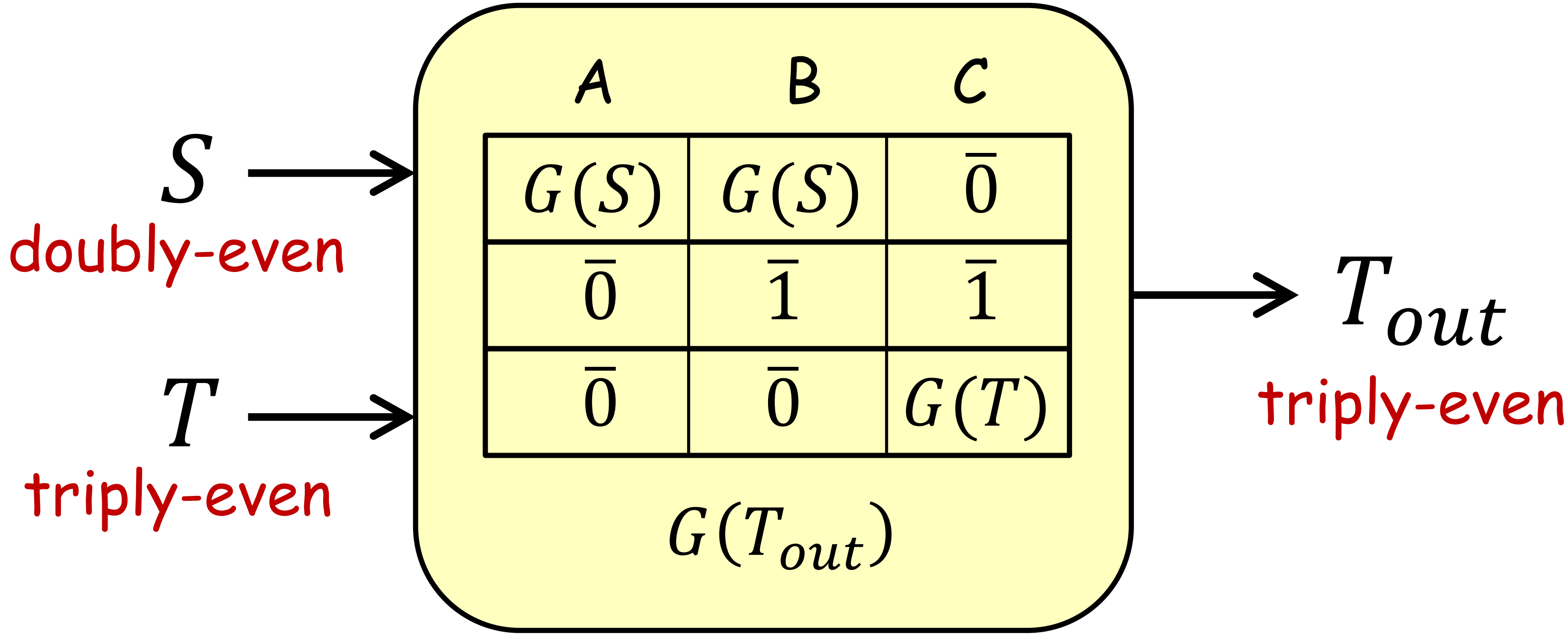
Does it have any structure ?

Doubling transformation [Betsumiya and Munemasa 2010]



$$n(S) + n(T) = 0 \pmod 8$$

Doubling transformation [Betsumiya and Munemasa 2010]



generating matrices of S, T, T_{out}

$$n(S) + n(T) = 0 \pmod 8$$

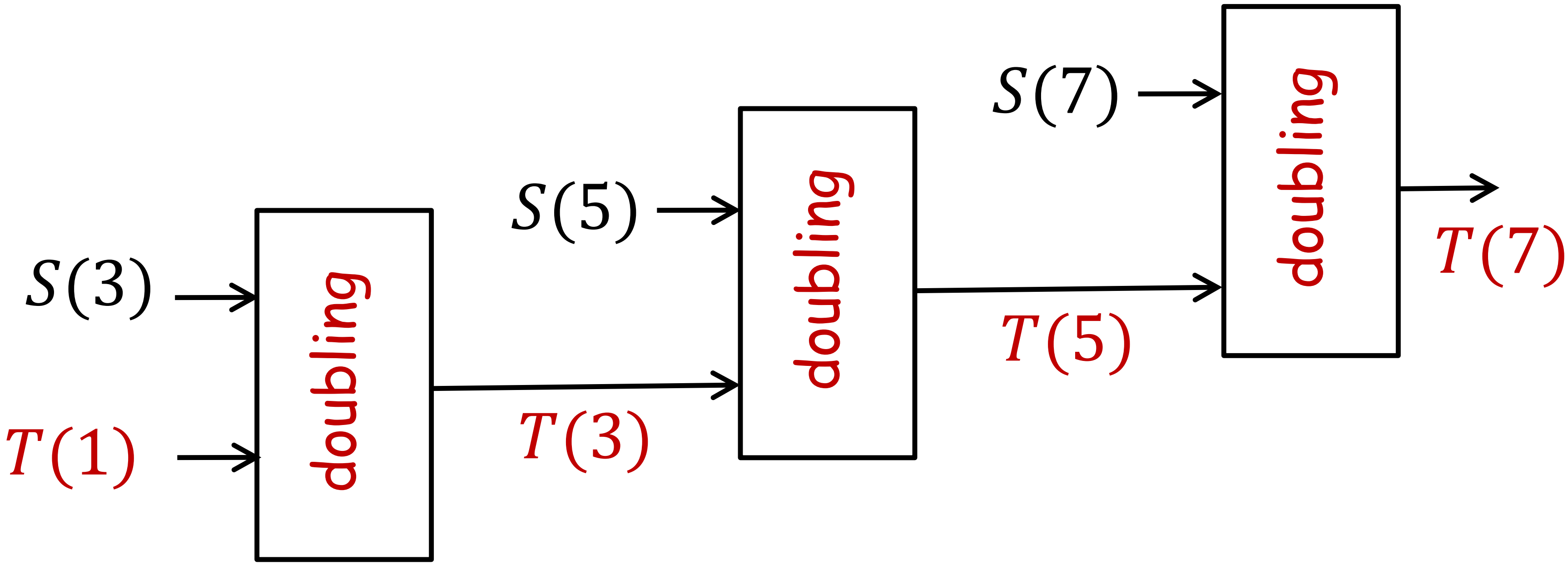
Code distance:

$$d(T_{out}) = \min\{\, d(S), 2 + d(T) \,\}$$

Doubled Color Codes

$S(d)$: 2D color code with distance d (doubly even)

$T(1)$: unencoded qubit (distance 1)



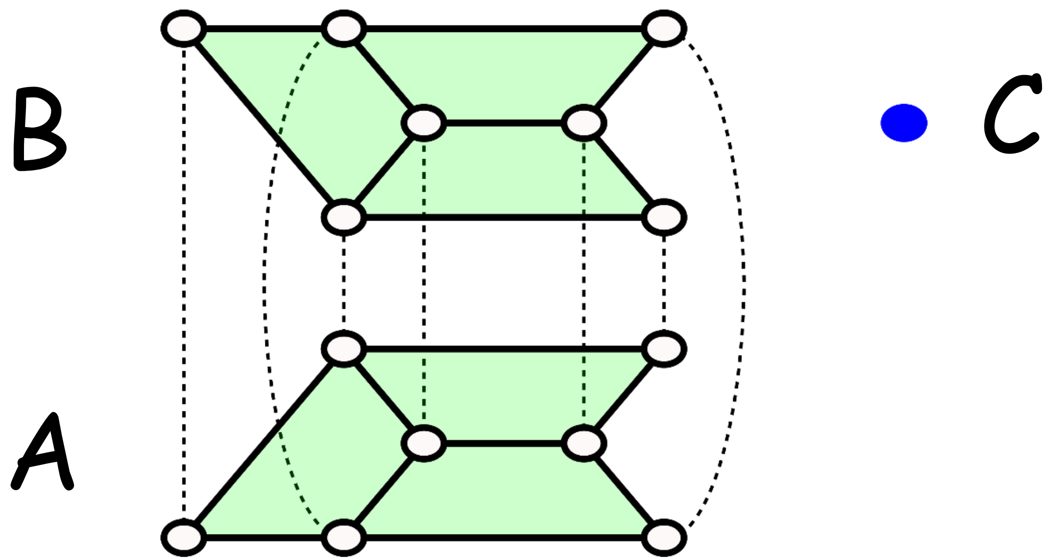
Each doubling step increases the distance by two.

Doubled Color Codes: $T(3), T(5), T(7), \ldots$

Doubled color codes: small examples

$$T(3) = [\![15,1,3]\!]$$

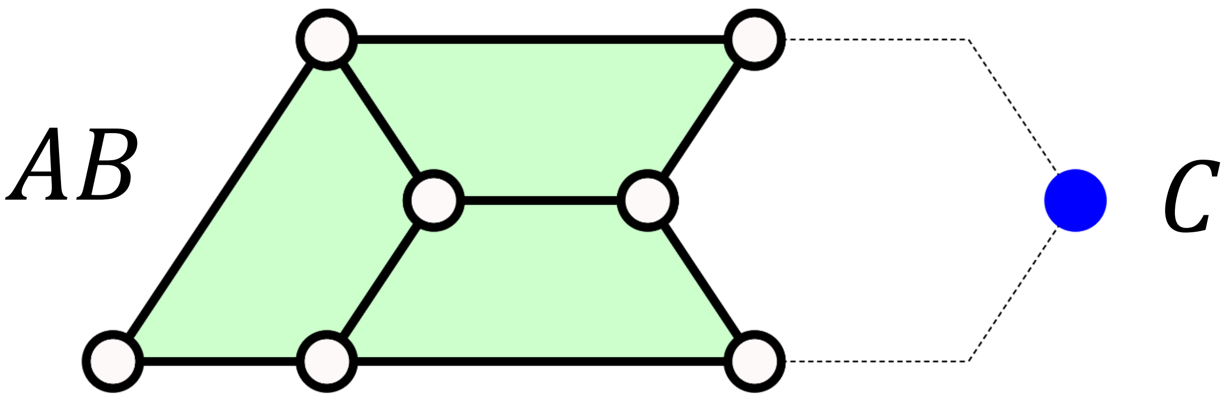
Bilayer geometry (quasi-2D). One qubit per site.



Doubled color codes: small examples

$$T(3) = [\![15,1,3]\!]$$

Single layer geometry. Two qubits per site in AB.

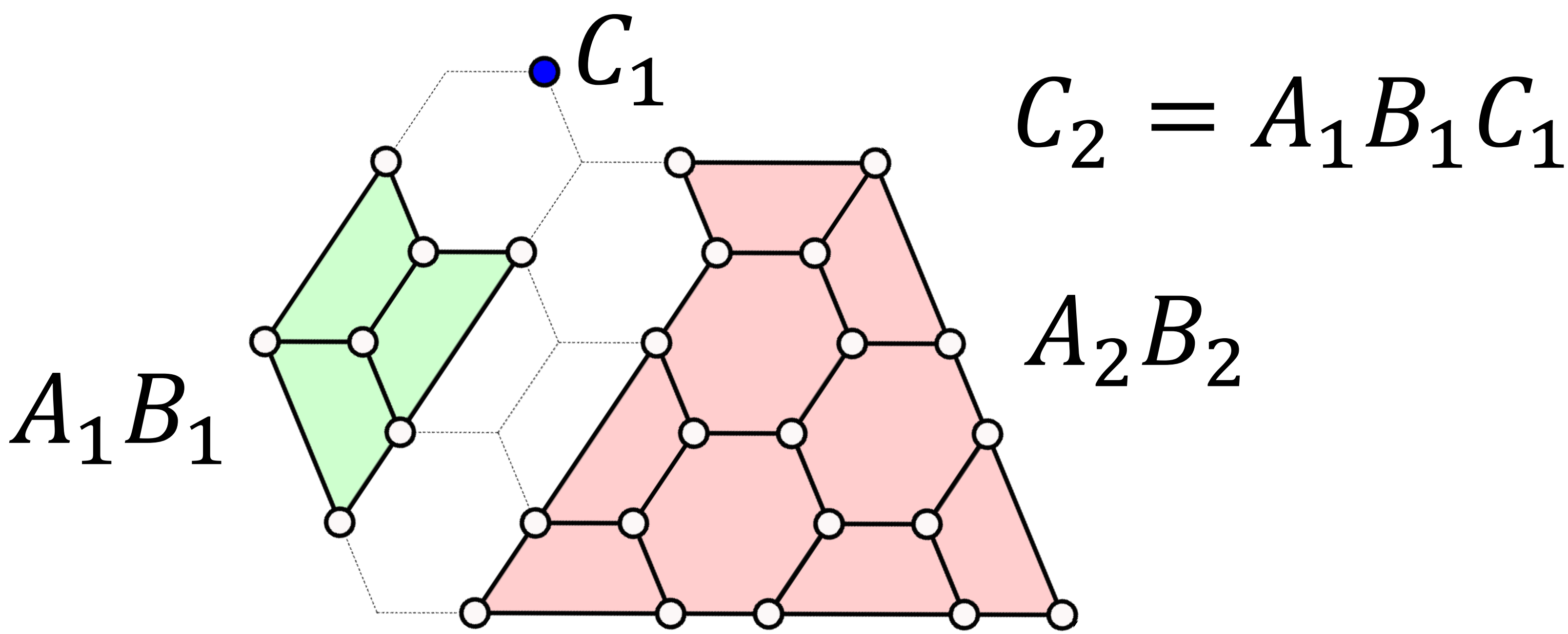


X-stabilizers: double faces of AB; full BC

Z-stabilizers: single faces of A; single faces of B;
double edges of AB; full BC

Doubled color codes: small examples

$$T(5) = [\![53,1,5]\!]$$



X-stabilizers: double faces of A_iB_i; full B_2C_2 and B_1C_1

Z-stabilizers: single faces of A_i and B_i

double edges of A_iB_i; full B_2C_2 and B_1C_1

Technical remark: color codes on the honeycomb lattice are doubly-even in a weak sense.

$$S^{\sigma_1} \otimes S^{\sigma_2} \otimes \cdots \otimes S^{\sigma_n} \longrightarrow \text{Logical S-gate}$$

$$\sigma_j = 0, \pm 1$$

Weak doubly-even condition:

$$\sum_{j=1}^{n} \sigma_j f_j = 0 \pmod 4 \quad \forall\, f \in \mathcal{A}$$

$$\sum_{j=1}^{n} \sigma_j = 1 \pmod 2$$

Technical remark: color codes on the honeycomb lattice are doubly-even in a weak sense.

$$T^{\sigma_1} \otimes T^{\sigma_2} \otimes \cdots \otimes T^{\sigma_n} \longrightarrow \text{Logical T-gate}$$

$$\sigma_j = 0, \pm 1$$

Weak triply-even condition:

$$\sum_{j=1}^{n} \sigma_j f_j = 0 \pmod 8 \quad \forall\, f \in \mathcal{A}$$

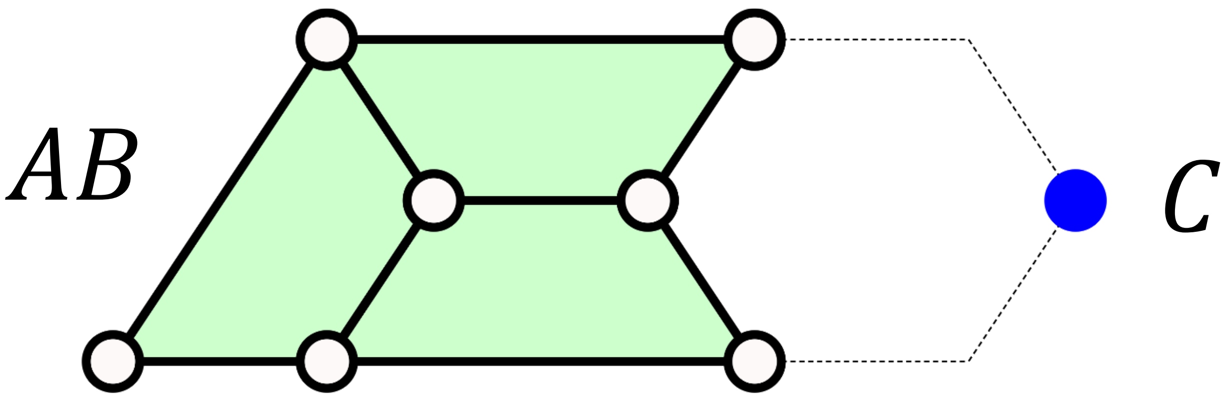
$$\sum_{j=1}^{n} \sigma_j = 1 \pmod 2$$

The doubling transformation works for the weak version of doubly/triply even codes.

How to measure high-weight non-local stabilizers ?

Step 1 (trivial): choose a new basis set of stabilizers

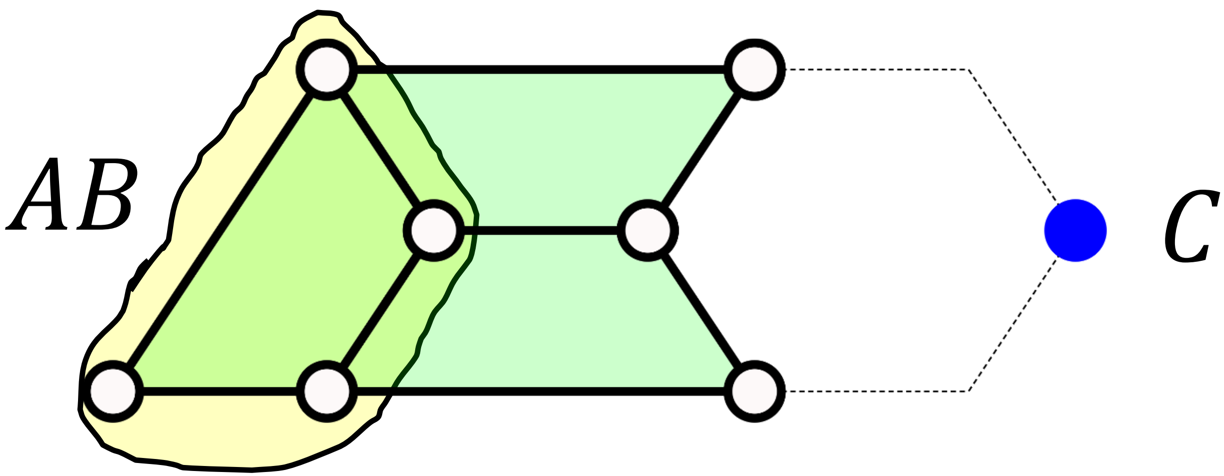
Stabilizer $Z(BC)$



How to measure high-weight non-local stabilizers ?

Step 1 (trivial): choose a new basis set of stabilizers

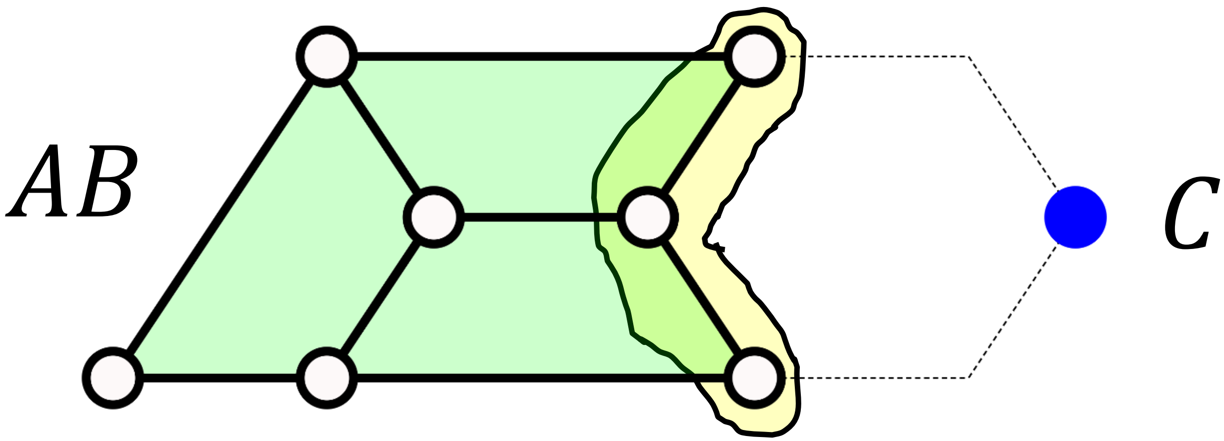
Stabilizer $Z(BC)$



How to measure high-weight non-local stabilizers ?

Step 1 (trivial): choose a new basis set of stabilizers

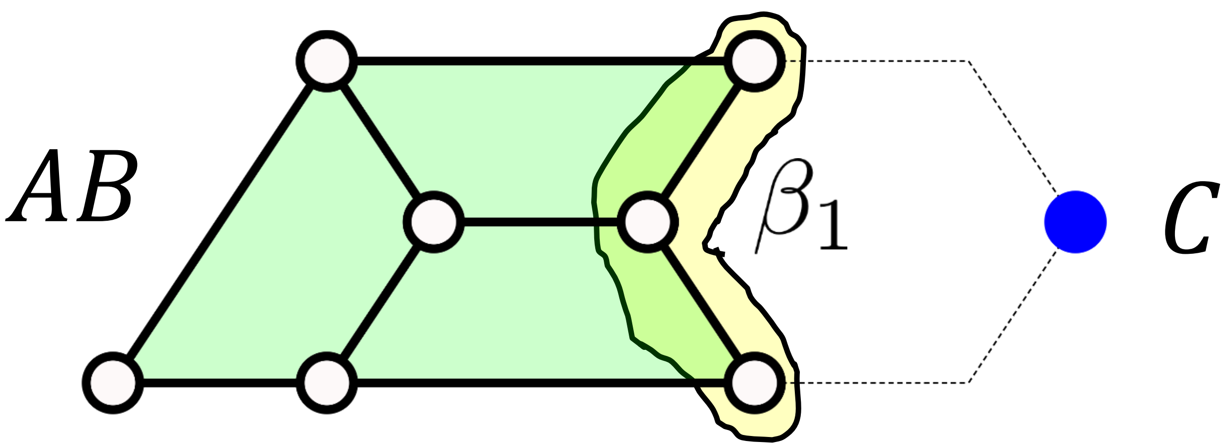
Stabilizer $Z(BC)$



How to measure high-weight non-local stabilizers ?

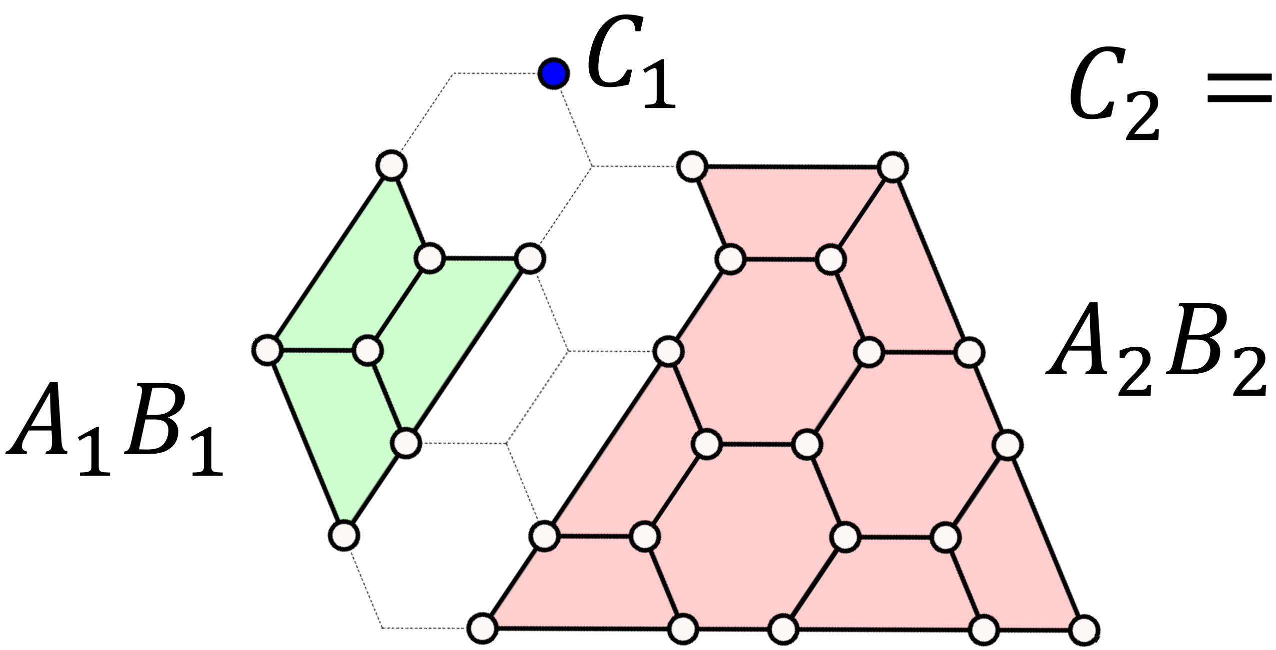
Step 1 (trivial): choose a new basis set of stabilizers

Stabilizer $Z(BC)$



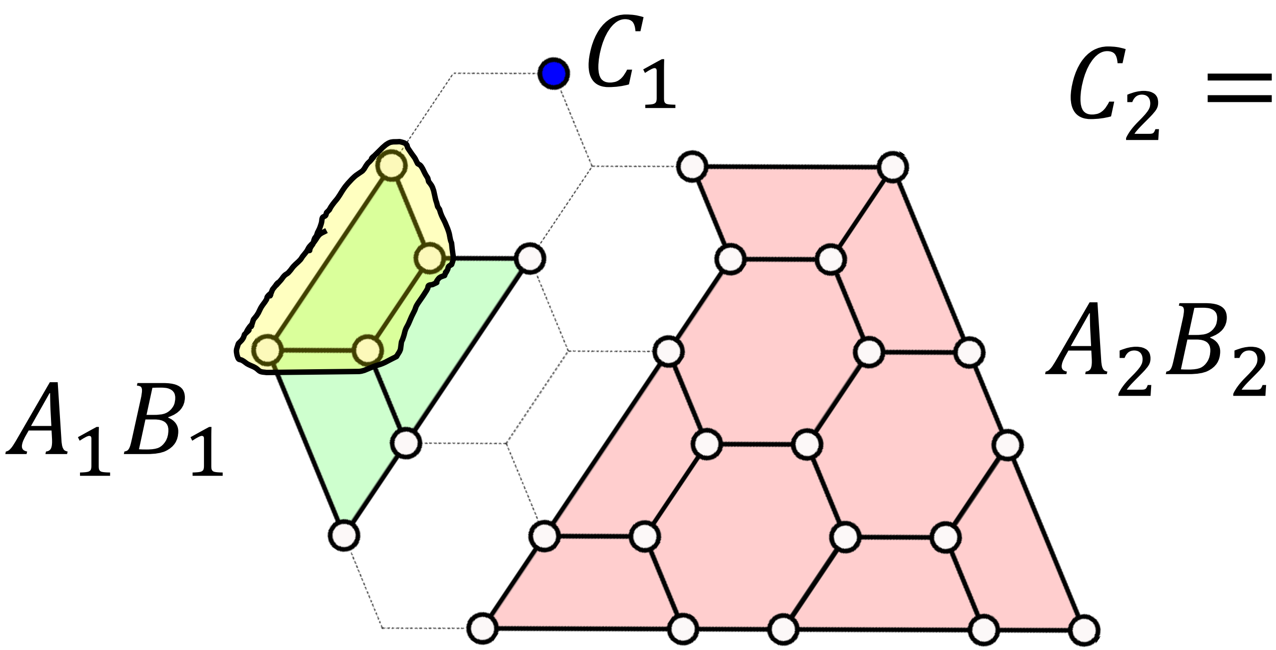
$$Z(BC) \sim Z(\beta_1)Z(C)$$

Stabilizer $Z(B_2C_2)$



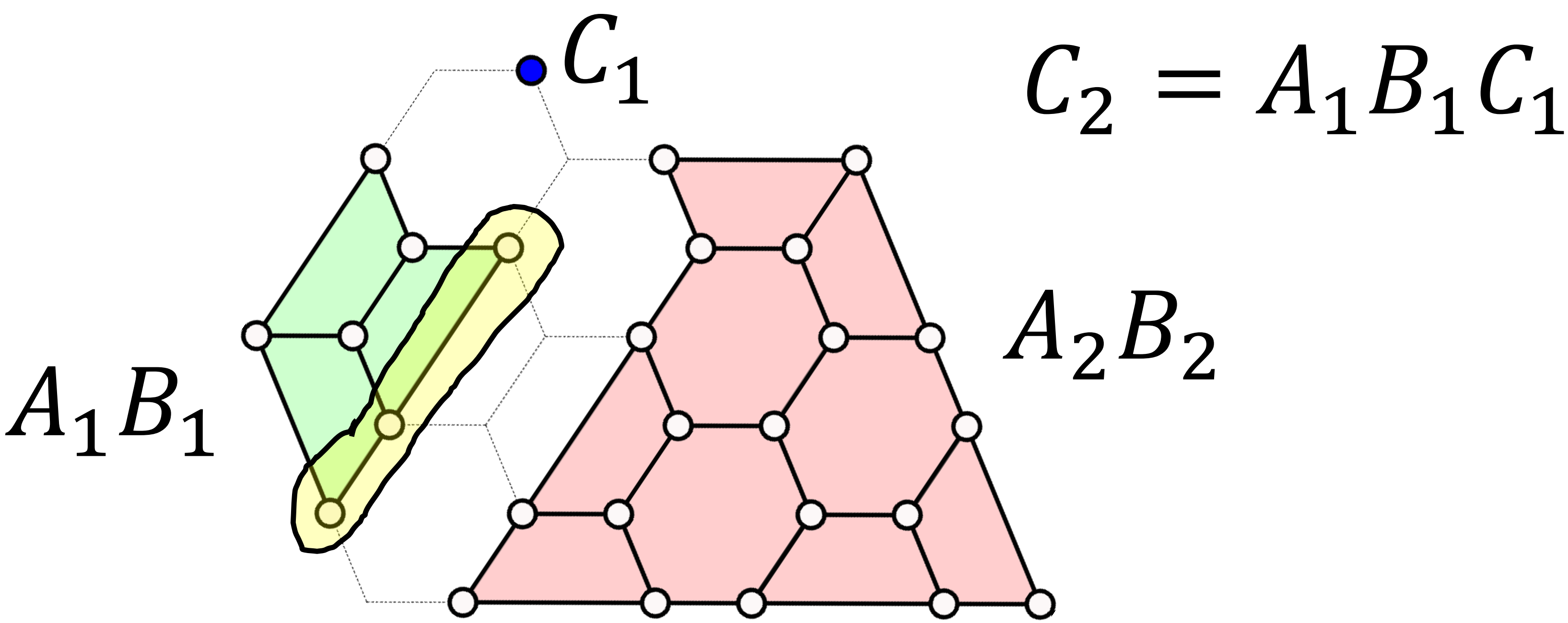
$$C_2 = A_1B_1C_1$$

Stabilizer $Z(B_2C_2)$



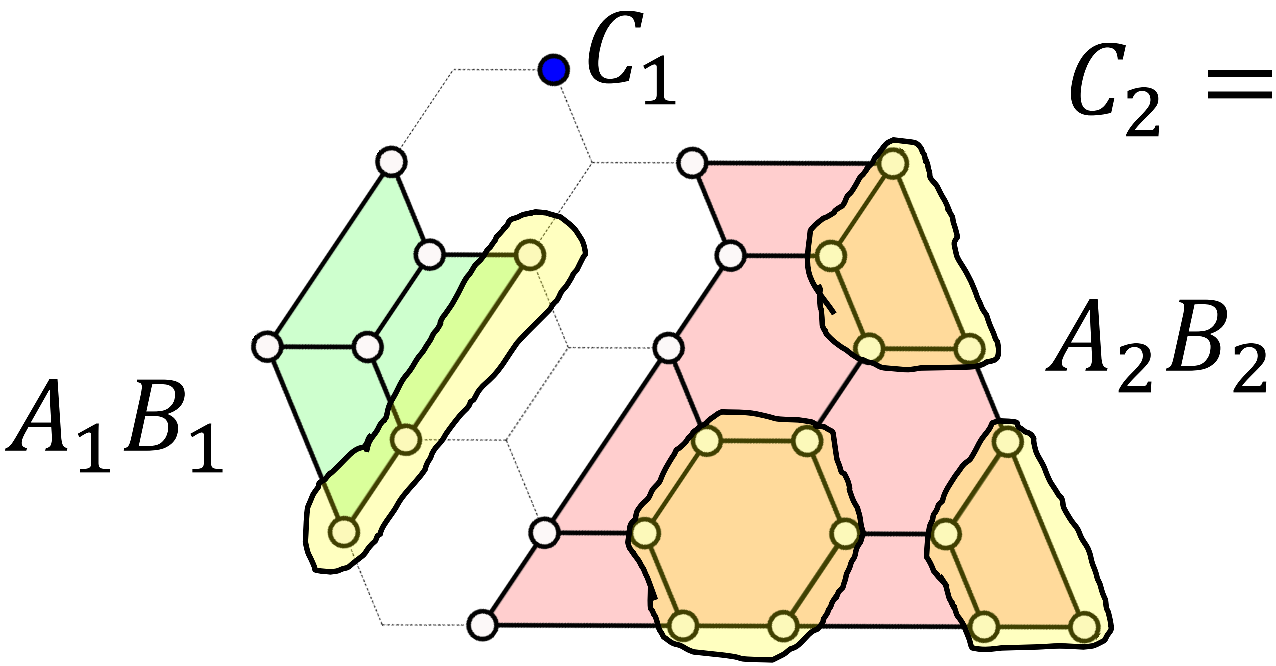
$C_2 = A_1B_1C_1$

Stabilizer $Z(B_2C_2)$

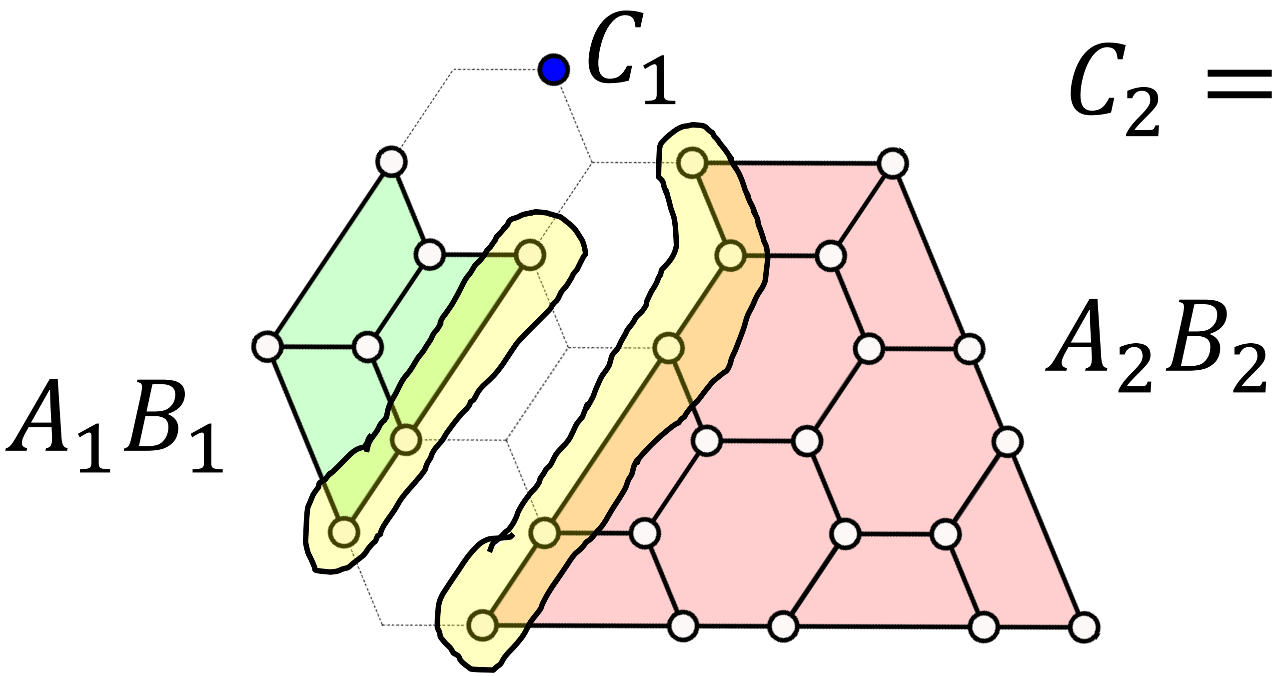


Stabilizer $Z(B_2C_2)$

$$C_2 = A_1 B_1 C_1$$

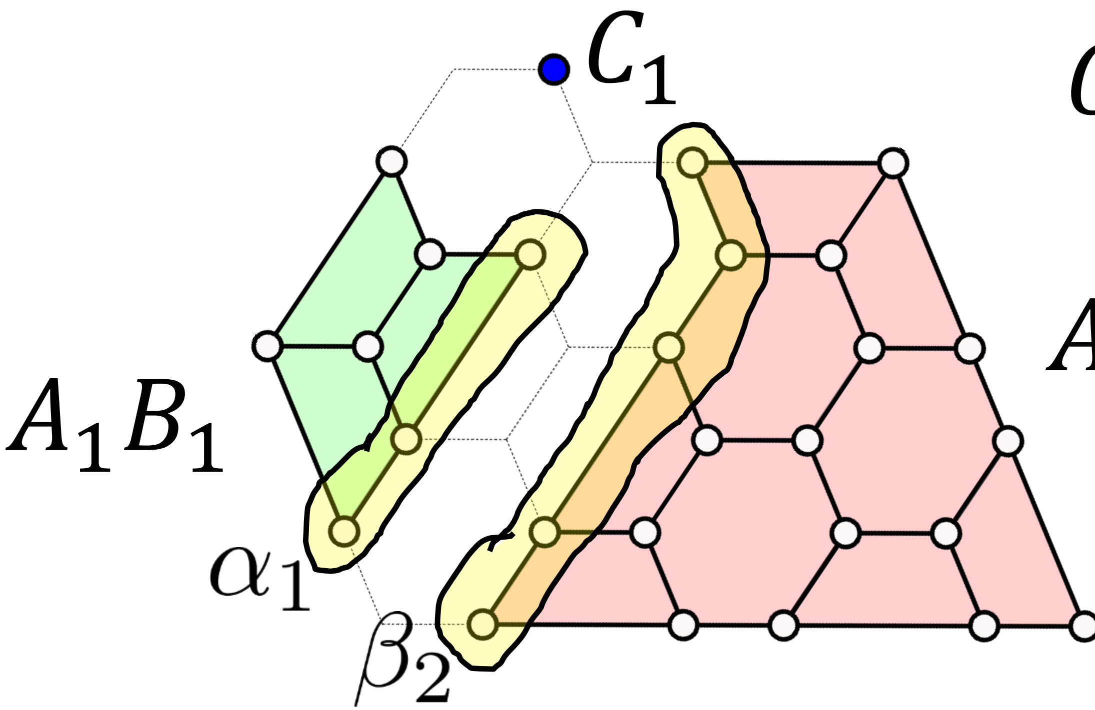


Stabilizer $Z(B_2C_2)$



$C_2 = A_1B_1C_1$

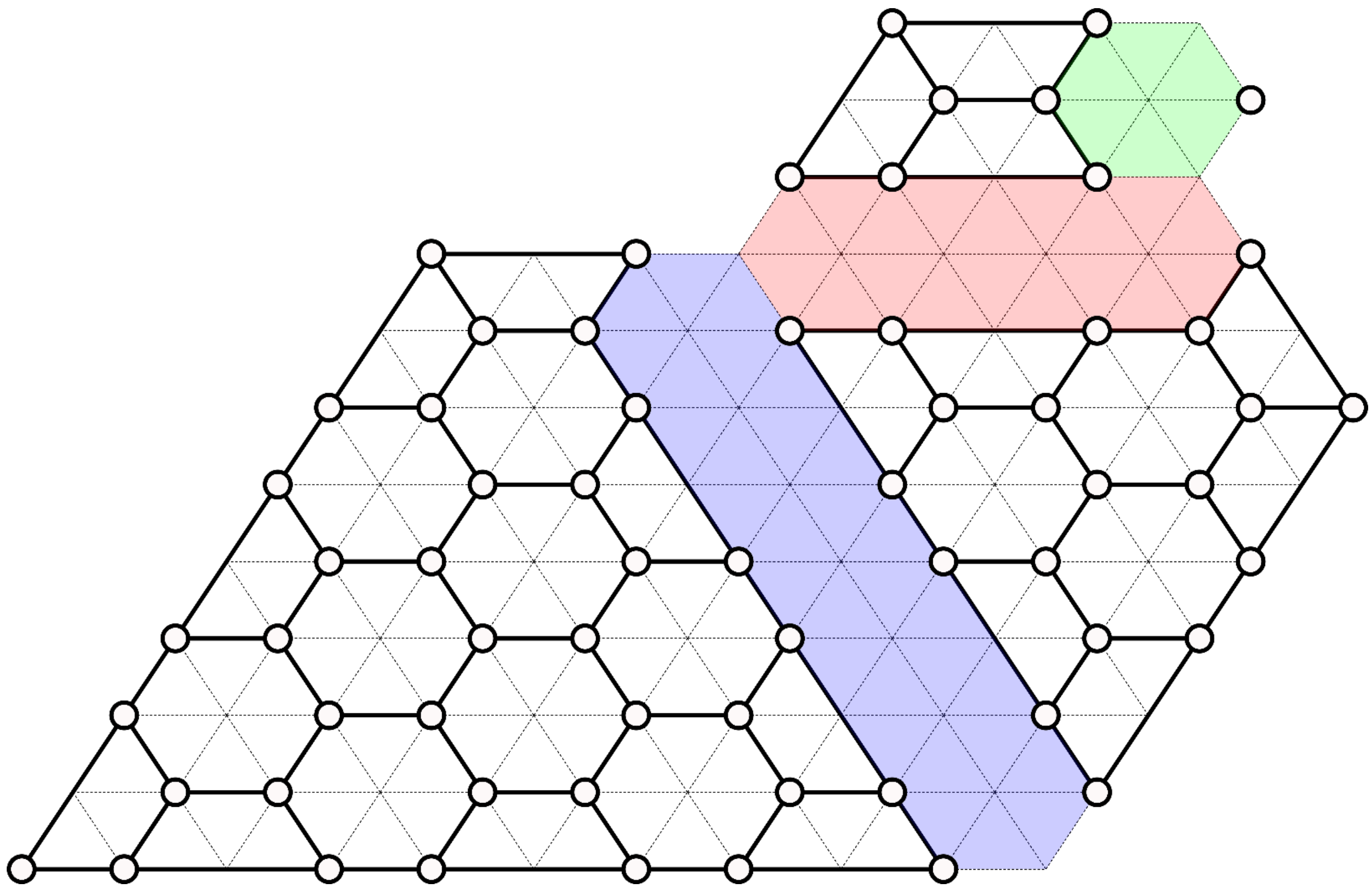
Stabilizer $Z(B_2C_2)$



$C_2 = A_1B_1C_1$

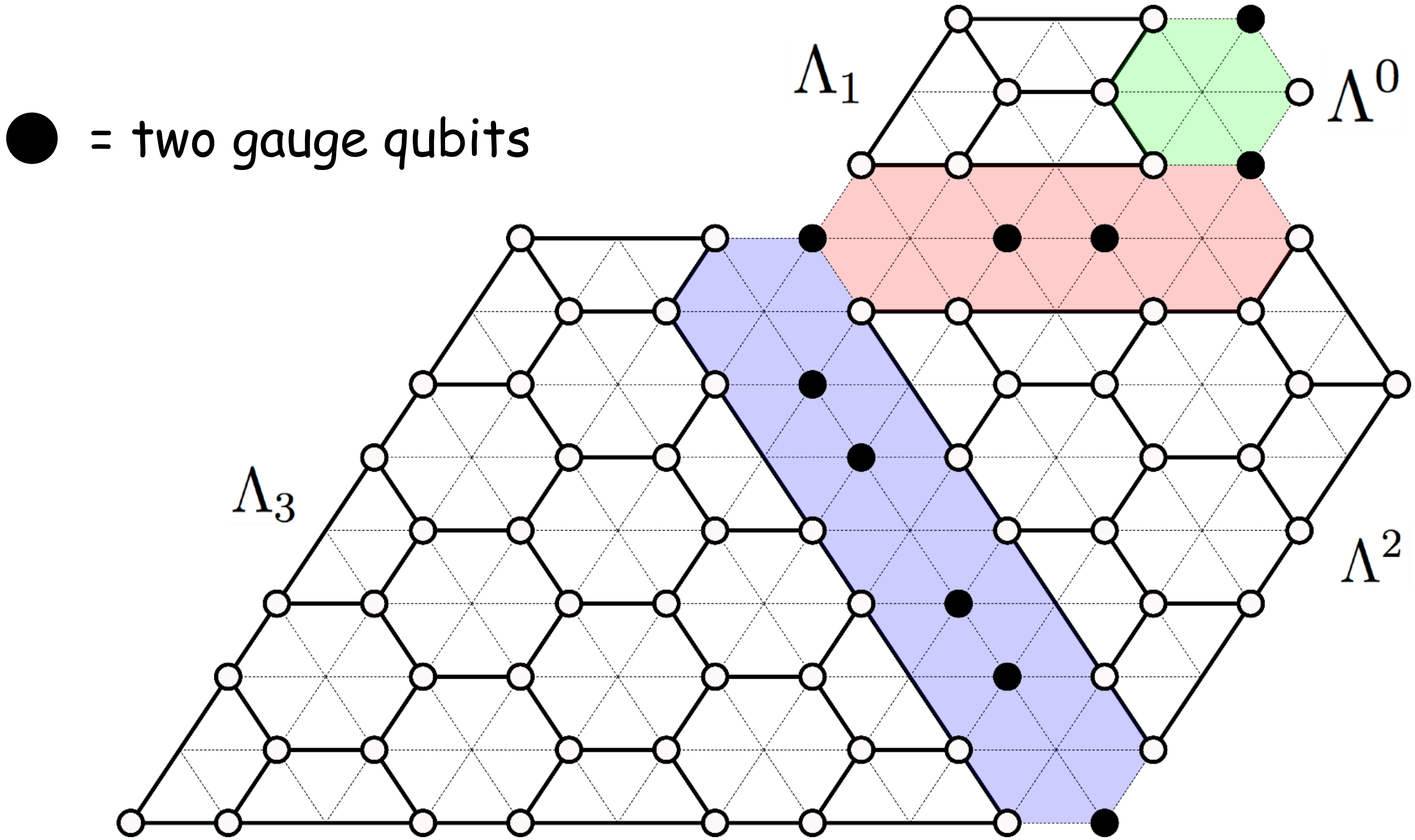
$$Z(B_2C_2) \sim Z(\beta_2)Z(\alpha_1)$$

The only non-local Z-stabilizers are those connecting the boundaries of consecutive color code patches.

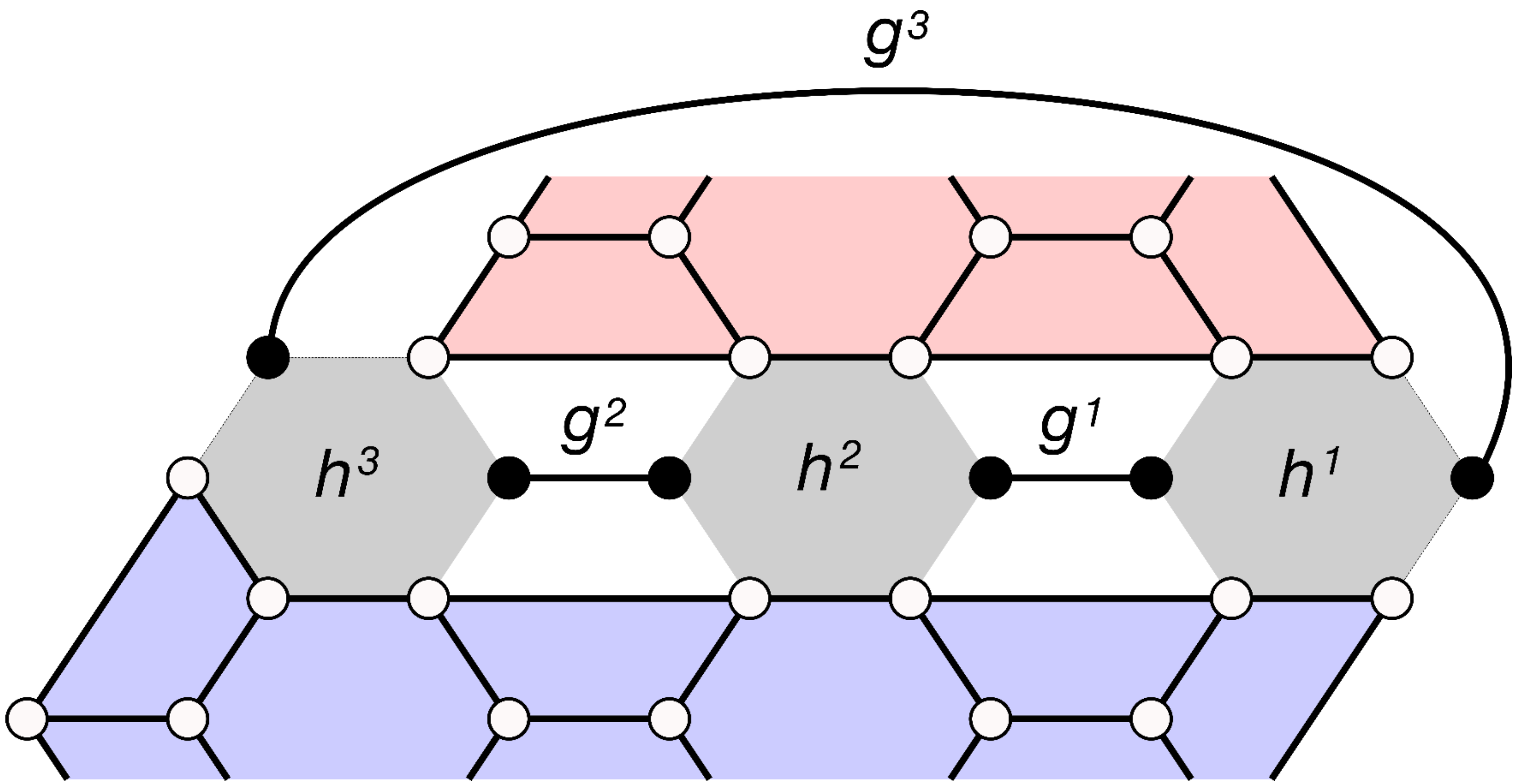


Step 2: decompose non-local Z-stabilizers into a product of local gauge operators

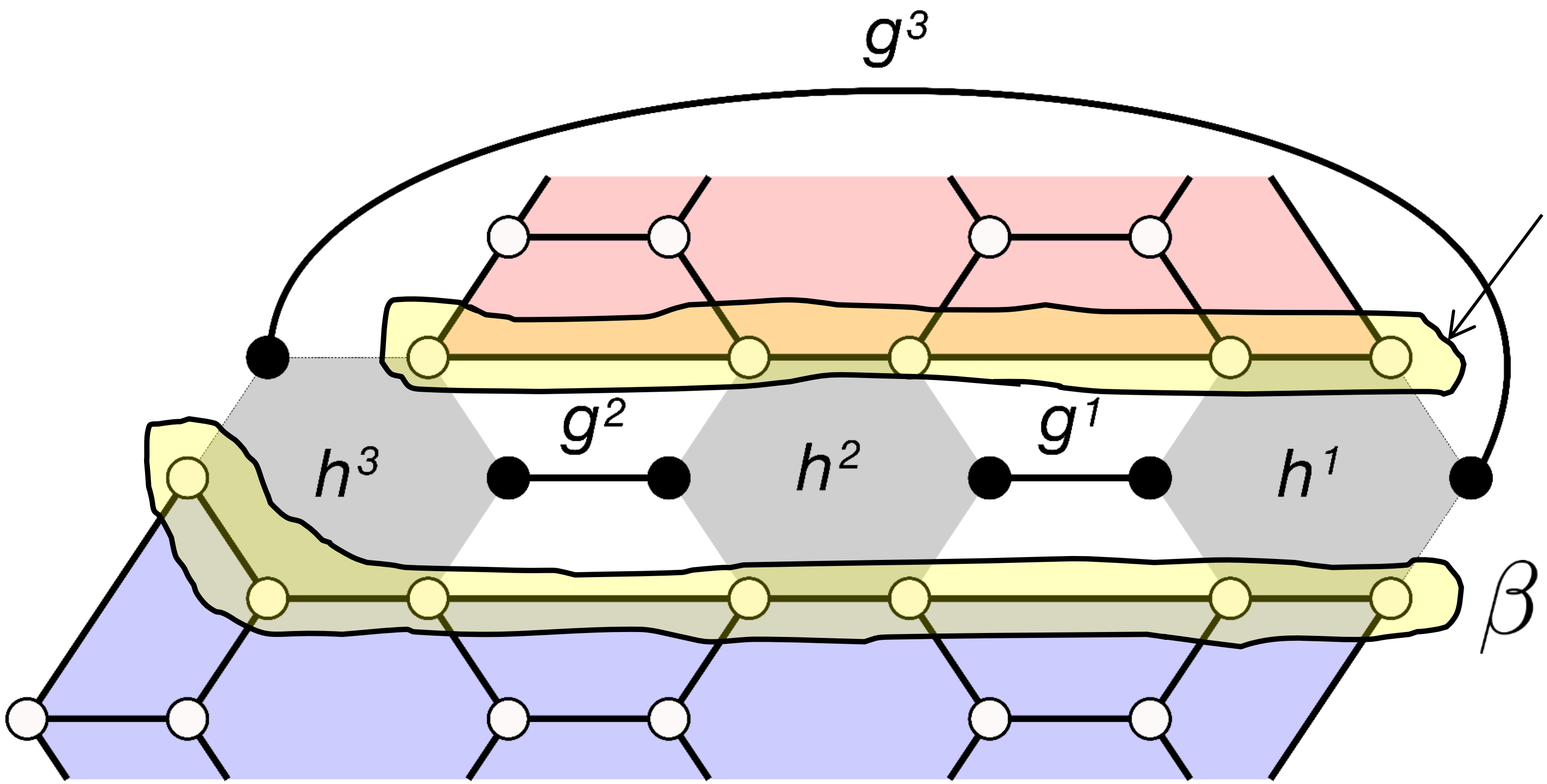
Connector region: add two gauge qubits at each site (ancillary qubits that do not store any information)



New Z-type gauge generators:



New Z-type gauge generators:



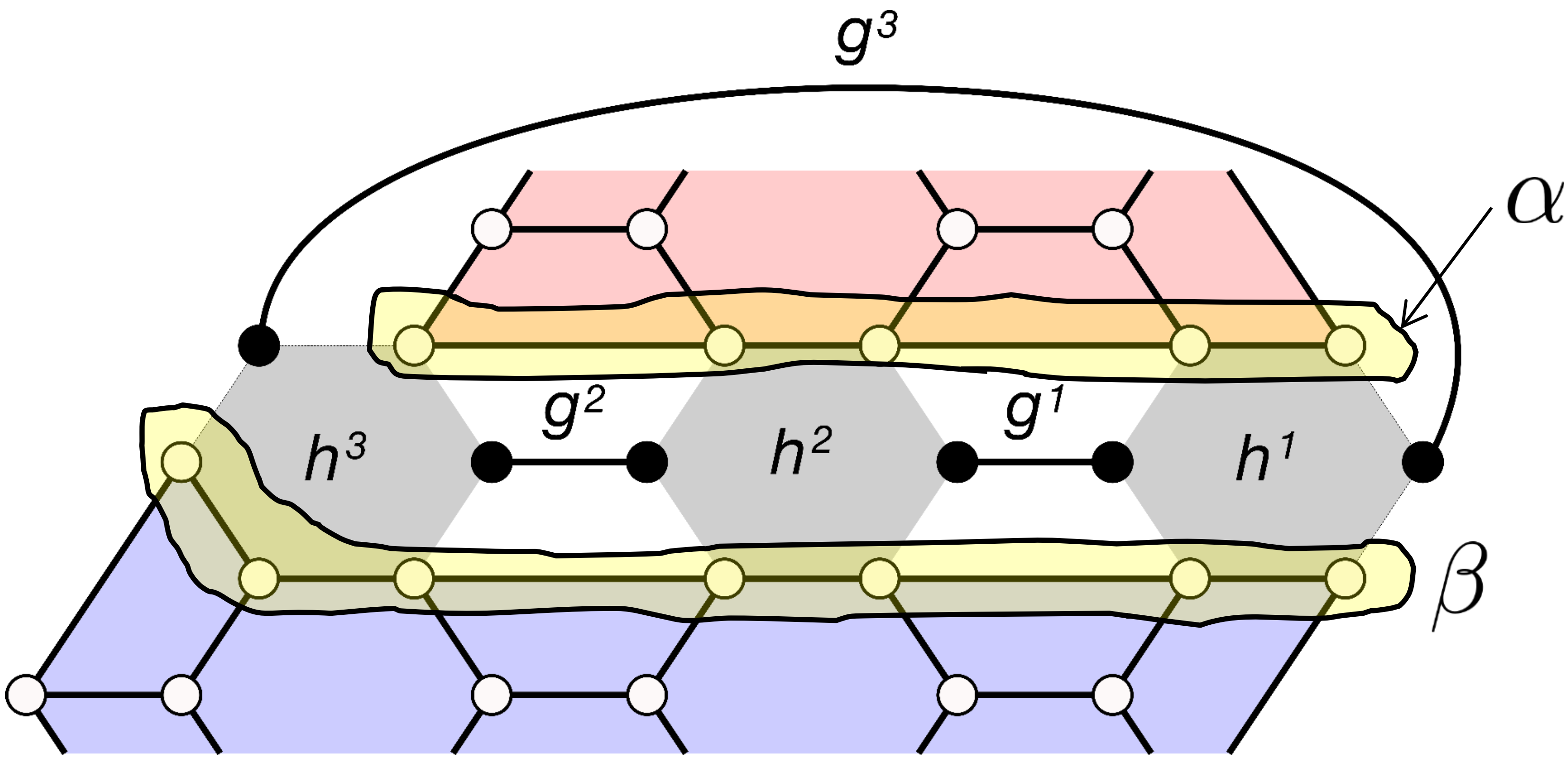
$$Z(\alpha\beta) \sim Z(h^1)Z(h^2)Z(h^3)Z(g^1)Z(g^2)Z(g^3)$$

non-local

local

non-local

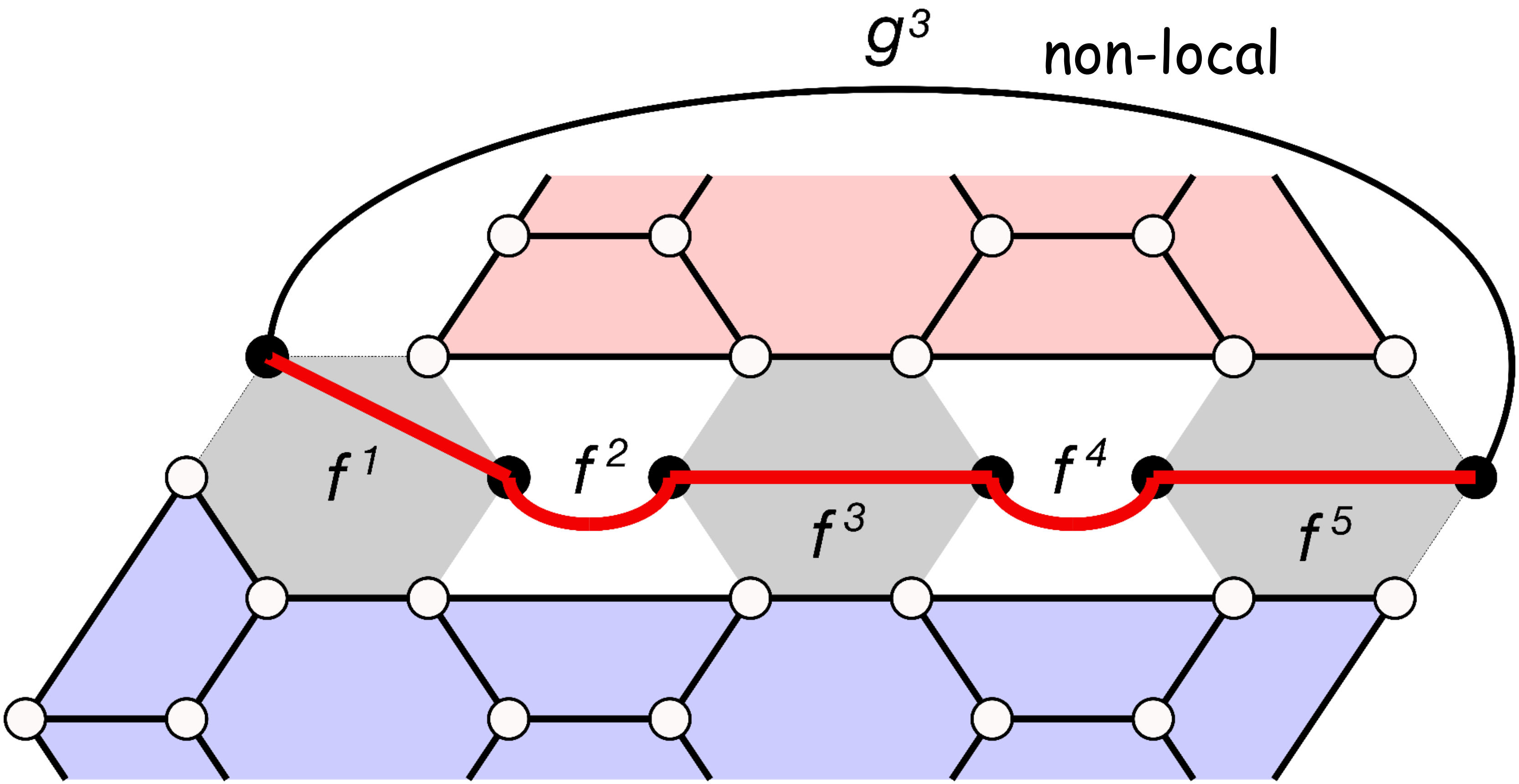
New Z-type gauge generators:



Similar to the lattice surgery method

Landahl and Ryan-Anderson (2014)

New Z-type gauge generators:



$$Z(g^3) = \prod_i Z(f^i)$$

non-local local

Now all Z-stabilizers are locally measurable: they can be decomposed into a product of local gauge generators.

Main technical work: compute the distance of the new code with the extra gauge qubits.

Step 3: decompose non-local X-stabilizers into a product of local gauge operators

Now all Z-stabilizers are locally measurable: they can be decomposed into a product of local gauge generators.

Main technical work: compute the distance of the new code with the extra gauge qubits.

~~Step 3: decompose non-local X-stabilizers into a product of local gauge operators~~

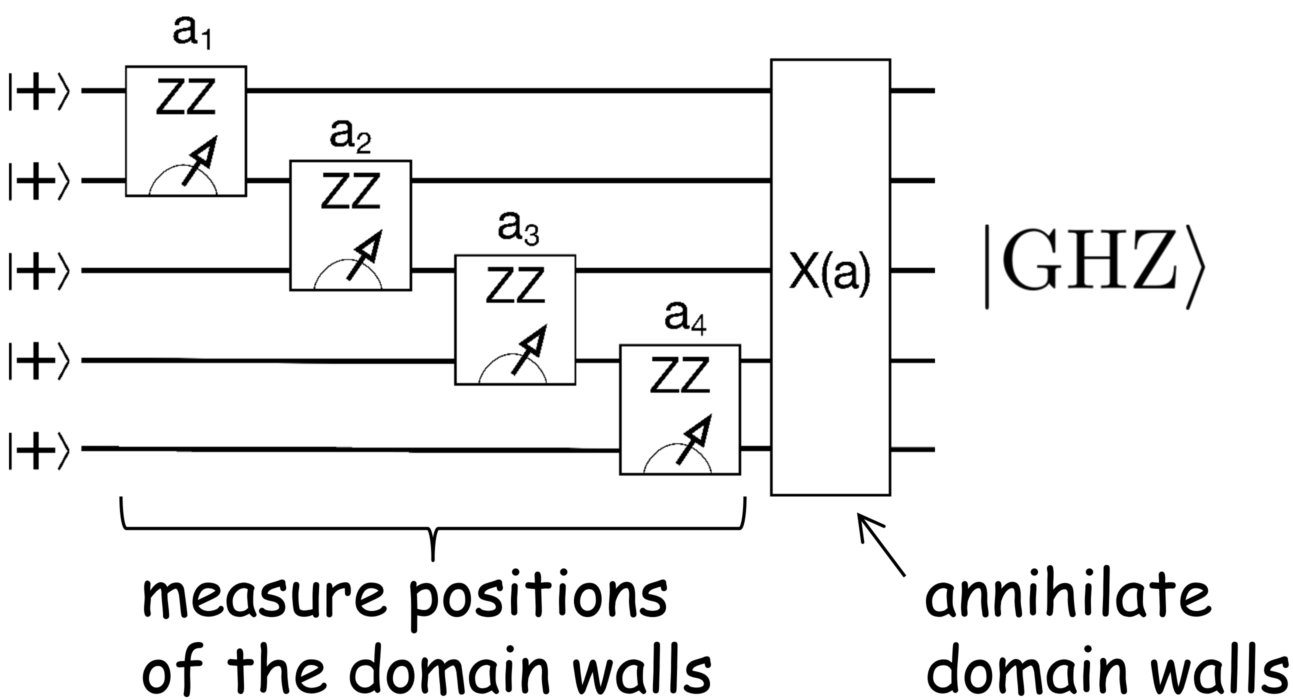
Better solution: don't measure any X-stabilizers

Illustrative example

$$|\mathrm{GHZ}\rangle \sim |00000\rangle + |11111\rangle$$

Stabilizers: $Z_1 Z_2, \ldots, Z_4 Z_5$ local

$X_1 X_2 X_3 X_4 X_5$ non-local

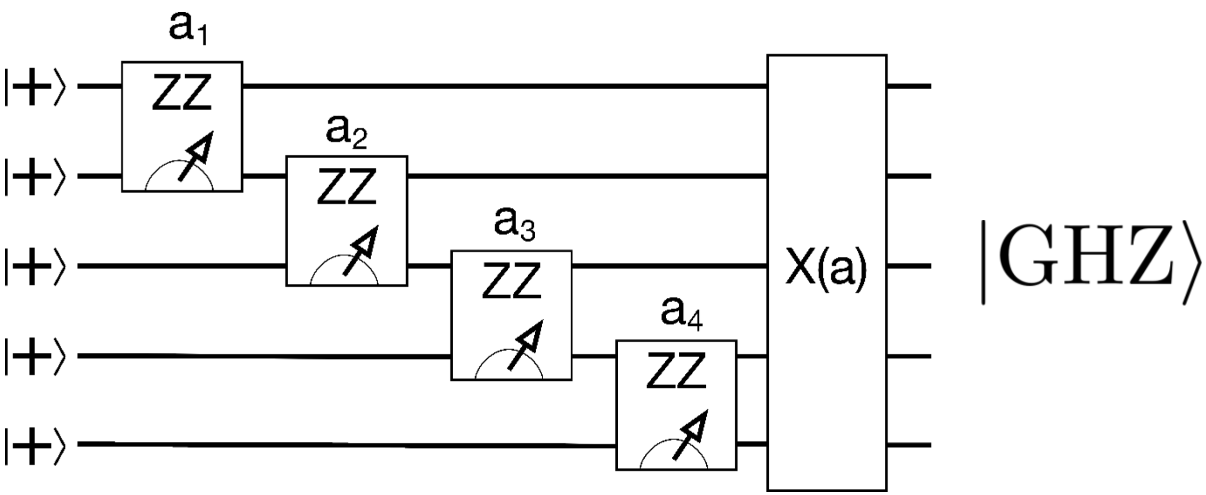


Illustrative example

$$|\mathrm{GHZ}\rangle \sim |00000\rangle + |11111\rangle$$

Stabilizers: $Z_1 Z_2, \ldots, Z_4 Z_5$ local

$X_1 X_2 X_3 X_4 X_5$ non-local

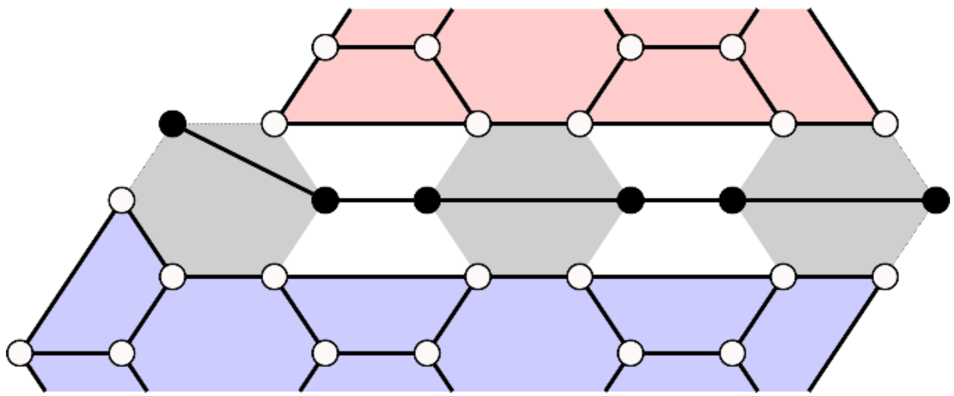


No need to measure the non-local stabilizer $X_1 X_2 X_3 X_4 X_5$

It is already in the stabilizer group of the initial state.

$|+\rangle^{\otimes n}$ $|\mathrm{GHZ}\rangle$

Extended Color Code Doubled Color Code

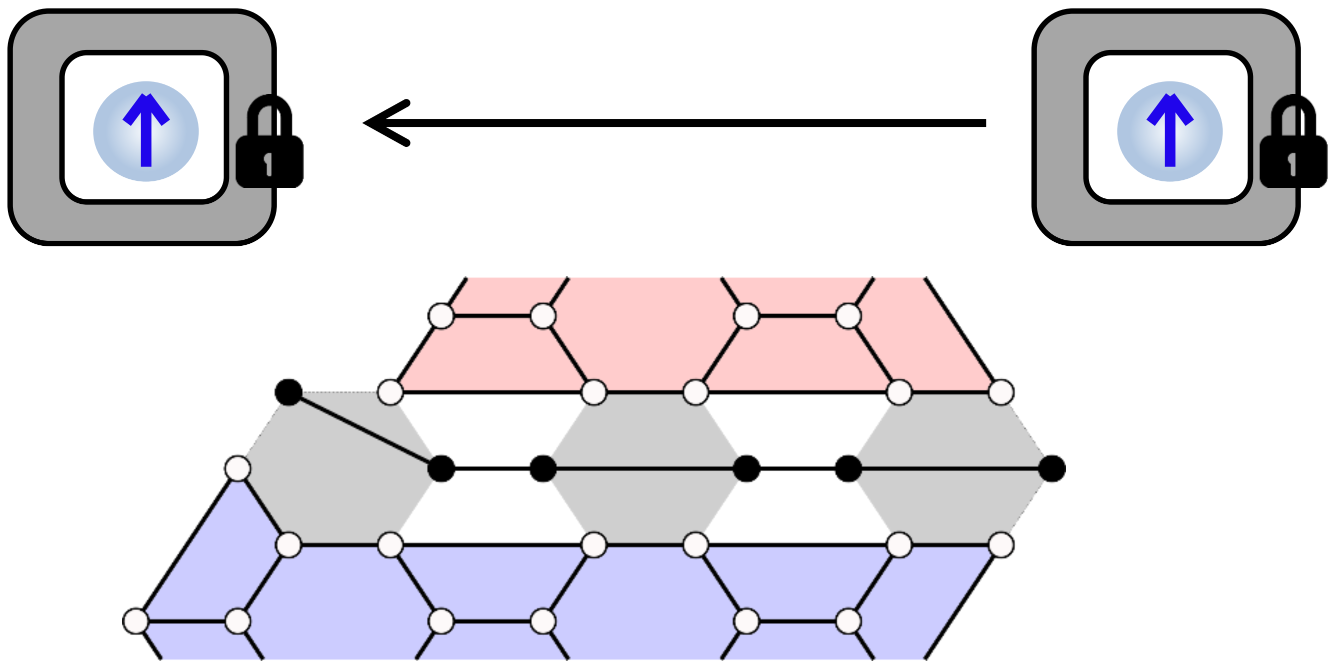


Measure Z-generators on all edges

Measure Z-generators on all faces
in the connector regions

$|+\rangle^{\otimes n}$ $|\mathrm{GHZ}\rangle$

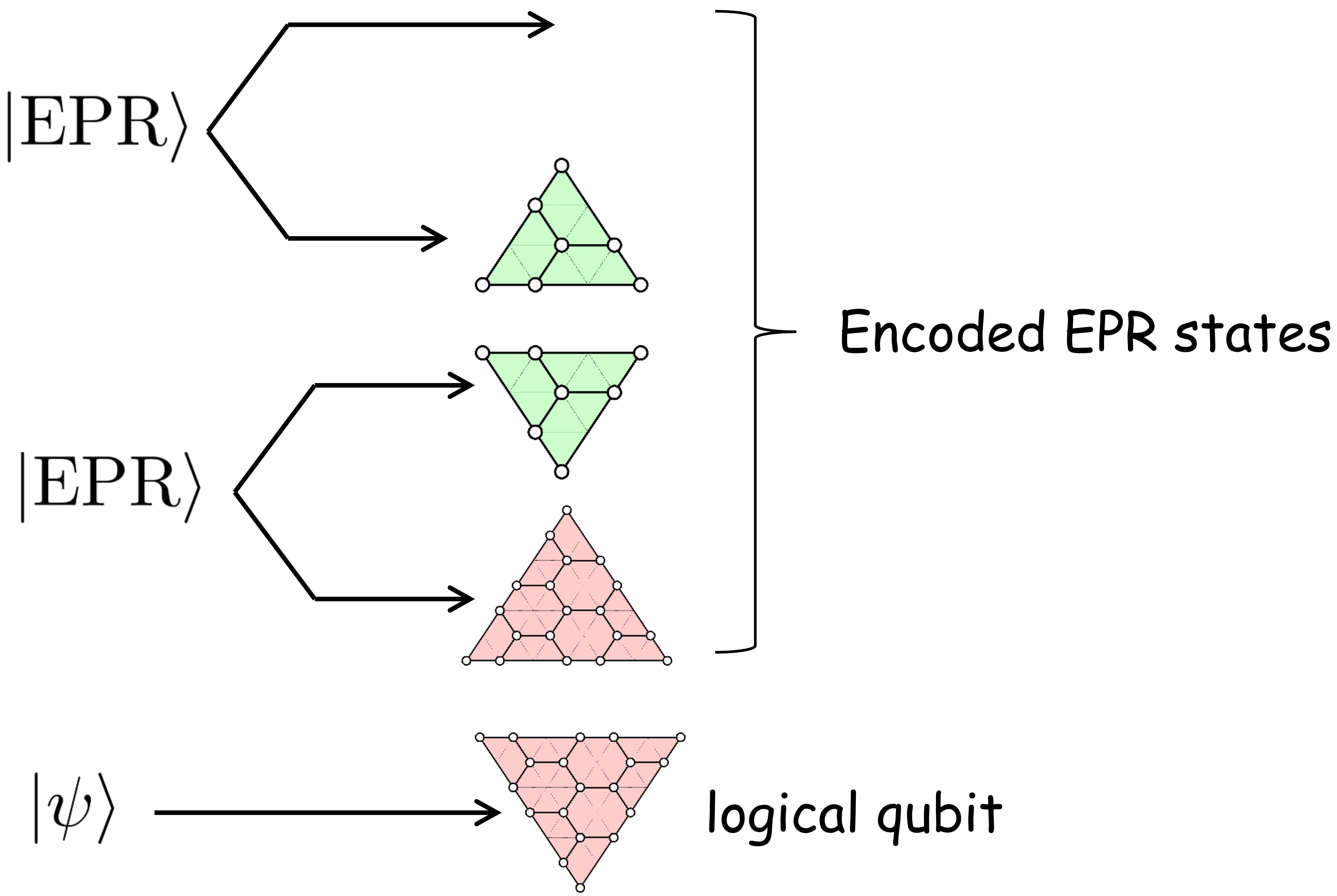
Extended Color Code Doubled Color Code



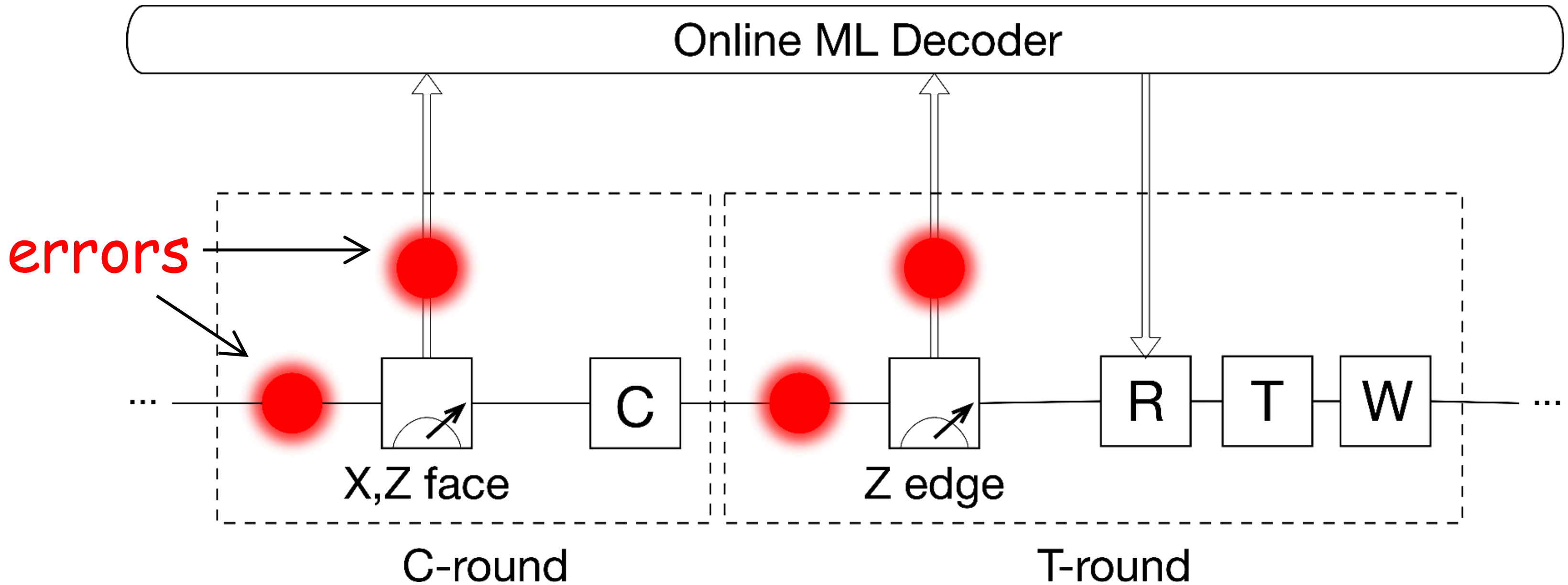
Measure both X- and Z-generators on all faces.

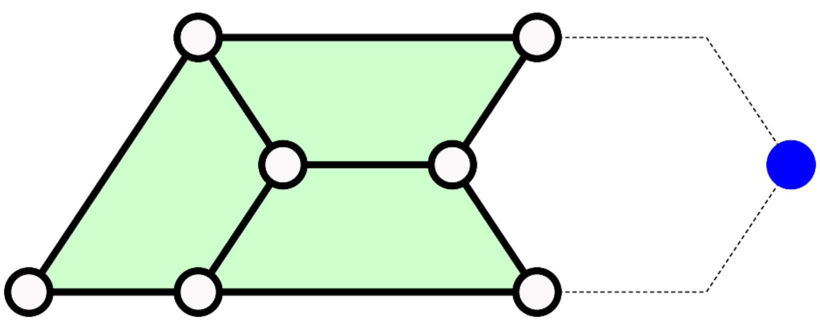
Measure both X- and Z-generators on edges in the connector regions

Extended Color Code:



Simulation of logical Clifford+T circuits





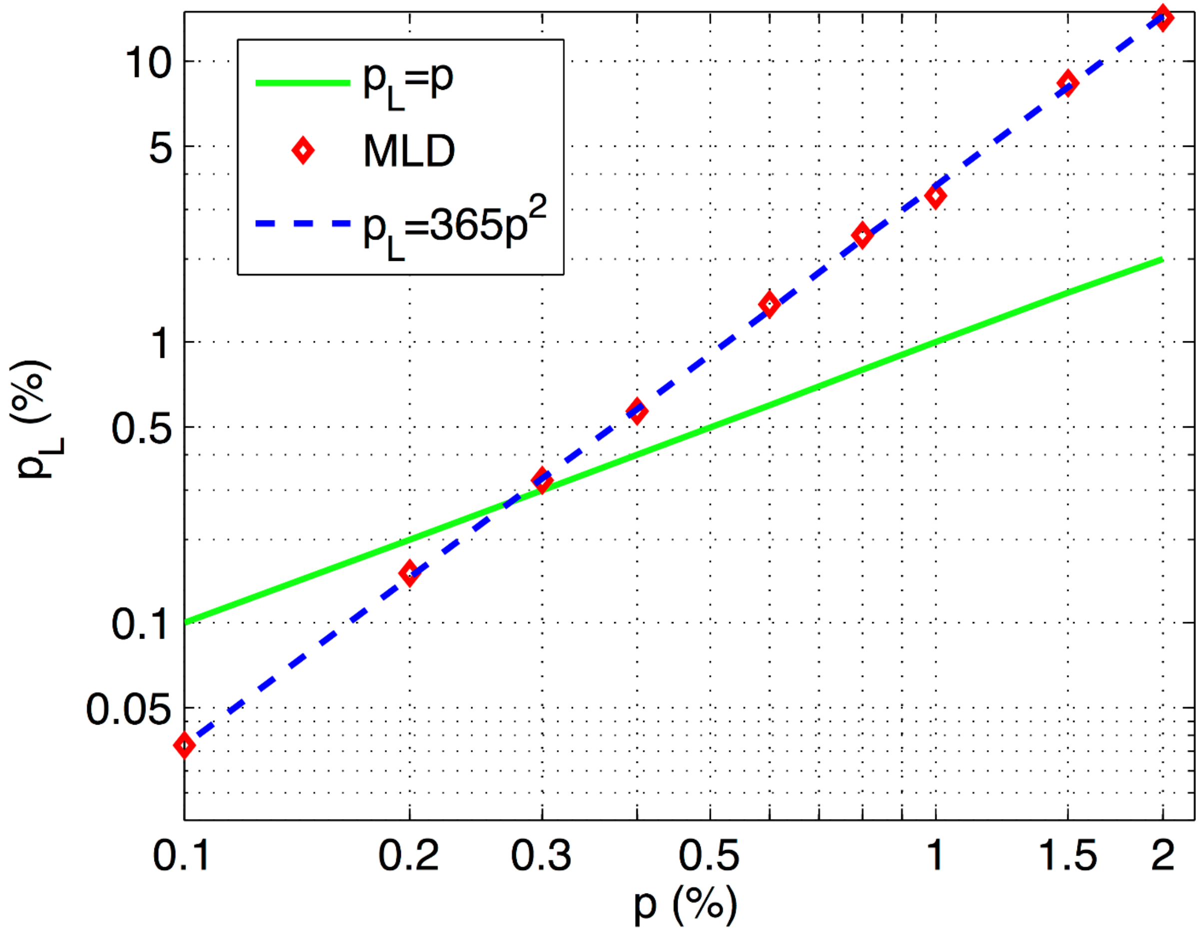
C : transversal Clifford gate (random)

T : transversal T-gate

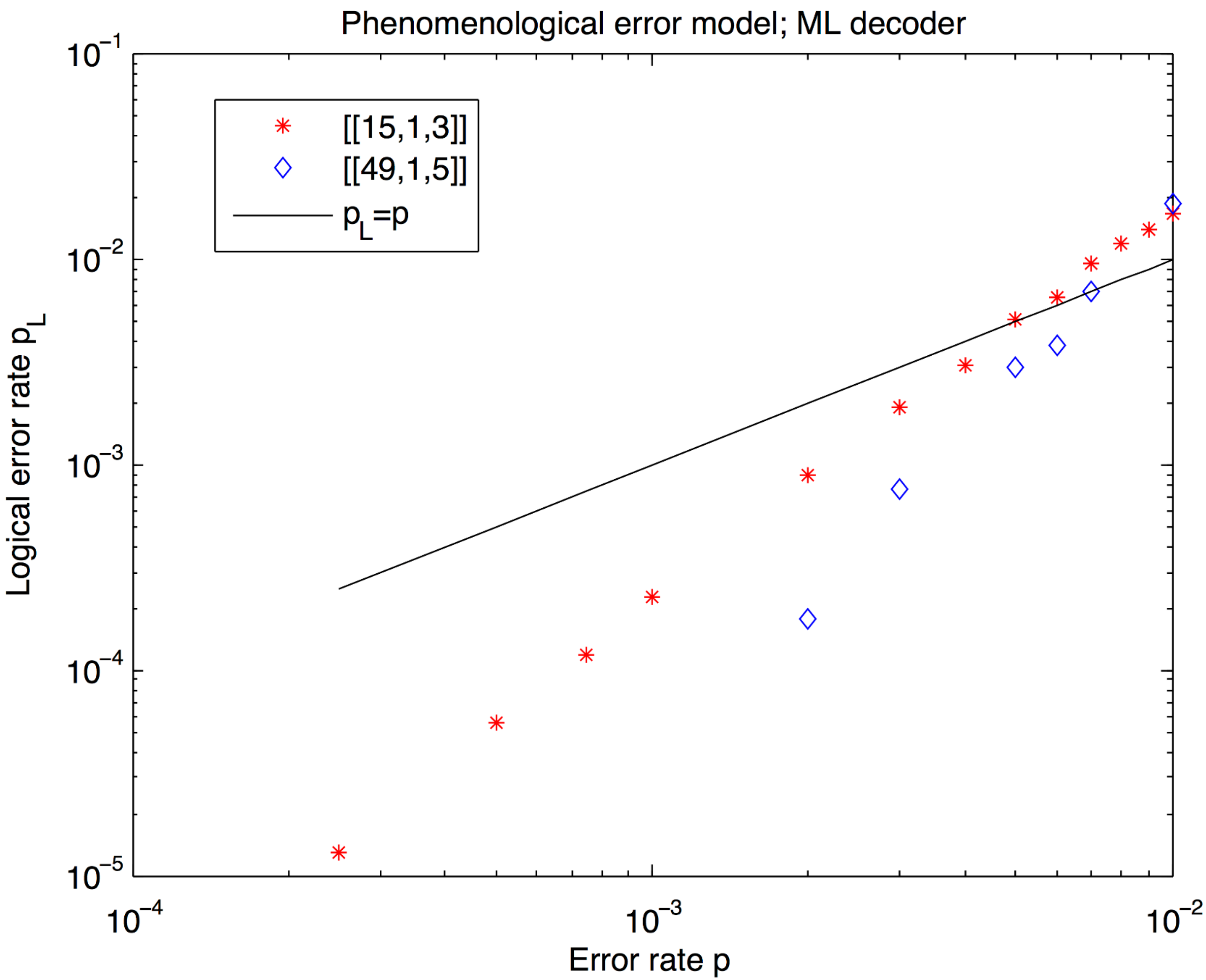
R : error correction + gauge fixing

t : the average number of T-gates implemented before the first logical error

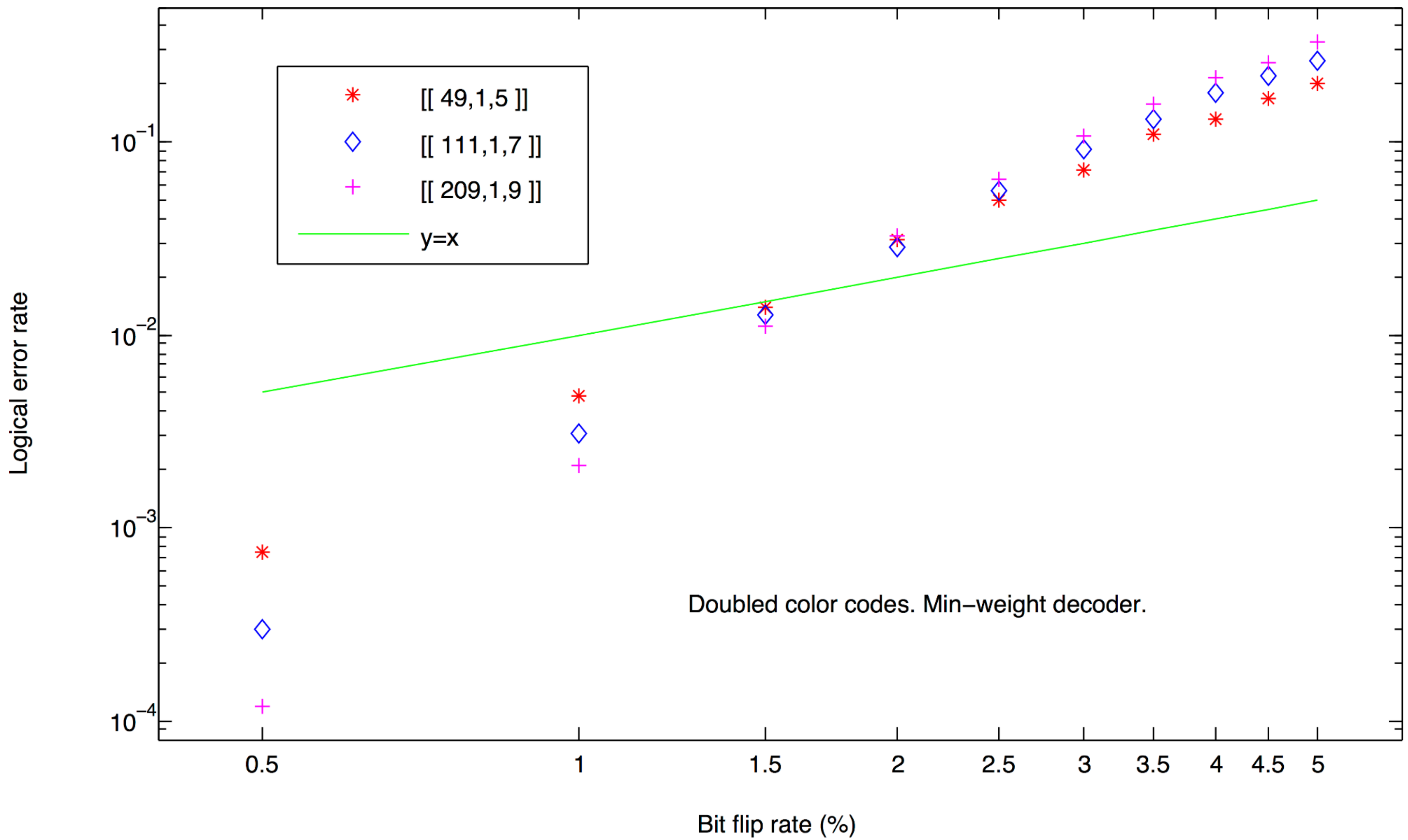
Logical error probability: $p_L = 1/t$



Depolarizing noise + syndrome errors

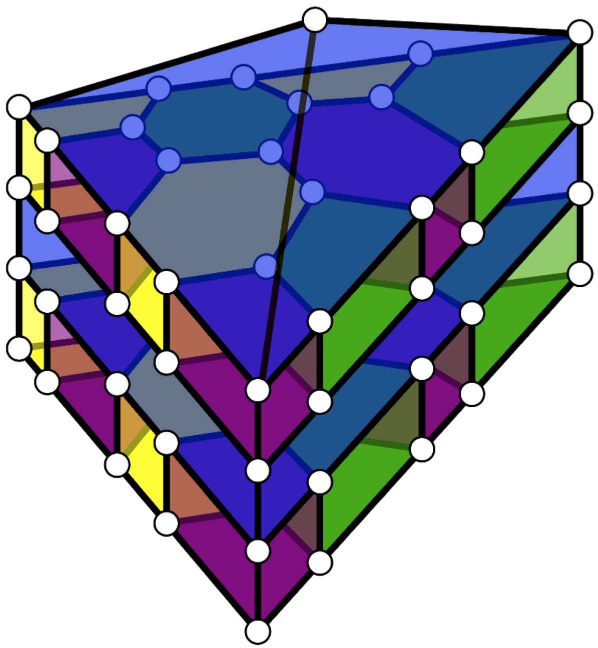


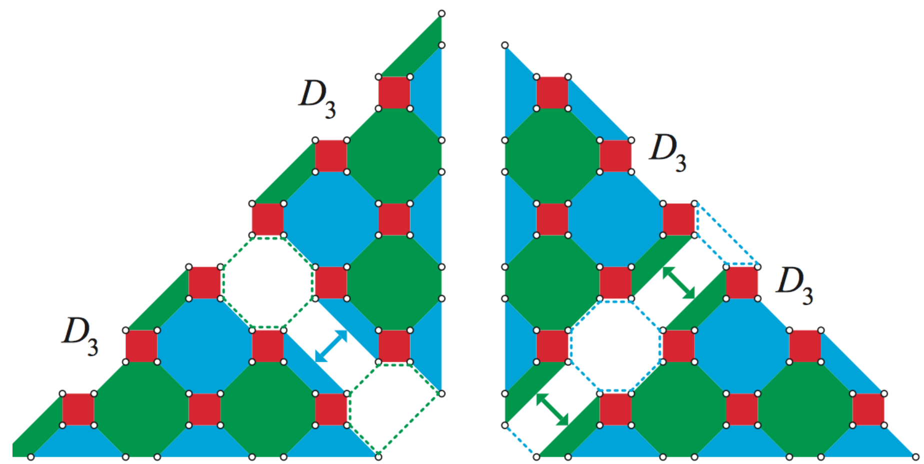
Bit-flip noise, no syndrome errors



Related work:

Jochym-O'Connor and Bartlett,
"Stacked codes: universal quantum computation in a two-dimensional layout", arxiv:1509.04255

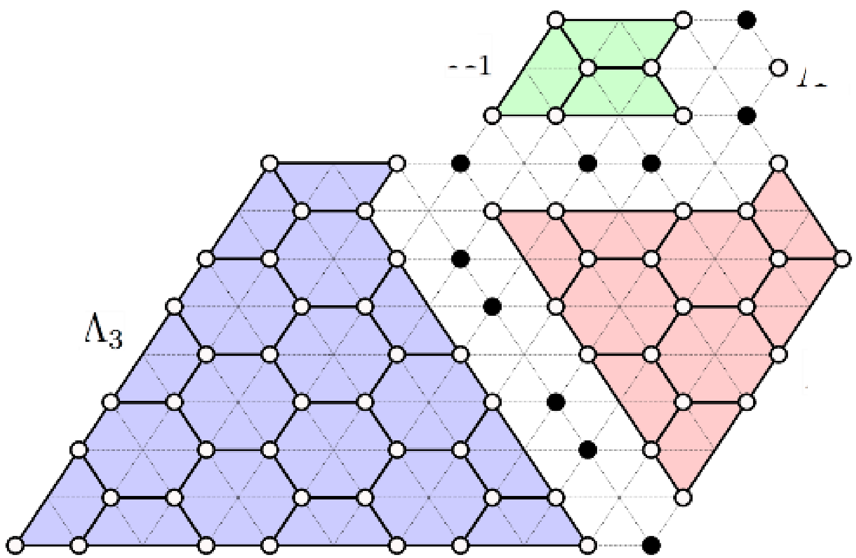




Jones, Brooks, Harrington,
"Gauge color codes in two dimensions",
arxiv:1512.04193

Summary

- Subsystem codes on the honeycomb lattice with two qubits per site. Local gauge generators.
- Infinite family with a diverging code distance.
- Transversal Clifford+T gates by gauge fixing.



$$[[n, 1, d]]$$

$$d = 2t + 1$$

$$n = 2t^3 + 8t^2 + 6t - 1$$