# Algebraic Coding Problems from Quantum Fault Tolerance

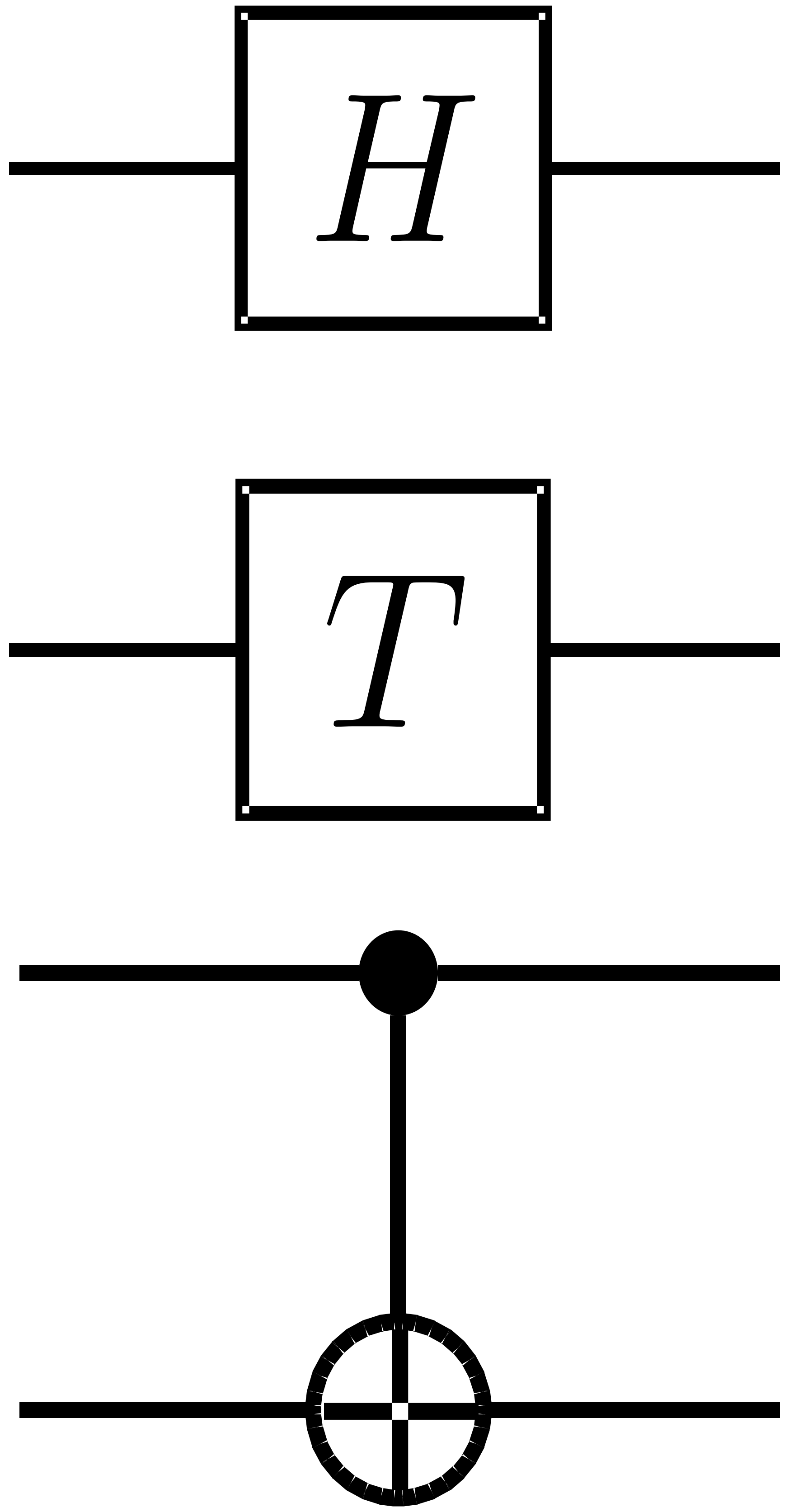
**Application-Driven Coding Theory Workshop**
**March 5, 2024**
**Simons Institute, UC Berkeley, CA**

---

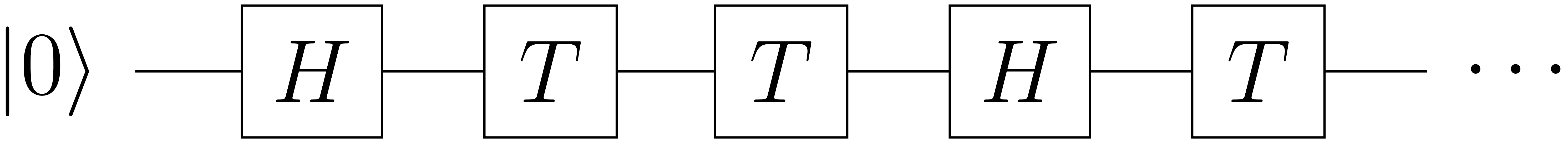
NARAYANAN RENGASWAMY
DEPT. OF ELECTRICAL AND COMPUTER ENGINEERING
UNIVERSITY OF ARIZONA

# Quantum Circuits

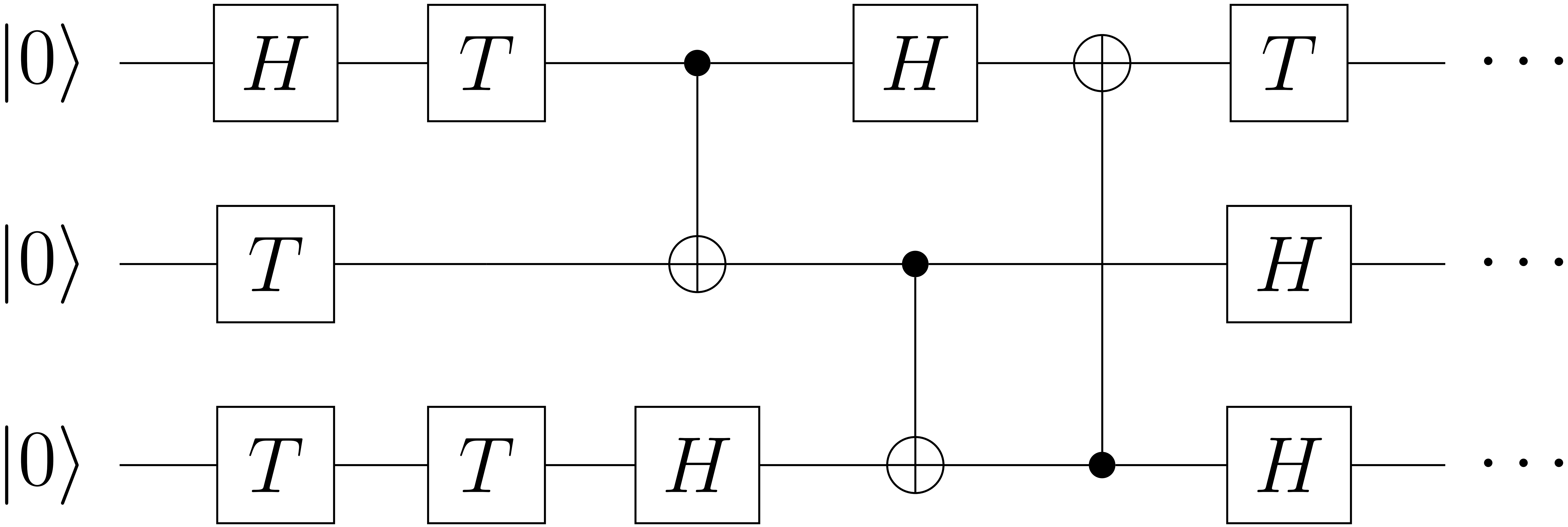
## Universal set of “gates”



## $k = 1$ qubit

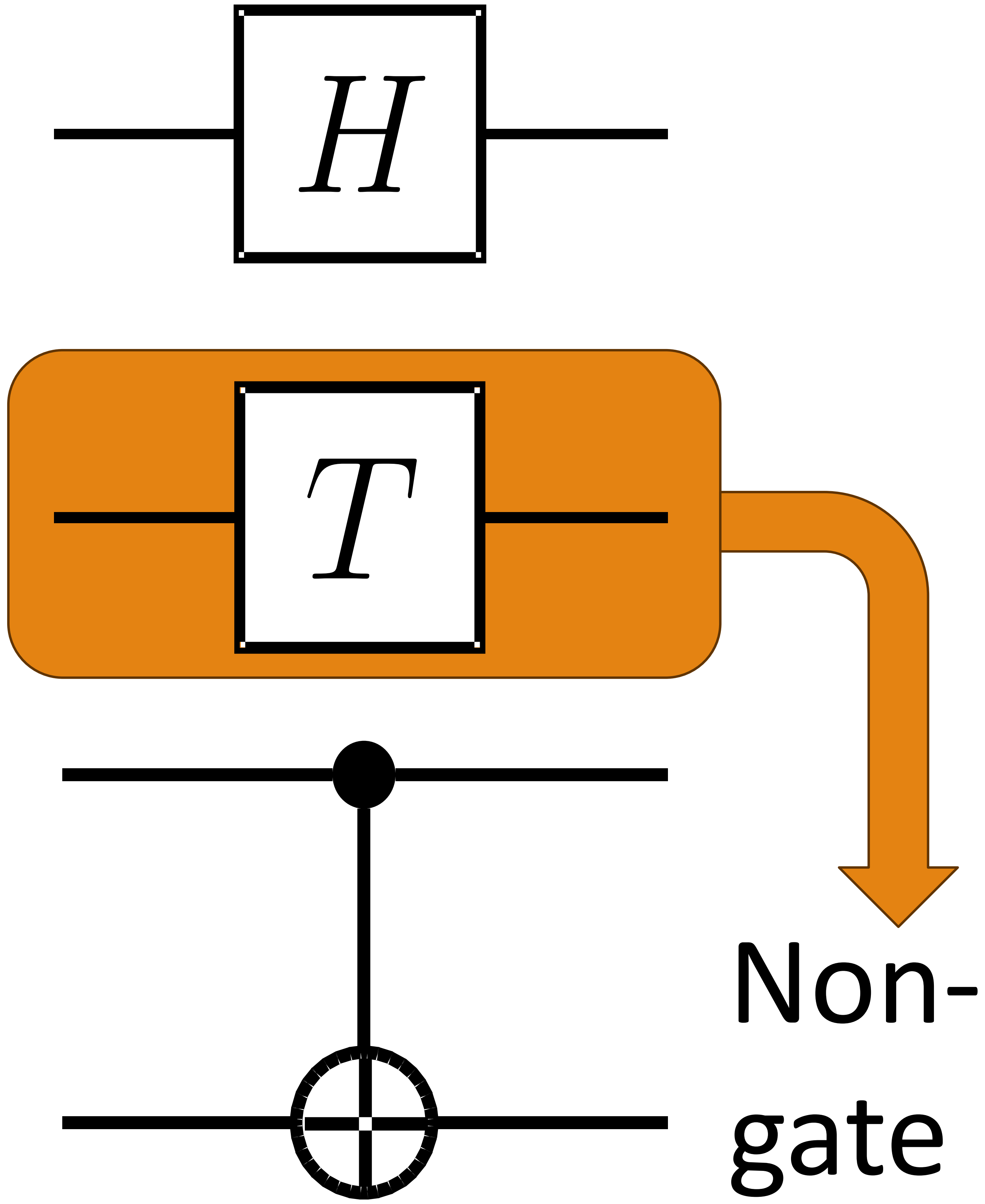


## $k = 3$ qubits

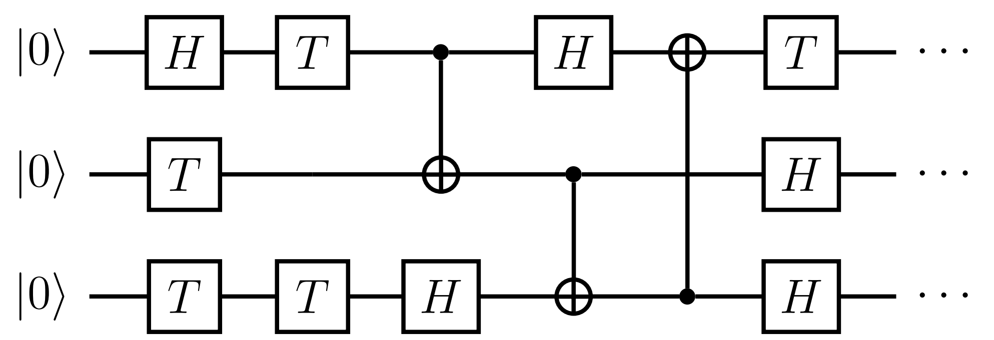


# QEC: Quantum Error Correction

Universal set of “gates”



Non-Clifford gate



**Noisy**

Logical gates

$$\begin{tikzcd}
k \text{ qubits} \quad |\psi\rangle_L \arrow[r] \arrow[d, "\text{QECC Encode}"'] & |\phi\rangle_L \\
n \text{ qubits} \quad \overline{|\psi\rangle} \arrow[r] & \overline{|\phi\rangle} \arrow[u, "\text{QECC Decode}"']
\end{tikzcd}$$

Translate (Synthesize)

Physical gates

**Fault-tolerant**

???

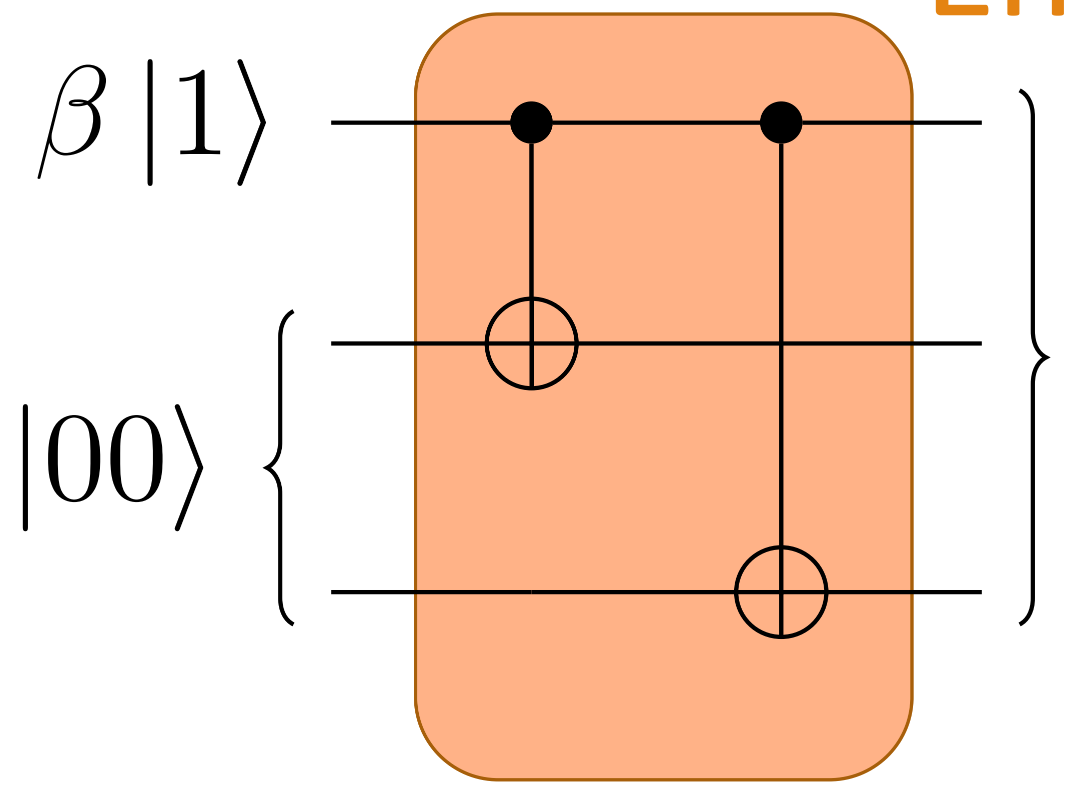
QECC: Quantum Error Correcting Code

# QEC: Syndrome-Based Error Correction

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

$k = 1$

Logical Qubit



Encoding

$\overline{|\psi\rangle} = \alpha|000\rangle + \beta|111\rangle$

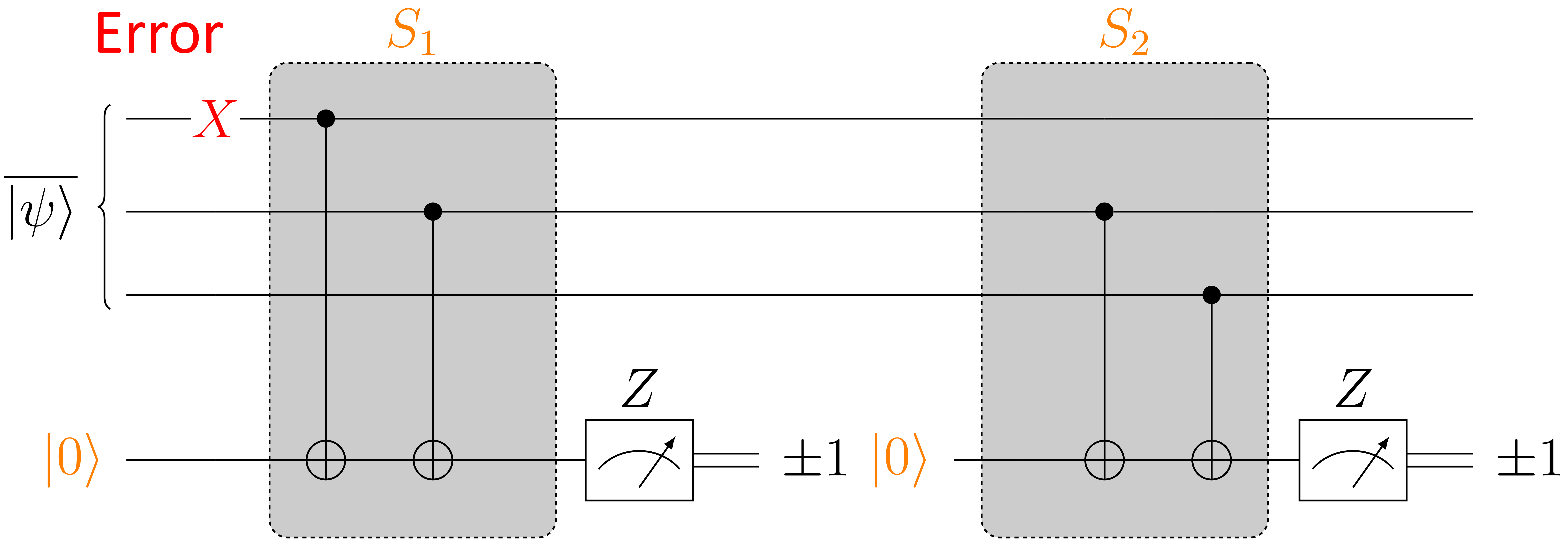
$n = 3$

Physical Qubits

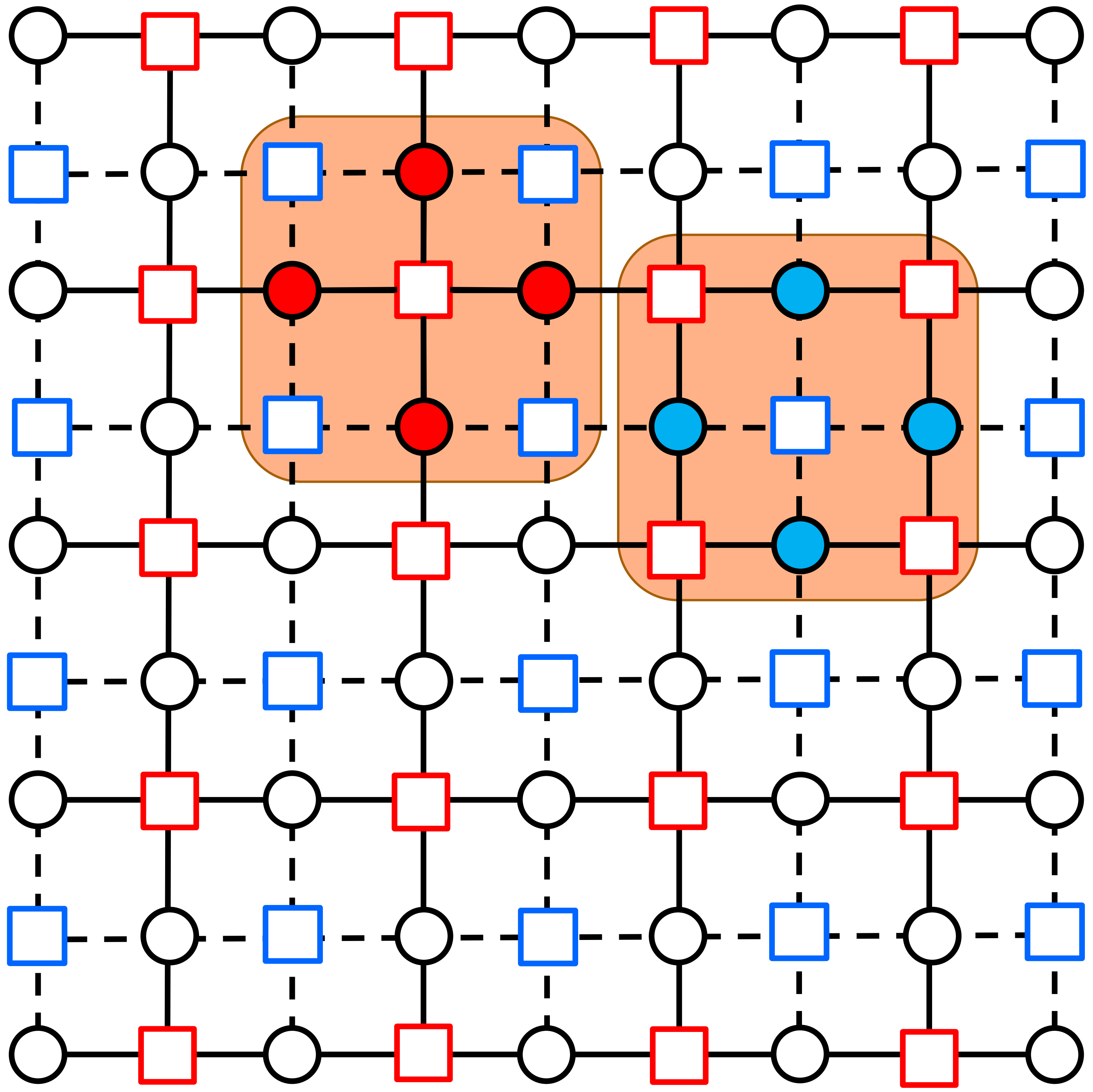
+1-Eigenvectors of $S_1 = Z \otimes Z \otimes I$ and $S_2 = I \otimes Z \otimes Z$

$$S_1\overline{|\psi\rangle} = \overline{|\psi\rangle}\ ,\ S_2\overline{|\psi\rangle} = \overline{|\psi\rangle}$$

Measure the stabilizer generators $S_1 = ZZI$ and $S_2 = IZZ$:



# Towards QEC with Constant Overhead



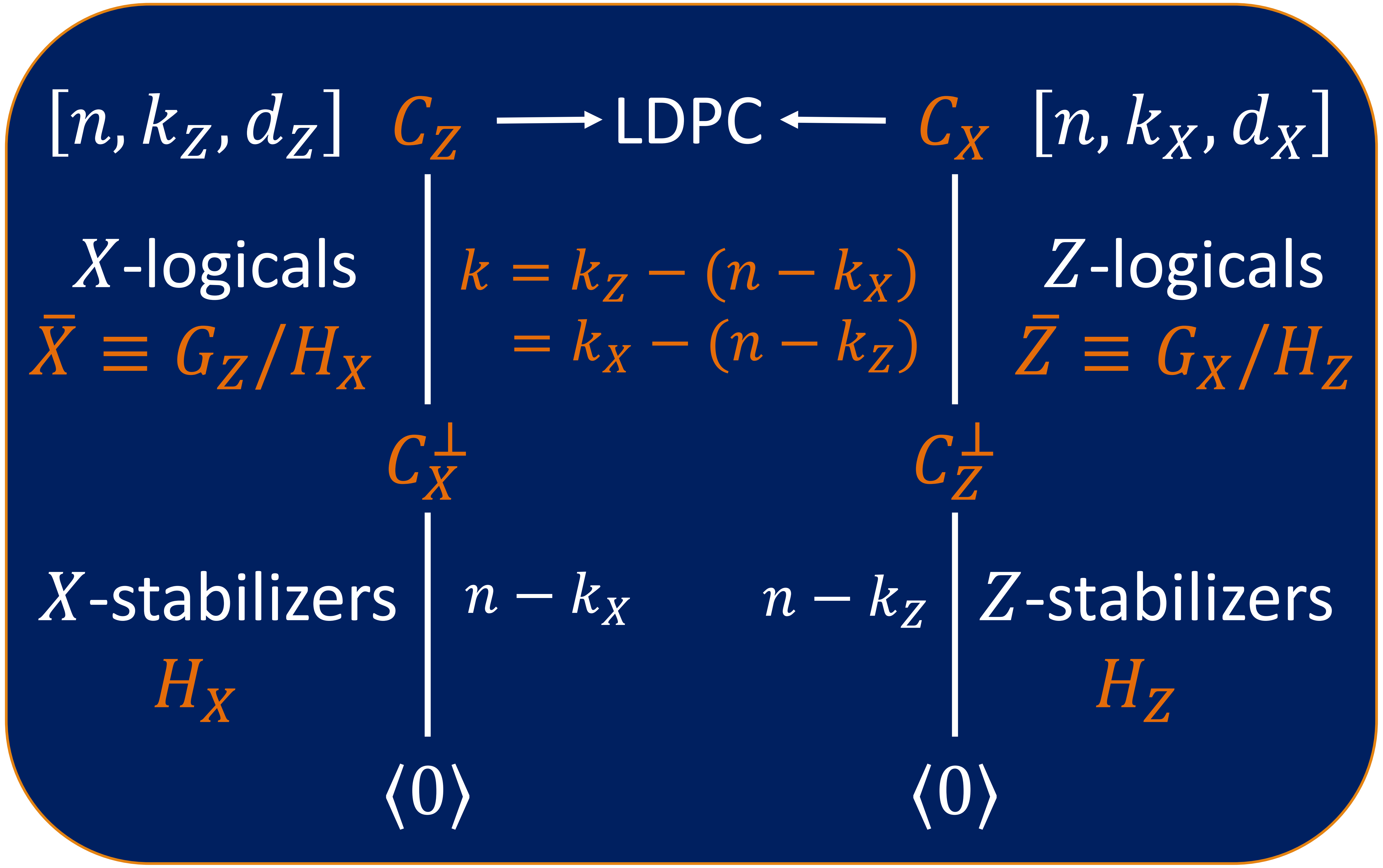
🟥 - vertex checks ($H_X$)

🟦 - plaquette checks ($H_Z$)

| Topological Codes | Optimal QLDPC Codes |
|---|---|
| $[[\, n, 1, \Theta(\sqrt{n}) \,]]$ | $[[\, n, \Theta(n), \Theta(n) \,]]$ |
| High error thresholds | Promising thresholds |
| $\sim$ Linear-time decoder | Linear-time decoder* |
| Logical gates known | **Very little research** |
| Nearest-neighbor | **Long-range interactions** |
| **Not scalable;**<br>**large overhead** | **Scalable with**<br>**constant overhead???** |

QLDPC: Quantum Low-Density Parity-Check

# Calderbank-Shor-Steane (CSS) Codes



$[n, k_Z, d_Z]$ $C_Z$ $\longrightarrow$ LDPC $\longleftarrow$ $C_X$ $[n, k_X, d_X]$

$X$-logicals

$\bar{X} \equiv G_Z / H_X$

$$k = k_Z - (n - k_X)$$
$$= k_X - (n - k_Z)$$

$Z$-logicals

$\bar{Z} \equiv G_X / H_Z$

$C_X^{\perp}$ $C_Z^{\perp}$

$X$-stabilizers

$H_X$

$n - k_X$

$n - k_Z$

$Z$-stabilizers

$H_Z$

$\langle 0 \rangle$ $\langle 0 \rangle$

## $[[n, k, d]]$ CSS Code

$$H_S = \left[\begin{array}{c|c} H_X & 0 \\ \hline 0 & H_Z \end{array}\right] \begin{array}{c} X \\ Z \end{array}$$

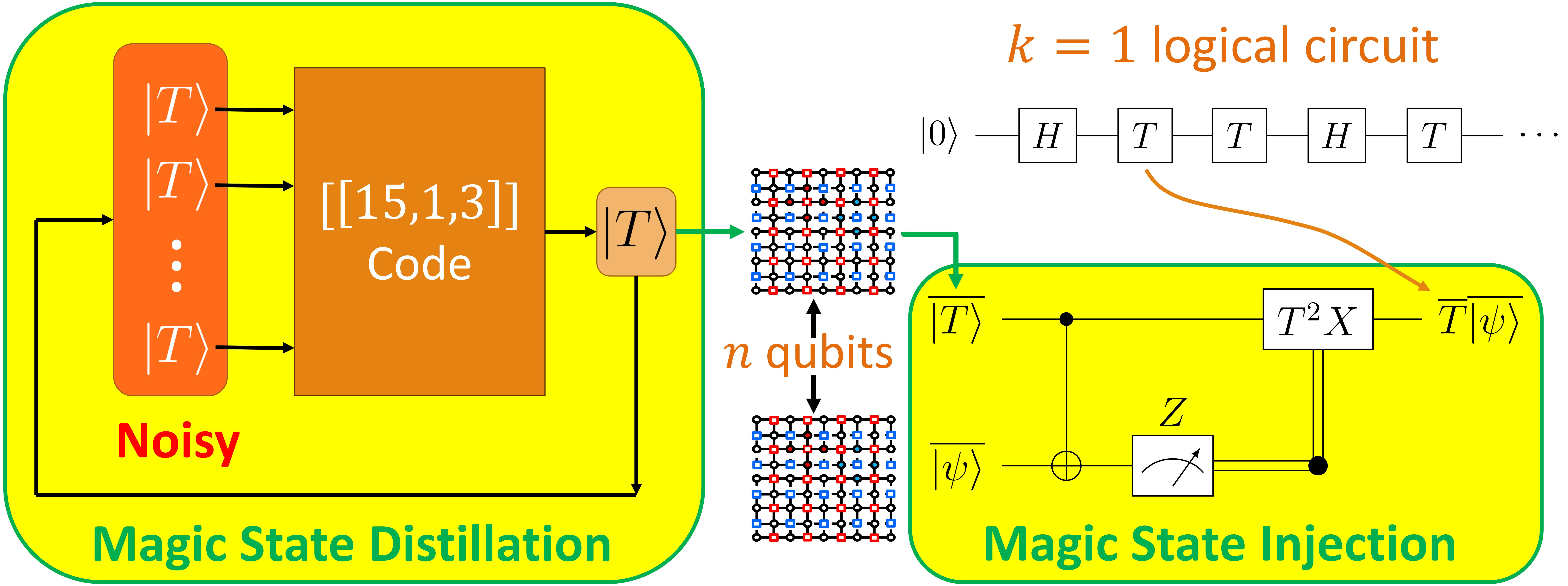
$$H_X H_Z^T = 0$$

$$\boxed{k = n - \mathrm{rank}(H_X) - \mathrm{rank}(H_Z)}$$

$$\boxed{d = \text{minimum weight of } \bar{X}, \bar{Z}}$$

How to implement logical non-Clifford gate fault-tolerantly?

# Magic State Distillation and Injection



$k = 1$ logical circuit

$|0\rangle$ — $H$ — $T$ — $T$ — $H$ — $T$ — $\cdots$

$|T\rangle$
$|T\rangle$
$\vdots$
$|T\rangle$

**Noisy**

[[15,1,3]] Code

$|T\rangle$

$n$ qubits

$\overline{|T\rangle}$

$\overline{|\psi\rangle}$

$Z$

$T^2X$

$\overline{T|\psi\rangle}$

**Magic State Distillation**

**Magic State Injection**

"Triorthogonal" Codes: https://arxiv.org/abs/1209.2426

# Triorthogonal Codes

$$\begin{tikzcd}
C_Z \arrow[d, dash, "\bar{X}"'] & C_X \arrow[d, dash, "\bar{Z}"] \\
C_X^{\perp} \arrow[d, dash, "H_X"'] & C_Z^{\perp} \arrow[d, dash, "H_Z"] \\
\langle 0 \rangle & \langle 0 \rangle
\end{tikzcd}$$

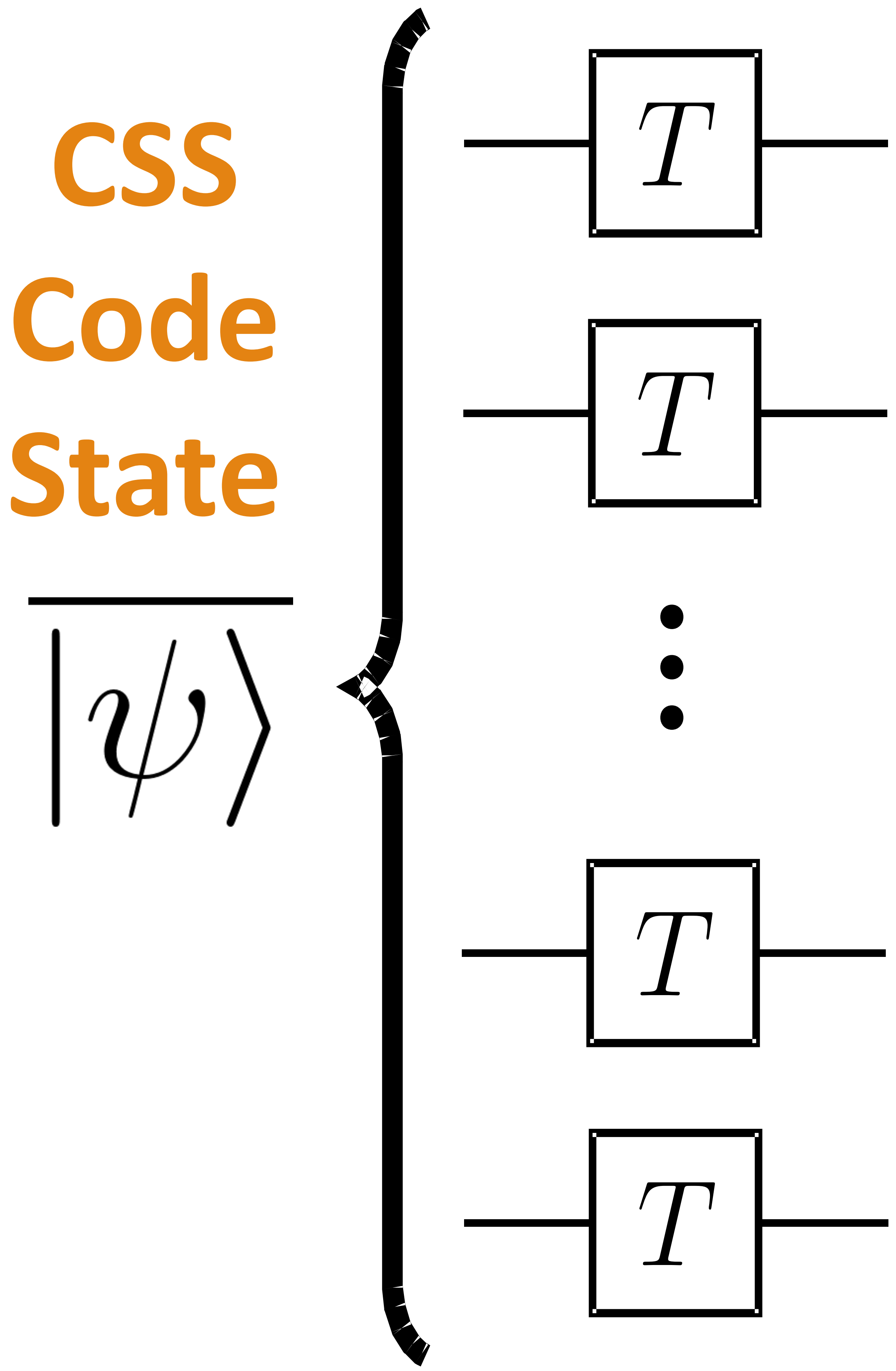
$C_Z$ = Punctured RM(1,4)
$C_X$ = [15,11,3] Hamming
$C_Z^{\perp}$ = Even weight subcode
$C_X^{\perp}$ = [15,4,8] Simplex

## [[15,1,3]] Code: Transversal $T$ induces logical $T^{\dagger}$

$$H_Z = \left[\begin{array}{ccccccccccccccc} 1 & & 1 & & 1 & & 1 & & 1 & & 1 & & 1 & & 1 \\ & 1 & 1 & & & 1 & 1 & & & 1 & 1 & & & 1 & 1 \\ & & & 1 & 1 & 1 & 1 & & & & & 1 & 1 & 1 & 1 \\ & & & & & & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ & & 1 & & & & 1 & & & & 1 & & & & 1 \\ & & & & 1 & & 1 & & & & & & 1 & & 1 \\ & & & & & & & & 1 & & 1 & & 1 & & 1 \\ & & & & & 1 & 1 & & & & & & & 1 & 1 \\ & & & & & & & & & 1 & 1 & & & 1 & 1 \\ & & & & & & & & & & & 1 & 1 & 1 & 1 \end{array}\right] \begin{array}{ll} x_1 & \\ x_2 & \Big\} H_X \\ x_3 & \\ x_4 & \\ x_1x_2 & \\ x_1x_3 & \\ x_1x_4 & \\ x_2x_3 & \\ x_2x_4 & \\ x_3x_4 & \end{array}$$

"Triorthogonal" Codes: https://arxiv.org/abs/1209.2426

# Transversal $T$: Naturally Fault-Tolerant



When does this preserve the CSS code space?

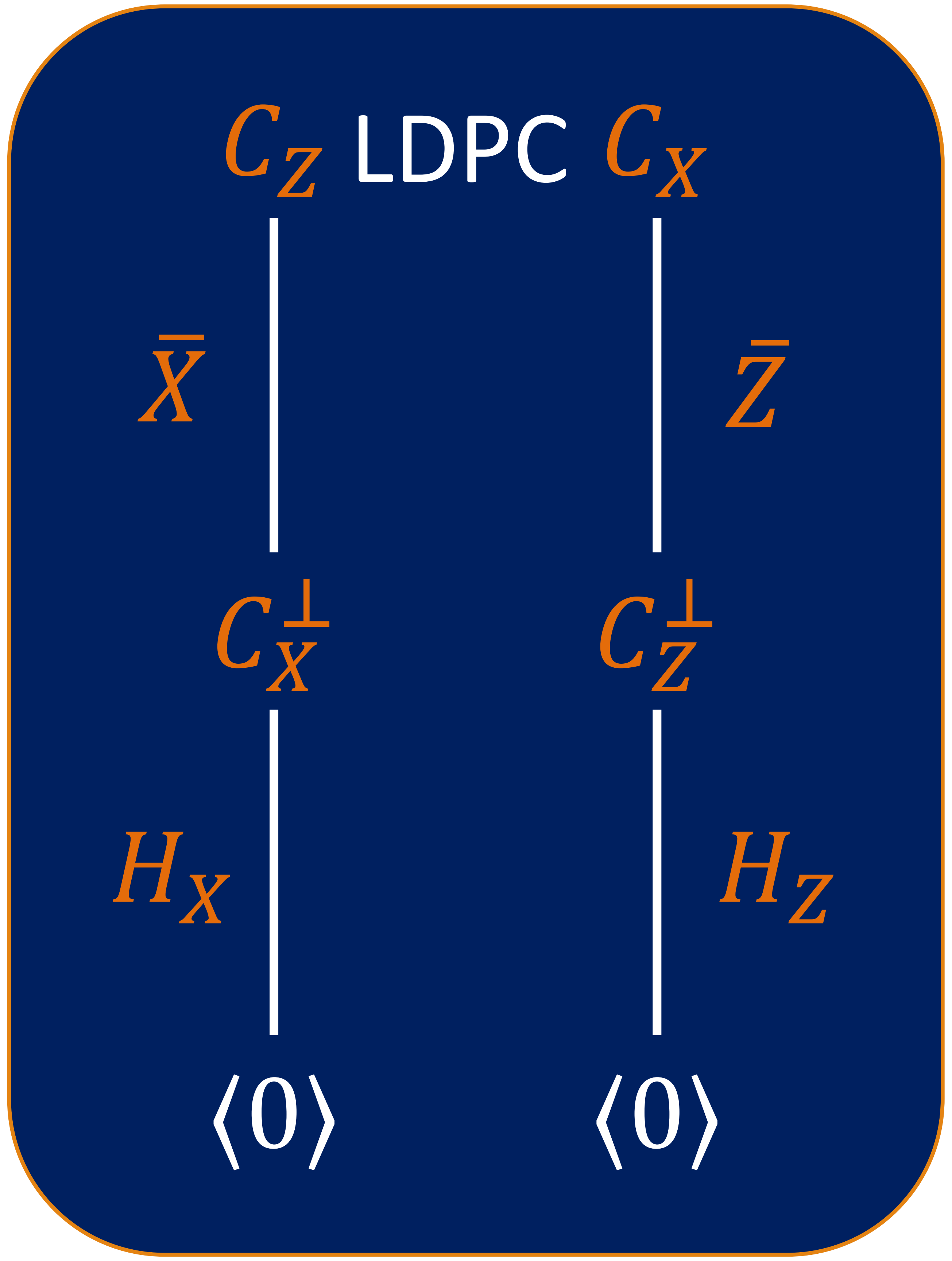
Consider the projector $\Pi_S$ to the code space. For transversal $T$ to fix the code space, we need

$$T^{\otimes n}\Pi_S\left(T^{\otimes n}\right)^{\dagger} = \Pi_S.$$

Solving this equality leads to necessary and sufficient conditions that the code must meet.

For more details, see https://arxiv.org/abs/1910.09333 and https://arxiv.org/abs/2001.04887

# Transversal $T$: Classical Coding Problem



$$\begin{tikzcd}
C_Z \arrow[d, dash, "\bar{X}"'] & \text{LDPC} & C_X \arrow[d, dash, "\bar{Z}"] \\
C_X^{\perp} \arrow[d, dash, "H_X"'] & & C_Z^{\perp} \arrow[d, dash, "H_Z"] \\
\langle 0 \rangle & & \langle 0 \rangle
\end{tikzcd}$$

## “CSS-T” Problem for Quantum Codes

Construct pair $(C_Z, C_X)$ of classical codes s.t.:

1. All codewords of $C_X^{\perp}$ have even Hamming weight
2. For each $x \in C_X^{\perp}$, the code $C_Z^{\perp}$ contains a self-dual code $Z_x$ supported only on $x \in C_X^{\perp}$

($Z_x$ is essentially a $[w_H(x), \frac{w_H(x)}{2}]$ self-dual code)

For more details, see https://arxiv.org/abs/1910.09333 and https://arxiv.org/abs/2001.04887

# CSS-T Example 1

$$\begin{tikzcd}
C_Z \arrow[d, dash, "\bar{X}"'] & C_X \arrow[d, dash, "\bar{Z}"] \\
C_X^{\perp} \arrow[d, dash, "H_X"'] & C_Z^{\perp} \arrow[d, dash, "H_Z"] \\
\langle 0 \rangle & \langle 0 \rangle
\end{tikzcd}$$

Construct pair $(C_Z, C_X)$ of classical (LDPC) codes s.t.:

1. All codewords of $C_X^{\perp}$ have even Hamming weight
2. For each $x \in C_X^{\perp}$, the code $C_Z^{\perp}$ contains a self-dual code $Z_x$ supported only on $x \in C_X^{\perp}$

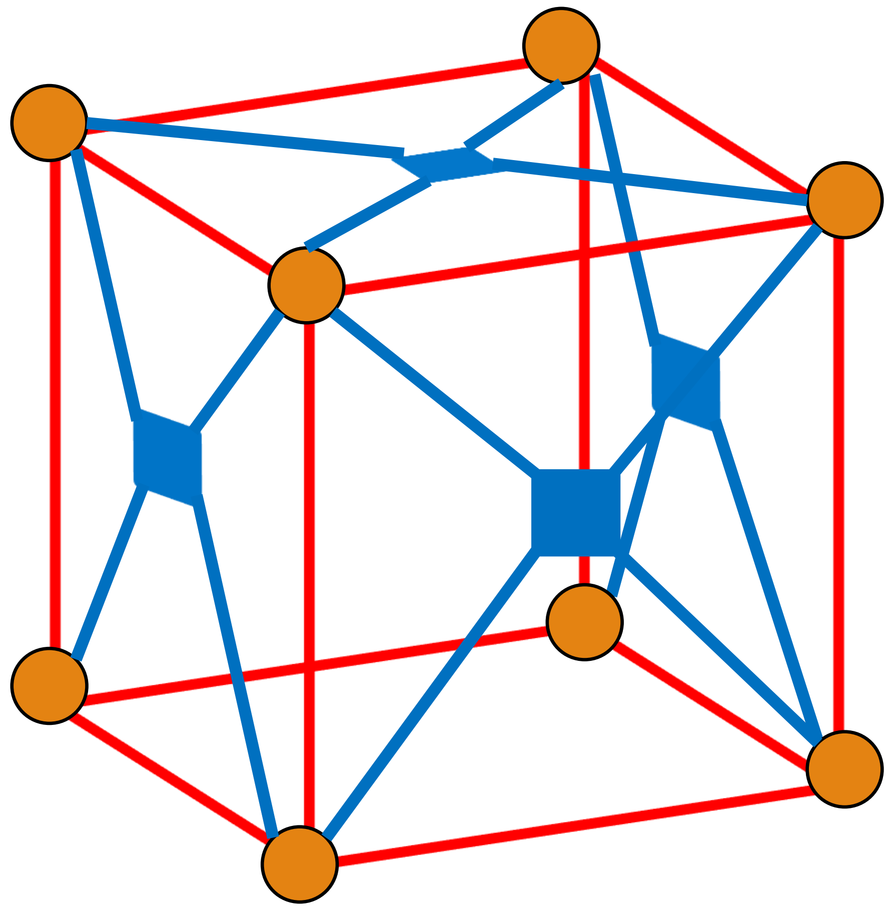
## [[8,3,2]] Code: Transversal $T$ induces logical $CCZ$

$$H_X = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

8-bit Repetition

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

Reed-Muller RM(1,3)
a.k.a.
Extended Hamming



# CSS-T Example 2 (Triorthogonal Code)

$$\begin{tikzcd} C_Z \arrow[d, dash, "\bar{X}"'] & C_X \arrow[d, dash, "\bar{Z}"] \\ C_X^{\perp} \arrow[d, dash, "H_X"'] & C_Z^{\perp} \arrow[d, dash, "H_Z"] \\ \langle 0 \rangle & \langle 0 \rangle \end{tikzcd}$$

## [[15,1,3]] Code: Transversal $T$ induces logical $T^{\dagger}$

$$H_Z = \left[\begin{array}{ccccccccccccccc}
1 & & 1 & & 1 & & 1 & & 1 & & 1 & & 1 & & 1 \\
 & 1 & 1 & & & 1 & 1 & & & 1 & 1 & & & 1 & 1 \\
 & & & 1 & 1 & 1 & 1 & & & & & 1 & 1 & 1 & 1 \\
 & & & & & & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\hline
 & & 1 & & & & 1 & & & & 1 & & & & 1 \\
 & & & & 1 & & 1 & & & & & & 1 & & 1 \\
 & & & & & & & & 1 & & 1 & & 1 & & 1 \\
 & & & & & 1 & 1 & & & & & & & 1 & 1 \\
 & & & & & & & & & 1 & 1 & & & 1 & 1 \\
 & & & & & & & & & & & 1 & 1 & 1 & 1
\end{array}\right]
\begin{array}{l}
\left.\begin{array}{l} x_1 \\ x_2 \\ x_3 \\ x_4 \end{array}\right\} H_X \\
x_1 x_2 \\ x_1 x_3 \\ x_1 x_4 \\ x_2 x_3 \\ x_2 x_4 \\ x_3 x_4
\end{array}$$

$C_Z$ = Punctured RM(1,4)

$C_X$ = [15,11,3] Hamming

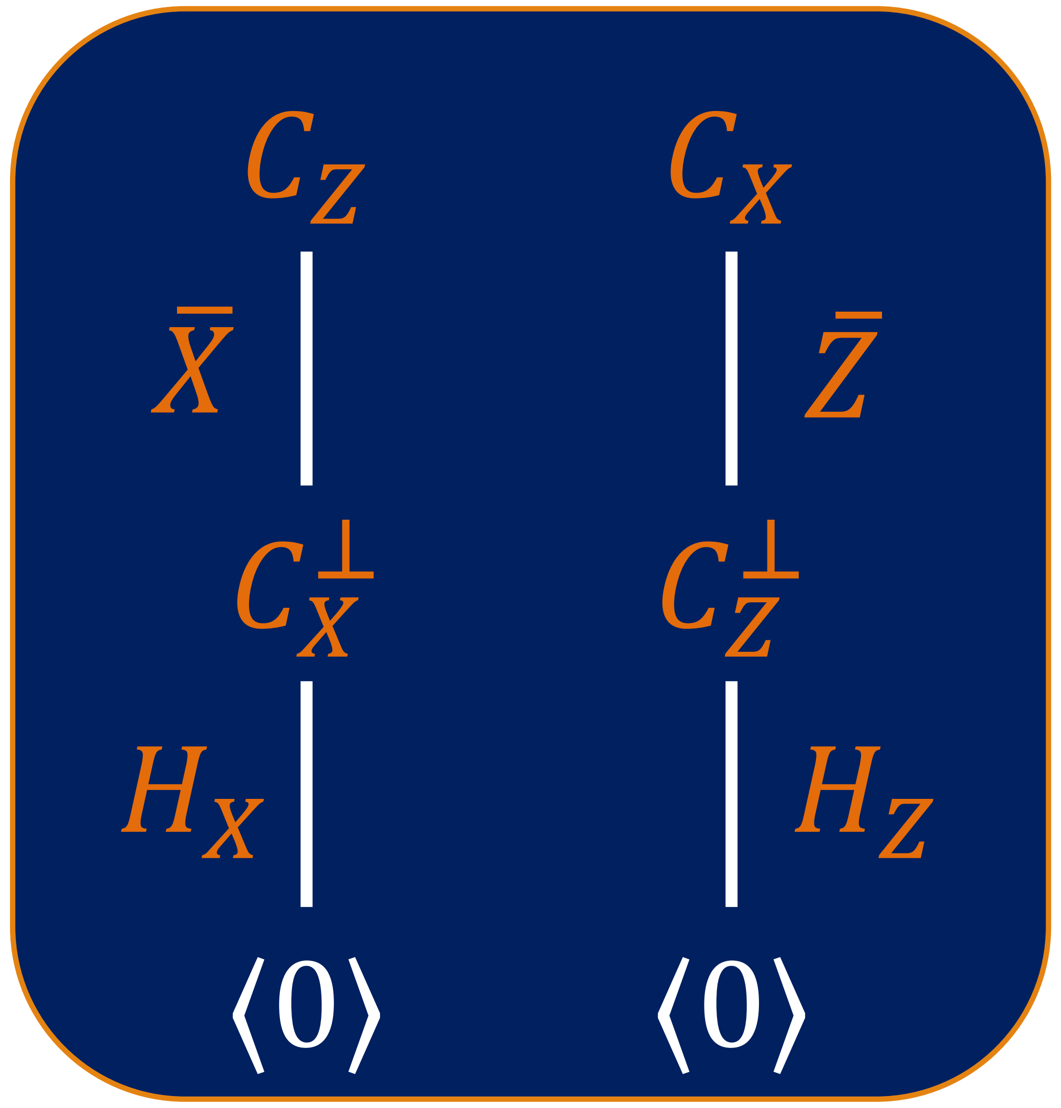
$C_Z^{\perp}$ = Even weight subcode

$C_X^{\perp}$ = [15,4,8] Simplex

"Triorthogonal" Codes: https://arxiv.org/abs/1209.2426

# CSS-T Example 3

## $[[16,3,2]]$ Code: Transversal $T$ induces logical $CCZ$



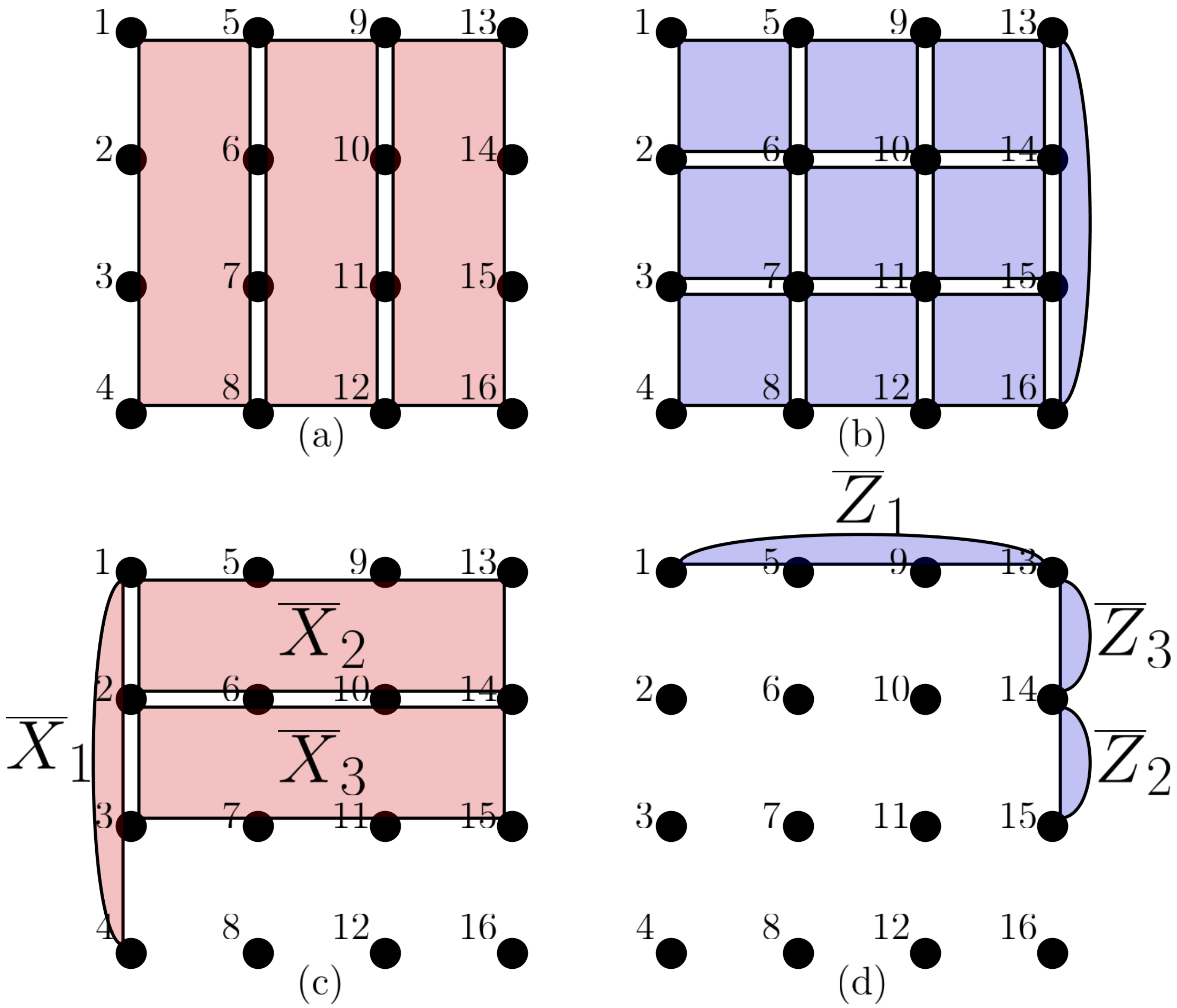
$C_Z$ $C_X$

$\bar{X}$ $\bar{Z}$

$C_X^{\perp}$ $C_Z^{\perp}$

$H_X$ $H_Z$

$\langle 0\rangle$ $\langle 0\rangle$



$H_X = \{1, x_1, x_2\}$

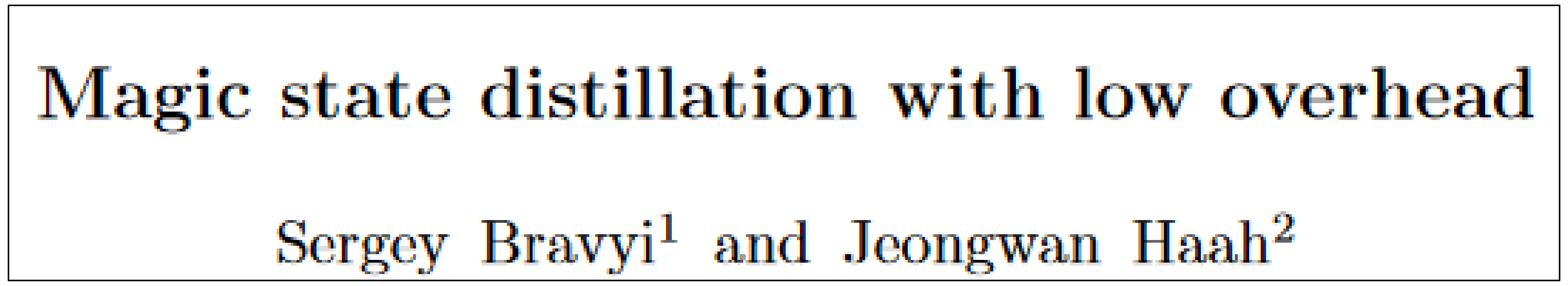
$G_Z = H_X \cup \{x_3, x_4, x_1x_2\}$

$H_Z = G_Z \cup \{x_1x_3, x_1x_4, x_2x_3, x_2x_4\}$

$G_X = H_Z \cup \{x_3x_4, x_1x_2x_3, x_1x_2x_4\}$

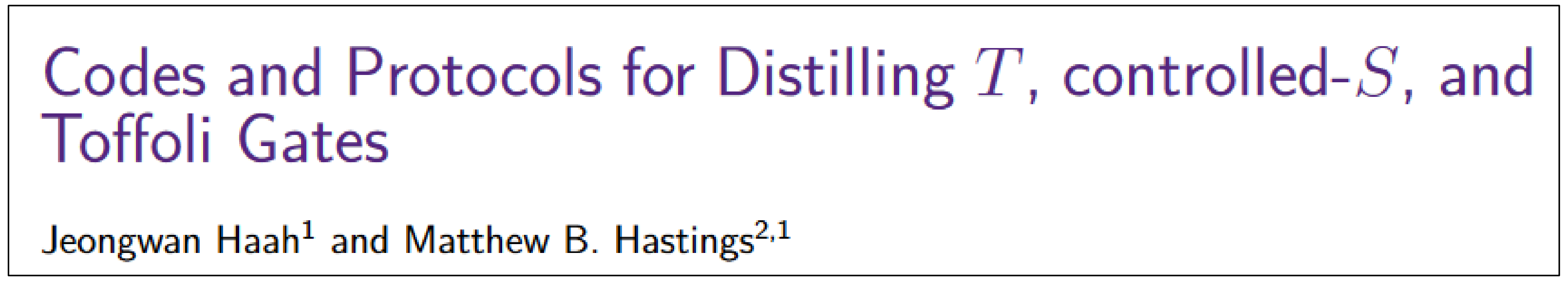
Decreasing Monomial Codes: https://arxiv.org/abs/1601.06215

# Related Work (Partial List)



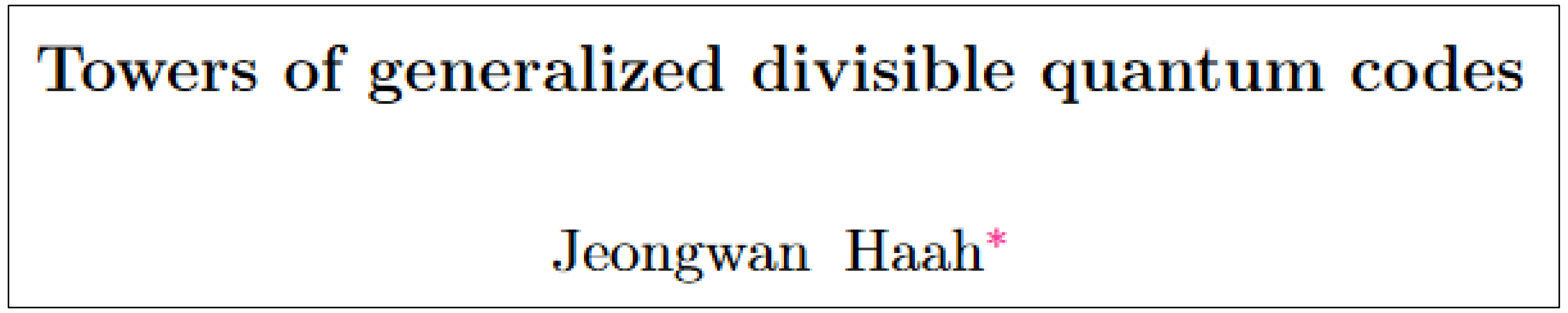
### Magic state distillation with low overhead

Sergey Bravyi[1] and Jeongwan Haah[2]



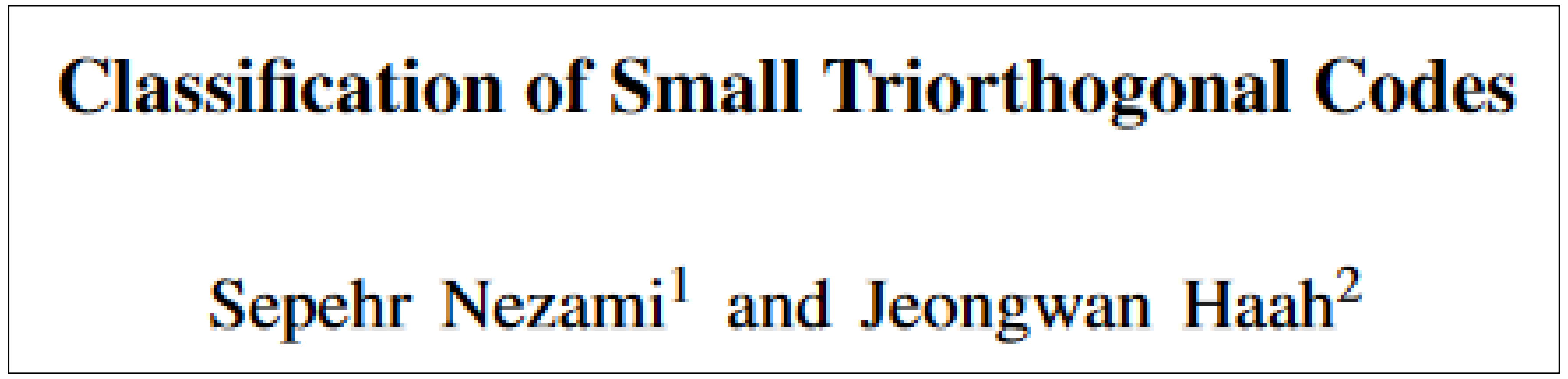
### Codes and Protocols for Distilling $T$, controlled-$S$, and Toffoli Gates

Jeongwan Haah[1] and Matthew B. Hastings[2,1]



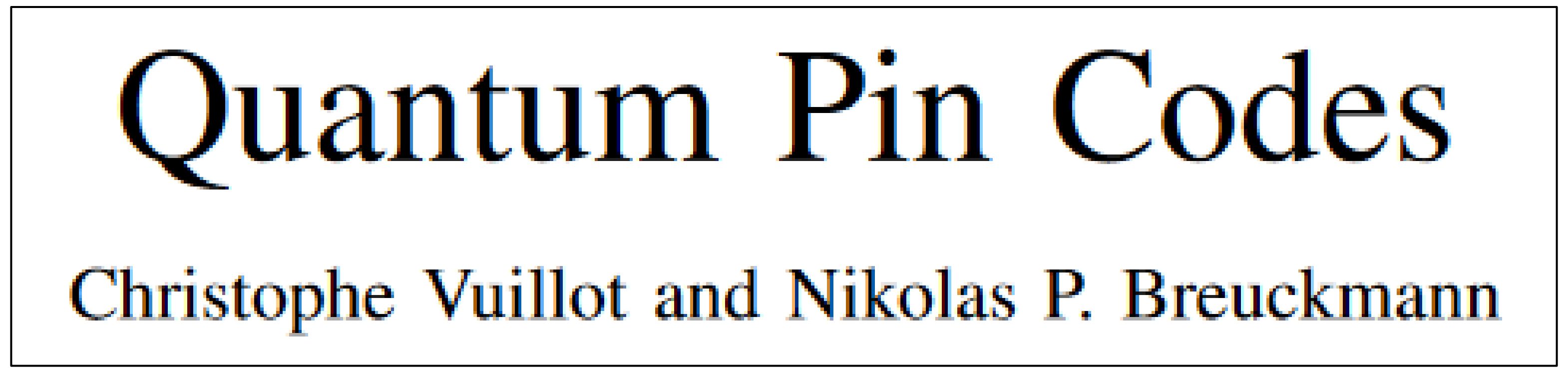
### Towers of generalized divisible quantum codes

Jeongwan Haah*



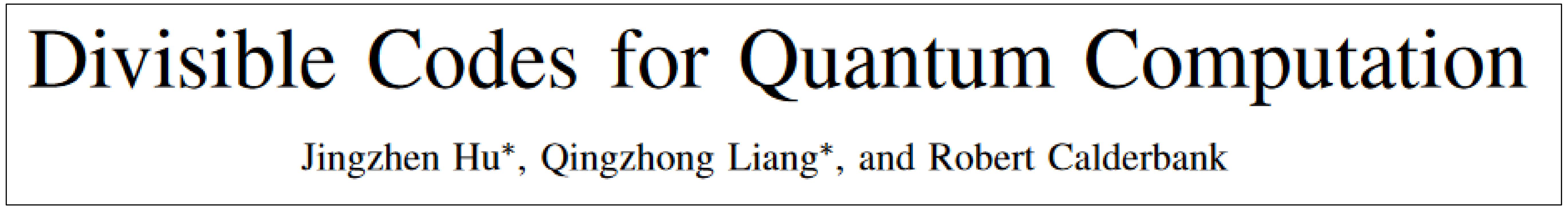
### Classification of Small Triorthogonal Codes

Sepehr Nezami[1] and Jeongwan Haah[2]



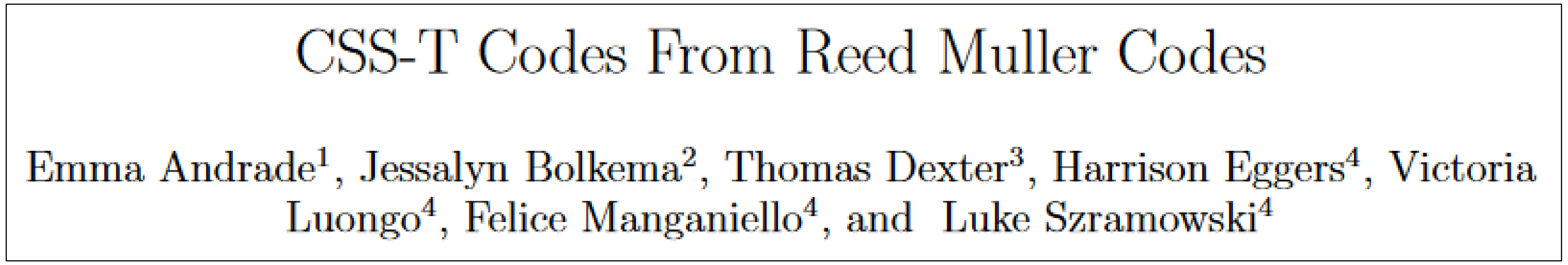
### Quantum Pin Codes

Christophe Vuillot and Nikolas P. Breuckmann



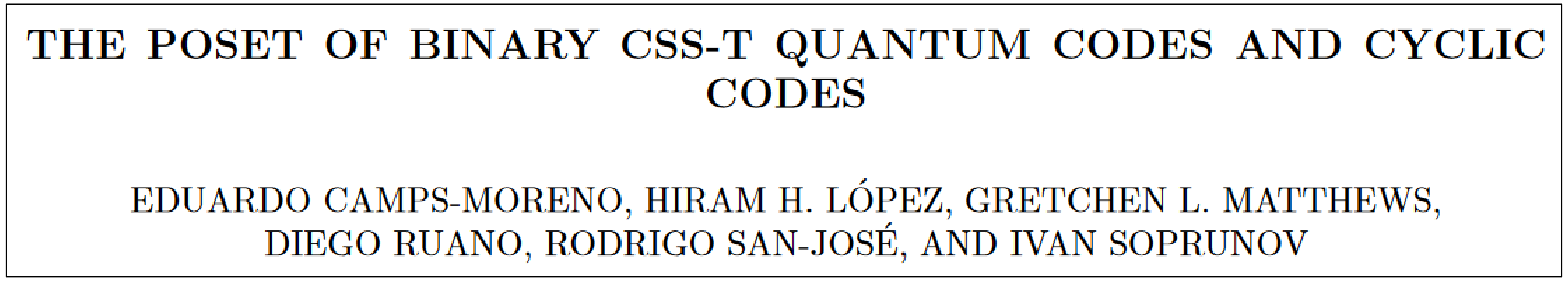
### Divisible Codes for Quantum Computation

Jingzhen Hu*, Qingzhong Liang*, and Robert Calderbank



### CSS-T Codes From Reed Muller Codes

Emma Andrade[1], Jessalyn Bolkema[2], Thomas Dexter[3], Harrison Eggers[4], Victoria Luongo[4], Felice Manganiello[4], and Luke Szramowski[4]

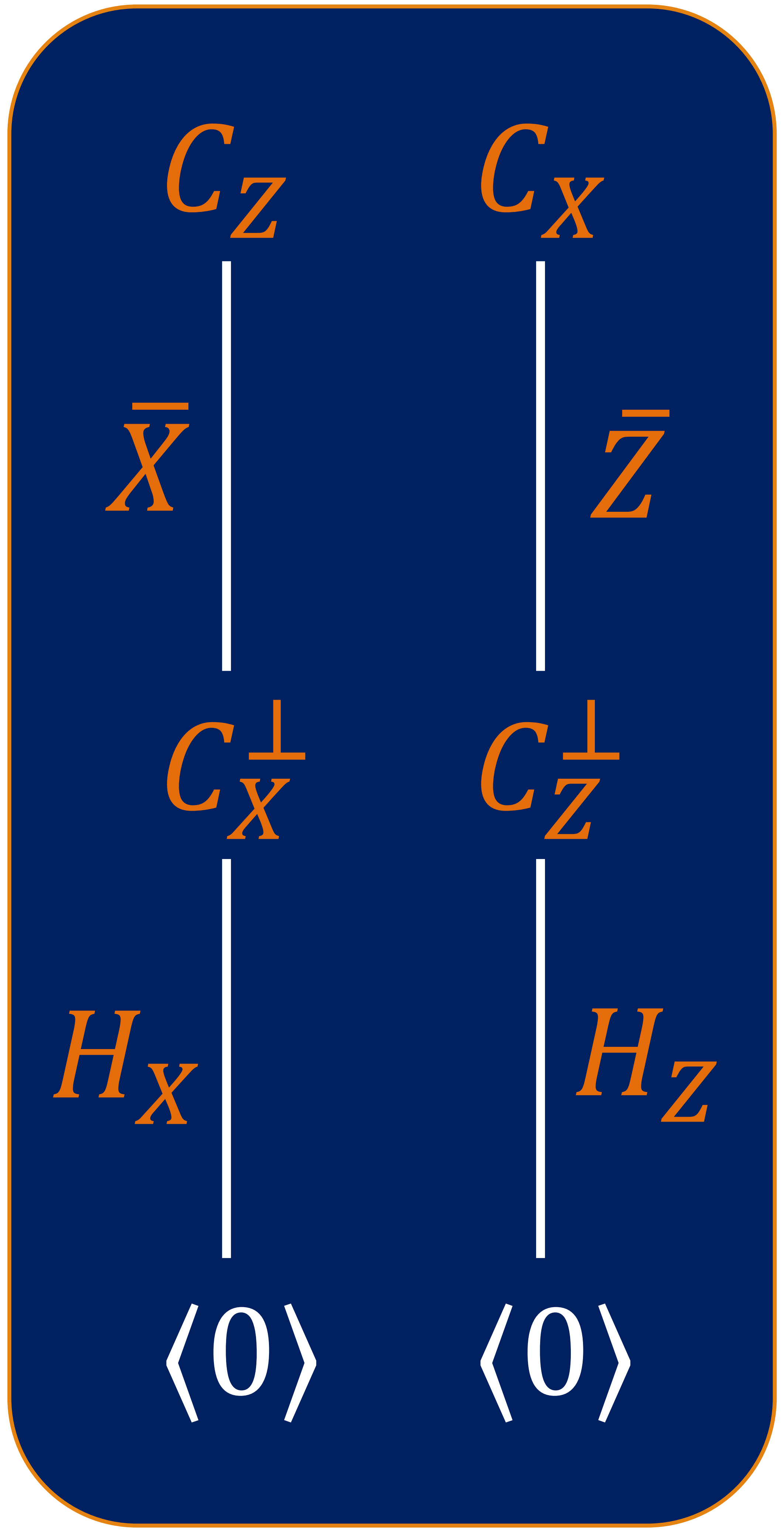


### THE POSET OF BINARY CSS-T QUANTUM CODES AND CYCLIC CODES

EDUARDO CAMPS-MORENO, HIRAM H. LÓPEZ, GRETCHEN L. MATTHEWS, DIEGO RUANO, RODRIGO SAN-JOSÉ, AND IVAN SOPRUNOV

Classical Coding Theorists!

# CSS-T: Search for Good Codes



Construct pair $(C_Z, C_X)$ of classical (LDPC) codes s.t.:

1. All codewords of $C_X^{\perp}$ have even Hamming weight
2. For each $x \in C_X^{\perp}$, the code $C_Z^{\perp}$ contains a self-dual code $Z_x$ supported only on $x \in C_X^{\perp}$

$[[n, k, d]]$ CSS-T Codes:

1. Current solution: polynomial evaluation codes
2. Problem: high-weight checks, poor parameters

3. Towards optimality: need $k = \Theta(n), d = \Theta(n)$
4. Practicality: need both $C_Z$ and $C_X$ to be LDPC

# 1st Quantum Information Knowledge (QuIK) Workshop

- Date: July 7, 2024 @ ISIT (full day)
- Venue: Athens, Greece
- Theme: Quantum Error Correction
- Goal: Interactions between QEC community and coding theorists
- Components: Tutorial, Invited Talks, Poster Session, Panel Discussion
- Submit posters by 3/17 !!

Details: https://isit-quik24.com

# Thank you!

https://arxiv.org/abs/1910.09333

https://arxiv.org/abs/2001.04887

Center for
Quantum Networks
*NSF Engineering Research Center*

narayananr@arizona.edu

https://ece.arizona.edu

https://cqn-erc.org

https://sqmscenter.fnal.gov/

SQMS
SUPERCONDUCTING QUANTUM
MATERIALS & SYSTEMS CENTER