Center for
Quantum Networks
NSF Engineering Research Center

https://cqn-erc.org/

Error Correction for Quantum Networks

Narayanan Rengaswamy
Assistant Professor
University of Arizona
narayananr@arizona.edu

Michele Pacenti
Ph.D. Candidate
University of Arizona
mpacenti@arizona.edu

Sijie Cheng
Ph.D. Student
University of Arizona
sijiecheng@arizona.edu

2025 CQN Winter School on Quantum Networks

Funded by National Science Foundation Grant #1941583

Agenda

Part 1 Classical Error Correction

By Michele Pacenti

Part 2 Quantum Error Correction

By Narayanan Rengaswamy

Part 3 Application to Quantum Networks

By Sijie Cheng

Agenda

Part 1 Classical Error Correction

By Michele Pacenti

Part 2 Quantum Error Correction

By Narayanan Rengaswamy

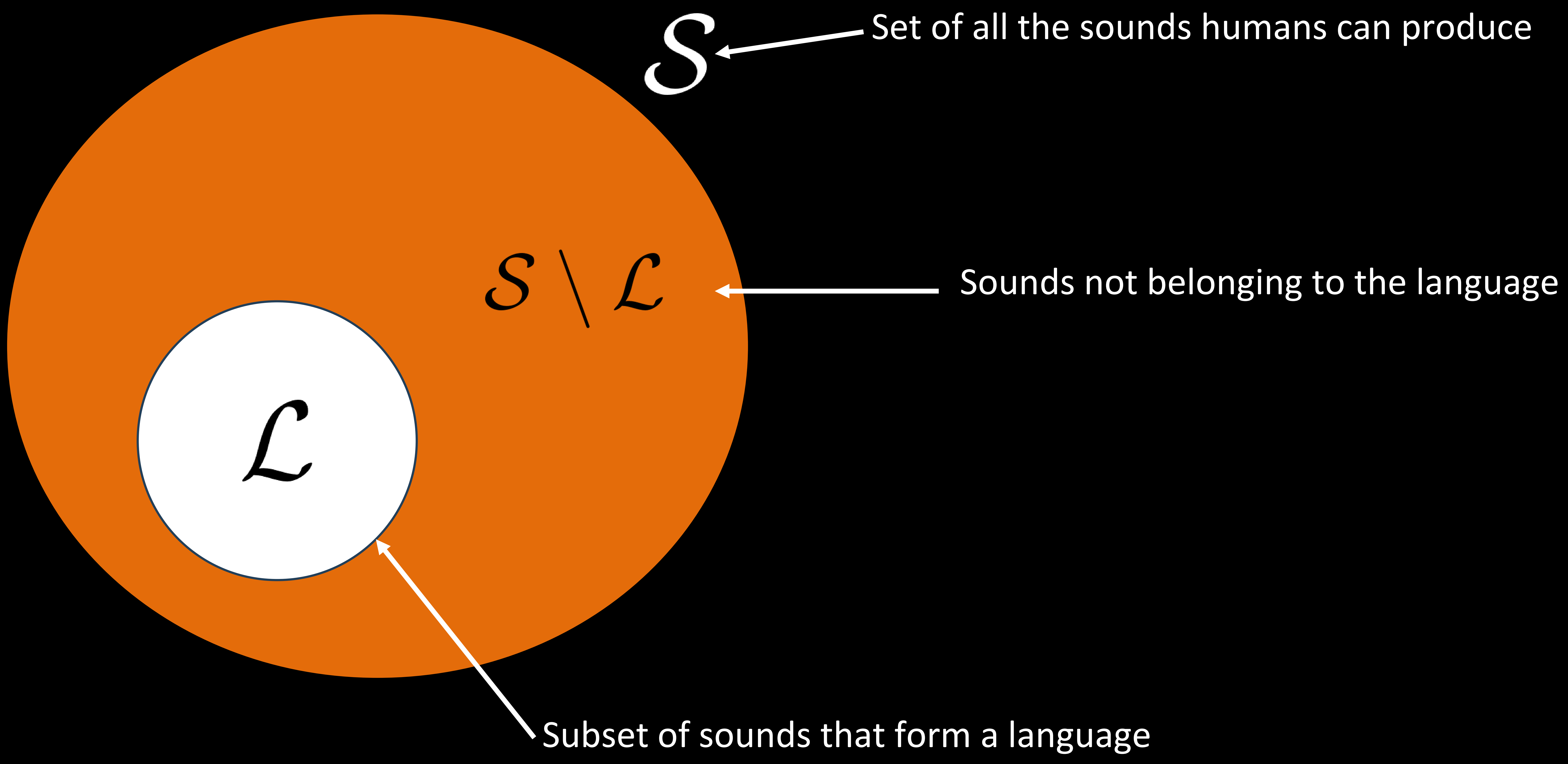
Part 3 Application to Quantum Networks

By Sijie Cheng

T#ping is no substit$te for th?nking.

Typing is no substitute for thinking.

Human languages are error correcting codes



Maximum likelihood decoding

T#ping	→	Typing
is	→	is
no	→	no
substit$te	→	substitute
for	→	for
th?nking	→	thinking

Binary – the language of machines

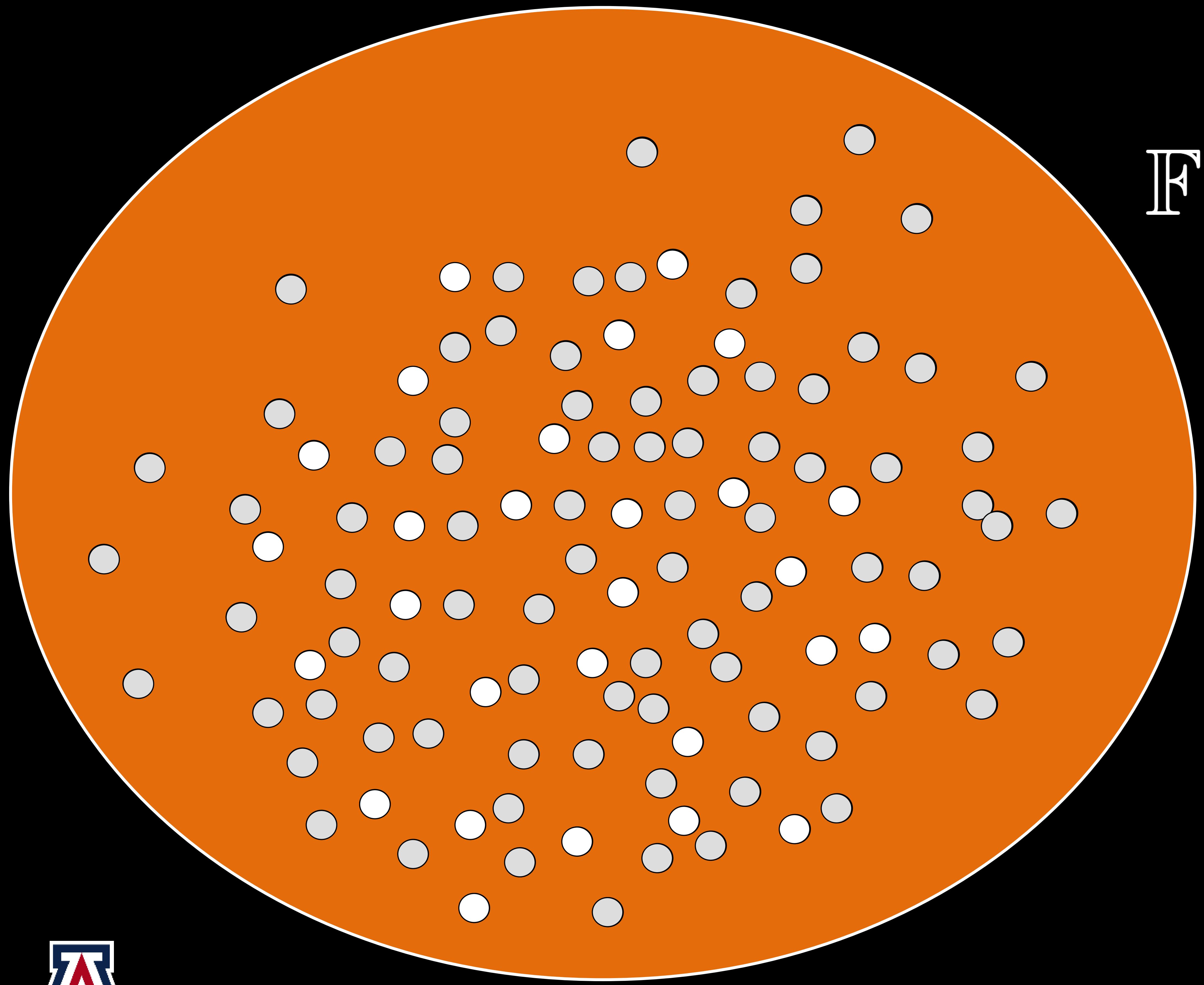
- Binary alphabet: operations over $\mathbb{F}_2^n$

$+$	0	1
0	0	1
1	1	0

$\times$	0	1
0	0	0
1	0	1

- Each word is independent from the others (block codes)

Error correcting codes

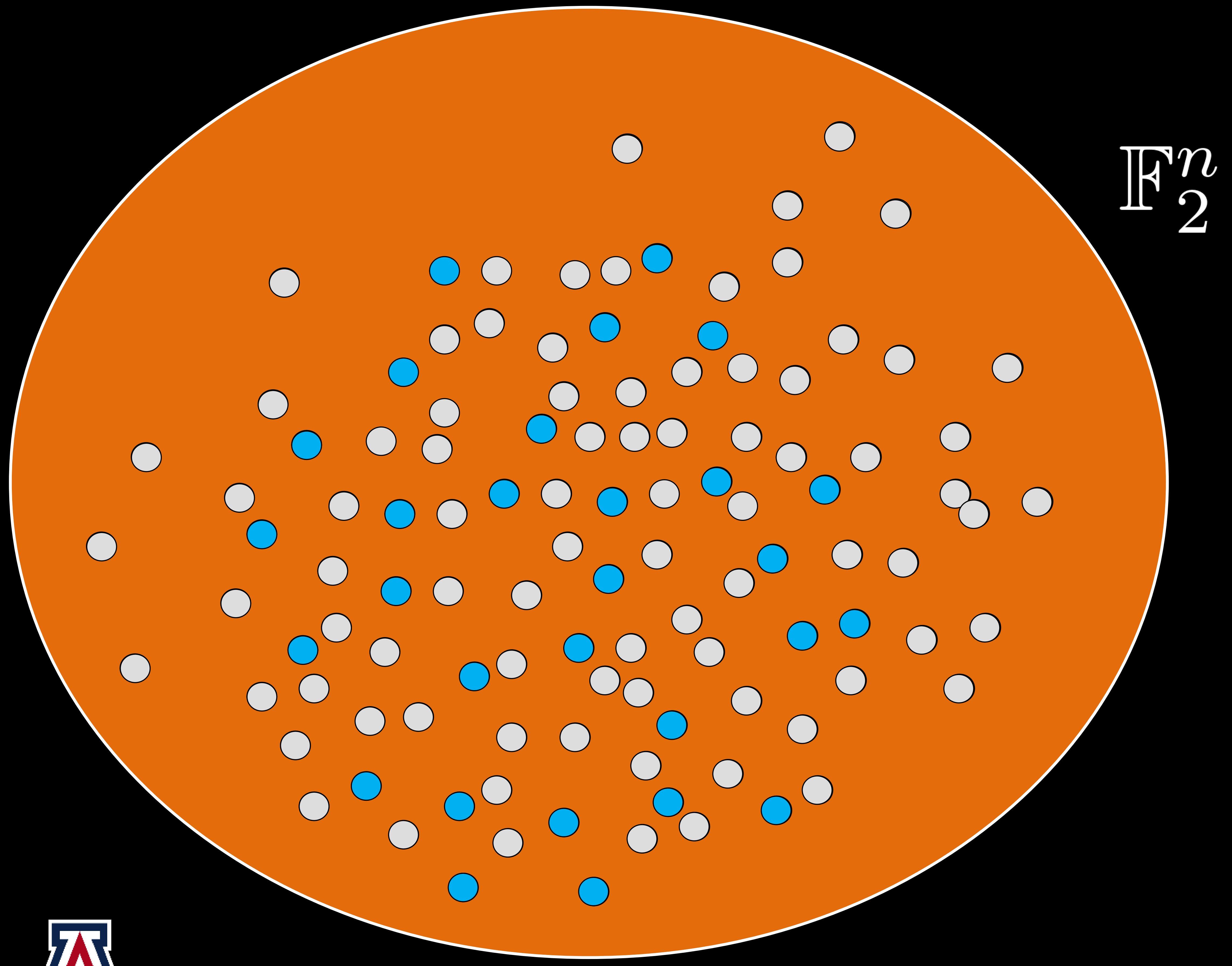


$\mathbb{F}_2^n$ ← Set (field) of all the binary vectors of length n

Size of the space

$$|\mathbb{F}_2^n| = 2^n$$

Error correcting codes



Set (field) of all the binary vectors of length n

- 2^k vectors $\in C$
- $(2^n - 2^k)$ vectors $\notin C$

$k \leq n$

The [3,1,3] repetition code

$$[n, k, d]$$

Codelength — Code dimension — Minimum (Hamming) distance

$$R = \frac{k}{n}$$

Rate

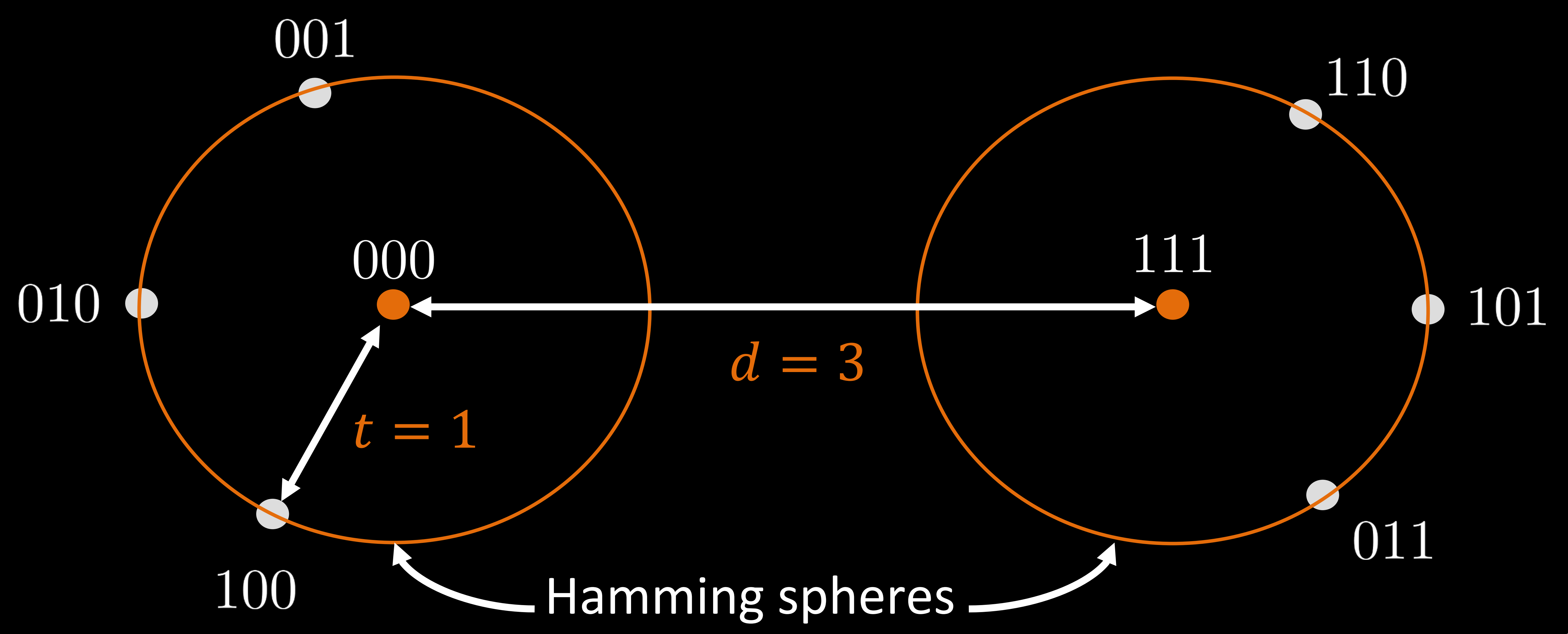
Hamming distance = the number of positions in which two binary vectors differ

$$0 \rightarrow 000$$
$$1 \rightarrow 111$$

Codewords

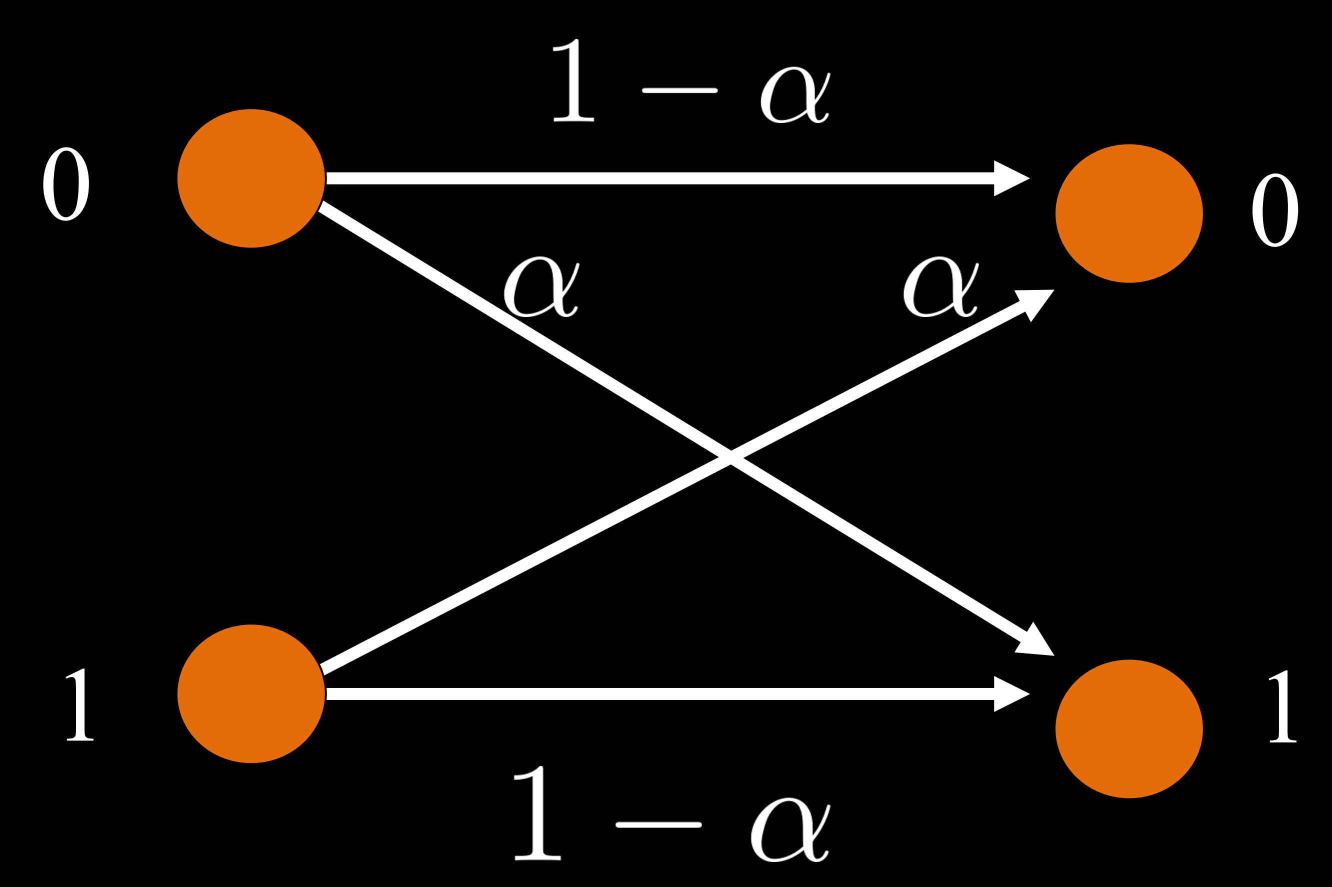
$$t = \left\lfloor \frac{d-1}{2} \right\rfloor$$

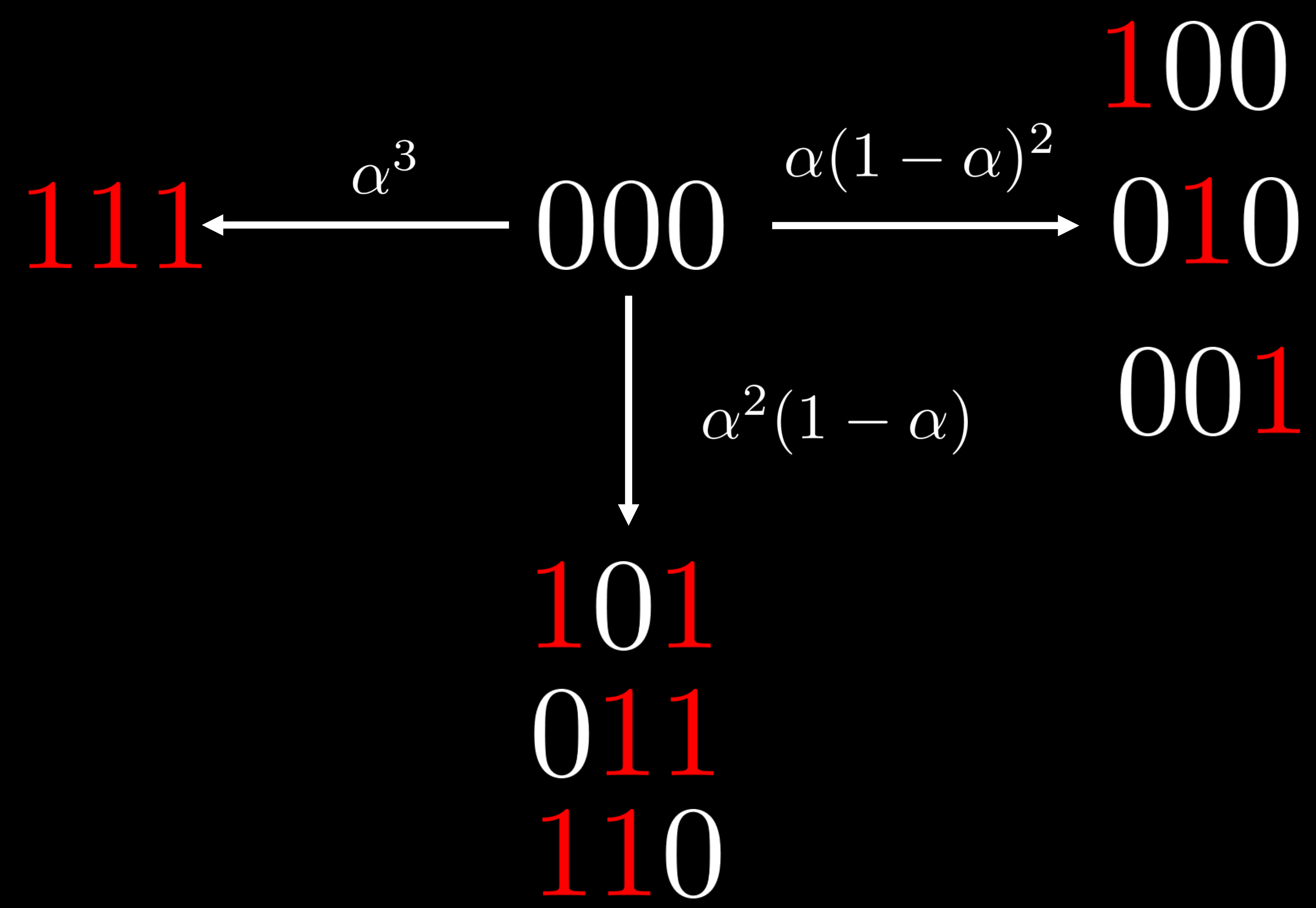
Correctable errors



Binary Symmetric Channel (BSC)

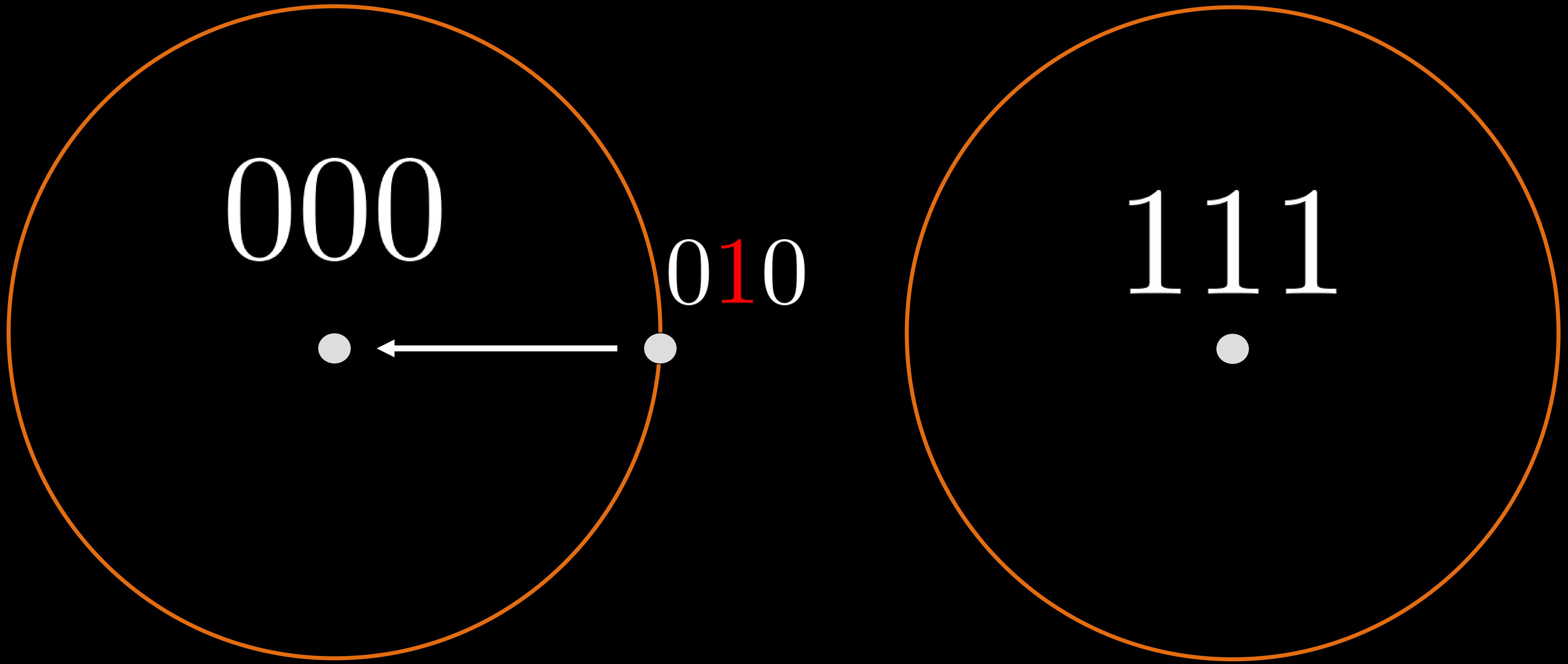
- Each bit is flipped with probability α





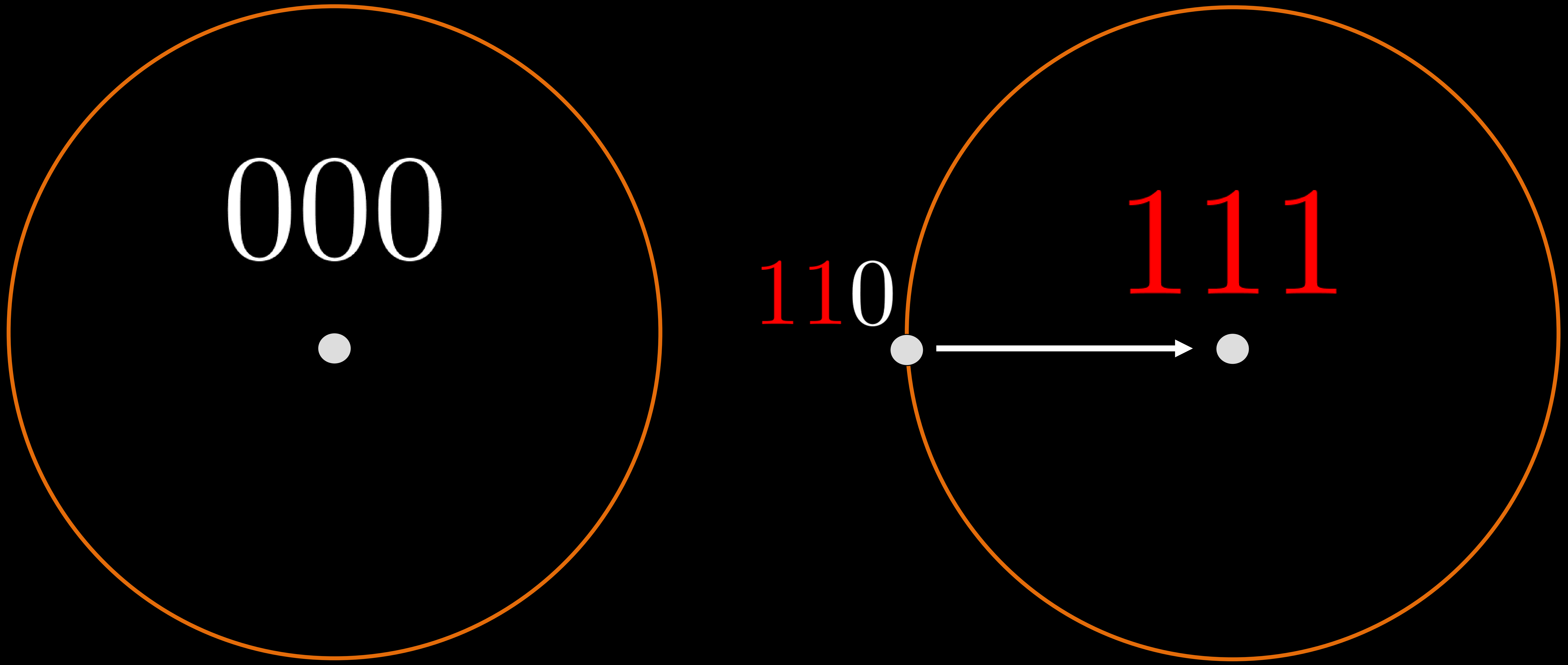
Maximum likelihood decoding of the repetition code

$$\underset{000}{\mathbf{x}} \xrightarrow{\text{BSC}} \underset{010}{\mathbf{y}} \xrightarrow{\text{Decoder}} \underset{000}{\hat{\mathbf{x}}}$$

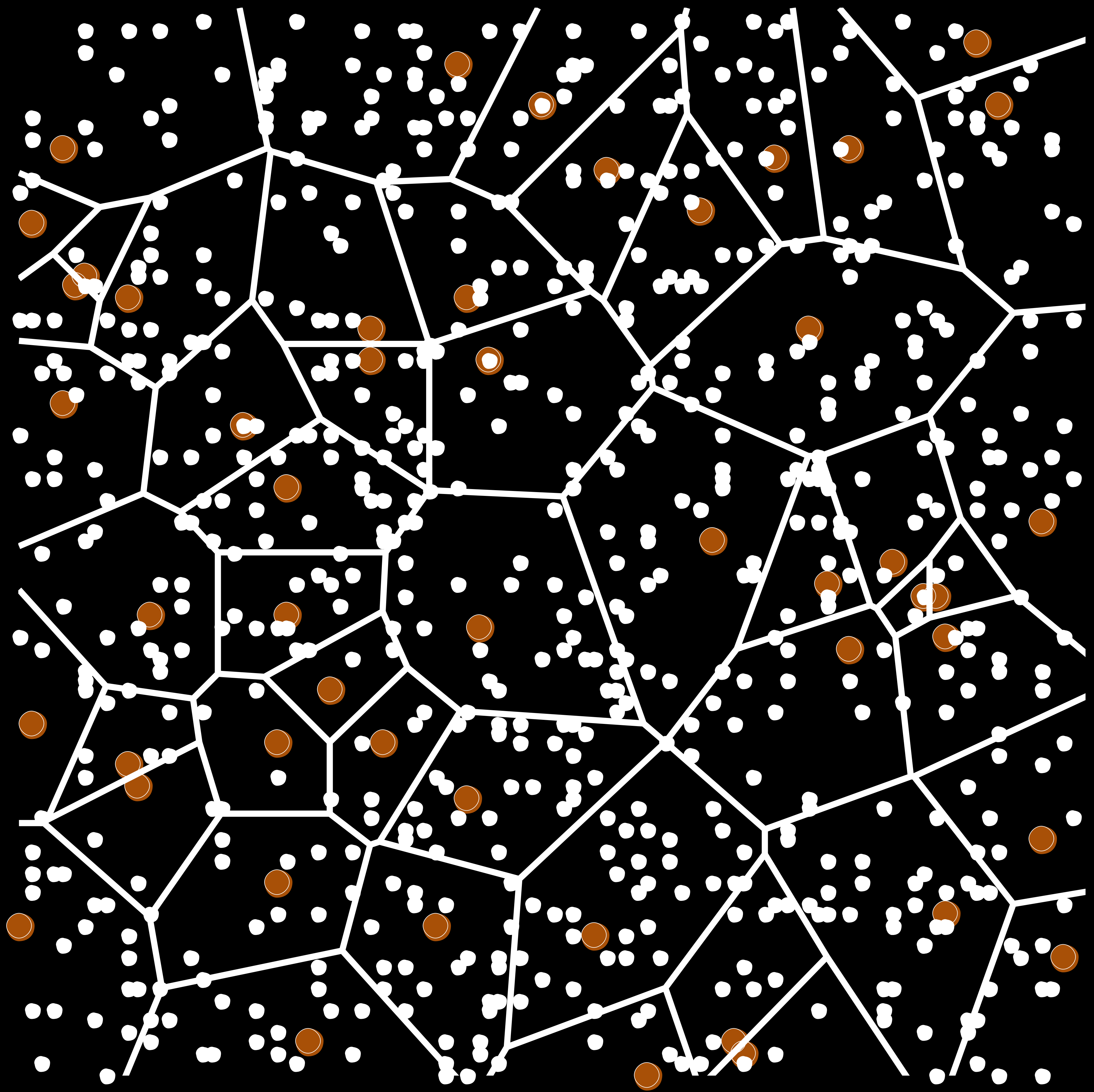


Decoding failure

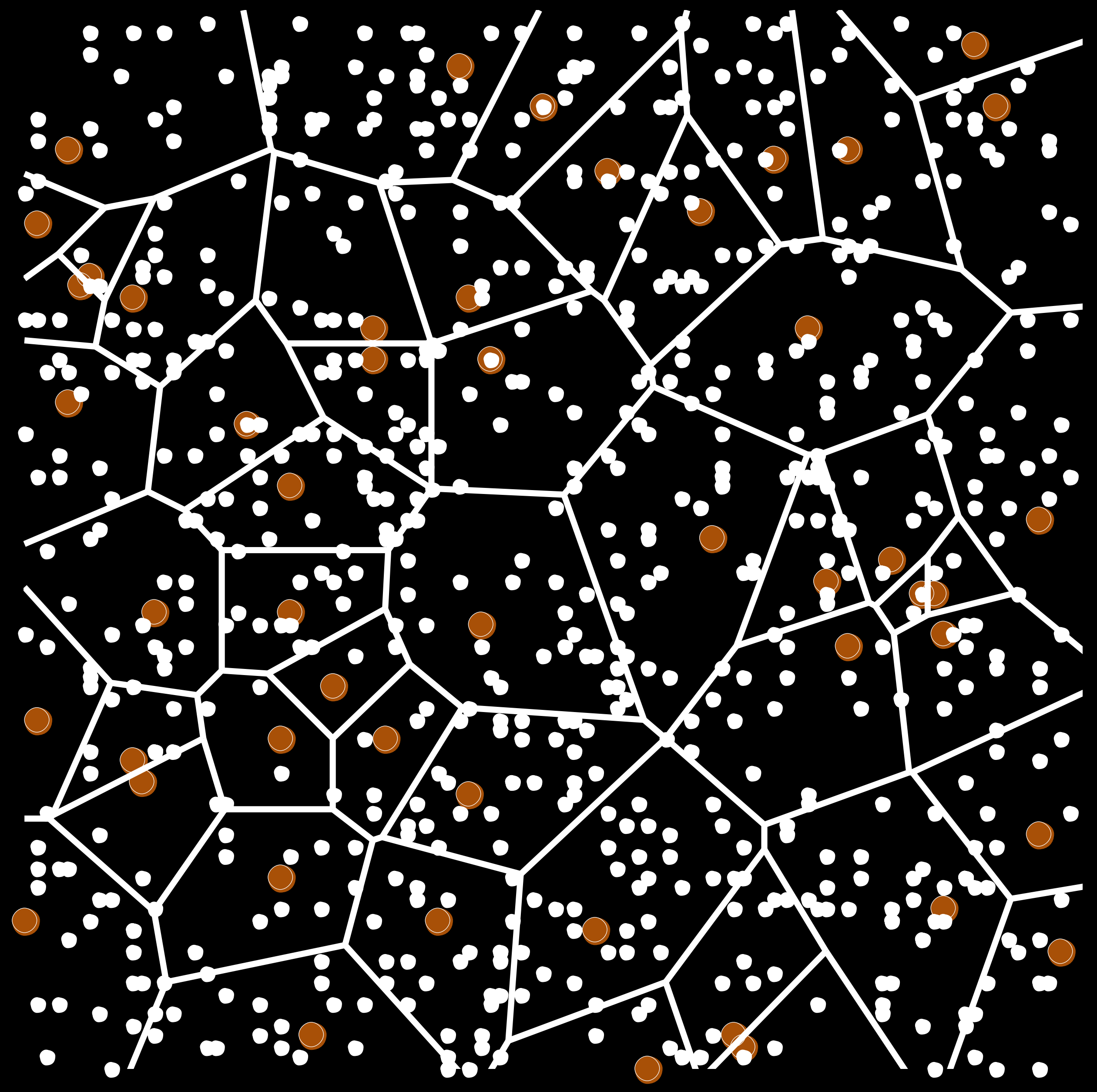
$$\underset{\mathbf{x}}{000} \xrightarrow{\text{BSC}} \underset{\mathbf{y}}{110} \xrightarrow{\text{Decoder}} \underset{\hat{\mathbf{x}}}{111}$$



Maximum likelihood decoding



Maximum likelihood decoding



$d = d(x, y)$: the Hamming distance between x and y

For the BSC channel: $P(y|x) = \alpha^d (1-\alpha)^{n-d}$

ML decoding rule:

$$\hat{x} = \underset{x \in C}{\operatorname{argmax}}\ P(x|y) = \underset{x \in C}{\operatorname{argmax}}\ \frac{P(x)P(y|x)}{P(y)}$$

$$\log_2 P(y|x) = d \log_2 \alpha + (n-d)\log_2(1-\alpha)$$

$$= d \log_2 \frac{\alpha}{1-\alpha} + \cancel{n \log_2(1-\alpha)}$$

negative for $\alpha < 1/2$

$$\hat{x} = \underset{x \in C}{\operatorname{argmax}}\ \log_2 P(y|x) \longleftrightarrow \hat{x} = \underset{x \in \mathcal{C}}{\operatorname{argmin}}\, d(x, y)$$

Maximum likelihood decoding

Minimum distance decoding

Codes as matrices

- Let $\boldsymbol{u}$ be a binary vector of length k
- Let $\boldsymbol{G}$ be the generator matrix of the code $\mathcal{C}$
- A codeword can be defined as $\boldsymbol{x} = \boldsymbol{u} \cdot \boldsymbol{G}$

$$\begin{cases} \mathbf{x} \in \mathrm{Im}\{\mathbf{G}\} \\ \mathbf{x} \in \ker\{\mathbf{H}\} \end{cases}$$

- Let $\boldsymbol{H}$ be the parity check matrix of the code $\mathcal{C}$
- $\boldsymbol{x}$ is a codeword if and only if $\boldsymbol{x} \cdot \boldsymbol{H}^T = \boldsymbol{0}$

$$\mathbf{G} \cdot \mathbf{H}^T = \mathbf{0}$$

Matrices of the [3,1,3] repetition code

$$\mathbf{u} \in \{0, 1\}$$

$$\begin{cases} 0 \cdot \mathbf{G} = 000 \\ 1 \cdot \mathbf{G} = 111 \end{cases} \longrightarrow \quad \mathbf{G} = \underbrace{\begin{pmatrix} 1 & 1 & 1 \end{pmatrix}}_{n} \}k$$

$$\mathbf{H} = \underbrace{\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}}_{n} \Bigg\} n-k$$

$$111 \cdot \mathbf{H}^T = \begin{pmatrix} 1+1+0 \\ 0+1+1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Error syndrome

- Assume we transmit a codeword $\boldsymbol{x}$ over a BSC
- The effect of the channel can be modeled as adding an error vector $\boldsymbol{e}$ to the codeword

- Received sequence $\mathbf{y} = \mathbf{x} \oplus \mathbf{e}$

- Syndrome
$$\mathbf{s} = \mathbf{y} \cdot \mathbf{H}^T$$
$$\mathbf{s} = (\mathbf{x} + \mathbf{e}) \cdot \mathbf{H}^T$$
$$\mathbf{s} = \cancel{\mathbf{x} \cdot \mathbf{H}^T} + \mathbf{e} \cdot \mathbf{H}^T$$
$$\mathbf{s} = \mathbf{e} \cdot \mathbf{H}^T$$

Example

$$\mathbf{x} = 000 \qquad \mathbf{e} = 100 \longrightarrow \mathbf{y} = 100 \longrightarrow \mathbf{s} = 100 \cdot \mathbf{H}^T = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Standard Array Table

Codewords

Syndromes

000	**111**	**00**
100	011	10
010	101	11
001	110	01

2^{n-k}

2^k

Another example: the [7,4,3] Hamming code

$$G = \begin{pmatrix} 1 & & & & 1 & & 1 \\ & 1 & & & 1 & 1 & \\ & & 1 & & & 1 & 1 \\ & & & 1 & 1 & 1 & 1 \end{pmatrix}$$

$$H = \begin{pmatrix} 1 & 1 & 1 & & 1 & & \\ 1 & & 1 & 1 & & 1 & \\ 1 & 1 & & 1 & & & 1 \end{pmatrix}$$

$$C = \left\{ \begin{array}{c} 0000000, \\ 1110000, \\ 1001100, \\ 0111100, \\ 0101010, \\ 1011010, \\ 1100110, \\ 0010110, \\ 1101001, \\ 0011001, \\ 0100101, \\ 1010101, \\ 1000011, \\ 0110011, \\ 0001111, \\ 1111111. \end{array} \right\}$$

Center for Quantum Networks NSF-ERC

Another example: the [7,4,3] Hamming code

0000000	1110000	1001100	0111100	0101010	1011010	1100110	0010110	1101001	0011001	0100101	1010101	1000011	0110011	0001111	1111111	000
1000000	0110000	0001100	1111100	1101010	0011010	0100110	1010110	0101001	1011001	1100101	0010101	0000011	1110011	1001111	0111111	111
0100000	1010000	1101100	0011100	0001010	1111010	1000110	0110110	1001001	0111001	0000101	1110101	1100011	0010011	0101111	1011111	101
0010000	1100000	1011100	0101100	0111010	1001010	1110110	0000110	1111001	0001001	0110101	1000101	1010011	0100011	0011111	1101111	110
0001000	1111000	1000100	0110100	0100010	1010010	1101110	0011110	1100001	0010001	0101101	1011101	1001011	0111011	0000111	1110111	011
0000100	1110100	1001000	0111000	0101110	1011110	1100010	0010010	1101101	0011101	0100001	1010001	1000111	0110111	0001011	1111011	100
0000010	1110010	1001110	0111110	0101000	1011000	1100100	0010100	1101011	0011011	0100111	1010111	1000001	0110001	0001101	1111101	010
0000001	1110001	1001101	0111101	0101011	1011011	1100111	0010111	1101000	0011000	0100100	1010100	1000010	0110010	0001110	1111110	001

The size of the table scales exponentially

[3,2,2] Single Parity Check code

[3,1,3] Repetition code

[3,2,2] Single Parity Check code

$$\mathbf{G}_R = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$

$$\mathbf{G}_S = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

$$\mathbf{H}_R = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

$$\mathbf{H}_S = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$

$$\mathbf{H}_R \cdot \mathbf{H}_S^T = \mathbf{0}$$

Dual codes $\mathcal{C},\, \mathcal{C}^{\perp}$

Low Density Parity Check (LDPC) codes

Sparse parity check matrix

↓

Low complexity iterative decoding (on graphs)

Tanner graph

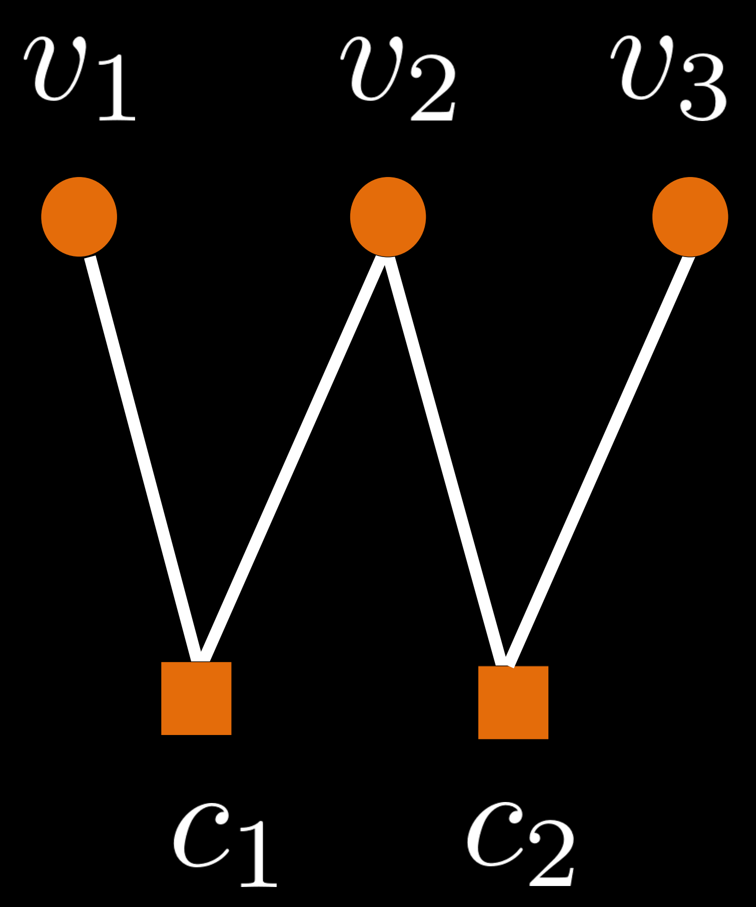
$$\begin{array}{cccc} v_1 & v_2 & v_3 & \\ \end{array}$$

$$\mathbf{H} = \begin{array}{c} \\ \left(\begin{array}{ccc} 1 & 1 & 0 \\ 0 & 1 & 1 \end{array}\right) \end{array} \begin{array}{c} c_1 \\ c_2 \end{array}$$

$V = \{v_1, v_2, v_3\}$ Variable nodes

$C = \{c_1, c_2\}$ Check nodes

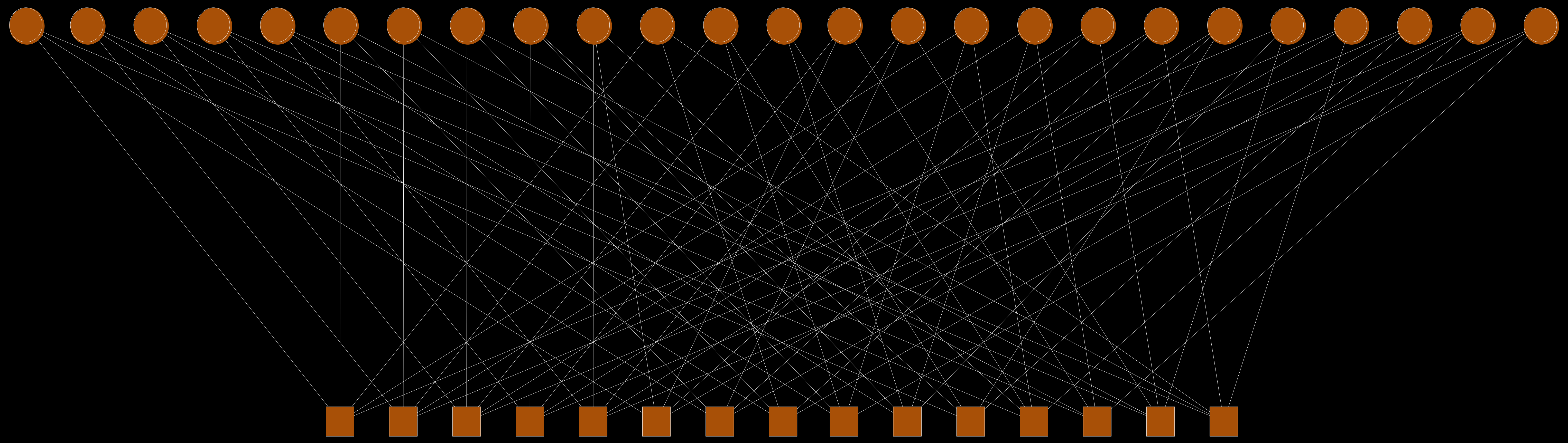
$E = \{(v_j, c_i) \mid \mathbf{H}_{i,j} = 1\}$



- Bipartite graph

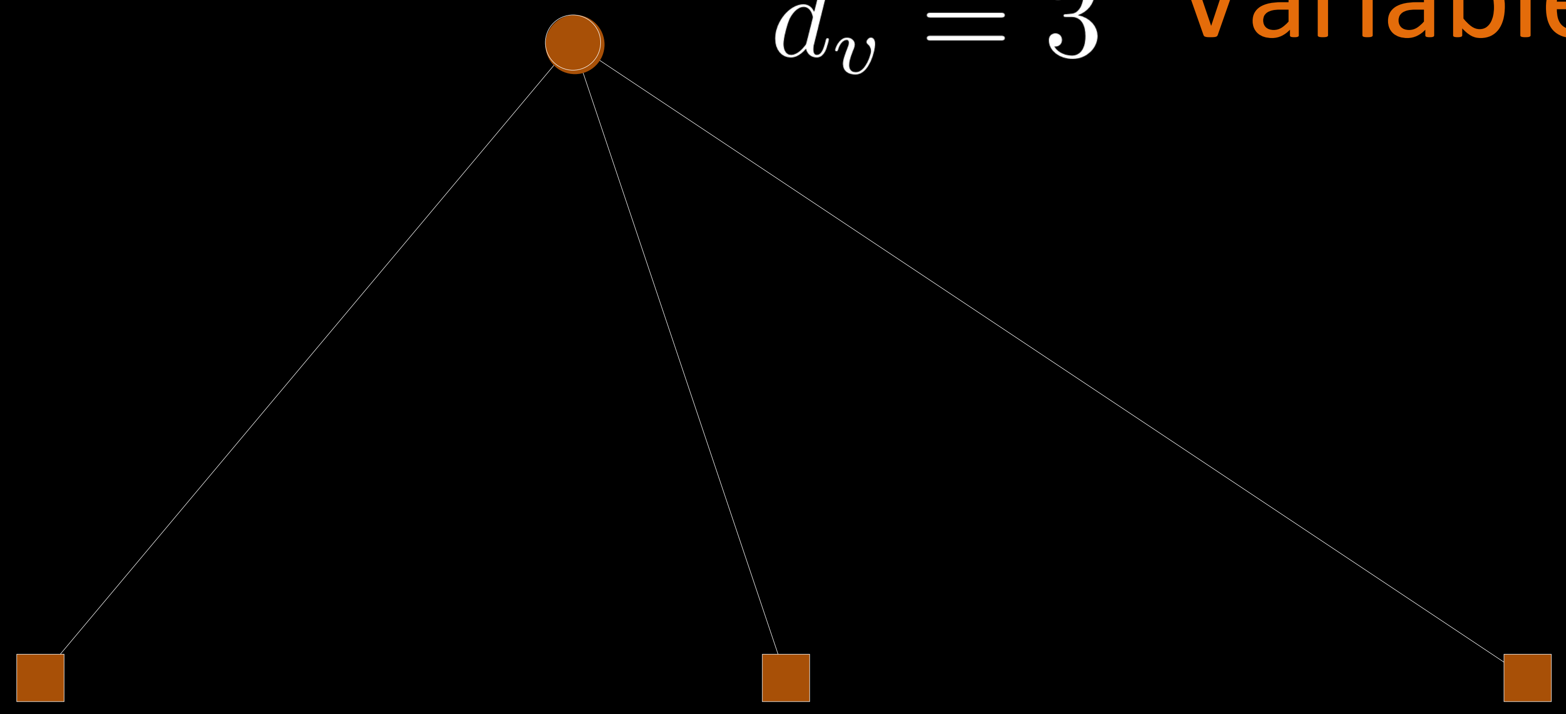
Tanner graph

(d_v, d_c) - regular code



v

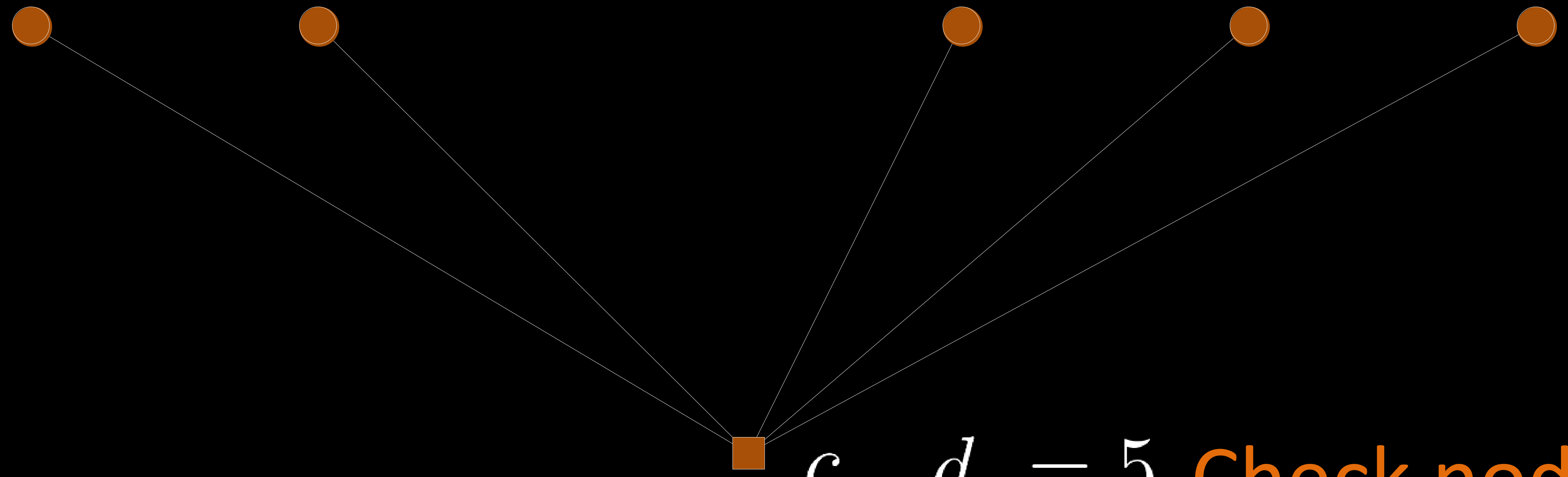
$d_v = 3$ Variable node degree



$\mathcal{N}(v)$

Neighborhood

$\mathcal{N}(c)$



c $d_c = 5$ Check node degree

Iterative decoding

- Low complexity algorithms
- Operate on the Tanner graph
- Each iteration involves an exchange of messages between variable nodes and check nodes
- If the estimated error matches the syndrome, the decoder succeeds, otherwise it continues for another iteration

Bit-Flipping algorithm

Initialization – variable-to-check messages

$$\nu_{j\to i}^{(0)} = y_j \in \{0,1\}$$

(received value from channel)

$$\mu_{i\to j}^{(k)} = \bigoplus_{j\in\mathcal{N}(i)} \nu_{j\to i}^{(k-1)}$$

(number of unsatisfied checks)

Check-to-variable messages

$$\nu_{j\to i}^{(k)} = \begin{cases} \nu_{j\to i}^{(k-1)} \oplus 1, \ \text{if} \ \left(\sum_{i\in\mathcal{N}(j)} \mu_{i\to j}^{(k)}\right) > \beta \\ \nu_{j\to i}^{(k-1)}, \ \text{otherwise} \end{cases}$$

$$\beta = \left\lfloor \frac{d_v}{2} \right\rfloor$$

Variable-to-check messages

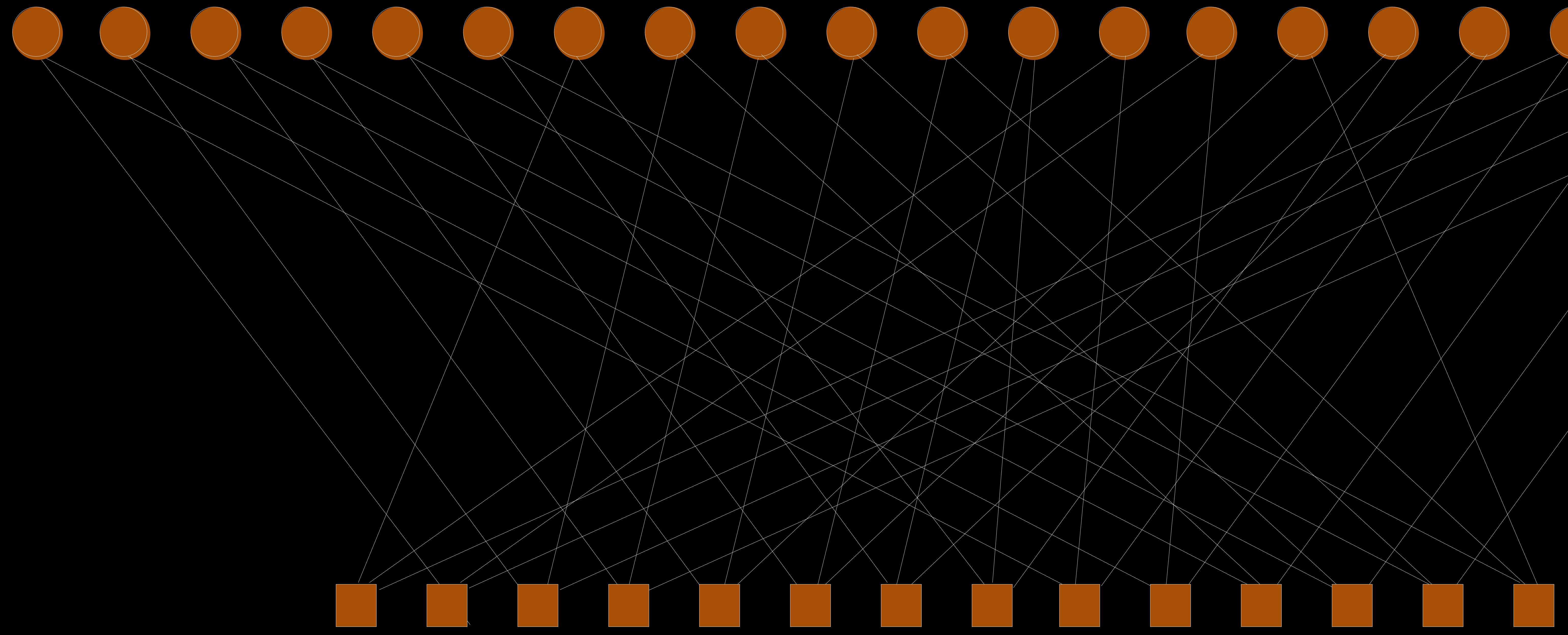
Threshold

Example

$$H = \begin{pmatrix}
 & & & & & & 1 & & & & & & 1 & & & & & 1 & & & \\
1 & & & & & & & & & & & & & 1 & & & & & 1 & & \\
 & 1 & & & & & & 1 & & & & & & & & & & & & 1 & \\
 & & 1 & & & & & & 1 & & & & & & & & & & & & 1 \\
 & & & 1 & & & & & & 1 & & & & & 1 & & & & & & \\
 & & & & 1 & & & & & & 1 & & & & & 1 & & & & & \\
 & & & & & 1 & & & & & & 1 & & & & & 1 & & & & \\
 & & & & & & 1 & & & & & 1 & & & & 1 & & & & & \\
1 & & & & & & & & & & & & 1 & & & & 1 & & & & \\
 & 1 & & & & & & & & & & & & 1 & & & & 1 & & & \\
 & & 1 & & & & & 1 & & & & & & & & & & & 1 & & \\
 & & & 1 & & & & & 1 & & & & & & & & & & & 1 & \\
 & & & & 1 & & & & & 1 & & & & & & & & & & & 1 \\
 & & & & & 1 & & & & & 1 & & & & 1 & & & & & &
\end{pmatrix}$$

Example

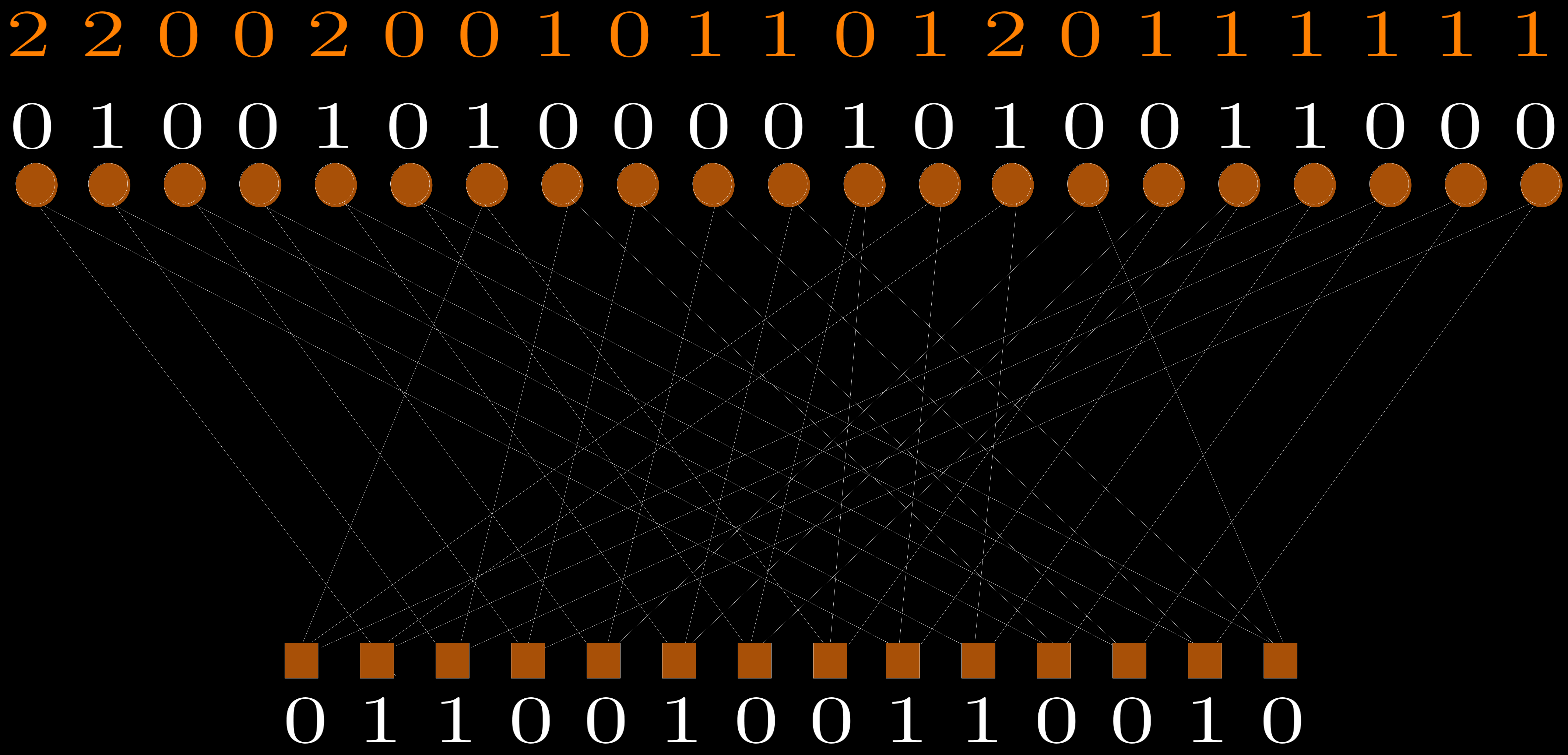
$$\mathbf{y} = 0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 0$$



$$\mathbf{s} = 0\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 0$$

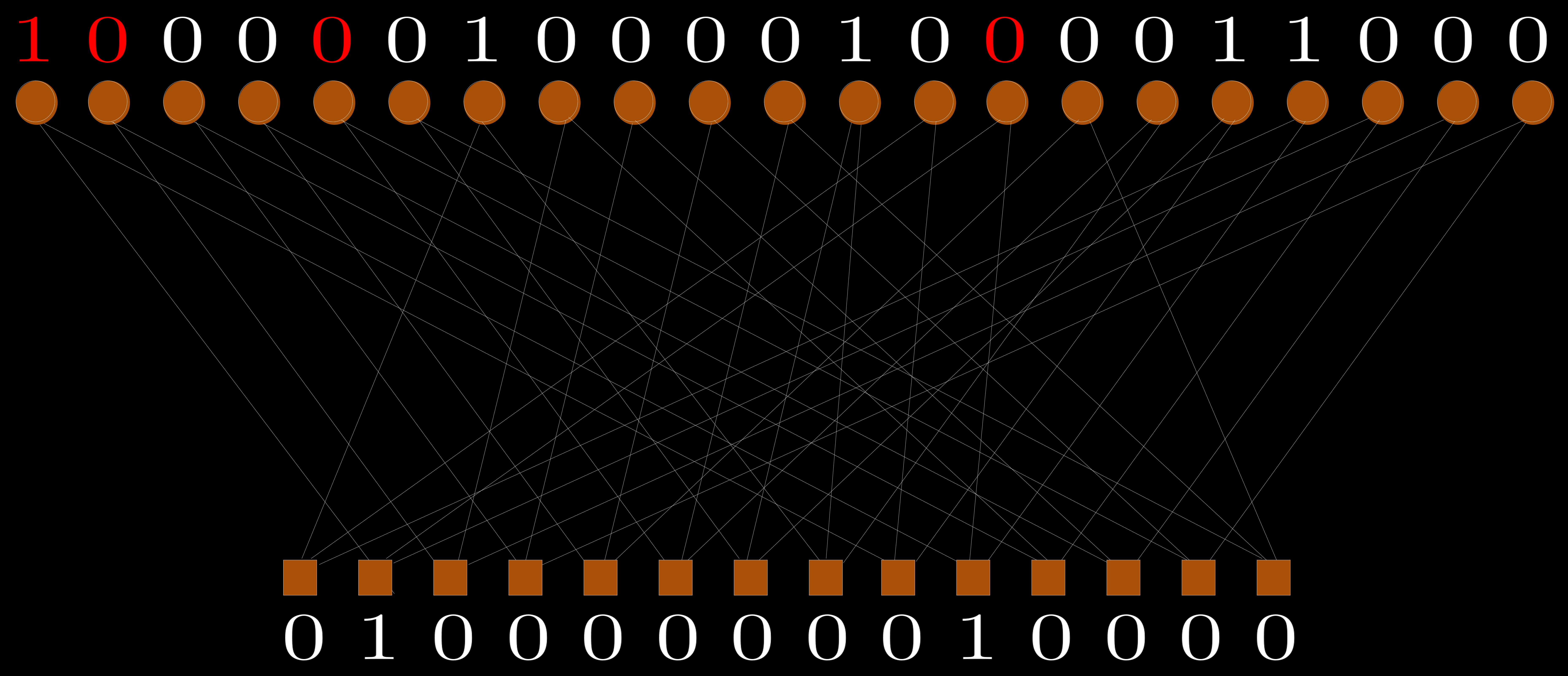
$$\beta = 1 \qquad \mu^{(1)}_{i\to j} = \bigoplus_{j \in \mathcal{N}(i)} \nu^{(0)}_{j \to i}$$

Example



$$\nu_{j\to i}^{(1)} = \begin{cases} \nu_{j\to i}^{(0)} \oplus 1, \text{ if } \left(\sum_{i \in \mathcal{N}(j)} \mu_{i\to j}^{(1)} \right) > \beta \\ \nu_{j\to i}^{(0)}, \text{ otherwise} \end{cases}$$

Example



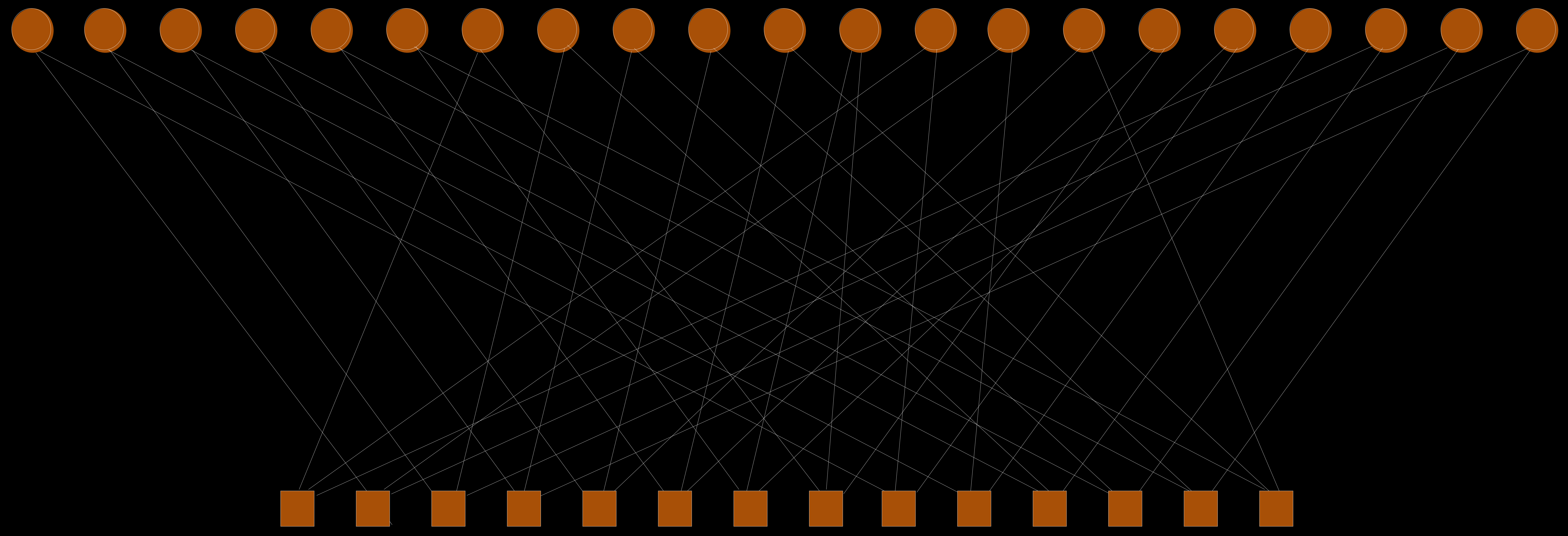
$$\mu^{(2)}_{i\to j} = \bigoplus_{j\in\mathcal{N}(i)} \nu^{(1)}_{j\to i}$$

Center for Quantum Networks
NSF-ERC

Example

1 1 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 1

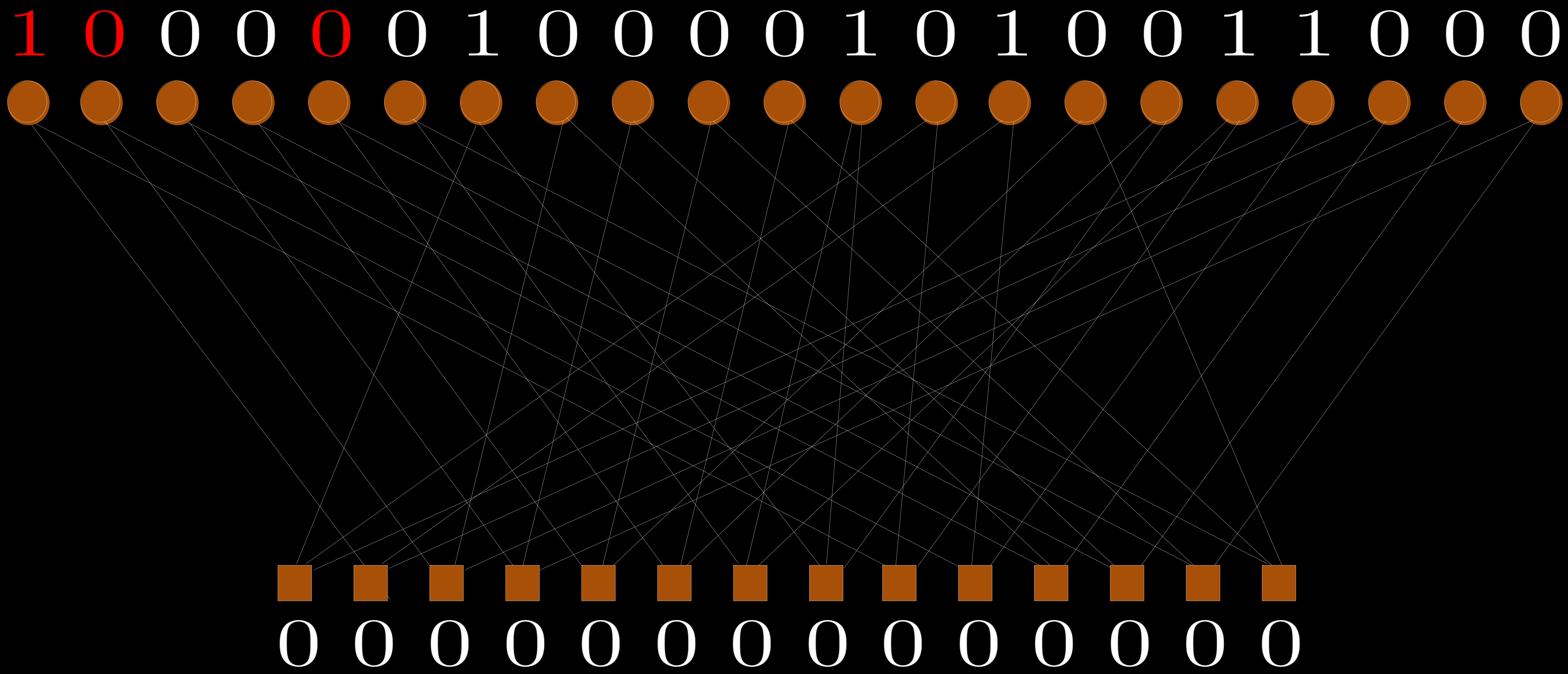
1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0



0 1 0 0 0 0 0 0 0 1 0 0 0 0

$$\nu^{(2)}_{j\to i} = \begin{cases} \nu^{(1)}_{j\to i} \oplus 1, \text{ if } \left(\sum_{i\in\mathcal{N}(j)} \mu^{(2)}_{i\to j} \right) > \beta \\ \nu^{(1)}_{j\to i}, \text{ otherwise} \end{cases}$$

Example



Belief propagation (or sum-product)

Transmitted codeword x

Received sequence y

$$\hat{\mathbf{x}} = \underset{\mathbf{x} \in C}{\operatorname{argmax}} \, P(\mathbf{x}|\mathbf{y})$$

Posterior joint probability distribution

$$\hat{x}_j = \underset{x_j \in \{0,1\}}{\operatorname{argmax}} \, P(x_j|\mathbf{y}) = \underset{x_j \in \{0,1\}}{\operatorname{argmax}} \, \sum_{x_1,\ldots,x_{j-1},x_{j+1},\ldots,x_n \in \{0,1\}^{n-1}} P(x|\mathbf{y})$$

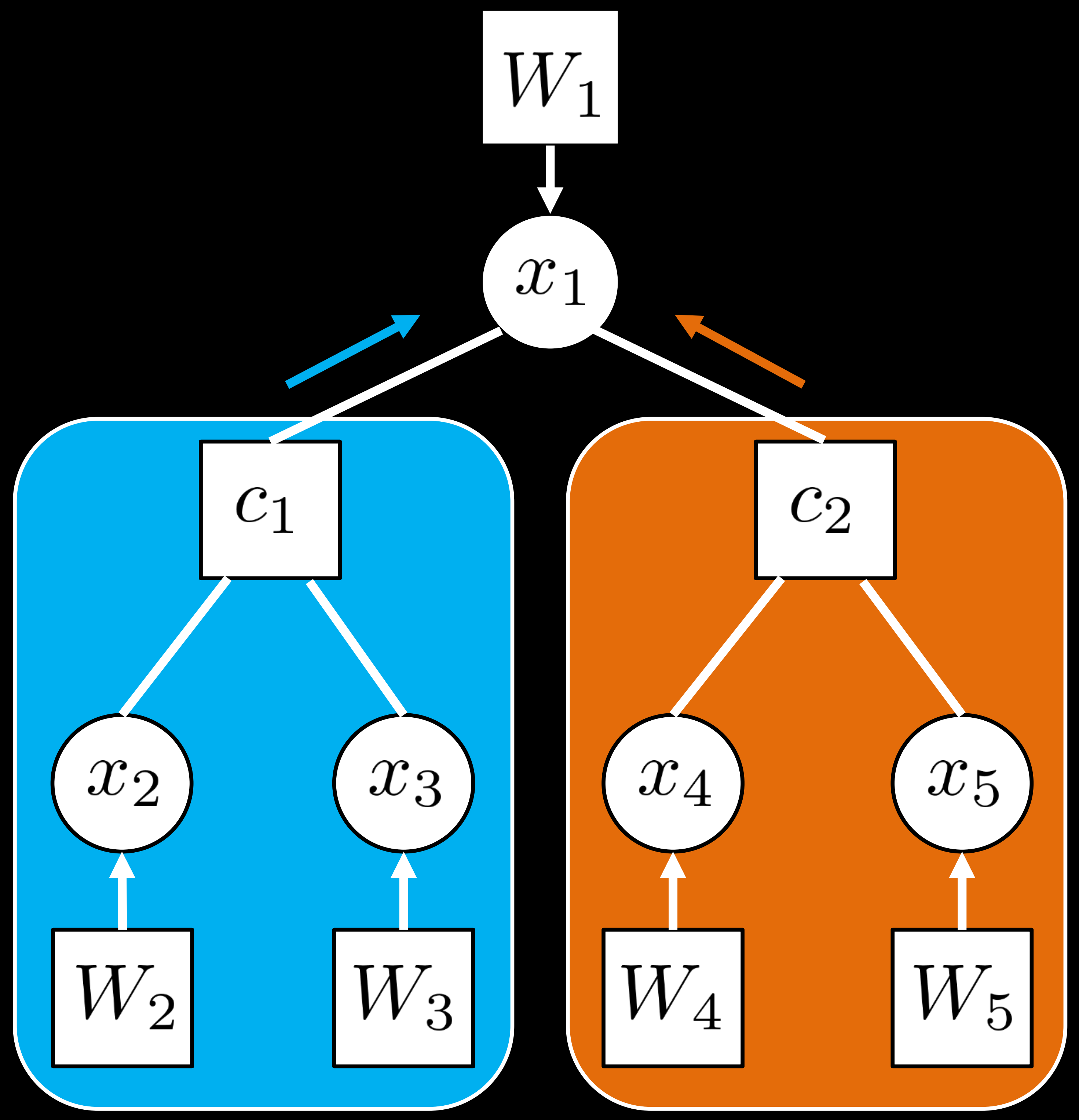
Posterior marginal distributions

Belief propagation (or sum-product)

$$\hat{x}_1 = \underset{x_1\in\{0,1\}}{\operatorname{argmax}}\ P(x_1|y) = \underset{x_1\in\{0,1\}}{\operatorname{argmax}} \sum_{x_2,x_3,x_4,x_5\in\{0,1\}^4} P(x|y)$$

$$= \underset{x_1\in\{0,1\}}{\operatorname{argmax}} \sum_{x_2,x_3,x_4,x_5\in\{0,1\}^4} P(y|x)P(x)$$

$$= \underset{x_1\in\{0,1\}}{\operatorname{argmax}} \sum_{x_2,x_3,x_4,x_5\in\{0,1\}^4} \left(\prod_{j=1}^{5} W_j\right) \mathbb{I}(c_1=0)\mathbb{I}(c_2=0)$$



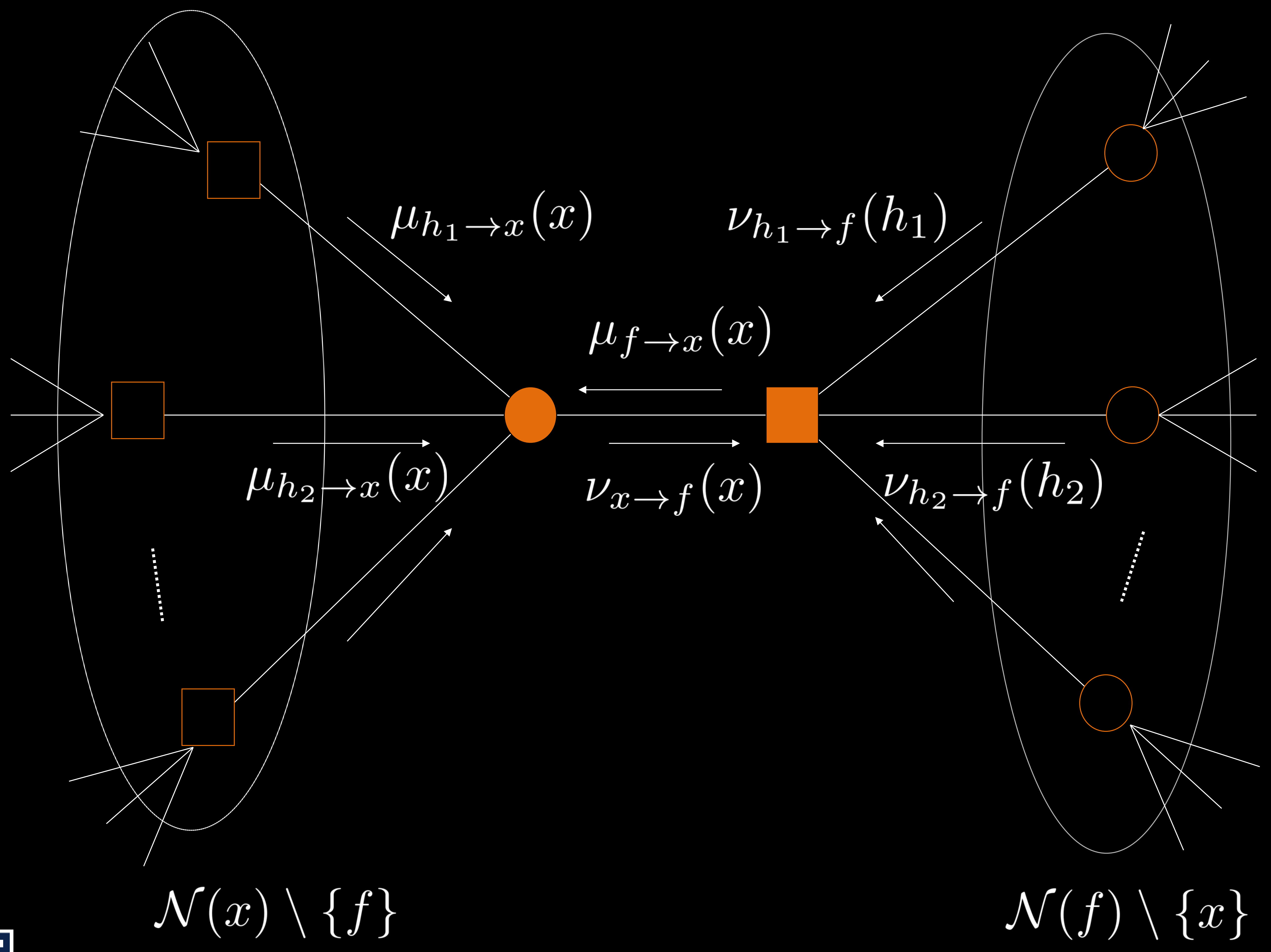
$$W_j = P(y_j|x_j)$$

$$= \underset{x_1\in\{0,1\}}{\operatorname{argmax}}\ W_1 \cdot \left(\sum_{x_2,x_3\in\{0,1\}^2} \mathbb{I}(c_1=0)\cdot W_2\cdot W_3\right)\left(\sum_{x_4,x_5\in\{0,1\}^2} \mathbb{I}(c_2=0)\cdot W_4\cdot W_5\right)$$

$c_1 :\ x_1 + x_2 + x_3 = 0$

$c_2 :\ x_1 + x_4 + x_5 = 0$

Belief propagation (or sum-product)



$$\nu_{x \to f}(x) = \prod_{h \in \mathcal{N}(x) \setminus \{f\}} \mu_{h \to x}(x)$$

$$\mu_{f \to x}(x) = \sum_{\{x\}} \left(f(X) \prod_{h \in \mathcal{N}(f) \setminus \{x\}} \nu_{h \to f}(h) \right)$$

$$\sigma_i(x_i) = \prod_{h \in \mathcal{N}(x_i)} \mu_{h \to x_i}(x_i)$$

Monte Carlo simulations of error correction

- Repeated random sampling to obtain numerical results
- MC simulation is the standard method for assessing the performance of a code/decoder
- Simulation software needs to be fast to generate the adequate number of samples in a relatively short time

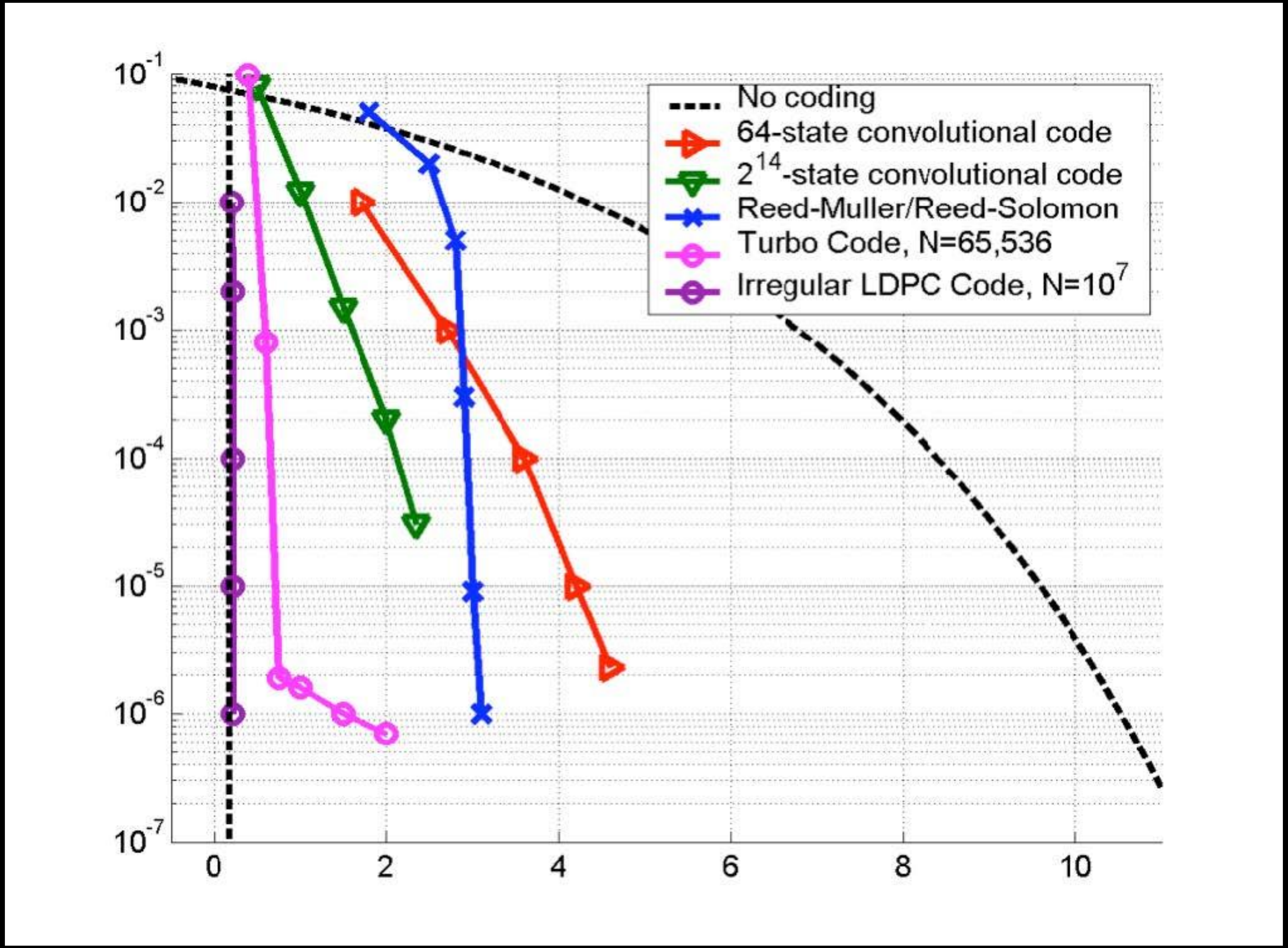
Monte Carlo simulations of error correction

- Choose an error model
- Choose a code and a decoder
- Fix the number of samples (large enough)
- Fix the parameters of the error model (e.g., for BSC)

- For each sample:
  - Generate a codeword (usually the all-zero codeword is sufficient)
  - Generate a random sample of noise, add it to the codeword
  - Compute the syndrome and start decoding
  - If the decoder fails (doesn't converge or estimates a different codeword) → count a failure

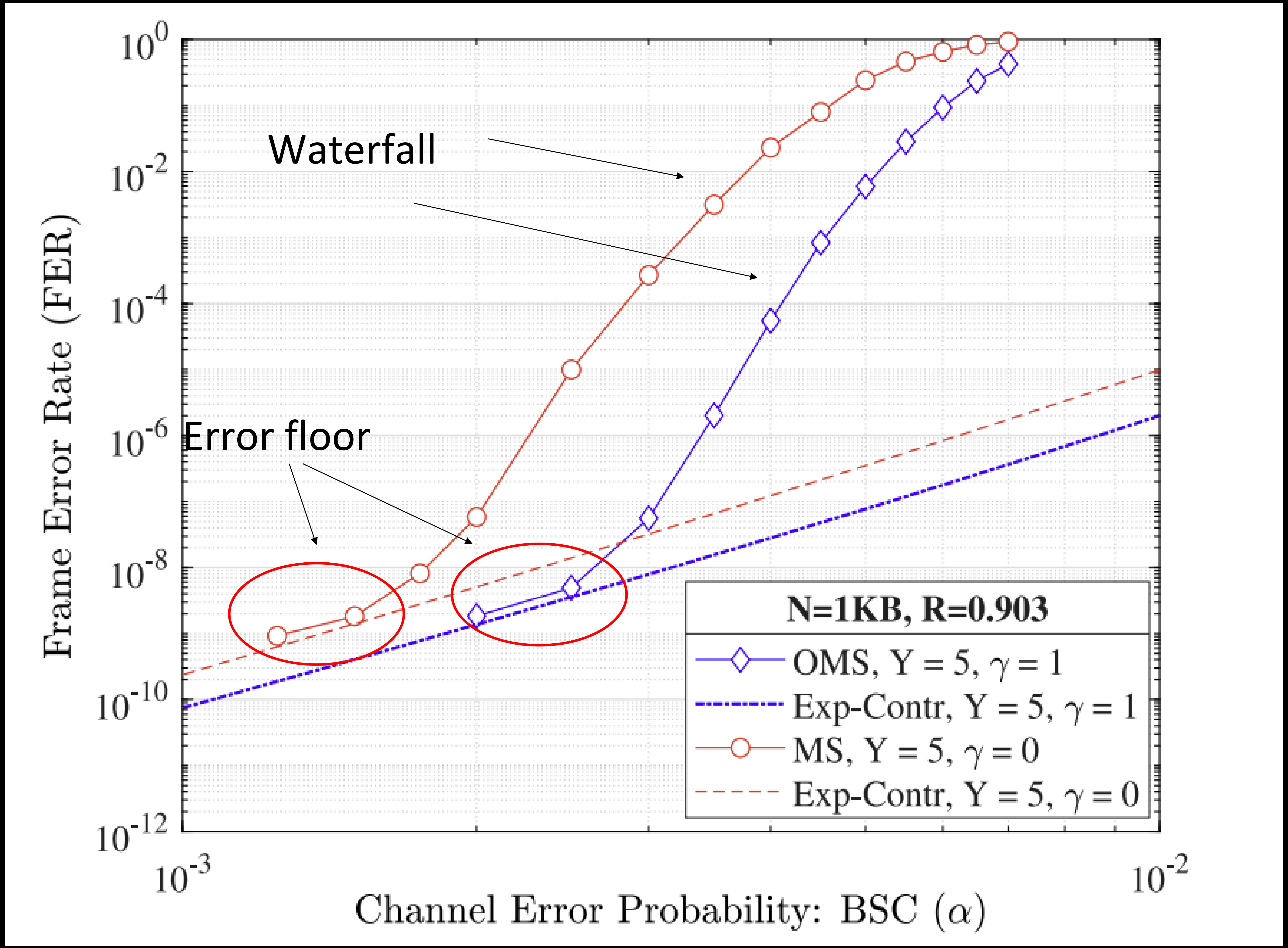
- Divide the number of failures for the total number of samples
- Repeat for different parameters of the error model
- Plot error parameters vs failure rates

Plot examples



http://www.jaist.ac.jp/~kurkoski/teaching/portfolio/uec_s05/S05-LDPC%20Lecture%201.pdf

Plot examples



N. Raveendran, D. Declercq and B. Vasić, "A Sub-Graph Expansion-Contraction Method for Error Floor Computation," in *IEEE Transactions on Communications*, vol. 68, no. 7, pp. 3984-3995, 2020

Agenda

Part 1 Classical Error Correction

By Michele Pacenti

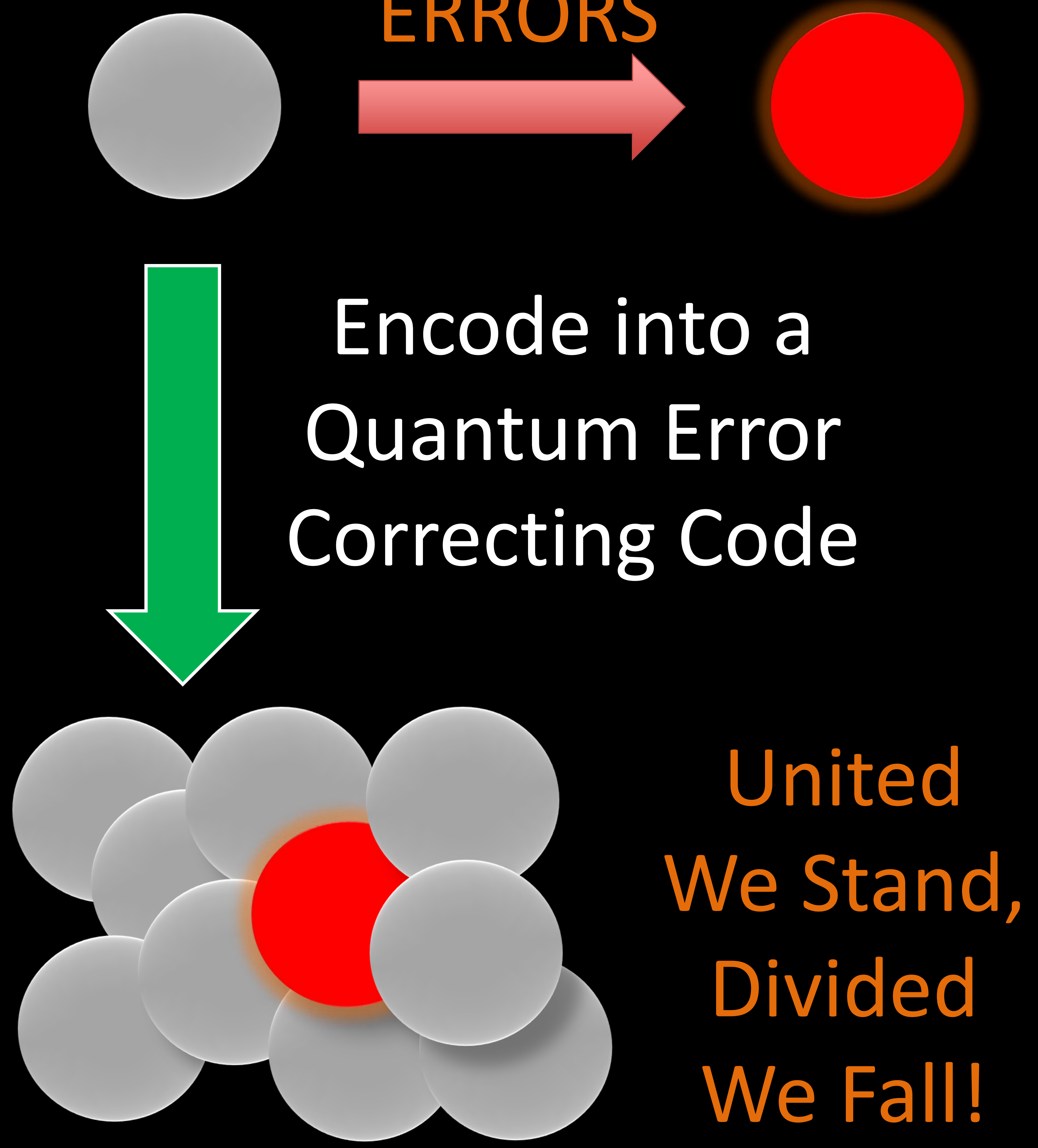
Part 2 Quantum Error Correction

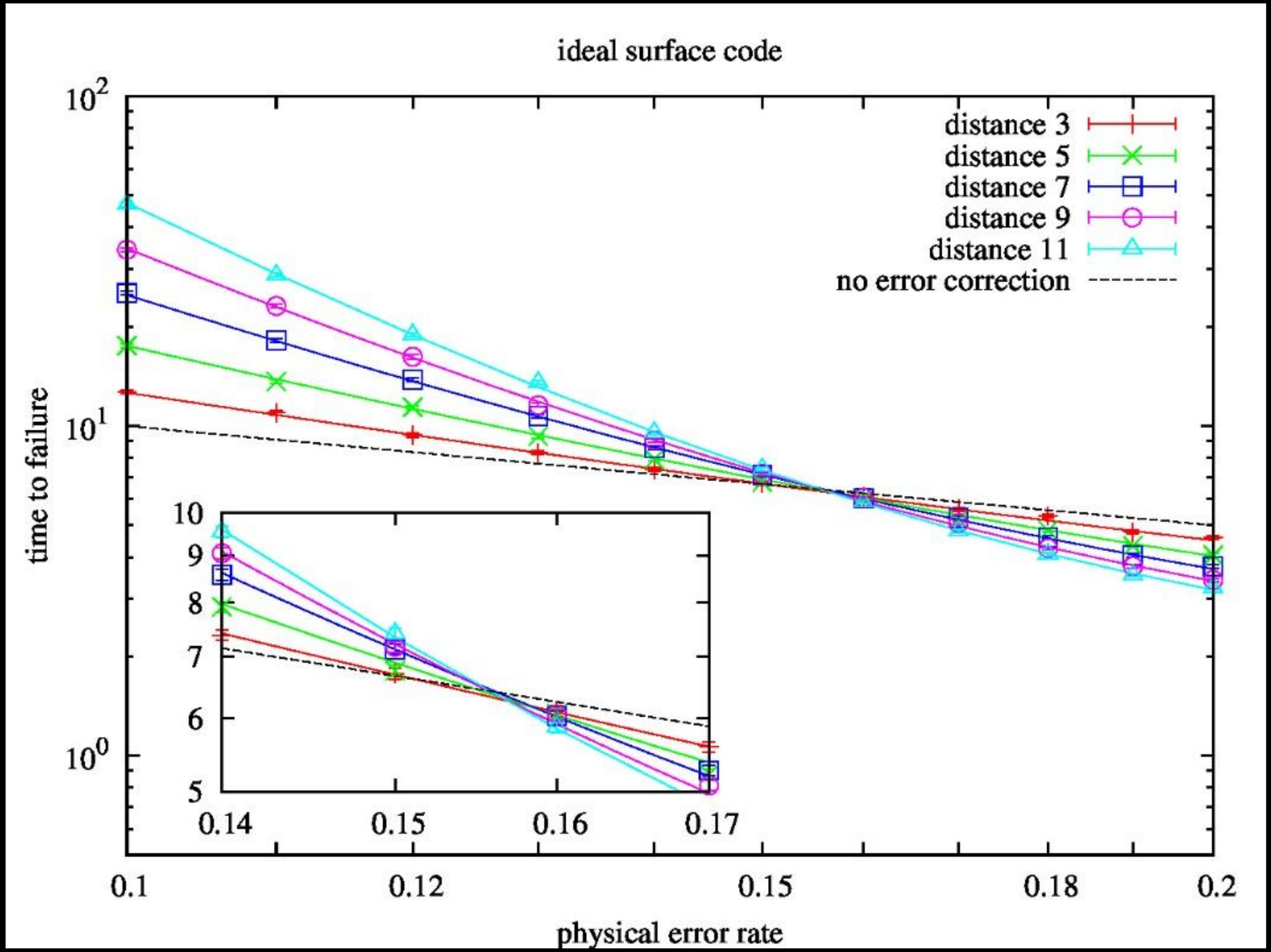
By Narayanan Rengaswamy

Part 3 Application to Quantum Networks

By Sijie Cheng

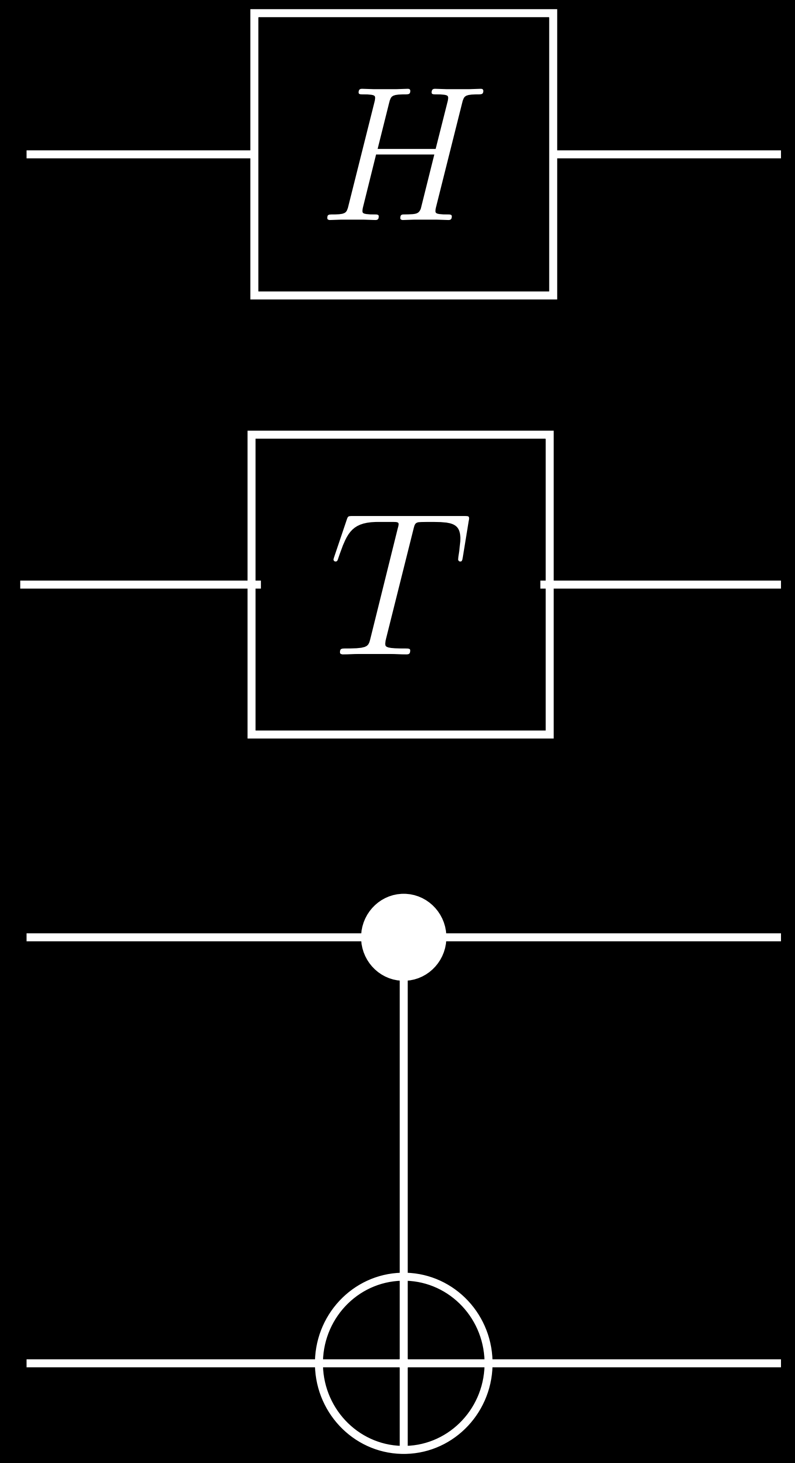
What do we gain from QEC?





QEC Goals: Computation & Error Correction

Universal gates on
k logical qubits:



$$
\begin{tikzcd}[column sep=huge, row sep=huge]
k \text{ qubits} \quad |\psi\rangle_L \arrow[rr, "\substack{\text{Noisy}\\ \text{Logical operation}}"] \arrow[d, "\substack{\text{QECC}\\ \text{Encode}}"'] \arrow[rd, phantom, "\substack{\text{Translate}\\ \text{(Synthesize)}}"] & & |\phi\rangle_L \\
n \text{ qubits} \quad \overline{|\psi\rangle} \arrow[rr, "\substack{\text{Physical operation}\\ \text{Fault-tolerant}}"'] & & \overline{|\phi\rangle} \arrow[u, "\substack{\text{QECC}\\ \text{Decode}}"']
\end{tikzcd}
$$

QECC: Quantum Error Correcting Code

Pauli Operators and Bloch Sphere

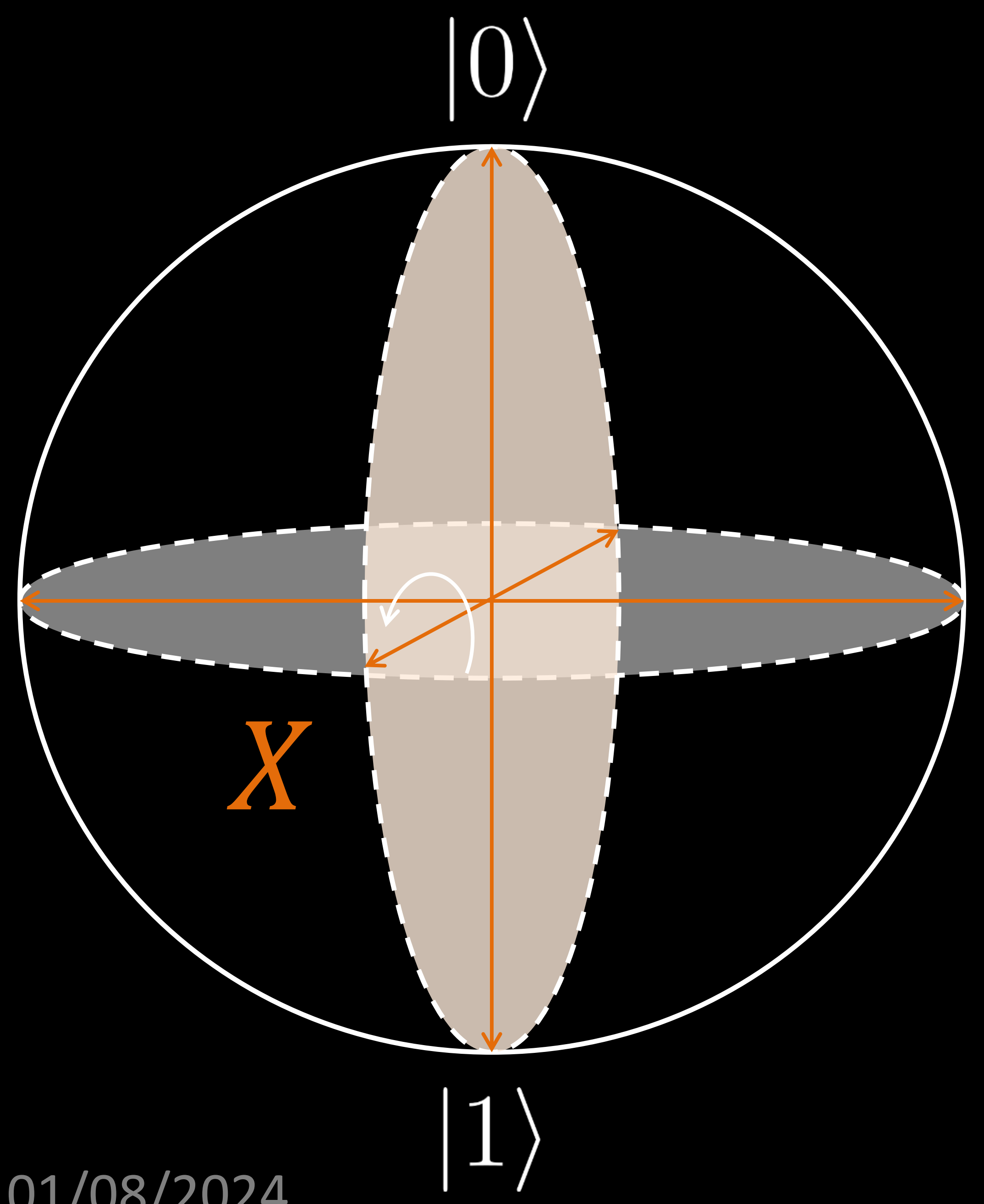
Center for Quantum Networks NSF-ERC

Bit Flip

"Quantum" NOT gate

$$|0\rangle \; \text{—}\boxed{X}\text{—} \; |1\rangle$$

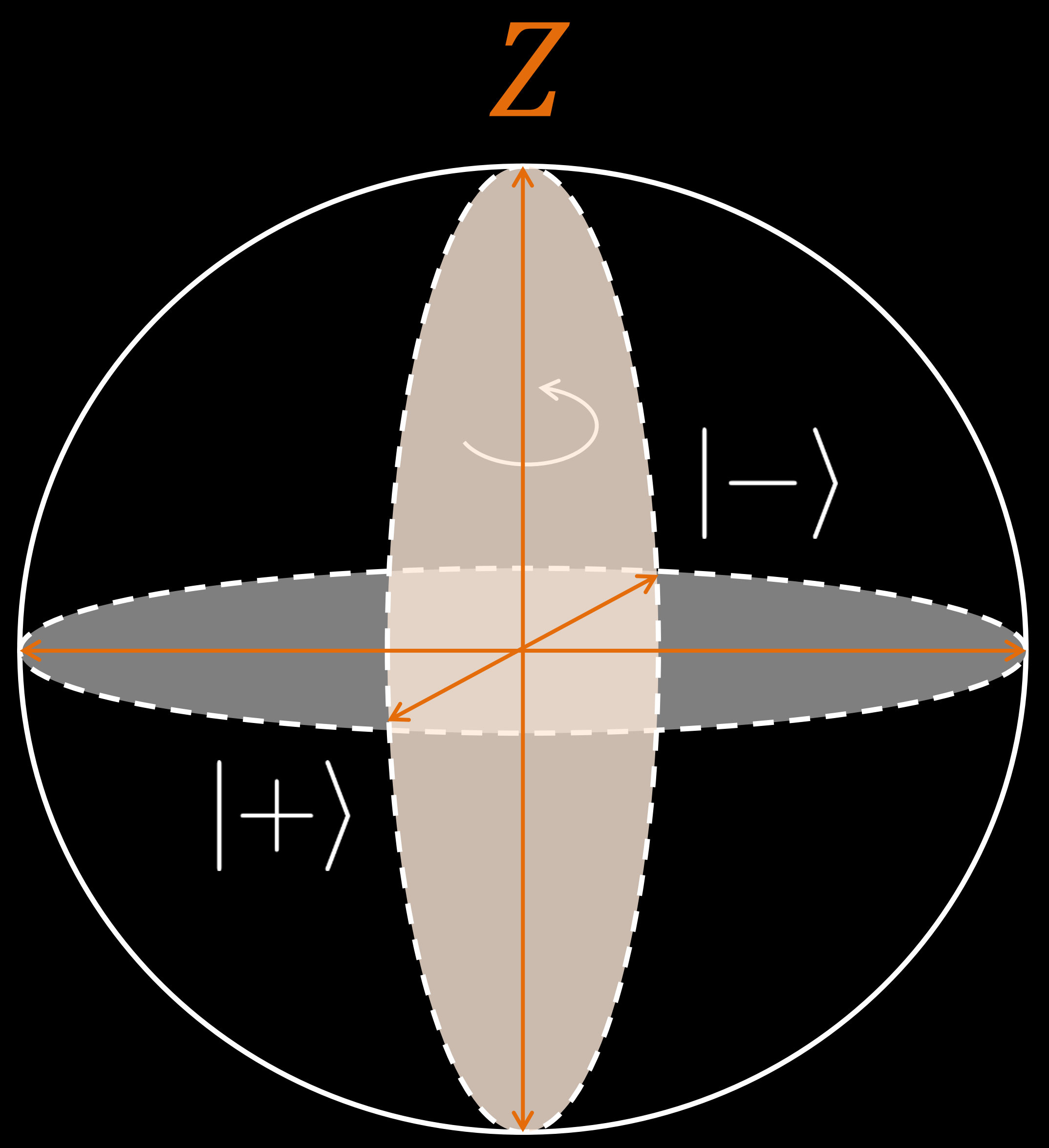
$$|1\rangle \; \text{—}\boxed{X}\text{—} \; |0\rangle$$



Phase Flip

$$|0\rangle \; \text{—}\boxed{Z}\text{—} \; |0\rangle$$

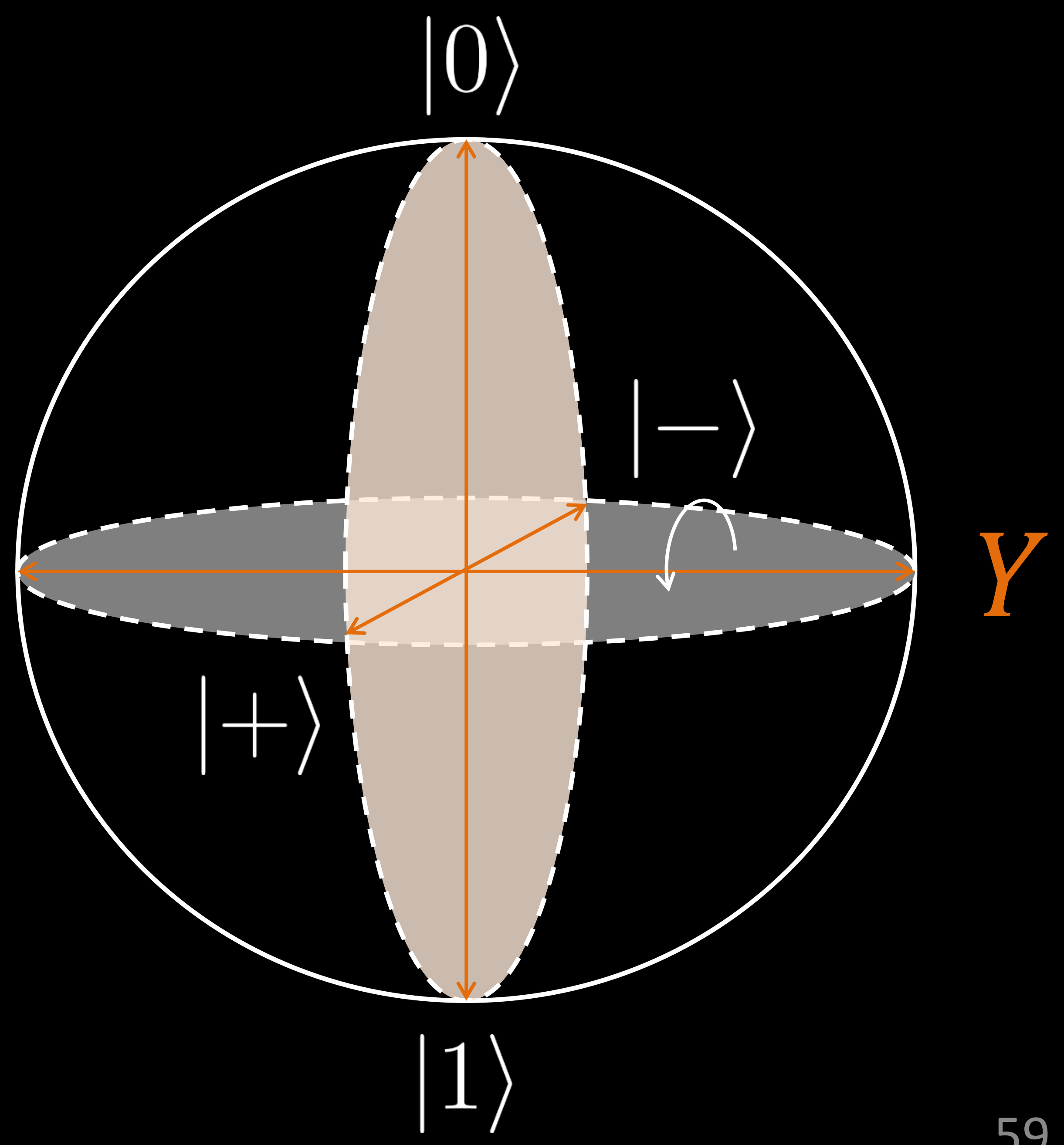
$$|1\rangle \; \text{—}\boxed{Z}\text{—} \; -|1\rangle$$



Bit-Phase Flip

$$|0\rangle \; \text{—}\boxed{Y}\text{—} \; i\,|1\rangle$$

$$|1\rangle \; \text{—}\boxed{Y}\text{—} \; -i\,|0\rangle$$



Pauli Errors and Depolarizing Channel

Bit Flip

$$|0\rangle \;—\boxed{X}—\; |1\rangle$$

$$|1\rangle \;—\boxed{X}—\; |0\rangle$$

Phase Flip

$$|0\rangle \;—\boxed{Z}—\; |0\rangle$$

$$|1\rangle \;—\boxed{Z}—\; -|1\rangle$$

Bit-Phase Flip

$$|0\rangle \;—\boxed{Y}—\; i\,|1\rangle$$

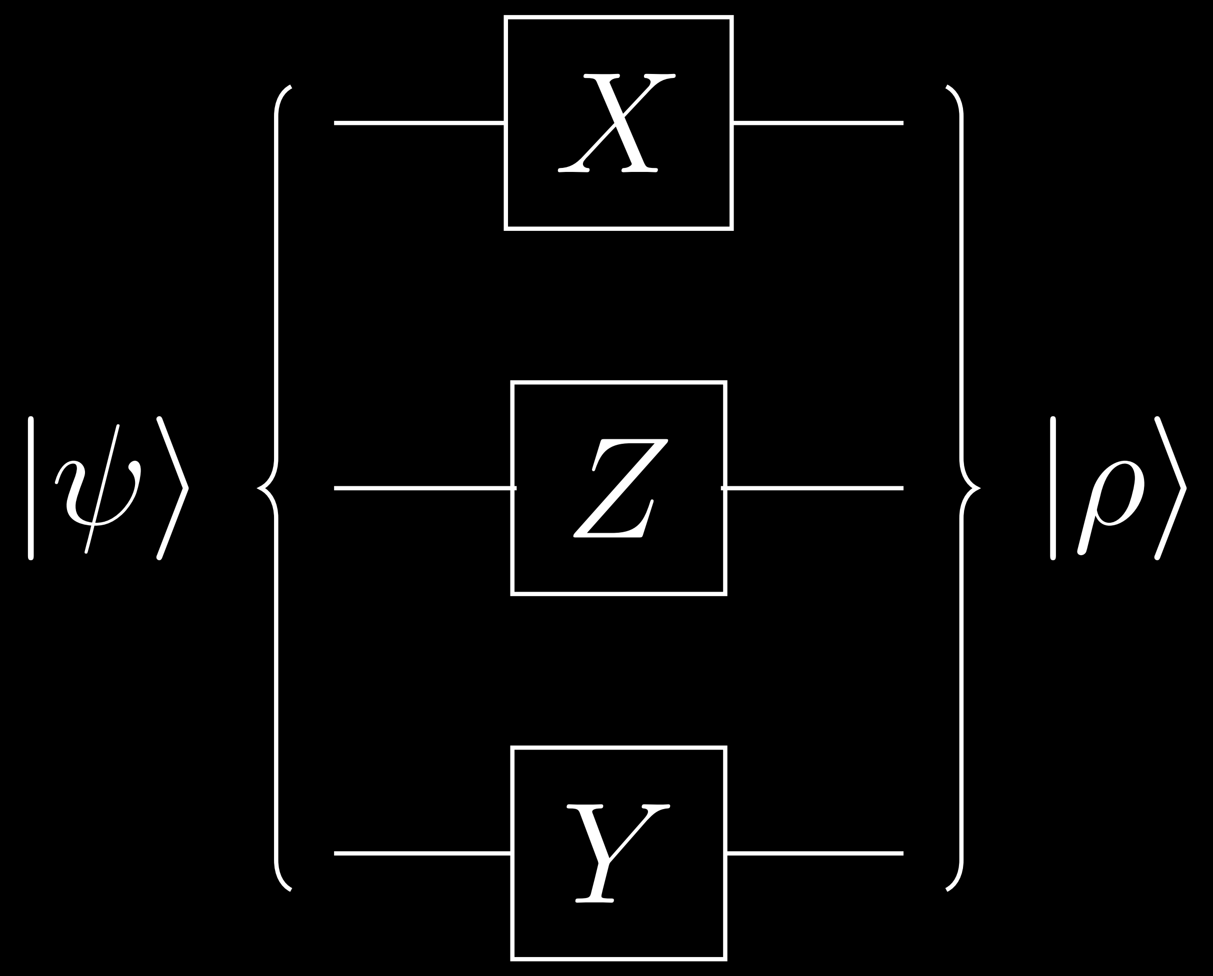
$$|1\rangle \;—\boxed{Y}—\; -i\,|0\rangle$$

The "Quantum" ~~Binary~~ Symmetric Channel:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \;—\boxed{\epsilon(I, X, Z, Y)}—\; |\rho\rangle = \begin{cases} |\psi\rangle & \text{with prob. } 1-\epsilon, \\ X\,|\psi\rangle & \text{with prob. } \frac{\epsilon}{3}, \\ Z\,|\psi\rangle & \text{with prob. } \frac{\epsilon}{3}, \\ Y\,|\psi\rangle & \text{with prob. } \frac{\epsilon}{3}. \end{cases}$$

Depolarizing Channel

Multi-Qubit Pauli Operators/Errors



$n = 3$ qubits

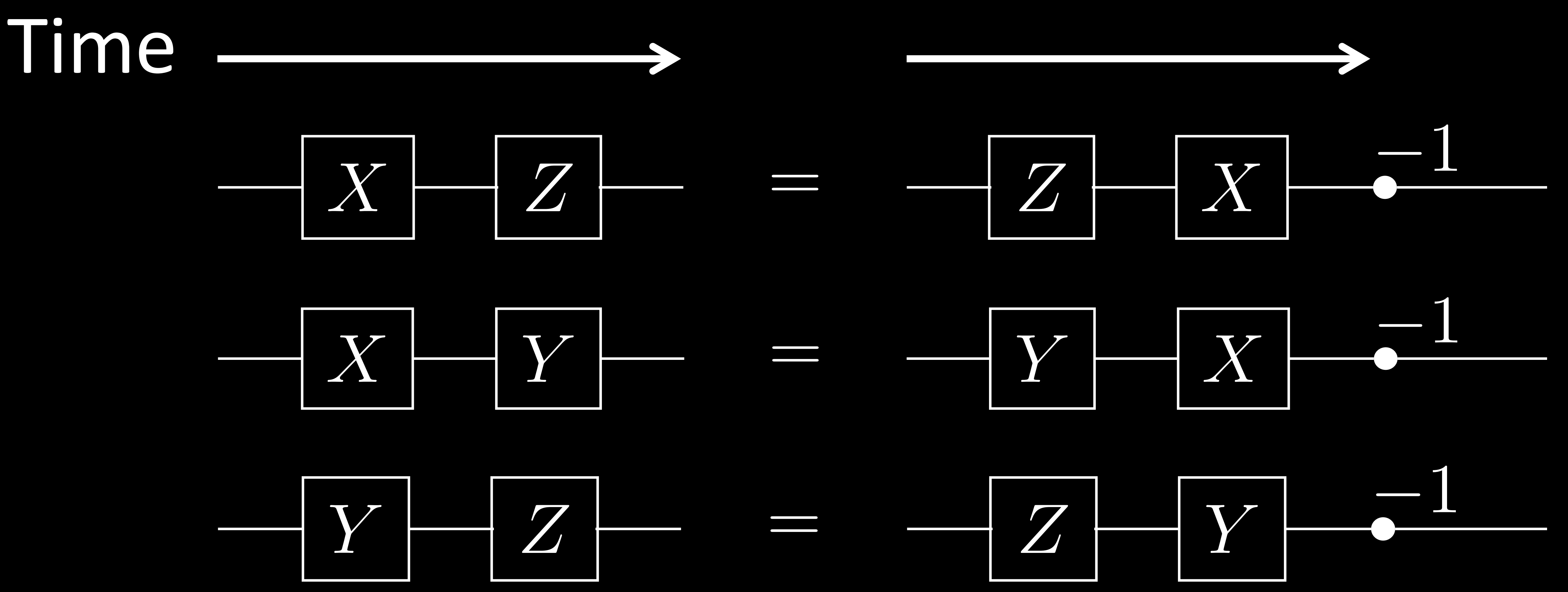
$X \otimes Z \otimes Y$
$\equiv X_1 Z_2 Y_3$
$\equiv XZY$

Pauli Group on n qubits:

All possible tensor products of I, X, Z, Y with global phase from the set $\{1, i, -1, -i\}$

1. Group generated by I_j, X_j, Z_j, Y_j on j^{th} qubit
2. The 4^n Hermitian operators form an orthogonal basis for all $2^n \times 2^n$ matrices
3. Trace Inner Product: $\langle A, B \rangle \coloneqq \mathrm{Tr}(A^\dagger B)$

Commutativity of Pauli Operators

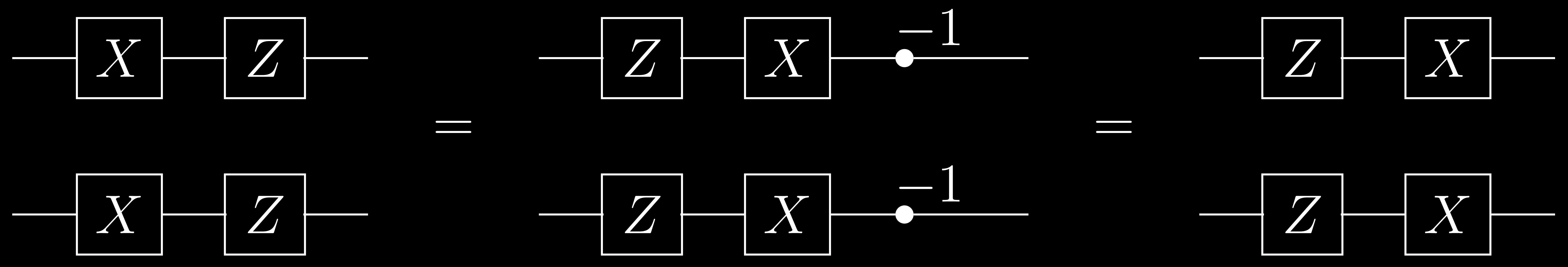


$$ZX|\psi\rangle = -XZ|\psi\rangle$$

$$YX|\psi\rangle = -XY|\psi\rangle$$

$$ZY|\psi\rangle = -YZ|\psi\rangle$$

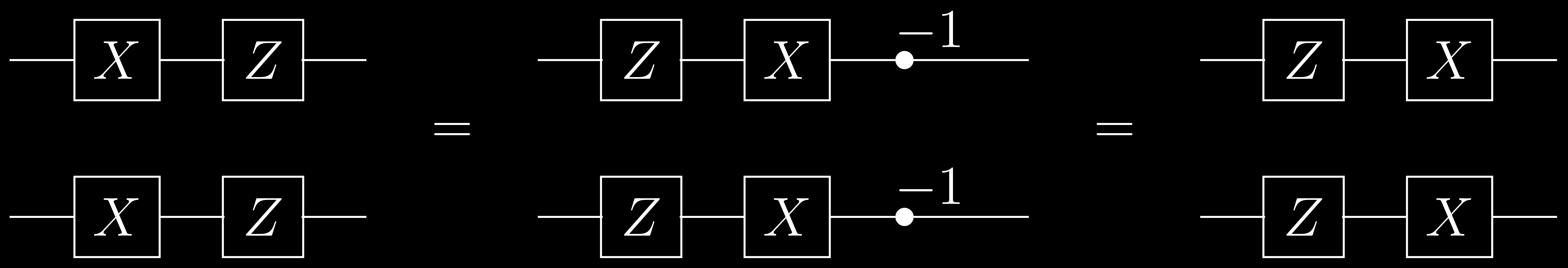
$n = 2$ qubits Property: $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$



$$\begin{aligned}&(Z \otimes Z)(X \otimes X)\\ &= (ZX) \otimes (ZX)\\ &= (-XZ) \otimes (-XZ)\\ &= (X \otimes X)(Z \otimes Z)\end{aligned}$$

Stabilizers: Commuting Pauli Operators

$n = 2$ qubits



$(Z \otimes Z)(X \otimes X)$
$= (X \otimes X)(Z \otimes Z)$

Stabilizer Group: $S = \langle X \otimes X, Z \otimes Z \rangle = \{II, XX, ZZ, -YY\}$

Linear Algebra: When Hermitian matrices commute, they can be simultaneously diagonalized, i.e., they have a <u>common basis of eigenvectors</u>

Stabilizer Codes from Stabilizer Groups

Stabilizer Group: $S = \langle X \otimes X, Z \otimes Z \rangle = \{II, XX, ZZ, -YY\}$

"Bell Basis": The common eigenbasis of these "Bell state" stabilizers

$$|\Phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \quad , \quad |\Phi^-\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}} \quad , \quad |\Psi^+\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}} \quad , \quad |\Psi^-\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \Rightarrow XX\,|\Phi^+\rangle = \frac{X\,|0\rangle\, X\,|0\rangle + X\,|1\rangle\, X\,|1\rangle}{\sqrt{2}} = \frac{|11\rangle + |00\rangle}{\sqrt{2}} = (+1)\,|\Phi^+\rangle$$

Stabilizer Code $Q(S)$: Subspace spanned by the +1 eigenvectors of S

$\boxed{\text{In this case we only get a 1-dimensional space spanned just by } |\Phi^+\rangle}$

Stabilizer Codes from Stabilizer Groups

Stabilizer Group: $S = \langle \cancel{X \otimes X}, Z \otimes Z \rangle \rightarrow \langle Z \otimes Z \rangle \equiv \langle ZZ \rangle$

For n qubits, remove some of the n generators to make space for qubits!

$$|\Phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \;,\; |\Phi^-\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}} \;,\; \cancel{|\Psi^+\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}}} \;,\; \cancel{|\Psi^-\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}}$$

Stabilizer Code $Q(S)$: Subspace spanned by the +1 eigenvectors of S

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \xrightarrow{\text{Encoding}} \overline{|\psi\rangle} = \alpha|\Phi^+\rangle + \beta|\Phi^-\rangle$$

$$\equiv \alpha|00\rangle + \beta|11\rangle$$

$k = 1$ logical qubit

$n = 2$ physical qubits

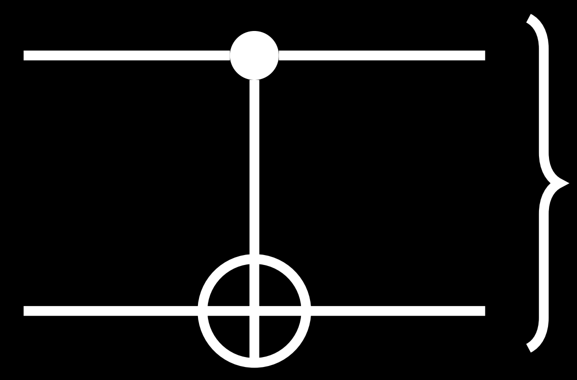
Stabilizer Codes: Logical Operators

Stabilizer Group: $S = \langle \cancel{X \otimes X}, Z \otimes Z \rangle \rightarrow \langle Z \otimes Z \rangle \equiv \langle ZZ \rangle$

$k = 1$
logical qubit

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

$|0\rangle$



$\overline{|\psi\rangle} = \alpha\,|00\rangle + \beta\,|11\rangle$

$n = 2$
physical qubits

Logical Pauli Operators

$$X\,|\psi\rangle = \alpha\,|1\rangle + \beta\,|0\rangle \longleftrightarrow XX\overline{|\psi\rangle} = \alpha\,|11\rangle + \beta\,|00\rangle \Rightarrow \boxed{\overline{X} = XX}$$

$$Z\,|\psi\rangle = \alpha\,|0\rangle - \beta\,|1\rangle \longleftrightarrow ZI\overline{|\psi\rangle} = \alpha\,|00\rangle - \beta\,|11\rangle \Rightarrow \boxed{\overline{Z} = ZI \equiv IZ}$$

$$Y = iXZ \Rightarrow \boxed{\overline{Y} = i\overline{XZ} = YX}$$

Synthesis of Logical Operators: Translate a <u>universal</u> set of quantum gates

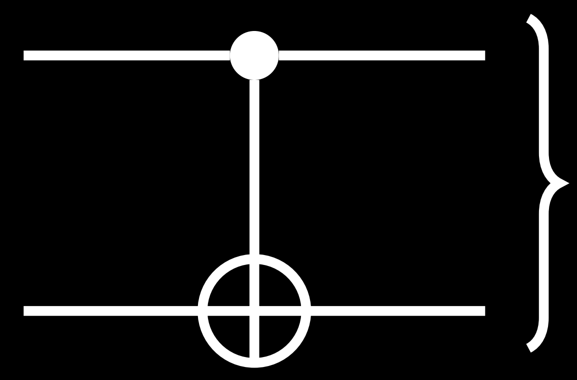
Stabilizer Codes: Error Correction

Stabilizer Group: $S = \langle Z \otimes Z \rangle \equiv \langle ZZ \rangle$; Logicals: $\bar{X} = XX, \bar{Z} = ZI \equiv IZ$

$k = 1$
logical qubit

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

$|0\rangle$



$\overline{|\psi\rangle} = \alpha\,|00\rangle + \beta\,|11\rangle$

$n = 2$
physical qubits

Recall: Pauli operators form an orthogonal basis for all square matrices
$\Rightarrow$ Suffices to consider only Pauli errors on the physical qubits*

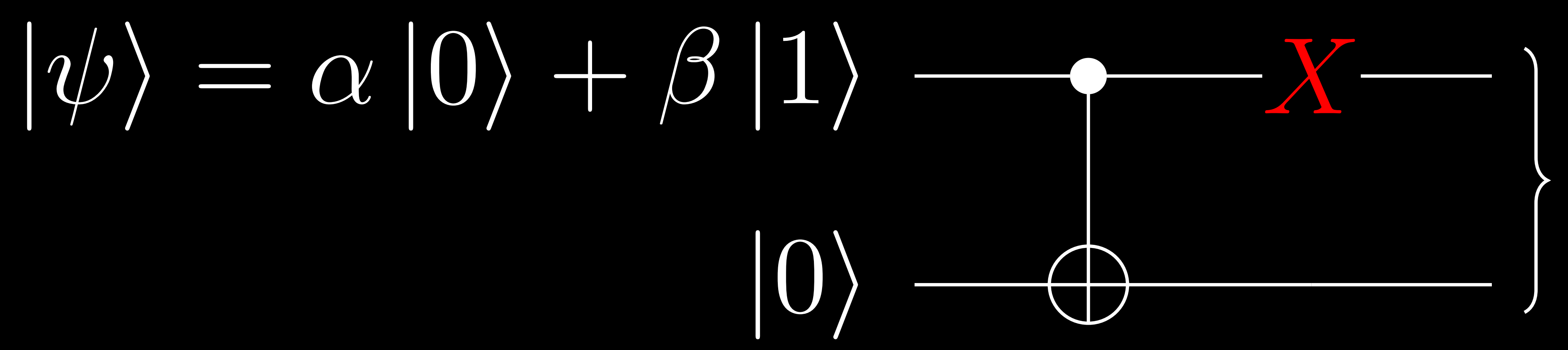
1. Error $E = XI$ occurs on the code state: $\overline{|\psi\rangle} \mapsto E\overline{|\psi\rangle} = \alpha|10\rangle + \beta|01\rangle$

2. Measure ZZ, i.e., the property that the state belongs to the code

3. Since "syndrome" is -1, decoder estimates error to be $\hat{E} = XI$ or $\widehat{E'} = IX = \hat{E}\bar{X}$

Stabilizer Codes: Error Correction

Stabilizer Group: $S = \langle Z \otimes Z \rangle \equiv \langle ZZ \rangle$; **Logicals:** $\bar{X} = XX, \bar{Z} = ZI \equiv IZ$



$$E\overline{|\psi\rangle} = \alpha|10\rangle + \beta|01\rangle$$

Measure Syndrome: Eigenvalues of $E\overline{|\psi\rangle}$ with respect to stabilizer(s)

$$(ZZ)E\overline{|\psi\rangle} = (ZZ)(XI)\overline{|\psi\rangle} = (-1)(XI)(ZZ)\overline{|\psi\rangle} = (-1)E\overline{|\psi\rangle}$$

Decoder: Takes syndrome as input and outputs most likely (Pauli) error $\hat{E}$

$$\hat{E} = XI : \hat{E}E\overline{|\psi\rangle} = (XI)(XI)\overline{|\psi\rangle} = \overline{|\psi\rangle} \qquad \hat{E} = IX : \hat{E}E\overline{|\psi\rangle} = (IX)(XI)\overline{|\psi\rangle} = \overline{X}\overline{|\psi\rangle}$$

Correct Decoding

Logical Error!!

Clearly, this is a trivial code that cannot store much information (i.e., logical qubits) or correct many errors

We need a systematic approach to design good quantum codes and efficient decoders to correct errors on them

There is a rich literature on classical codes and decoders

How can we establish a connection between quantum and classical codes to leverage that?

Bridge to Classical Codes

Map an n-qubit Hermitian Pauli matrix to a pair of binary vectors:

Example for $n = 3$: $X \otimes Z \otimes Y \longrightarrow E(\boldsymbol{a}, \boldsymbol{b})$

$$\boldsymbol{a} = [1 \quad 0 \quad 1] \quad (X \text{ component})$$

$$\boldsymbol{b} = [0 \quad 1 \quad 1] \quad (Z \text{ component})$$

How to check if $E(\boldsymbol{a}, \boldsymbol{b}) = X \otimes Z \otimes Y$ and $E(\boldsymbol{c}, \boldsymbol{d}) = Z \otimes Z \otimes X$ commute?

Compare operators on each qubit:

$$X \otimes Z \otimes Y \mapsto ([1\ 0\ 1], [0\ 1\ 1])$$

$$Z \otimes Z \otimes X \mapsto ([0\ 0\ 1], [1\ 1\ 0])$$

Commutativity: Symplectic Inner Product

How to check if $E(\boldsymbol{a},\boldsymbol{b}) = X \otimes Z \otimes Y$ and $E(\boldsymbol{c},\boldsymbol{d}) = Z \otimes Z \otimes X$ commute?

Compare operators on each qubit:

$$X \otimes Z \otimes Y \mapsto (\underbrace{[1\ 0\ 1]}_{\boldsymbol{a}}, \underbrace{[0\ 1\ 1]}_{\boldsymbol{b}})$$

$$Z \otimes Z \otimes X \mapsto (\underbrace{[0\ 0\ 1]}_{\boldsymbol{c}}, \underbrace{[1\ 1\ 0]}_{\boldsymbol{d}})$$

Symplectic inner product: $\langle [\boldsymbol{a},\boldsymbol{b}], [\boldsymbol{c},\boldsymbol{d}] \rangle_{\text{sym}} \coloneqq \boldsymbol{a}\boldsymbol{d}^T + \boldsymbol{b}\boldsymbol{c}^T$ (modulo 2)

$$= \begin{cases} 0 & \text{iff they commute,} \\ 1 & \text{iff they anticommute} \end{cases}$$

Commutativity: Symplectic Inner Product

How to check if $E(\boldsymbol{a},\boldsymbol{b}) = X \otimes Z \otimes Y$ and $E(\boldsymbol{c},\boldsymbol{d}) = Z \otimes Z \otimes X$ commute?

Compare operators on each qubit:

$$X \otimes Z \otimes Y \mapsto (\underbrace{[1\ 0\ 1]}_{\boldsymbol{a}}, \underbrace{[0\ 1\ 1]}_{\boldsymbol{b}})$$

$$Z \otimes Z \otimes X \mapsto (\underbrace{[0\ 0\ 1]}_{\boldsymbol{c}}, \underbrace{[1\ 1\ 0]}_{\boldsymbol{d}})$$

$$\begin{aligned} \boldsymbol{a}\boldsymbol{d}^T + \boldsymbol{b}\boldsymbol{c}^T &= [1 \quad 0 \quad 1]\begin{bmatrix}1\\1\\0\end{bmatrix} + [0 \quad 1 \quad 1]\begin{bmatrix}0\\0\\1\end{bmatrix} \\ &= 1 + 1 \\ &= 0 \text{ (modulo 2)} \Rightarrow \text{they commute} \end{aligned}$$

Quantum Parity-Check Matrices

Stabilizers ($n = 3$): $Z \otimes Z \otimes I$ $\quad$ $I \otimes Z \otimes Z$

(X component) $\boldsymbol{a_1} = [0 \quad 0 \quad 0]$ $\quad$ $\boldsymbol{a_2} = [0 \quad 0 \quad 0]$

(Z component) $\boldsymbol{b_1} = [1 \quad 1 \quad 0]$ $\quad$ $\boldsymbol{b_2} = [0 \quad 1 \quad 1]$

$$H_S = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right] = [H_a \mid H_b] \; ; \; \boldsymbol{H_a H_b^T + H_b H_a^T = 0}$$

Recall: Syndrome is the commutation relations of error $E(\boldsymbol{c}, \boldsymbol{d})$ with stabilizers

$$(ZZ)E\overline{|\psi\rangle} = (ZZ)(XI)\overline{|\psi\rangle} = (-1)(XI)(ZZ)\overline{|\psi\rangle} = (-1)E\overline{|\psi\rangle}$$

$$\text{Syndrome} = \begin{bmatrix} \langle [\boldsymbol{a_1}, \boldsymbol{b_1}], [\boldsymbol{c}, \boldsymbol{d}] \rangle_{\text{sym}} \\ \langle [\boldsymbol{a}_2, \boldsymbol{b_2}], [\boldsymbol{c}, \boldsymbol{d}] \rangle_{\text{sym}} \end{bmatrix} = H_a \boldsymbol{d}^T + H_b \boldsymbol{c}^T \text{ (modulo 2)}$$

Bell, GHZ States → 2- and 3-Qubit Codes

$$|\Phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \quad , \quad |\Phi^-\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}$$

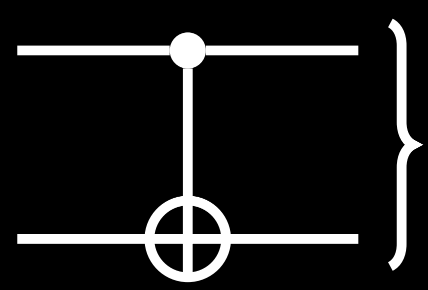
Stabilizers: $S = \langle \cancel{XX}, ZZ \rangle \rightarrow \langle ZZ \rangle$

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \xrightarrow{\text{Encoding}} \overline{|\psi\rangle} = \alpha\,|\Phi^+\rangle + \beta\,|\Phi^-\rangle$$
$$\equiv \alpha\,|00\rangle + \beta\,|11\rangle$$

Encoding

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

$|0\rangle$



$\overline{|\psi\rangle} = \alpha\,|00\rangle + \beta\,|11\rangle$

Logicals: $\bar{X} = XX, \bar{Z} = ZI \equiv IZ$

$$|\Phi_3^+\rangle = \frac{|000\rangle + |111\rangle}{\sqrt{2}} \quad , \quad |\Phi_3^-\rangle = \frac{|000\rangle - |111\rangle}{\sqrt{2}}$$

$S = \langle \cancel{XXX}, ZZI, IZZ \rangle \rightarrow \langle ZZI, IZZ \rangle$

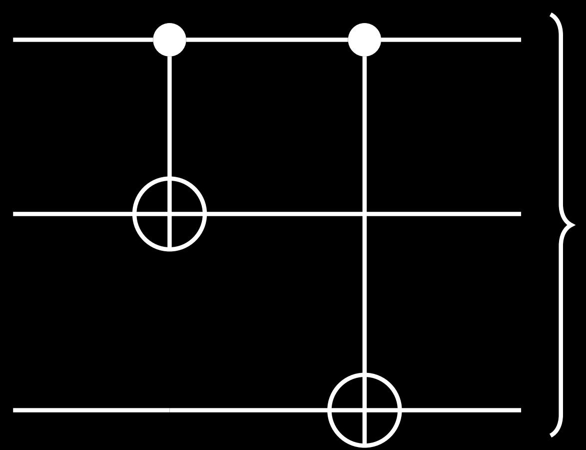
$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \xrightarrow{\text{Encoding}} \overline{|\psi\rangle} = \alpha\,|\Phi_3^+\rangle + \beta\,|\Phi_3^-\rangle$$
$$\equiv \alpha\,|000\rangle + \beta\,|111\rangle$$

Encoding

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

$|0\rangle$

$|0\rangle$

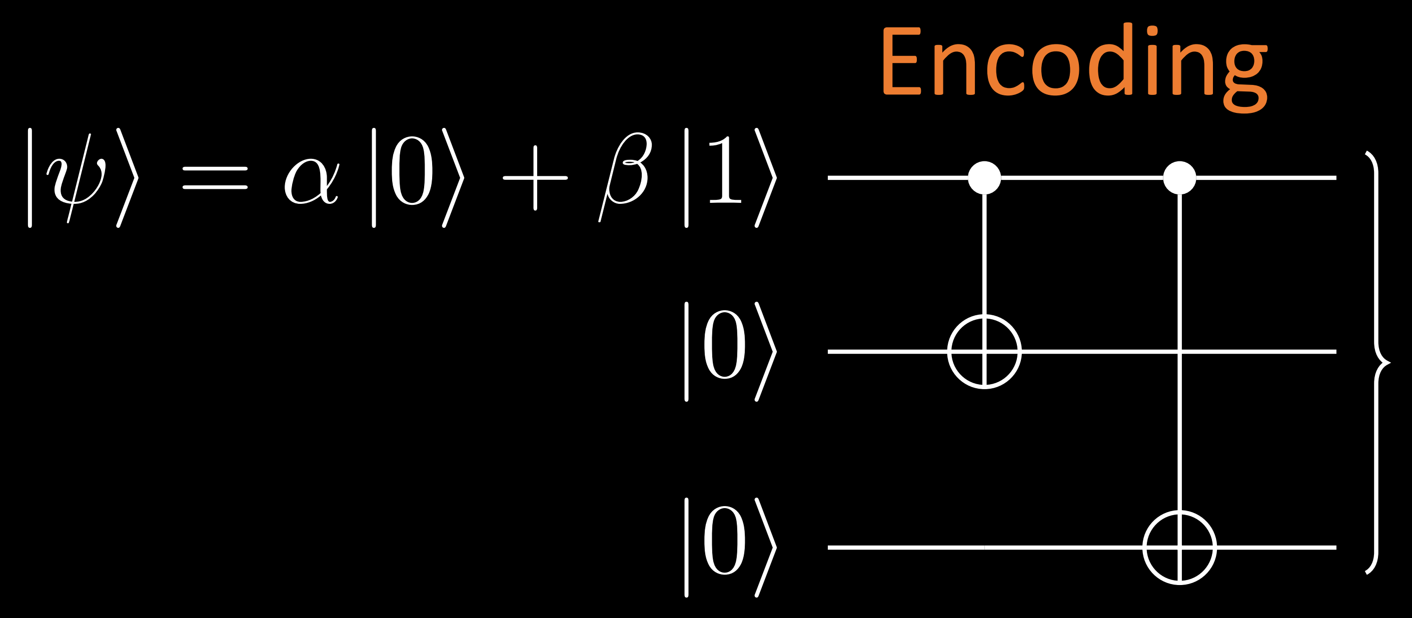


$\overline{|\psi\rangle} = \alpha\,|000\rangle + \beta\,|111\rangle$

$\bar{X} = XXX, \bar{Z} = ZII \equiv IZI \equiv IIZ$

3-Qubit Code ≠ Quantum Repetition Code

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \xrightarrow{\text{Encoding}} \overline{|\psi\rangle} = \alpha|\Phi_3^+\rangle + \beta|\Phi_3^-\rangle$$
$$\equiv \alpha|000\rangle + \beta|111\rangle$$



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

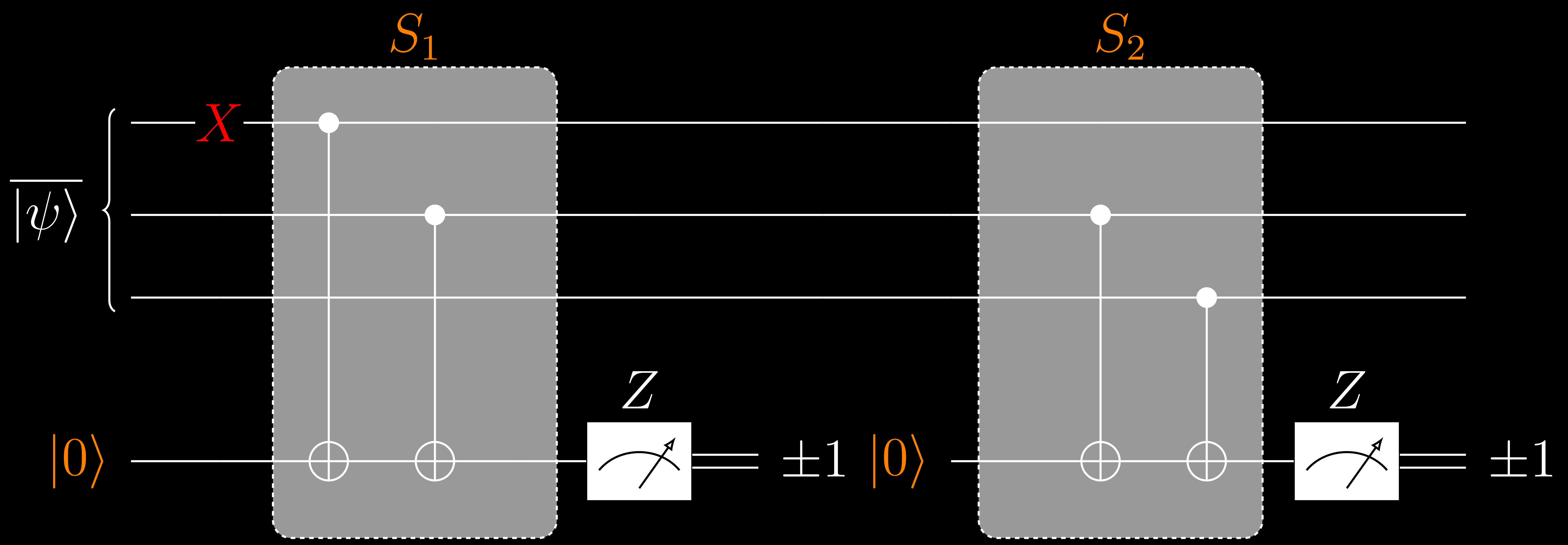
This is NOT the same as cloning (i.e., copying) the logical qubit into 3 physical qubits!!

Recall Classical Repetition Code: $C = \{000, 111\}$

Quantum “No Cloning” Theorem: Arbitrary quantum state can’t be cloned!

Syndrome Measurement for 3-Qubit Code

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

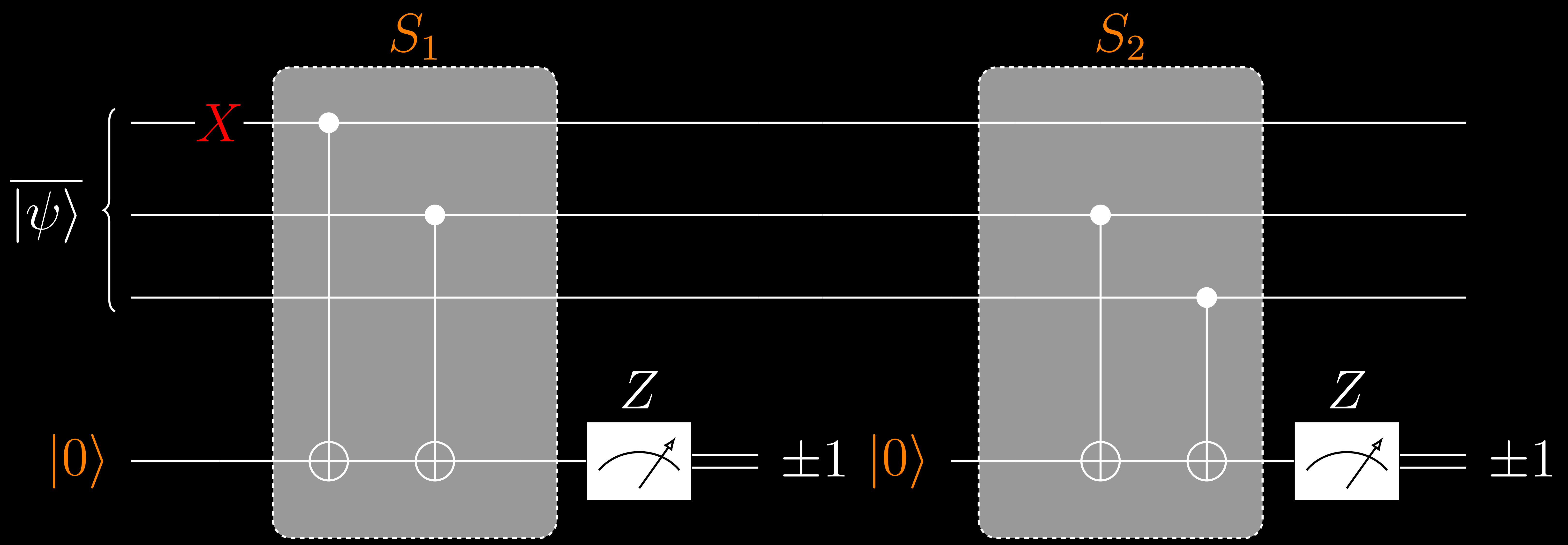


The error X propagates through the CNOT and flips the measurement

Hence, the measurement results in -1 whenever there are an odd number of X's on the ancilla (through the CNOT gates), i.e., **when the error anticommutes with the stabilizer S_i (syndrome!)**

Syndrome Measurement for 3-Qubit Code

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



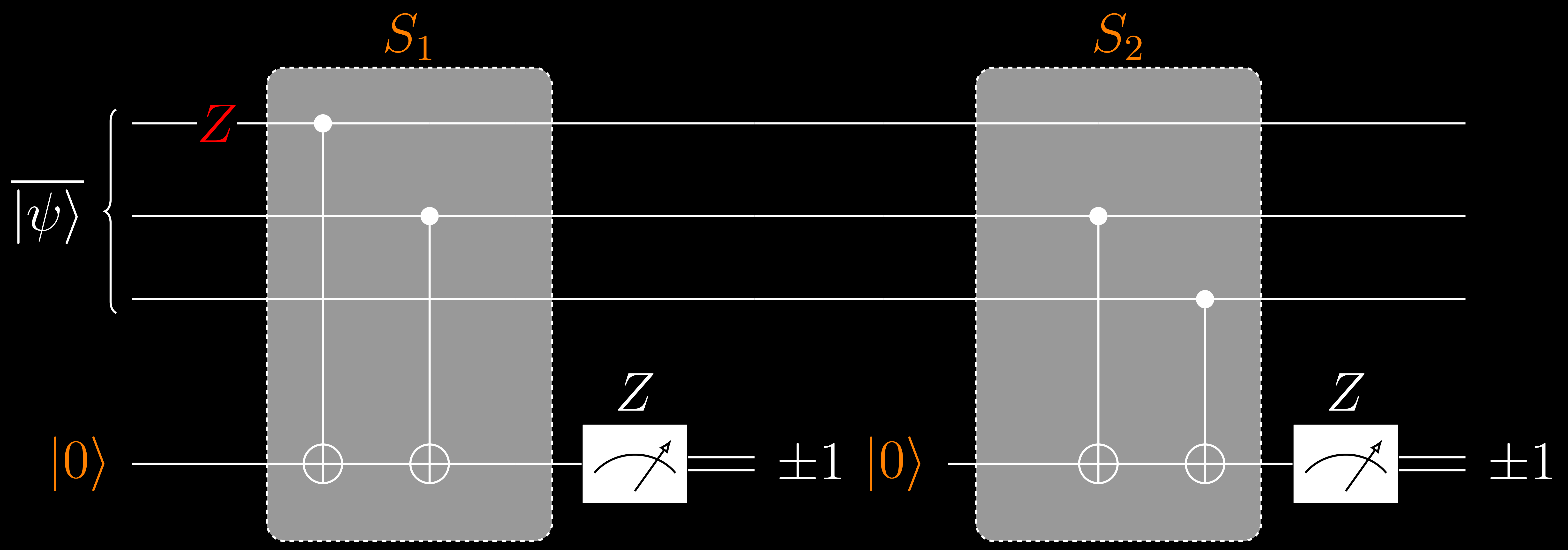
The error X propagates through the CNOT and flips the measurement

$$H_S = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right] = [H_a \mid H_b] \;;\; \text{Syndrome } = H_a \mathbf{d}^T + H_b \mathbf{c}^T$$

$$E(\mathbf{c}, \mathbf{d}) = X_1 = XII$$

Minimum Distance of 3-Qubit Code

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



The error Z DOES NOT propagate through the CNOT, so doesn't flip the measurement

Minimum Distance of a Code: The minimum weight of an undetectable error $=$ The minimum weight of a logical operator

Logicals: $\bar{X} = XXX, \bar{Z} = ZII \equiv IZI \equiv IIZ \Rightarrow$ Minimum Distance $d = 1$

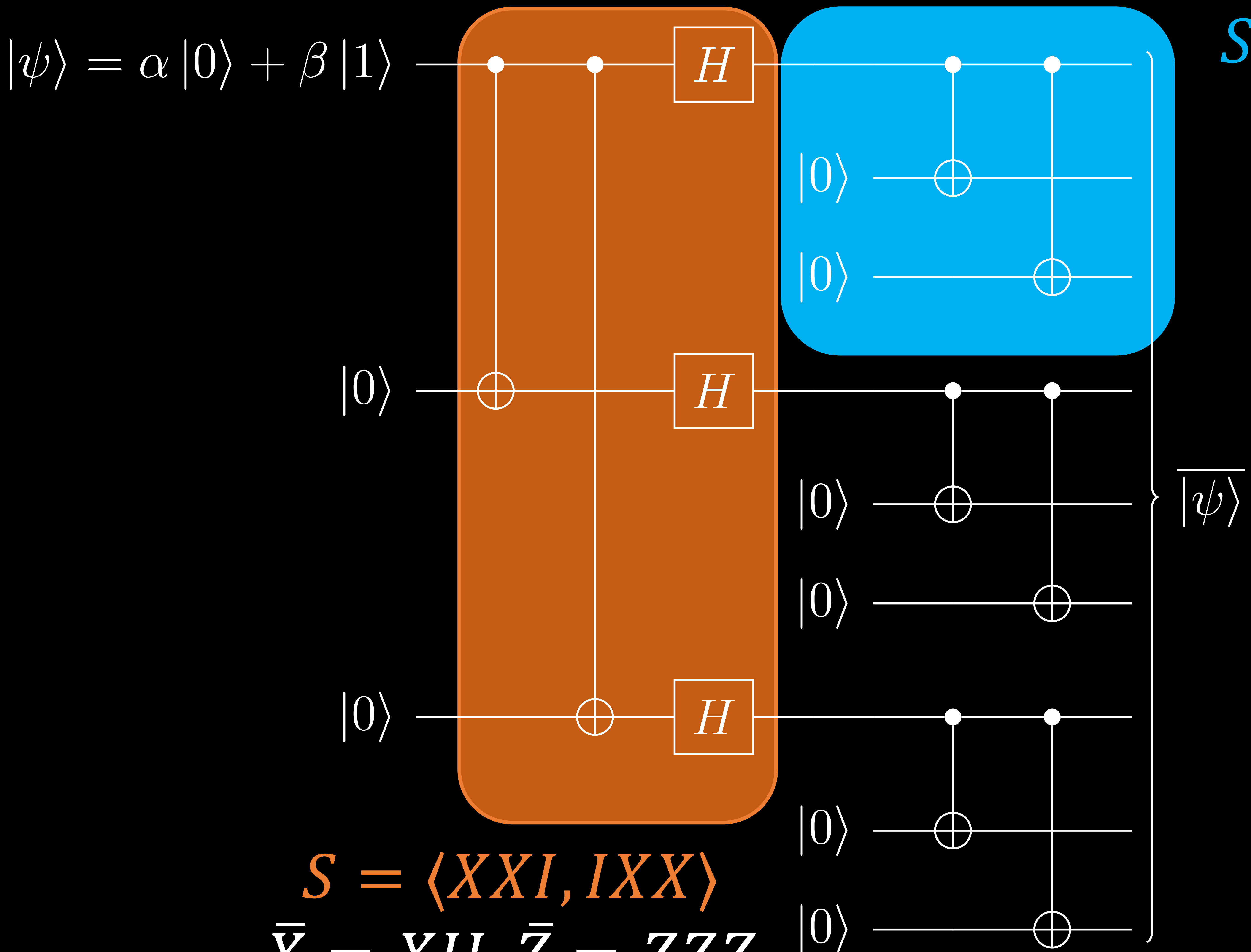
Clearly, this code can only correct 1 error of X-type and cannot even detect 1 error of Z-type

How can we construct a non-trivial code out of this 3-qubit code that can correct 1 error of any type?

Concatenate the 3-qubit code with its X-type version!

This leads to the famous 9-qubit code of Peter Shor, called the "Shor code", which was the first time QEC was shown to work!

Shor's 9-Qubit Code



$S = \langle XXI, IXX\rangle$
$\bar{X} = XII, \bar{Z} = ZZZ$

$S = \langle ZZI, IZZ\rangle, \bar{X} = XXX, \bar{Z} = ZII$

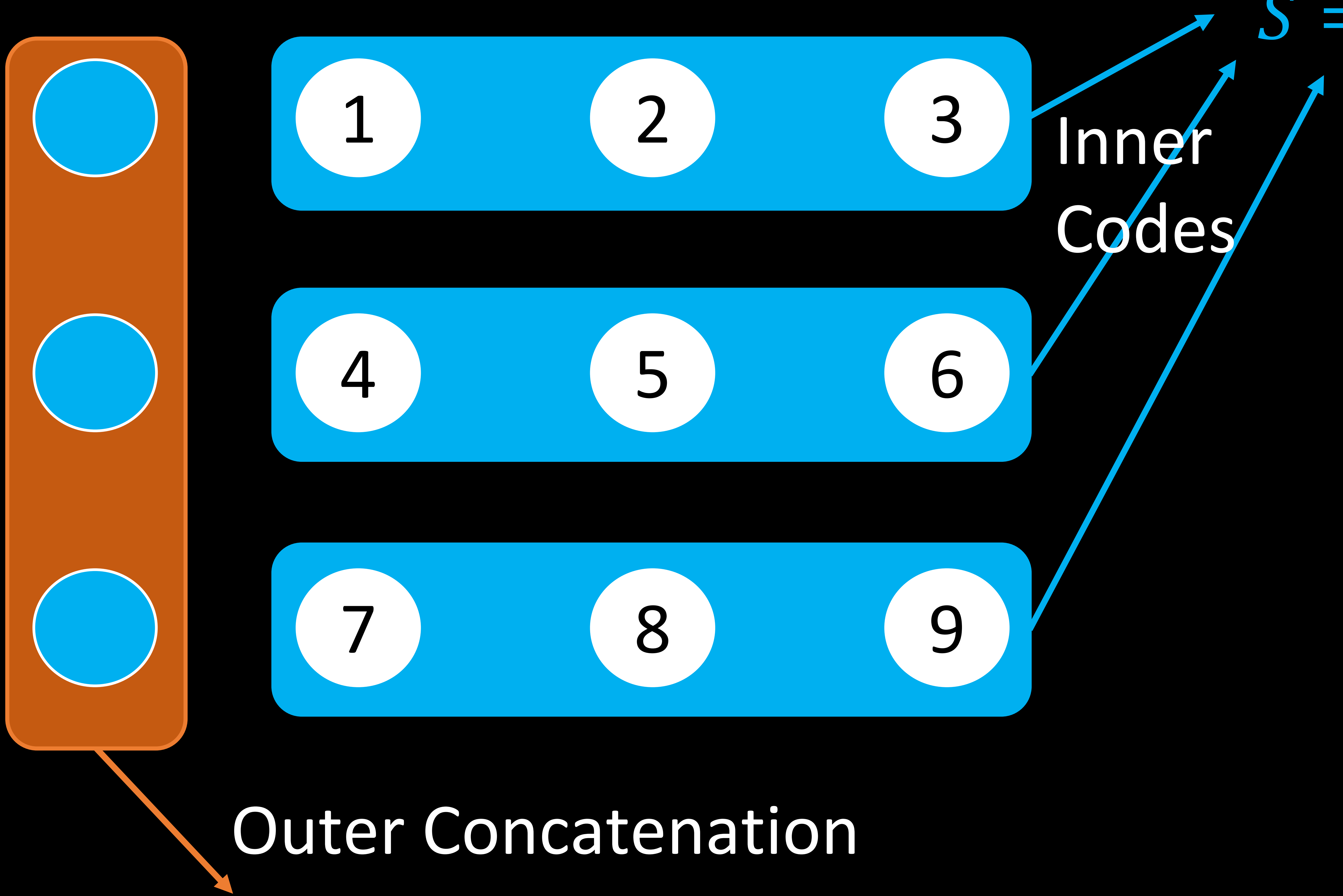
Each of the 3 physical qubits of the initial 3-qubit "phase-flip" code is further encoded into the 3-qubit "bit-flip" code

This is a $[[n = 9, k = 1, d = 3]]$ code since the minimum weight of a logical operator is $d = 3$

There are 8 stabilizer generators, 6 of Z-type and 2 of X-type

Shor's 9-Qubit Code

$S = \langle ZZI, IZZ \rangle, \bar{X} = XXX, \bar{Z} = ZII$



$S = \langle XXI, IXX \rangle$
$\bar{X} = XII, \bar{Z} = ZZZ$

Each of the 3 physical qubits of the initial 3-qubit "phase-flip" code is further encoded into the 3-qubit "bit-flip" code

This is a $[[n = 9, k = 1, d = 3]]$ code since the minimum weight of a logical operator is $d = 3$

There are 8 stabilizer generators, 6 of Z-type and 2 of X-type

Steane's 7-Qubit Code

[7,4,3] Hamming Code:

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

[[7,1,3]] Steane Code:

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

Commutativity Check:

$$H_a H_b^T + H_b H_a^T = \left[\begin{array}{c|c} 0 & H H^T \\ \hline 0 & 0 \end{array}\right] = 0$$

Logical Pauli Operators:

$$\bar{X} = \left[\begin{array}{ccccccc|ccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

$$\bar{Z} = \left[\begin{array}{ccccccc|ccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right]$$

Steane's 7-Qubit Code

[7,4,3] Hamming Code:

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ \boxed{0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1} \end{bmatrix} \Rightarrow X_3X_5X_6X_7, Z_3Z_5Z_6Z_7$$

[[7,1,3]] Steane Code:

$$H_S = \left[\, H_a \,\middle|\, H_b \,\right] = \begin{array}{c} \left[\begin{array}{c|c} H & 0 \\ \hline 0 & H \end{array}\right] \end{array} \begin{array}{c} X \\ Z \end{array}$$

Logical Pauli Operators:

$$\bar{X} = \left[\begin{array}{ccccccc|ccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

$$\bar{Z} = \left[\begin{array}{ccccccc|ccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right]$$

Since stabilizers fix code state, $\bar{X} \equiv (XXXXXXX)(X_3X_5X_6X_7) = X_1X_2X_4$

$$Z \equiv (ZZZZZZZ)(Z_3Z_5Z_6Z_7) = Z_1Z_2Z_4$$

Decoding Steane's 7-Qubit Code

[[7,1,3]] Steane Code: $H_S = \left[\begin{array}{c|c} H_a & H_b \end{array}\right] = \left[\begin{array}{c|c} H & 0 \\ \hline 0 & H \end{array}\right] \begin{array}{c} X \\ Z \end{array}$

Error $E(\boldsymbol{c}, \boldsymbol{d})$: $\text{Syndrome} = H_a \boldsymbol{d}^T + H_b \boldsymbol{c}^T = \begin{bmatrix} H \\ 0 \end{bmatrix} \boldsymbol{d}^T + \begin{bmatrix} 0 \\ H \end{bmatrix} \boldsymbol{c}^T = \begin{bmatrix} H\boldsymbol{d}^T \\ H\boldsymbol{c}^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_X \\ \boldsymbol{s}_Z \end{bmatrix}$

Decoding: Use the X-syndrome $\boldsymbol{s}_X$ (Z-syndrome $\boldsymbol{s}_Z$) to estimate Z-error (X-error)

This reduces to decoding the Hamming code for either type of error!

If X- and Z-errors are correlated, then decoders aren't independent

Recall Standard Array Decoding

[[7,1,3]] Steane Code: $$H_S = \left[\, H_a \mid H_b \,\right] = \left[\begin{array}{c|c} H & 0 \\ \hline 0 & H \end{array}\right] \begin{array}{c} X \\ Z \end{array}$$

0000000	1110000	1001100	0111100	0101010	1011010	1100110	0010110	1101001	0011001	0100101	1010101	1000011	0110011	0001111	1111111	000
1000000	0110000	0001100	1111100	1101010	0011010	0100110	1010110	0101001	1011001	1100101	0010101	0000011	1110011	1001111	0111111	111
0100000	1010000	1101100	0011100	0001010	1111010	1000110	0110110	1001001	0111001	0000101	1110101	1100011	0010011	0101111	1011111	101
0010000	1100000	1011100	0101100	0111010	1001010	1110110	0000110	1111001	0001001	0110101	1000101	1010011	0100011	0011111	1101111	110
0001000	1111000	1000100	0110100	0100010	1010010	1101110	0011110	1100001	0010001	0101101	1011101	1001011	0111011	0000111	1110111	011
0000100	1110100	1001000	0111000	0101110	1011110	1100010	0010010	1101101	0011101	0100001	1010001	1000111	0110111	0001011	1111011	100
0000010	1110010	1001110	0111110	0101000	1011000	1100100	0010100	1101011	0011011	0100111	1010111	1000001	0110001	0001101	1111101	010
0000001	1110001	1001101	0111101	0101011	1011011	1100111	0010111	1101000	0011000	0100100	1010100	1000010	0110010	0001110	1111110	001

Calderbank-Shor-Steane (CSS) Codes

[[7,1,3]] Steane Code

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

$$HH^T = 0$$

$$\text{Syndrome} = \begin{bmatrix} H\boldsymbol{d}^T \\ H\boldsymbol{c}^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_X \\ \boldsymbol{s}_Z \end{bmatrix}$$

$$\overline{X} = E(\boldsymbol{c}, \boldsymbol{0}) \ ; \ H\boldsymbol{c}^T = \boldsymbol{0}$$

$$\overline{Z} = E(\boldsymbol{0}, \boldsymbol{d}) \ ; \ H\boldsymbol{d}^T = \boldsymbol{0}$$

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

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

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

$$H_X H_Z^T = 0$$

$$\text{Syndrome} = \begin{bmatrix} H_X\boldsymbol{d}^T \\ H_Z\boldsymbol{c}^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_X \\ \boldsymbol{s}_Z \end{bmatrix}$$

$$\overline{X} = E(\boldsymbol{c}, \boldsymbol{0}) \ ; \ H_Z\boldsymbol{c}^T = \boldsymbol{0}$$

$$\overline{Z} = E(\boldsymbol{0}, \boldsymbol{d}) \ ; \ H_X\boldsymbol{d}^T = \boldsymbol{0}$$

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

CSS Quantum LDPC (QLDPC) 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

$$H_S = \begin{bmatrix} H_X & 0 \\ 0 & H_Z \end{bmatrix} \begin{matrix} X \\ Z \end{matrix}$$

$$H_X H_Z^T = 0$$

$$\text{Syndrome} = \begin{bmatrix} H_X \boldsymbol{d}^T \\ H_Z \boldsymbol{c}^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_X \\ \boldsymbol{s}_Z \end{bmatrix}$$

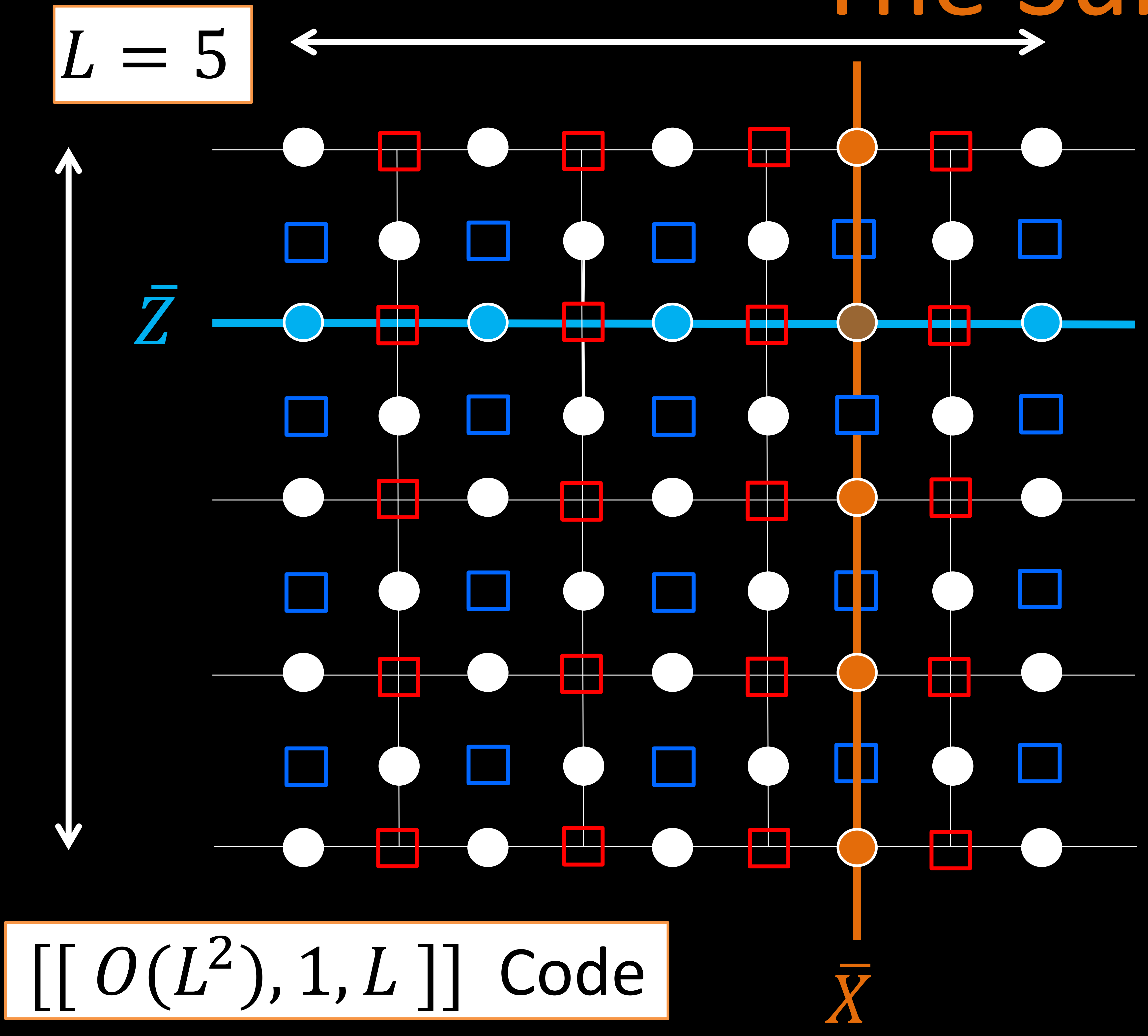
$$\overline{X} = E(\boldsymbol{c}, \boldsymbol{0}) \; ; \; H_Z \boldsymbol{c}^T = \boldsymbol{0}$$

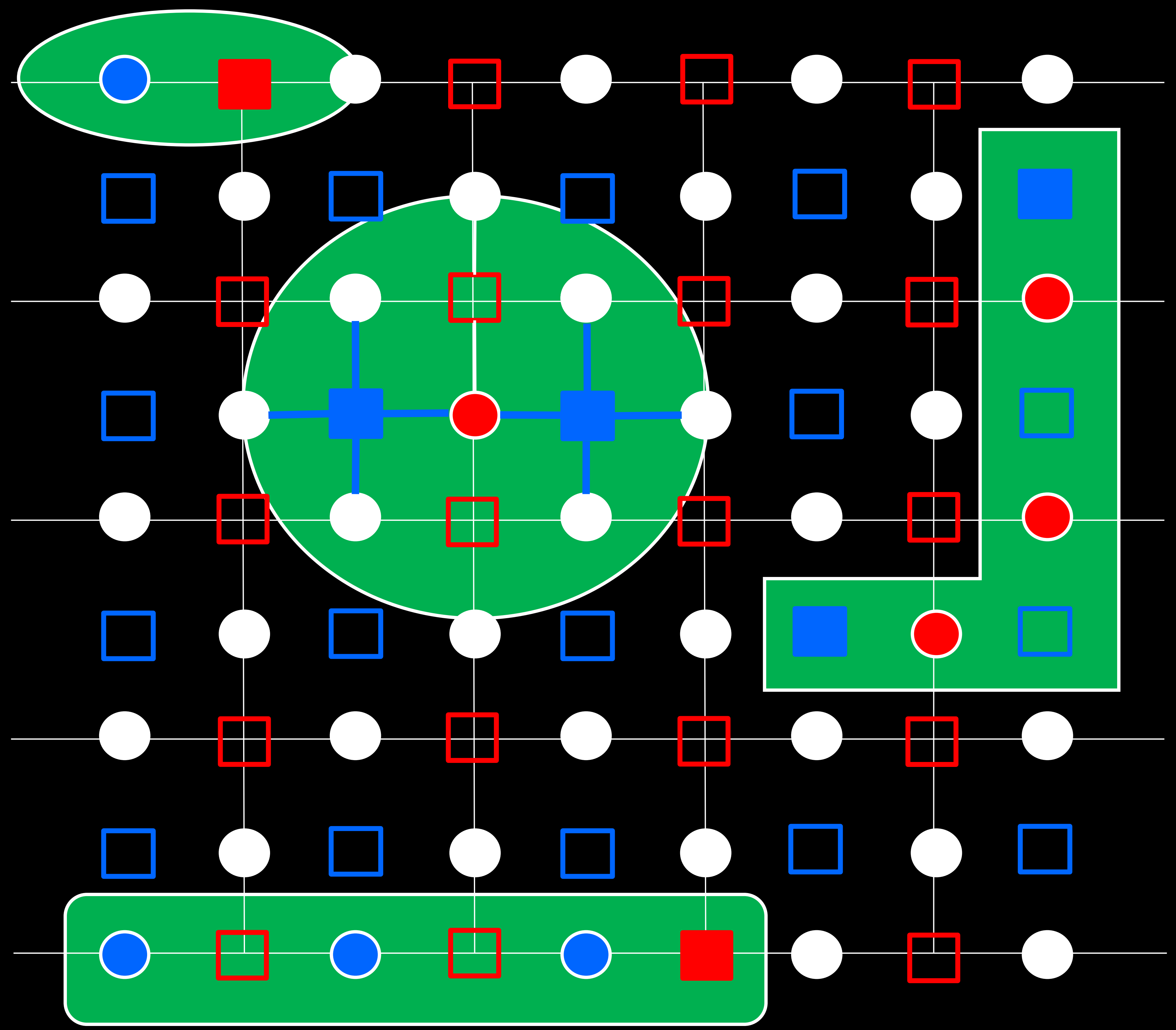
$$\overline{Z} = E(\boldsymbol{0}, \boldsymbol{d}) \; ; \; H_X \boldsymbol{d}^T = \boldsymbol{0}$$

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

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

The Surface Code



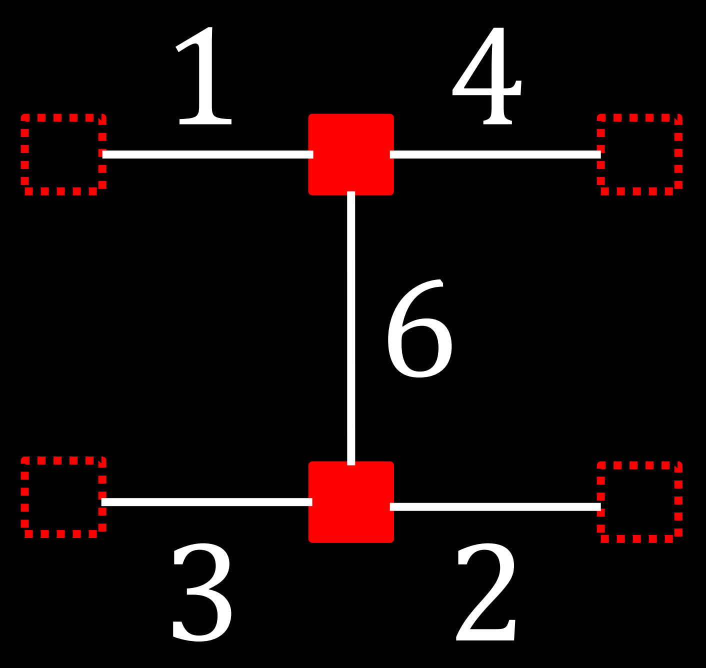


Errors cause syndrome at endpoints

□ - vertex checks (H_X) □ - plaquette checks (H_Z)

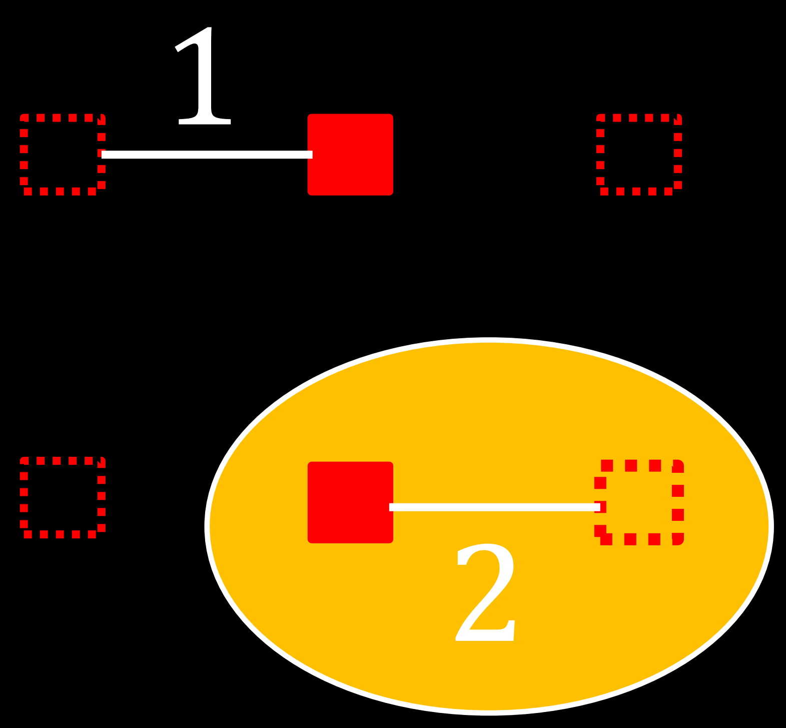
Minimum Weight Perfect Matching

X-Syndrome Graph

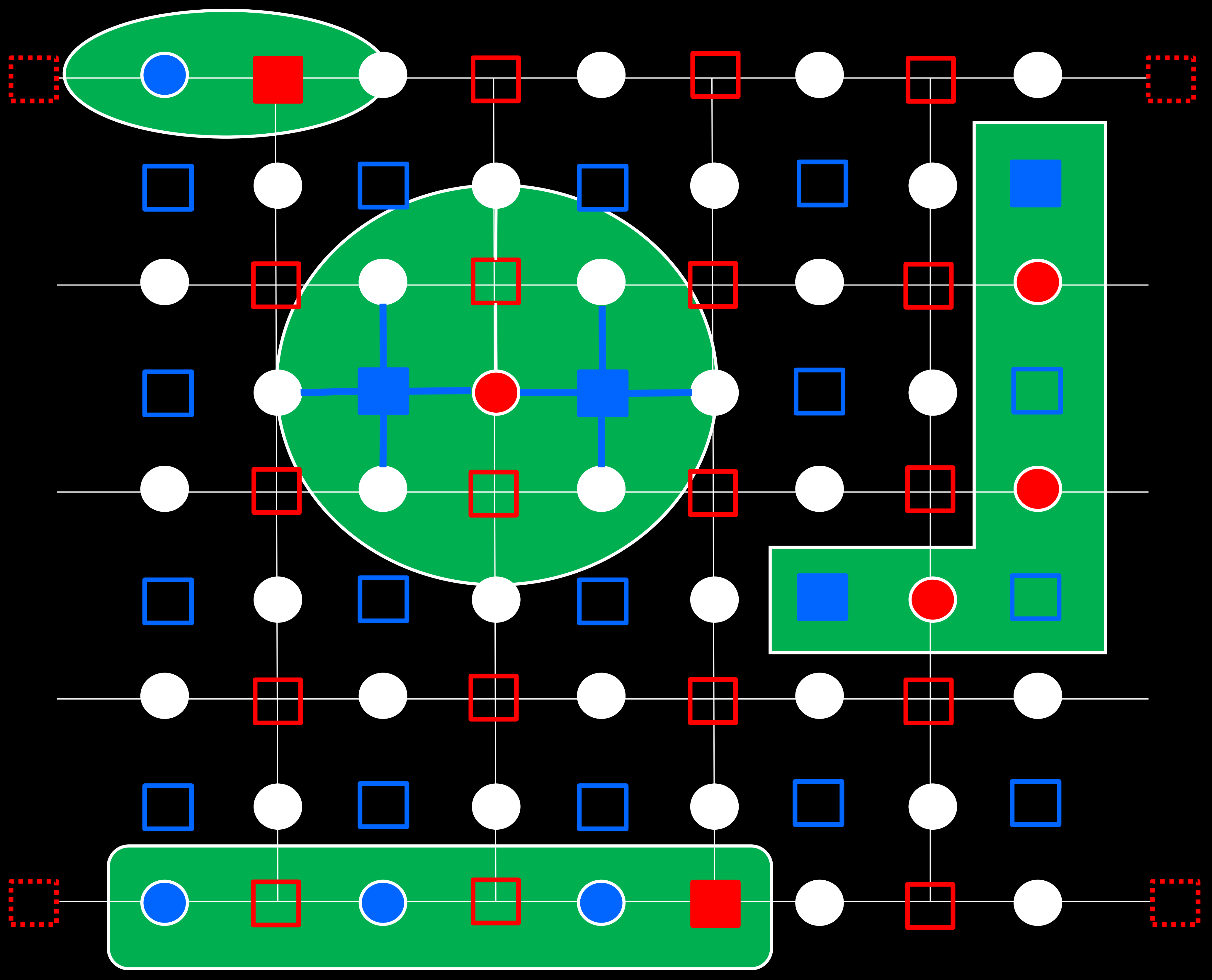


MWPM

Matched Graph



Causes Logical Error!!

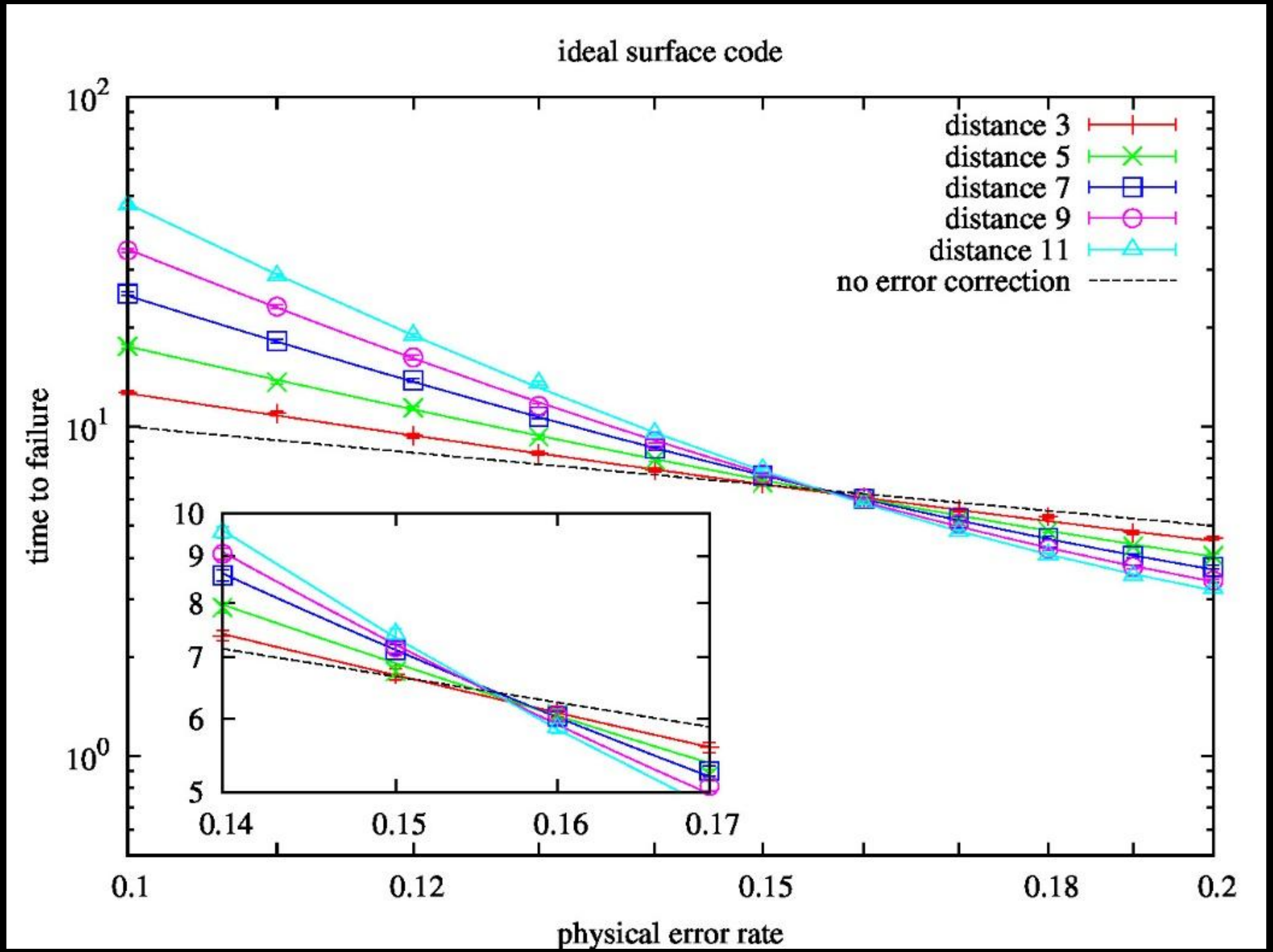


Virtual Check for Matching

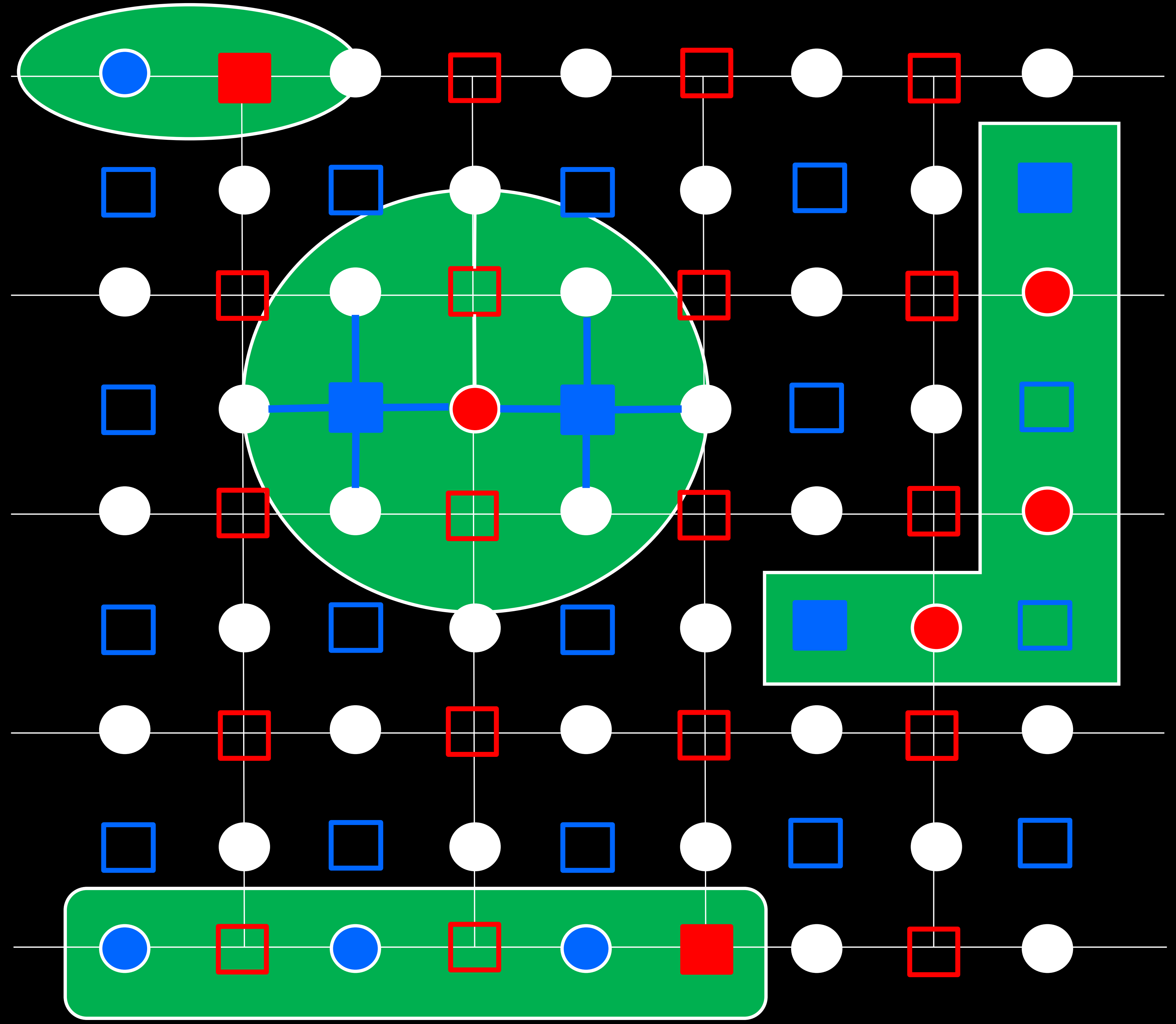
Errors cause syndrome at endpoints

- vertex checks (H_X) - plaquette checks (H_Z)

MWPM Decoding Threshold



Wang et al. http://arxiv.org/abs/0905.0531



Errors cause syndrome at endpoints

□ - vertex checks (H_X) □ - plaquette checks (H_Z)

Hypergraph Product (HP) QLDPC Codes

$[[N, K, d]]$ HP Code

$$C_a : [n_a, k_a, d_a] \to H_a : m_a \times n_a$$
$$C_b : [n_b, k_b, d_b] \to H_b : m_b \times n_b$$

$$H_X = \left[\boxed{H_a \otimes I_{m_b}} \;\; \boxed{I_{m_a} \otimes H_b} \right]$$
$$H_Z = \left[\boxed{I_{n_a} \otimes H_b^T} \;\; \boxed{H_a^T \otimes I_{n_b}} \right]$$

$$N = \boxed{n_a m_b} + \boxed{n_b m_a}$$
$$K = 2k_a k_b - k_a(n_b - m_b) - k_b(n_a - m_a) \sim \Theta(N)$$
$$d \geq \min(d_a, d_b, d_a^T, d_b^T) \sim \Theta(\sqrt{N})$$

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

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

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

$$H_X H_Z^T = 0$$

$$\text{Syndrome} = \begin{bmatrix} H_X \boldsymbol{d}^T \\ H_Z \boldsymbol{c}^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_X \\ \boldsymbol{s}_Z \end{bmatrix}$$

$$\overline{X} = E(\boldsymbol{c}, \boldsymbol{0}) \; ; \; H_Z \boldsymbol{c}^T = \boldsymbol{0}$$
$$\overline{Z} = E(\boldsymbol{0}, \boldsymbol{d}) \; ; \; H_X \boldsymbol{d}^T = \boldsymbol{0}$$

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

Hypergraph Product (HP) QLDPC Codes

$[[N, K, d]]$ HP Code

$$C_a : [5,1,5] \rightarrow H_a : m_a \times n_a = 4 \times 5$$

$$C_b : [4,0,0] \rightarrow H_b = H_a^T : m_b \times n_b = 5 \times 4$$

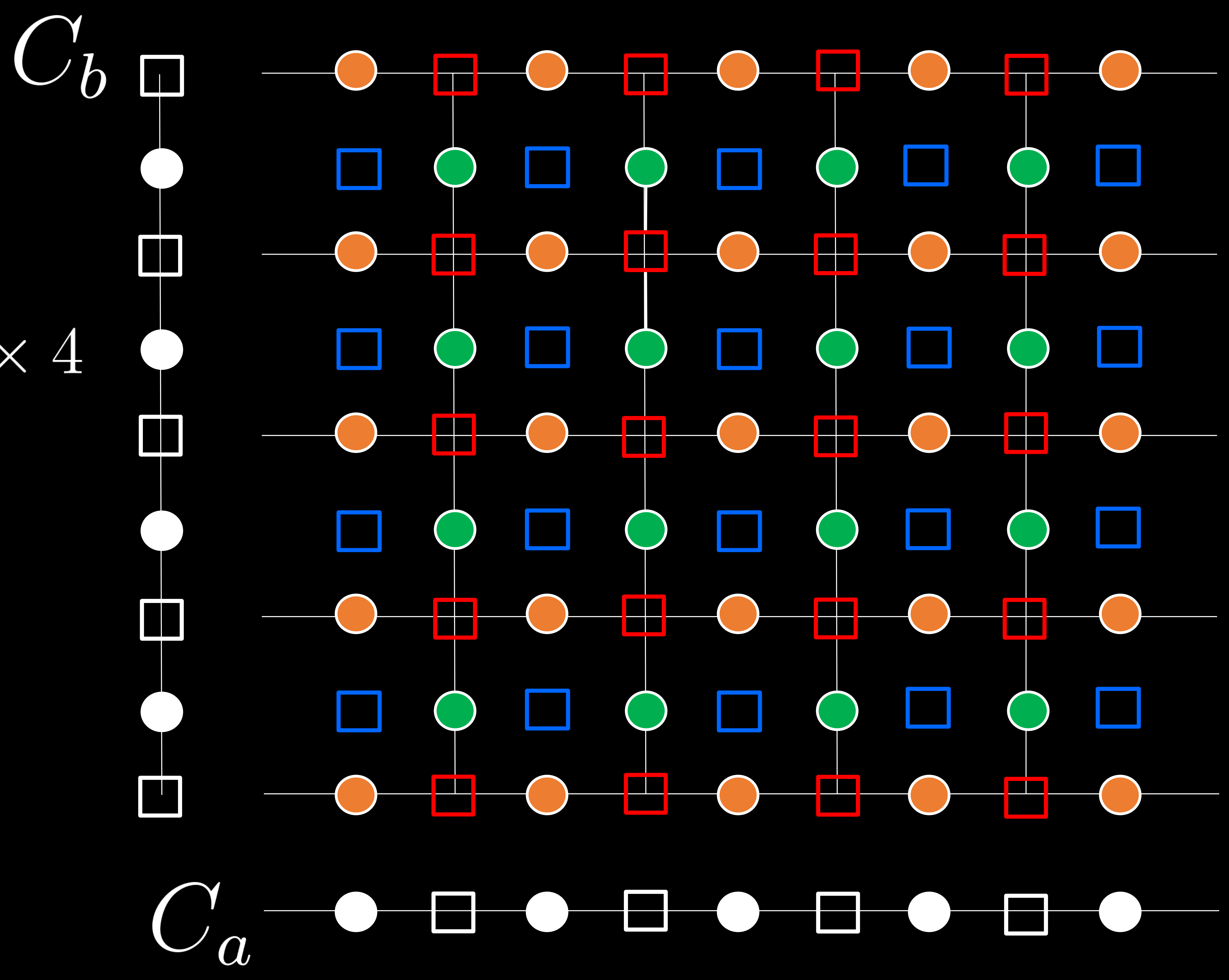
$$H_X = \left[H_a \otimes I_{m_b} \quad I_{m_a} \otimes H_b \right]$$

$$H_Z = \left[I_{n_a} \otimes H_b^T \quad H_a^T \otimes I_{n_b} \right]$$

$$N = n_a m_b + n_b m_a$$

$$K = \cancel{2k_a k_b} - k_a(n_b - m_b) - \cancel{k_b(n_a - m_a)} = 1$$

$$d = 5$$



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

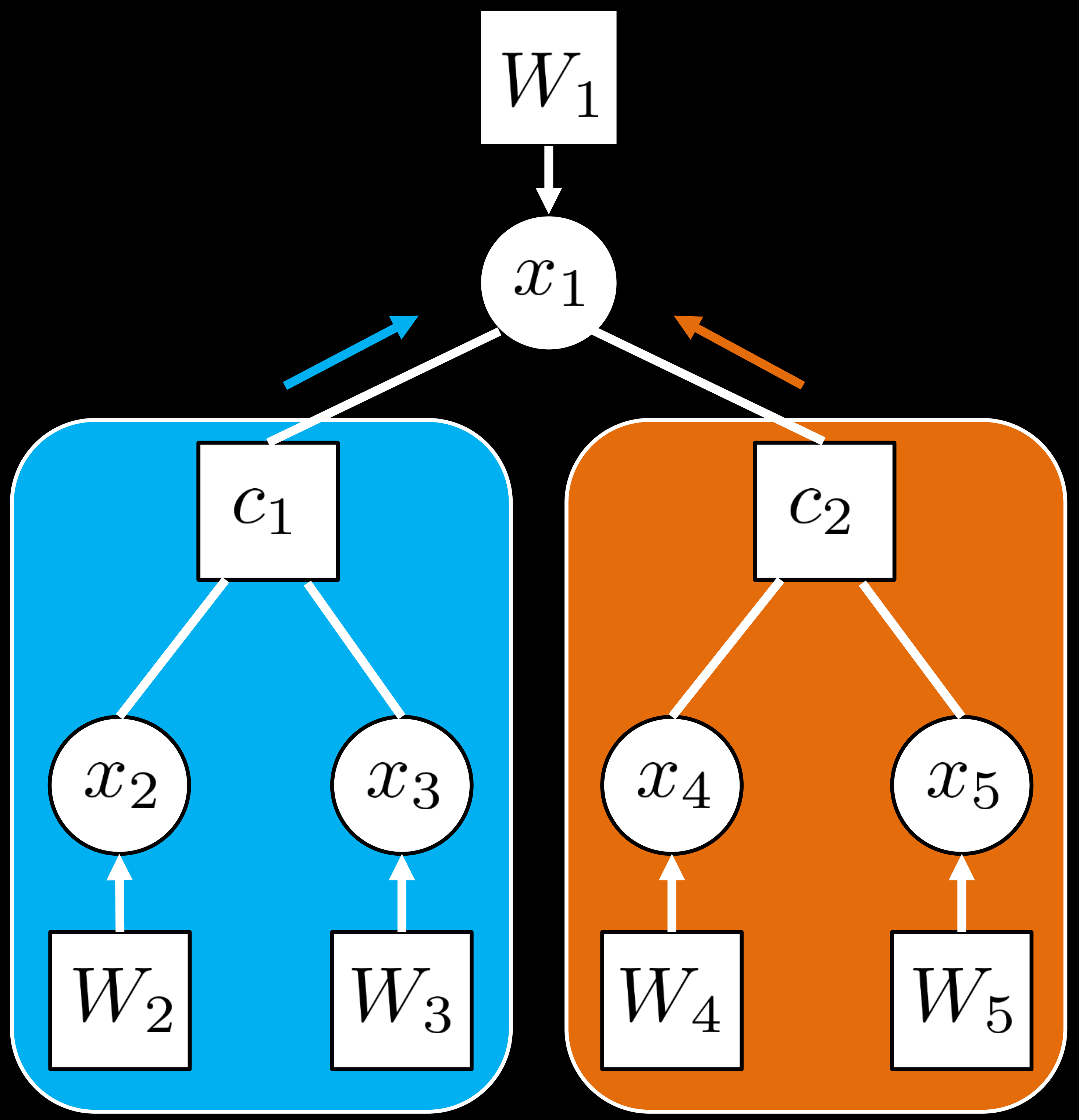
$$d = \text{minimum weight of } \bar{X}, \bar{Z}$$

Belief propagation (or sum-product)

$$\hat{x}_1 = \underset{x_1\in\{0,1\}}{\text{argmax}}\ P(x_1|y) = \underset{x_1\in\{0,1\}}{\text{argmax}} \sum_{x_2,x_3,x_4,x_5\in\{0,1\}^4} P(x|y)$$

$$= \underset{x_1\in\{0,1\}}{\text{argmax}} \sum_{x_2,x_3,x_4,x_5\in\{0,1\}^4} P(y|x)P(x)$$

$$= \underset{x_1\in\{0,1\}}{\text{argmax}} \sum_{x_2,x_3,x_4,x_5\in\{0,1\}^4} \left(\prod_{j=1}^{5} W_j\right) \mathbb{I}(c_1=0)\mathbb{I}(c_2=0)$$



$$W_j = P(y_j|x_j)$$

$$= \underset{x_1\in\{0,1\}}{\text{argmax}}\ W_1 \cdot \left(\sum_{x_2,x_3\in\{0,1\}^2} \mathbb{I}(c_1=0)\cdot W_2 \cdot W_3\right) \left(\sum_{x_4,x_5\in\{0,1\}^2} \mathbb{I}(c_2=0)\cdot W_4 \cdot W_5\right)$$

$c_1 :\ x_1 + x_2 + x_3 = 0$

$c_2 :\ x_1 + x_4 + x_5 = 0$

Iterative Decoding of QLDPC 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

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

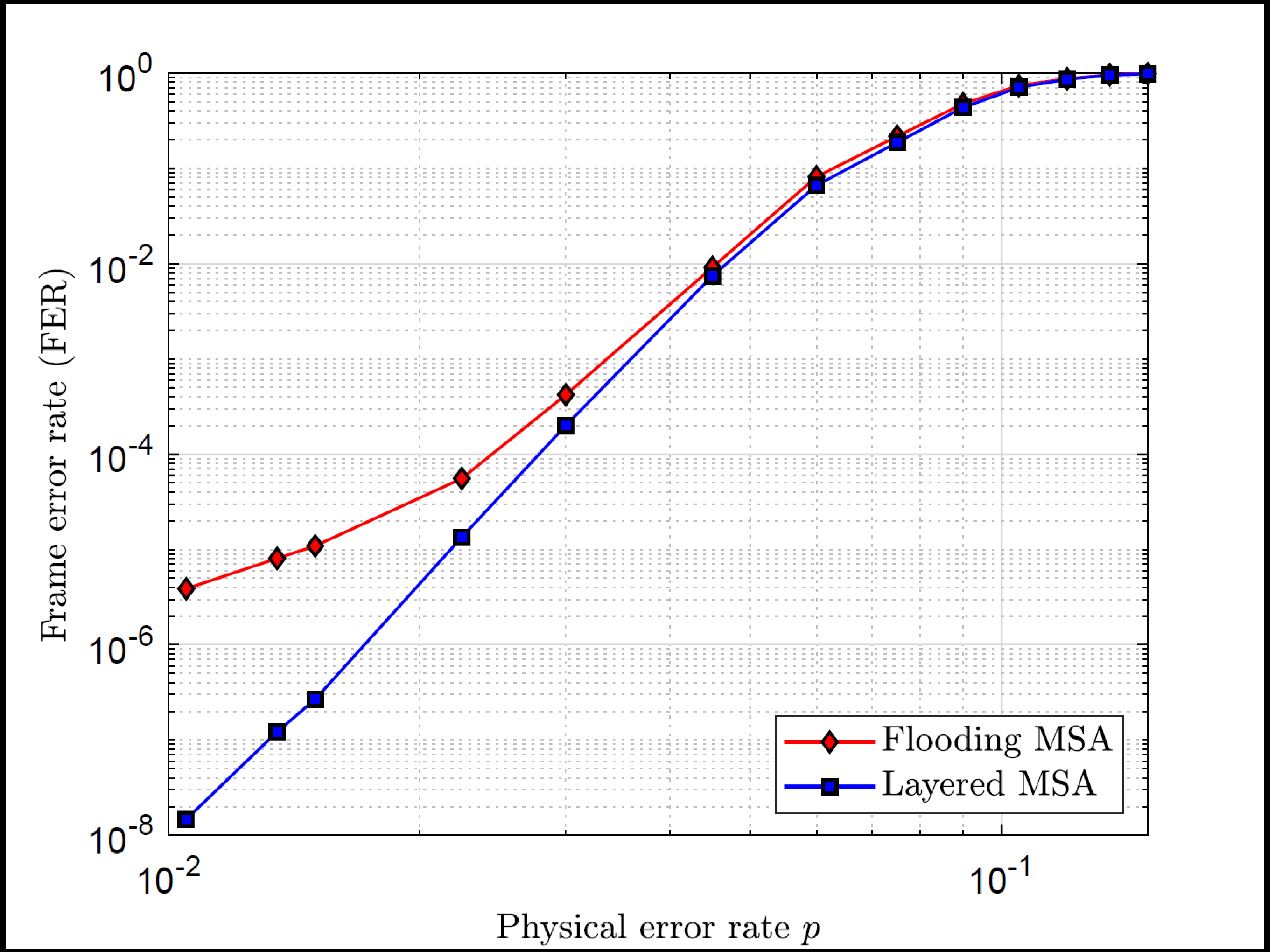
$$\text{Syndrome} = \begin{bmatrix} H_X \boldsymbol{d}^T \\ H_Z \boldsymbol{c}^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_X \\ \boldsymbol{s}_Z \end{bmatrix}$$

1. Decoding is based on syndrome, no received vector
2. No X/Z Correlations in Error: Run two independent iterative decoders based on H_X, H_Z
3. Depolarizing Channel: Exchange information or run non-binary decoder for X, Z, Y

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

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

Iterative Decoding at Finite Lengths



Raveendran, Vasić: https://arxiv.org/abs/2012.15297

$$\text{Syndrome} = \begin{bmatrix} H_X \boldsymbol{d}^T \\ H_Z \boldsymbol{c}^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_X \\ \boldsymbol{s}_Z \end{bmatrix}$$

1. Decoding Schedule: An iterative decoder can run parallel or serial node updates
2. Error Floor: The logical error rate saturates (“floors”) once physical error rate is small
3. Degeneracy: Two errors differing by a stabilizer are equivalent, are “degenerate”

Center for Quantum Networks *NSF-ERC*

Recent Exciting QLDPC Codes

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 60, NO. 2, FEBRUARY 2014 1193

Quantum LDPC Codes With Positive Rate and Minimum Distance Proportional to the Square Root of the Blocklength

Jean-Pierre Tillich and Gilles Zémor

Degenerate Quantum LDPC Codes With Good Finite Length Performance

Pavel Panteleev and Gleb Kalachev

Faculty of Mechanics and Mathematics, Moscow State University, GSP-1, Leninskie Gory, Moscow, 119991, Russian Federation

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 68, NO. 1, JANUARY 2022 213

Quantum LDPC Codes With Almost Linear Minimum Distance

Pavel Panteleev and Gleb Kalachev

Fiber Bundle Codes: Breaking the $N^{1/2}$ polylog(N) Barrier for Quantum LDPC Codes

Matthew B. Hastings
mahastin@microsoft.com
Station Q. and Microsoft Quantum
Redmond, WA, USA

Jeongwan Haah
Jeongwan.Haah@microsoft.com
Station Q. and Microsoft Quantum
Redmond, WA, USA

Ryan O'Donnell
odonnell@cs.cmu.edu
Carnegie Melon University
Pittsburgh, PA, USA

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 67, NO. 10, OCTOBER 2021 6653

Balanced Product Quantum Codes

Nikolas P. Breuckmann and Jens N. Eberhardt

Asymptotically Good Quantum and Locally Testable Classical LDPC Codes

Pavel Panteleev
Lomonosov Moscow State University
Russia
panteleev@intsys.msu.ru

Gleb Kalachev
Lomonosov Moscow State University
Russia
kalachev@intsys.msu.ru

Quantum Low-Density Parity-Check Codes

Nikolas P. Breuckmann[1,*] and Jens Niklas Eberhardt[2,†]

[1] *Department of Computer Science, University College London, London, WC1E 6BT, United Kingdom*
[2] *Mathematical Institute, University of Bonn, 53115 Bonn, Germany*

(Received 7 March 2021; revised 27 May 2021; published 11 October 2021)

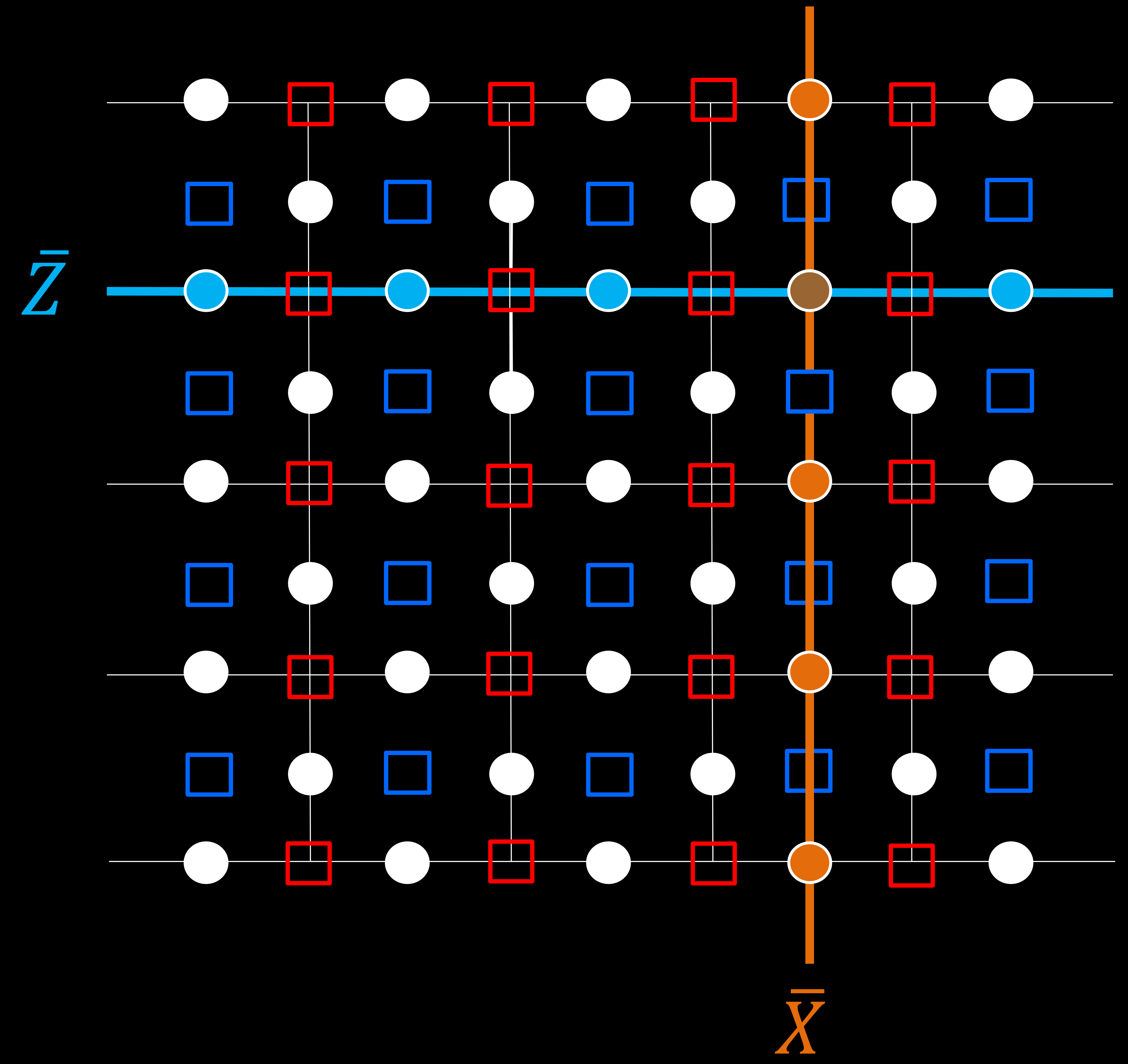
2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)

Quantum Tanner codes

Anthony Leverrier
Inria
Paris, France
anthony.leverrier@inria.fr

Gilles Zémor
Institut de Mathématiques de Bordeaux, UMR 5251
Bordeaux, France
zemor@math.u-bordeaux.fr

Towards "Good" Quantum LDPC Codes



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; large overhead	**Scalable with constant overhead???**

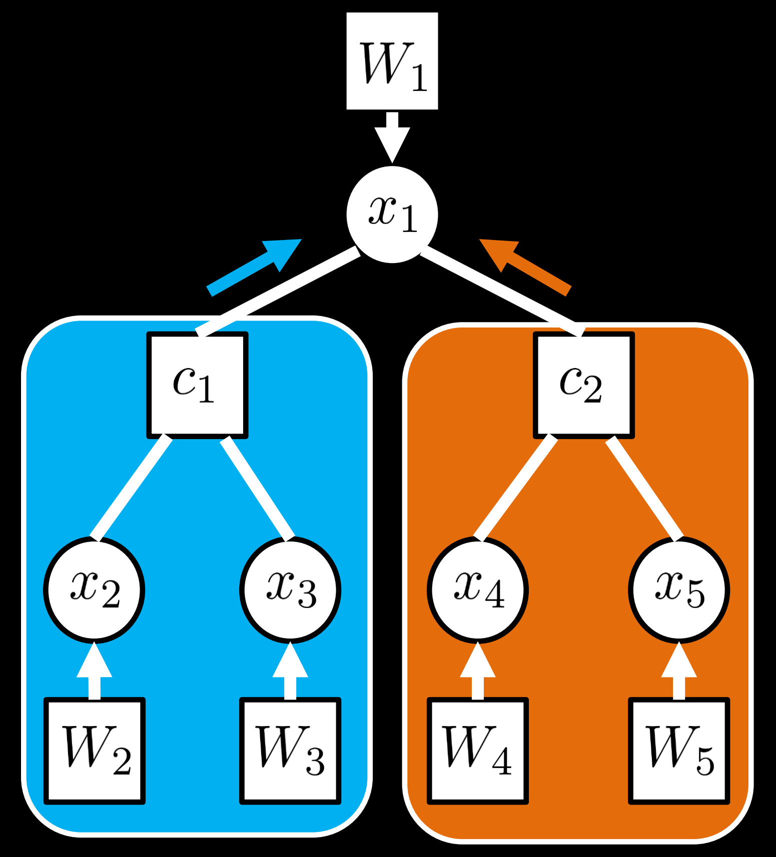
Summary

Code Parameters

$$[\![n, k, d]\!]$$

Physical Qubits → n · Logical Qubits → k · Minimum Distance → d

Decoder



[[3,1,1]] ``Bit-Flip'' Code:

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

$\overline{|\psi\rangle}_\varepsilon = \alpha|100\rangle + \beta|011\rangle$

Syndrome Measurement

Agenda

Part 1 Classical Error Correction

By Michele Pacenti

Part 2 Quantum Error Correction

By Narayanan Rengaswamy

Part 3 Application to Quantum Networks

By Sijie Cheng

Summary

[[3,1,1]] ``Bit-Flip'' Code:

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

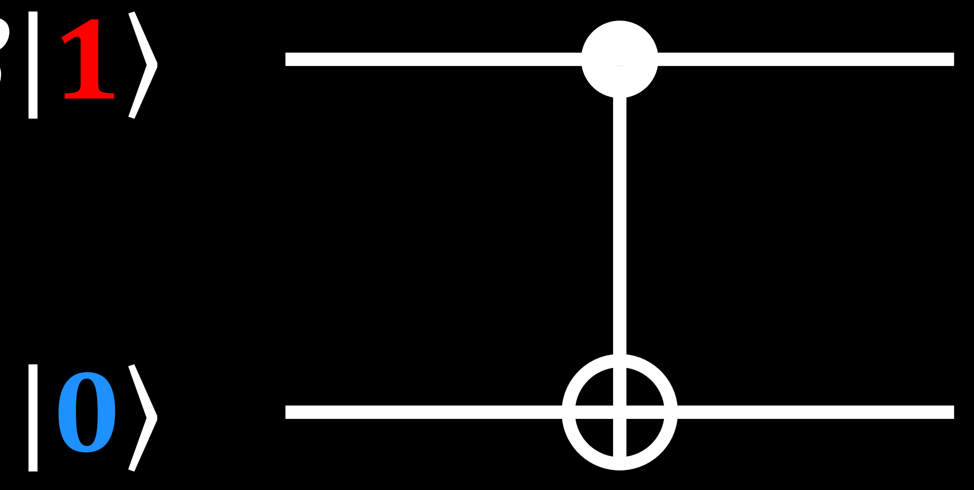
$\overline{|\psi\rangle}_{\varepsilon} = \alpha|100\rangle + \beta|011\rangle$

Syndrome Measurement

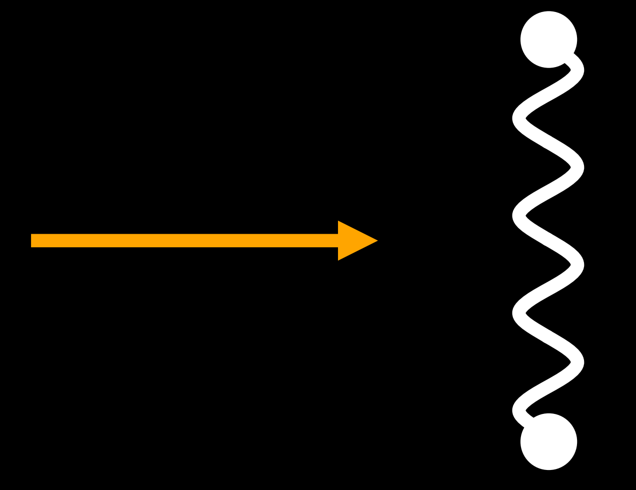
Bell State Encoder

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

$\boxed{\text{Set } \alpha = \beta = \frac{1}{\sqrt{2}}}$

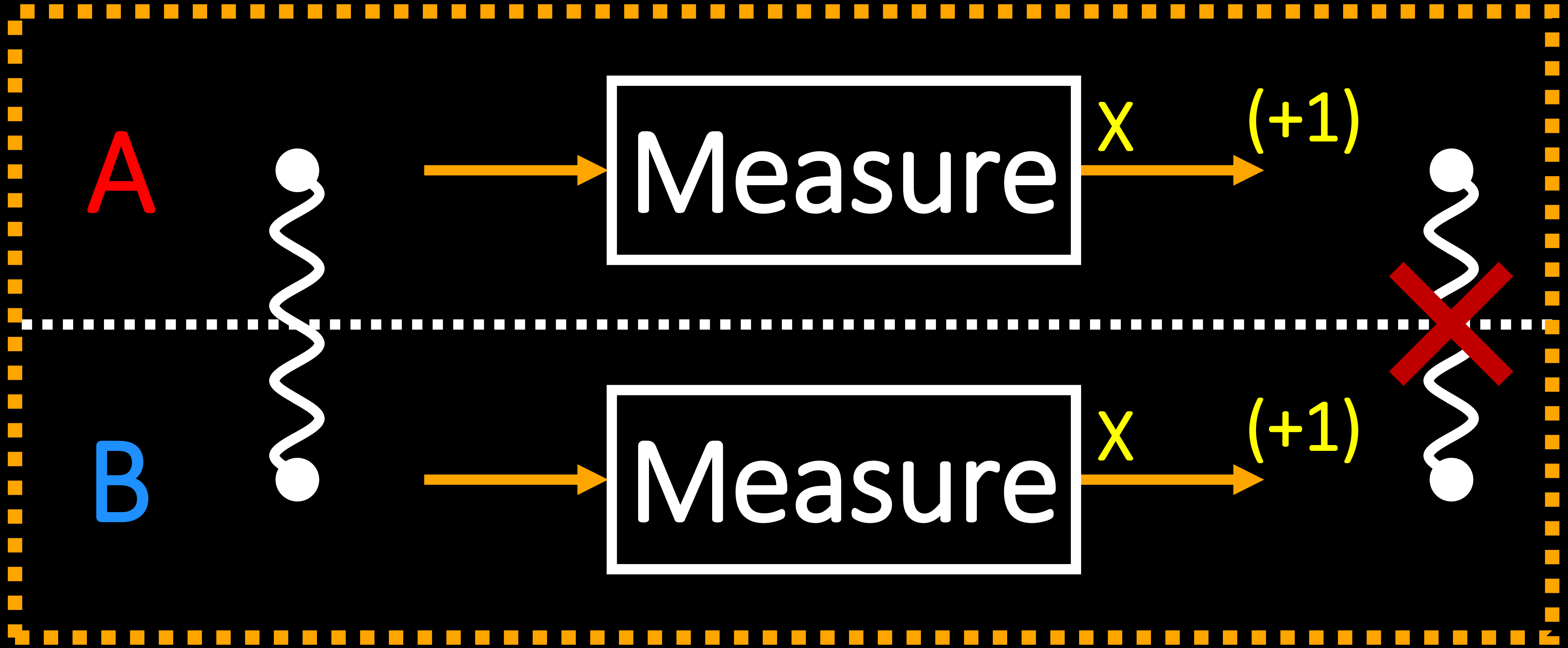


$$\overline{|\psi\rangle} = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$



Bell State Measurement Property

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

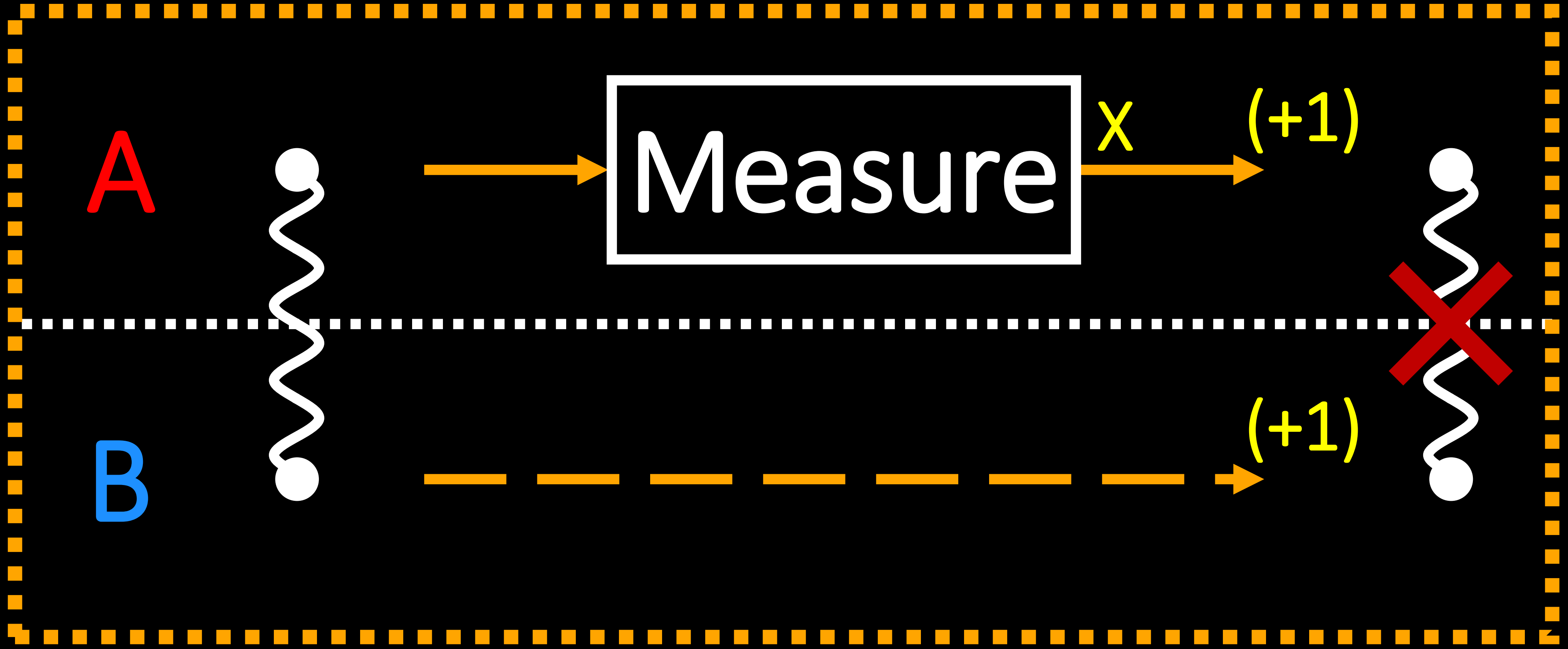


$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

=

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$



$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

Legend

Un-coded State without Error

Stabilizer Generator

Coded State without Error

Sign for Stabilizers

Un-coded State with Error

Coded State with Error

Syndrome for Error

Estimated Error

Classical Channel without Error

Quantum Channel without Error

Quantum Channel with Error

Measure

Decoder

Correct

Bell State

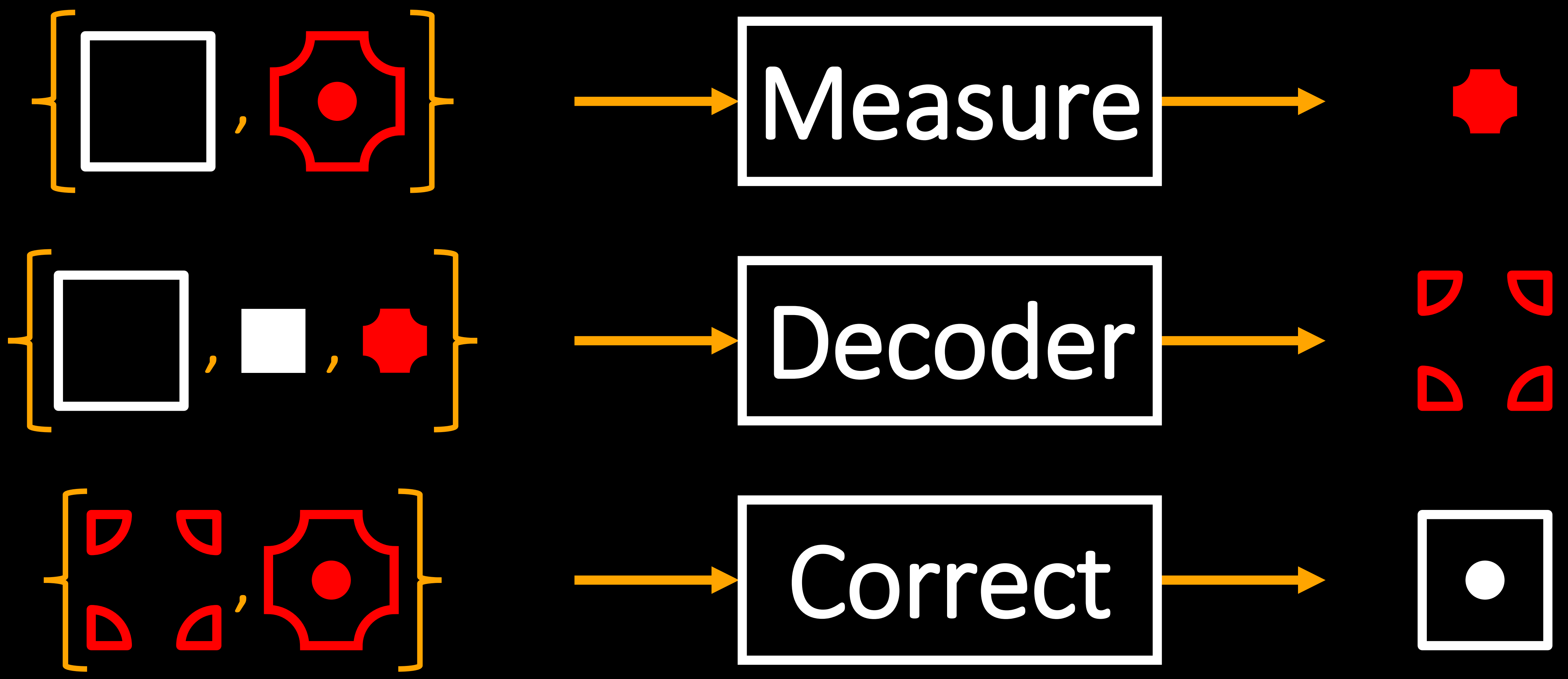
Bell Pair without Error

Bell Pair with Error

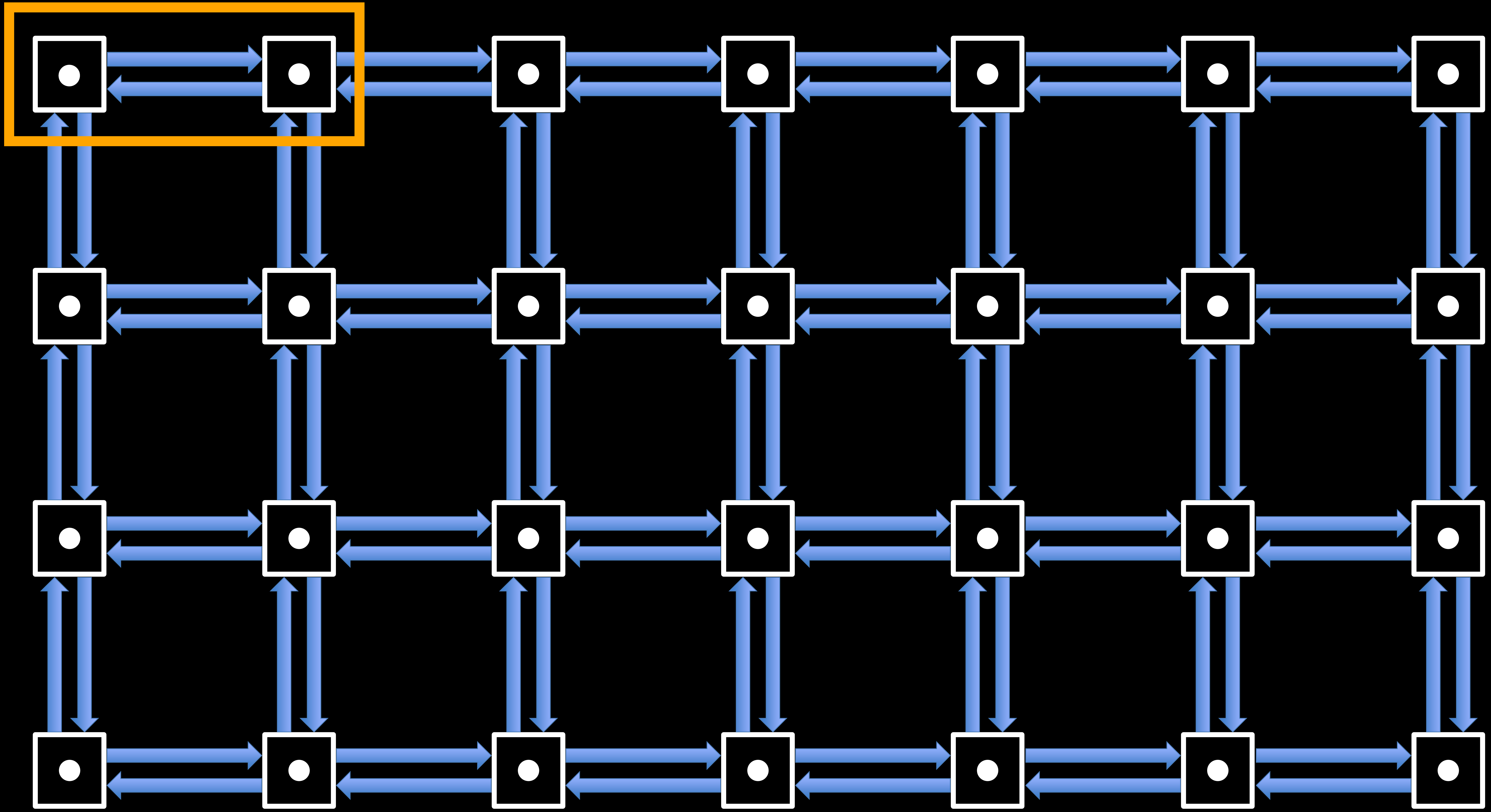
Equality

Error Correction Process

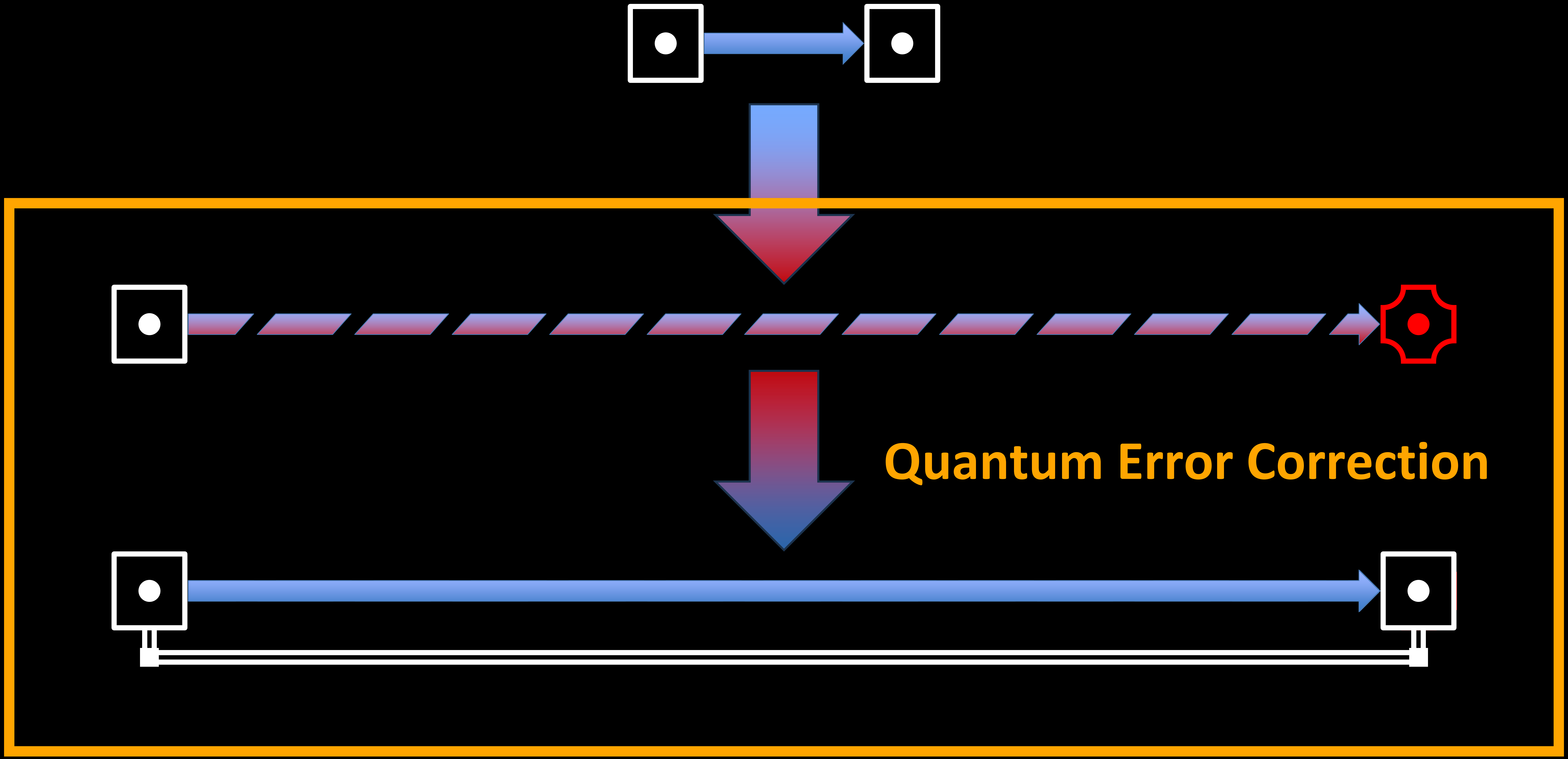
- Un-coded State without Error
- Stabilizer Generator
- Coded State without Error
- Sign for Stabilizers
- Un-coded State with Error
- Coded State with Error
- Syndrome for Error
- Estimated Error
- Classical Channel without Error
- Quantum Channel without Error
- Quantum Channel with Error



Quantum Network



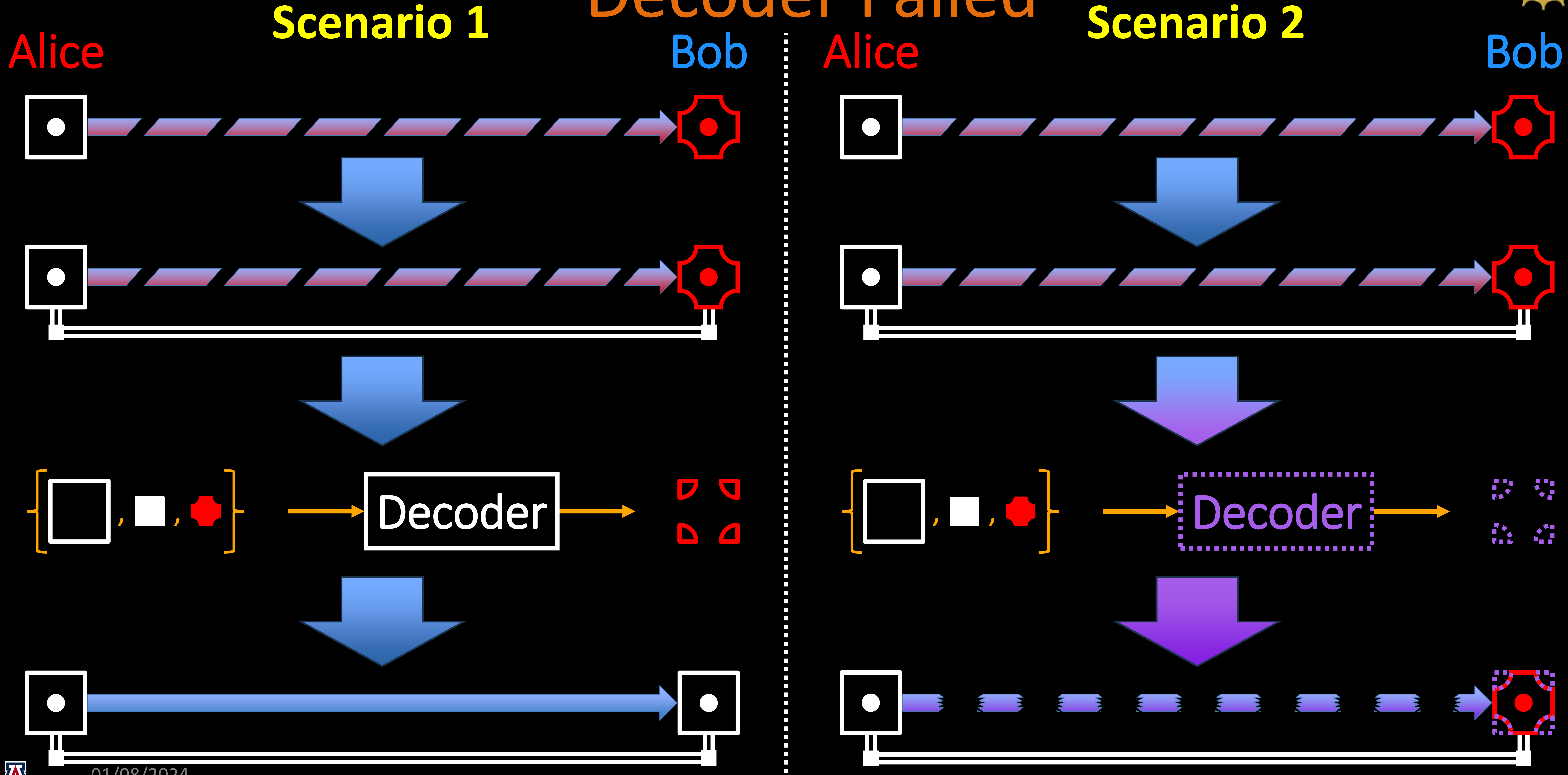
Direct Application



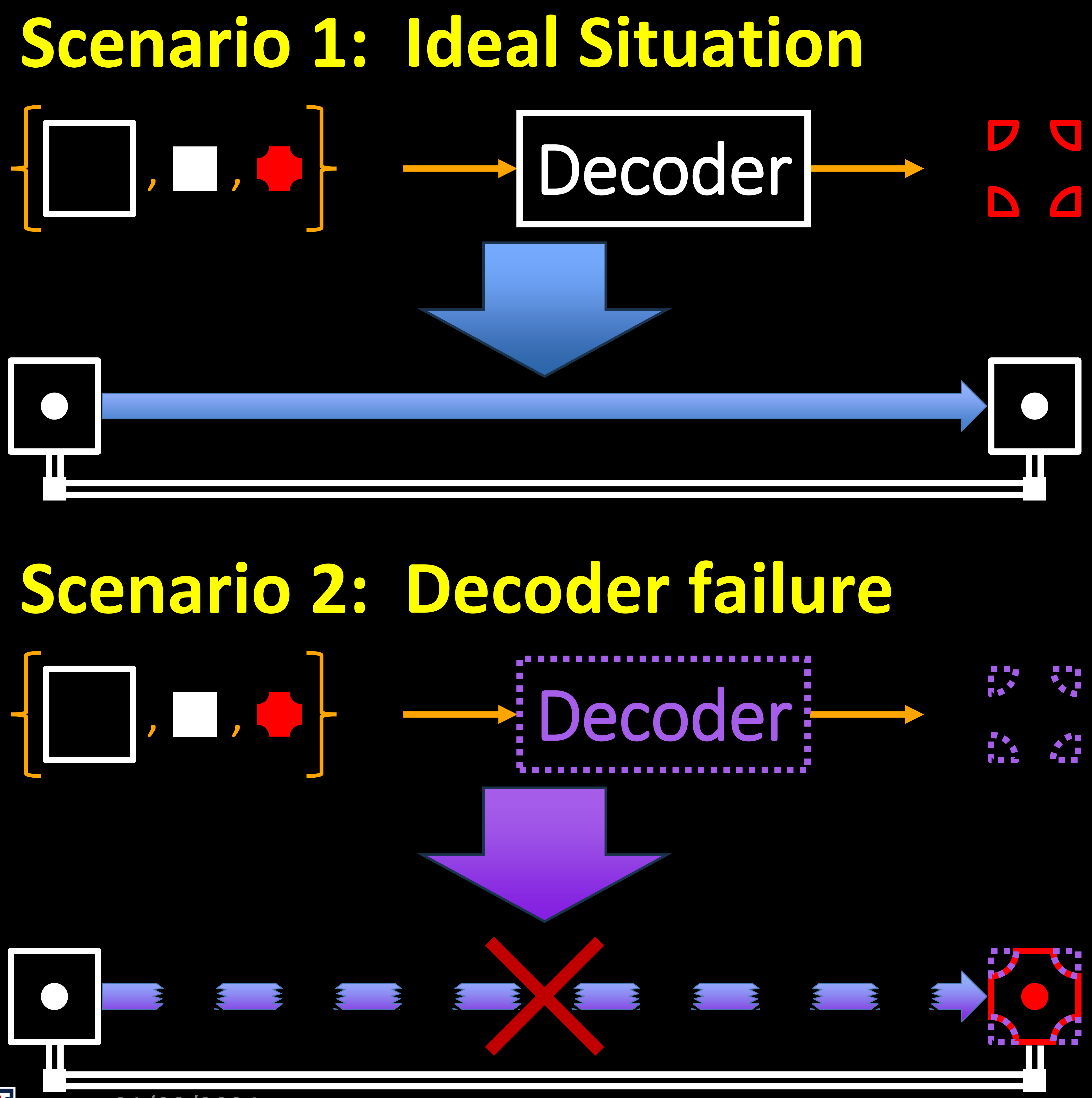
Center for Quantum Networks NSF-ERC

Quantum Error Correction

Decoder Failed



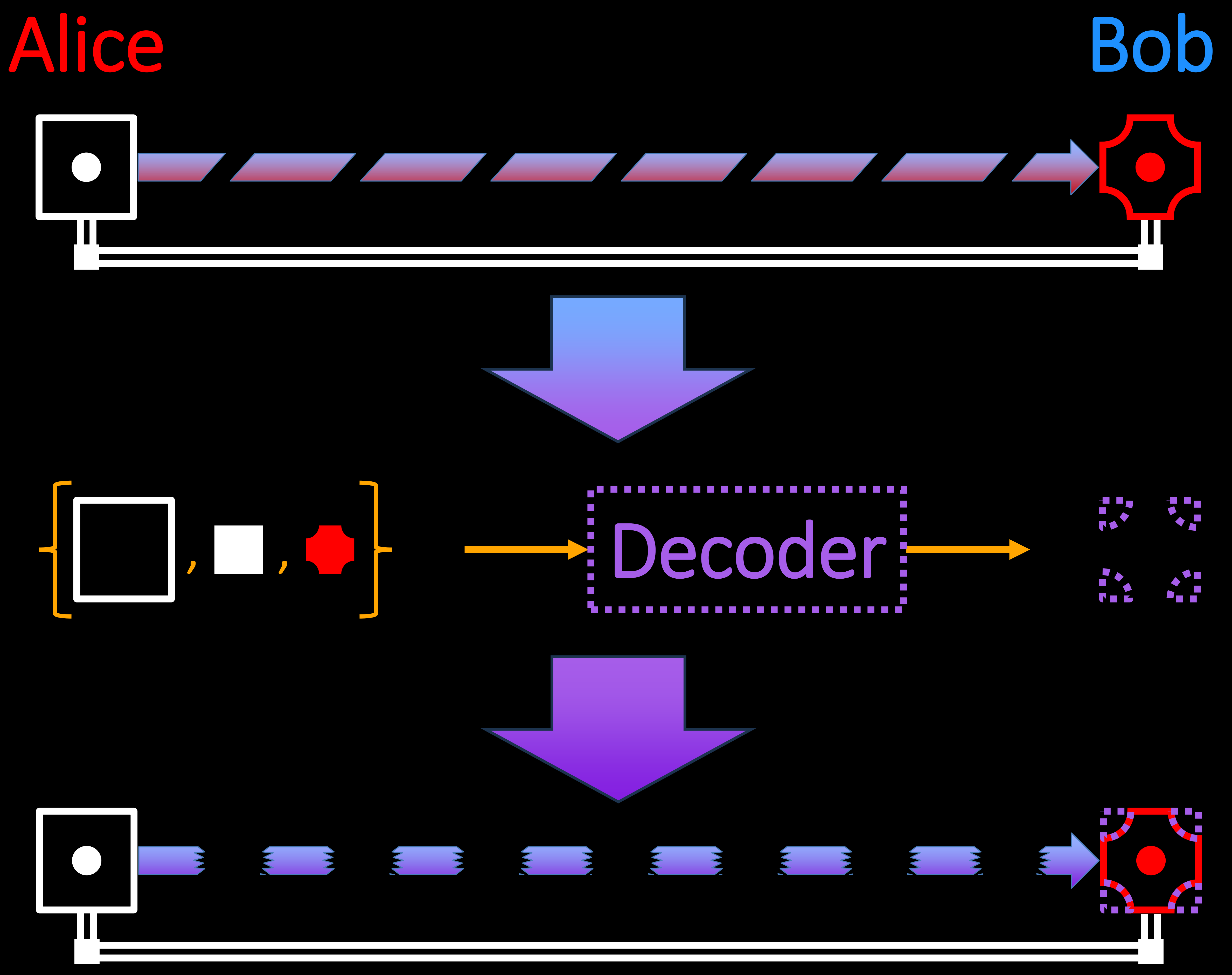
Problem 1 – Long Distance

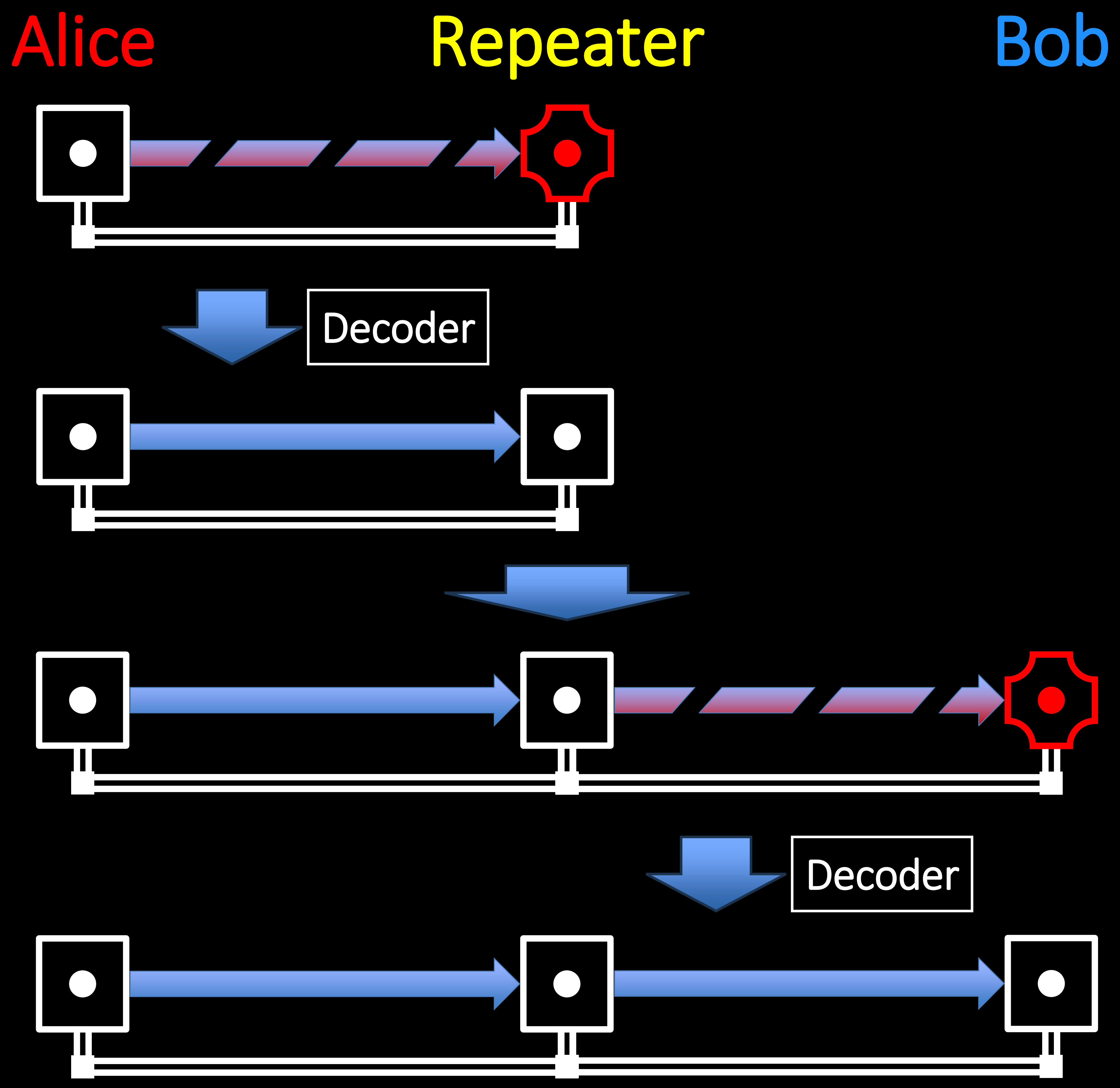


1. As the distance increases, the quantum channel may experience more errors, potentially causing the decoder to fail.

2. If decoding is performed in real-time, a decoder failure will result in the quantum link being cut off. We need to find a method to perform decoding earlier, yet not in real-time, to reduce the impact of such failures.

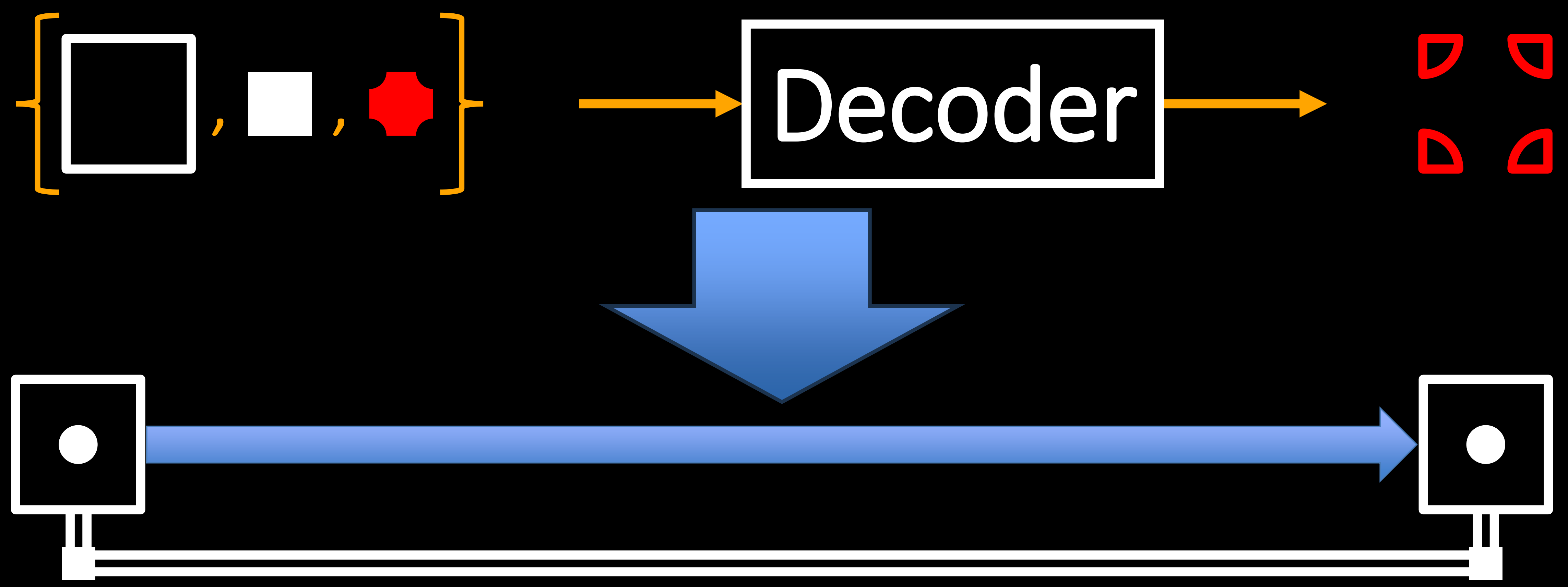
Solution for Long Distance



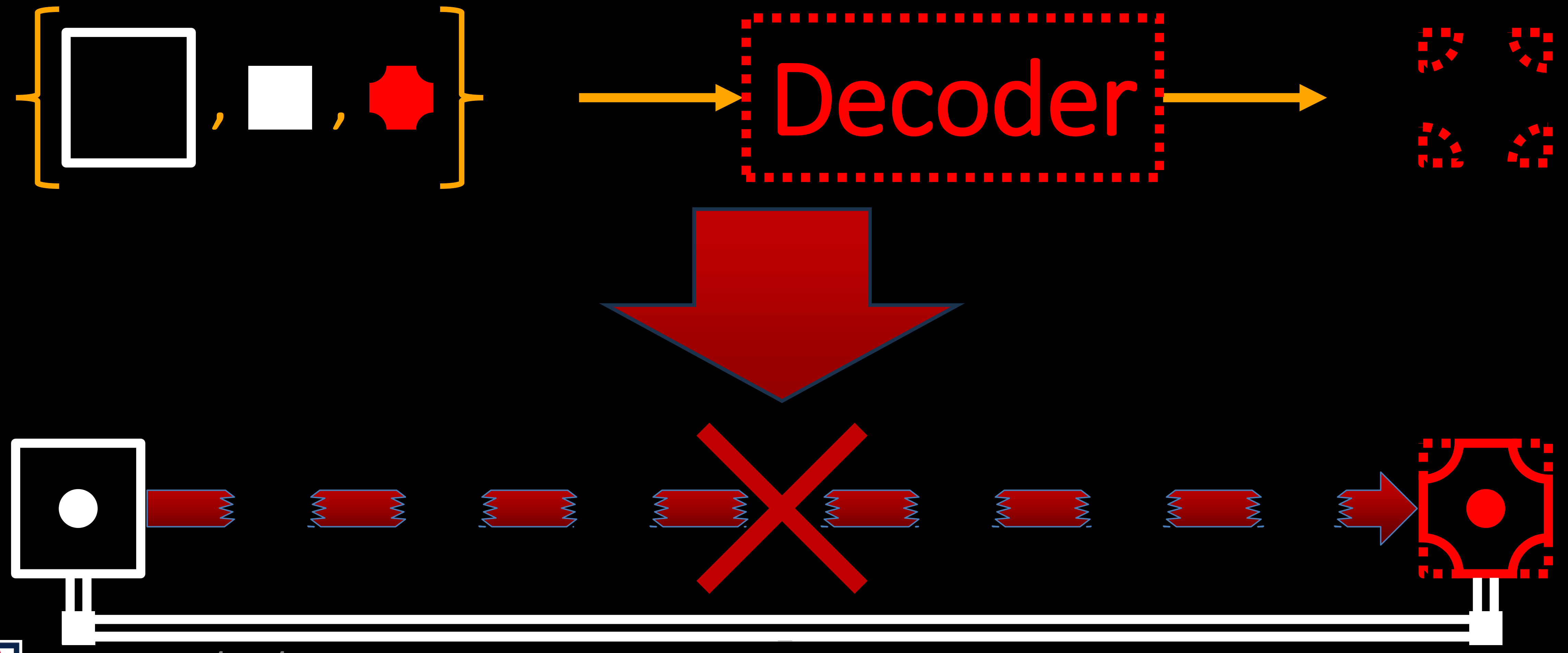


Problem 2 – Quantum Circuit Error

Scenario 1: Ideal Situation



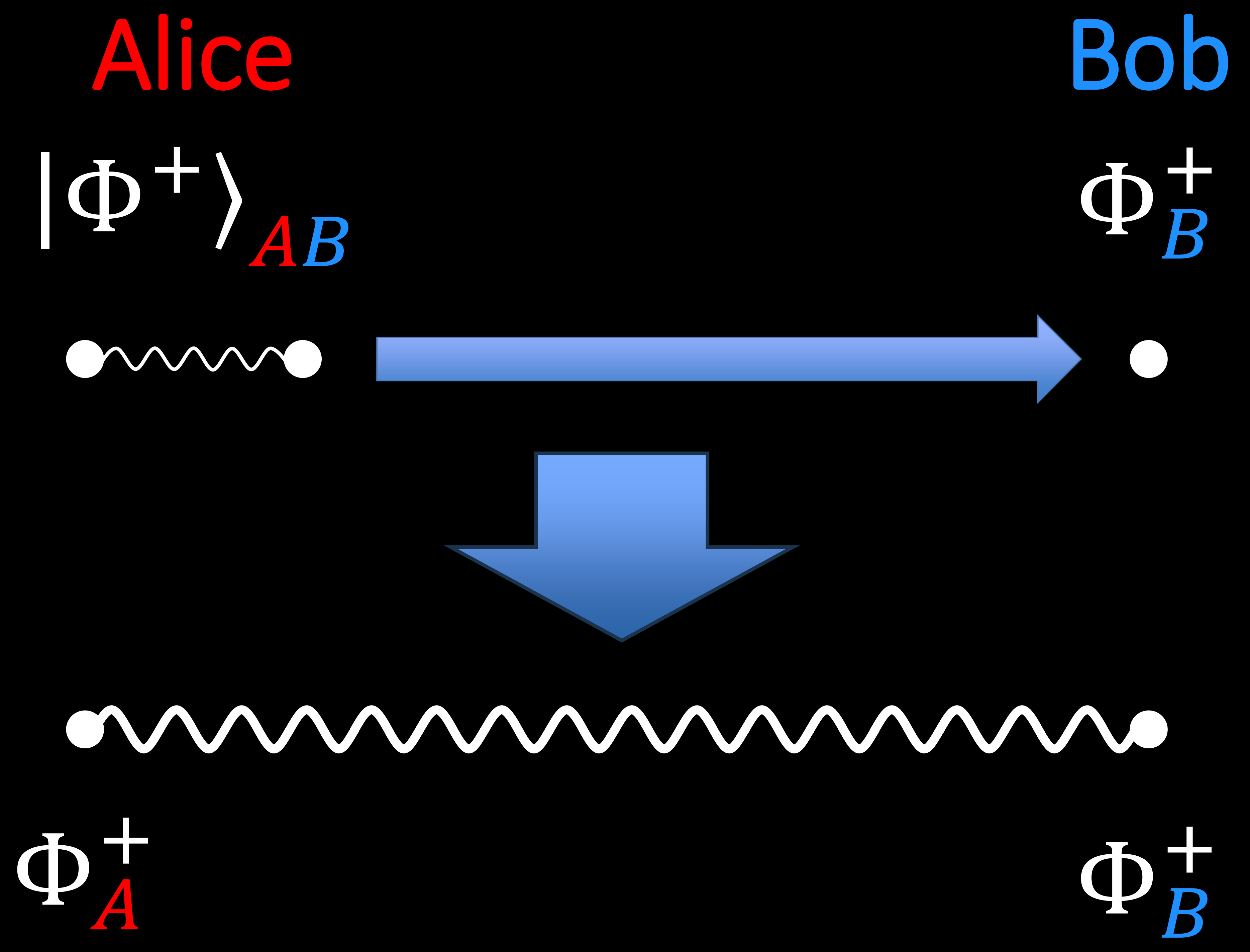
Scenario 2: Decoder failure



1. As the distance increases, the quantum channel may experience more errors, potentially causing the decoder to fail.

2. If decoding is performed in real-time, a decoder failure will result in the quantum link being cut off. We need to find a method to perform decoding earlier, yet not in real-time, to reduce the impact of such failures.

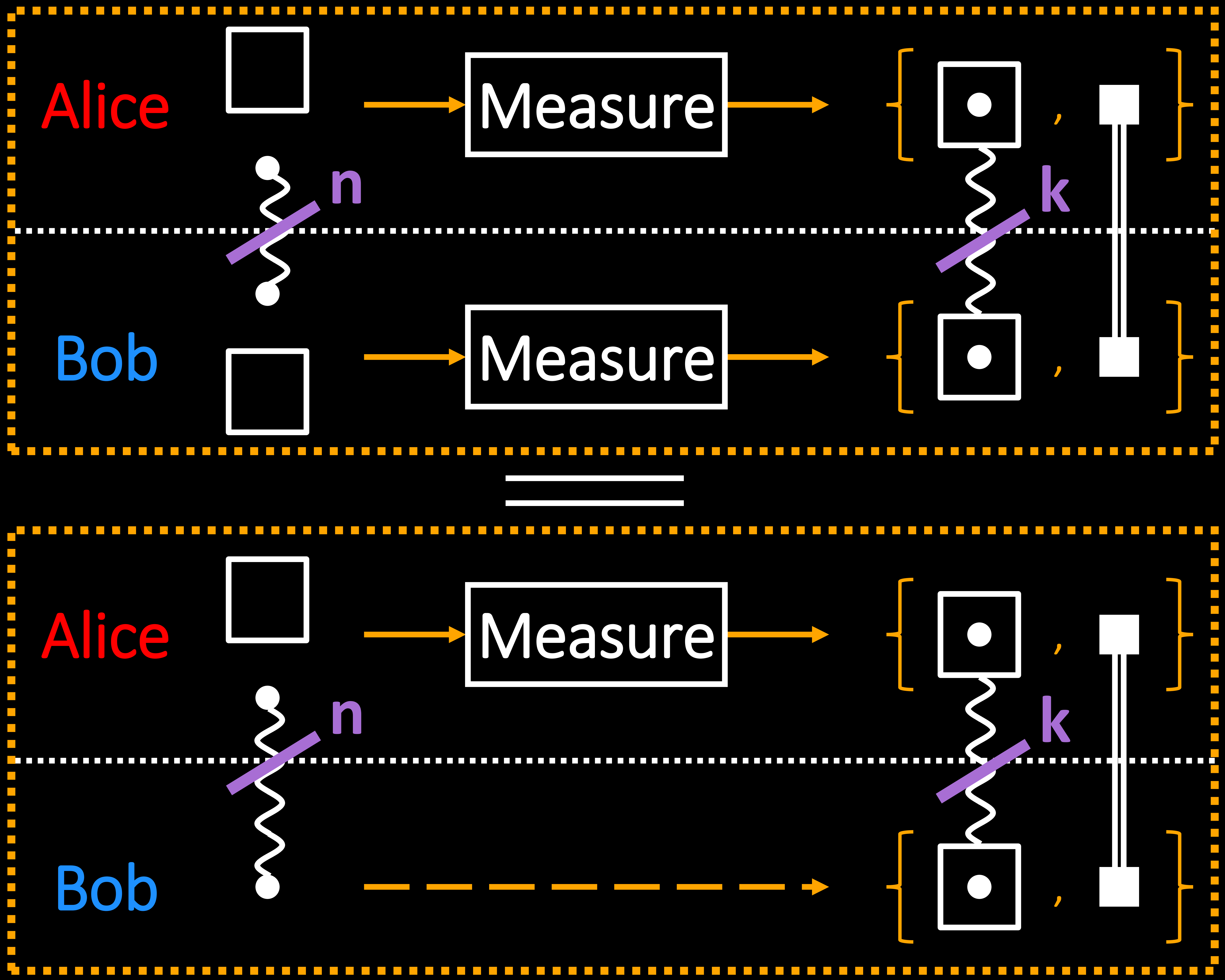
Stabilizer Measurement on Bell State



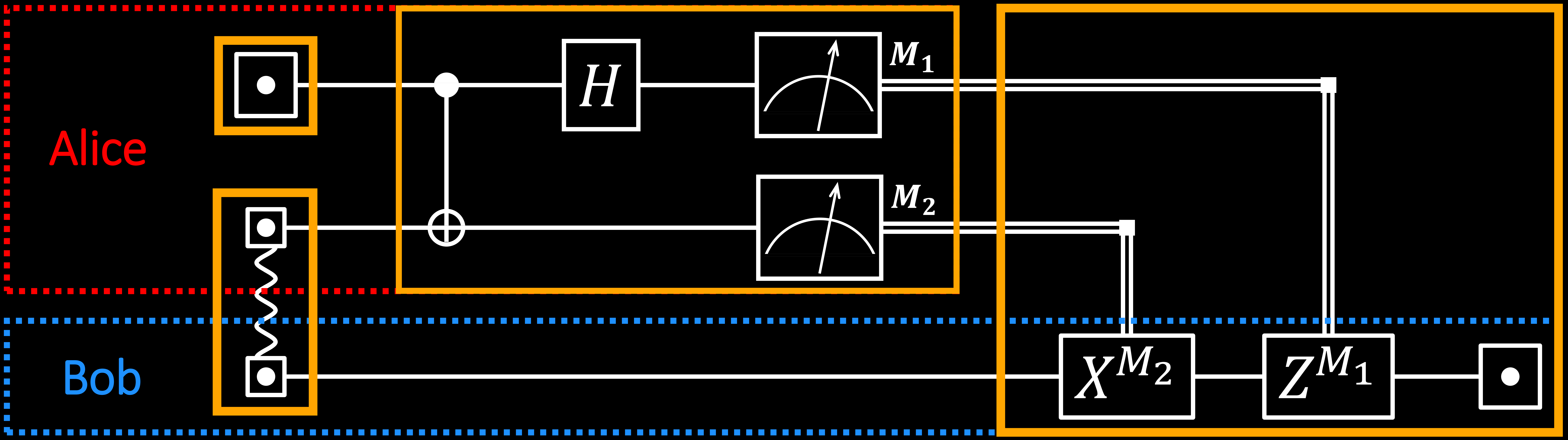
$|\Phi^+\rangle_{AB}$: Bell State

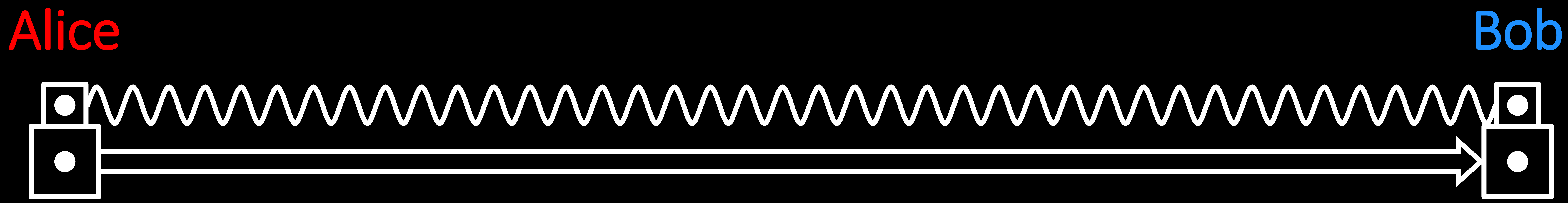
$\Phi_A^+ = Tr_B\left[|\Phi^+\rangle_{AB}\langle\Phi^+|\right]$: Alice Part

$\Phi_B^+ = Tr_A\left[|\Phi^+\rangle_{AB}\langle\Phi^+|\right]$: Bob Part

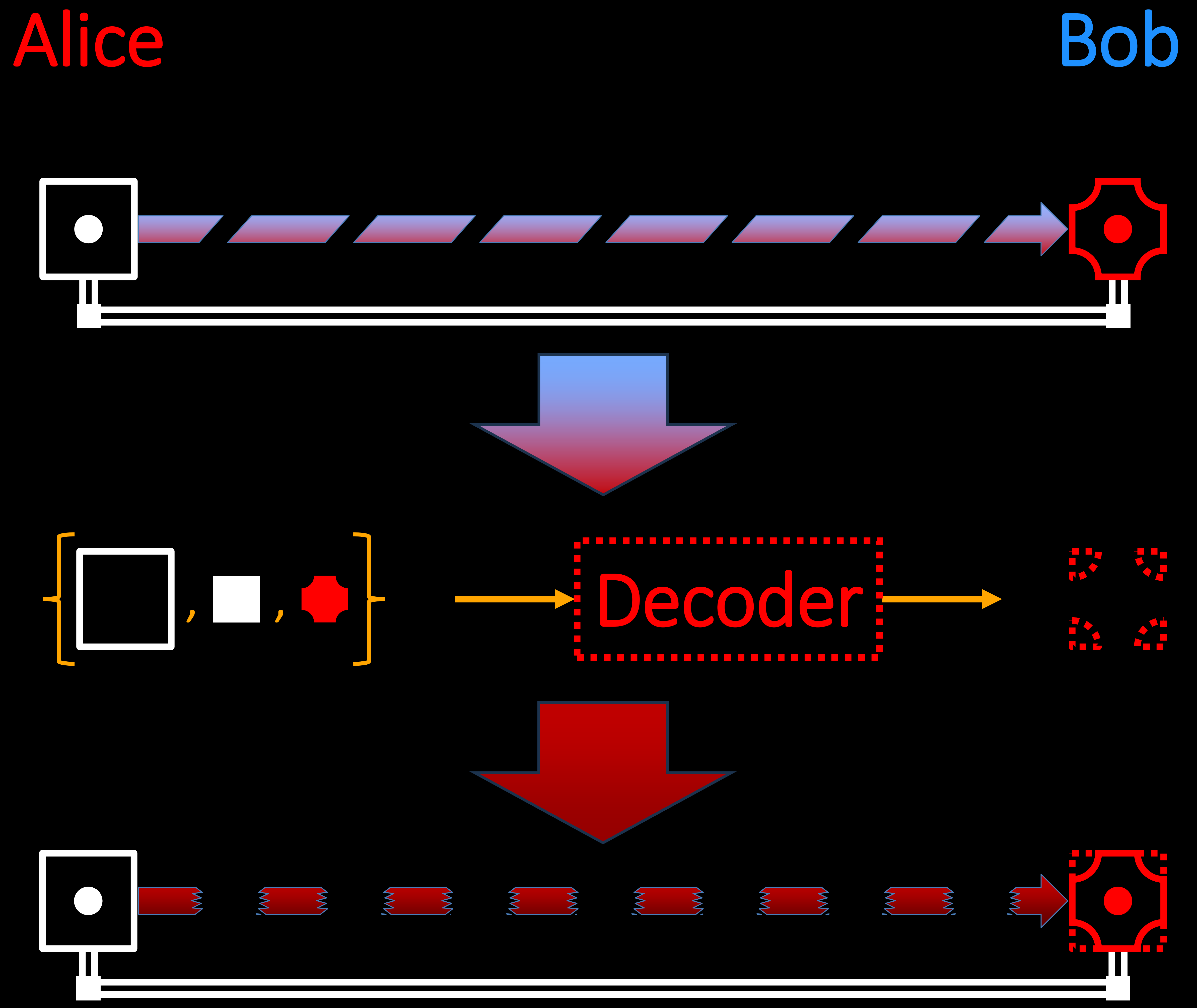


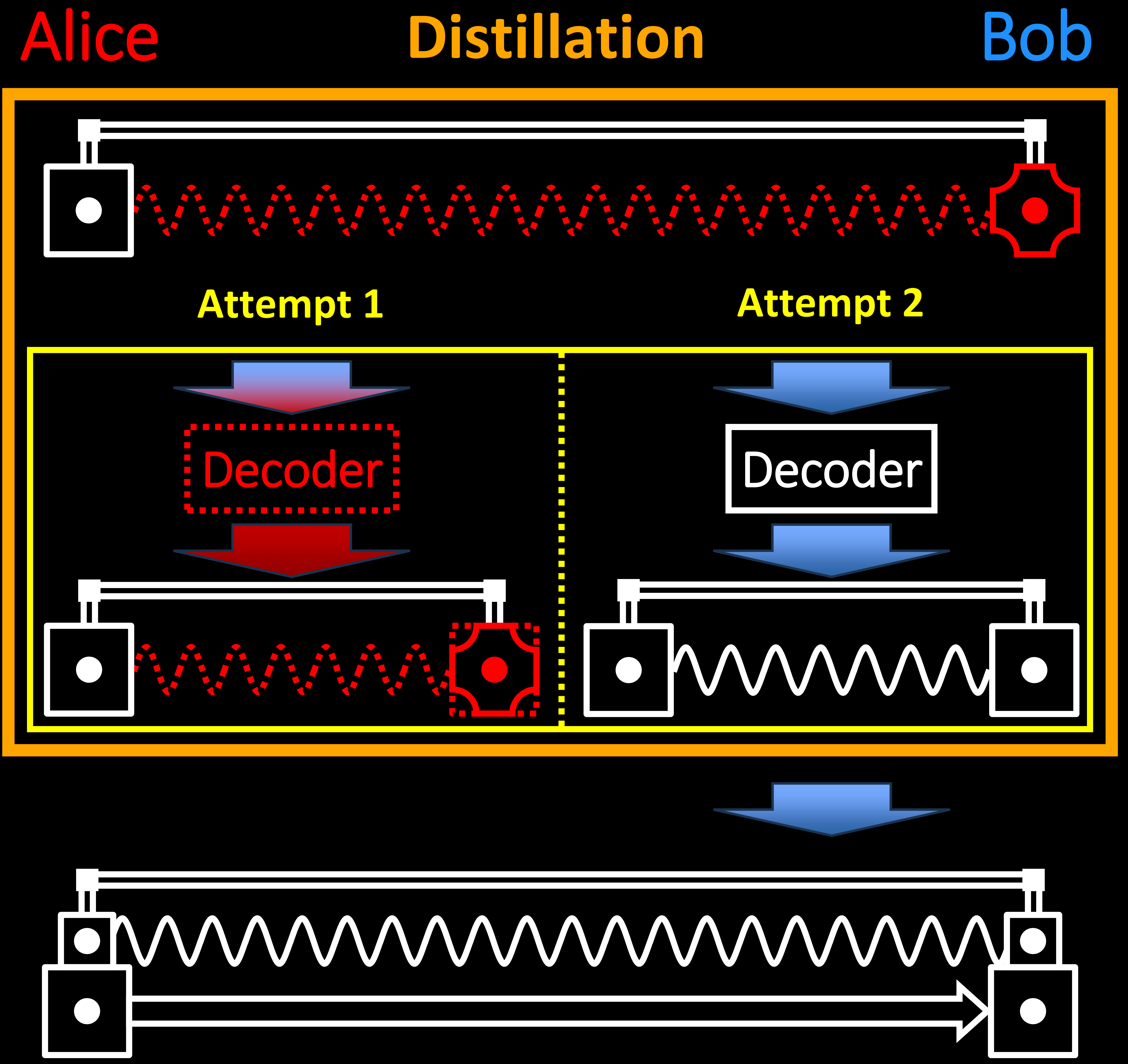
Teleportation





Solution for Quantum Circuit Error





Distillation based on Error Correction

Summary

Coded State without Error

Sign for Stabilizers

Bell State

Classical Channel without Error

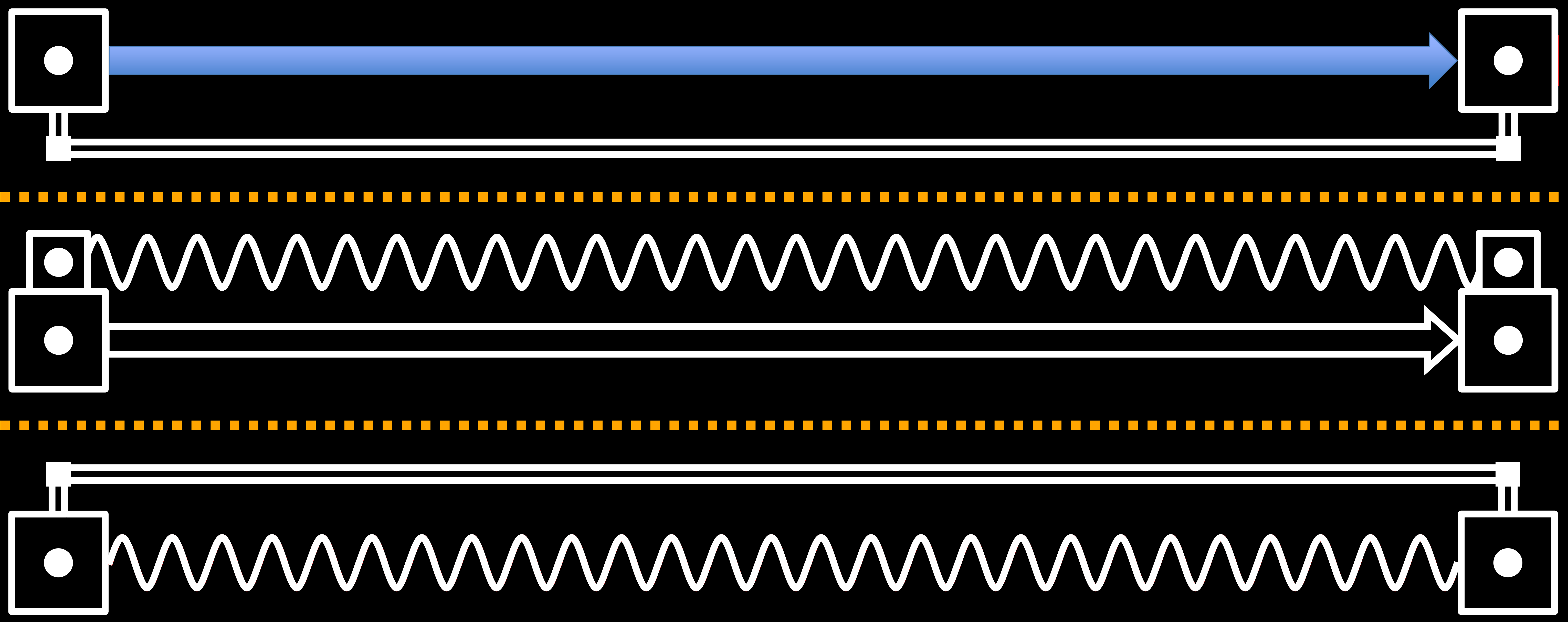
Quantum Channel without Error

Bell Pair without Error

Quantum Error Correction

Teleportation

Distillation



Center for Quantum Networks
NSF Engineering Research Center

Course Evaluation Survey

We value your feedback on all aspects of this short course. Please go to the link provided in the Zoom Chat or in the email you will soon receive to give your opinions of what worked and what could be improved.

2025 CQN Winter School on Quantum Networks

Funded by National Science Foundation Grant #1941583