Overview of quantum error correction

with a view on machine learning opportunities

Michael Vasmer, Future Horizons Workshop, 15 May 2024

Quantum error correction

Why should you care?

Quantum error correction

Why should you care?

- Noise severely limits the performance of today's quantum computers

Quantum error correction

Why should you care?

- Noise severely limits the performance of today's quantum computers
- Quantum error correction enables scalable quantum computation for applications

Quantum error correction

Why should you care?

- Noise severely limits the performance of today’s quantum computers
- Quantum error correction enables scalable quantum computation for applications
  - Factoring, chemical simulation, optimisation, ...

Quantum error correction

Why should you care?

- Noise severely limits the performance of today's quantum computers
- Quantum error correction enables scalable quantum computation for applications
  - Factoring, chemical simulation, optimisation, ...
  - But space & time overheads are currently prohibitive

Quantum error correction

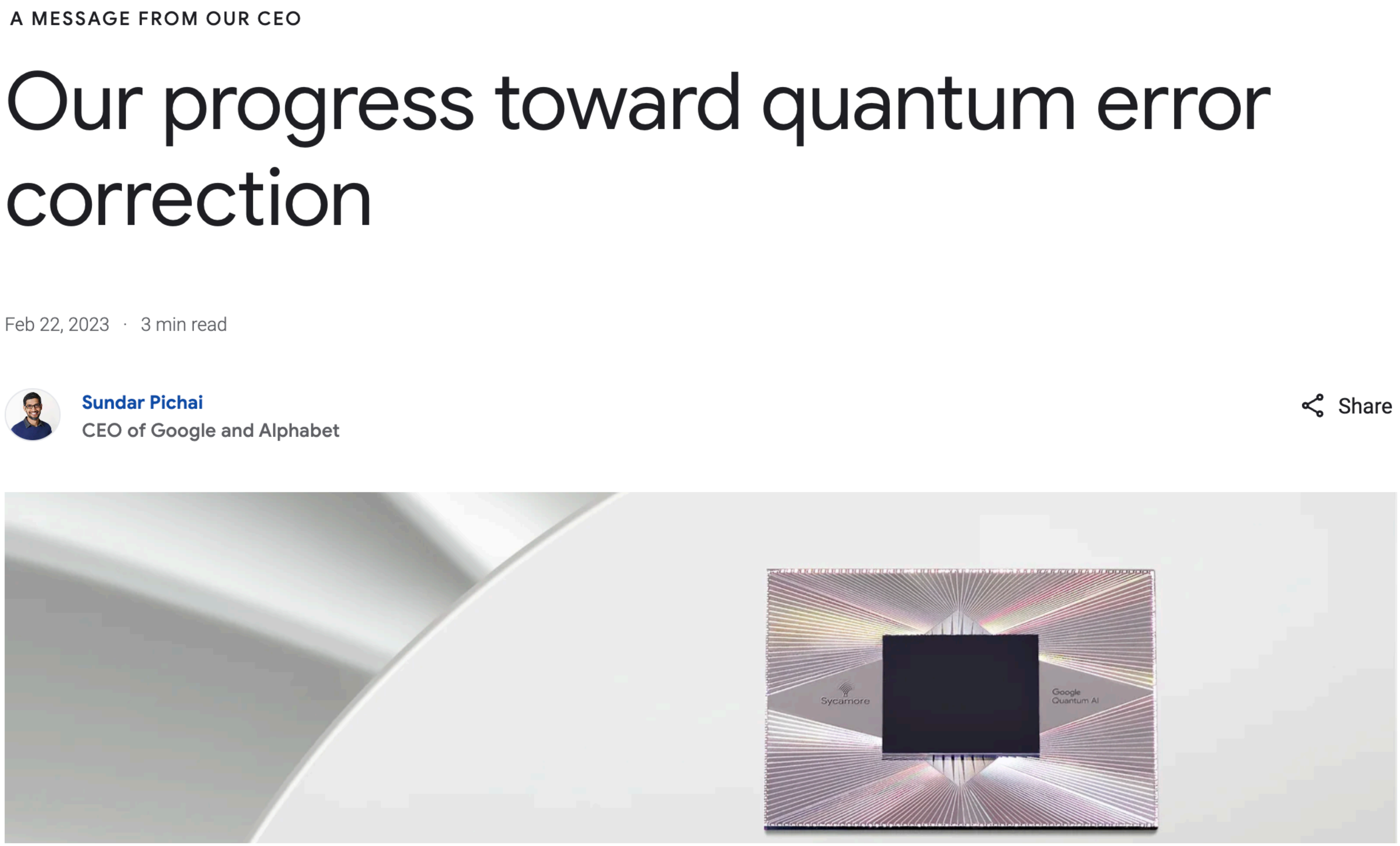
Why should you care?

- Noise severely limits the performance of today’s quantum computers
- Quantum error correction enables scalable quantum computation for applications
  - Factoring, chemical simulation, optimisation, …
  - But space & time overheads are currently prohibitive
- Big focus for academia, government, and industry

Quantum error correction

Why should you care?

- Noise severely
- Quantum erro ions
  - Factoring, c
  - But space &
- Big focus for c


A MESSAGE FROM OUR CEO

Our progress toward quantum error correction

Feb 22, 2023 · 3 min read

Sundar Pichai
CEO of Google and Alphabet

Share

Quantum error correction

Why should you care?

- Noise severely limits the performance of today's quantum computers
- Quantu
  - Factc
  - But sp
- Big focu

PsiQuantum to Build World's First Utility-Scale, Fault-Tolerant Quantum Computer in Australia

The Australian and Queensland Governments Will Invest $940M AUD ($620M USD) into PsiQuantum

April 29, 2024 04:00 PM Eastern Daylight Time

BRISBANE, Australia--(BUSINESS WIRE)--PsiQuantum today announced it will build the world's first utility-scale quantum computer at a strategically located site near Brisbane Airport in Brisbane, Australia. The Australian Commonwealth and Queensland Governments will invest $940M AUD ($620M USD) into PsiQuantum through a financial package, comprised of equity, grants, and loans. PsiQuantum is on an aggressive plan to have the site operational by the end of 2027. A fault-tolerant quantum computer will be able to solve commercially useful problems across industries built upon chemistry, math, and physics; thereby transforming critical industries – including renewable energy, minerals and metals, healthcare and transportation – that will propel the global economy for decades to come.

Quantum error correction

Why should you care?

- Noise severely limits the performance of today's quantum computers
- Quantum error correction enables scalable quantum computation for applications
  - Factoring, chemical simulation, optimisation, ...
  - But space & time overheads are currently prohibitive
- Big focus for academia, government, and industry

Quantum error correction

Why should you care?

- Noise severely limits the performance of today's quantum computers
- Quantum error correction enables scalable quantum computation for applications
  - Factoring, chemical simulation, optimisation, ...
  - But space & time overheads are currently prohibitive
- Big focus for academia, government, and industry
- Surprising connections to other fields in computer science & physics

Quantum error correction

Why should you care?

- Noise severely limits the performance of today's quantum computers
- Quantum error correction enables scalable quantum computation for applications
  - Factoring, chemical simulation, optimisation, ...
  - But space & time overheads are currently prohibitive
- Big focus for academia, government, and industry
- Surprising connections to other fields in computer science & physics
  - Topological phases of matter, black holes, quantum PCP conjecture, ...

Talk outline

- Quantum error correction 101
- The decoding problem
- Finding fault-tolerant logical gates

Quantum error correction 101

Quantum error correction 101

One quantum bit

Quantum error correction 101

One quantum bit

- Quantum bit (qubit) $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$

Quantum error correction 101

One quantum bit

- Quantum bit (qubit) $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$
- $|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}$, $|1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$, $\alpha, \beta \in \mathbb{C}$, $|\alpha|^2 + |\beta|^2 = 1$

Quantum error correction 101

One quantum bit

- Quantum bit (qubit) $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$
- $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $\alpha, \beta \in \mathbb{C}$, $|\alpha|^2 + |\beta|^2 = 1$
- Pauli matrices $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$, and $Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

Quantum error correction 101

One quantum bit

- Quantum bit (qubit) $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$
- $|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}$, $|1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$, $\alpha, \beta \in \mathbb{C}$, $|\alpha|^2 + |\beta|^2 = 1$
- Pauli matrices $X = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$, $Y = \begin{bmatrix}0 & -i\\i & 0\end{bmatrix}$, and $Z = \begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}$
- Quantum errors

Quantum error correction 101

One quantum bit

- Quantum bit (qubit) $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$
- $|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}$, $|1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$, $\alpha, \beta \in \mathbb{C}$, $|\alpha|^2 + |\beta|^2 = 1$
- Pauli matrices $X = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$, $Y = \begin{bmatrix}0 & -i\\i & 0\end{bmatrix}$, and $Z = \begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}$
- Quantum errors
  - Bit-flip $|0\rangle \xrightarrow{X} |1\rangle$, phase-flip $|0\rangle + |1\rangle \xrightarrow{Z} |0\rangle - |1\rangle$

Quantum error correction 101

Many qubits

Quantum error correction 101

Many qubits

- n-qubit (Hilbert) space $\mathscr{H} = (\mathbb{C}^2)^{\otimes n}$

Quantum error correction 101

Many qubits

- n-qubit (Hilbert) space $\mathscr{H} = (\mathbb{C}^2)^{\otimes n}$
    - Computational basis states = tensor products of $|0\rangle$ and $|1\rangle$

Quantum error correction 101

Many qubits

- n-qubit (Hilbert) space $\mathscr{H} = (\mathbb{C}^2)^{\otimes n}$
  - Computational basis states = tensor products of $|0\rangle$ and $|1\rangle$
- n-qubit Pauli group $\mathscr{P}_n$ generated by tensor products of X, Y, Z, I

Quantum error correction 101

Many qubits

- n-qubit (Hilbert) space $\mathscr{H} = (\mathbb{C}^2)^{\otimes n}$
  - Computational basis states = tensor products of $|0\rangle$ and $|1\rangle$
- n-qubit Pauli group $\mathscr{P}_n$ generated by tensor products of X, Y, Z, I
- Quantum error-correcting code: subspace $\mathscr{C} \subseteq \mathscr{H}$

Quantum error correction 101

Stabiliser codes [Got97]

Quantum error correction 101

Stabiliser codes [Got97]

- Stabiliser group $\mathcal{S} \leq \mathcal{P}_n$

Quantum error correction 101

Stabiliser codes [Got97]

- Stabiliser group $\mathcal{S} \leq \mathcal{P}_n$
  - Abelian and $-I \notin \mathcal{S}$

Quantum error correction 101

Stabiliser codes [Got97]

- Stabiliser group $\mathscr{S} \leq \mathscr{P}_n$
  - Abelian and $-I \notin \mathscr{S}$
- Code (sub)space $\mathscr{C} = \{ |\psi\rangle \in (\mathbb{C}^2)^{\otimes n} : S|\psi\rangle = |\psi\rangle \ \forall S \in \mathscr{S} \}$

Quantum error correction 101

Stabiliser codes [Got97]

- Stabiliser group $\mathcal{S} \leq \mathcal{P}_n$
  - Abelian and $-I \notin \mathcal{S}$
- Code (sub)space $\mathcal{C} = \{ |\psi\rangle \in (\mathbb{C}^2)^{\otimes n} : S|\psi\rangle = |\psi\rangle \ \forall S \in \mathcal{S} \}$
- Dimension of $\mathcal{C}$: $k = n - \mathrm{rank}(\mathcal{S})$

Quantum error correction 101

Stabiliser codes [Got97]

Quantum error correction 101

Stabiliser codes [Got97]

- Code space $\mathscr{C} = \{ |\psi\rangle \in (\mathbb{C}^2)^{\otimes n} : S|\psi\rangle = |\psi\rangle \; \forall S \in \mathcal{S} \}$

Quantum error correction 101

Stabiliser codes [Got97]

- Code space $\mathscr{C} = \{ |\psi\rangle \in (\mathbb{C}^2)^{\otimes n} : S|\psi\rangle = |\psi\rangle \; \forall S \in \mathcal{S} \}$
- Normaliser of $\mathcal{S}$, $N(\mathcal{S}) = \{P \in \mathscr{P}_n : [P, S] = 0 \; \forall S \in \mathcal{S}\}$

Quantum error correction 101

Stabiliser codes [Got97]

- Code space $\mathscr{C} = \{\, |\psi\rangle \in (\mathbb{C}^2)^{\otimes n} : S|\psi\rangle = |\psi\rangle \; \forall S \in \mathcal{S}\,\}$
- Normaliser of $\mathcal{S}$, $N(\mathcal{S}) = \{P \in \mathscr{P}_n : [P, S] = 0 \; \forall S \in \mathcal{S}\}$
- Logical operators $L \in N(\mathcal{S}) \backslash \mathcal{S}$

Quantum error correction 101

Stabiliser codes [Got97]

- Code space $\mathscr{C} = \{ |\psi\rangle \in (\mathbb{C}^2)^{\otimes n} : S|\psi\rangle = |\psi\rangle \ \forall S \in \mathscr{S} \}$
- Normaliser of $\mathscr{S}$, $N(\mathscr{S}) = \{P \in \mathscr{P}_n : [P, S] = 0 \ \forall S \in \mathscr{S}\}$
- Logical operators $L \in N(\mathscr{S}) \backslash \mathscr{S}$
- Code distance of $\mathscr{C}$: $d = \min_{L \in N(\mathscr{S}) \backslash \mathscr{S}} \mathrm{wt}(L)$

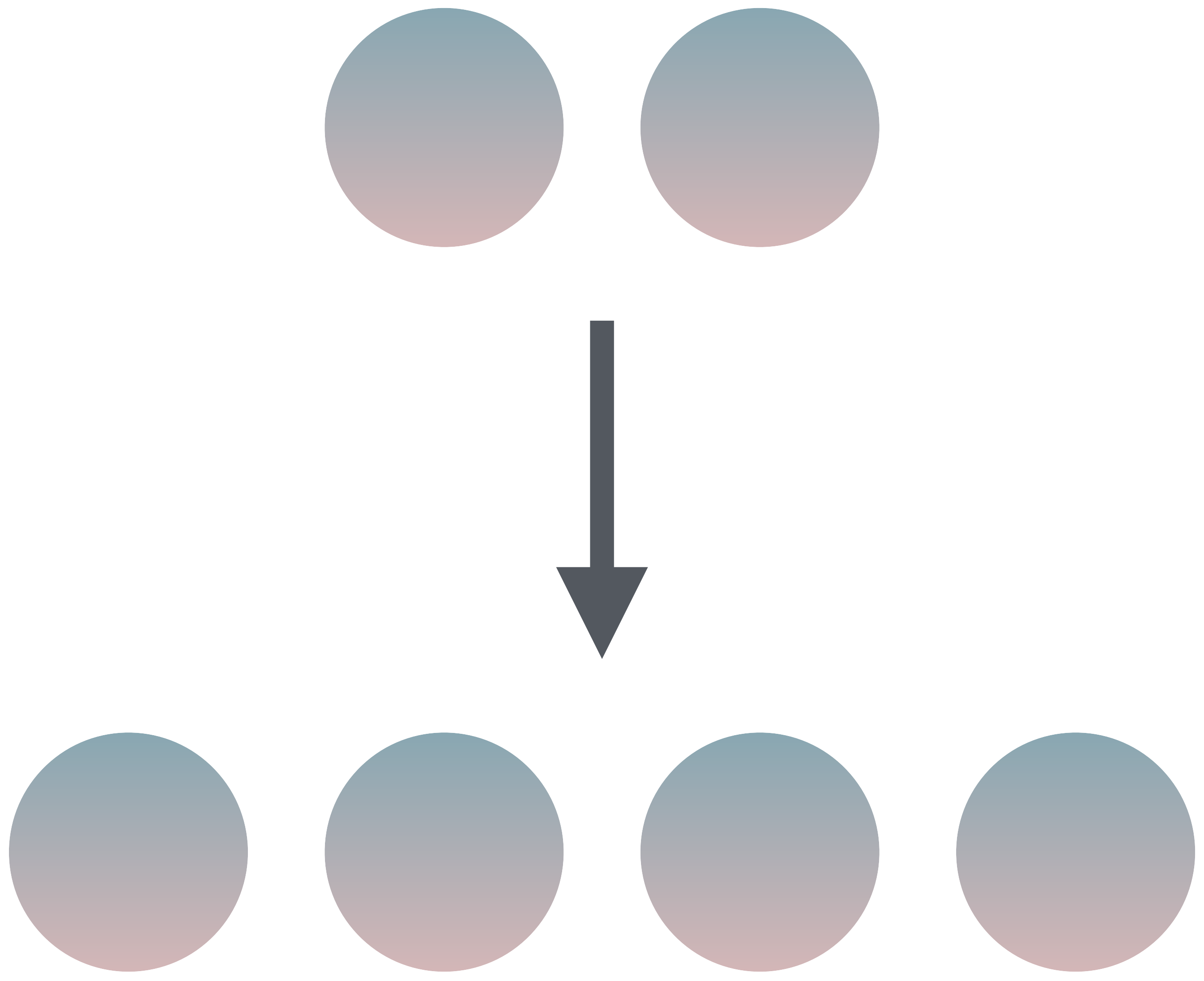
Quantum error correction 101

Stabiliser codes [Got97]

- Code space $\mathscr{C} = \{ |\psi\rangle \in (\mathbb{C}^2)^{\otimes n} : S|\psi\rangle = |\psi\rangle \; \forall S \in \mathcal{S} \}$
- Normaliser of $\mathcal{S}$, $N(\mathcal{S}) = \{P \in \mathscr{P}_n : [P, S] = 0 \; \forall S \in \mathcal{S}\}$
- Logical operators $L \in N(\mathcal{S})\backslash\mathcal{S}$
- Code distance of $\mathscr{C}$: $d = \min_{L \in N(\mathcal{S})\backslash\mathcal{S}} \mathrm{wt}(L)$
- Code parameters $[[n, k, d]]$

Quantum error correction 101

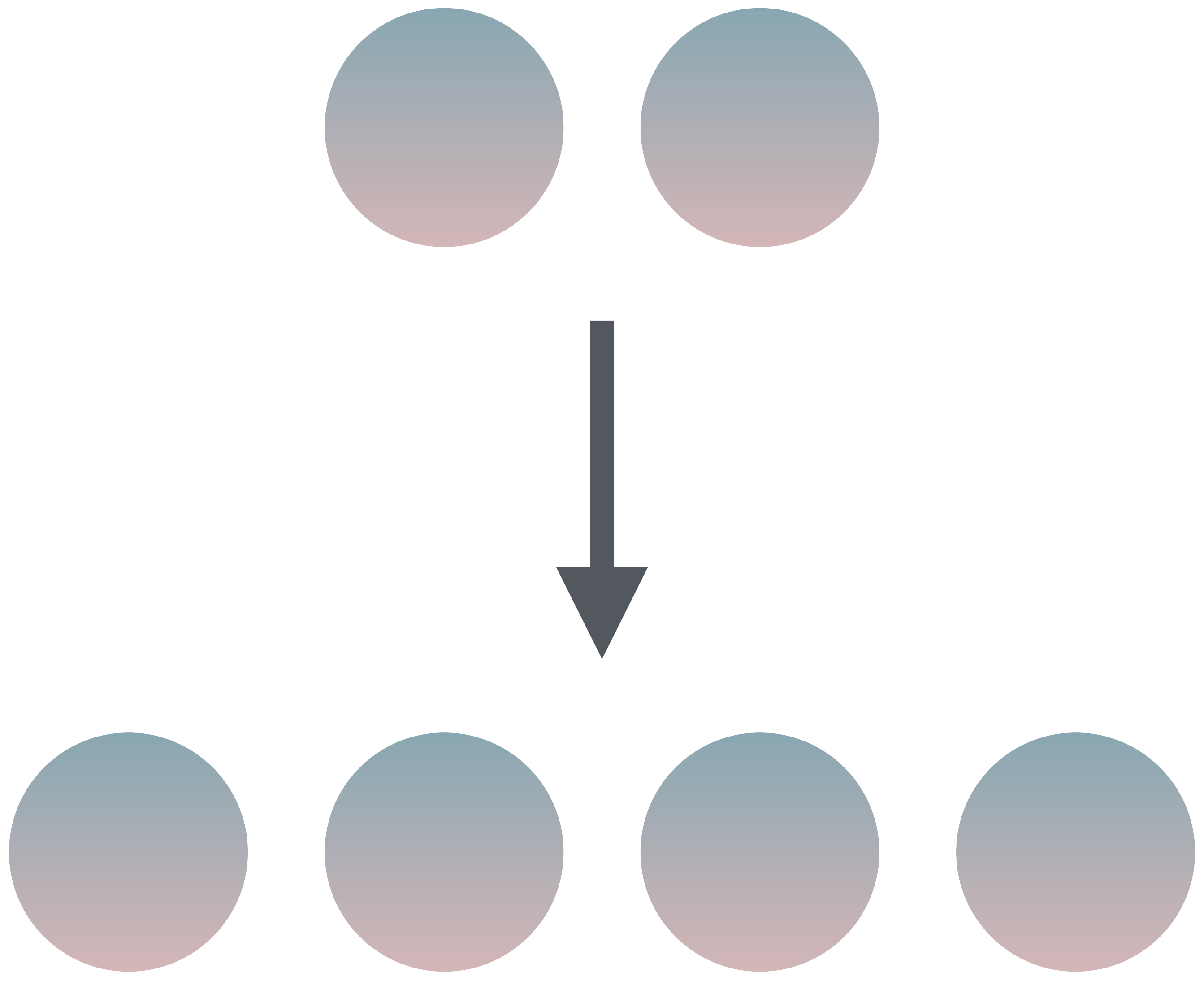
Example 1: ⟦4,2,2⟧ code



Quantum error correction 101

Example 1: $[\![4,2,2]\!]$ code

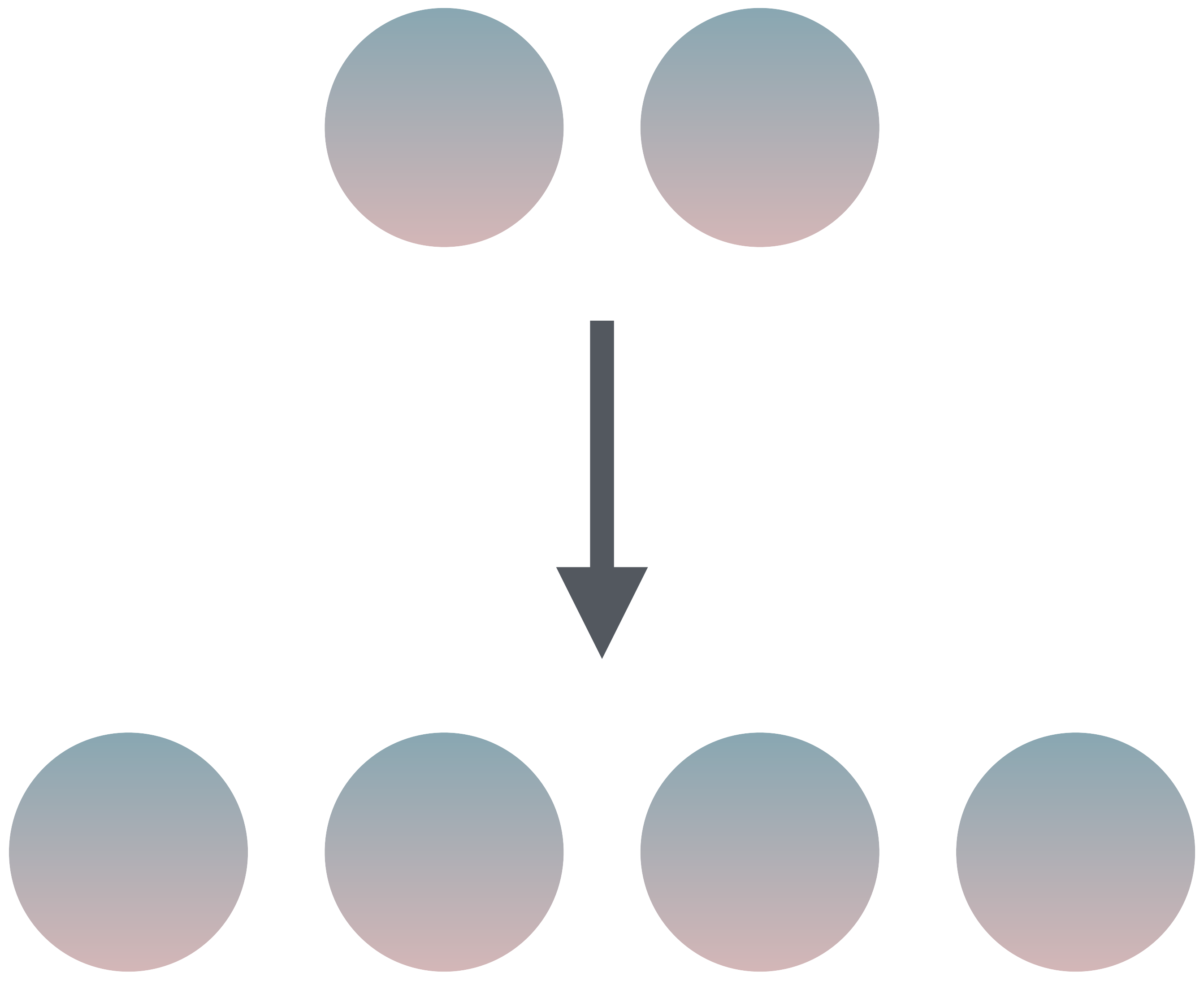
- Stabiliser group $\mathcal{S} = \langle X_1X_2X_3X_4, Z_1Z_2Z_3Z_4 \rangle$



Quantum error correction 101

Example 1: $[\![4,2,2]\!]$ code

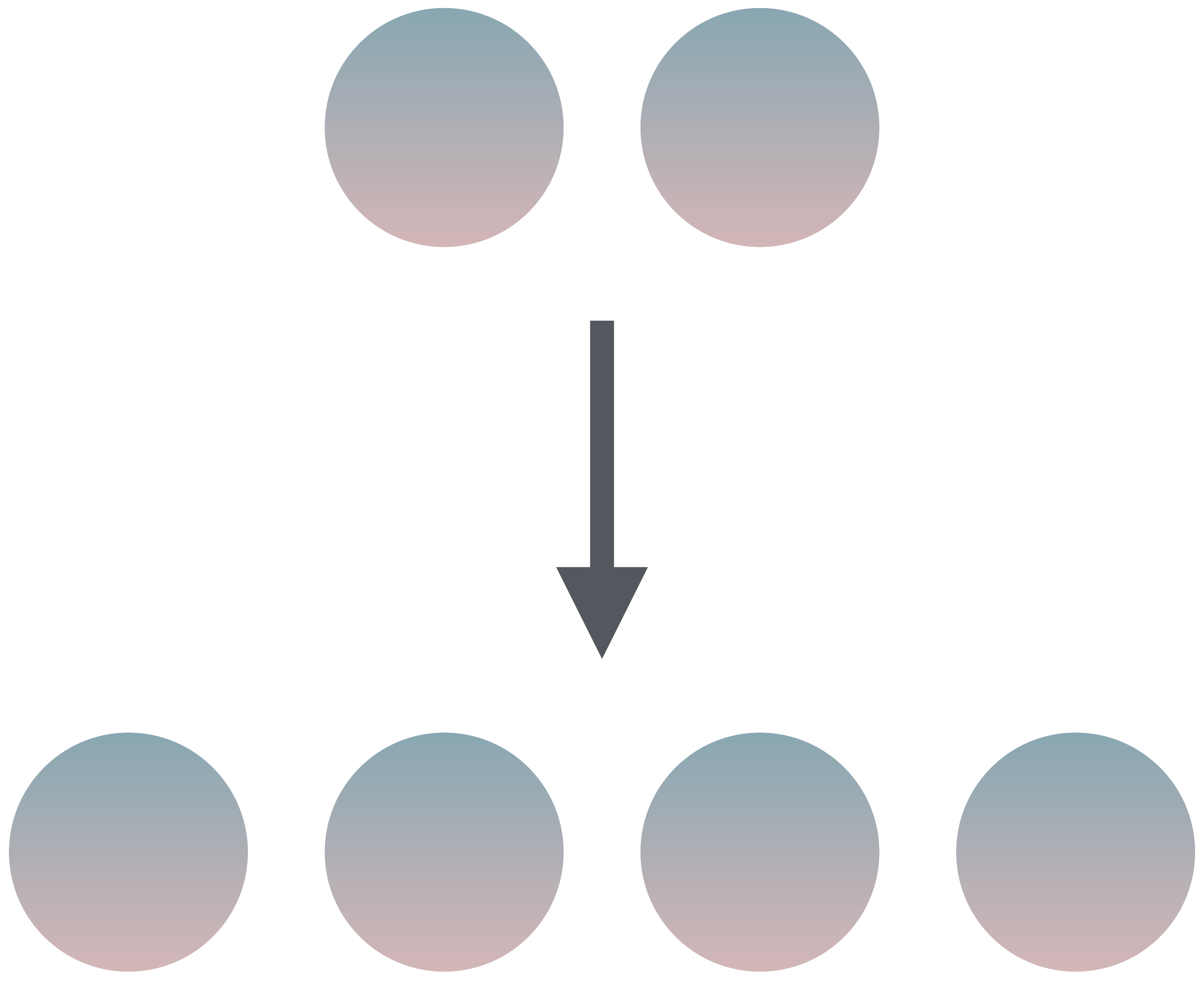
- Stabiliser group $\mathcal{S} = \langle X_1X_2X_3X_4, Z_1Z_2Z_3Z_4 \rangle$
- Logical operators



Quantum error correction 101

Example 1: $[\![4,2,2]\!]$ code

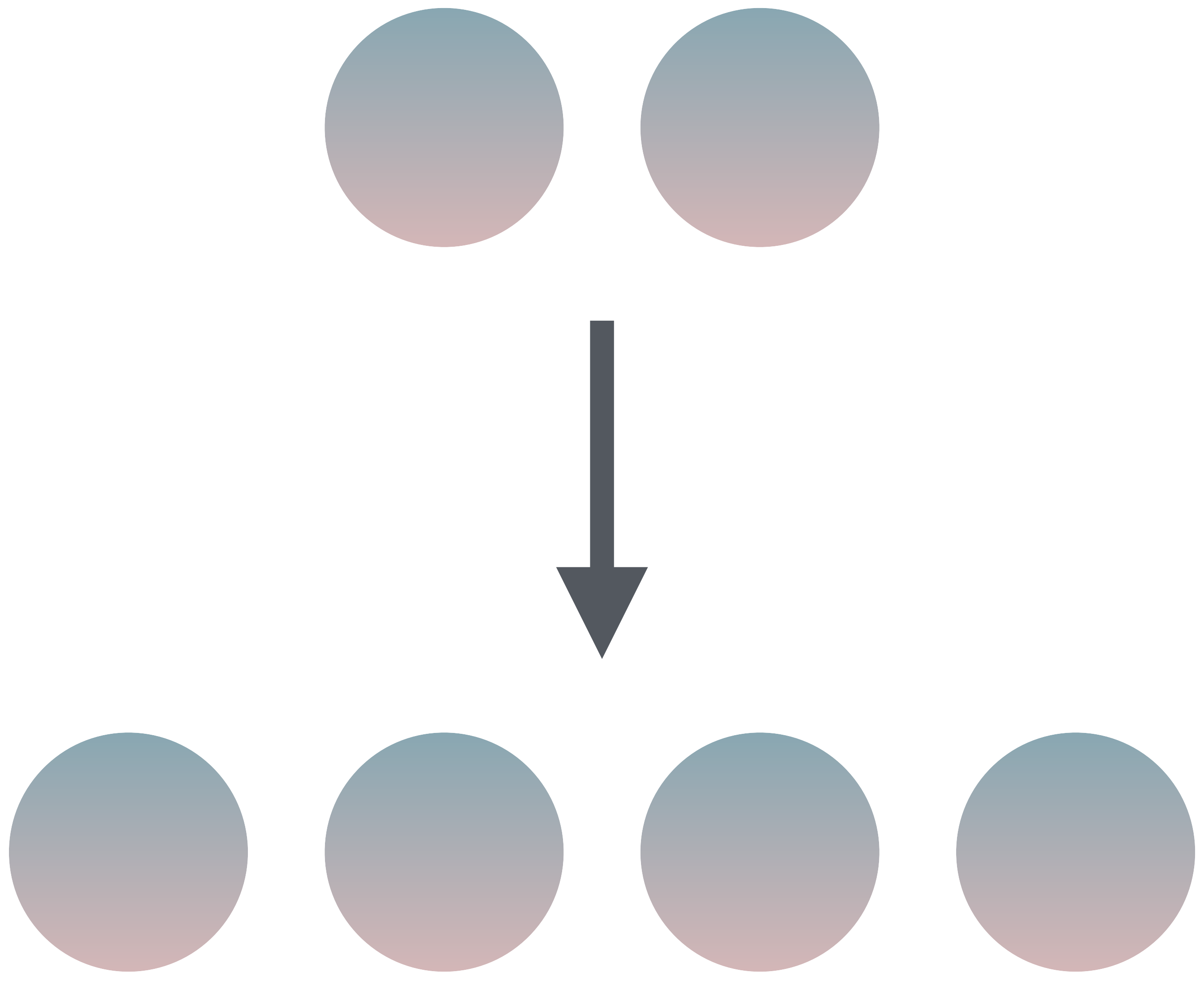
- Stabiliser group $\mathcal{S} = \langle X_1X_2X_3X_4, Z_1Z_2Z_3Z_4 \rangle$
- Logical operators
  - $\overline{X}_1 = X_1X_2, \quad \overline{Z}_1 = Z_1Z_3$



Quantum error correction 101

Example 1: $[\![4,2,2]\!]$ code

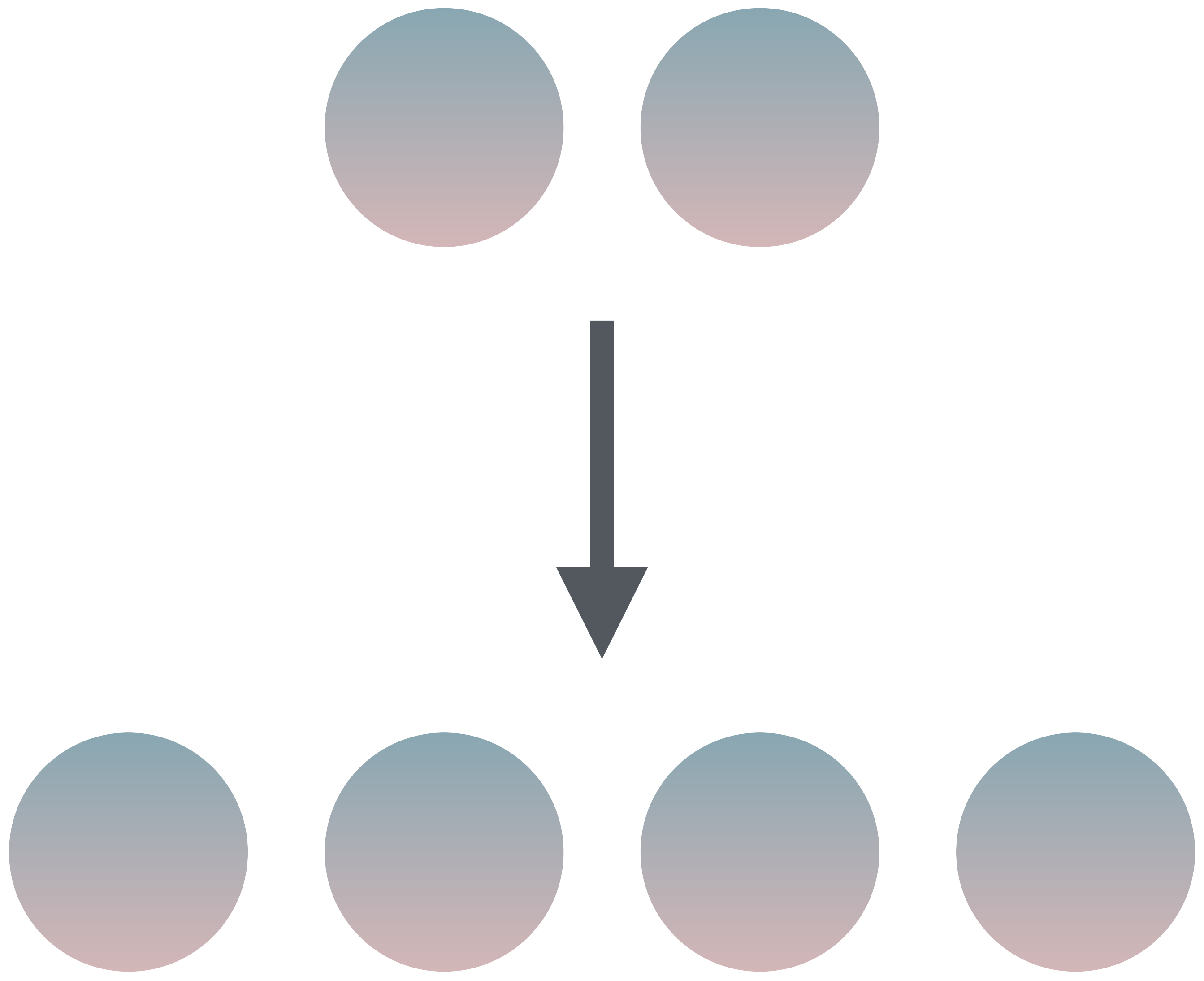
- Stabiliser group $\mathcal{S} = \langle X_1X_2X_3X_4, Z_1Z_2Z_3Z_4 \rangle$
- Logical operators
  - $\overline{X}_1 = X_1X_2, \quad \overline{Z}_1 = Z_1Z_3$
  - $\overline{X}_2 = X_1X_3, \quad \overline{Z}_2 = Z_1Z_2$



Quantum error correction 101

Example 1: ⟦4,2,2⟧ code

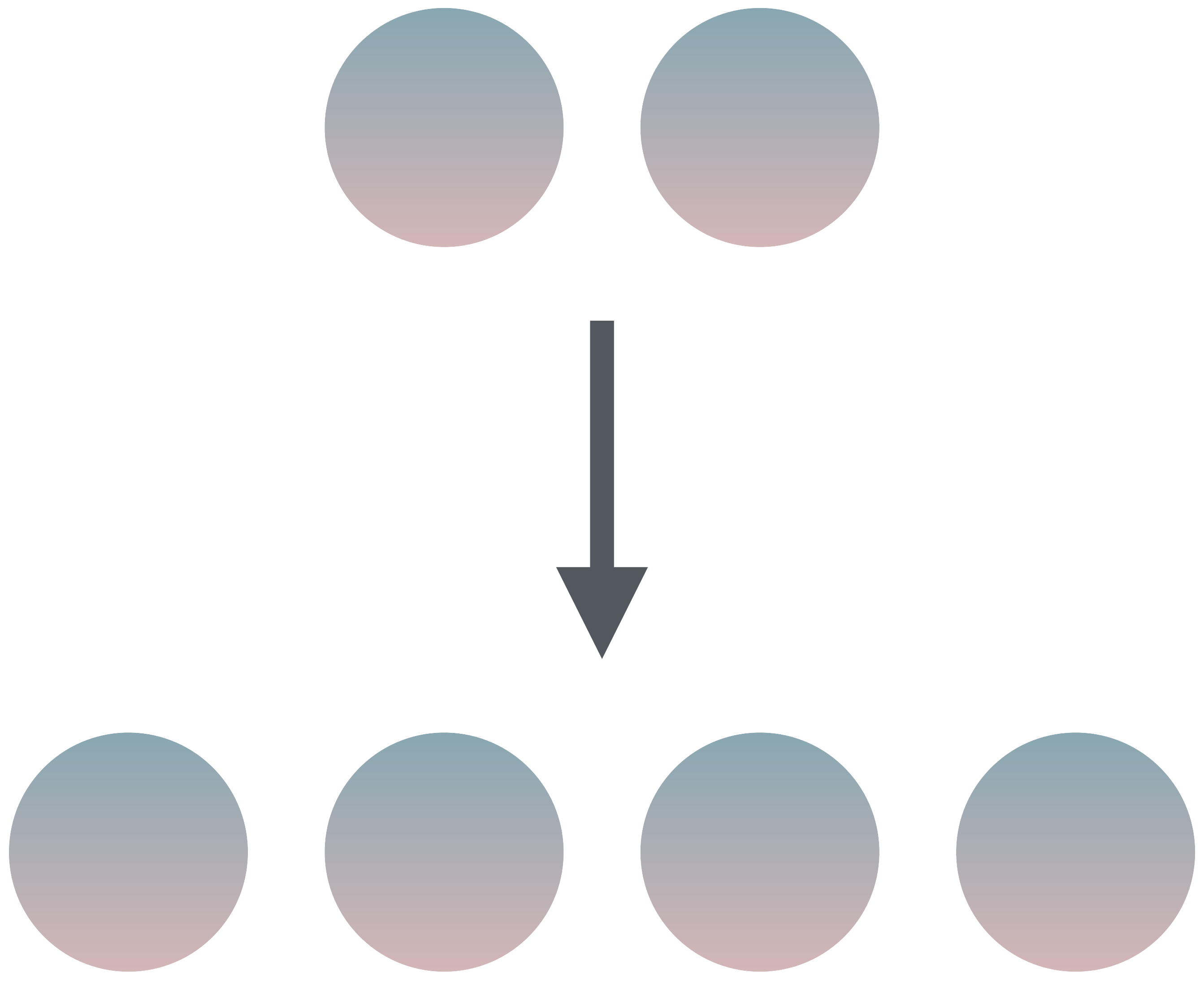
- Stabiliser group $\mathcal{S} = \langle X_1X_2X_3X_4, Z_1Z_2Z_3Z_4 \rangle$
- Logical operators
  - $\overline{X}_1 = X_1X_2, \quad \overline{Z}_1 = Z_1Z_3$
  - $\overline{X}_2 = X_1X_3, \quad \overline{Z}_2 = Z_1Z_2$
- Encoded states



Quantum error correction 101

Example 1: $[\![4,2,2]\!]$ code

- Stabiliser group $\mathcal{S} = \langle X_1X_2X_3X_4, Z_1Z_2Z_3Z_4 \rangle$
- Logical operators
  - $\overline{X}_1 = X_1X_2, \quad \overline{Z}_1 = Z_1Z_3$
  - $\overline{X}_2 = X_1X_3, \quad \overline{Z}_2 = Z_1Z_2$
- Encoded states
  - $|\overline{00}\rangle = \frac{1}{\sqrt{2}}\left(|0000\rangle + |1111\rangle\right), \quad |\overline{10}\rangle = \frac{1}{\sqrt{2}}\left(|1100\rangle + |0011\rangle\right), \quad \ldots$



Quantum error correction 101

More on logical gates

Quantum error correction 101

More on logical gates

- Logical gates map encoded states to encoded states

Quantum error correction 101

More on logical gates

- Logical gates map encoded states to encoded states
- Want fault-tolerant logical gates

Quantum error correction 101

More on logical gates

- Logical gates map encoded states to encoded states
- Want fault-tolerant logical gates
- Clifford gates map Pauli operators to Pauli operators

Quantum error correction 101

More on logical gates

- Logical gates map encoded states to encoded states
- Want fault-tolerant logical gates
- Clifford gates map Pauli operators to Pauli operators
    - Easy to implement fault-tolerantly

Quantum error correction 101

More on logical gates

- Logical gates map encoded states to encoded states
- Want fault-tolerant logical gates
- Clifford gates map Pauli operators to Pauli operators
  - Easy to implement fault-tolerantly
  - … but can be efficiently simulated classically

Quantum error correction 101

More on logical gates

- Logical gates map encoded states to encoded states
- Want fault-tolerant logical gates
- Clifford gates map Pauli operators to Pauli operators
  - Easy to implement fault-tolerantly
  - ... but can be efficiently simulated classically
- Clifford + one non-Clifford = universal quantum computation

Quantum error correction 101

Example 2: [[4,2,2]] code Clifford gate

Quantum error correction 101

Example 2: $[\![4,2,2]\!]$ code Clifford gate

- Stabiliser group $\mathcal{S} = \langle X_1X_2X_3X_4, Z_1Z_2Z_3Z_4 \rangle$

Quantum error correction 101

Example 2: $[\![4,2,2]\!]$ code Clifford gate

- Stabiliser group $\mathcal{S} = \langle X_1X_2X_3X_4, Z_1Z_2Z_3Z_4 \rangle$
- Logical operators $\overline{X}_1 = X_1X_2, \quad \overline{Z}_1 = Z_1Z_3, \quad \overline{X}_2 = X_1X_3, \quad \overline{Z}_2 = Z_1Z_2$

Quantum error correction 101

Example 2: $[\![4,2,2]\!]$ code Clifford gate

- Stabiliser group $\mathcal{S} = \langle X_1X_2X_3X_4, Z_1Z_2Z_3Z_4 \rangle$
- Logical operators $\overline{X}_1 = X_1X_2, \quad \overline{Z}_1 = Z_1Z_3, \quad \overline{X}_2 = X_1X_3, \quad \overline{Z}_2 = Z_1Z_2$
- Hadamard gate $H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, $HXH^\dagger = Z$, $HZH^\dagger = X$

Quantum error correction 101

Example 2: $[[4,2,2]]$ code Clifford gate

- Stabiliser group $\mathcal{S} = \langle X_1X_2X_3X_4, Z_1Z_2Z_3Z_4 \rangle$
- Logical operators $\overline{X}_1 = X_1X_2, \quad \overline{Z}_1 = Z_1Z_3, \quad \overline{X}_2 = X_1X_3, \quad \overline{Z}_2 = Z_1Z_2$
- Hadamard gate $H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, $HXH^\dagger = Z$, $HZH^\dagger = X$
- Logical gate $H_1H_2H_3H_4$

Quantum error correction 101

Example 2: $[\![4,2,2]\!]$ code Clifford gate

- Stabiliser group $\mathcal{S} = \langle X_1X_2X_3X_4, Z_1Z_2Z_3Z_4 \rangle$
- Logical operators $\overline{X}_1 = X_1X_2, \quad \overline{Z}_1 = Z_1Z_3, \quad \overline{X}_2 = X_1X_3, \quad \overline{Z}_2 = Z_1Z_2$
- Hadamard gate $H = \dfrac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, $HXH^\dagger = Z$, $HZH^\dagger = X$
- Logical gate $H_1H_2H_3H_4$
  - $\overline{H}_1\overline{H}_2\overline{\mathrm{SWAP}}$

The decoding problem

The decoding problem

Pauli errors

The decoding problem

Pauli errors

- Pauli group provides a basis for n-qubit errors

The decoding problem

Pauli errors

- Pauli group provides a basis for n-qubit errors
- $\Longrightarrow$ Pauli error models are a good approximation

The decoding problem

Pauli errors

- Pauli group provides a basis for n-qubit errors
- $\implies$ Pauli error models are a good approximation
  - Bit-flip channel $\mathscr{E}(\rho) = (1-p)\rho + pX\rho X^\dagger$

The decoding problem

Pauli errors

- Pauli group provides a basis for n-qubit errors
- $\Longrightarrow$ Pauli error models are a good approximation
  - Bit-flip channel $\mathscr{E}(\rho) = (1-p)\rho + pX\rho X^\dagger$
  - Depolarising channel $\mathscr{D}(\rho) = (1-p)\rho + \frac{p}{3}\left(X\rho X^\dagger + Y\rho Y^\dagger + Z\rho Z^\dagger\right)$

The decoding problem

Error syndrome

The decoding problem

Error syndrome

- Stabiliser group $\mathcal{S}$ with associated $[\![n, k, d]\!]$ code

The decoding problem

Error syndrome

- Stabiliser group $\mathcal{S}$ with associated $[\![n, k, d]\!]$ code
- Suppose $E \in \mathscr{P}_n$ and $\mathrm{wt}(E) < d$

The decoding problem

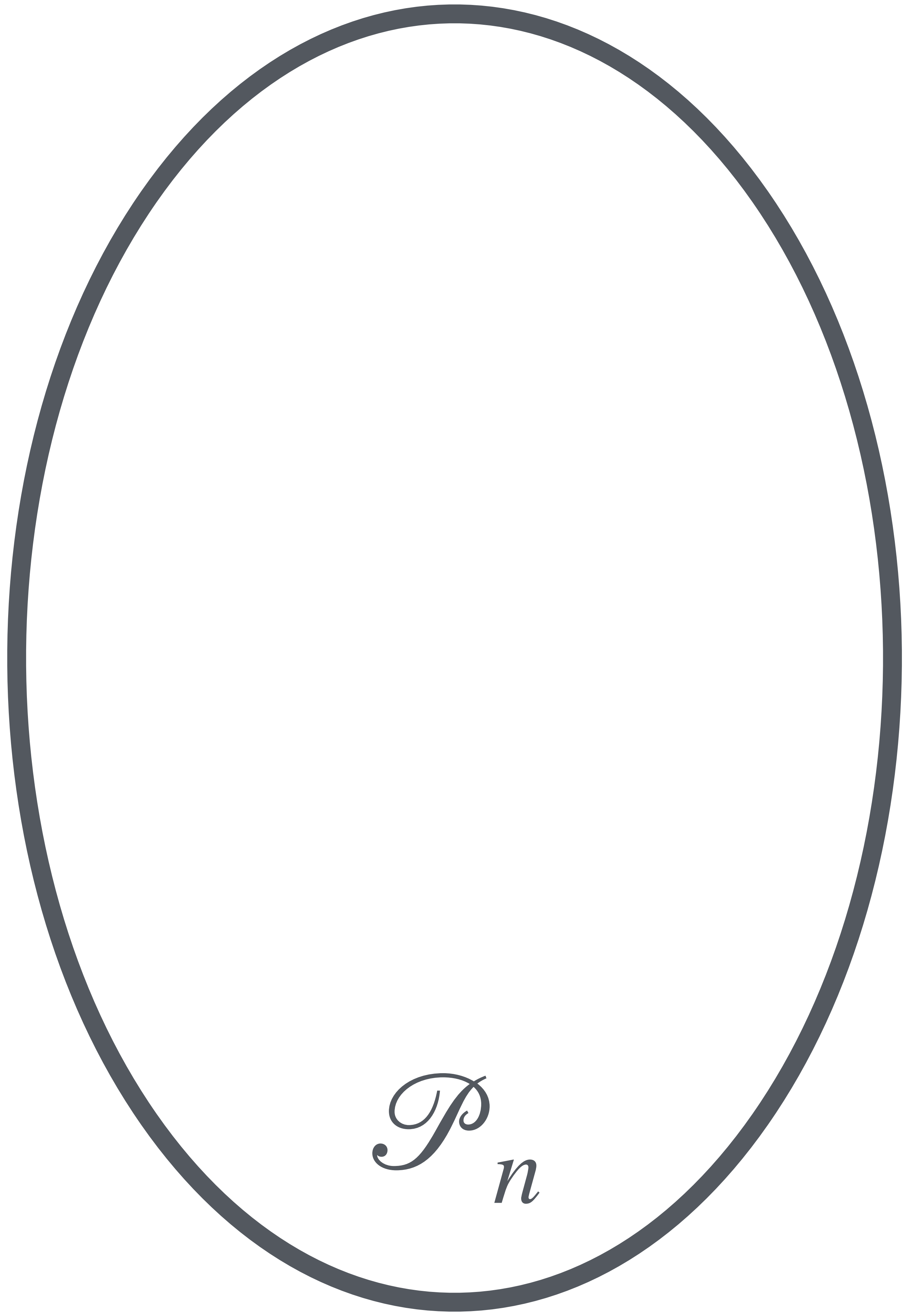
Error syndrome

- Stabiliser group $\mathcal{S}$ with associated $[\![n, k, d]\!]$ code
- Suppose $E \in \mathcal{P}_n$ and $\mathrm{wt}(E) < d$
- Either $E \in \mathcal{S}$ or $\exists S \in \mathcal{S} : \{E, S\} = ES + SE = 0$

The decoding problem

Error syndrome

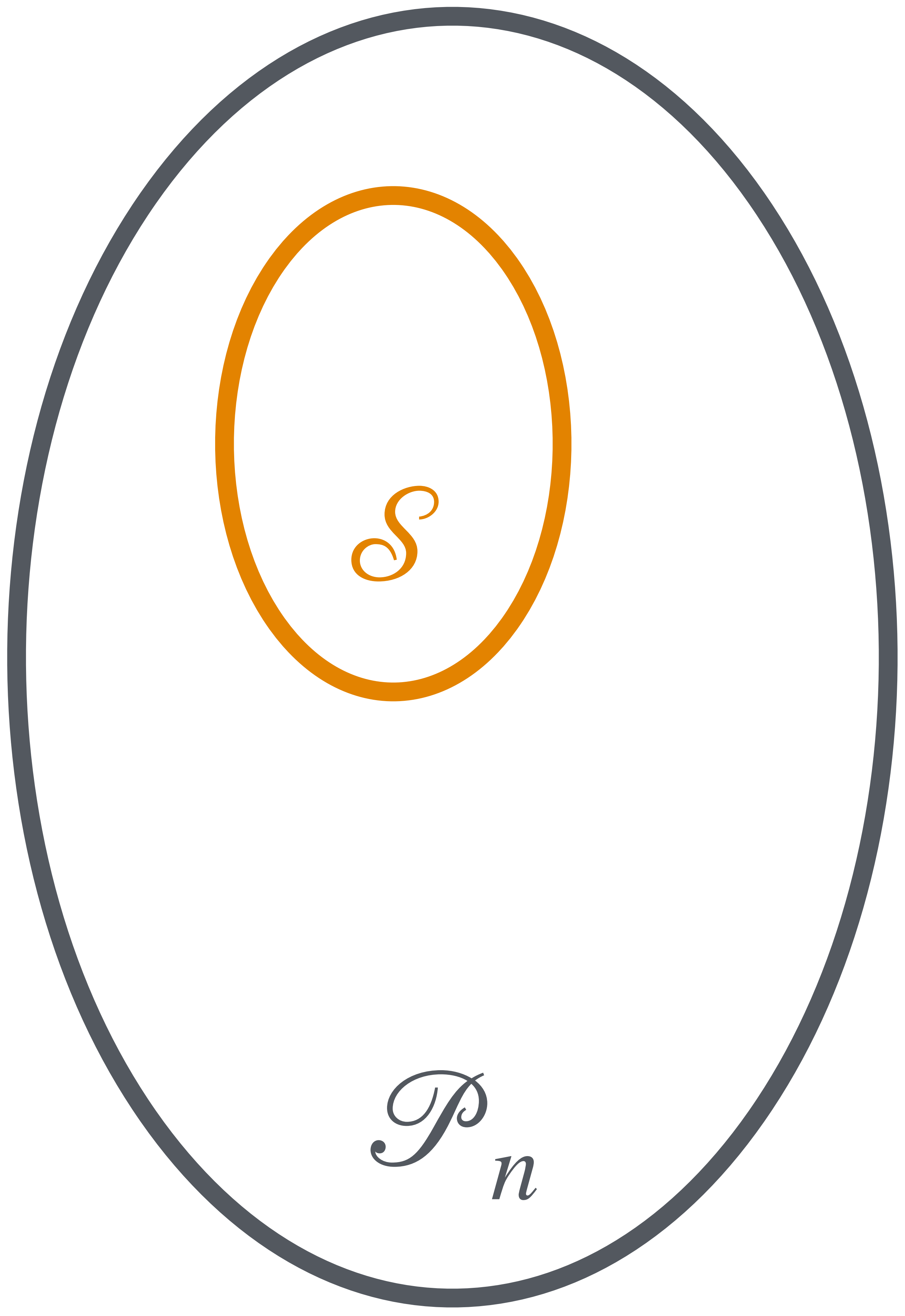
- Stabiliser group $\mathcal{S}$ with associated $[\![n,k,d]\!]$ code
- Suppose $E \in \mathcal{P}_n$ and $\mathrm{wt}(E) < d$
- Either $E \in \mathcal{S}$ or $\exists S \in \mathcal{S} : \{E, S\} = ES + SE = 0$



The decoding problem

Error syndrome

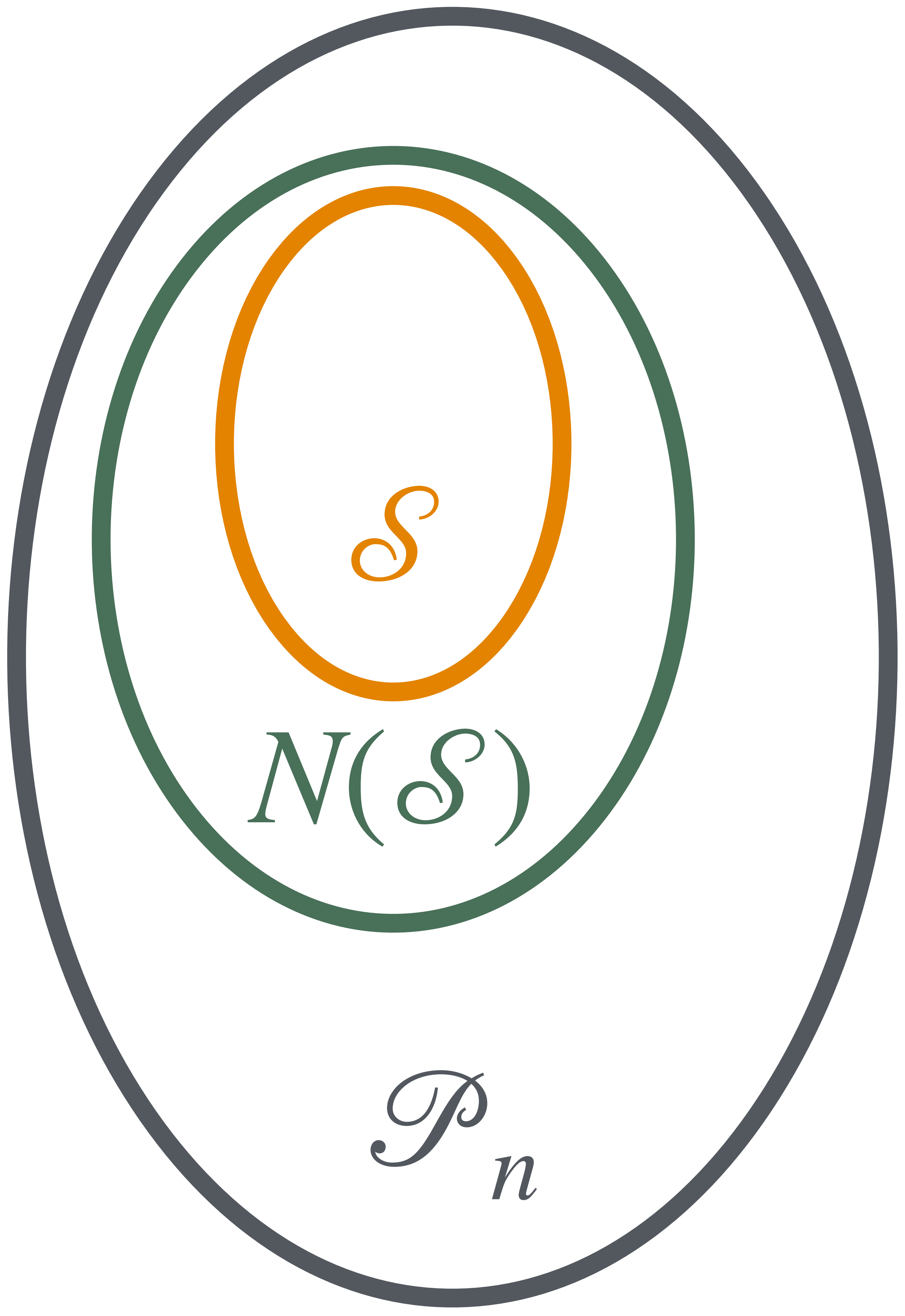
- Stabiliser group $\mathcal{S}$ with associated $[\![n, k, d]\!]$ code
- Suppose $E \in \mathcal{P}_n$ and $\mathrm{wt}(E) < d$
- Either $E \in \mathcal{S}$ or $\exists S \in \mathcal{S} : \{E, S\} = ES + SE = 0$



The decoding problem

Error syndrome

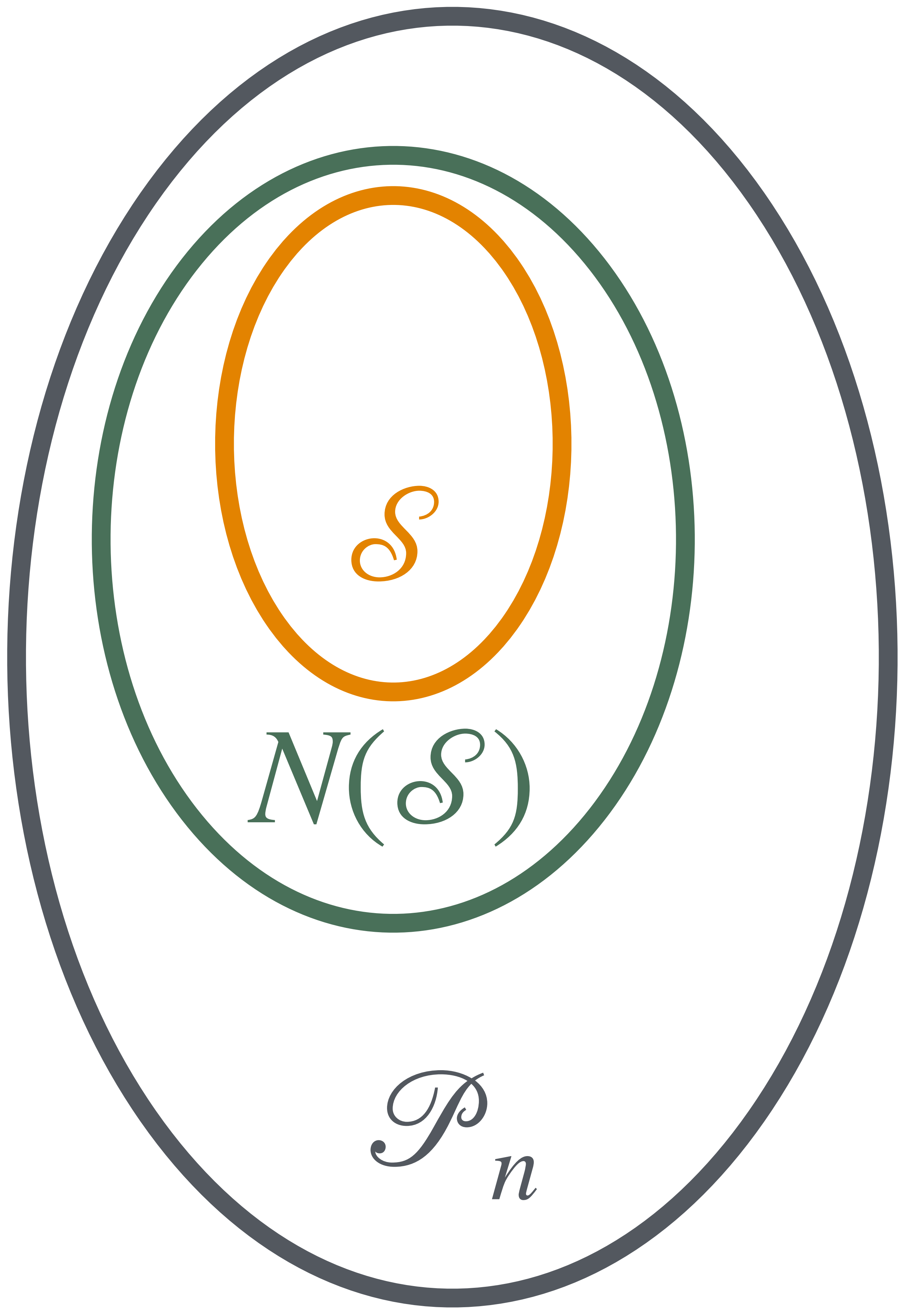
- Stabiliser group $\mathscr{S}$ with associated $[\![n, k, d]\!]$ code
- Suppose $E \in \mathscr{P}_n$ and $\mathrm{wt}(E) < d$
- Either $E \in \mathscr{S}$ or $\exists S \in \mathscr{S} : \{E, S\} = ES + SE = 0$



The decoding problem

Error syndrome

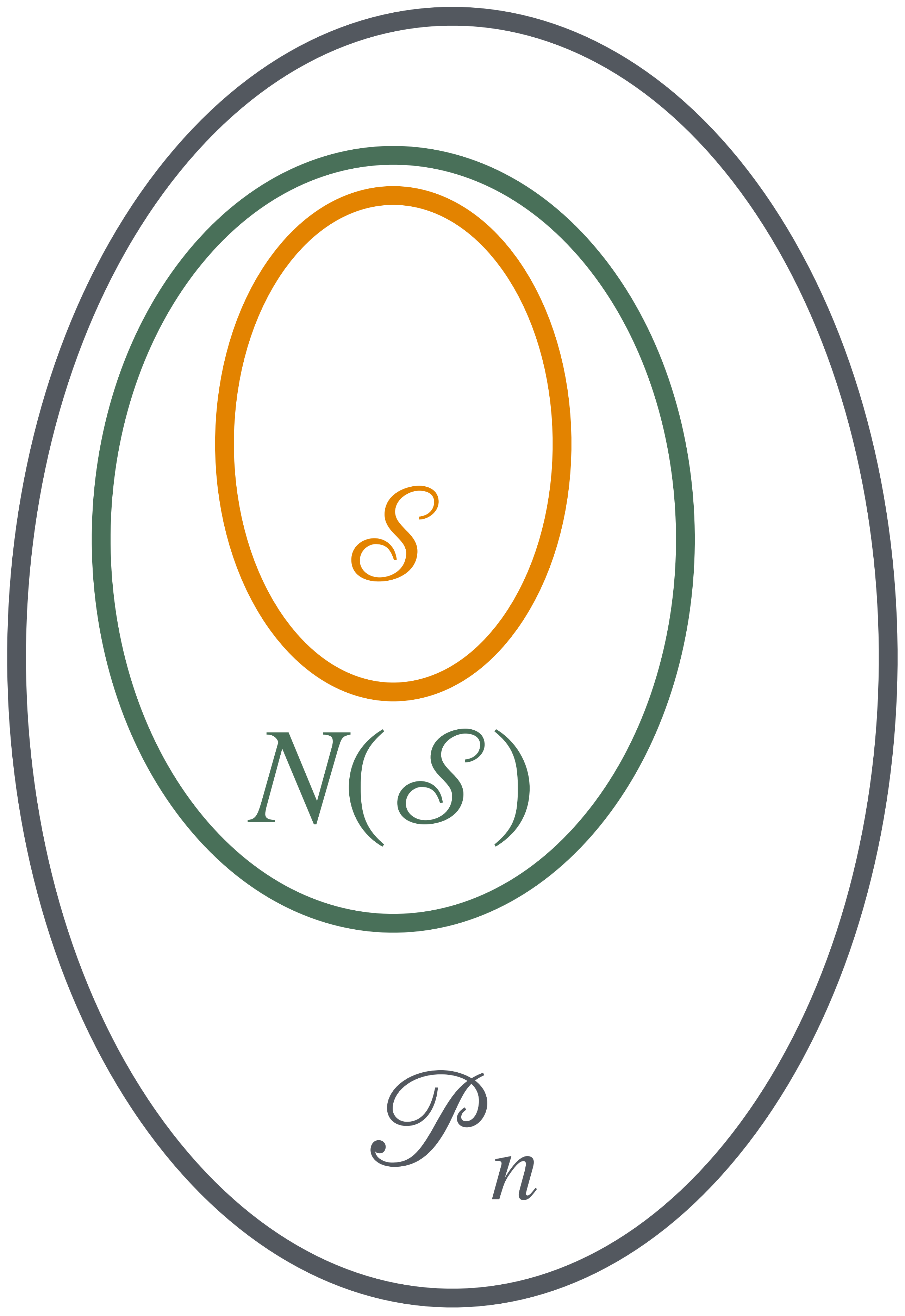
- Stabiliser group $\mathscr{S}$ with associated $[\![n, k, d]\!]$ code
- Suppose $E \in \mathscr{P}_n$ and $\mathrm{wt}(E) < d$
- Either $E \in \mathscr{S}$ or $\exists S \in \mathscr{S} : \{E, S\} = ES + SE = 0$
- Recall $d = \min_{L \in N(\mathscr{S}) \setminus \mathscr{S}} \mathrm{wt}(L)$



The decoding problem

Error syndrome

- Stabiliser group $\mathcal{S}$ with associated $[\![n, k, d]\!]$ code
- Suppose $E \in \mathcal{P}_n$ and $\mathrm{wt}(E) < d$
- Either $E \in \mathcal{S}$ or $\exists S \in \mathcal{S} : \{E, S\} = ES + SE = 0$
  - Recall $d = \min_{L \in N(\mathcal{S}) \setminus \mathcal{S}} \mathrm{wt}(L)$
- $S(E|\overline{\psi}\rangle) = -ES|\overline{\psi}\rangle = -E|\overline{\psi}\rangle$



The decoding problem

Error syndrome

The decoding problem

Error syndrome

- Measure generating set of stabiliser group

The decoding problem

Error syndrome

- Measure generating set of stabiliser group
- Error syndrome: list of measurement outcomes

The decoding problem

Error syndrome

- Measure generating set of stabiliser group
- Error syndrome: list of measurement outcomes
  - (Pauli operators have ± 1 eigenvalues)

The decoding problem

Error syndrome

- Measure generating set of stabiliser group
- Error syndrome: list of measurement outcomes
    - (Pauli operators have ± 1 eigenvalues)
    - If all $+1$, state is in the code space

The decoding problem

Error syndrome

- Measure generating set of stabiliser group
- Error syndrome: list of measurement outcomes
  - (Pauli operators have ± 1 eigenvalues)
  - If all $+1$, state is in the code space
  - -1 outcomes give us information about the error

The decoding problem

The decoding problem

- Stabiliser group $\mathcal{S}$ with associated $[\![n, k, d]\!]$ code

The decoding problem

- Stabiliser group $\mathcal{S}$ with associated $[\![n, k, d]\!]$ code
- Measure m stabiliser generators

The decoding problem

- Stabiliser group $\mathcal{S}$ with associated $[[n, k, d]]$ code
- Measure m stabiliser generators
- The decoding problem

The decoding problem

- Stabiliser group $\mathcal{S}$ with associated $[\![n, k, d]\!]$ code
- Measure m stabiliser generators
- The decoding problem
  - Input: error syndrome $\sigma(E) \in \{-1,1\}^m$

The decoding problem

- Stabiliser group $\mathcal{S}$ with associated $[\![n, k, d]\!]$ code
- Measure m stabiliser generators
- The decoding problem
  - Input: error syndrome $\sigma(E) \in \{-1,1\}^m$
  - Output: recovery operator $R \in \mathcal{P}_n$ such that $\sigma(R) = \sigma(E)$

The decoding problem

- Stabiliser group $\mathcal{S}$ with associated $[\![n, k, d]\!]$ code
- Measure m stabiliser generators
- The decoding problem
  - Input: error syndrome $\sigma(E) \in \{-1,1\}^m$
  - Output: recovery operator $R \in \mathcal{P}_n$ such that $\sigma(R) = \sigma(E)$
- Success if $RE \in \mathcal{S}$

The decoding problem

Example 3: Toric code decoding [Kit97] [DKL+02]

- See the hands-on session later today!
- https://github.com/MikeVasmer/future-horizons-qec-hands-on

The decoding problem

Machine learning solutions

The decoding problem

Machine learning solutions

- [TM17] Neural network decoder for topological codes

The decoding problem

Machine learning solutions

- [TM17] Neural network decoder for topological codes
- [BN18] Convolutional neural network decoder

The decoding problem

Machine learning solutions

- [TM17] Neural network decoder for topological codes
- [BN18] Convolutional neural network decoder
- [SKvN+20] Reinforcement learning decoder

The decoding problem

Machine learning solutions

- [TM17] Neural network decoder for topological codes
- [BN18] Convolutional neural network decoder
- [SKvN+20] Reinforcement learning decoder
- ... and many more!

The decoding problem

Two relevant regimes

The decoding problem

Two relevant regimes

- Near-term (memory experiments)

The decoding problem

Two relevant regimes

- Near-term (memory experiments)
  - All decoding done offline

The decoding problem

Two relevant regimes

- Near-term (memory experiments)
  - All decoding done offline
  - Performance most important

The decoding problem

Two relevant regimes

- Near-term (memory experiments)
  - All decoding done offline
  - Performance most important
  - Excellent results achievable

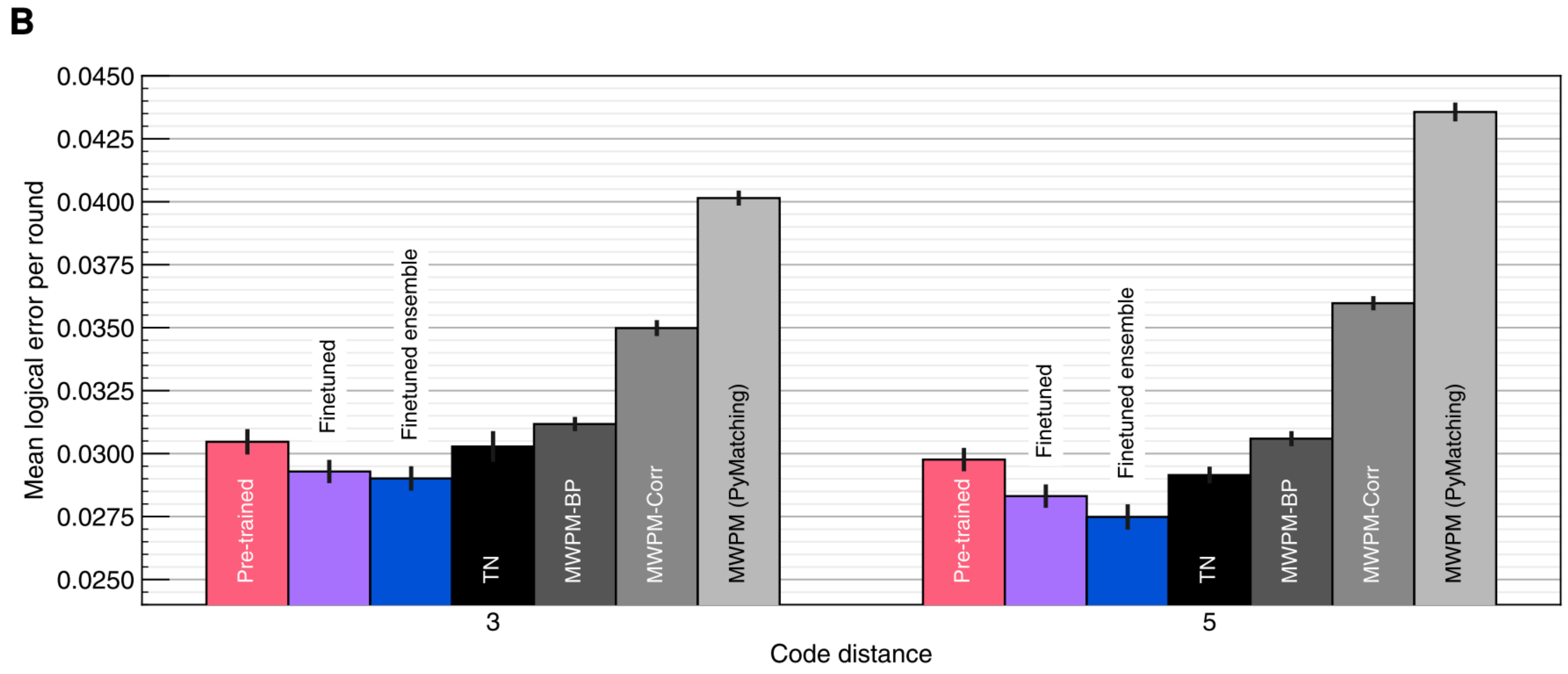


Figure from [BSH+23]

The decoding problem

Two relevant regimes

- Near-term (memory experiments)
  - All decoding done offline
  - Performance most important
  - Excellent results achievable
- Long-term (quantum computation)

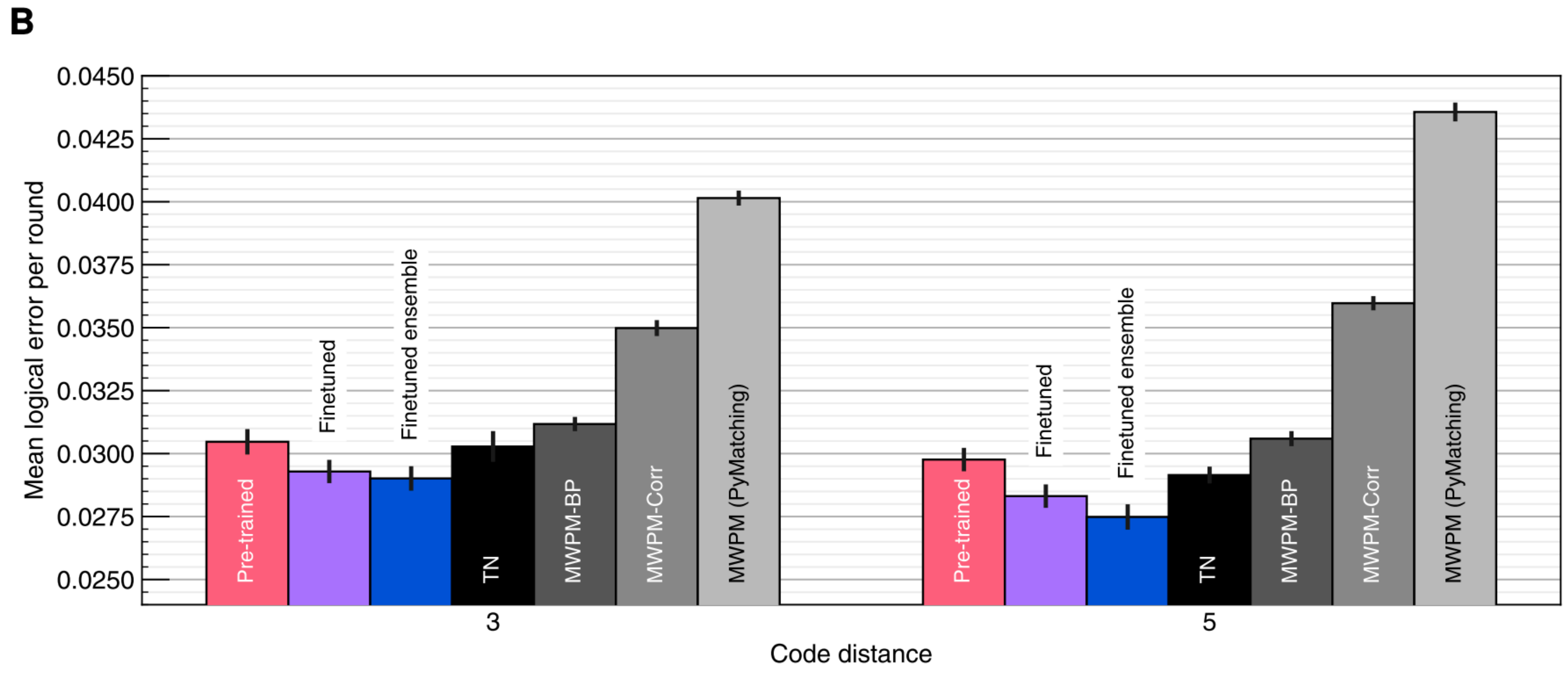


Figure from [BSH+23]

The decoding problem

Two relevant regimes

- Near-term (memory experiments)
  - All decoding done offline
  - Performance most important
  - Excellent results achievable
- Long-term (quantum computation)
  - Real-time decoding

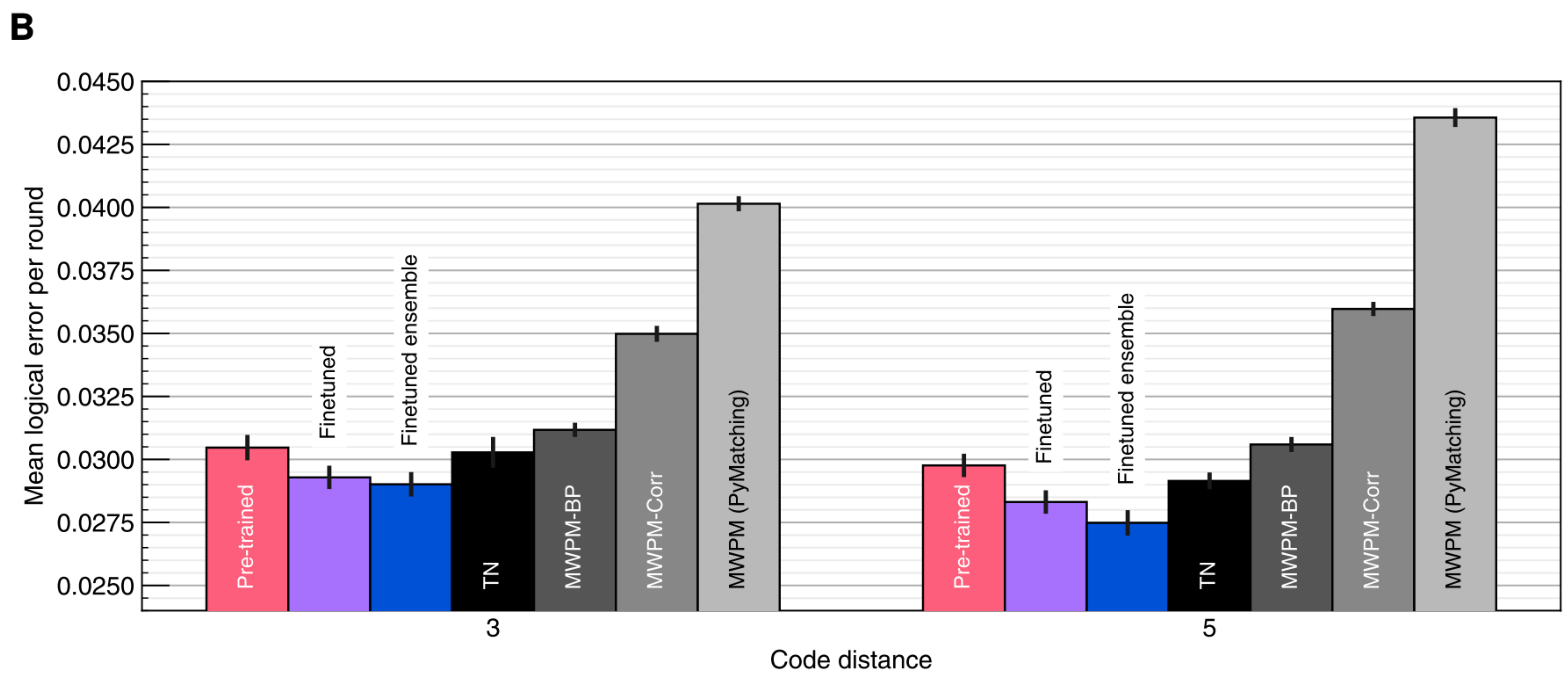


Figure from [BSH+23]

The decoding problem

Two relevant regimes

- Near-term (memory experiments)
  - All decoding done offline
  - Performance most important
  - Excellent results achievable
- Long-term (quantum computation)
  - Real-time decoding
  - Speed most important (μs timescales)

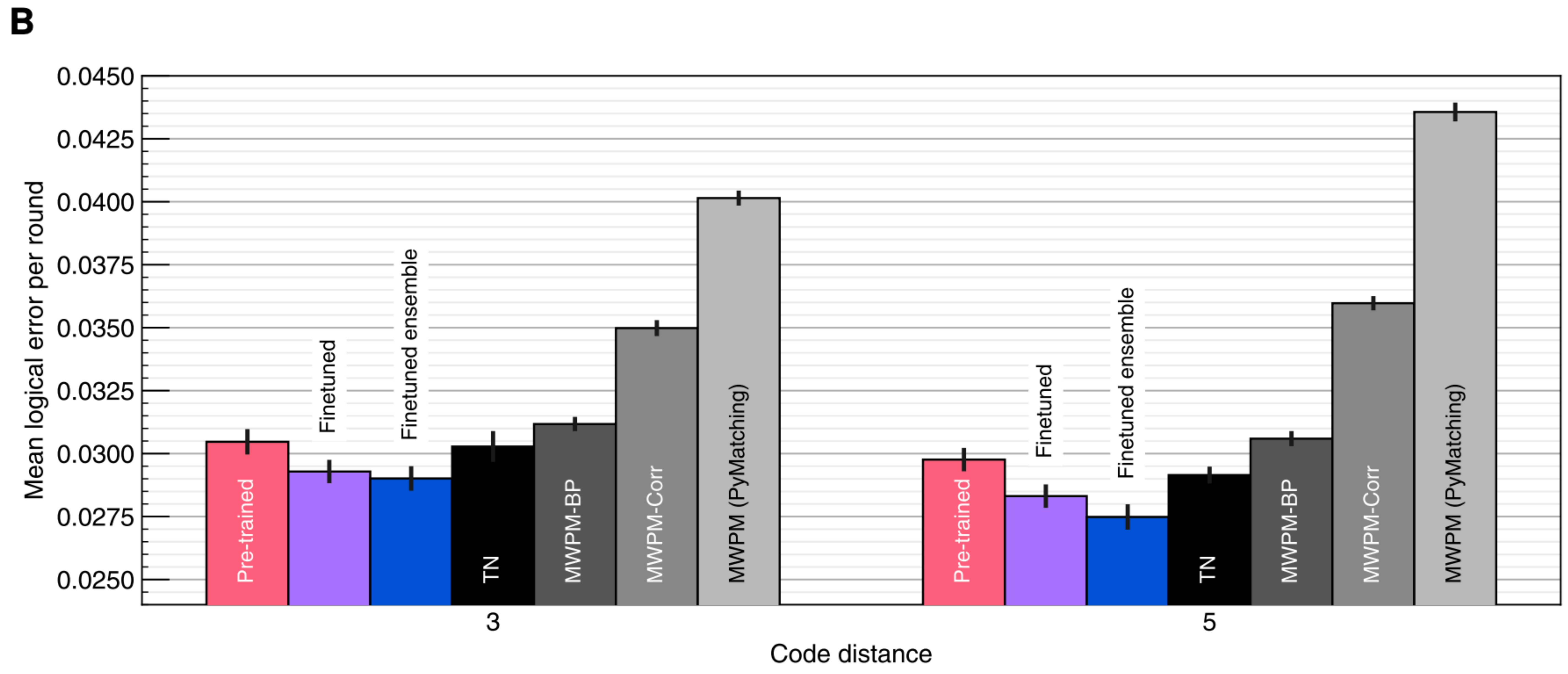


Figure from [BSH+23]

Finding fault-tolerant logical gates

Fault-tolerant logical gates

Definitions

Fault-tolerant logical gates

Definitions

- Logical gates map encoded states to encoded states

Fault-tolerant logical gates

Definitions

- Logical gates map encoded states to encoded states
- Correctable errors for an $[\![n, k, d]\!]$ code: $\{E \in \mathscr{P}_n : \mathrm{wt}(E) \leq t := \lfloor (d-1)/2 \rfloor\}$

Fault-tolerant logical gates

Definitions

- Logical gates map encoded states to encoded states
- Correctable errors for an $[\![n, k, d]\!]$ code: $\{E \in \mathscr{P}_n : \mathrm{wt}(E) \leq t := \lfloor (d-1)/2 \rfloor\}$
- A fault-tolerant logical gate should

Fault-tolerant logical gates

Definitions

- Logical gates map encoded states to encoded states
- Correctable errors for an $[\![n, k, d]\!]$ code: $\{E \in \mathscr{P}_n : \mathrm{wt}(E) \leq t := \lfloor (d-1)/2 \rfloor\}$
- A fault-tolerant logical gate should
  - Map correctable errors to correctable errors

Fault-tolerant logical gates

Definitions

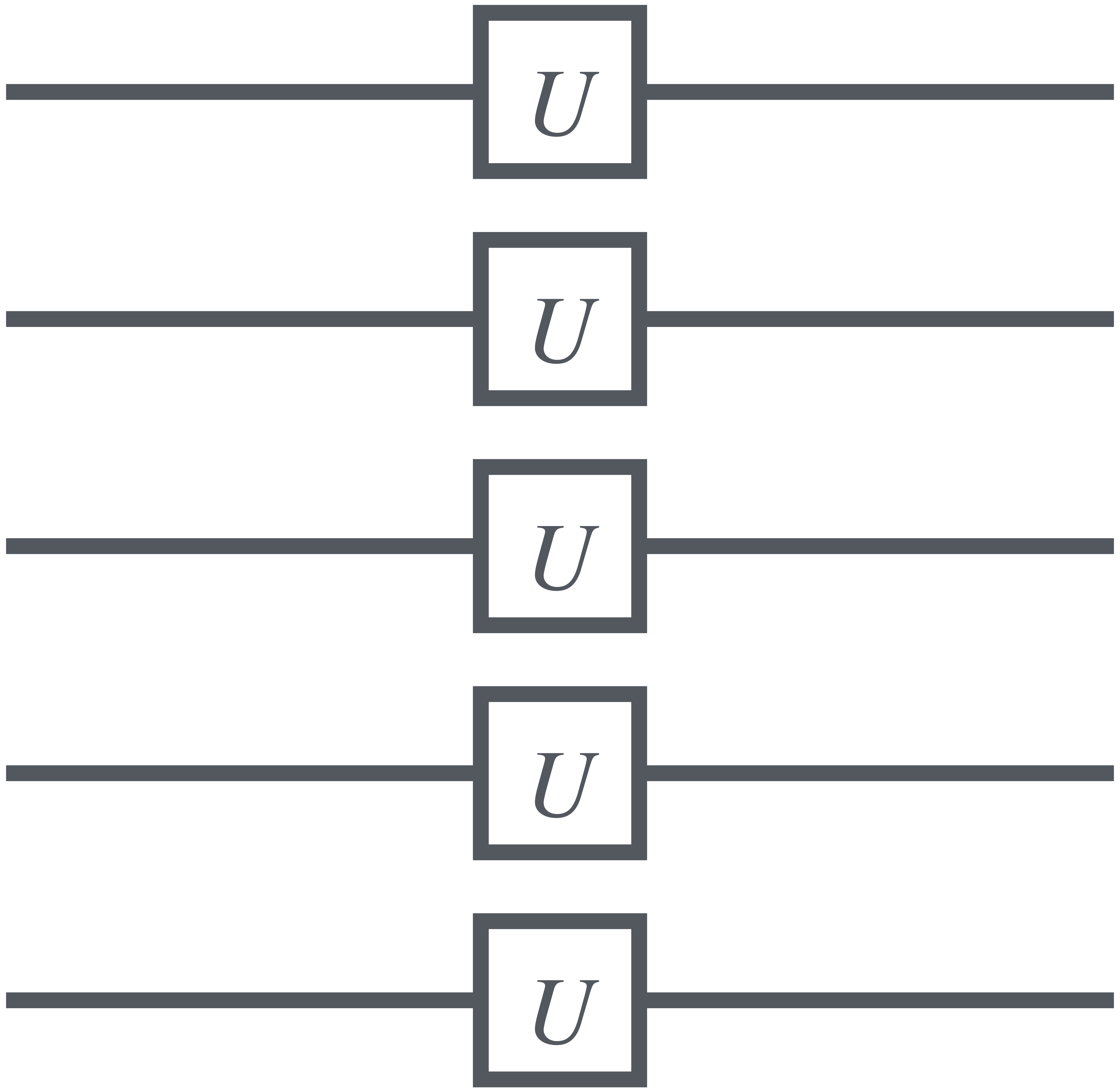
- Logical gates map encoded states to encoded states
- Correctable errors for an $[\![n,k,d]\!]$ code: $\{E \in \mathscr{P}_n : \mathrm{wt}(E) \leq t := \lfloor (d-1)/2 \rfloor\}$
- A fault-tolerant logical gate should
  - Map correctable errors to correctable errors
  - Be robust to t errors occurring during the gate

Fault-tolerant logical gates

Example 4: Transversal gates

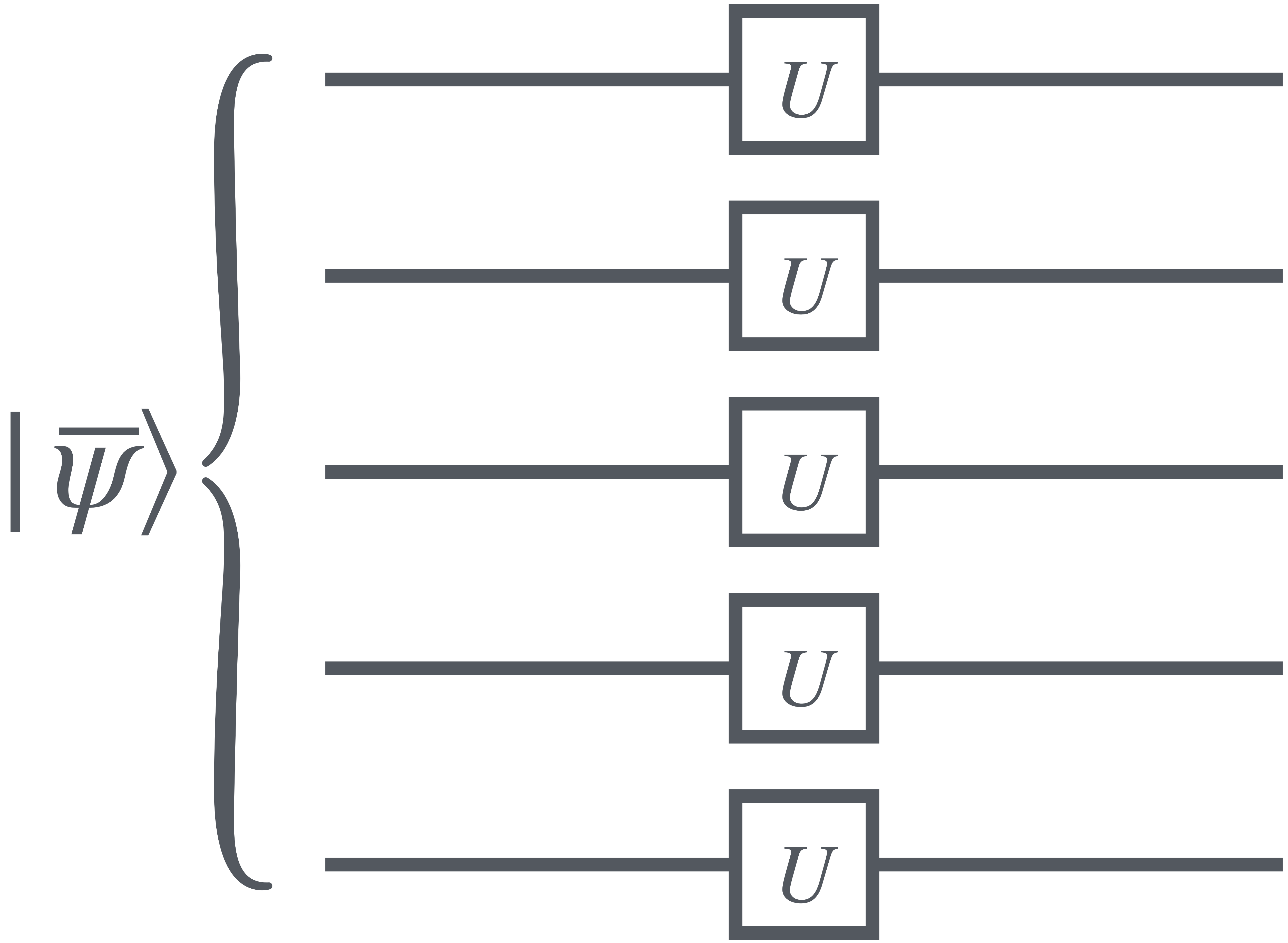
Fault-tolerant logical gates

Example 4: Transversal gates



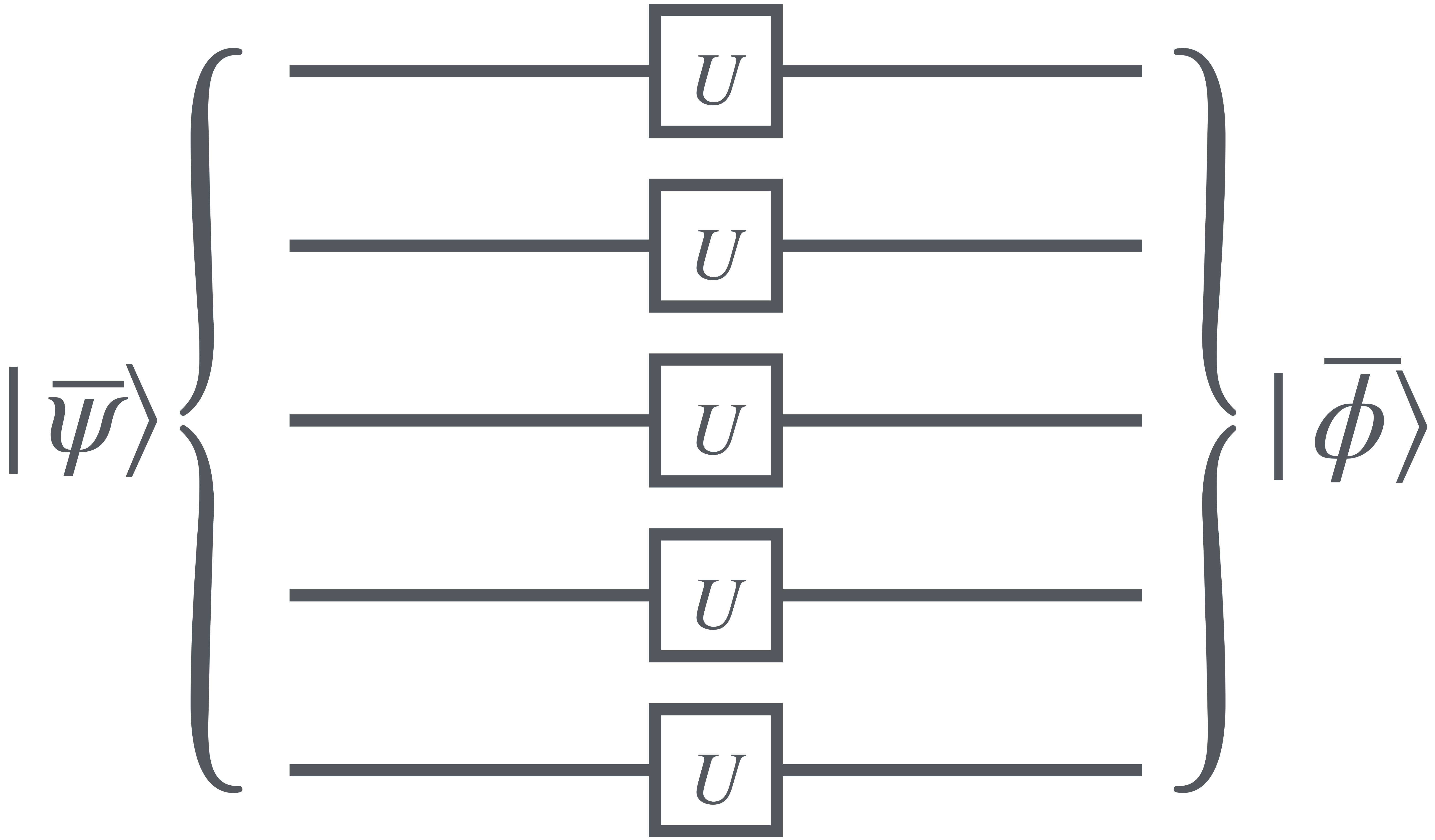
Fault-tolerant logical gates

Example 4: Transversal gates



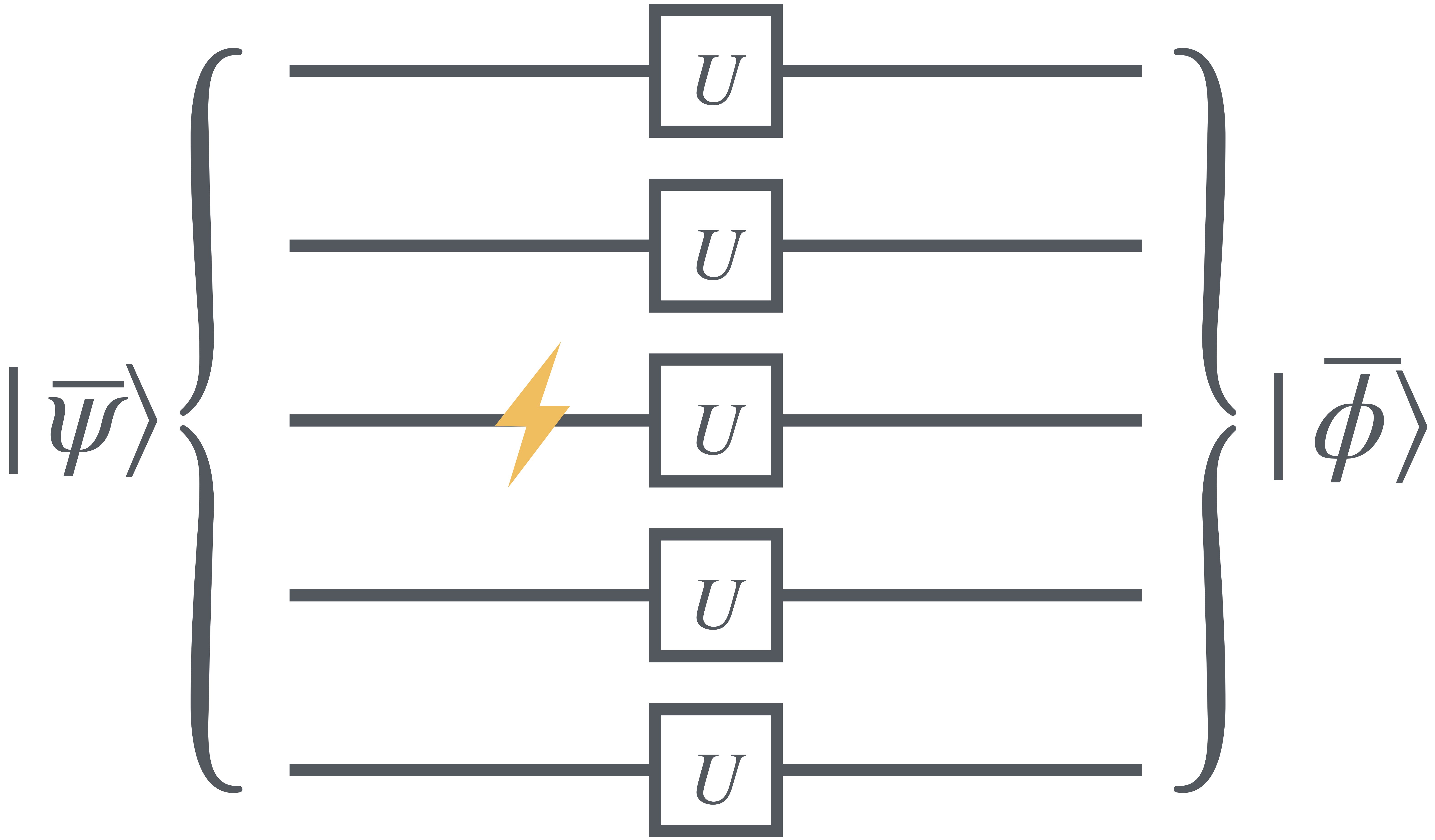
Fault-tolerant logical gates

Example 4: Transversal gates



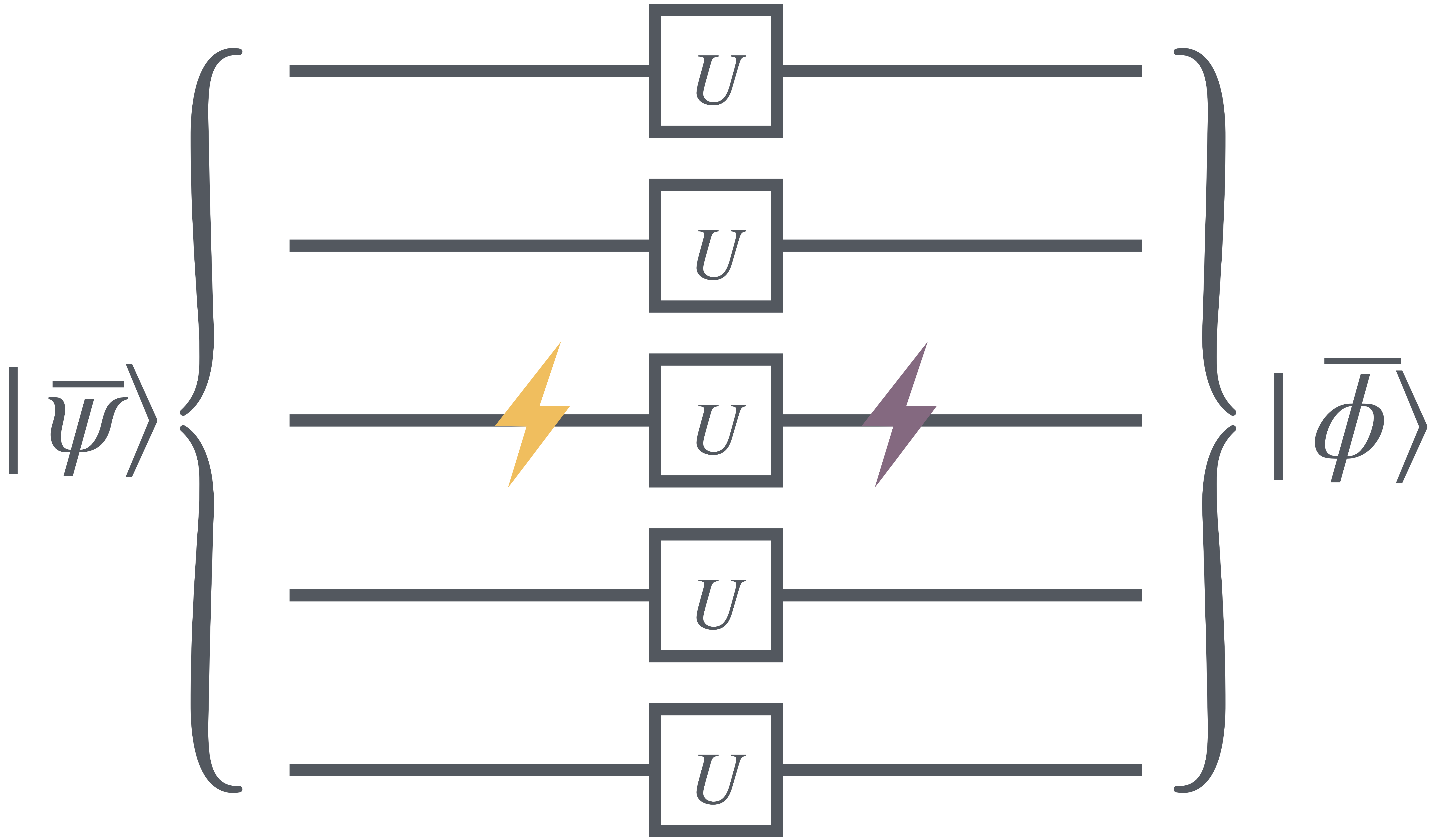
Fault-tolerant logical gates

Example 4: Transversal gates



Fault-tolerant logical gates

Example 4: Transversal gates



Finding fault-tolerant logical gates

Finding fault-tolerant logical gates

- Finding fault-tolerant logical gates

Finding fault-tolerant logical gates

- Finding fault-tolerant logical gates
  - Key problem in quantum error correction theory

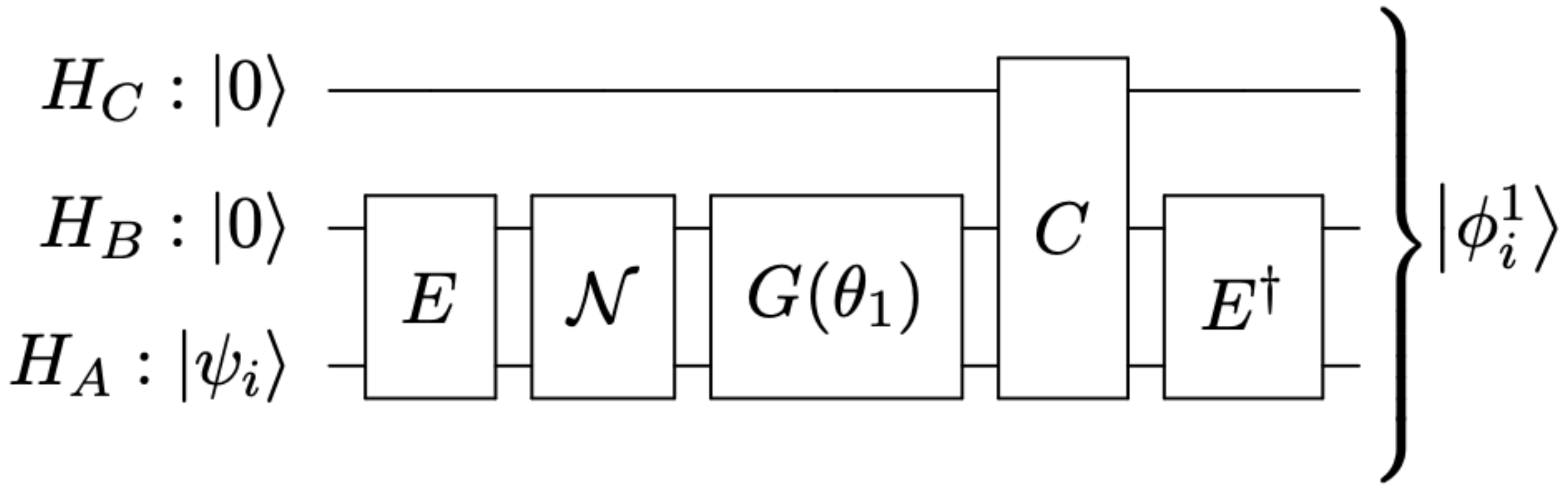
Finding fault-tolerant logical gates

- Finding fault-tolerant logical gates
  - Key problem in quantum error correction theory
  - Especially for codes with many encoded qubits (large k)

Finding fault-tolerant logical gates

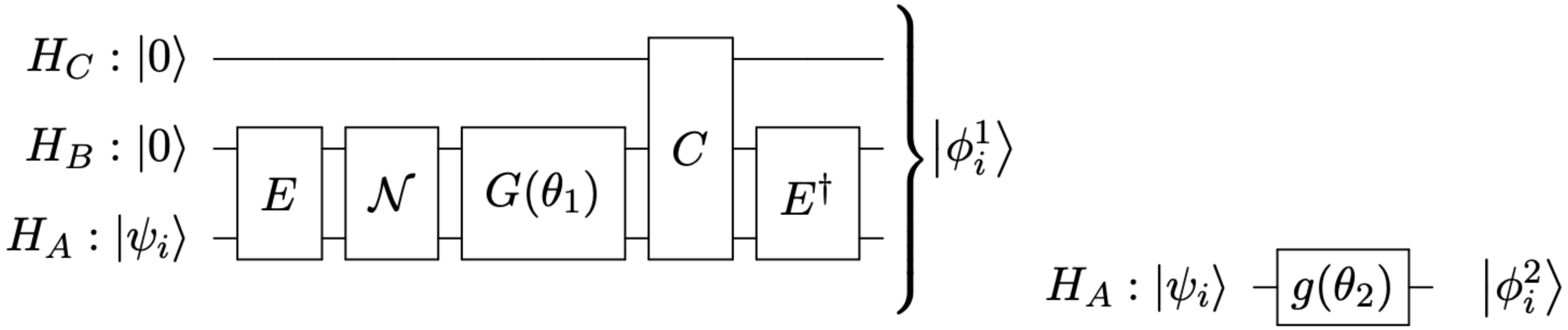
- Finding fault-tolerant logical gates
  - Key problem in quantum error correction theory
  - Especially for codes with many encoded qubits (large k)
- Can we automate the discovery of logical gates? Yes! [C**V**B+21]

Automated discovery of logical gates



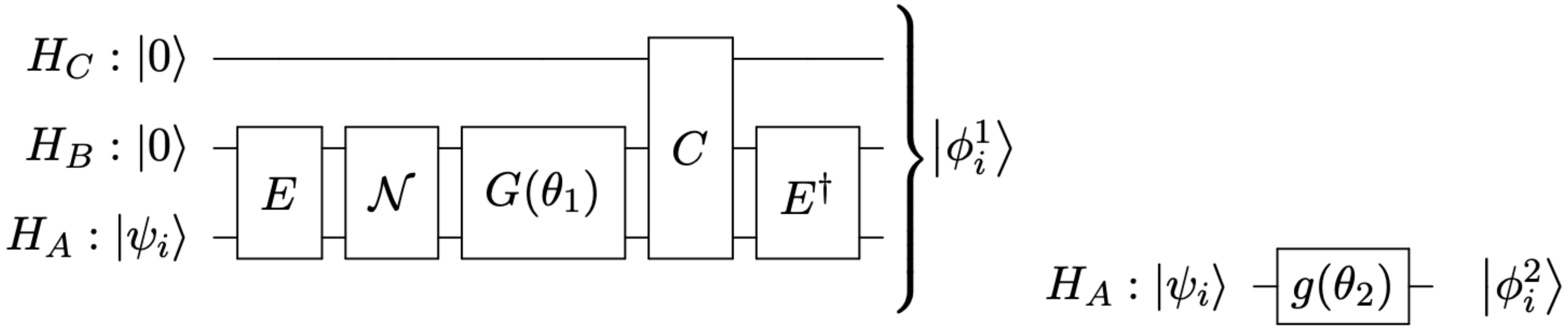
(a) Trial logical circuit

Automated discovery of logical gates



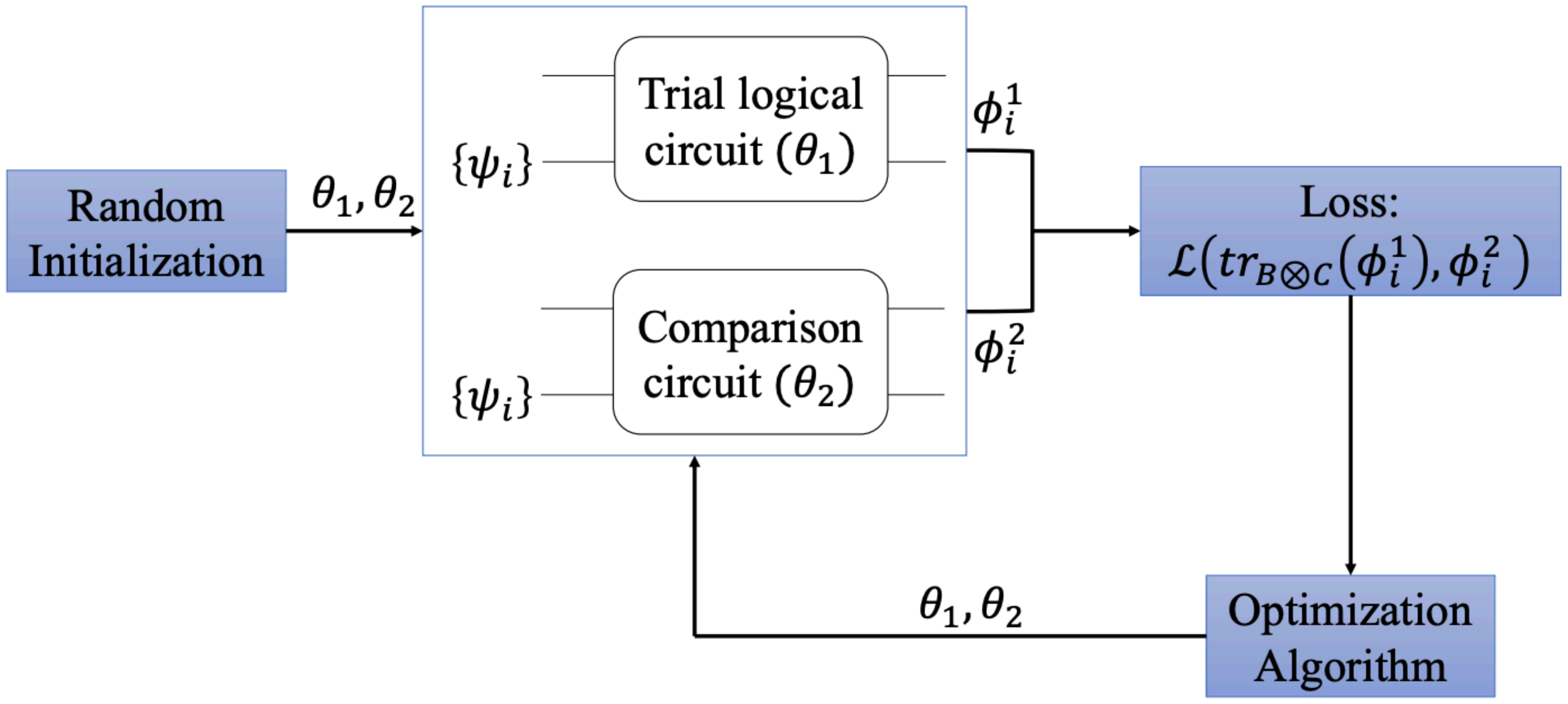
(a) Trial logical circuit (b) Comparison circuit

Automated discovery of logical gates



(a) Trial logical circuit

(b) Comparison circuit



Automated discovery of logical gates

Automated discovery of logical gates

- Rediscovered known logical gates and discovered new logical gates for small codes

Automated discovery of logical gates

- Rediscovered known logical gates and discovered new logical gates for small codes
- Found Pauli, Clifford and non-Clifford gates

Automated discovery of logical gates

- Rediscovered known logical gates and discovered new logical gates for small codes
- Found Pauli, Clifford and non-Clifford gates
- Found entangling gates between code blocks and within one code block

Automated discovery of logical gates

- Rediscovered known logical gates and discovered new logical gates for small codes
- Found Pauli, Clifford and non-Clifford gates
- Found entangling gates between code blocks and within one code block
- Results from classical simulations but method could be run on a quantum computer

Automated discovery of logical gates

- Rediscovered known logical gates and discovered new logical gates for small codes
- Found Pauli, Clifford and non-Clifford gates
- Found entangling gates between code blocks and within one code block
- Results from classical simulations but method could be run on a quantum computer
- Logical gate ansatz limited to transversal gates

Automated discovery of logical gates

- Rediscovered known logical gates and discovered new logical gates for small codes
- Found Pauli, Clifford and non-Clifford gates
- Found entangling gates between code blocks and within one code block
- Results from classical simulations but method could be run on a quantum computer
- Logical gate ansatz limited to transversal gates
    - Many limitations on transversal gates e.g. [EK09] [BK13]

Conclusion

Conclusion

- Quantum error correction overhead = main barrier to achieving useful quantum computing

Conclusion

- Quantum error correction overhead = main barrier to achieving useful quantum computing
- Machine learning has already had success in the decoding problem

Conclusion

- Quantum error correction overhead = main barrier to achieving useful quantum computing
- Machine learning has already had success in the decoding problem
- Potential opportunities for using machine learning to find fault-tolerant logical gates

Conclusion

- Quantum error correction overhead = main barrier to achieving useful quantum computing
- Machine learning has already had success in the decoding problem
- Potential opportunities for using machine learning to find fault-tolerant logical gates
- Things to remember

Conclusion

- Quantum error correction overhead = main barrier to achieving useful quantum computing
- Machine learning has already had success in the decoding problem
- Potential opportunities for using machine learning to find fault-tolerant logical gates
- Things to remember
  - Realistic error model almost always have measurement errors

Conclusion

- Quantum error correction overhead = main barrier to achieving useful quantum computing
- Machine learning has already had success in the decoding problem
- Potential opportunities for using machine learning to find fault-tolerant logical gates
- Things to remember
    - Realistic error model almost always have measurement errors
    - Constraints for different qubit technologies can be very different

References

- [Got97] D. Gottesman, “Stabilizer codes and quantum error correction”, *PhD Thesis (Caltech)* 1997. https://doi.org/10.48550/arXiv.quant-ph/9705052

- [Kit03] A. Yu. Kitaev, “Fault-tolerant quantum computation by anyons”, *Annals of Physics* 303(1), 2-30 (2003). https://doi.org/10.1016/S0003-4916(02)00018-0

- [DKL+02] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill, “Topological quantum memory”, *Journal of Mathematical Physics* 43, 4452-4505 (2002). https://doi.org/10.1063/1.1499754

References

- [TM17] Giacomo Torlai and Roger G. Melko, “Neural Decoder for Topological Codes”, *Physical Review Letters* 119, 030501 (2017). https://doi.org/10.1103/PhysRevLett.119.030501

- [BN18] Nikolas P. Breuckmann and Xiaotong Ni, “Scalable Neural Network Decoders for Higher Dimensional Quantum Codes”, *Quantum* 2, 68 (2018). https://doi.org/10.22331/q-2018-05-24-68

- [SKvN+20] Ryan Sweke, Markus S. Kesselring, Evert P. L. van Nieuwenburg and Jens Eisert, "Reinforcement learning decoders for fault-tolerant quantum computation”, *Machine Learning: Science and Technology* 2, 025005 (2020). https://doi/org/10.1088/2632-2153/abc609

References

- [BSH+23] Johannes Bausch, Andrew W. Senior, Francisco J. H. Heras, et al., “Learning to Decode the Surface Code with a Recurrent, Transformer-Based Neural Network”, *arXiv preprint* (2023). https://doi.org/10.48550/arXiv.2310.05900

- [CVB+21] H. Chen, M. Vasmer, N. P. Breuckmann, and E. Grant, “Automated discovery of logical gates for quantum error correction”, *Quantum Information and Computation* 22(11&12), 947-964 (2021). https://doi.org/10.48550/arXiv.1912.10063

References

- [EK09] Bryan Eastin and Emanuel Knill, "Restrictions on transversal encoded quantum gate sets", *Physical Review Letters* 102, 110502 (2009). https://doi.org/10.1103/PhysRevLett.102.110502

- [BK13] Sergey Bravyi and Robert Koenig, "Classification of Topologically Protected Gates for Local Stabilizer Codes", *Physical Review Letters* 110, 170503 (2013). https://doi.org/10.1103/PhysRevLett.110.170503

Thanks

- mikevasmer.github.io
- Research Scientist @ Perimeter Institute // Senior Quantum Architecture Scientist @ Xanadu
- Thanks to the Perimeter Institute Quantum Intelligence Lab (PIQuIL) for early feedback on this talk!