Tour de gross: A modular quantum computer based on bivariate bicycle codes

Theodore Yoder **Eddie Schoute** Patrick Rall Emily Pritchett
Jay Gambetta Andrew Cross Malcolm Carroll Michael Beverland[1]

Feb 2026
arXiv:2506.03094

[1] In reverse alphebatical order

Introduction

Scaling beyond the surface code

Surface code qubit

1 logical qubit requires $2(d+1)^2$ physical qubits.

Scaling beyond the surface code

Surface code qubit

1 logical qubit requires $2(d+1)^2$ physical qubits.

Bivariate bicycle codes

Gross ($d = 12$): 12 logical qubits per 288 physical qubits.

Scaling beyond the surface code

Surface code qubit

1 logical qubit requires $2(d+1)^2$ physical qubits.

Bivariate bicycle codes

Gross ($d = 12$): 12 logical qubits per 288 physical qubits.
Two-gross ($d = 18$): 12 logical qubits per 576 physical qubits.

Scaling beyond the surface code

Surface code qubit

1 logical qubit requires $2(d+1)^2$ physical qubits.

Bivariate bicycle codes

Gross ($d = 12$): 12 logical qubits per 288 physical qubits.
Two-gross ($d = 18$): 12 logical qubits per 576 physical qubits.

Surface codes require 288 ($d = 11$) and 648 ($d = 17$) physical qubits.

Scaling beyond the surface code

Surface code qubit

1 logical qubit requires $2(d+1)^2$ physical qubits.

Bivariate bicycle codes

Gross ($d = 12$): 12 logical qubits per 288 physical qubits.
Two-gross ($d = 18$): 12 logical qubits per 576 physical qubits.

Surface codes require 288 ($d = 11$) and 648 ($d = 17$) physical qubits.

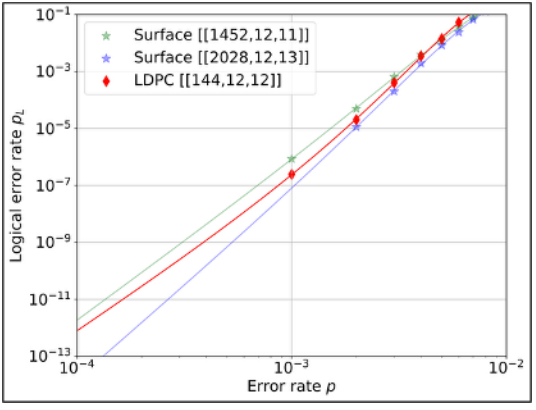


Figure: Compared to surface code, the qLDPC code requires about 10x fewer physical qubits at comparable k and d [Bravyi et al., Nature 627 (2024)].

The gross code and friends

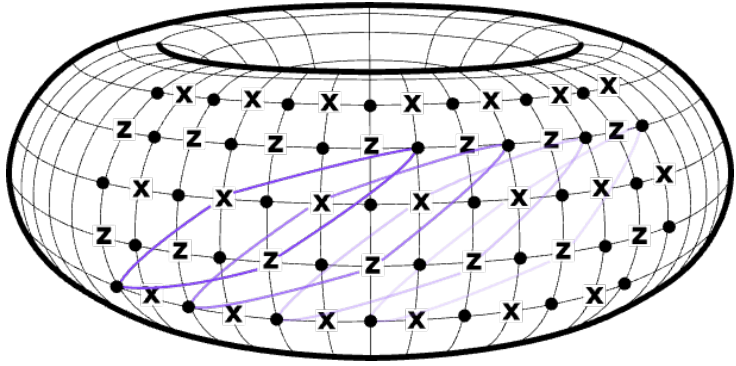


Figure: The gross code is visually represented by a torus due to its Tanner graph embedding plus two long-range connections.

HW demo: 6-way couplers

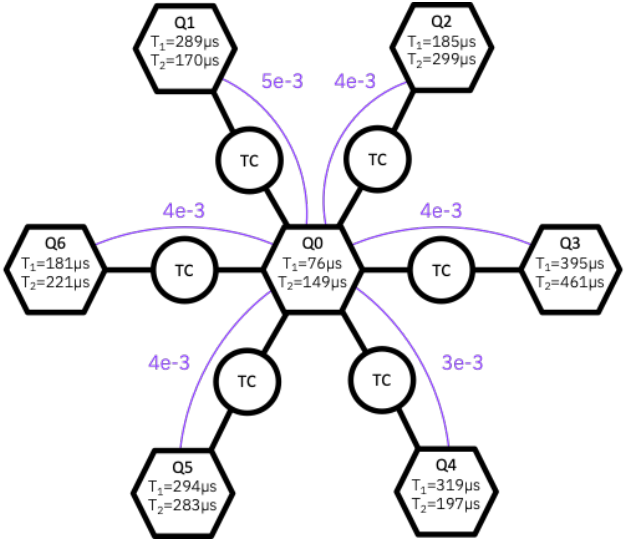


Figure: Degree-6 connectivity without loss of gate fidelity (purple) using tunable couplers (TC)

HW demo: Long-range gates

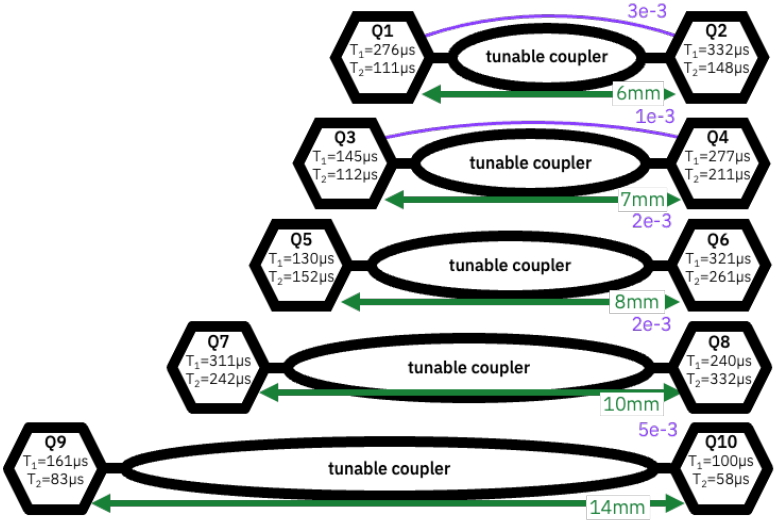


Figure: Demonstration of long-range gates required for the gross code.

HW demo: Non-planar interconnect

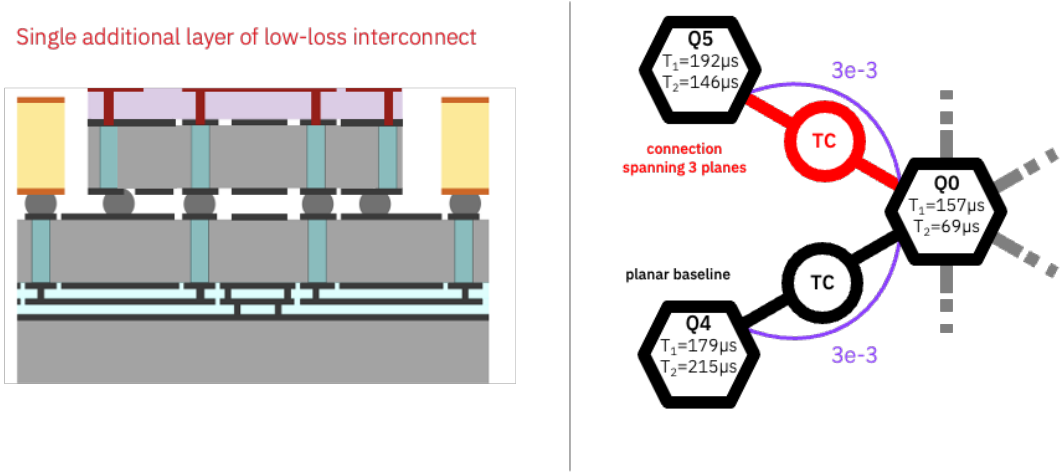
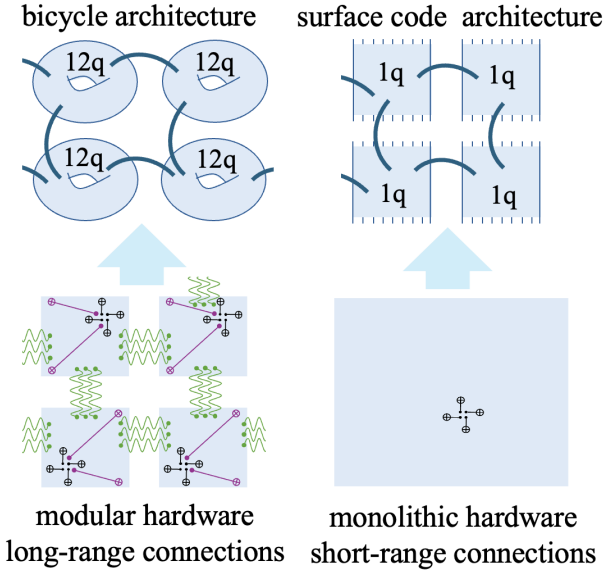


Figure: Demonstration of multi-layer interconnect required for non-planar connectivity in the gross code

Long-range connections enable modularity and scale



Logical capability estimates

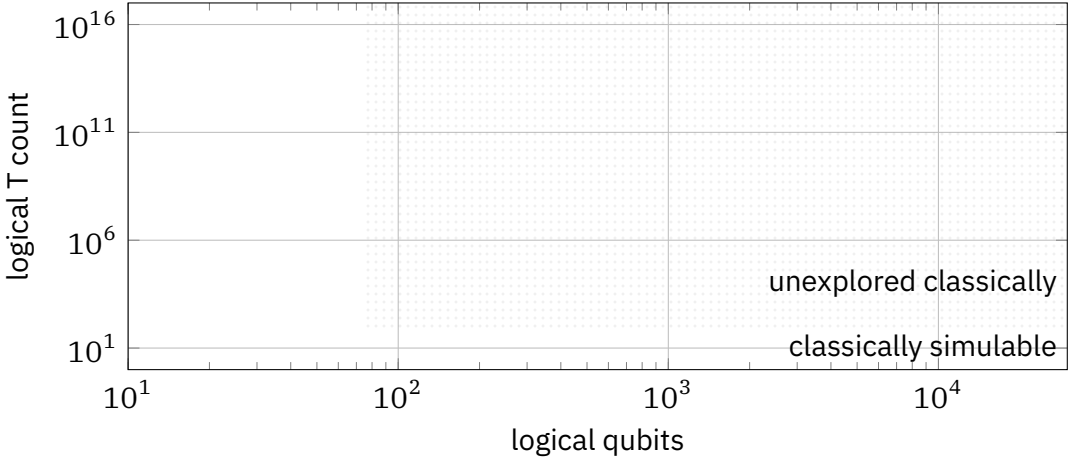


Figure: Bicycle architecture capabilities using gross (red filled ellipse) and two-gross (blue hollow ellipse) codes when compared to surface code architectures (black square) given q physical qubits and p physical error rate.

Logical capability estimates

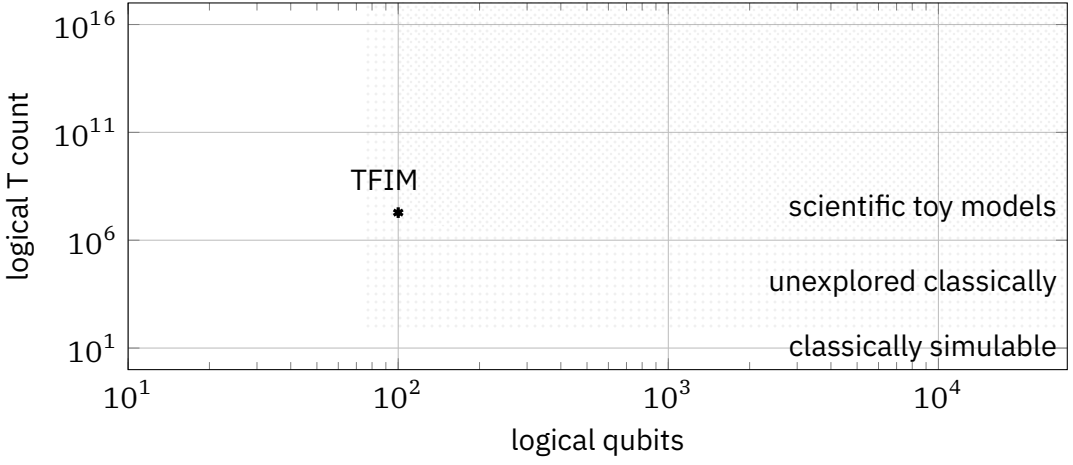


Figure: Bicycle architecture capabilities using gross (red filled ellipse) and two-gross (blue hollow ellipse) codes when compared to surface code architectures (black square) given q physical qubits and p physical error rate.

Logical capability estimates

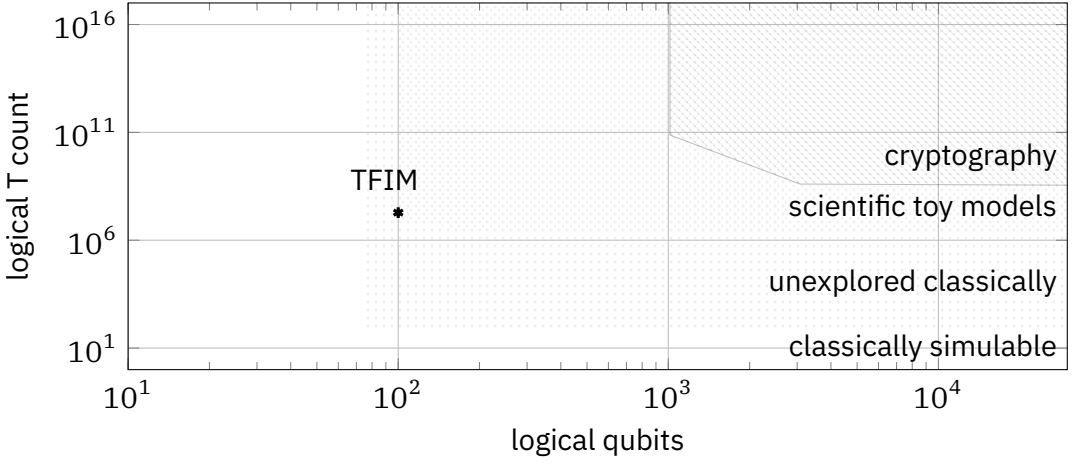


Figure: Bicycle architecture capabilities using gross (red filled ellipse) and two-gross (blue hollow ellipse) codes when compared to surface code architectures (black square) given q physical qubits and p physical error rate.

Logical capability estimates

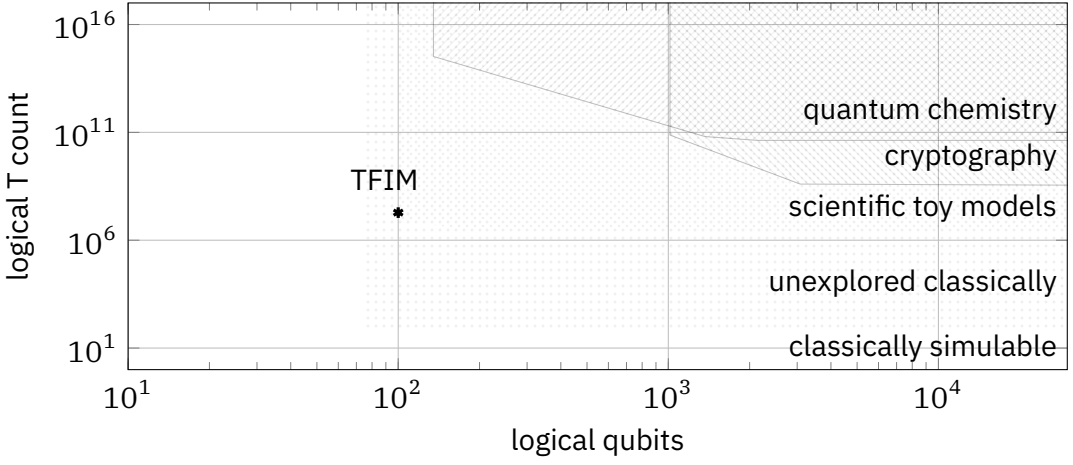


Figure: Bicycle architecture capabilities using gross (red filled ellipse) and two-gross (blue hollow ellipse) codes when compared to surface code architectures (black square) given q physical qubits and p physical error rate.

Logical capability estimates

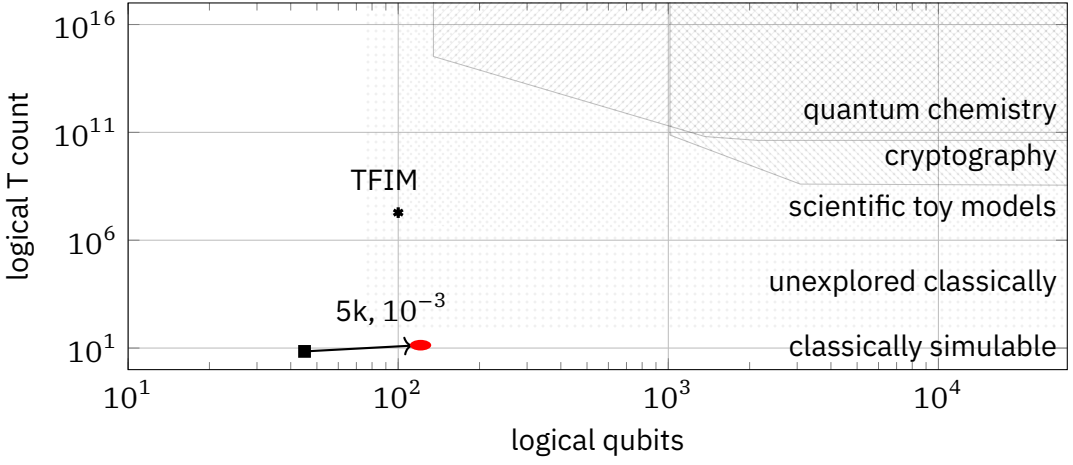


Figure: Bicycle architecture capabilities using gross (red filled ellipse) and two-gross (blue hollow ellipse) codes when compared to surface code architectures (black square) given q physical qubits and p physical error rate.

Logical capability estimates

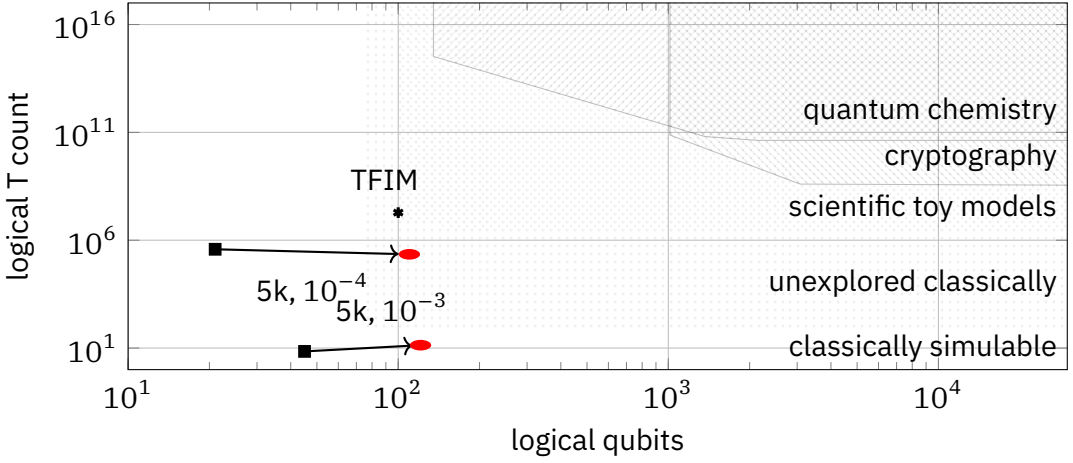


Figure: Bicycle architecture capabilities using gross (red filled ellipse) and two-gross (blue hollow ellipse) codes when compared to surface code architectures (black square) given q physical qubits and p physical error rate.

Logical capability estimates

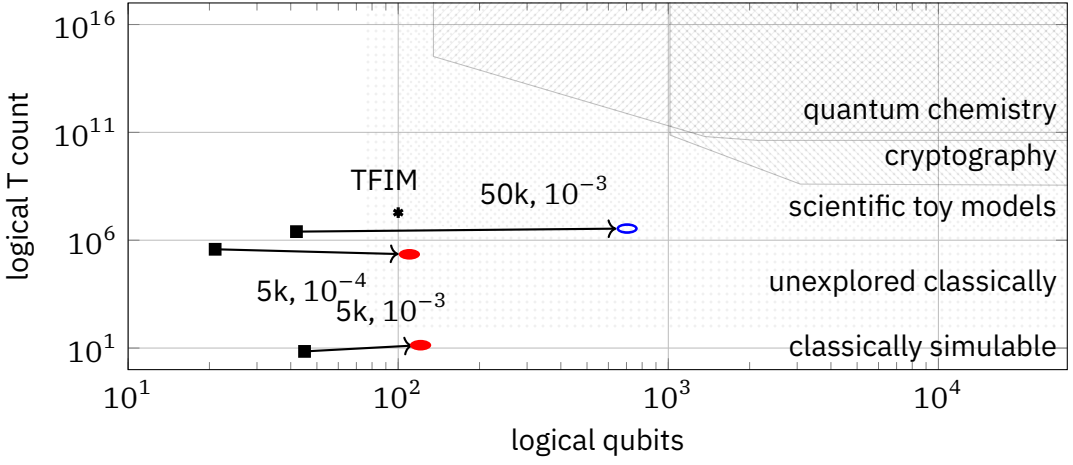


Figure: Bicycle architecture capabilities using gross (red filled ellipse) and two-gross (blue hollow ellipse) codes when compared to surface code architectures (black square) given q physical qubits and p physical error rate.

Logical capability estimates

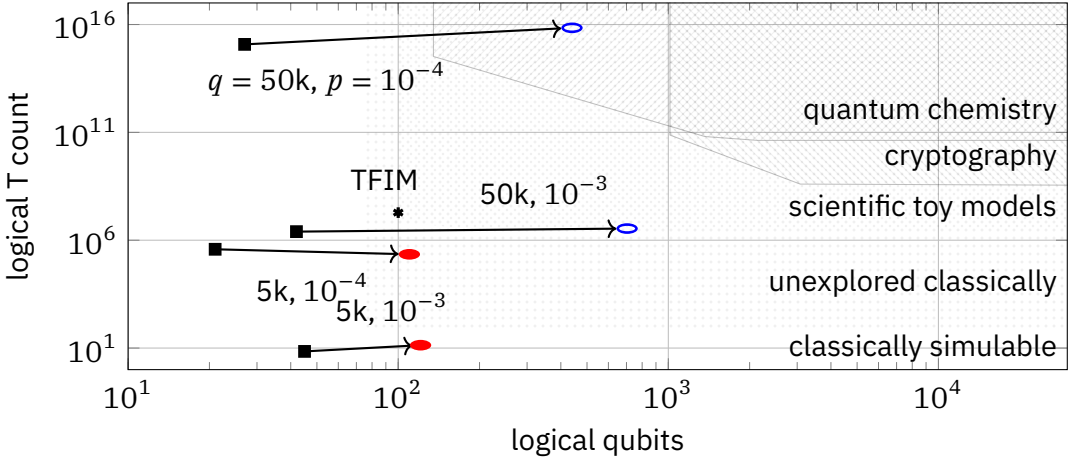


Figure: Bicycle architecture capabilities using gross (red filled ellipse) and two-gross (blue hollow ellipse) codes when compared to surface code architectures (black square) given q physical qubits and p physical error rate.

Logical capability estimates

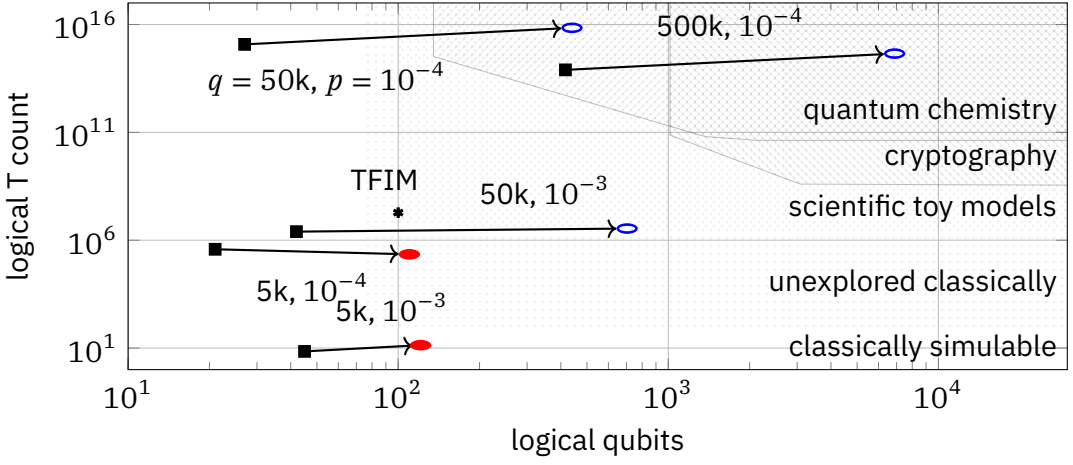


Figure: Bicycle architecture capabilities using gross (red filled ellipse) and two-gross (blue hollow ellipse) codes when compared to surface code architectures (black square) given q physical qubits and p physical error rate.

Bicycle architecture

Two code modules

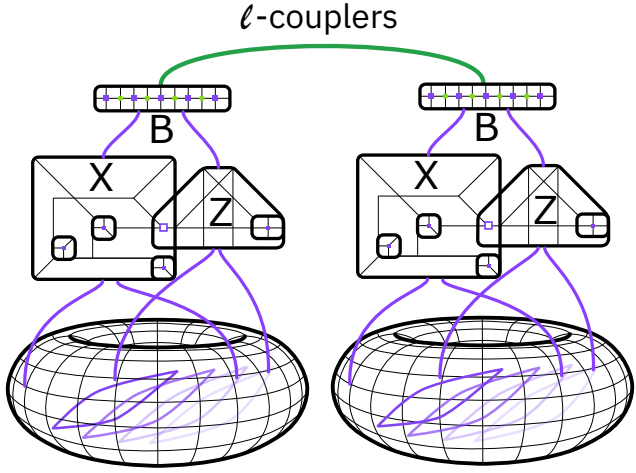


Figure: A gross code is just a memory. We can perform computation by attaching an ancilla system called the *logical processing unit* (LPU) [Cross, He, Rall, Yoder (2024); Williamson, Yoder (2024)]. By connecting LPUs through a bridge system, we can perform joint measurements between two codes.

Bicycle architecture

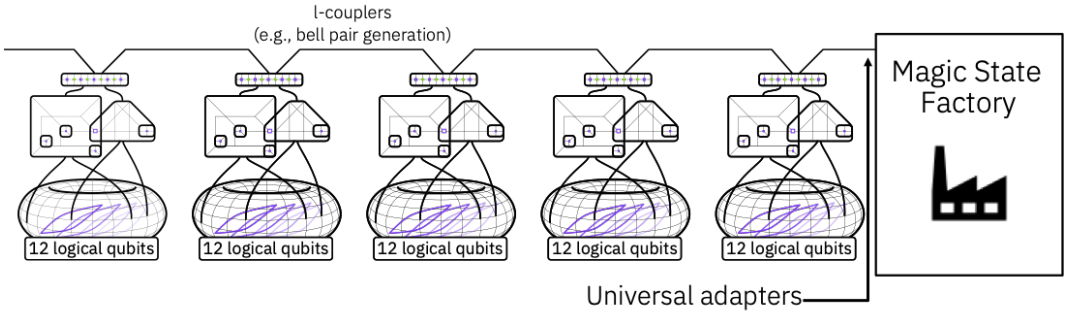
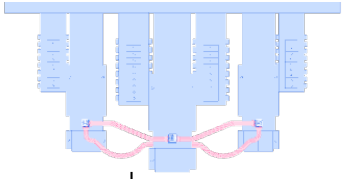


Figure: The bicycle architecture consists of many connected code modules and a (magic state) factory module.

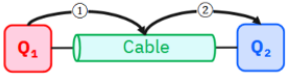
HW demo: L-coupler



"l-couplers"

Demonstrating key L-coupler technology required for Flamingo

Transfer of quantum state by relay operation



Vacuum rabi oscillations from qubit to cable modes

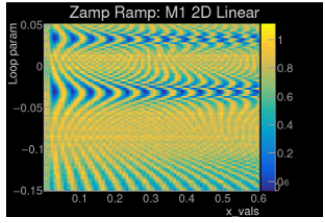


Figure: Demonstration of entangling gate operation over L-coupled interface of up to 96.5% fidelity through interconnect of about 1 meter [Shawn Hall, Kentaro Heya, Yadav Prasad Kandel, Moein Malek, Yves Martin, Jae-Woong Nah, Jason Orcutt, Timothy Phung, Rachel Steiner, Neereja Sundaresan].

Physical qubit costs

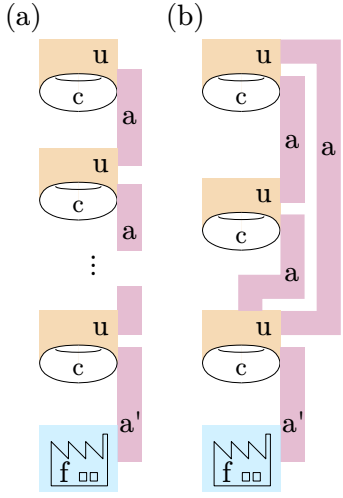


Figure: Code modules (c) with logical processing units (u) and a factory module (f) connected in two ways via adapters (a, a').

Physical qubit costs

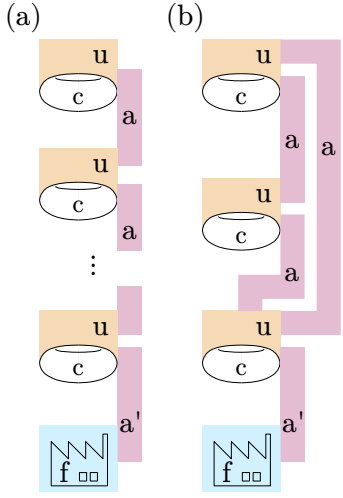


Figure: Code modules (c) with logical processing units (u) and a factory module (f) connected in two ways via adapters (a, a').

system	p	gross	two-gross
c		288	576
u		90	158
a		22	34
f	10^{-3}	454	810
	10^{-4}	463	18,600
a'	10^{-3}	29	13
	10^{-4}	29	49

Table: Physical qubit counts for each system in the architecture

Bicycle instructions

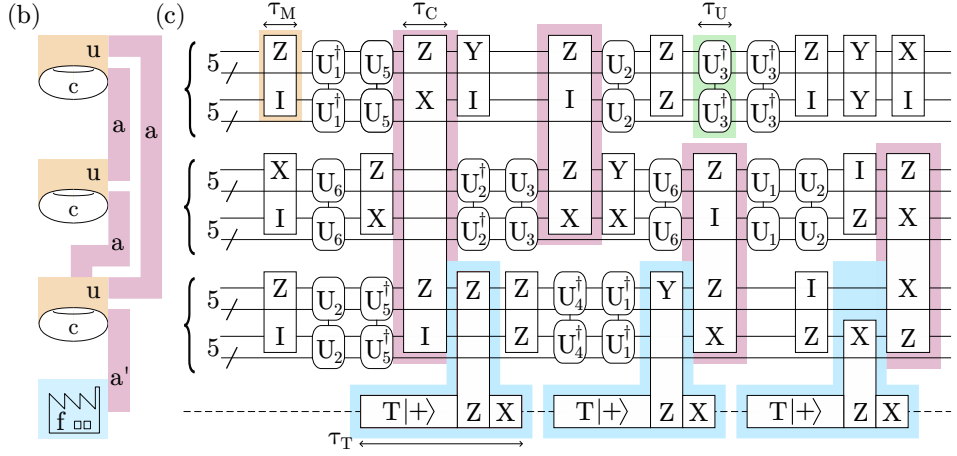


Figure: Given connectivity (b) we define the available *bicycle instructions* (c) on the bicycle architecture: In-module measurement (yellow) and inter-module measurement (red); Automorphisms $U_i \otimes U_i$ (green) and T state injection (blue).

Bicycle instruction error rates

Real-time decoding: Relay-BP

We use Relay-BP, which can achieve a orders of magnitude improvement over BP+OSD+CS-10 and is suitable for real-time implementation [Müller et al., arXiv:2506.01779]

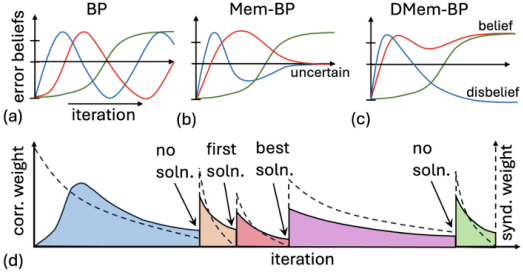


Figure: Relay-BP chains multiple runs of belief propagation with disordered memory strengths.

Real-time decoding: Relay-BP

We use Relay-BP, which can achieve a orders of magnitude improvement over BP+OSD+CS-10 and is suitable for real-time implementation [Müller et al., arXiv:2506.01779]

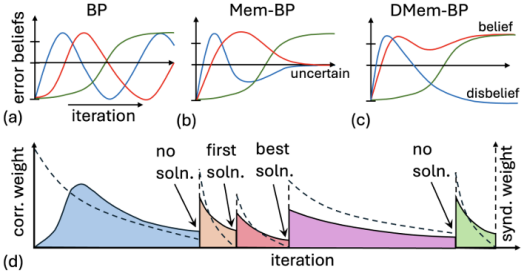


Figure: Relay-BP chains multiple runs of belief propagation with disordered memory strengths.

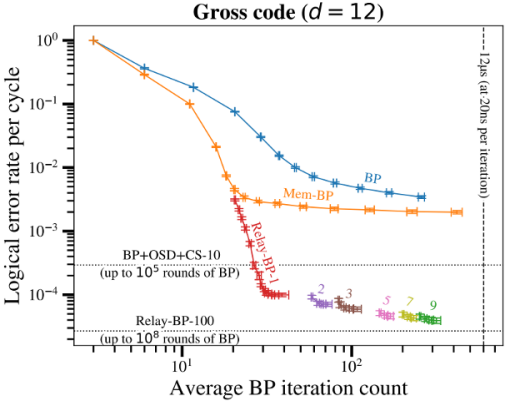
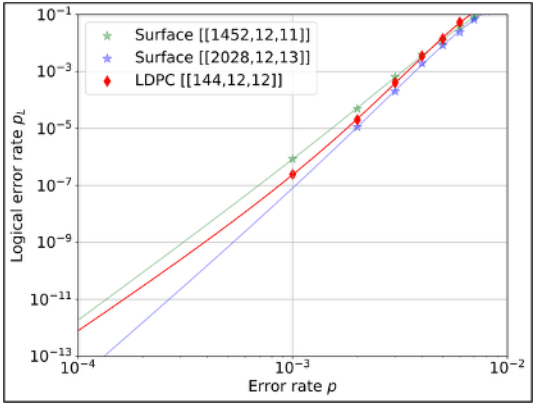


Figure: Substantially lower error rates within 600 iterations

IBM **Quantum**

Monte Carlo limitations



- Monte Carlo can sample at high error rates
- How do we figure out what's going on at low p?

Extrapolation from low-weight failure fraction (gross)

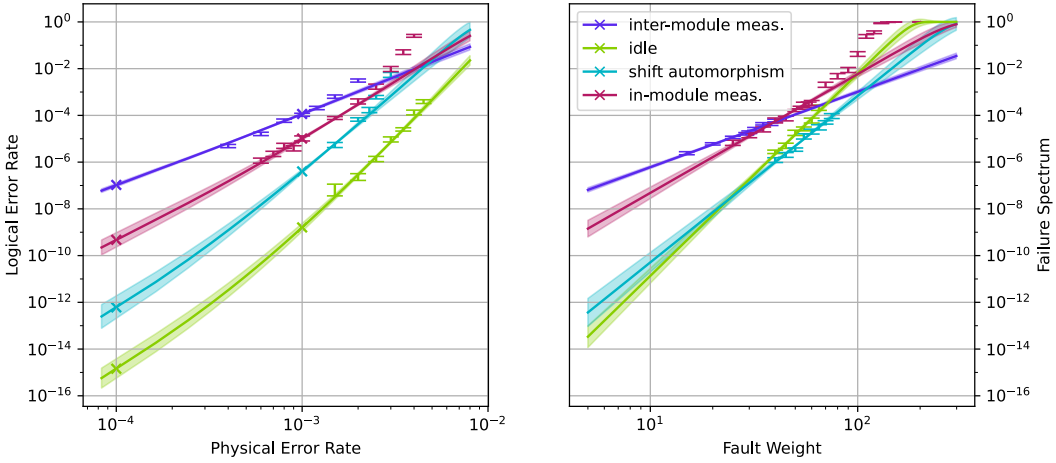


Figure: We sample sets of exactly w circuit faults and estimate failure rate (right). We then fit ansatz function and translate back to logical error rate (left) and observe agreement with independent Monte Carlo data. [Beverland et al. (2025)]

Extrapolation from low-weight failure fraction (two-gross)

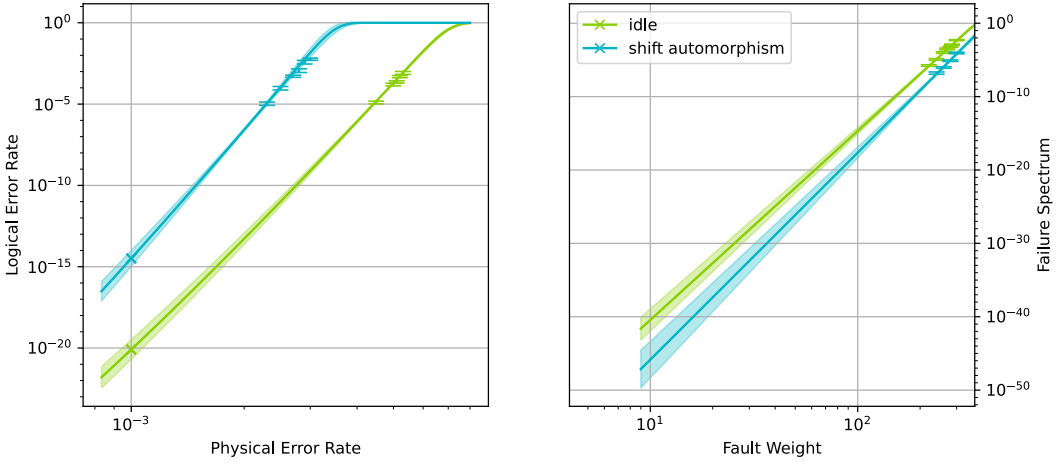


Figure: We sample sets of exactly w circuit faults and estimate failure rate (right). We then fit ansatz function and translate back to logical error rate (left) and observe agreement with independent Monte Carlo data. [Beverland et al. (2025)]

Bicycle instruction properties

			P_i (logical error rate)	
instruction	module	τ_i (timesteps)	$p = 10^{-3}$	$p = 10^{-4}$
idle	gross	8	$10^{-8.8\pm0.2}$	$10^{-14.8\pm0.4}$
	two-gross	8	$10^{-20.1\pm0.5}$	$10^{-38.3\pm0.9}$
shift automorphism	gross	14	$10^{-6.4\pm0.2}$	$10^{-12.2\pm0.5}$
	two-gross	14	$10^{-14.5\pm0.4}$	$10^{-37\pm1}$
in-module meas.	gross	120	$10^{-5.0\pm0.1}$	$10^{-9.0\pm0.2}$
	two-gross	216	$[10^{-11}]$	$[10^{-20}]$
inter-module meas.	gross	120	$10^{-2.7\pm0.1}$	$10^{-7.3\pm0.3}$
	two-gross	216	$[10^{-9}]$	$[10^{-18}]$

Table: Time duration and logical error rate properties of bicycle instructions. We omit here T state injection, using magic state cultivation [Gidney, Shutty, Jones (2024)] or distillation [Litinski (2019)]. Bracketed numbers are assumptions without data.

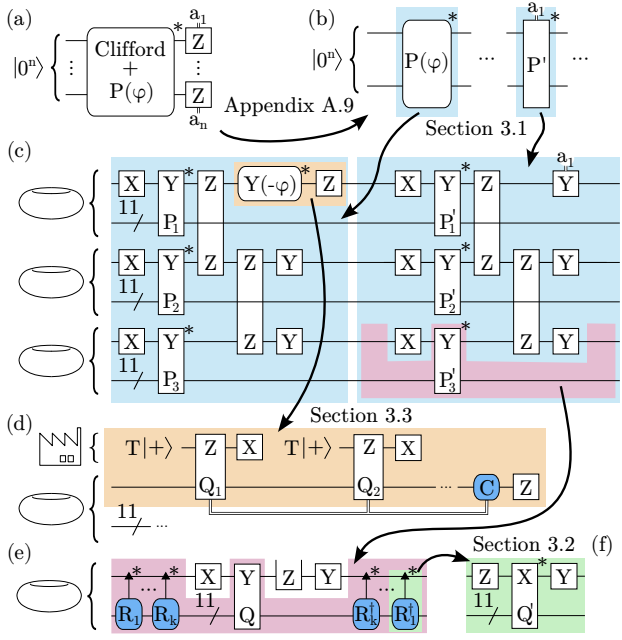
Bicycle instruction properties

instruction	module	τ_i (timesteps)	P_i (logical error rate) $p = 10^{-3}$	$p = 10^{-4}$
idle	gross	8	$10^{-8.8\pm0.2}$	$10^{-14.8\pm0.4}$
	two-gross	8	$10^{-20.1\pm0.5}$	$10^{-38.3\pm0.9}$
shift automorphism	gross	14	$10^{-6.4\pm0.2}$	$10^{-12.2\pm0.5}$
	two-gross	14	$10^{-14.5\pm0.4}$	$10^{-37\pm1}$
in-module meas.	gross	120	$10^{-5.0\pm0.1}$	$10^{-9.0\pm0.2}$
	two-gross	216	$[10^{-11}]$	$[10^{-20}]$
inter-module meas.	gross	120	$10^{-4.0\pm0.2}$	$10^{-7.3\pm0.3}$
	two-gross	216	$[10^{-9}]$	$[10^{-18}]$

Table: Time duration and logical error rate properties of bicycle instructions. We omit here T state injection, using magic state cultivation [Gidney, Shutty, Jones (2024)] or distillation [Litinski (2019)]. Bracketed numbers are assumptions without data.

Compiling for the bicycle architecture

Compiler summary



IBM **Quantum**

Compiler implementation

Compiler implementation available at
`github.com/qiskit-community/bicycle-architecture-compiler`

Graphical language

Round boxes are unitary gates

$V :=$

, $e^{i\frac{\pi}{4}P} :=$

, (1)

Graphical language

Round boxes are unitary gates

$$V \coloneqq \text{}, \qquad e^{i\frac{\pi}{4}P} \coloneqq \text{  }, \tag{1}$$

and square boxes are measurements

$$\frac{\mathbb{1}+(-1)^a P}{2} \coloneqq \text{}, \tag{2}$$

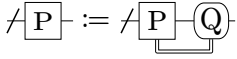
for measurement outcome $a \in \{0, 1\}$.

Measurement projection

Frequently, we don’t care about the measurement outcome because it’s random.

Measurement projection

Frequently, we don't care about the measurement outcome because it's random. We draw a measurement without an outcome wire and define it as a *measurement projection*,

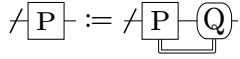


(3)

Effectively, it is the projection $\frac{\mathbb{1}+P}{2}$.

Measurement projection

Frequently, we don't care about the measurement outcome because it's random. We draw a measurement without an outcome wire and define it as a *measurement projection*,



(3)

Effectively, it is the projection $\frac{\mathbb{1}+P}{2}$.
This is only possible if we know an anti-commuting stabilizer Q before measuring P so that

$$Q^a \frac{\mathbb{1}+(-1)^a P}{2}|\psi\rangle = \frac{\mathbb{1}+P}{2} Q^a |\psi\rangle = \frac{\mathbb{1}+P}{2}|\psi\rangle. \tag{4}$$

Measurement ancilla

Let us focus on Pauli-generated rotations,

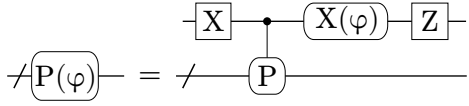
$$P(\varphi) \coloneqq e^{i\varphi P}. \tag{5}$$

Measurement ancilla

Let us focus on Pauli-generated rotations,

$$P(\varphi) \coloneqq e^{i\varphi P}. \tag{5}$$

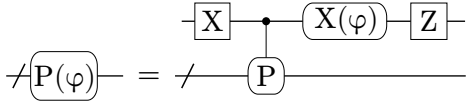
A Pauli-generated rotation can be implemented by



(6)

Distributing across data modules

We can add some ancillas to

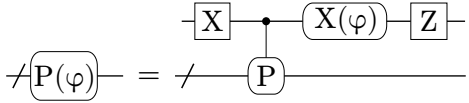


(7)

and assume $P = P_1 \otimes P_2 \otimes P_3$

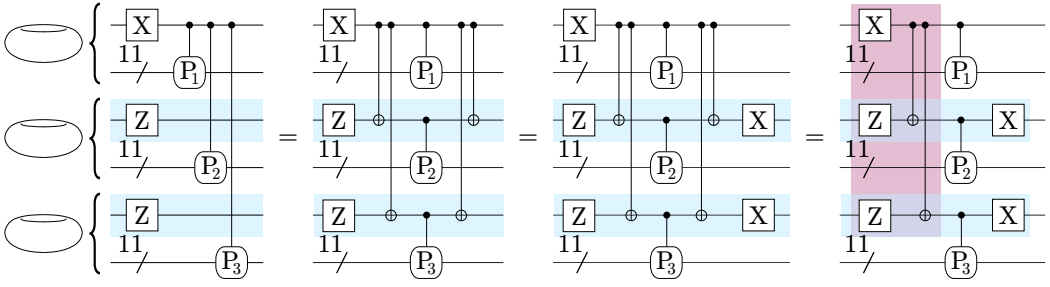
Distributing across data modules

We can add some ancillas to



(7)

and assume $P = P_1 \otimes P_2 \otimes P_3$

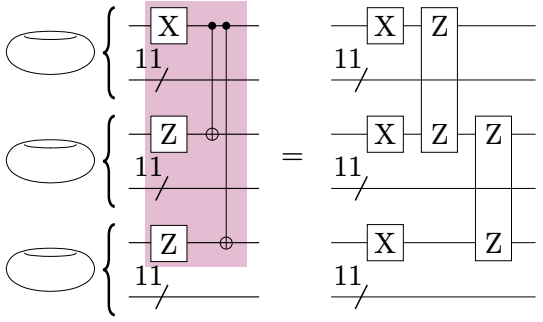


(8)

IBM **Quantum**

It’s a GHZ state

We identify the state preparation as a GHZ state

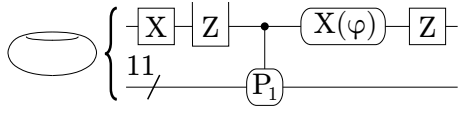


, (9)

which uses just bicycle instructions (assuming connected modules).

The result

Thus, the first code module has the following circuit

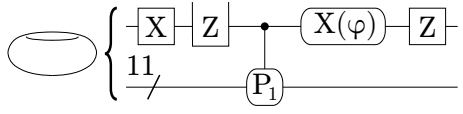


, (10)

with one end of a ZZ-measurement depicted.

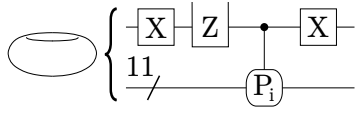
The result

Thus, the first code module has the following circuit



, (10)

with one end of a ZZ-measurement depicted.
Code modules $i = 2, \ldots, M$ have

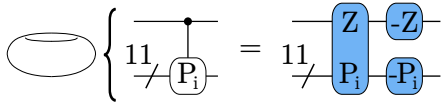


, (11)

where we depict only one ZZ measurement.
(If we were compiling a Pauli measurement, then there are no instances of (10).)

Decomposing controlled-P_i

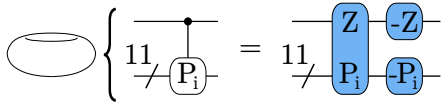
We use the identity, for any Pauli P_i,



(12)

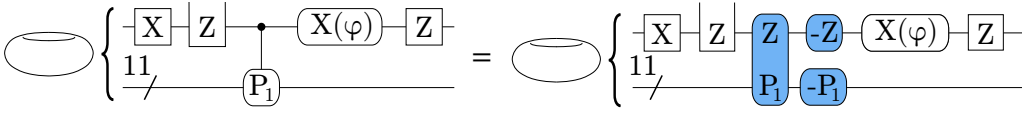
Decomposing controlled-P_i

We use the identity, for any Pauli P_i,

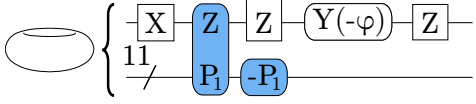


(12)

to simplify



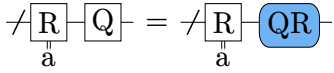
(13)



(14)

Measurement to rotation identity

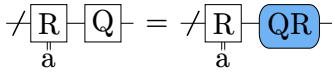
For anti-commuting Paulis Q and R, it holds that



. (15)

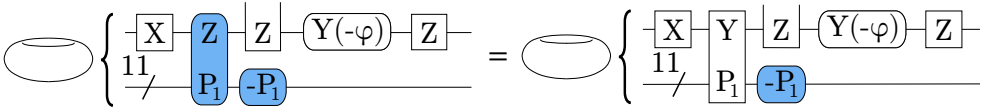
Measurement to rotation identity

For anti-commuting Paulis Q and R, it holds that



. (15)

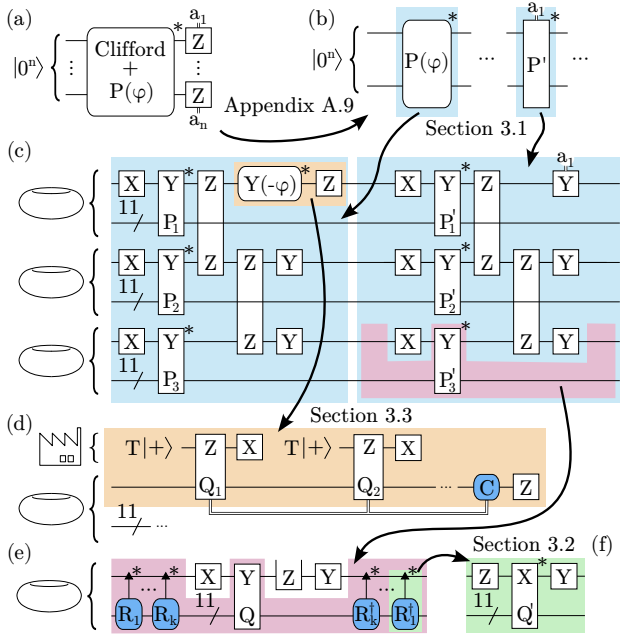
Now, we can rewrite



(16)

The Clifford rotation generated by P_1 can be compiled away using standard techniques.

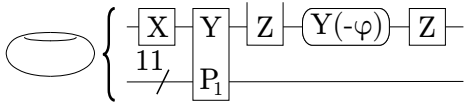
Recap



IBM **Quantum**

Measuring arbitrary Paulis in a module

We are faced with implementing the circuit

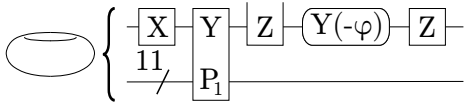


(17)

The measurement $Y \otimes P_1$ is not a bicycle instruction.

Measuring arbitrary Paulis in a module

We are faced with implementing the circuit

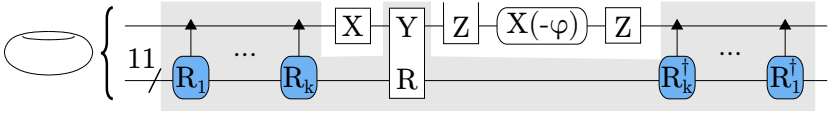


(17)

The measurement $Y \otimes P_1$ is not a bicycle instruction.
But we can conjugate P_1 with some 11-qubit Clifford gate so that it becomes easy to implement [Cross, He, Rall, Yoder (2024)].

Implementing a Clifford rotation

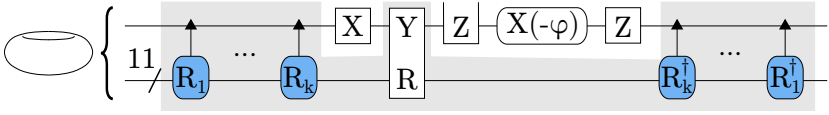
We need at most three native Clifford rotations for any P_1 in



(18)

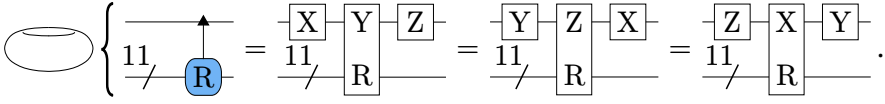
Implementing a Clifford rotation

We need at most three native Clifford rotations for any P_1 in



(18)

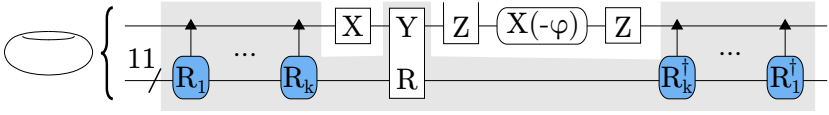
and each native Clifford rotation is easily implemented using an ancilla by



(19)

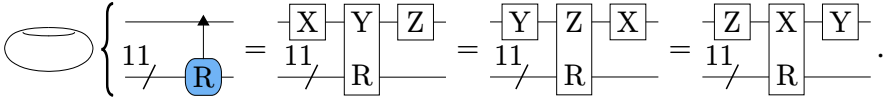
Implementing a Clifford rotation

We need at most three native Clifford rotations for any P_1 in



(18)

and each native Clifford rotation is easily implemented using an ancilla by



(19)

This gives an overhead for *Pauli addressing* some $Y \otimes P_1$ measurement

Overhead of arbitrary Pauli addressing

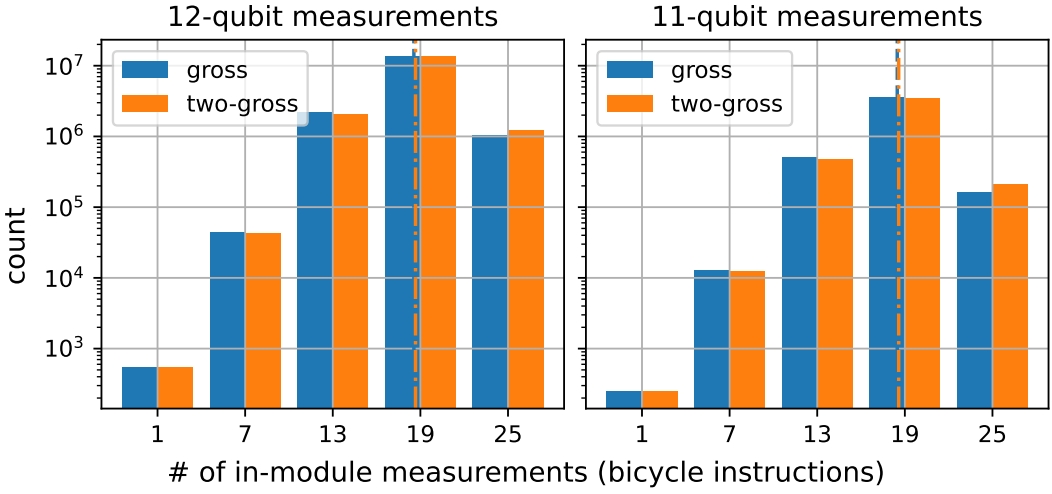


Figure: Number of bicycle measurements needed to measure the 12-qubit Paulis, $Q \otimes P_t$, (left) with mean values 18.55 (gross) and 18.67 (two-gross). In practice, we can minimize over Q (right) to get mean values 18.48 (gross) and 18.58 (two-gross).

Automorphism group structure

The automorphism unitaries are representations of a group. The parity matrices

$$A_x^{\text{gross}} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}, \qquad A_y^{\text{gross}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \tag{20}$$

are each of order 6.

Automorphism group structure

The automorphism unitaries are representations of a group. The parity matrices

$$A_x^{\text{gross}} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}, \qquad A_y^{\text{gross}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \tag{20}$$

are each of order 6.
This gives us $15 \times 6^2 = 540$ *easy* measurements and 510 easy Clifford rotations.

Automorphism group structure

The automorphism unitaries are representations of a group. The parity matrices

$$A_x^{\text{gross}} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}, \qquad A_y^{\text{gross}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \tag{20}$$

are each of order 6.
This gives us $15 \times 6^2 = 540$ *easy* measurements and 510 easy Clifford rotations.

Combinatorial open question: Reducing overhead

Given measurements $\{P_1, \ldots, P_k\}$ can we map each P_j to easier measurements through $l \ll k$ easy Clifford rotations?

End-to-end resource estimates

Logical capability estimates

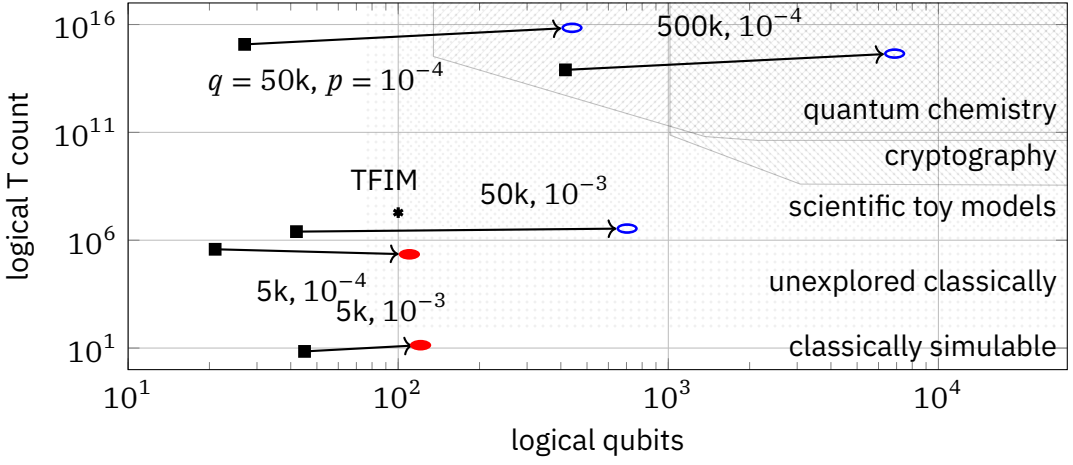


Figure: Bicycle architecture capabilities using gross (red filled ellipse) and two-gross (blue hollow ellipse) codes when compared to surface code architectures (black square) given q physical qubits and p physical error rate.

Methodology

We consider circuits of the form

$$\prod_j e^{i\frac{\varphi_j}{2}Q_j}.$$

After compiling, we associate identical logical errors and timing information to all bicycle instructions and compute a simple additive failure probability,

$$\sum_j N_j P_j,$$

including any required idling.

Capability estimates: Random circuits

p	q	code	n	N_T	timesteps / T gate
10^{-3}	5k	gross	121	1.4×10^{1}	3367
		surface ($d = 5$)	45	7.0	381
	50k	two-gross	704	3.5×10^{6}	7057
		surface ($d = 17$)	42	2.5×10^{6}	2269
10^{-4}	5k	gross	110	2.2×10^{5}	3091
		surface ($d = 7$)	21	3.8×10^{5}	115
	50k	gross	1342	3.7×10^{4}	3135
		surface ($d = 7$)	250	3.2×10^{4}	115
		two-gross	440	7.0×10^{15}	5436
		surface ($d = 17$)	27	1.2×10^{15}	509
	500k	two-gross	6886	4.5×10^{14}	5437
		surface ($d = 17$)	416	7.9×10^{13}	509

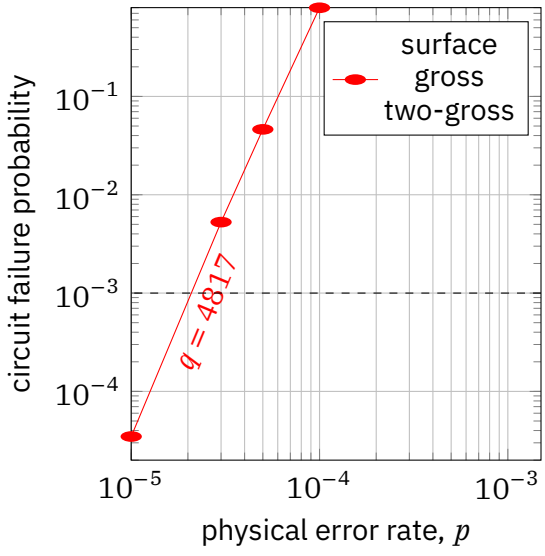
Table: The T count, N_T, estimates the number of $e^{i\frac{\pi}{8}Q}$ gates with full support on n qubits such that circuit failure probability $\leq \frac{1}{3}$.

Resource requirements for transverse-field Ising model

- We choose the same parameters for the 10×10 2D TFIM Hamiltonian as [Beverland et al. (2022)] and require $\epsilon \leq 10^{-3}$.
- The resulting circuit has 1.8×10^5 Pauli-generated rotations (of small-angle).

Resource requirements for transverse-field Ising model

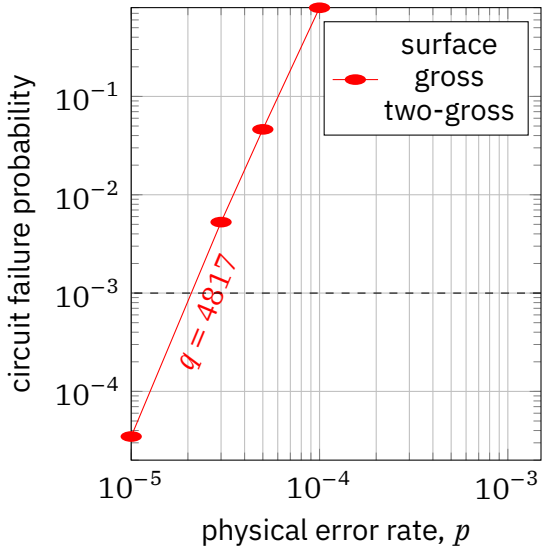
- We choose the same parameters for the 10×10 2D TFIM Hamiltonian as [Beverland et al. (2022)] and require $\epsilon \leq 10^{-3}$.
- The resulting circuit has 1.8×10^5 Pauli-generated rotations (of small-angle).



IBM **Quantum**

Resource requirements for transverse-field Ising model

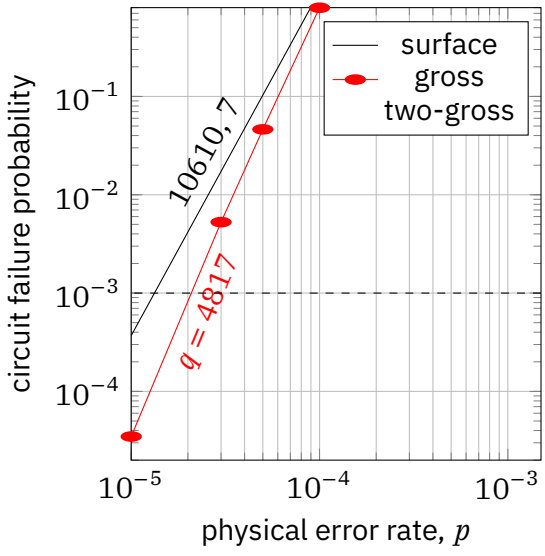
- We choose the same parameters for the 10×10 2D TFIM Hamiltonian as [Beverland et al. (2022)] and require $\epsilon \leq 10^{-3}$.
- The resulting circuit has 1.8×10^5 Pauli-generated rotations (of small-angle).
- Gross requires $p \leq 2.1 \times 10^{-5}$



IBM **Quantum**

Resource requirements for transverse-field Ising model

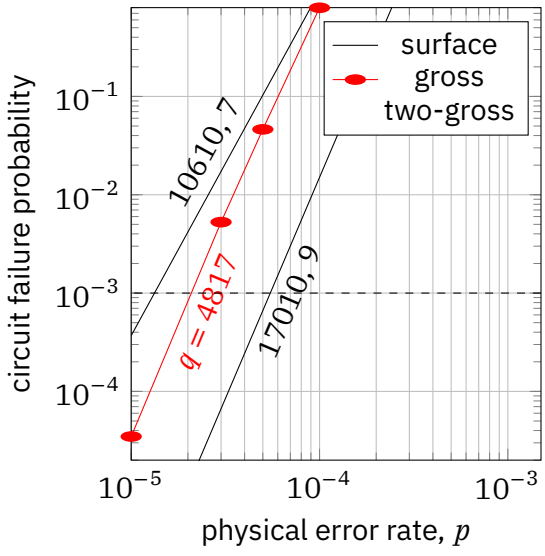
- We choose the same parameters for the 10×10 2D TFIM Hamiltonian as [Beverland et al. (2022)] and require $\epsilon \leq 10^{-3}$.
- The resulting circuit has 1.8×10^5 Pauli-generated rotations (of small-angle).
- Gross requires $p \leq 2.1 \times 10^{-5}$



IBM **Quantum**

Resource requirements for transverse-field Ising model

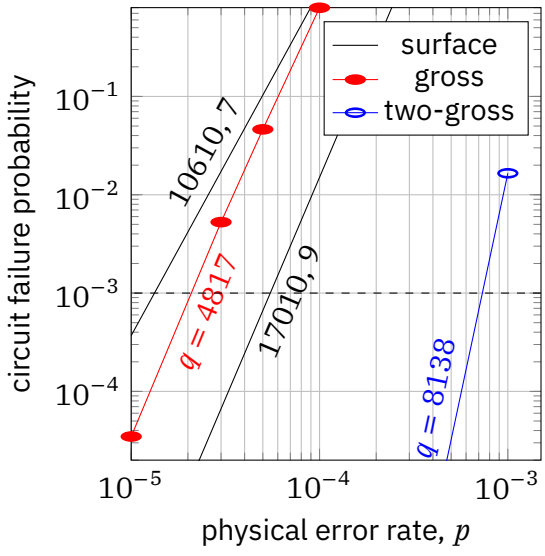
- We choose the same parameters for the 10×10 2D TFIM Hamiltonian as [Beverland et al. (2022)] and require $\epsilon \leq 10^{-3}$.
- The resulting circuit has 1.8×10^5 Pauli-generated rotations (of small-angle).
- Gross requires $p \leq 2.1 \times 10^{-5}$



IBM **Quantum**

Resource requirements for transverse-field Ising model

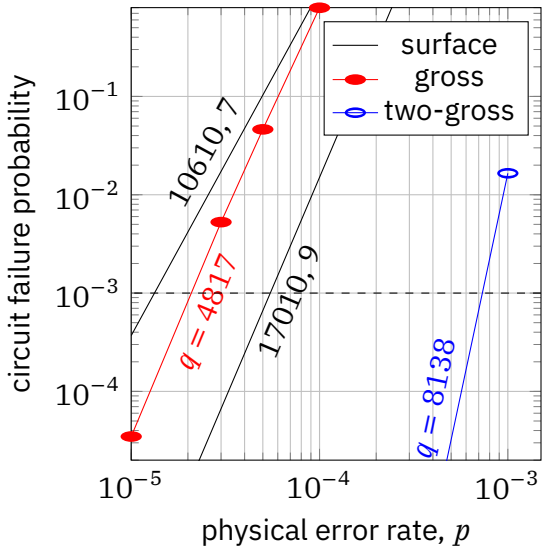
- We choose the same parameters for the 10×10 2D TFIM Hamiltonian as [Beverland et al. (2022)] and require $\epsilon \leq 10^{-3}$.
- The resulting circuit has 1.8×10^5 Pauli-generated rotations (of small-angle).
- Gross requires $p \leq 2.1 \times 10^{-5}$



IBM **Quantum**

Resource requirements for transverse-field Ising model

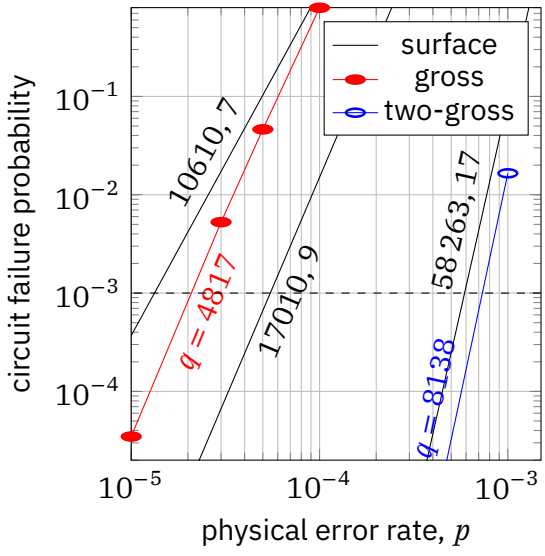
- We choose the same parameters for the 10×10 2D TFIM Hamiltonian as [Beverland et al. (2022)] and require $\epsilon \leq 10^{-3}$.
- The resulting circuit has 1.8×10^5 Pauli-generated rotations (of small-angle).
- Gross requires $p \leq 2.1 \times 10^{-5}$
- Two-gross requires $p \leq 7.3 \times 10^{-4}$



IBM **Quantum**

Resource requirements for transverse-field Ising model

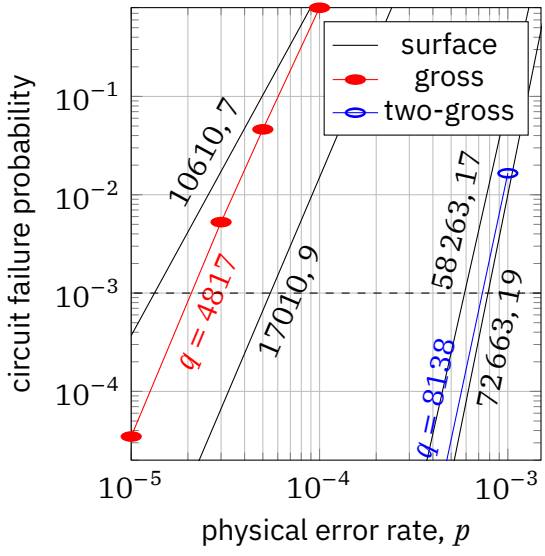
- We choose the same parameters for the 10×10 2D TFIM Hamiltonian as [Beverland et al. (2022)] and require $\epsilon \leq 10^{-3}$.
- The resulting circuit has 1.8×10^5 Pauli-generated rotations (of small-angle).
- Gross requires $p \leq 2.1 \times 10^{-5}$
- Two-gross requires $p \leq 7.3 \times 10^{-4}$



IBM **Quantum**

Resource requirements for transverse-field Ising model

- We choose the same parameters for the 10×10 2D TFIM Hamiltonian as [Beverland et al. (2022)] and require $\epsilon \leq 10^{-3}$.
- The resulting circuit has 1.8×10^5 Pauli-generated rotations (of small-angle).
- Gross requires $p \leq 2.1 \times 10^{-5}$
- Two-gross requires $p \leq 7.3 \times 10^{-4}$



IBM **Quantum**

Conclusion

Scalable fault-tolerance criteria

The bicycle architecture satisfies our six sufficient critera for a scalable fault-tolerant architecture:

(i) *Fault-tolerant* – logical errors are suppressed enough for meaningful algorithms to succeed.

Scalable fault-tolerance criteria

The bicycle architecture satisfies our six sufficient critera for a scalable fault-tolerant architecture:

(i) *Fault-tolerant* – logical errors are suppressed enough for meaningful algorithms to succeed.

(ii) *Addressable* – individual logical qubits can be prepared or measured throughout the computation.

Scalable fault-tolerance criteria

The bicycle architecture satisfies our six sufficient critera for a scalable fault-tolerant architecture:

(i) *Fault-tolerant* – logical errors are suppressed enough for meaningful algorithms to succeed.

(ii) *Addressable* – individual logical qubits can be prepared or measured throughout the computation.

(iii) *Universal* – a universal set of quantum instructions can be applied to the logical qubits.

Scalable fault-tolerance criteria

The bicycle architecture satisfies our six sufficient critera for a scalable fault-tolerant architecture:

(i) *Fault-tolerant* – logical errors are suppressed enough for meaningful algorithms to succeed.

(ii) *Addressable* – individual logical qubits can be prepared or measured throughout the computation.

(iii) *Universal* – a universal set of quantum instructions can be applied to the logical qubits.

(iv) *Adaptive* – measurements are real-time decoded and can alter subsequent quantum instructions.

Scalable fault-tolerance criteria

The bicycle architecture satisfies our six sufficient critera for a scalable fault-tolerant architecture:

(i) *Fault-tolerant* – logical errors are suppressed enough for meaningful algorithms to succeed.

(ii) *Addressable* – individual logical qubits can be prepared or measured throughout the computation.

(iii) *Universal* – a universal set of quantum instructions can be applied to the logical qubits.

(iv) *Adaptive* – measurements are real-time decoded and can alter subsequent quantum instructions.

(v) *Modular* – the hardware is distributed across a set of replaceable modules connected quantumly.

Scalable fault-tolerance criteria

The bicycle architecture satisfies our six sufficient critera for a scalable fault-tolerant architecture:

(i) *Fault-tolerant* – logical errors are suppressed enough for meaningful algorithms to succeed.

(ii) *Addressable* – individual logical qubits can be prepared or measured throughout the computation.

(iii) *Universal* – a universal set of quantum instructions can be applied to the logical qubits.

(iv) *Adaptive* – measurements are real-time decoded and can alter subsequent quantum instructions.

(v) *Modular* – the hardware is distributed across a set of replaceable modules connected quantumly.

(vi) *Efficient* – meaningful algorithms can be executed with reasonable physical resources.

Opportunities for improvement

Reduce the time overhead The synthesis of arbitrary Pauli measurements incurs significant overhead.

Opportunities for improvement

Reduce the time overhead The synthesis of arbitrary Pauli measurements incurs significant overhead.

Increase parallelism The Pauli-based compilation scheme leads to sequential T gate execution.

Opportunities for improvement

Reduce the time overhead The synthesis of arbitrary Pauli measurements incurs significant overhead.

Increase parallelism The Pauli-based compilation scheme leads to sequential T gate execution.

Decoding with speed and accuracy For instance, decoding speed and error rates of inter-module measurement [Maurya et al. (2025)].

Opportunities for improvement

Reduce the time overhead The synthesis of arbitrary Pauli measurements incurs significant overhead.

Increase parallelism The Pauli-based compilation scheme leads to sequential T gate execution.

Decoding with speed and accuracy For instance, decoding speed and error rates of inter-module measurement [Maurya et al. (2025)].

Increasing code and circuit distances With long-range coupling, can consider a wide variety of qLDPC codes.

Opportunities for improvement

Reduce the time overhead The synthesis of arbitrary Pauli measurements incurs significant overhead.

Increase parallelism The Pauli-based compilation scheme leads to sequential T gate execution.

Decoding with speed and accuracy For instance, decoding speed and error rates of inter-module measurement [Maurya et al. (2025)].

Increasing code and circuit distances With long-range coupling, can consider a wide variety of qLDPC codes.

Improved magic state factories Similarly, long range connectivity can be used for better magic state factories [Jacob et al. (2025); Li et al. (2025); Guanyu Zhu (2025)].

Opportunities for improvement

Reduce the time overhead The synthesis of arbitrary Pauli measurements incurs significant overhead.

Increase parallelism The Pauli-based compilation scheme leads to sequential T gate execution.

Decoding with speed and accuracy For instance, decoding speed and error rates of inter-module measurement [Maurya et al. (2025)].

Increasing code and circuit distances With long-range coupling, can consider a wide variety of qLDPC codes.

Improved magic state factories Similarly, long range connectivity can be used for better magic state factories [Jacob et al. (2025); Li et al. (2025); Guanyu Zhu (2025)].

Modifying the bicycle instructions Simplify circuits for bicycle instructions, especially for complicated and error-prone instructions.