Benchmarking Quantum Computers



A lecture series by IQM and HS RM

Authors: Stefan Seegerer and Nadia Milazzo

Last Updated 06/2025

www.meetiqm.com

What metric to use to make a buying decision?



I want to be able to carry more.



I want to go fast.



I want to protect the environment.

We want **measurements** about the car that helps our decision.

What metric to use to make a buying decision?



Trunk space or passenger count



Top speed or max acceleration



Emissions specs

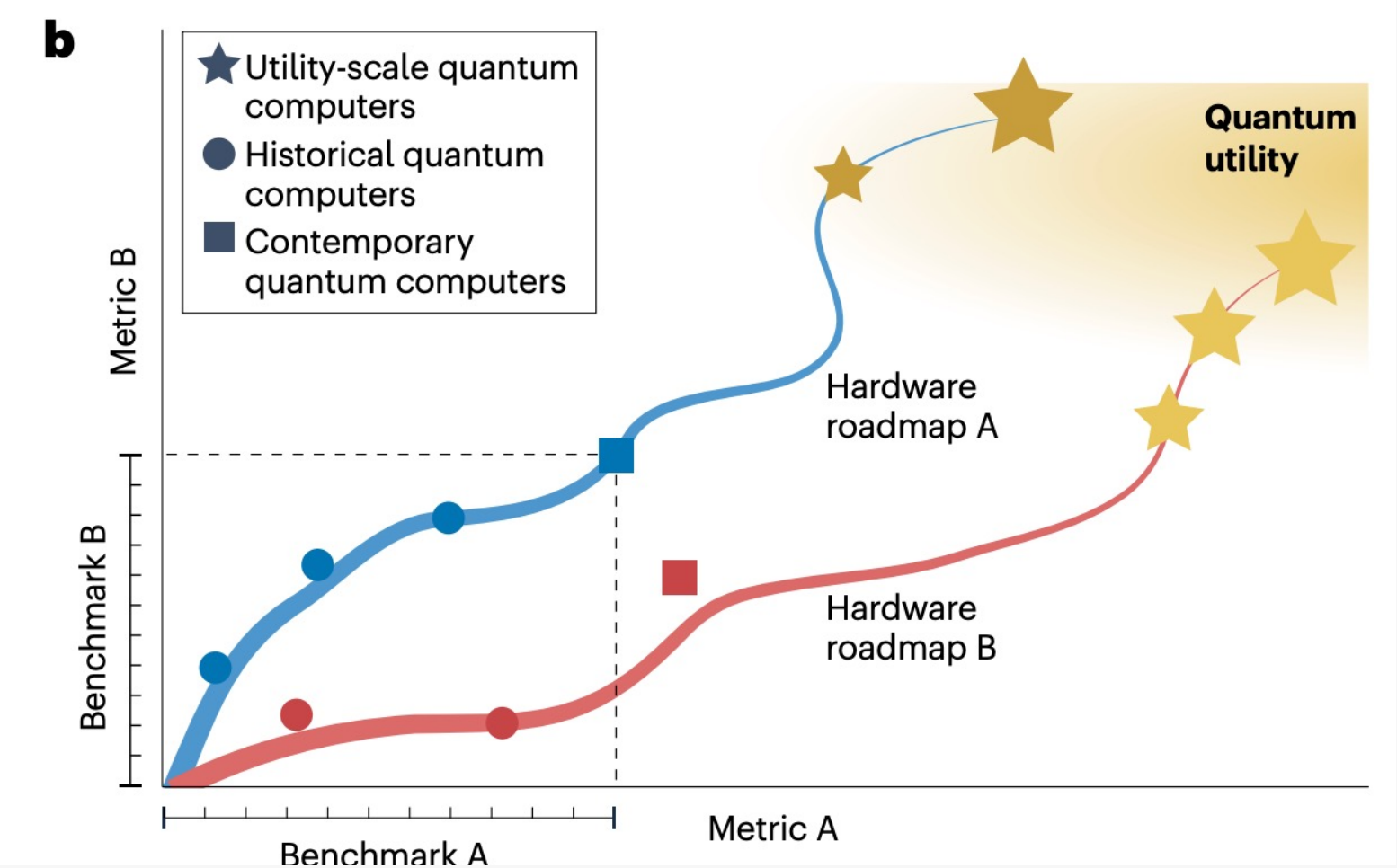
Standard benchmarks help us compare cars and check if a car suits our needs.

Classical Computer Benchmarking

- How fast can a computer perform a useful task? For example:
  - **LINPACK:** performing specific linear algebra tasks
  - **SPEC** (Standard Performance Evaluation Corporation): suite of tasks ranging from analyzing proteins to playing Go to compressing images
  - **Cinebench**: rendering images, videos, or animations

→ Different users care about different benchmarks

Quantum Computer Benchmarking



Benchmarks let us compare quantum computers and guide their progress towards practical usefulness.

Proctor, Timothy, et al. "Benchmarking quantum computers." Nature Reviews Physics 7.2 (2025): 105-118.

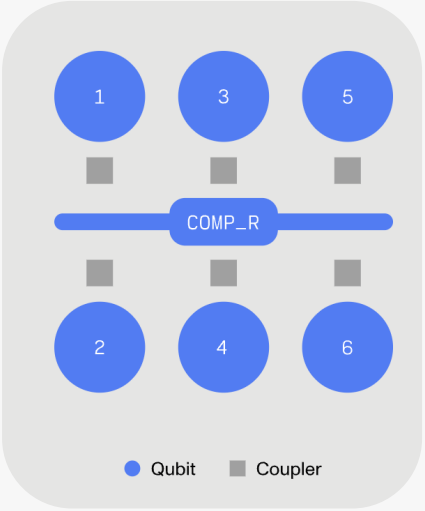
But how do we compare quantum computers?

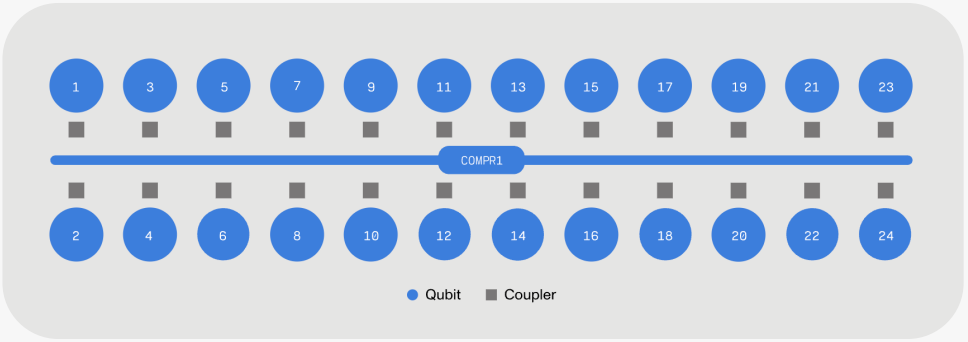
QUIZ TIME!

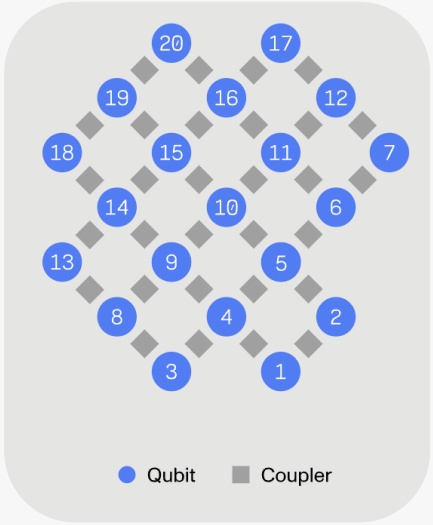


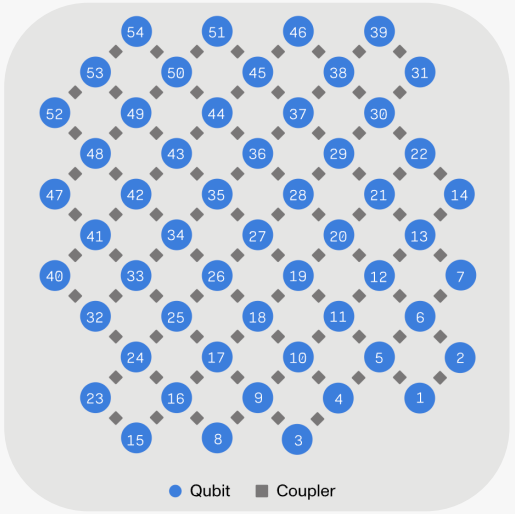
If you were buying a quantum computer, what would you want to know about it to make an informed decision?

A simple benchmark: Qubit count



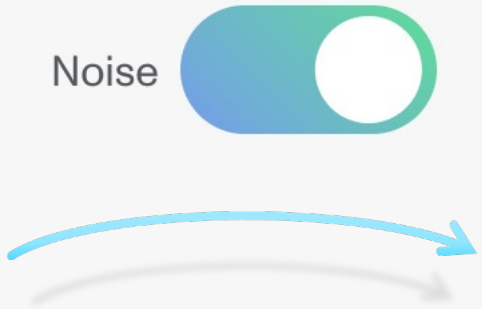






Qubit count isn’t enough…

1	|00000)	0% chance
2	|00001)	0% chance
3	|00010)	0% chance
4	|00011)	0% chance
5	|00100)	0% chance
6	|00101)	0% chance
7	|00110)	0% chance
8	|00111)	0% chance
9	|01000)	0% chance
10	|01001)	0% chance
11	|01010)	0% chance
12	|01011)	0% chance
13	|01100)	0% chance
14	|01101)	100% chance
15	|01110)	0% chance
16	|01111)	0% chance
17	|10000)	0% chance
18	|10001)	0% chance
19	|10010)	0% chance
20	|10011)	0% chance
21	|10100)	0% chance
22	|10101)	0% chance
23	|10110)	0% chance
24	|10111)	0% chance
25	|11000)	0% chance
26	|11001)	0% chance
27	|11010)	0% chance
28	|11011)	0% chance
29	|11100)	0% chance
30	|11101)	0% chance
31	|11110)	0% chance
32	|11111)	0% chance



1	|00000)	0% chance
2	|00001)	1% chance
3	|00010)	0% chance
4	|00011)	0% chance
5	|00100)	0% chance
6	|00101)	6% chance
7	|00110)	0% chance
8	|00111)	2% chance
9	|01000)	0% chance
10	|01001)	0% chance
11	|01010)	0% chance
12	|01011)	0% chance
13	|01100)	0% chance
14	|01101)	87% chance
15	|01110)	2% chance
16	|01111)	0% chance
17	|10000)	0% chance
18	|10001)	0% chance
19	|10010)	0% chance
20	|10011)	0% chance
21	|10100)	0% chance
22	|10101)	0% chance
23	|10110)	0% chance
24	|10111)	0% chance
25	|11000)	0% chance
26	|11001)	0% chance
27	|11010)	0% chance
28	|11011)	0% chance
29	|11100)	0% chance
30	|11101)	2% chance
31	|11110)	0% chance
32	|11111)	0% chance

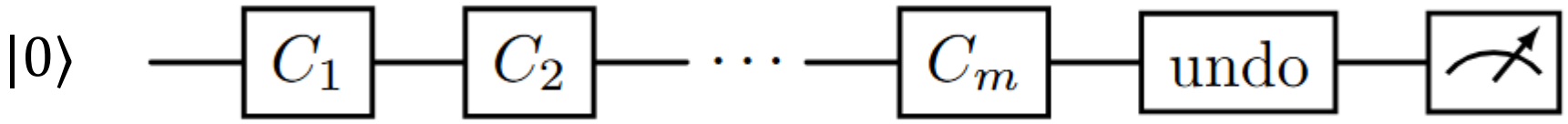
Errors have a significant impact on the performance of quantum computers, so most quantum computer benchmarks measure the impact of those errors.

Benchmarking gates

We need to be able to apply gates (manipulate qubits) without too much error. How do we measure this error?

Randomized Benchmarking

- Measures the error in applying a typical 1-qubit gate directly:

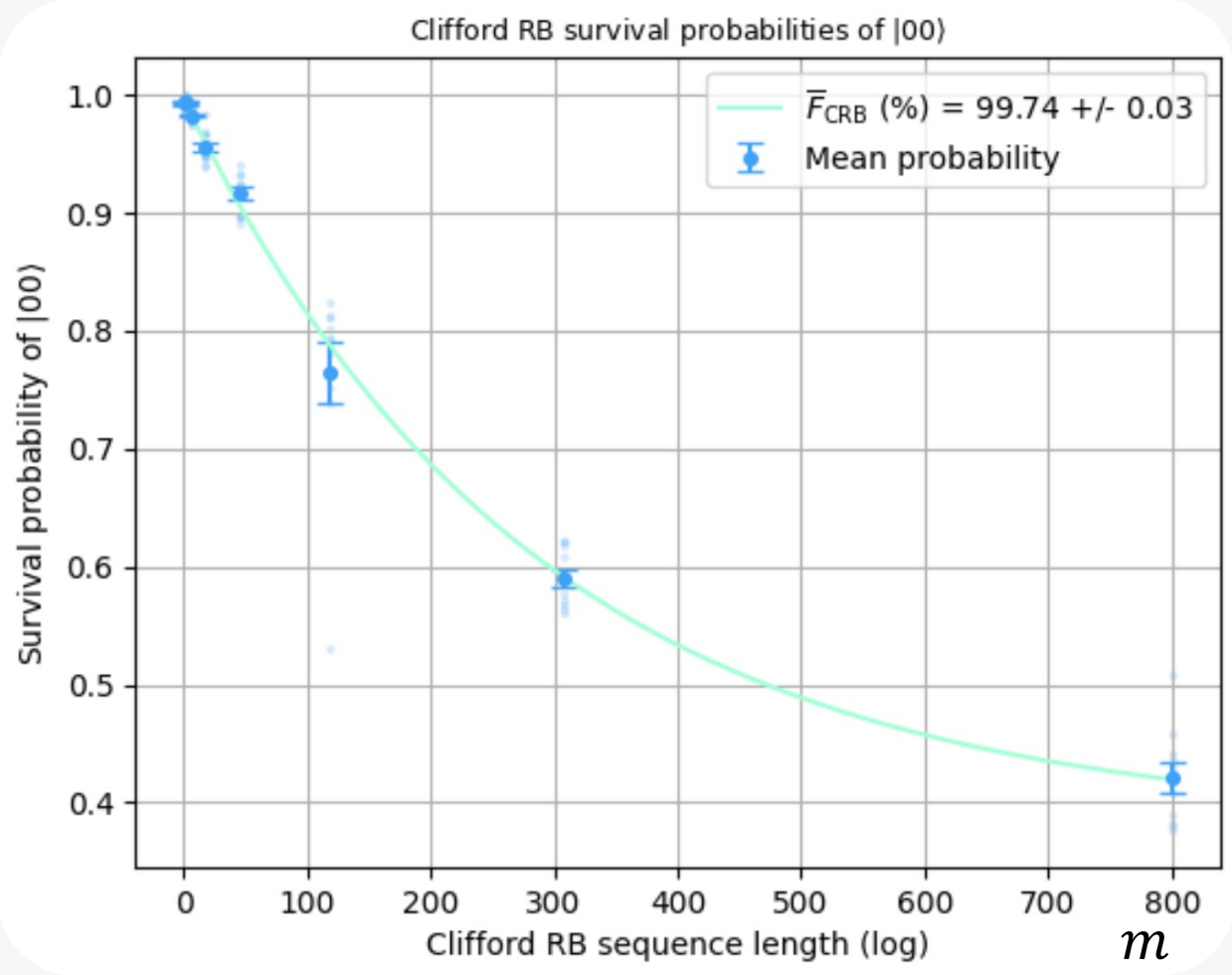


- The gates are randomly chosen*. We average over many attempts with many random choices.
- Thanks to “undo” gate (different for each random choice), no error means if we start in state $|0\rangle$, we should end in state $|0\rangle$ - the state “survives.”
- The probability of survival decreases exponentially as a function of m.

IQM

*For technical reasons, we choose them from a special set of gates called Clifford gates which are easy to simulate classically.

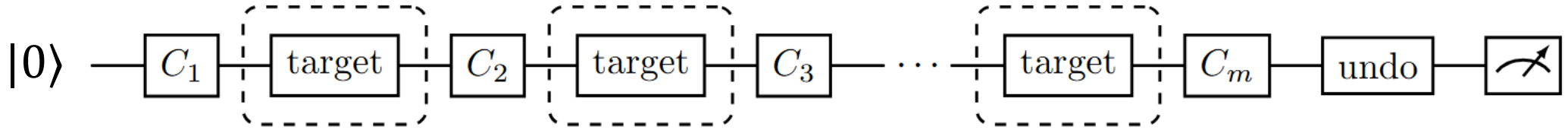
Randomized Benchmarking Result



Gate fidelity is the percent chance one gate succeeds. It can be extracted by fitting this graph. Faster drop = lower fidelity.

Interleaved Randomized Benchmarking

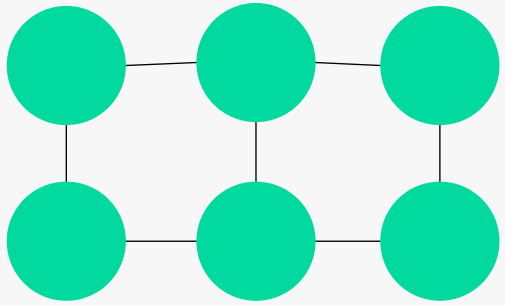
- Measures the fidelity of a specific target 1- or 2-qubit gate directly



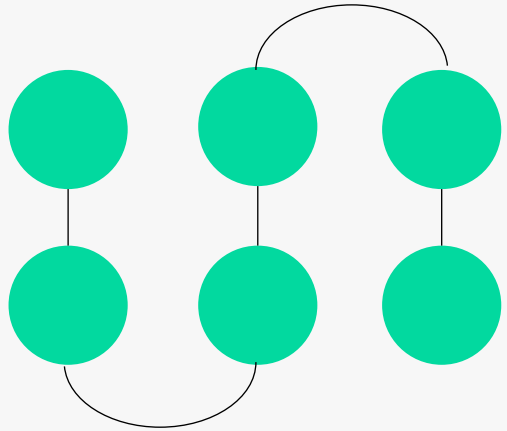
- We can extract the fidelity of the target gate from the survival probability as before (if we have already done standard randomized benchmarking).

- There are other variants of randomized benchmarking that measure related quantities, but the benchmarks can become more complicated.

Gates don’t give a full picture



vs.



- Low connectivity between qubits decreases algorithmic performance even if gate fidelity is high.
- Nearby qubits can interact in complex, non-random ways (crosstalk) and therefore lead to additional errors.
- Gate fidelity alone is not enough to assess the performance

Benchmarking entanglement

Entanglement is key to quantum algorithms. Can we measure how many qubits you can meaningfully entangle?

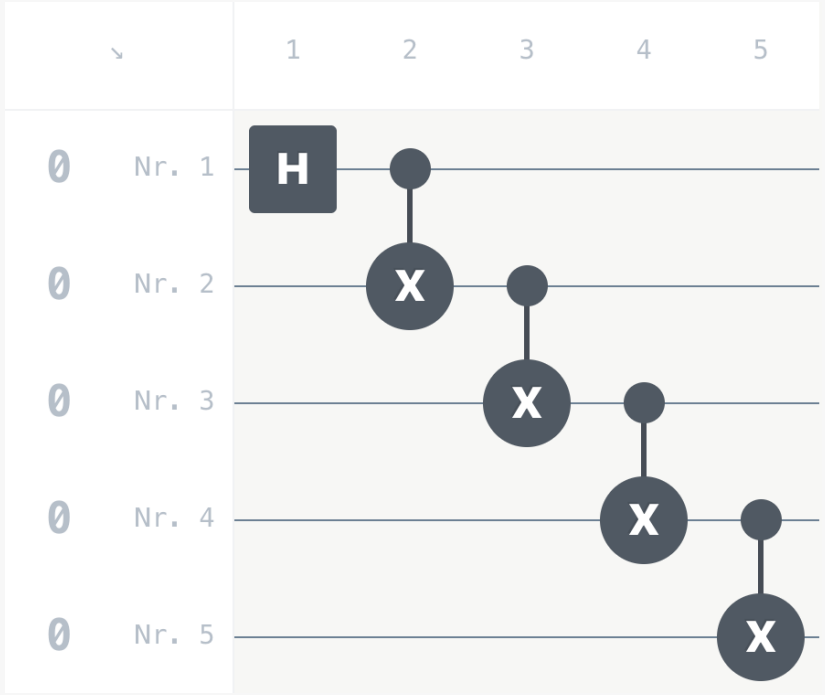
GHZ state fidelity

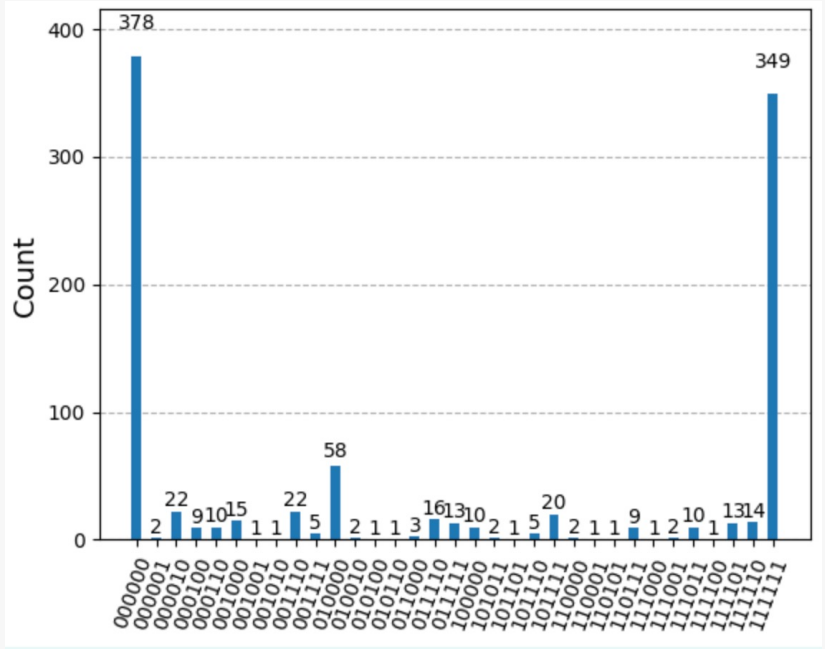
- The GHZ state is an entangled quantum state that involves at least three qubits

$$\frac{1}{\sqrt{2}}(|00\ldots0\rangle + |11\ldots1\rangle)$$

- With noise, measurement will lead to outcomes other than $00\ldots0$ or $11\ldots1$.

→ The question is: Can we measure if the qubits are truly "entangled"?




Count
400
300
200
100
0
378
349
58
22
22
20
16
15
14
13
10
10
9
5
5
3
2
2
2
2
2
2
2
1
1
1

GHZ state fidelity

- We can verify genuine multi-qubit entanglement by measuring the GHZ state fidelity

$$F(\text{ideal}, \text{measured}) = \left(Tr\sqrt{\sqrt{\rho_{ideal}}\,\rho_{measured}\sqrt{\rho_{ideal}}} \right)^2$$

- If F is above 0.5, you can **prove** that the measurement results cannot be explained without entanglement involving all of the qubits
  - This is a great way to evaluate the "quantumness" of the computer

- Problem: the probabilities of different outcomes only tells us the diagonal elements of the matrix $\rho_{measured}$
- To measure the off-diagonal entries, one can use the method of multiple quantum coherences (Mooney, 2021) – see the exercises for lecture 5.

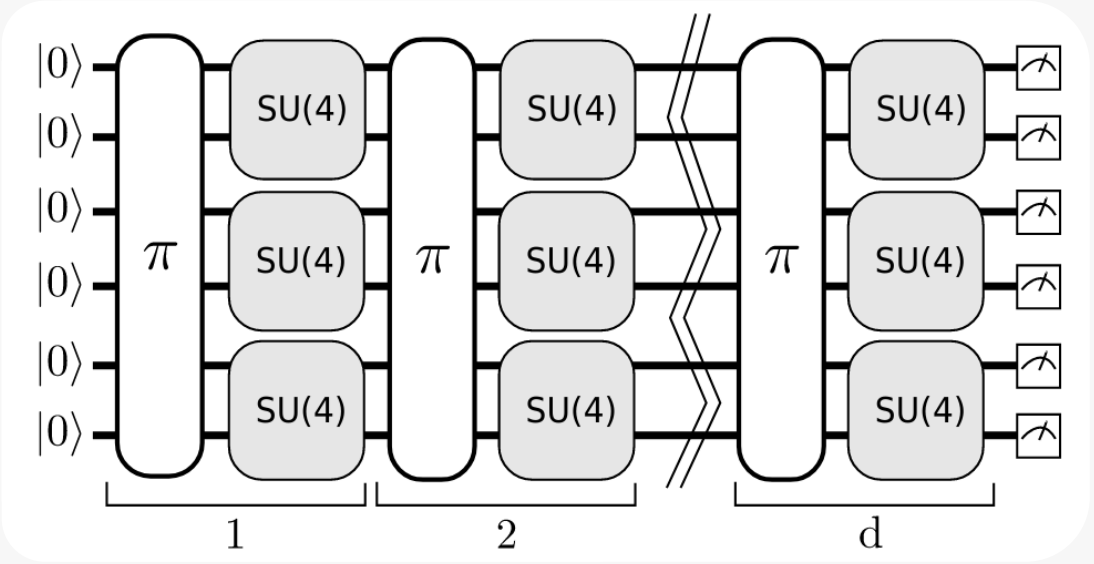
Holistic benchmarks

We need benchmarks for other important (classical) tooling: transpiler, compilers, scheduler, error reduction solutions, …

Quantum Volume

Measures how big of a quantum circuit of a certain type you can execute reliably.

1. Pick m qubits.
2. Run a certain type of random quantum circuit.
3. Test if the circuit was run successfully.
4. If so, repeat, but this time run two random circuits in succession.
5. Continue adding "layers" of quantum circuits until the circuit fails. Call this number of layers $d(m)$. Write down the smaller of m and $d(m)$.
6. Repeat for as many values of m as you can.



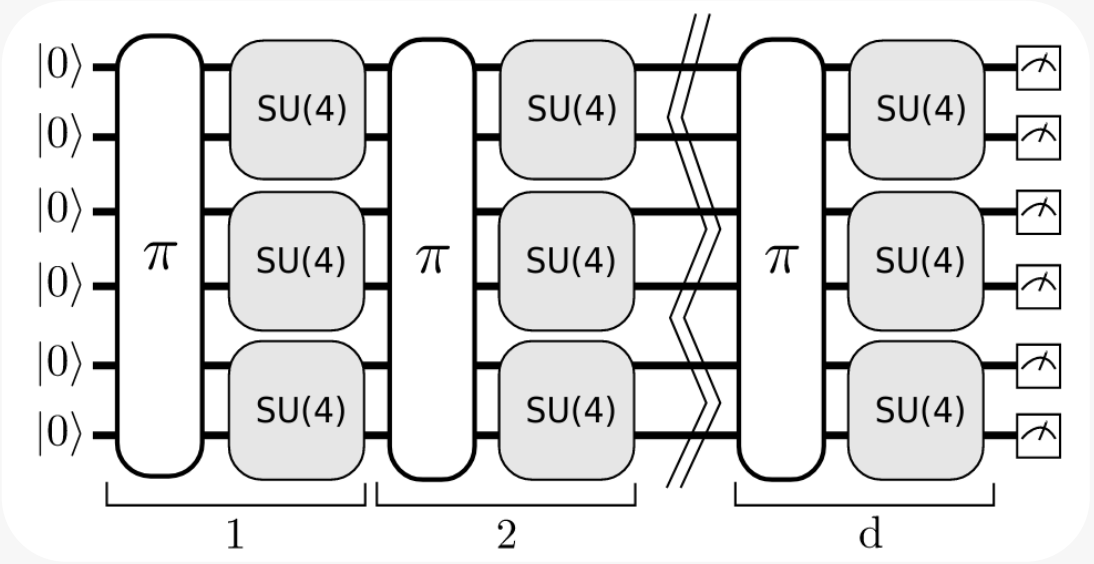
Cross et al., 2019

(The logarithm of the) quantum volume $\tilde{V}_Q$ is defined as the m which produces the largest number you have written. More precisely,

$$\log_2 \tilde{V}_Q = \mathrm{argmax}_m \min[m, d(m)]$$

Quantum Volume

- Quantum volume is holistic because it depends on:
  - Error rates of both gates and qubits, which decrease $d(m)$
  - Number and connectivity of qubits, which must be high to run big circuits
  - How you actually run the circuit – optimizing the transpiler, compiler, etc. can improve it.

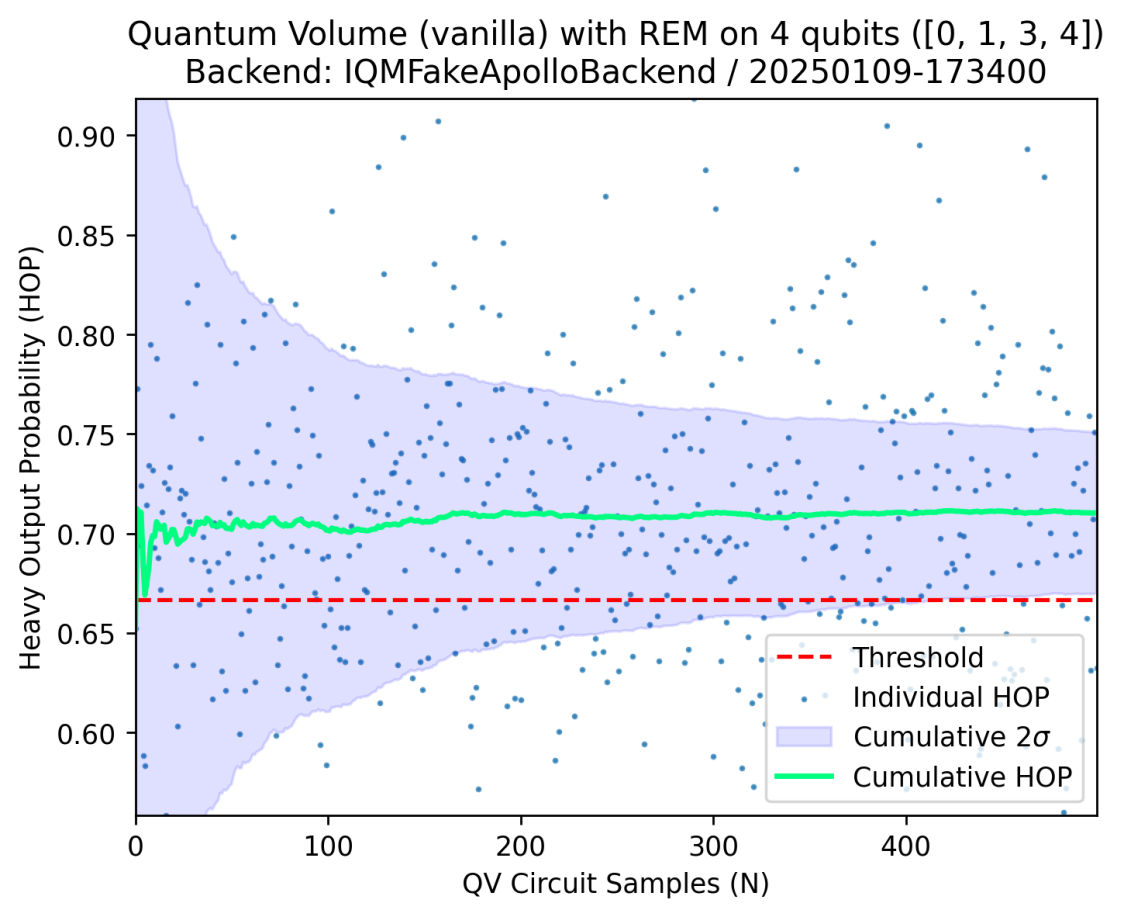


Cross et al., 2019

Quantum Volume – what does “running the circuit reliably” mean?

- When running random quantum circuits, we get a probability distribution over all possible outcomes (bitstrings)
- Some outputs are more likely than average – these are called “heavy outputs”
- An ideal quantum computer produces heavy outputs more often than a classical guess would
- So, measuring the Heavy Output Probability (HOP) tells us how “quantum” and reliable the device is

Quantum Volume – interpreting test results



- Y-axis : Probability that the quantum computer produces a "heavy" output.
- Red dashed line: The minimum success probability (2/3) that counts as passing the test.
- Blue dots: Success probability for each individual circuit run. These fluctuate a lot.
- Green line: Running average of all circuit results up to that point – smooths out the noise.
- Shaded blue region: Statistical confidence interval (±2 standard deviations) around the average – shows uncertainty.

Interpretation summary:

If the green line and the shaded blue area stay above the red threshold, the device can reliably pass this volume test.

Does this device pass this test?

CLOPS (Circuit layer operations per second)

- CLOPS quanitfies how fast a quantum computer can execute circuits – it estimates how long it takes to execute one layer of a quantum volume test (Wack, 2021 for one variant)

$$\text{CLOPS} = \frac{M \cdot K \cdot S \cdot \log_2 \tilde{V}_Q}{total\ time}.$$

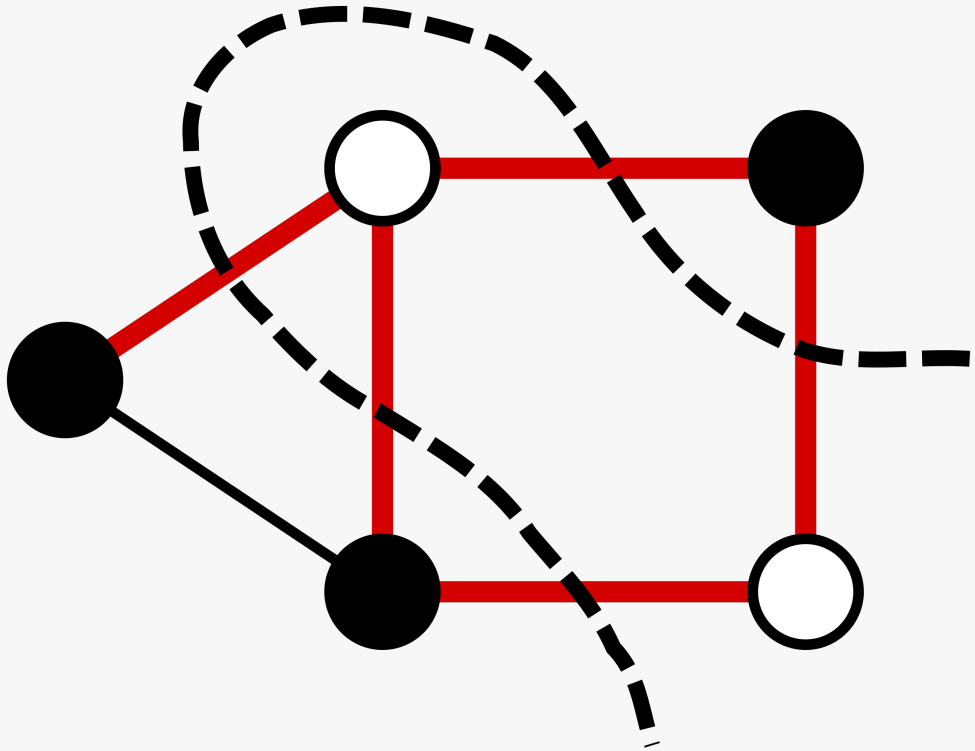
- M: number of different types of circuits (“templates”)
- K: number of updates made to the parameters of each template
- S: number of shots per circuit
- $\log_2 \tilde{V}_Q$: number of quantum volume layers

- Notice that the total CLOPS time includes that of assignment of parameters, submission of circuits and retrieval of results.

Holistic (algorithmic) benchmarks

Algorithmic Benchmarks

- A last category of benchmarks looks at what algorithmic problem classes can be solved with a given device
- Theoretically, many problems can be chosen here
  - A popular choice is solving MaxCut using QAOA (see lecture 7, and Martiel, Ayral, & Allouche, 2021)
  - Recall MaxCut: separating a group of people (represented as nodes in a graph) into two rooms. Goal is to break up as many friendships (represented as edges in the graph) as possible.



By Miym, CC BY-SA 3.0

Example: Q-Score

Measures how well a quantum computer does at solving the MaxCut problem

1. Pick a problem size n, the number of nodes in a random graph
2. Run the quantum algorithm (QAOA-MaxCut) on many randomly generated graphs of size n
   Compute $C(n) =$ -(average value of the cost function achieved by QAOA)
3. Compute the "β" score for size n

$$\beta(n) = \frac{\text{how much better than a random guess you have done}}{\text{typical cost of optimal solution}} = \frac{C(n) - \text{how well a random guess would do}}{\text{typical cost of optimal solution}}$$

$$= \frac{C(n) - \frac{n^2}{8}}{0.178n^{3/2}}$$

4. The device *passes* for size n if $\beta(n)$, is big enough (bigger than 0.2 in the original publication)
   - This means that the quantum device performs sufficiently better than random guessing.

Q-Score is the maximum n where the quantum computer passes the test

Conclusion: Benchmarking quantum computers

- Just counting qubits is not enough; errors and architecture matter
- Benchmarks provide a fair way to compare different quantum devices

Component benchmarks

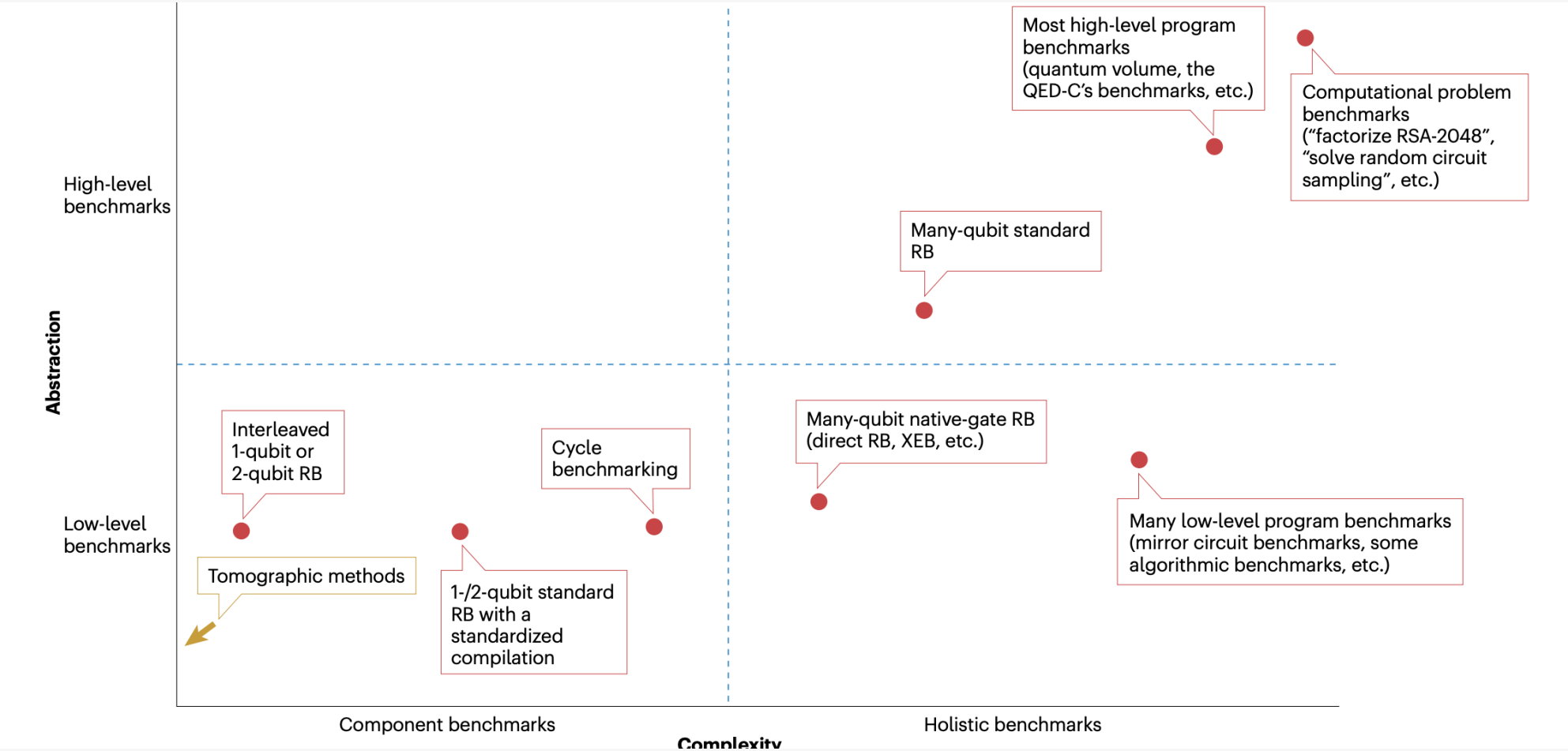
- Randomized Benchmarking
  - Measures average gate fidelities
- Entanglement Benchmarks
  - e.g., GHZ state fidelity, multiple quantum coherences

Holistic benchmarks

- Quantum Volume (QV)
  - Tests how well devices run random circuits (via Heavy Output Probability)
- CLOPS
  - Measures circuit throughput (speed)
- Algorithmic Benchmarks (e.g., Q-Score)
  - Test performance on specific problem classes (MaxCut via QAOA).

Both are needed to guide progress toward quantum utility / advantage.

Conclusion: Component benchmarks & holistic benchmarks



Proctor, Timothy, et al. "Benchmarking quantum computers." Nature Reviews Physics 7.2 (2025): 105-118.