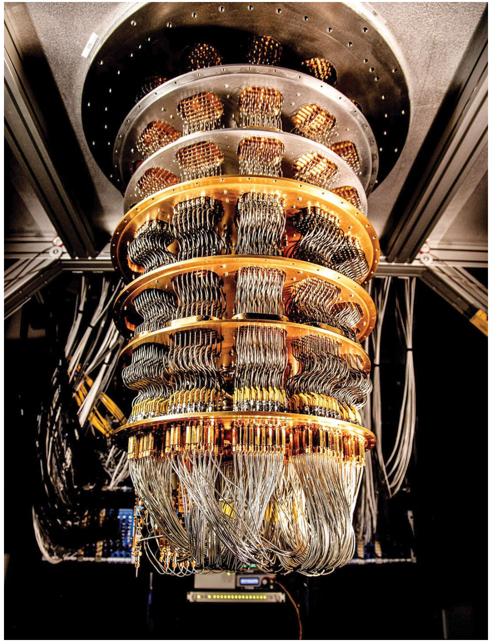
Validating quantum computers using randomized model circuits

Andrew W. Cross, Lev S. Bishop, Sarah Sheldon, Paul D. Nation, and Jay M. Gambetta

Reliability of Quantum Computers

- This paper is mainly to do with measuring reliability of Quantum Computers, why is that such a pressing issue?

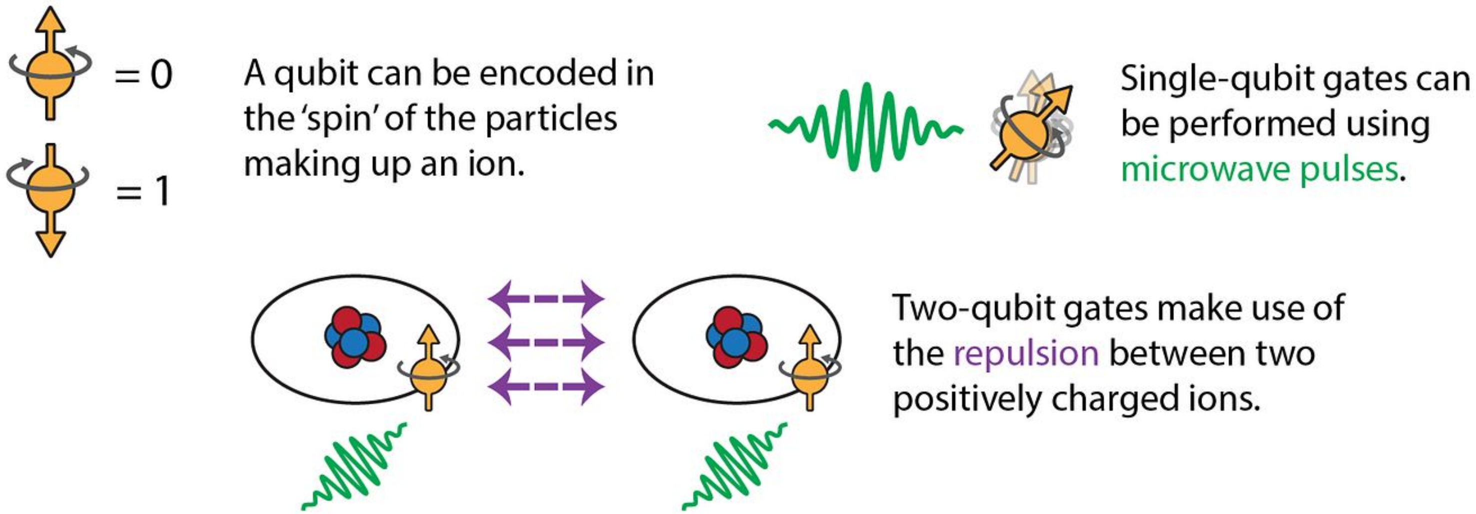


Quantum Gates

- Just as we have logic gates for traditional computing AND OR NOT, there are many quantum gates that can be stringed together to form the larger quantum computer.

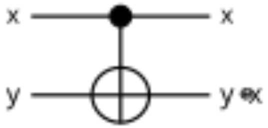
- However, we can’t just flip bits, the ways we interact with one qubit, and one gate doesn’t happen in isolation.

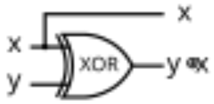
Quantum Gates



Example Quantum Gate

CNOT (Controlled Not)





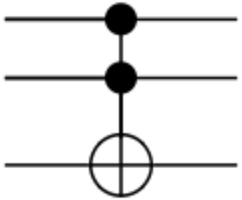
input		output	
x	y	x	y+x
$\vert 0\rangle$	$\vert 0\rangle$	$\vert 0\rangle$	$\vert 0\rangle$
$\vert 0\rangle$	$\vert 1\rangle$	$\vert 0\rangle$	$\vert 1\rangle$
$\vert 1\rangle$	$\vert 0\rangle$	$\vert 1\rangle$	$\vert 1\rangle$
$\vert 1\rangle$	$\vert 1\rangle$	$\vert 1\rangle$	$\vert 0\rangle$

input		output	
x	y	x	y+x
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

- One qubit is labeled the target and the other the control
- If the control qubit is 1 then it will flip the target
- In the left diagram y appears to be the target and x the control.

Example Quantum Gate

CCNOT or Toffoli Gate



Truth table

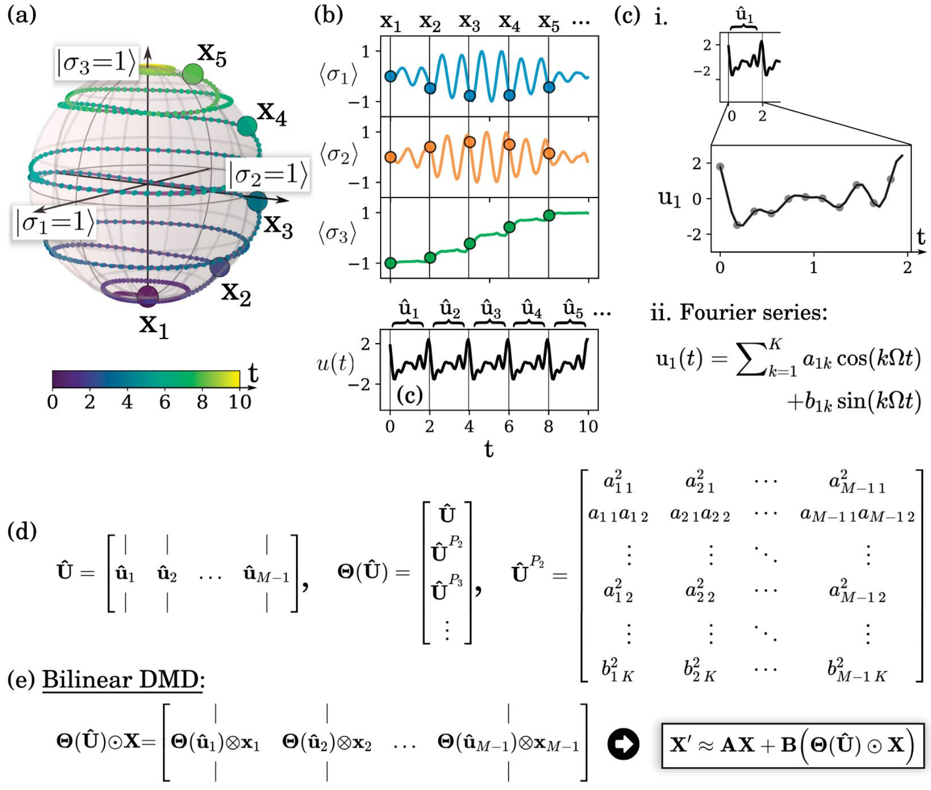
INPUT			OUTPUT		
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Similar to the CNOT but only applies the NOT operation to the third bit if it receives both of the other two inputs.

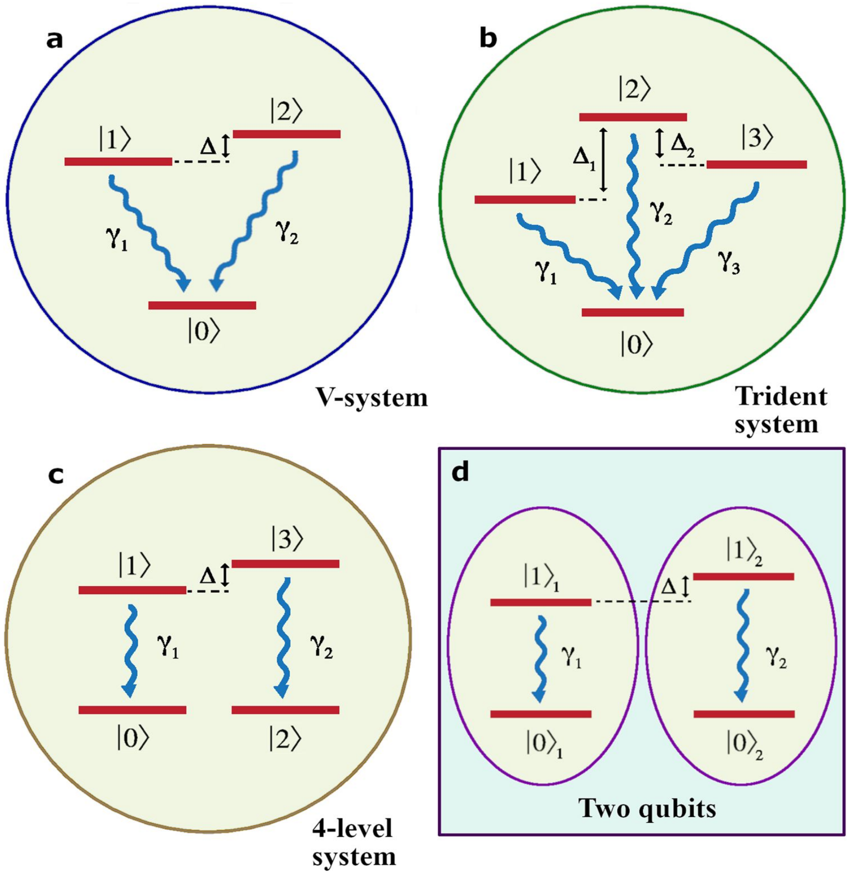
Quantum Control

There have been pre-existing methods of controlling and managing the interactions between quantum gates.

- Open Control
    - The idea that we can’t control these elements and that they will “unavoidably” interact and influence each other.
    - Ex. Markovian Master Equations (MME)
- Closed Control
    - This method assumes no input other than the direct input of the experimenter.
    - Unlike open control, this allows for planned input to influence the component deliberately.
    - Ex. Bilinear Models (BLM)
        - Manipulates spin and magnetic fields.



Bilinear Models (BLM)



Markovian Master Equations (MME)

What are we missing?

With Quantum Control we can try to identify how gates interact with each other, we can attempt to influence it. But we aren’t measuring it.

How would we measure the amount of error generated from gates interfering with each other?

Quantum Volume

Table 1: Quantum volume for some near-term devices

Device		
Topology	Error rate	Quantum Volume
IBM QX 5Q[a]	5×10^{-2}	16
5Q	10^{-2}	25
4×4	10^{-2}	36
4×4	10^{-3}	256
7×7	10^{-3}	256
7×7	10^{-4}	1296
10×10	10^{-4}	1296
10×10	10^{-5}	8100

[a] IBM Quantum Experience[1]

What is it?

Validating quantum computers using randomized model circuits

Andrew W. Cross,* Lev S. Bishop,† Sarah Sheldon, Paul D. Nation, and Jay M. Gambetta
IBM T. J. Watson Research Center, Yorktown Heights, NY 10598

We introduce a single-number metric, quantum volume, that can be measured using a concrete protocol on near-term quantum computers of modest size ($n \lesssim 50$), and measure it on several state-of-the-art transmon devices, finding values as high as 16. The quantum volume is linked to system error rates, and is empirically reduced by uncontrolled interactions within the system. It quantifies the largest random circuit of equal width and depth that the computer successfully implements. Quantum computing systems with high-fidelity operations, high connectivity, large calibrated gate sets, and circuit rewriting toolchains are expected to have higher quantum volumes. The quantum volume is a pragmatic way to measure and compare progress toward improved system-wide gate error rates for near-term quantum computation and error-correction experiments.

Here we propose an architecture-neutral metric, the *quantum volume*, to summarize performance against these factors. The quantum volume measures the useful amount of quantum computing done by a device in space and time. Table 1 summarizes predicted quantum volumes for potential near-term devices.

Basically, it's a performance metric

- ➔ Circuit depth, d
- ➔ Active Qubits, n'
- ➔ Effective Error Rate, ε_{eff}

$$\epsilon_{\text{eff}} = \epsilon_{\text{eff}}(n')$$

$$d = \frac{1}{n' \epsilon_{\text{eff}}(n')}$$

- ➔ Potential effective circuit depth as the inverse of n' * ε_{eff}

Quantum Volume lets us gauge performance with higher fidelity

Mathematical Definition

$$V_Q = \max_{n' \leq n} \min \left[n', \frac{1}{n' \epsilon_{\text{eff}}(n')} \right]^2.$$

V_Q	Quantum Volume
n′	Number of **Active** qubits
ε_{eff}	Effective Error Rate
d	Circuit depth

$$d = \frac{1}{n' \epsilon_{\text{eff}}(n')}$$

Mathematical Definition cont.

$$V_Q = \max_{n' \leq n} \left(\min(n', d)\right)^2$$

- For each possible n', return the min of n' and d

$$\min(n', d)$$

- Our Quantum Volume is the square of the maximum of all of these returned values.

Mathematical Definition Finale

$$V_Q = \max_{n' \leq n} \min\left[n', \frac{1}{n'\epsilon_{\mathrm{eff}}(n')}\right]^2.$$

n'	$\boldsymbol{\varepsilon}_{\mathrm{eff}}$ = 2n′	d	min(n',d)
1	1/5	5	Min = n' = 1
2	2/5	5/4	Min = d = 5/4
3	3/5	5/9	Min = d = 5/9
4	4/5	5/16	Min = d = 5/16
5	1	1/5	Min = d = 1/5

Table 1: Quantum volume for some near-term devices

Device		
Topology	Error rate	Quantum Volume
IBM QX 5Q[a]	5×10^{-2}	16
5Q	10^{-2}	25
4×4	10^{-2}	36
4×4	10^{-3}	256
7×7	10^{-3}	256
7×7	10^{-4}	1296
10×10	10^{-4}	1296
10×10	10^{-5}	8100

[a] IBM Quantum Experience[1]

As Error Rate falls, QV rises.

Measuring Quantum Volume

Measuring Quantum Volume

Measuring Quantum Volume takes a number of steps

1. Create a structure that we can standardize and closely examine.
2. Tune that structure to be representative of the machine we are measuring
3. Run Simulation
4. Evaluate output into Quantum Volume

Measuring Quantum Volume

Each structure that we are going to use to measure Volume involves a number of circuits.

These circuits are made up of a number of qubit gates and labels to distinguish each group.

A *model circuit*, shown in Fig. 1, with depth d and width m, is a sequence $U = U^{(d)} \dots U^{(2)} U^{(1)}$ of d layers

$$U^{(t)} = U^{(t)}_{\pi_t(m'-1),\pi_t(m')} \otimes \cdots \otimes U^{(t)}_{\pi_t(1),\pi_t(2)}, \qquad (1)$$

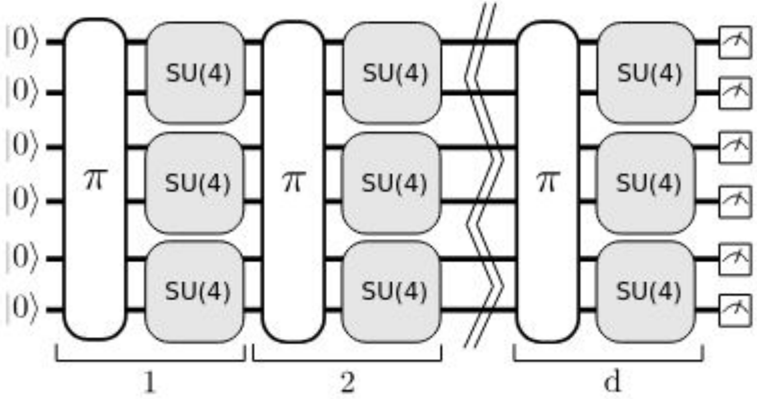


FIG. 1. **Model circuit.** A model circuit consists of d layers of random permutations of the qubit labels, followed by random two-qubit gates. When the circuit width m is odd, one of the qubits is idle in each layer. A final permutation can be applied to the labels of the measurement outcomes.

U and U’

- U refers to the model circuit.
- U’ is referring to the version of this circuit which is specifically tuned to the target machine.
- To achieve this we mirror the gates that are being used in the model circuit and limit our circuit to the same number of possible interactions between the gates.

$$F_{\text{avg}}(U,U') = \frac{\left|\mathrm{Tr}(U^\dagger U')\right|^2/2^m + 1}{2^m + 1}.$$

Evaluating Circuits

- Once we have finalized our circuits, we need to start evaluating them.
- To achieve this we use **heavy output**.

$$p_U(x) = |\langle x|U|0\rangle|^2$$

$$H_U = \{x \in \{0,1\}^m \text{ such that } p_U(x) > p_{med}\}.$$

Evaluating Heavy Output

- How to tell if output is successfully Heavy?

Algorithm 1 provides pseudocode for testing when each $h_d > 2/3$.

Algorithm 1 Check heavy output generation

function IsHeavy$(m, d; n_c \geq 100, n_s)$
  $n_h \leftarrow 0$
  for n_c repetitions **do**
    $U \leftarrow$ random model circuit, width m, depth d
    $H_U \leftarrow$ heavy set of U from classical simulation
    $U' \leftarrow$ compiled U for available hardware
    for n_s repetitions **do**
      $x \leftarrow$ outcome of executing U'
      if $x \in H_U$ **then** $n_h \leftarrow n_h + 1$
  return $\frac{n_h - 2\sqrt{n_h(n_s - n_h/n_c)}}{n_c n_s} > \frac{2}{3}$

Volume

- If the output is **heavy** then we can move on to quantum volume.
- We use both the number of qubits *m* and the depth *d(m)*

$$\log_2 V_Q = \underset{m}{\operatorname{argmax}} \min(m, d(m))$$

- The definition changed, the paper cites the ‘circuit structure’ as the reason.

$$V_Q = \max_{n' \leq n} \left(\min(n', d)\right)^2$$

Experiments

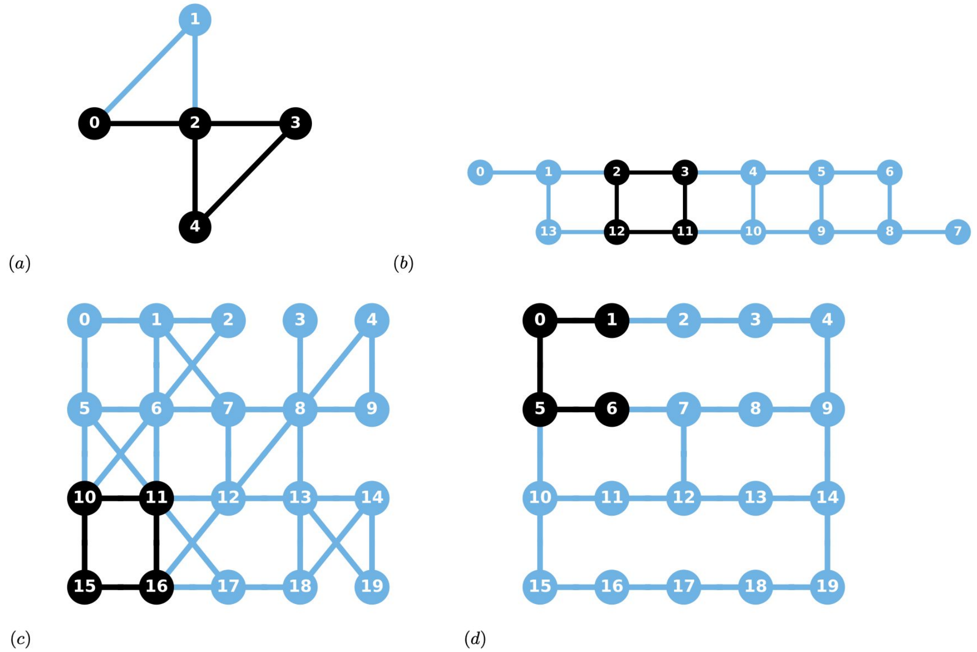
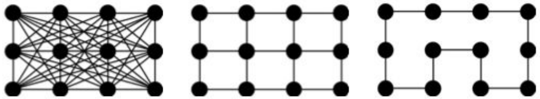


FIG. 8. Device diagrams used for the experimental data in Table I: (a) *Tenerife*, (b) *Melbourne*, (c) *Tokyo*, and (d) *Johannesburg*. The highlighted qubits are those selected for the experiments discussed here. CX gates are available between pairs of qubits connected by a highlighted line.

Goal: Measure QV of a QC while running the Heavy Output Generation Algorithm

➔ Step One: Simulate a QC running Heavy Output Generation on a classical machine.

➔ Step Two: Calculate the QV of said simulation to provide a baseline against which to compare our actual results.



$\log_2 V_Q$	All-to-All	Square Grid	Loop
4	0.03	0.028	0.028
6	0.015	0.011	0.011
8	0.008	0.005	0.0047
12	0.0032	0.0015	0.0014

	Standard	KAK	1% approx.	3% approx.	5% approx.
Average # CX Gates	28.1	21.0	17.7	16.1	15.1
Noisy Simulation	0.676 (0.003)	0.687 (0.004)	0.693 (0.004)	0.692 (0.004)	0.685 (0.005)
Experiment	0.614 (0.003)	0.632 (0.005)	0.649 (0.005)	0.647 (0.005)	0.646 (0.005)

Experimental Procedure cont.

- Step Three: Run Heavy Output on 4 different IBM Quantum Machines, measuring QV for each.

Circuit	Tenerife	Melbourne	Tokyo	Johannesburg
$m = d = 2$	0.685 (0.001)*	0.638 (0.006)	0.718 (0.006)	0.711 (0.006)
$m = d = 3$	0.651 (0.006)	0.641 (0.009)	0.682 (0.002)*	0.729 (0.007)
$m = d = 4$	0.516 (0.002)	0.523 (0.002)	0.614 (0.003)	0.664 (0.004)
$m = d = 4\dagger$			0.649 (0.005)	0.699 (0.001)**
$m = d = 5$				0.601 (0.004)

TABLE I. Experimentally estimated heavy output probabilities for four IBM Q devices: 5-qubit *Tenerife*, 16-qubit *Melbourne*, 20-qubit *Tokyo*, and 20-qubit *Johannesburg*, for circuits of equal width m and depth d. For each m, 200 circuits were run on every device. The experiments (*/**) were repeated with (5000/1000) circuits to ensure a 97.5% one-sided confidence interval as descriped in Appendix C. $m = d = 4\dagger$ experiments used circuits optimized with the KAK and approximate SU(4) decompositions assuming a 1% CX error rate.

Physical Results vs Simulation

➔ Step Four: Compare measured QV's against predicted QV's.

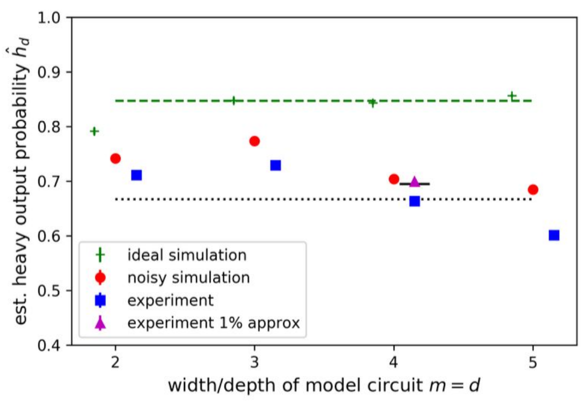


FIG. 3. Experimental data for square (width = depth) quantum volume circuits using the IBM Q Johannesburg 20-qubit device. As in Figure 2, the ideal simulation results are green

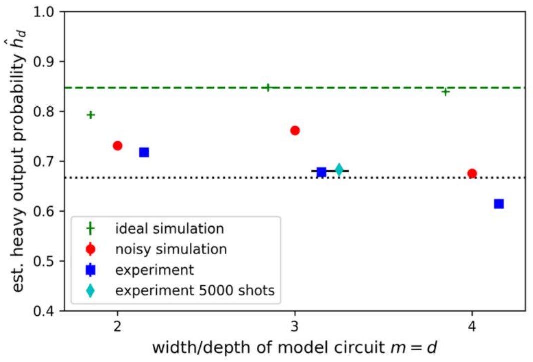


FIG. 2. Experimental data for square (width = depth) quantum volume circuits using the IBM Q 20-qubit device, *Tokyo*. The ideal simulation results are green plus signs. The noisy simulations, using a depolarizing noise model with average error rates from the qubits used on the device, are red circles. The experiments using 200 circuits are blue squares. The dotted line is the threshold of 2/3 for heavy output generation, and the dashed (green) line is the asymptotic ideal heavy output probability of $\frac{1+\ln 2}{2}$ [19], which the ideal simulations quickly approach. In order to set a high confidence level that h_d surpasses the threshold, the point at $m = d = 3$ was repeated with 5000 circuits (cyan diamond). This number of shots corresponds to a stricter threshold of 0.68 indicated by the solid line at the experimental points for $m = 3$.

Conclusions?

- ➔ Current machines far from ideal (obviously)
- ➔ That said, Quantum Volume CAN be reliably measured on different machines
- ➔ Having a unified means of measuring success will go a long way to validating machine performance.
- ➔ Potential increases of quantum volume from different qubit configuration schema

Takeaways

Takeaways

Some things this paper was able to solidify:

- A universal measurement standard that takes into account
  - Differences in implementation (how many gates, how much interference are they generating)
  - Differences in design (the layout and type of gates)

- A type of comparison that is completely agnostic to the specific architecture, as long as it can run quantum circuits it can be measured with quantum volume.

Takeaways

At the end of this paper, the authors didn’t outline what they are going to do next. So I thought to take a look at what they are researching currently and see how it involves quantum volume/quantum computers.

A huge part of this paper was developing a standard that can be used across all implementations, they are currently expanding on that with **Quantum Assembly Language.**

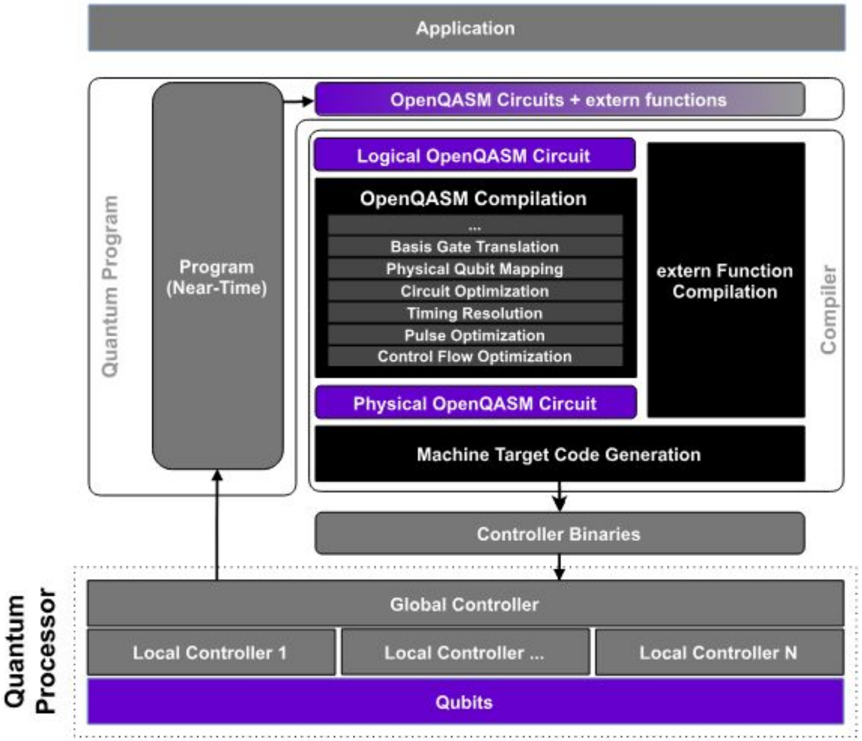
Quantum Assembly Language

Quantum Assembly language is a language to help define quantum circuits and expand their functionality.

This paper introduces many features quantum circuits haven't had up until this point, particularly **control flow** allowing for both conditionals and loops!

This language also integrates traditional computing into quantum computing to allow for the use of traditional data types (boolean, int, etc.)

Quantum Assembly Cont.



Merits and Shortcomings

Merits

- ➔ Effective use of diagrams / tables to summarize findings
- ➔ Thorough appendices detailing their Qiskit implementations of model circuits
- ➔ Expansion of potential research by simulating various qubit connection schema

Shortcomings

- ➔ Some reference hyperlinks 404
- ➔ Discrepancies in mathematical symbology from previous papers
- ➔ Somewhat difficult to parse out their experimental methods