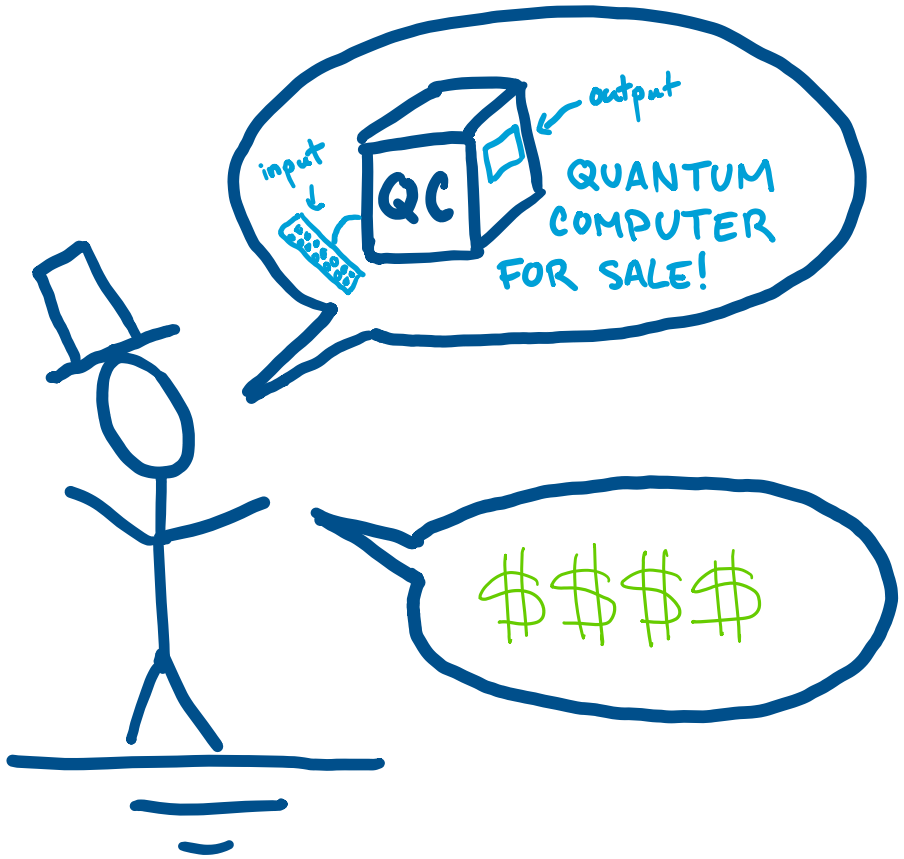
Is my device really quantum?

Demonstrating quantum supremacy with today's devices

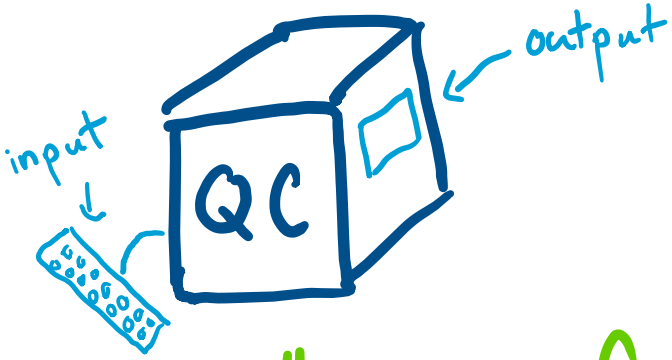
Chinmay Nirkhe

nirkhe@cs.berkeley.edu cs.berkeley.edu/~nirkhe

Berkeley
UNIVERSITY OF CALIFORNIA

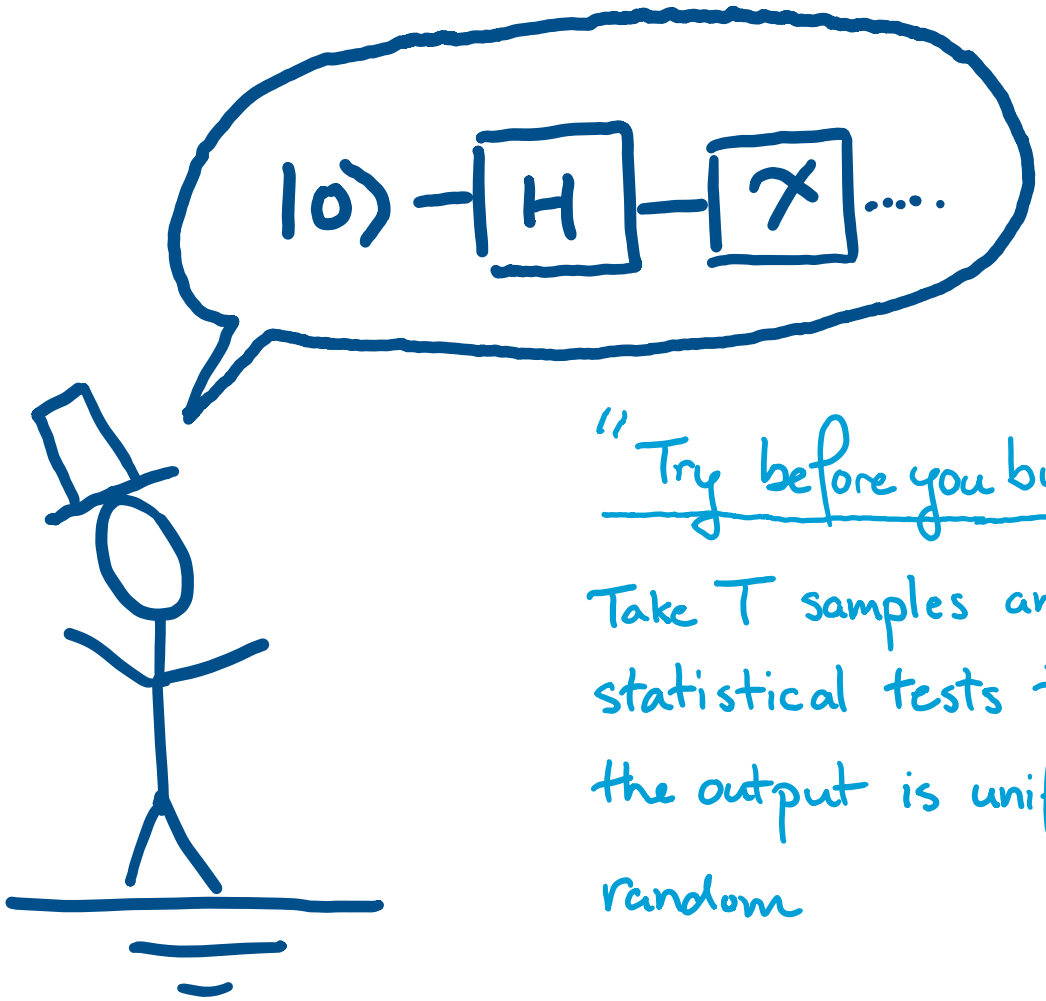


How do you tell if the box is actually a quantum computer?



"black-box" model of testing

Warm-up: Randomness generation



How do you test this is the ckt?

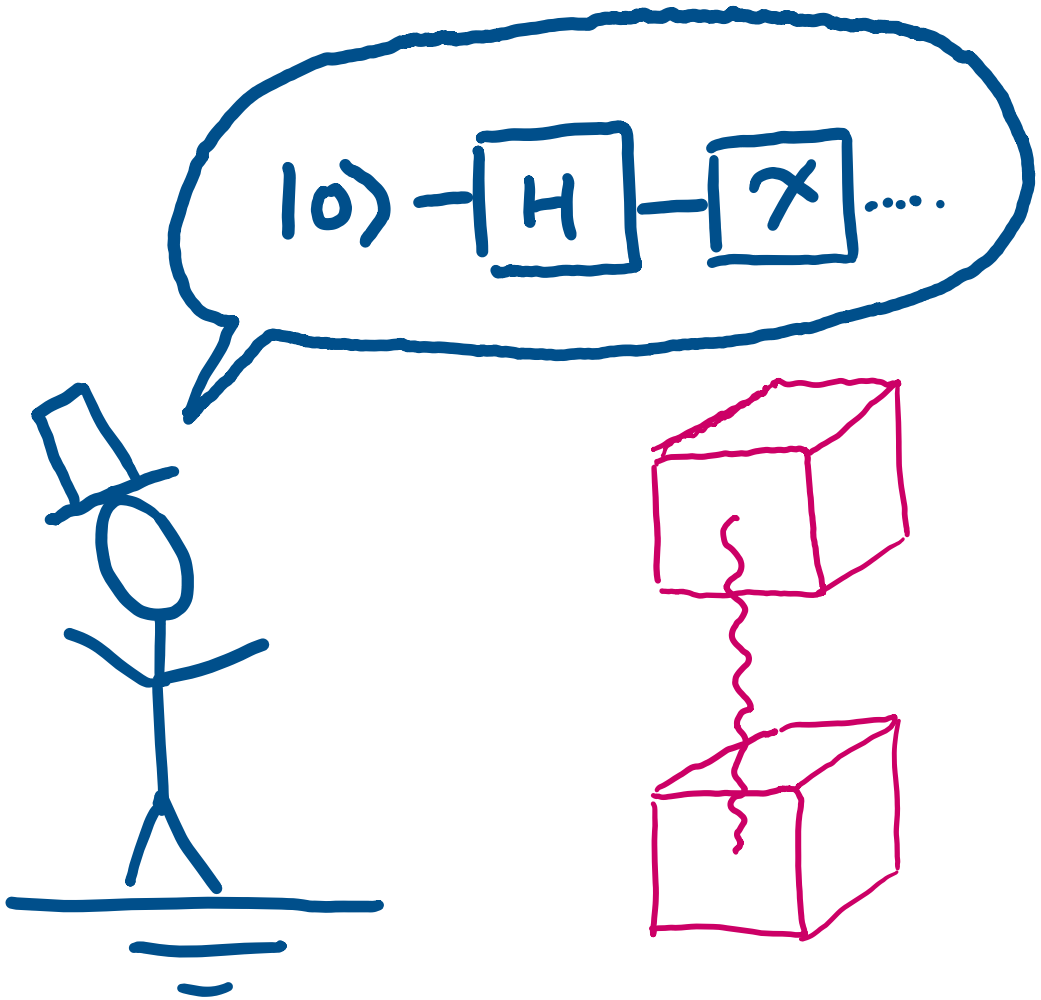
"Try before you buy"

Take T samples and run statistical tests to ensure the output is uniformly random

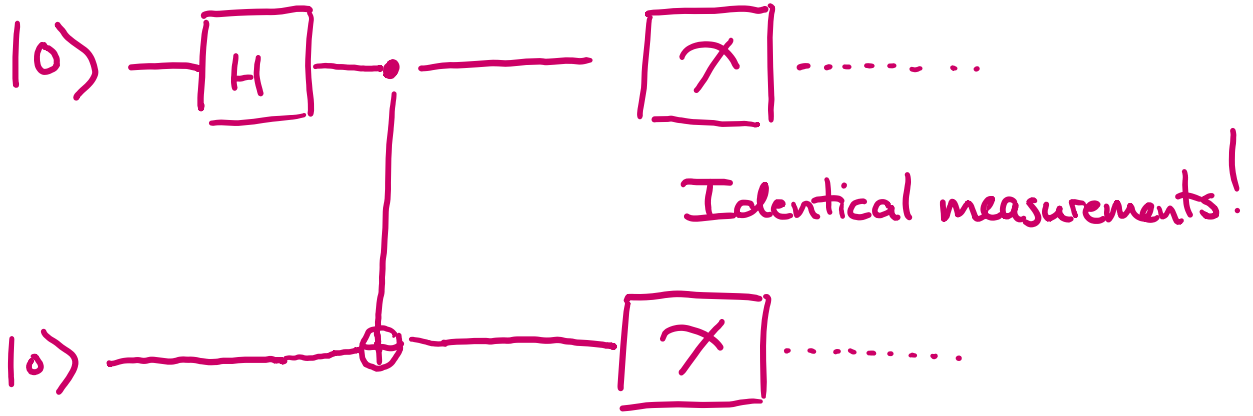
Issue

First 100T could be uniformly random and then afterwards according to the manufacturer's preferance.

Warm-up: Randomness generation



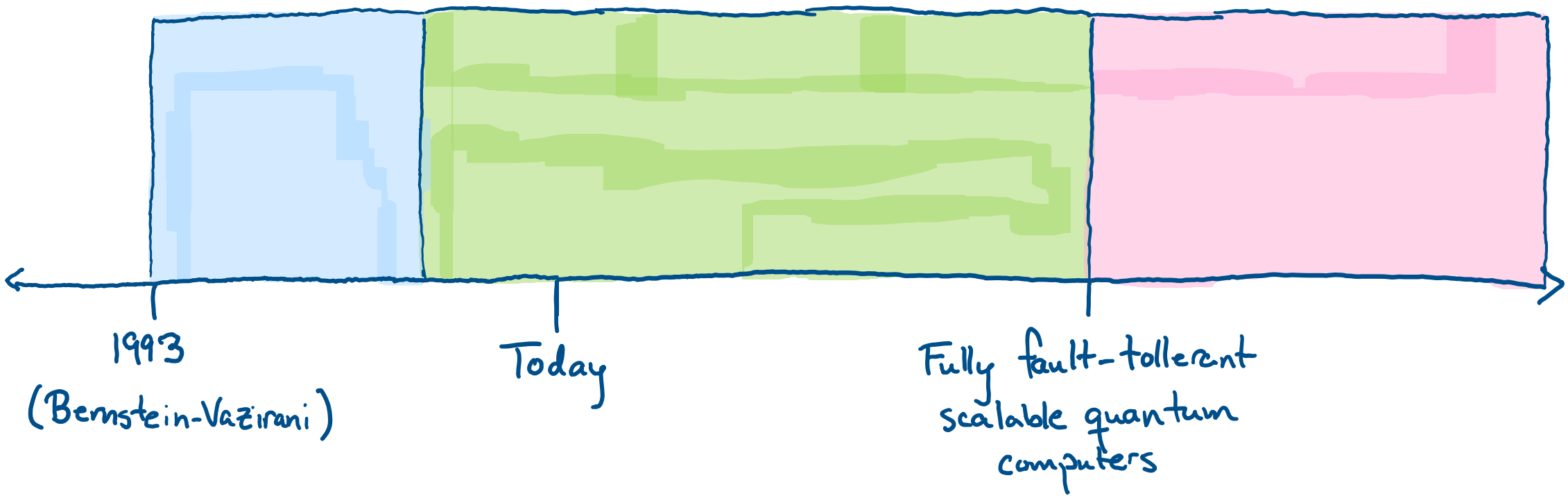
How do you test this is the ckt?



Identical measurements!

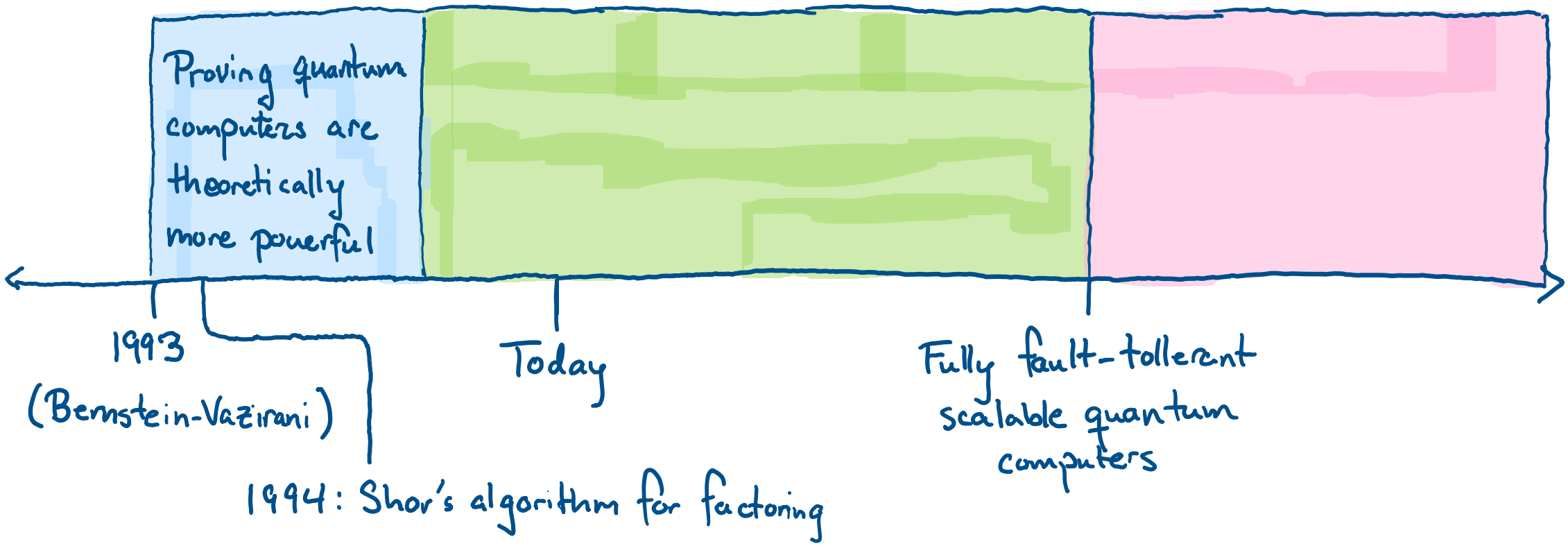
Timeline for this question

How do you tell if the box is an actual quantum computer?



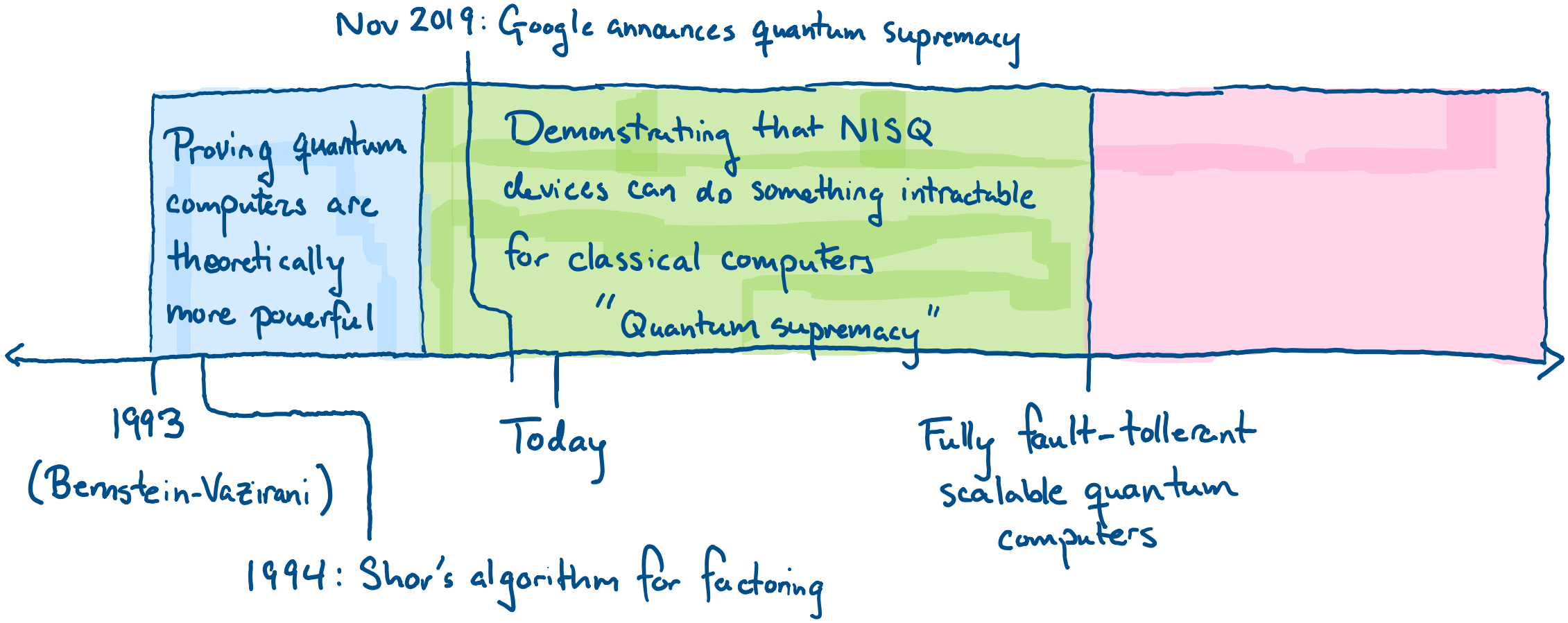
Timeline for this question

How do you tell if the box is an actual quantum computer?



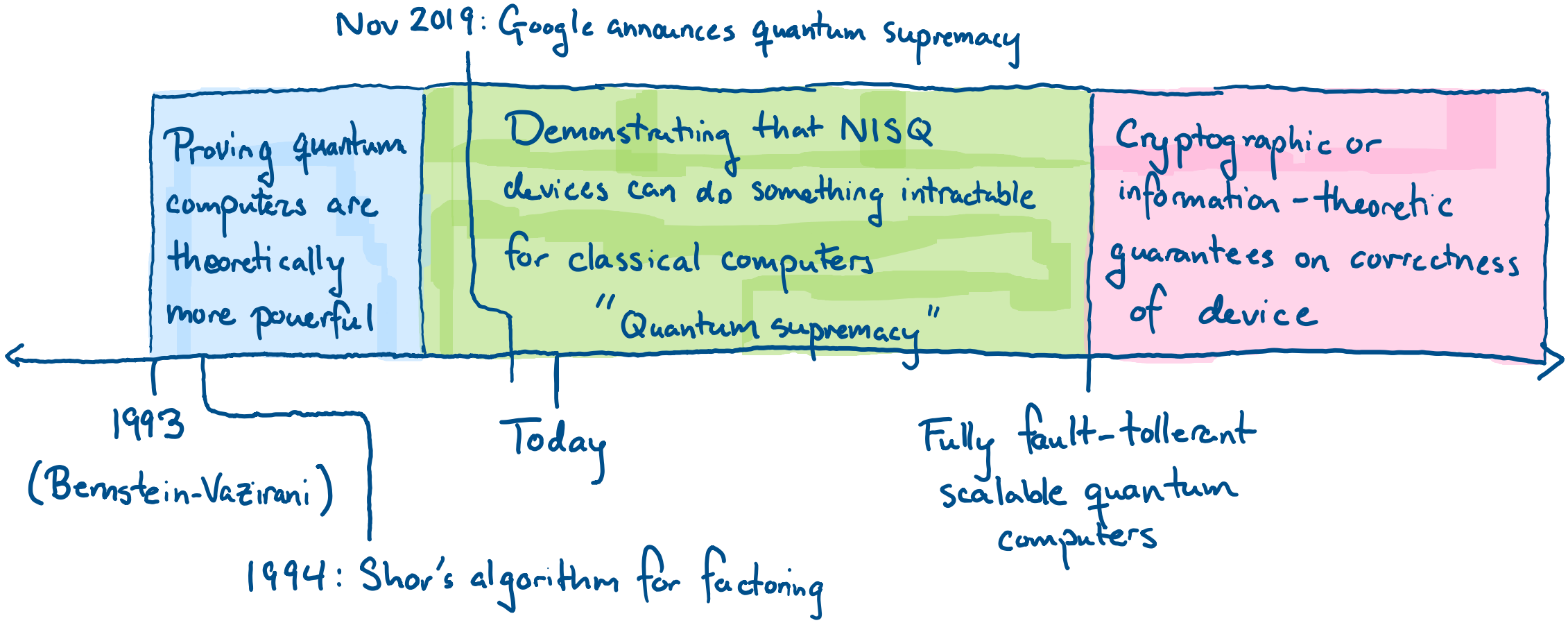
Timeline for this question

How do you tell if the box is an actual quantum computer?



Timeline for this question

How do you tell if the box is an actual quantum computer?



Step 1: Disprove the Extended Church-Turing thesis

give theoretical evidence that quantum computers can perform tasks intractable for classical computers

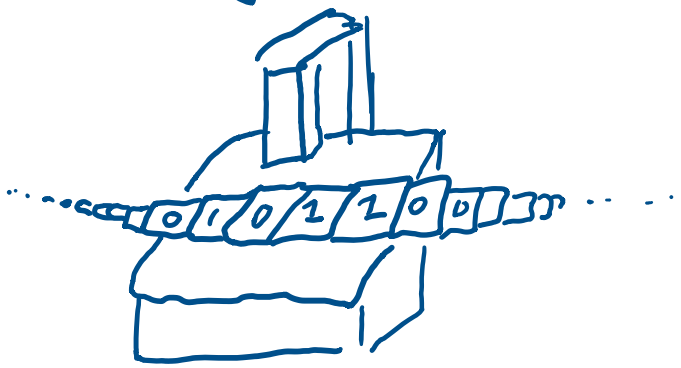
Turing machines

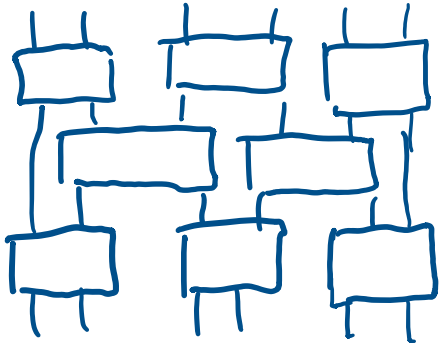
efficiently simulable

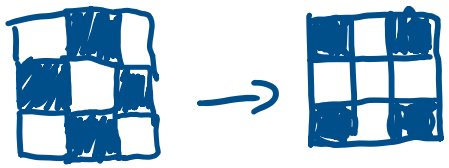
Uniform Circuits

efficiently simulable

Cellular Automata







ETC : Any "reasonable" model can be simulated efficiently on a standard model.

Step 1: Disprove the Extended Church-Turing thesis

give theoretical evidence that quantum computers can perform tasks intractable for classical computers

① $\exists\, O$ s.t. $BPP^O \neq BQP^O$ [Bernstein-Vazirani[93], Simon[93]]

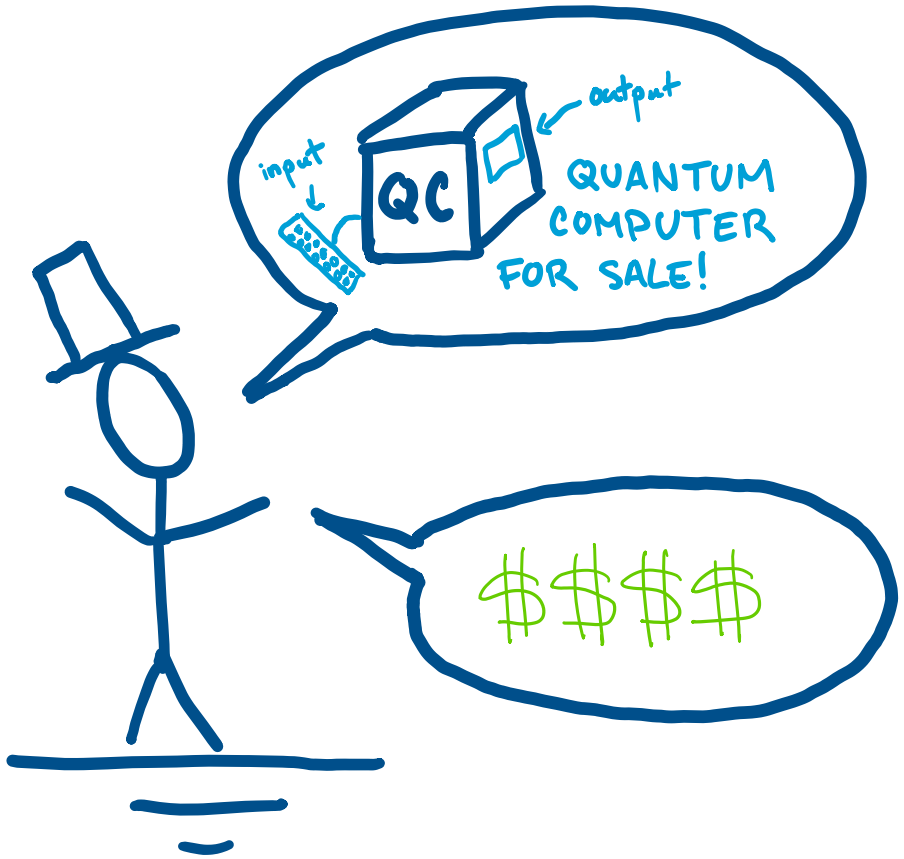
② FACTORING $\in$ BQP [Shor[94]]

$$BQP = \left\{ \begin{array}{c} \text{languages decidable by a polynomial} \\ \text{time quantum algorithm} \end{array} \right\}.$$

Step 1: Disprove the Extended Church-Turing thesis

give theoretical evidence that quantum computers can perform tasks intractable for classical computers

$$BQP = \{ \text{languages decidable by a polynomial time quantum algorithm} \}.$$



How do you tell if the box is actually a quantum computer?

Requirement:

Have it run a task (theoretically) intractable for classical computers.

Step 2: Experimental Progress

rigetti

Alibaba Group
阿里巴巴集团

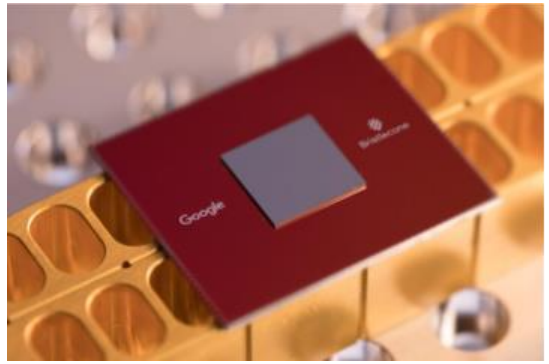
JOINT CENTER FOR QUANTUM INFORMATION AND COMPUTER SCIENCE

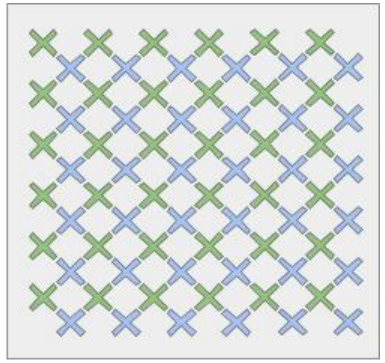
ZAPATA

Microsoft

XANADU

TU Delft Delft University of Technology





Google

TECH & SCIENCE

REVOLUTIONARY QUANTUM COMPUTER IS ONE STEP CLOSER TO REALITY AFTER MAJOR BREAKTHROUGH

BY **ARISTOS GEORGIOU** ON 3/8/18 AT 9:22 AM

And hype...

Is Government Ready for the Brewing Quantum Storm?

China's race for the mother of all supercomputers just got more crowded

Baidu, Alibaba and Tencent jockey for position in the development of quantum computing, which delivers a faster and more efficient approach to processing information than today's fastest computers

Why law firms need to worry about quantum computing

BY AGNESE SMITH December 7, 2018

Safe and secure with blockchain

Will quantum computing break blockchain?

December 12, 2018 Gary Stevens

TECH & SCIENCE

REVOLUTIONARY QUANTU COMPUTER IS ONE STEP CL REALITY AFTER MAJOR BREAKTHROUGH

BY ARISTOS GEORGIOU ON 3/8/18 AT 9:22 AM

China's race fo

And

Researchers Reverse the Flow of Time on a Quantum Computer

Time reversal may actually be possible. The quantum research could also help the world build better quantum computers.

Why law firms need to worry about quan computing

BY AGNESE SMITH December 7, 2018

Is Government Ready for the Brewing

SCIENTISTS HAVE REVERSED TIME IN A QUANTUM COMPUTER

BY HANNAH OSBORNE ON 3/13/19 AT 7:13 AM EDT

TECH & SCIENCE

Is Government Ready for the Brewing

REVOLUTIONARY QUANTUM SCIENTISTS HAVE REVERSED TIME IN A COMPUTER IS C
REALITY AFTEF
BREAKTHROUC

BY ARISTOS GEORGIOU ON 3/8/18 AT 9

And

Researchers Reverse t
Computer

Time reversal may actually be possible. The c
computers.

Why law firms need to v
computing

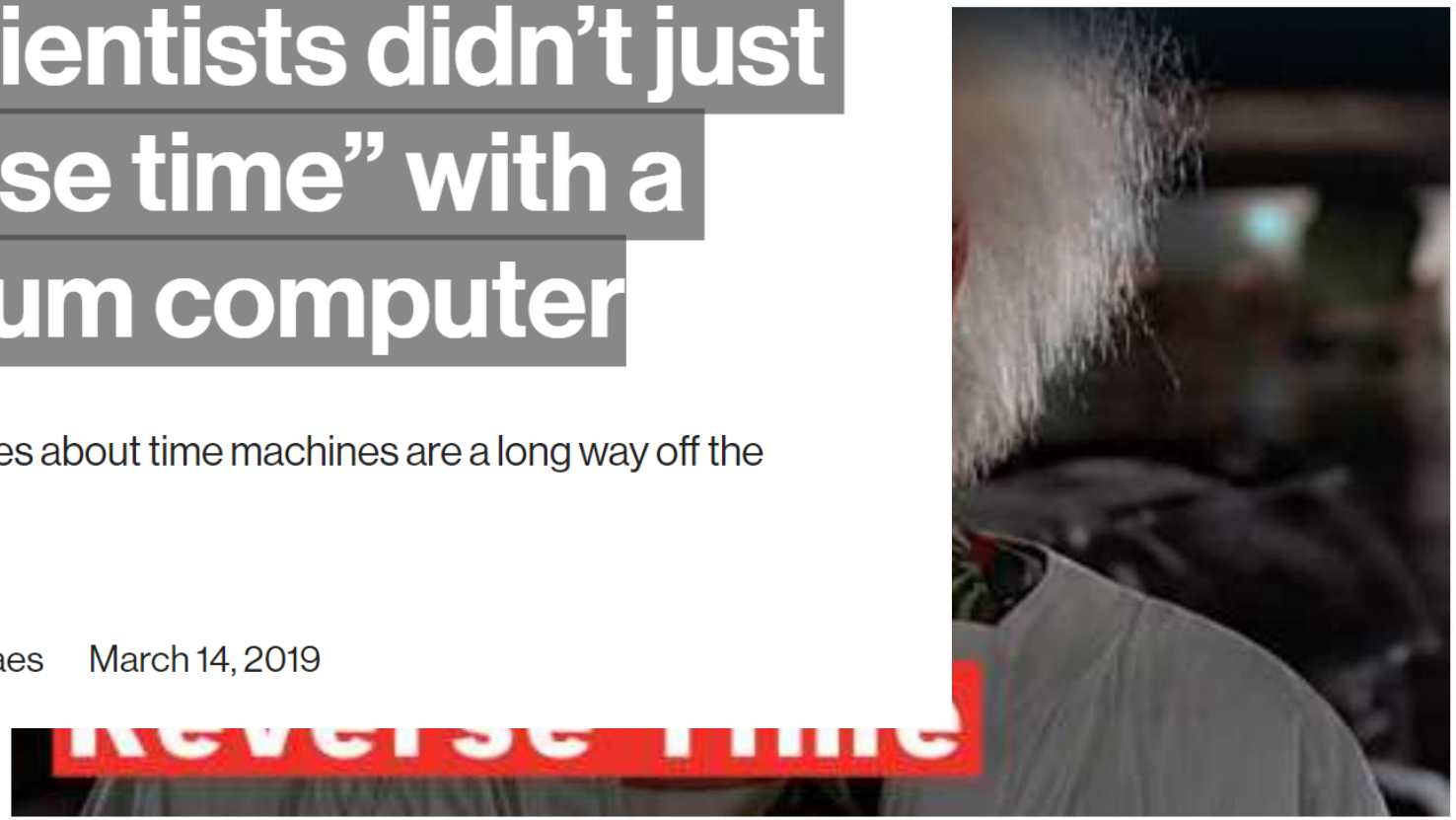
BY AGNESE SMITH December 7, 2018

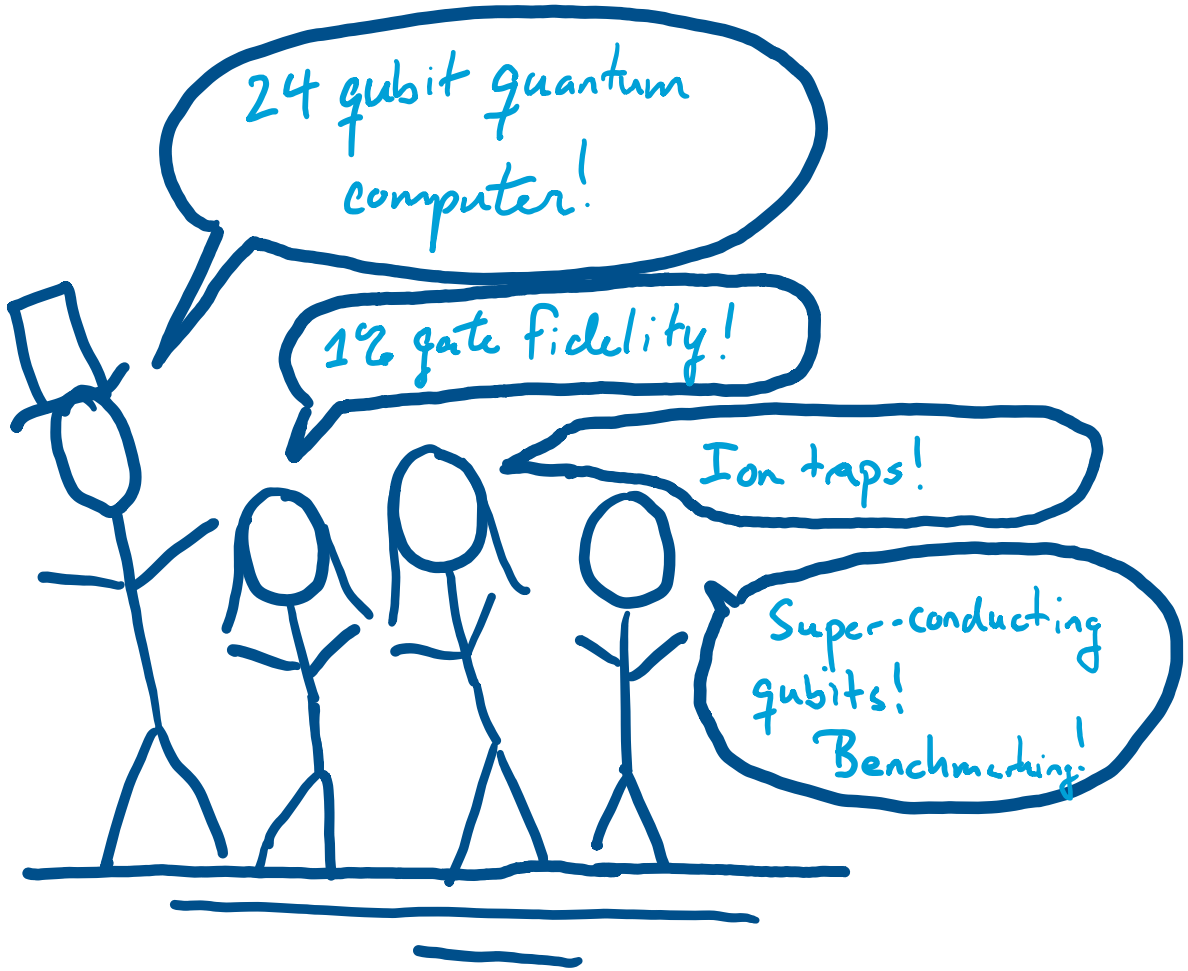
Intelligent Machines

No, scientists didn't just "reverse time" with a quantum computer

Amazing headlines about time machines are a long way off the mark, sadly.

by Konstantin Kakaes March 14, 2019





As experiments progress and hype builds, we need an undisputable demonstration of quantum computers achieving a task intractable for classical computers.

Can we demonstrate this seperation with today's noisy and intermediate-scale devices?

"an experimental violation of the Extended Church-Turing Thesis" - U. Vazirani

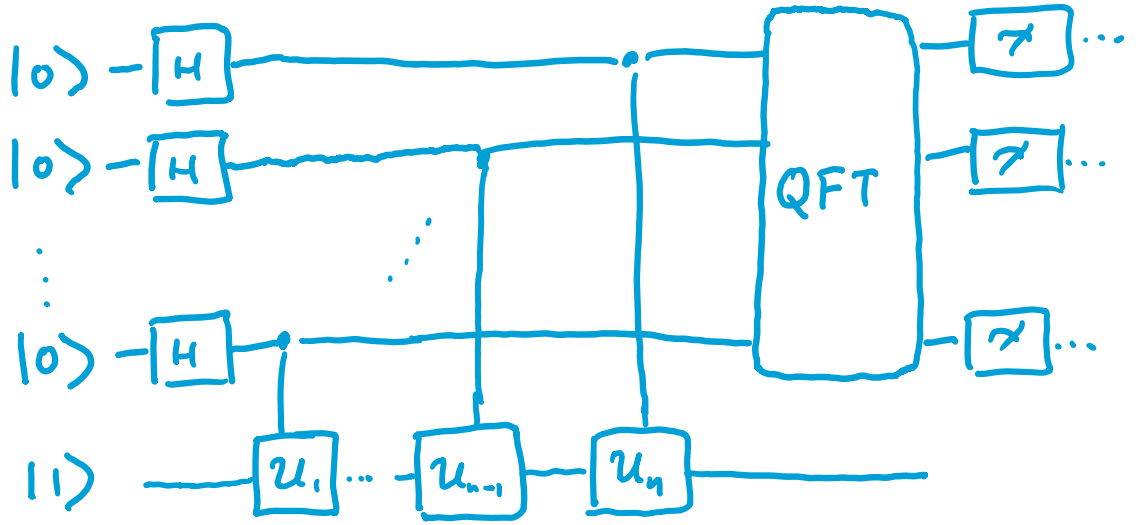
Quantum Supremacy Proposals

A practical demonstration of a quantum computation which is

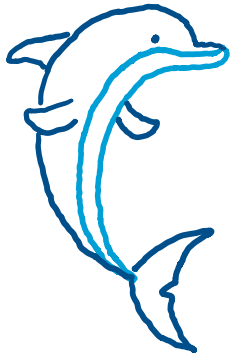
1. Experimentally feasible
2. Has theoretical evidence of classical hardness
3. Verifiable

Why factoring is not the right proposal

Speedups come from carefully engineered interference patterns which are hard to experimentally realize on noisy-intermediate scale devices



Period finding subroutine

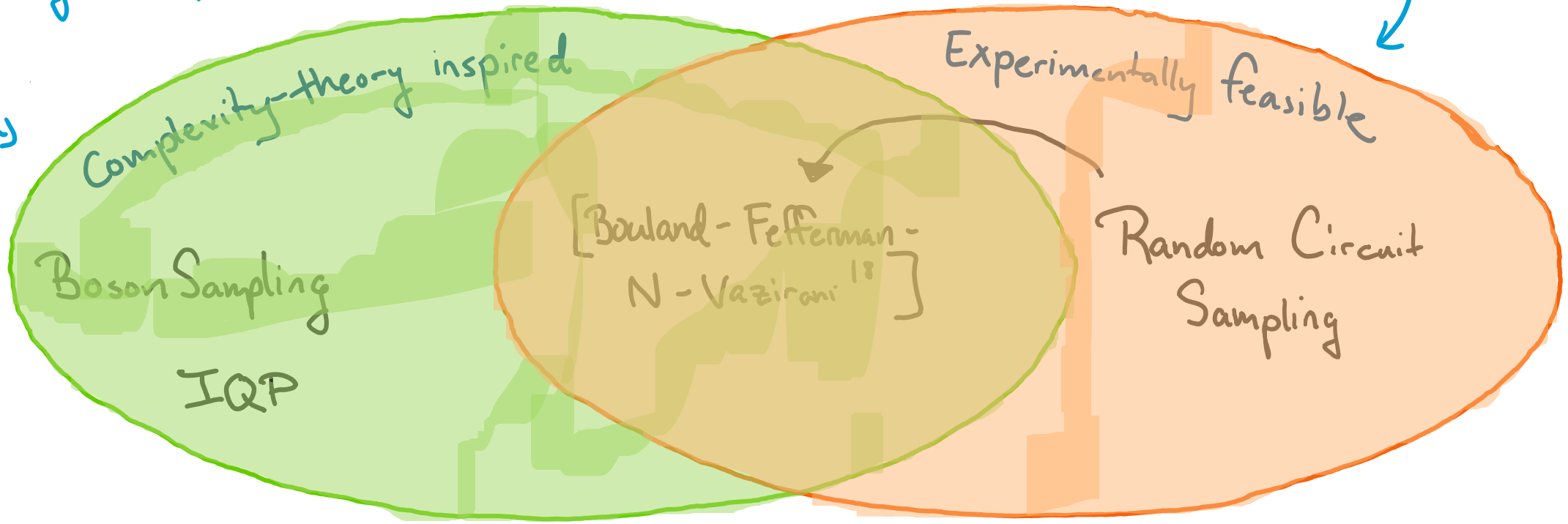


"Proving a quantum computer's computational power by having it factor integers is a bit like proving a dolphin's intelligence by teaching it to solve arithmetic problems." - Aaronson-Arkhipov[13]

Supremacy proposals

Problems for which no efficient classical algorithms exist (under complexity-theory conjectures)

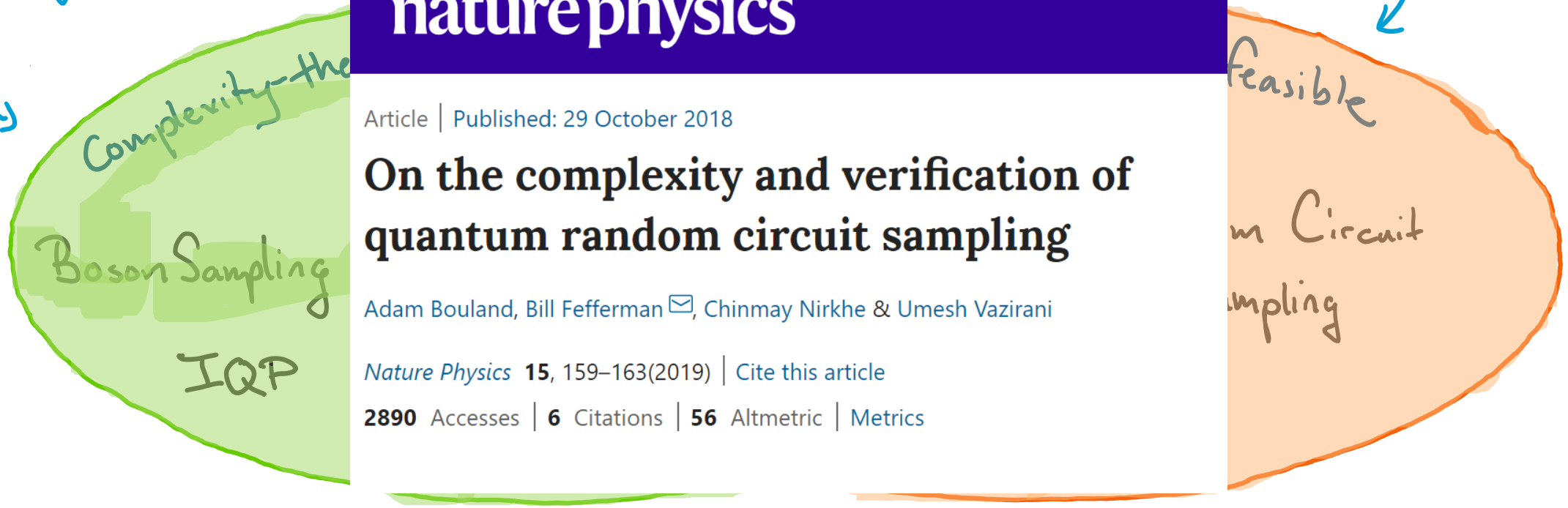
Problems which we can experimentally test in the next ~5 years



Supremacy proposals

Problems for which no efficient classical algorithms exist (under complexity-theory conjectures)

Problems which we can experimentally test in the next ~5 years



nature physics

Article | Published: 29 October 2018

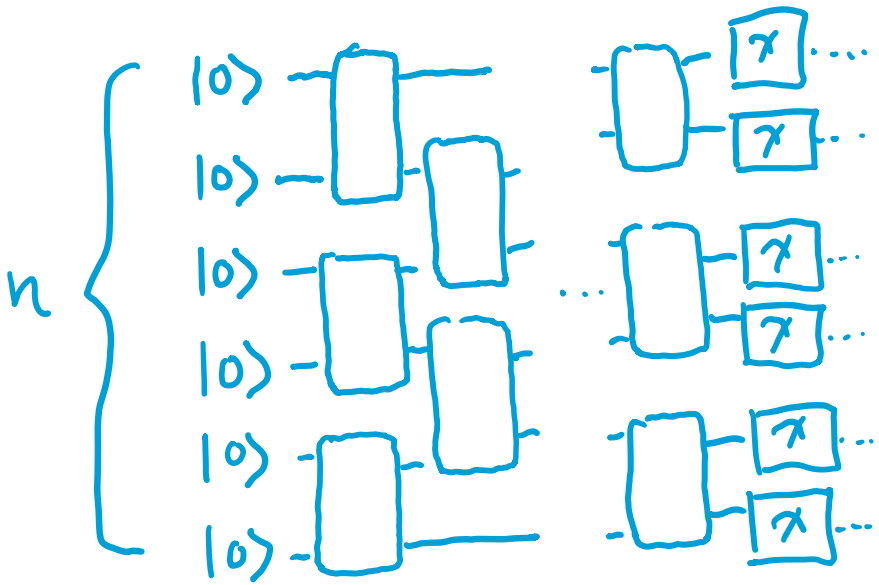
On the complexity and verification of quantum random circuit sampling

Adam Bouland, Bill Fefferman ✉, Chinmay Nirkhe & Umesh Vazirani

Nature Physics **15**, 159–163(2019) | Cite this article

Random Circuit Sampling

"canonical quantum problem"



Every quantum circuit has a classical probability distribution associated with it on $\{0,1\}^n$:

$$P_C(x) = |\langle x|C|0^n\rangle|^2$$

Sampling from this distribution, is an easy task for an ideal quantum computer

Claim: If the gates are choosen Haar-randomly, then it is intractable for a classical device to output samples from P_C.

Random Circuit Sampling

Article

Quantum supremacy using a programmable superconducting processor

https://doi.org/10.1038/s41586-019-1666-5
Received: 22 July 2019
Accepted: 20 September 2019
Published online: 23 October 2019

Frank Arute[1], Kunal Arya[1], Ryan Babbush[1], Dave Bacon[1], Joseph C. Bardin[1,2], Rami Barends[1], Rupak Biswas[3], Sergio Boixo[1], Fernando G. S. L. Brandao[1,4], David A. Buell[1], Brian Burkett[1], Yu Chen[1], Zijun Chen[1], Ben Chiaro[5], Roberto Collins[1], William Courtney[1], Andrew Dunsworth[1], Edward Farhi[1], Brooks Foxen[1,5], Austin Fowler[1], Craig Gidney[1], Marissa Giustina[1], Rob Graff[1], Keith Guerin[1], Steve Habegger[1], Matthew P. Harrigan[1], Michael J. Hartmann[1,6], Alan Ho[1], Markus Hoffmann[1], Trent Huang[1], Travis S. Humble[7], Sergei V. Isakov[1], Evan Jeffrey[1], Zhang Jiang[1], Dvir Kafri[1], Kostyantyn Kechedzhi[1], Julian Kelly[1], Paul V. Klimov[1], Sergey Knysh[1], Alexander Korotkov[1,8], Fedor Kostritsa[1], David Landhuis[1], Mike Lindmark[1], Erik Lucero[1], Dmitry Lyakh[9], Salvatore Mandrà[3,10], Jarrod R. McClean[1], Matthew McEwen[5], Anthony Megrant[1], Xiao Mi[1], Kristel Michielsen[11,12], Masoud Mohseni[1], Josh Mutus[1], Ofer Naaman[1], Matthew Neeley[1], Charles Neill[1], Murphy Yuezhen Niu[1], Eric Ostby[1], Andre Petukhov[1], John C. Platt[1], Chris Quintana[1], Eleanor G. Rieffel[3], Pedram Roushan[1], Nicholas C. Rubin[1], Daniel Sank[1], Kevin J. Satzinger[1], Vadim Smelyanskiy[1], Kevin J. Sung[1,13], Matthew D. Trevithick[1], Amit Vainsencher[1], Benjamin Villalonga[1,14], Theodore White[1], Z. Jamie Yao[1], Ping Yeh[1], Adam Zalcman[1], Hartmut Neven[1] & John M. Martinis[1,5]*

The promise of quantum computers is that certain computational tasks might be executed exponentially faster on a quantum processor than on a classical processor[1]. A fundamental challenge is to build a high-fidelity processor capable of running quantum algorithms in an exponentially large computational space. Here we report the use of a processor with programmable superconducting qubits[2–7] to create quantum states on 53 qubits, corresponding to a computational state-space of dimension 2^{53} (about 10^{16}). Measurements from repeated experiments sample the resulting probability distribution, which we verify using classical simulations. Our Sycamore processor takes about 200 seconds to sample one instance of a quantum circuit a million times—our benchmarks currently indicate that the equivalent task for a state-of-the-art classical supercomputer would take approximately 10,000 years. This dramatic increase in speed compared to all known classical algorithms is an experimental realization of quantum supremacy[8–14] for this specific computational task, heralding a much-anticipated computing paradigm.

Google's recent Nature article on supremacy

Every quantum circuit has a classical probability distribution associated with it on $\{0,1\}^n$:

$$P_C(x) = |\langle x|C|0^n\rangle|^2$$

Sampling from this distribution, is an easy task for an ideal quantum computer

Claim: If the gates are choosen Haar-randomly, then it is intractable for a classical device to output samples from P_C.

Quantum Supremacy Proposals

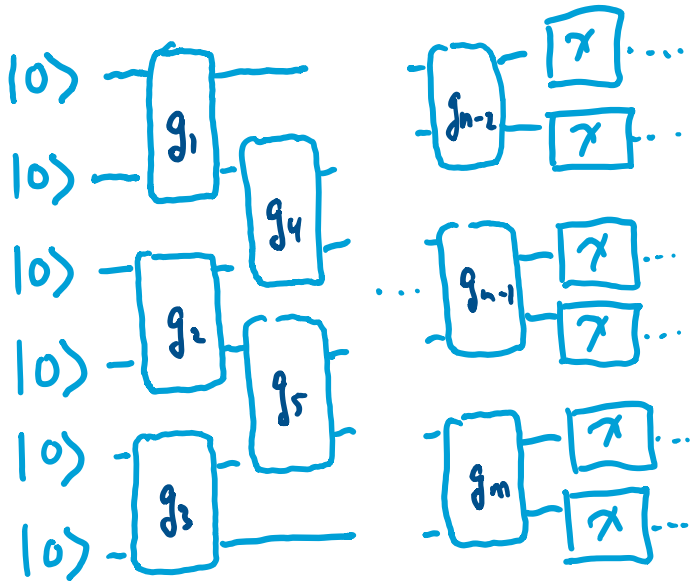
A practical demonstration of a quantum computation which is

1. Experimentally feasible ✓
2. Has theoretical evidence of classical hardness
3. Verifiable

] Bouland-Fefferman-N-Vazirani Nature Physics 2018.

Random Circuit Sampling

Fix an architecture over quantum circuits.



Choose gates $g_1, \ldots, g_m \sim$ Haar.

Task:

Output, whp over choice of gates, samples from the canonical prob. distribution of the circuit.

Establishing classical hardness

Goal: Show that sampling from the output distribution is #P-hard.

#P = { counting problems }.

examples:

# of solutions to a SAT problem

# of Hamiltonian cycles in a graph

# of 3-colourings of a graph

Idea: Show that if you had a sampler for the distribution, then you could calculate the probability $p_C(x)$ approximately.

Second, show that approximating $p_C(x)$ is #P-hard.

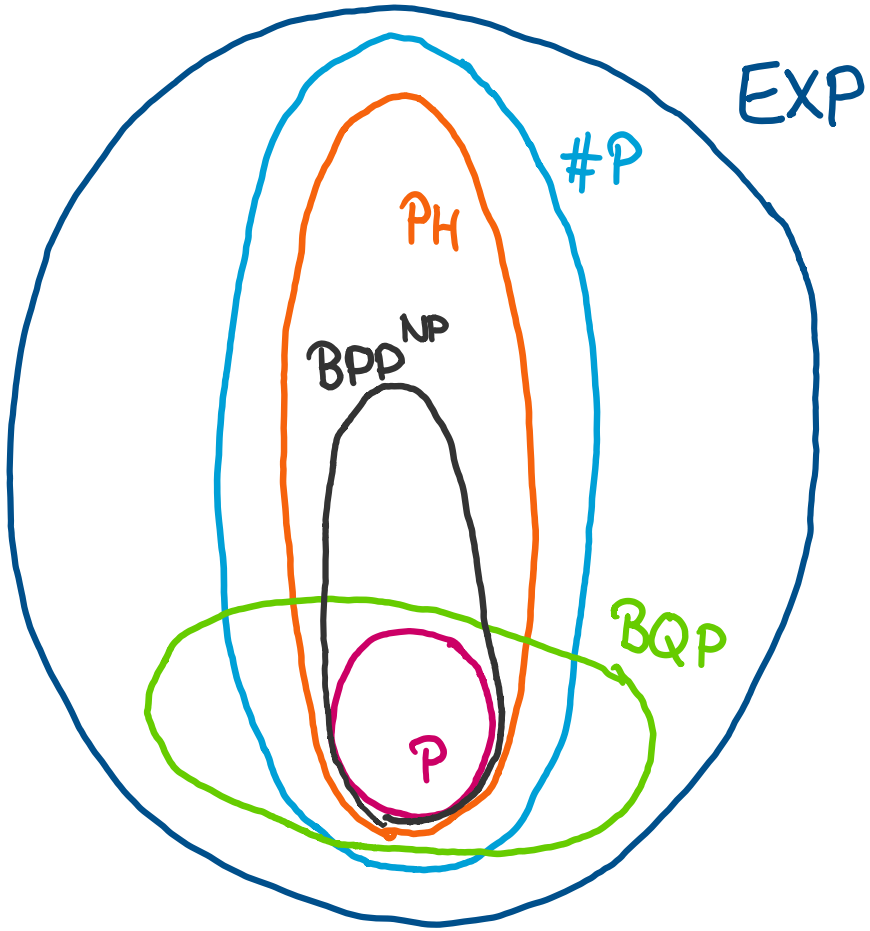
Establishing classical hardness

Idea: Show that if you had a sampler for the distribution, then you could calculate the probability $p_C(x)$ approximately.

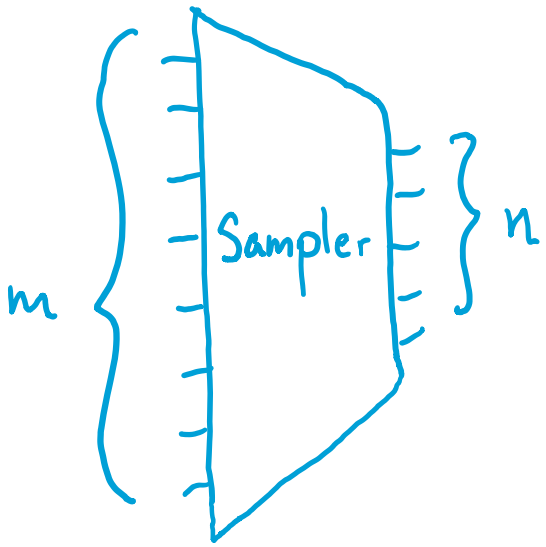
We will establish a $\mathrm{BPP}^{\mathrm{NP}} \subseteq \mathrm{PH} \subseteq \#\mathrm{P}$

- reduction to show this statement.

A known result by Stockmeyer '85.



Stockmeyer's Theorem [85]



$$\Pr(S \text{ outputs } x \in \{0,1\}^n) = \frac{\#\{y : S(y) = x\}}{2^m}$$

Let h be a random fn $\{0,1\}^m \to \{0,1\}^r$.

If $\#\{y : S(y) = x\} \geq 10 \cdot 2^r$, then w.h.p.

$\exists\, y$ s.t. $S(y) = x$ & $h(y) = 0^r$.

$\therefore$ $\mathrm{BPP}^{\mathrm{NP}^{\mathrm{sampler}}}$ can approximate

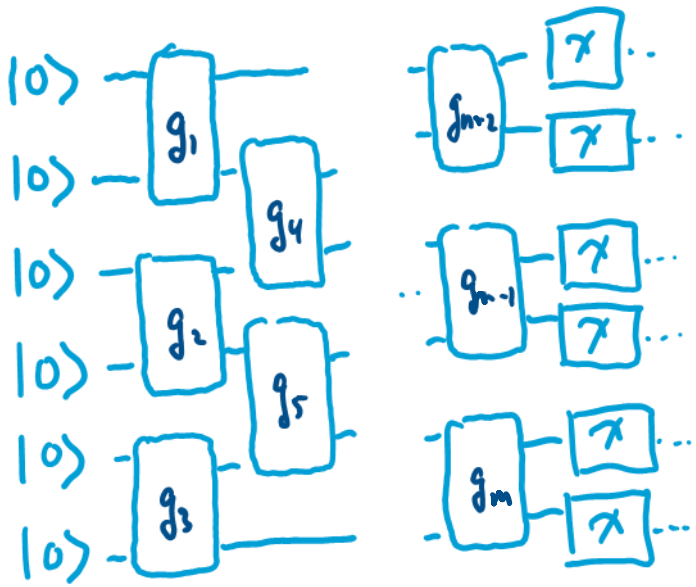
$\Pr(S \text{ outputs } x)$ to mult. 10.

Can be amplified to any ε using standard techniques.

Establishing classical hardness

Second, show that approximating $p_C(x)$ is #P-hard.

"Feynman Path Integral"



$$p_C(x) = |\langle x|C|0\rangle|^2 = |\langle x|g_m g_{m-1} \cdots g_1|0\rangle|^2$$

$$= \left| \sum_{y_1,\ldots,y_m \in \{0,1\}^n} \langle x|g_m|y_m\rangle\langle y_m|g_{m-1}|y_{m-1}\rangle \cdots \langle y_1|g_1|0\rangle \right|^2$$

$$= \left| \sum_{\substack{y_0,\ldots y_{m+1} \in \{0,1\}^n \\ y_0 = 0 \\ y_{m+1} = x}} \prod_{j=1}^{m+1} \langle y_j|g_j|y_{j-1}\rangle \right|^2$$

Establishing classical hardness

Second, show that approximating $p_c(x)$ is #P-hard.

$$p_c(x) = \left| \sum_{y_0, \ldots y_{m+1} \in \{0,1\}^* \; y_0 = 0 \; y_{m+1} = x} \prod_{j=1}^{m+1} \langle y_j | g_j | y_{j-1} \rangle \right|^2$$

With a little work, it can be seen as the difference of two #P-hard quantities, or is therefore GapP-hard.

GapP-hard quantities are hard to multiplicatively approximate.

Putting it all together

Assume we can sample from the output distribution of a #P-hard circuit.

Then, using Stockmeyer's theorem, we can solve this #P-hard problem in $\mathrm{BPP}^{\mathrm{NP}}$.

Non-collapse of the Polynomial Heirarchy:

$$\mathrm{BPP}^{\mathrm{NP}} \subseteq \Sigma_3 \subsetneq \mathrm{PH} \subseteq \#\mathrm{P}$$

Contradiction!

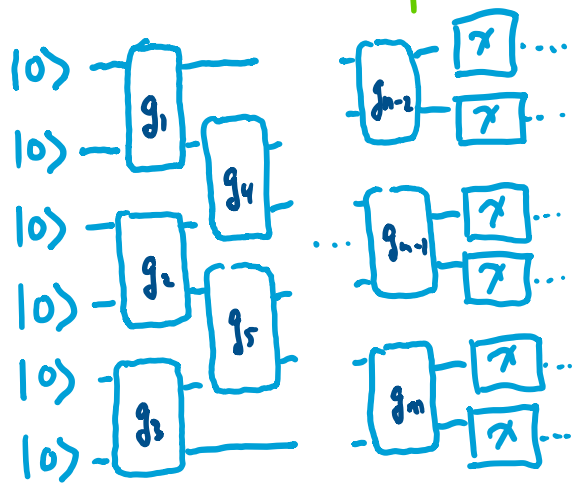
$\therefore$ Exact classical sampling of output distributions is intractable.

Close, but not quite there...

We have shown that exact sampling is #P-hard.

But exact sampling isn't feasible for near-term quantum devices.

Fix an architecture over quantum circuits.



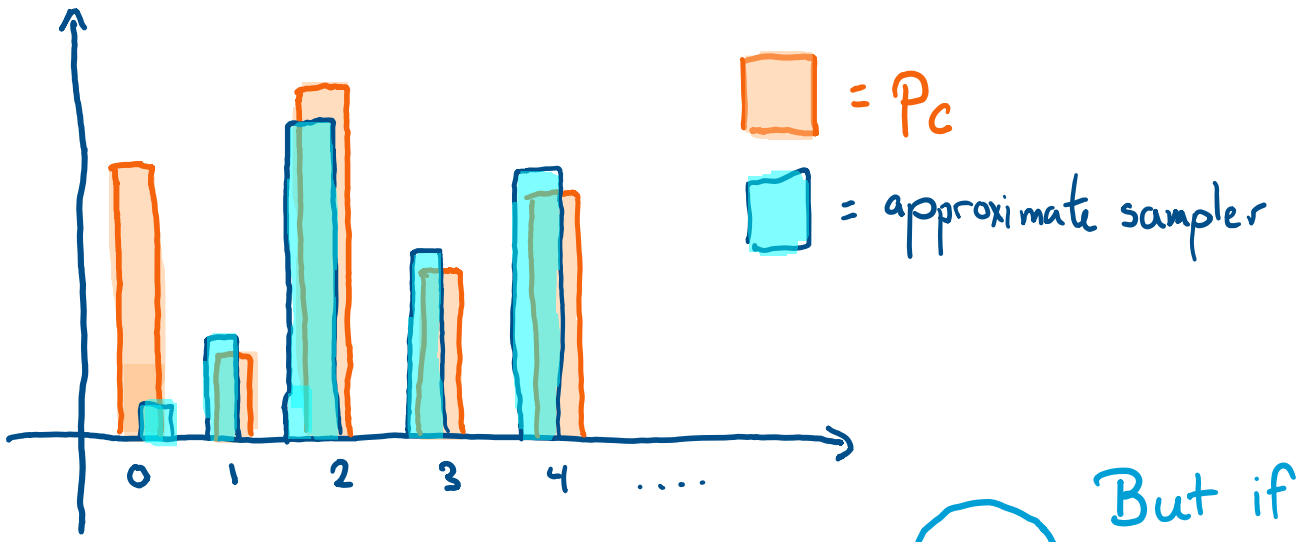
Choose gates $g_1, \ldots, g_m \sim$ Haar.

Task:

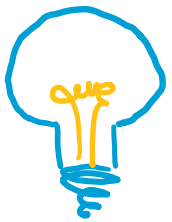
Output, whp over choice of gates, samples from a distribution near the canonical distribution of the circuit.

Showing that approximate sampling is also hard...

Let's assume that $p_c(0)$ is the GapP-hard quantity.



Even if $p_c(0)$ is hard to approximate, an ε-approximate sampler q may have $q(0)$ far from $p_c(0)$, so q may not be hard!



But if for most x, $p_c(x)$ is hard to approximate, then an approximate sampler will still be hard!

Equivalently, we need to show that the quantity $P_C(x)$ is average-case hard ~~to approximate~~. to calculate exactly.

Currently, we don't know how to prove such a statement for any supremacy proposal including Boson Sampling or IQP.

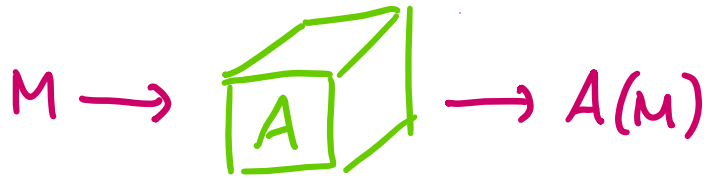
Due to a property called hiding, we need to show a statement like:

Calculating $p_c(0)$ for > 0.76 fraction of circuits w.r.t. the Haar-distribution is #P-hard.

average-to-worst-case reduction

What known problems have avg-to-worst case reductions?

$$\mathrm{perm}(M) = \sum_{\sigma \in S_n} \prod_{j=1}^{n} M_{j,\sigma(j)}$$

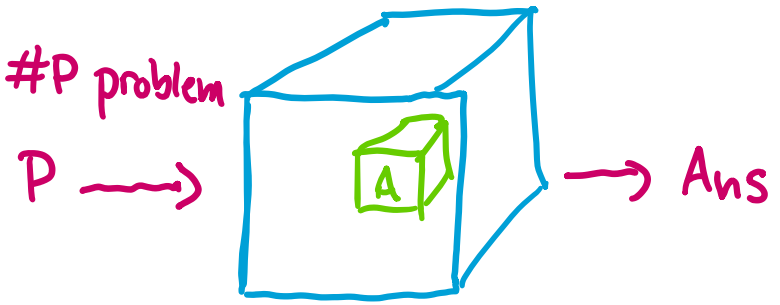


If $\Pr_M\left(A(M) = \mathrm{perm}(M)\right) > 0.76$

then, $\exists$

Theorem (Lipton[91], GLR+[91])

The following is #P-hard: For sufficiently large prime power q, given uniformly random matrix $M \in \mathbb{F}_q^{n \times n}$, calculating $\mathrm{perm}(M)$ with prob. > 0.76.



in particular, can solve permanents on worst-case inputs.

Proof that permanent is avg-case hard

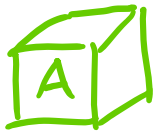
$$\text{perm}(M) = \sum_{\sigma \in S_n} \prod_{j=1}^{n} M_{j,\sigma(j)}$$

perm(M) = deg n polynomial in the entries of M.

Choose R uniformly random $\in \mathbb{F}_q^{n \times n}$. Set $M(t) = M + Rt$.

$M(0) = M$, $M(t)$ uniformly random $t \neq 0$.

perm(M(t)) is a degree n poly (univariate).

Choose random $t_1 \ldots t_{10n}$ and calculate $\text{perm}(M(t_i))$ using 

.

Interpolate using Berlekamp-Welch to learn perm(M(t)) despite errors.

Input $t = 0$ for answer.

Goal: Find a similar polynomial structure in the problem of Random Circuit Sampling

$$\mathrm{perm}(M) = \sum_{\sigma \in S_n} \prod_{j=1}^{n} M_{j,\sigma(j)}$$

$$p_C(x) = \left| \sum_{\substack{y_0, \ldots y_{m+1} \in \{0,1\}^* \\ y_0 = 0 \\ y_{m+1} = x}} \prod_{j=1}^{m+1} \langle y_j | g_j | y_{j-1} \rangle \right|^2$$

Feynman Path Integral

$$P_C(x) = \left| \sum_{y_0, \ldots y_{m+1} \in \{0,1\}^n} \prod_{j=1}^{m+1} \langle y_j | g_j | y_{j-1} \rangle \right|^2$$

$y_0 = 0$

$y_{m+1} = x$

Feynman Path Integral

$P_C(x)$ is a low-deg polynomial in the entries of $g_1, \ldots, g_m$. We can apply a similar interpolation technique to demonstrate that Random Circuit Sampling is worst-to-average case hard.

Quantum Supremacy Proposals

A practical demonstration of a quantum computation which is

1. Experimentally feasible ✓
2. Has theoretical evidence of classical hardness
3. Verifiable

] Bouland-Fefferman-N-Vazirani Nature Physics 2018.

Google's quantum supremacy experiment

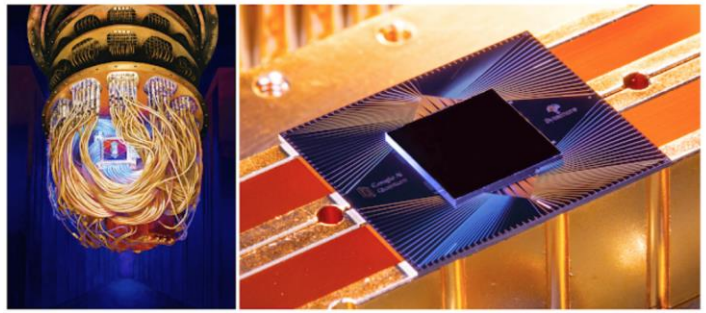
Quantum Supremacy Using a Programmable Superconducting Processor

Wednesday, October 23, 2019

Posted by John Martinis, Chief Scientist Quantum Hardware and Sergio Boixo, Chief Scientist Quantum Computing Theory, Google AI Quantum

Physicists have been talking about the power of quantum computing for over 30 years, but the questions have always been: will it ever do something useful and is it worth investing in? For such large-scale endeavors it is good engineering practice to formulate decisive short-term goals that demonstrate whether the designs are going in the right direction. So, we devised an experiment as an important milestone to help answer these questions. This experiment, referred to as a quantum supremacy experiment, provided direction for our team to overcome the many technical challenges inherent in quantum systems engineering to make a computer that is both programmable and powerful. To test the total system performance we selected a sensitive computational benchmark that fails if just a single component of the computer is not good enough.

Today we published the results of this quantum supremacy experiment in the *Nature* article, "Quantum Supremacy Using a Programmable Superconducting Processor". We developed a new 54-qubit processor, named "Sycamore", that is comprised of fast, high-fidelity quantum logic gates, in order to perform the benchmark testing. Our machine performed the target computation in 200 seconds, and from measurements in our experiment we determined that it would take the world's fastest supercomputer 10,000 years to produce a similar output.



Left: Artist's rendition of the Sycamore processor mounted in the cryostat. (Full Res Version; Forest Stearns, Google AI Quantum Artist in Residence) **Right:** Photograph of the Sycamore processor. (Full Res Version; Erik Lucero, Research Scientist and Lead Production Quantum Hardware)

QUANTIZED COLUMNS

Why I Called It 'Quantum Supremacy'

22

Researchers finally seem to have a quantum computer that can outperform a classical computer. But what does that really mean?

Why scientists are so excited about "quantum supremacy"

With a quantum computer, scientists are dipping into deeply weird physics to solve problems.

By Brian Resnick | @B_resnick | brian@vox.com | Oct 24, 2019, 3:30pm EDT

Opinion

Why Google's Quantum Supremacy Milestone Matters

The company says its quantum computer can complete a calculation much faster than a supercomputer. What does that mean?

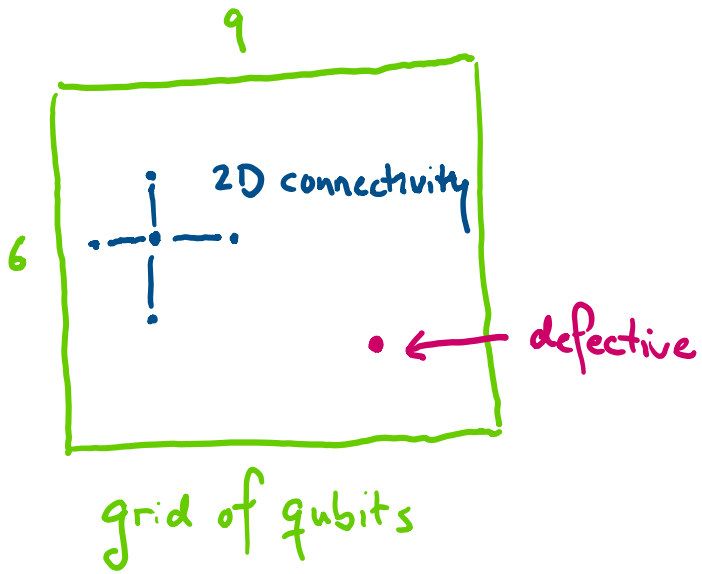
By Scott Aaronson

Dr. Aaronson is the founding director of the Quantum Information Center at the University of Texas at Austin.

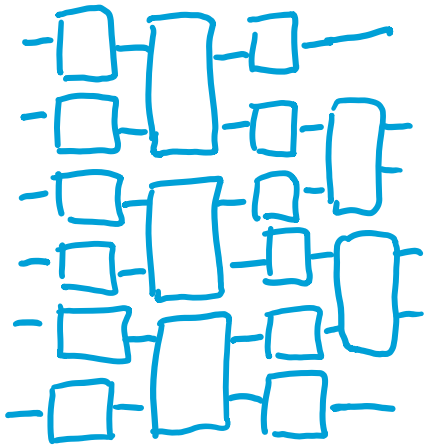
Oct. 30, 2019

Google's quantum supremacy experiment

Ran a similar experiment to Random Circuit Sampling



depth = 20 circuit



alternating 1 & 2 qubit gates.

... 1 qubit gates $\sim_u \{\sqrt{X}, \sqrt{Y}, \sqrt{W}\}$

2 qubit gates = $\sqrt{i\,\mathrm{SWAP}}$

Goal: Produce samples from a distribution close to the output dist.

Google's quantum supremacy experiment

Goal: Produce samples from a distribution close to the output dist.

How can we tell if the device is sampling from this distribution?

Issues

1. What is the target distribution?
2. How many samples do we need to take to verify closeness (as a fn of n)?

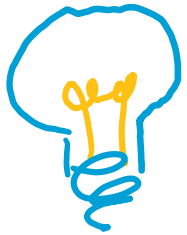
Measuring "quantumness" of the device

Utilize the supremacy "sweet spot": $KL(P_c \| P_{dev}) = \sum_x P_{dev}(x) \log\left(\frac{P_{dev}(x)}{P_c(x)}\right)$

① Computing $P_c(x)$ is $\sim 2^{40}$ computation for specific x.

② Computing all of P_c is too expensive.

Takes $\sim 2^n$ samples to approximate KL.



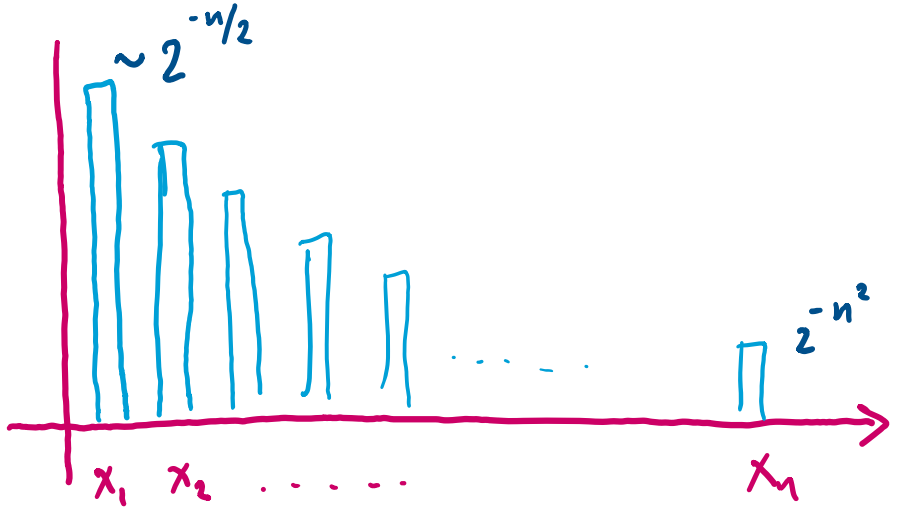
Approximate $\sum_x P_{dev}(x) P_c(x) \stackrel{def}{=} XEB(P_c \| P_{dev})$.

The intuition behind cross-entropy benchmarking

Random circuits behave like t-designs (emulate random unitaries).

The prob. dist. p_C is exponentially distributed.

$$p_C(x_1) \geq p_C(x_2) \geq \ldots\ldots \geq p_C(x_n)$$



$$KL = \mathop{\mathbb{E}}_{p_{dev}} \log\left(\frac{p_{dev}(x)}{p_C(x)}\right)$$

$$\sim -n + \mathop{\mathbb{E}}_{p_{dev}} \log\left(\frac{1}{p_C(x)}\right).$$

$$XEB = \mathop{\mathbb{E}}_{p_{dev}} p_C(x).$$

Measuring "quantumness" of the device

Approximate $\sum_x p_{dev}(x)\, p_c(x) \stackrel{\text{def}}{=} \mathrm{XEB}(p_c \| p_{dev})$.

If $p_{Dev} = p_c$, then $2^n \mathrm{XEB} - 1 = 1$ whp.

If $p_{Dev} = \text{uniform}$, then $2^n \mathrm{XEB} - 1 = 0$.

Google's supremacy claim:

Achieving $2^n \mathrm{XEB} - 1 > 0$ is proof of quantumness.

Their result:

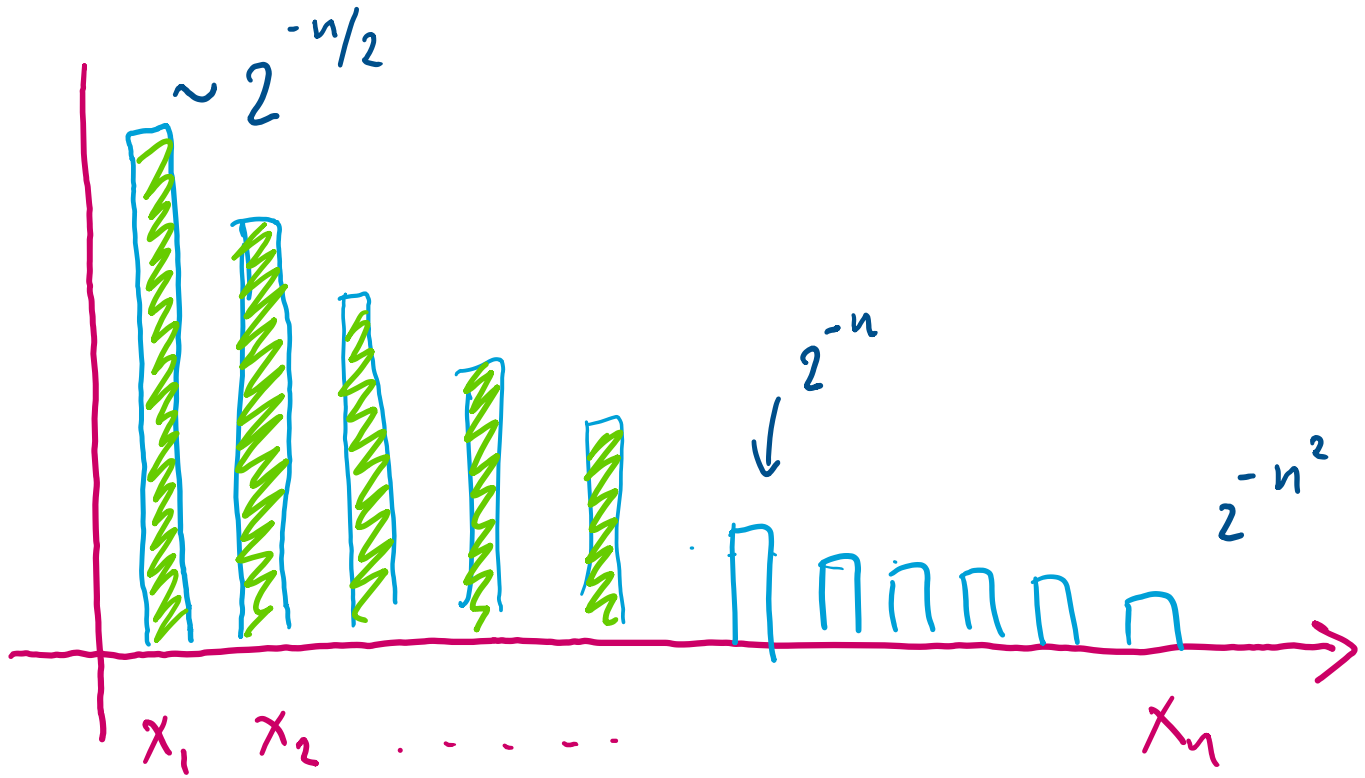
$2^n \mathrm{XEB} - 1 =$

$(2.24 \pm 0.21) \cdot 10^{-3}$

with 5σ confidence

$> 1 \cdot 10^{-3}$.

Measuring "quantumness" of the device



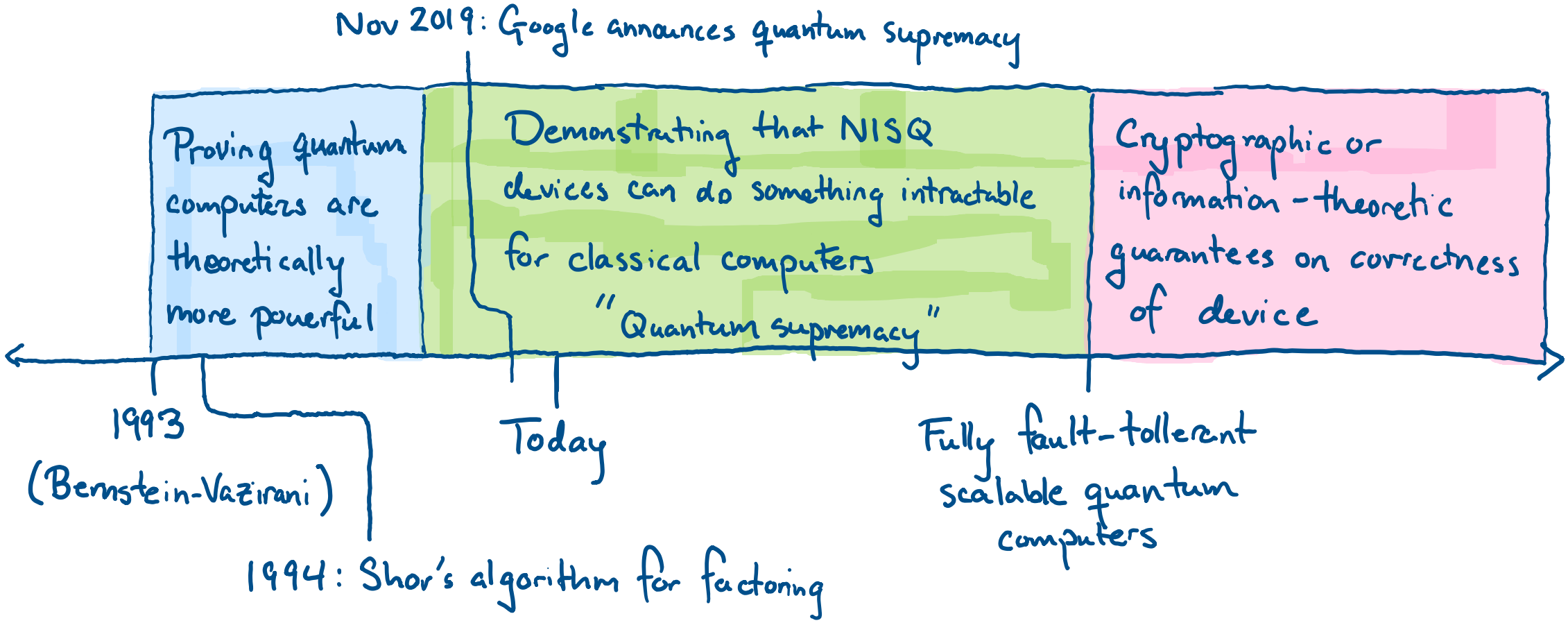
Google claims it can find these "heavy" elements slightly better than uniform guessing

&

Classical computers in limited time cannot do better than uniform guessing.

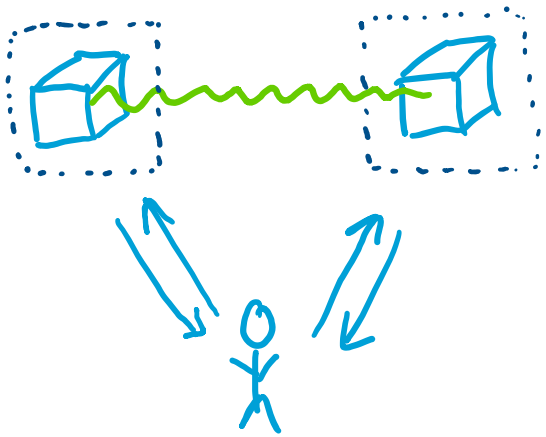
Timeline for this question

How do you tell if the box is an actual quantum computer?



Getting a leash on quantum devices

Use entanglement to your advantage!

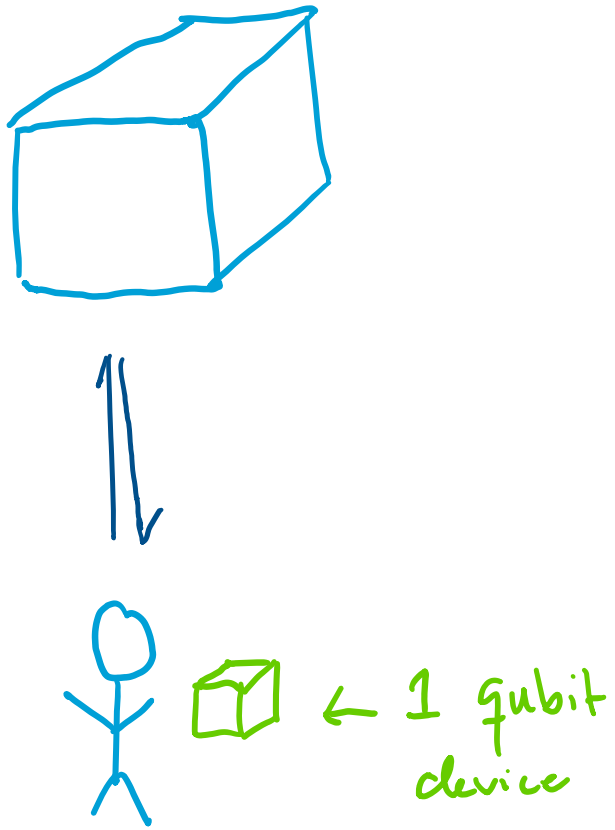


Many known results show how to use entanglement between 2 seperated quantum devices to ensure that they behave correctly.

"Power of quantum correlations"

[Reichardt-Unger-Vazirani[12]], [Natarajan-Vidick[16]] for ex.

Testing quantum devices with small quantum devices



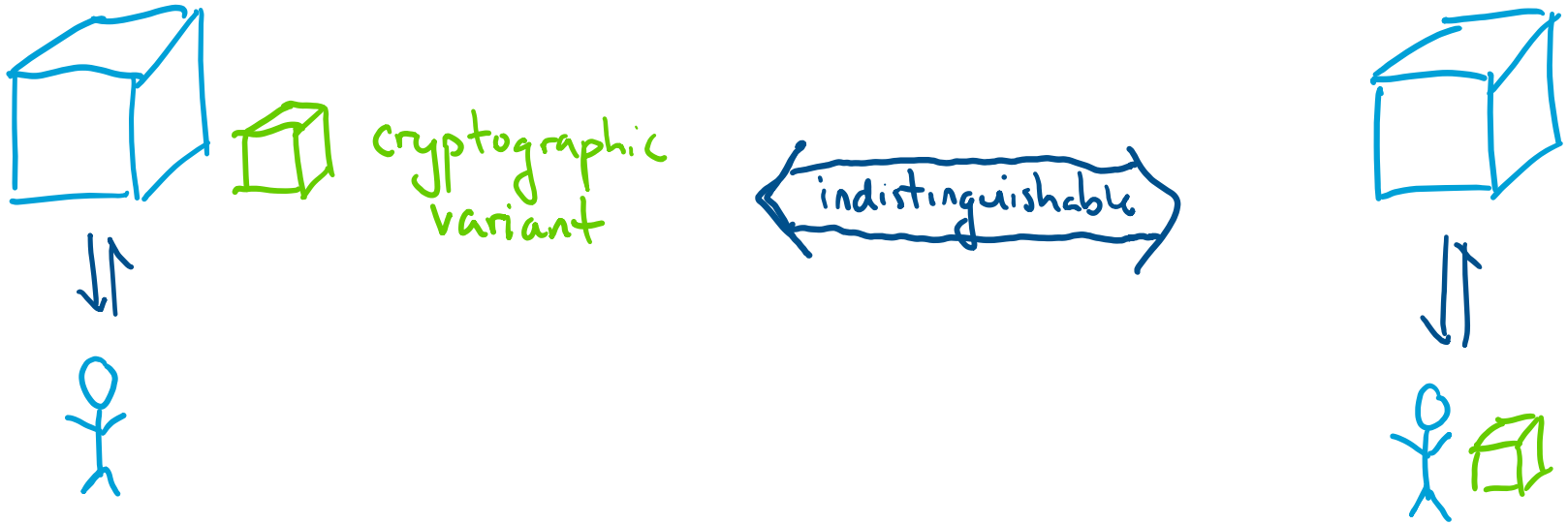
A one qubit device can be used to ensure that the larger quantum device is behaving correctly. Moreover, we can achieve properties like blindness, where the larger device doesn't know what its computing.

[Fitzsimmons-Kashefi'16] for ex.

Testing quantum devices with cryptography

Use post-quantum cryptography to ensure that the device behaves correctly.

Reduce to the small quantum device setting.



[Mahadev '19] for ex.

Timeline for this question

How do you tell if the box is an actual quantum computer?