Complexity Foundations of Quantum Supremacy

Quantum Supremacy

Preskill 2012: arXiv:1203.5813

Classical systems cannot in general simulate quantum systems efficiently.

We cannot yet prove this claim, either mathematically or experimentally, but we have reason to believe it is true...

We therefore hope to hasten the onset of the era of quantum supremacy, when we will be able to perform tasks with controlled quantum systems going beyond what can be achieved with ordinary digital computers.

Circuit Sampling

Imagine that Bob says he has a quantum computer and Alice does not believe him.

They can play the following game:

1) Alice draws an n-qubit quantum circuit B on a piece of paper and says "implement this!".

Perfectly executing $B|0^n\rangle$ and measuring gives a distribution p_B over $\{0,1\}^n$ where

$$p_B(x) = |\langle x|B|0^n\rangle|^2$$

Circuit Sampling

2) Bob implements B on his (noisy) quantum device Q. Measuring $Q|0^n\rangle$ leads to a distribution p_Q over $\{0,1\}^n$.

Bob creates $Q|0^n\rangle$ and measures a bunch of times to get samples $x_1, \ldots, x_k \in \{0,1\}^n$ from p_Q and sends these to Alice.

3) Alice verifies that the samples are close to what is expected from p_B.

Random Circuit Sampling

If Alice is skeptical, she wants to choose a circuit B that is hard to simulate classically.

That way if Bob succeeds Alice is convinced he actually has a quantum computer.

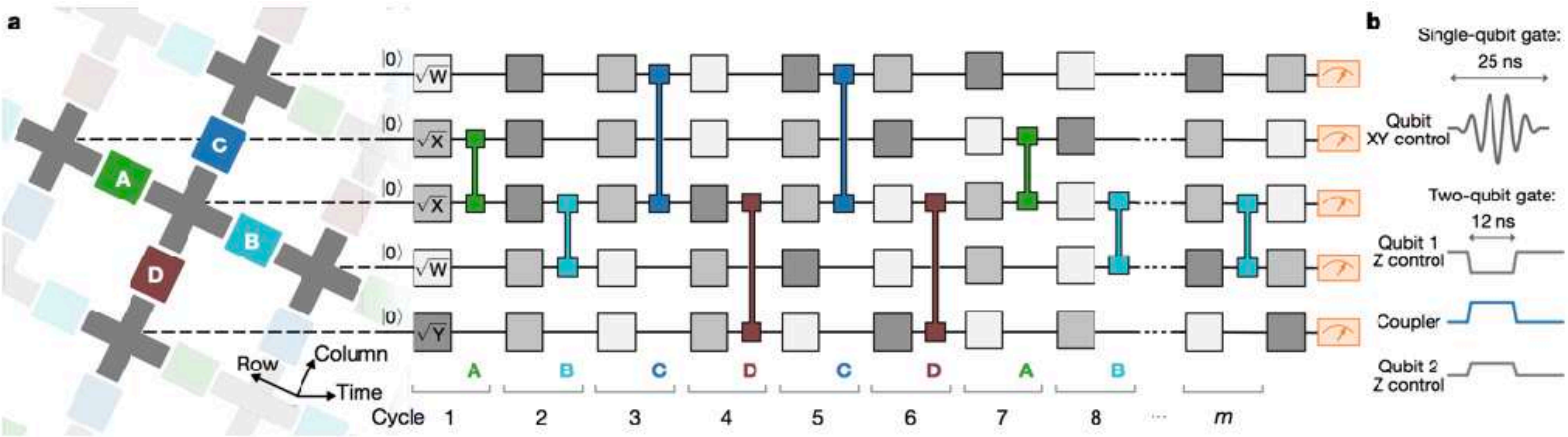
We think that approximating p_B for a random quantum circuit B should be hard classically.

Random circuit sampling is one of the main proposals for quantum supremacy, recently implemented by Google.

Google Experiment

53 qubits in a 2D array.

Nature 574, 505-510 (2019)



Single qubits chosen randomly from $\sqrt{X}, \sqrt{Y}, \sqrt{W}$ where $W = (X+Y)/\sqrt{2}$.

Two qubit gates between pairs are fixed.

20 cycles.

Verification

Google generates $30 \cdot 10^6$ samples from p_Q.

It takes about 200 seconds to generate 10^6 samples.

These samples are sent to Alice to verify that they "look like" samples from p_B.

The quantum supremacy claim is that no classical computer can generate samples to pass this verification test.

Verification

What should the verification test be?

How can Alice efficiently verify that the samples look like they were taken from the distribution p_B?

This is a very interesting question!

In the quantum supremacy regime p_B is unknowable by definition.

Verification

Ideally, one would like to measure how close p_Q is to p_B.

One natural choice is the Kullback-Leibler divergence:

$$D_{KL}(p_B||p_Q) = \sum_{x\in\{0,1\}^n} p_B(x) \ln\left(\frac{p_B(x)}{p_Q(x)}\right)$$

- If $p_B = p_Q$ this is 0.
- It can be unbounded.
- It is not symmetric.

Verification

One natural choice is the Kullback-Leibler divergence:

$$D_{KL}(p_B||p_Q) = \sum_{x\in\{0,1\}^n} p_B(x) \ln\left(\frac{p_B(x)}{p_Q(x)}\right)$$

We can't directly compute this because we don't know p_B or p_Q .

Let's focus on p_B first. We do know properties of p_B that hold whp when B is a random unitary.

Random Unitary

A random unitary is a unitary sampled from the Haar distribution:

1) Let A be an n-by-n matrix where each entry is a complex Gaussian random variable.

2) Let U be the unitary obtained by orthonormalizing the columns of A using Gram-Schmidt.

This process generates a Haar distributed unitary.

Random Unitary

Does the Google circuit generate a random unitary?

The larger the depth the closer the distribution gets to that of a random unitary.

You need depth at least $\Omega(\sqrt{n})$ on a $\sqrt{n}$-by- $\sqrt{n}$ array for each qubit to interact with all the others.

Harrow and Mehraban show that random 2D circuits of depth $O(\sqrt{n})$ are close to random unitaries.

arXiv:1809.06957

Random Unitary

Recall that p_B is the distribution induced by measuring $B|0^n\rangle$ in the comp. basis.

$B|0^n\rangle$ is just the first column of B .

When B is a random unitary $B|0^n\rangle$ is distributed like a vector with ind. Gaussian entries, normalized.

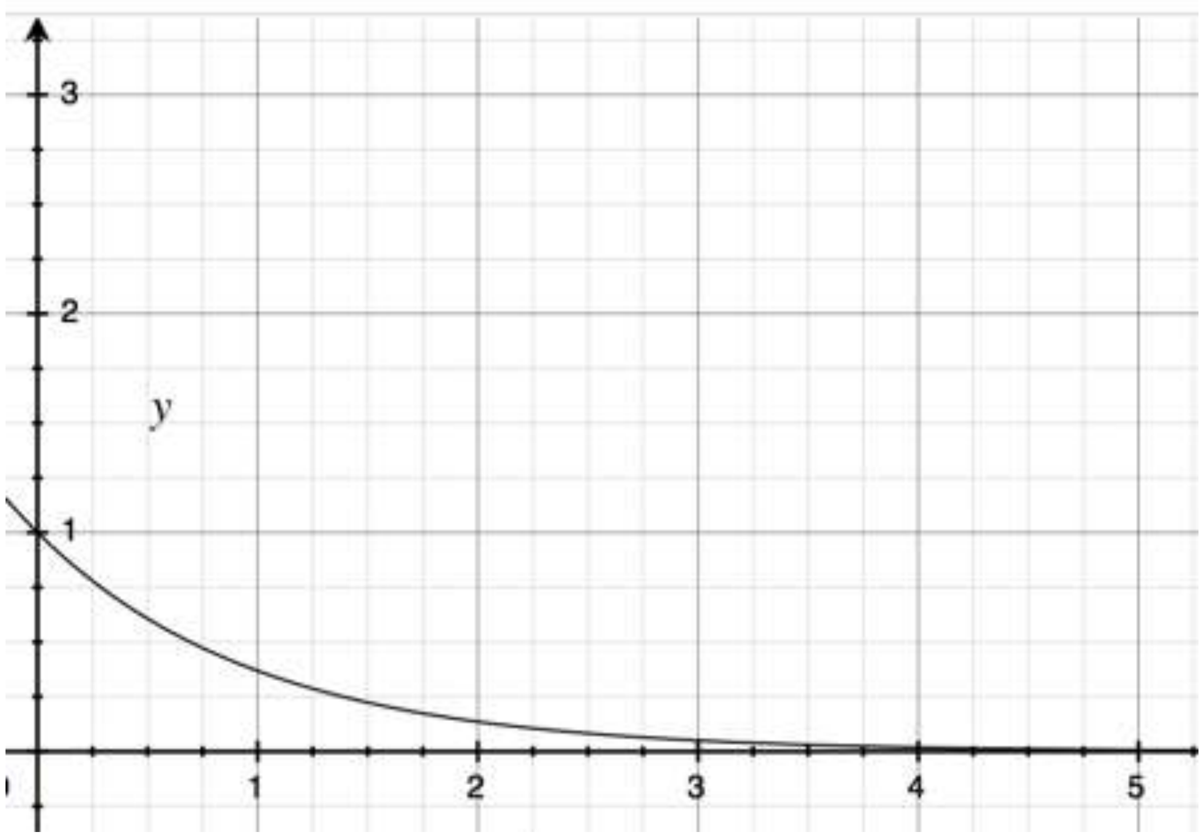
Then $|\langle x|B|0^n\rangle|^2$ is distributed like the square of a Gaussian.

Exponential RV

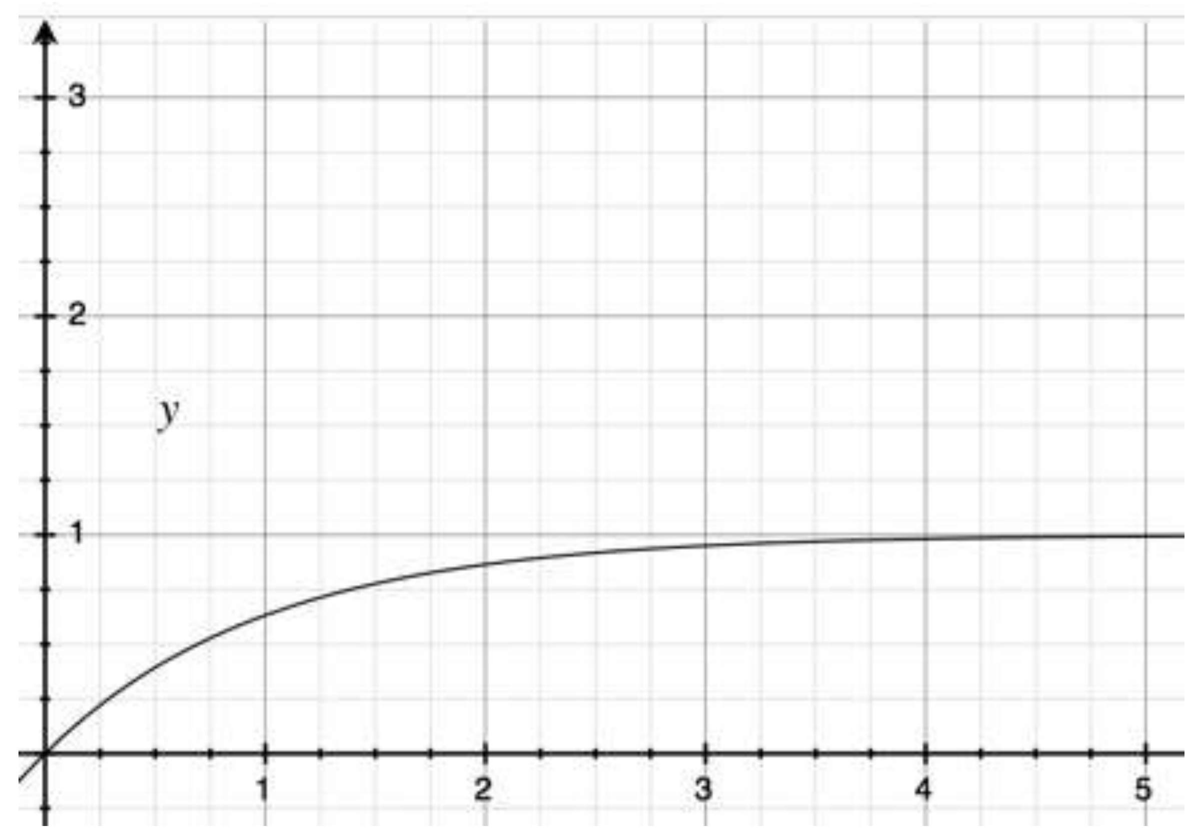
Let $N = 2^n$. Then

$$|\langle x|B|0^n\rangle|^2 \sim \frac{\mathrm{Exp}(1)}{N}$$

$\mathrm{Exp}(1)$ has mean and variance 1 and its pdf is e^{-z}.



pdf: e^{-z}



cdf: $1 - e^{-z}$

Comparing with classical

Google posits that the best thing a classical computer can do is output samples from the uniform dist.

$$D_{KL}(p_B||U) = \sum_{x\in\{0,1\}^n} p_B(x)\ln\left(2^n p_B(x)\right)$$

The expected value of this is known to be $\gamma \approx 0.577$.

Thus if Google could show $D_{KL}(p_B||p_Q) < 0.56$ they would be winning.

Verification

$$D_{KL}(p_B||p_Q) = \sum_{x \in \{0,1\}^n} p_B(x) \ln\left(\frac{p_B(x)}{p_Q(x)}\right)$$

The problem remains that we also don't know p_Q.

We could instead look at

$$D_{KL}(p_Q||p_B) = \sum_{x \in \{0,1\}^n} p_Q(x) \ln\left(\frac{p_Q(x)}{p_B(x)}\right)$$

$$= \mathbb{E}_{x \sim P_Q}\left[\ln\left(\frac{p_Q(x)}{p_B(x)}\right)\right]$$

Linear Cross Entropy

What Google actually uses is the linear cross entropy

$$\mathcal{F}_{\mathrm{XEB}}(q) = 2^n \mathbb{E}_{x \sim q}[p_B(x)] - 1$$

$$= 2^n \sum_{x \in \{0,1\}^n} q(x) p_B(x) - 1$$

A larger value is better.

The intuition is that q should put higher weight on strings that are more likely under p_B .

Linear Cross Entropy

$$\mathcal{F}_{\mathrm{XEB}}(q) = 2^n \sum_{x \in \{0,1\}^n} q(x) p_B(x) - 1$$

If q is uniform this is 0.

If $q = P_B$ then the expected value is 2 (not obvious).

The best thing to do is to set $q(x) = 1$ for the string x with highest probability under p_B .

In expectation this achieves $\Omega(n)$.

Linear Cross Entropy

$$\mathcal{F}_{\mathrm{XEB}}(q) = 2^n \sum_{x \in \{0,1\}^n} q(x) p_B(x) - 1$$

The best thing to do is to set $q(x) = 1$ for the string x with highest probability under p_B .

In expectation this achieves $\Omega(n)$.

Note that with $n = 53$ and taking $30 \cdot 10^6$ samples we don't expect to see the same string twice.

Linear Cross Entropy

$$\mathcal{F}_{\mathrm{XEB}}(q) = 2^n \mathbb{E}_{x \sim q}[p_B(x)] - 1$$

Google argue that for a classical algorithm sampling from p_B is as hard as explicitly approximating p_B .

For p_B they estimate this to take 10,000 years on a classical supercomputer.

From this they argue the best score of a classical algorithm is 0, from sampling uniformly at random.

Linear Cross Entropy

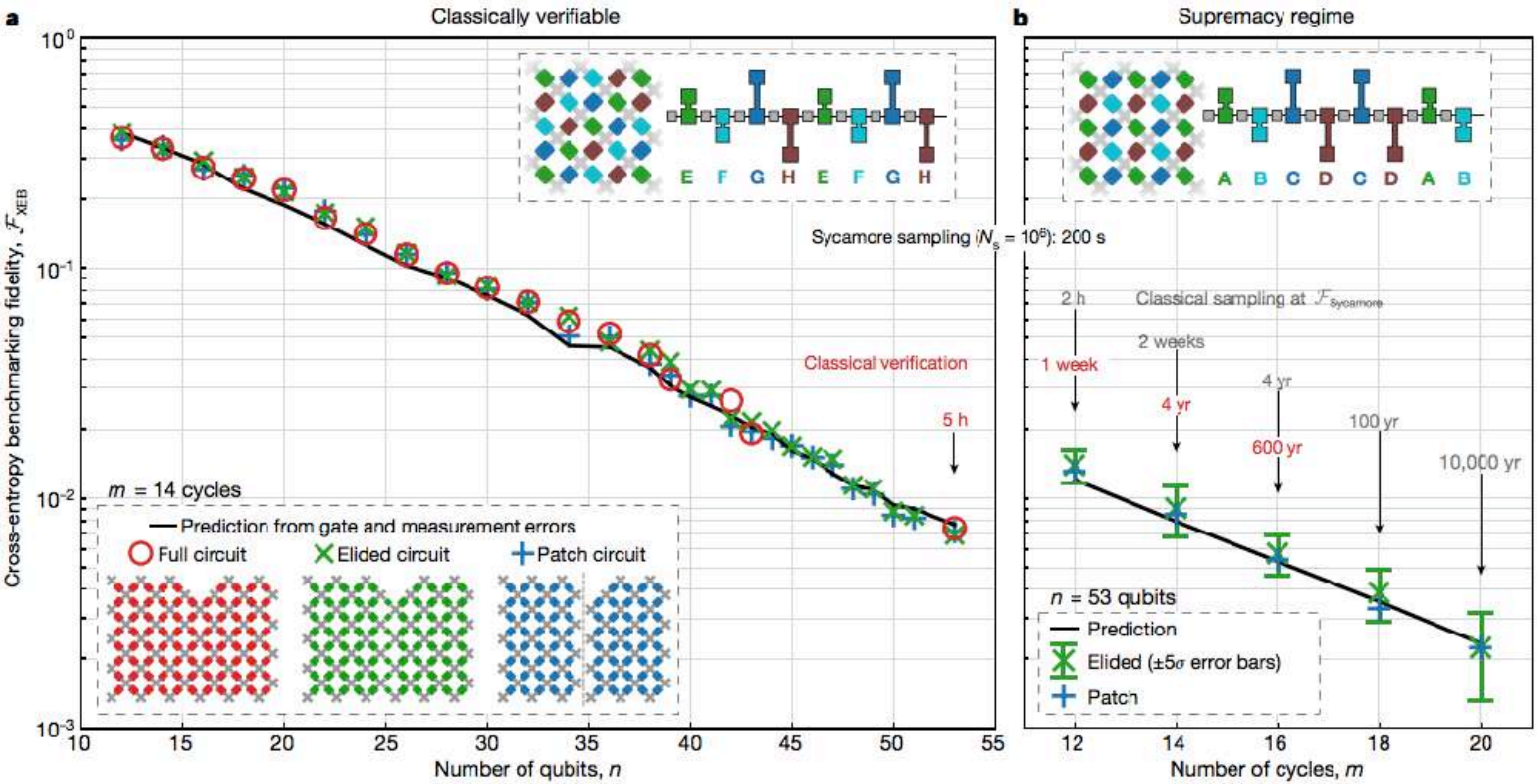
$$\mathcal{F}_{\mathrm{XEB}}(q) = 2^n \mathbb{E}_{x \sim q}[p_B(x)] - 1$$

To compute $\mathcal{F}_{\mathrm{XEB}}(p_Q)$ we still have the problem that we want to do this in a regime where we can't compute p_B .

Google reason about $\mathcal{F}_{\mathrm{XEB}}(p_Q)$ by extrapolating from experiments where p_B can be computed.

If we know p_B we can estimate $\mathcal{F}_{\mathrm{XEB}}(p_Q)$ well empirically by drawing samples from p_Q .

Linear Cross Entropy



From this trend they estimate $\mathcal{F}_{\mathrm{XEB}}(p_Q) = 2.24 \cdot 10^{-3}$.

Further Work

Other companies have pushed back on the 10,000 year claim:

- IBM: 2.5 days arXiv:1910.09534
- Alibaba: 20 days arXiv:2005.06787

Can one spoof linear cross entropy without computing the full probability distribution?

- Quasi-polynomial time randomized algorithm to achieve linear cross entropy $1/\mathrm{poly}(n)$ with a quantum circuit of depth arXiv:2005.02421

Support from complexity

Complexity theory

Now we change gears and talk about theoretical results related to quantum supremacy.

In complexity theory it is very hard to prove unconditional hardness results.

We try to show "amplification of craziness" results.

If X is something you don't think can happen, you try to show that X being true implies something really crazy.

Craziness amplification

The gold standard for something really crazy is $\mathsf{P} = \mathsf{NP}$.

Related to QS, ideally we would like to show:

If there is a randomized polynomial time algorithm that given a random quantum circuit B outputs samples from a distribution P_C with

$$d_{KL}(P_B||P_C) < 0.5$$

then $\mathsf{P} = \mathsf{NP}$.

We currently don't know how to show anything like this.

Complexity Result

What I'll talk about today is a statement with a stronger hypothesis and weaker conclusion.

- Instead of hardness for random quantum circuits we will talk about hardness in the worst case.

- Instead of KL divergence we talk about outputting every $x \in \{0,1\}^n$ with the right probability, up to a multiplicative constant.

- Instead of implying $\mathsf{P} = \mathsf{NP}$ the conclusion is that the polynomial hierarchy collapses to the third level.

Polynomial time

Better known as P .

A language $L \subseteq \{0,1\}^*$ is in P iff there is a deterministic Turing machine M that

- Always terminates in time polynomial in the size of the input.
- $M(x) = 1$ for every $x \in L$.
- $M(x) = 0$ for every $x \notin L$.

Example: Set of graphs that are connected.

Nondeterministic polynomial time

Better known as NP .

A language L is in NP iff there is a deterministic Turing machine M that takes input (x, y) where $|y| \in O(|x|^c)$ and

- $M(x, y)$ always terminates in polynomial time.
- If $x \in L$ then there exists a y s.t. $M(x, y) = 1$.
- If $x \notin L$ then $M(x, y) = 0$ for all y .

Example: Set of formulas that are satisfiable.

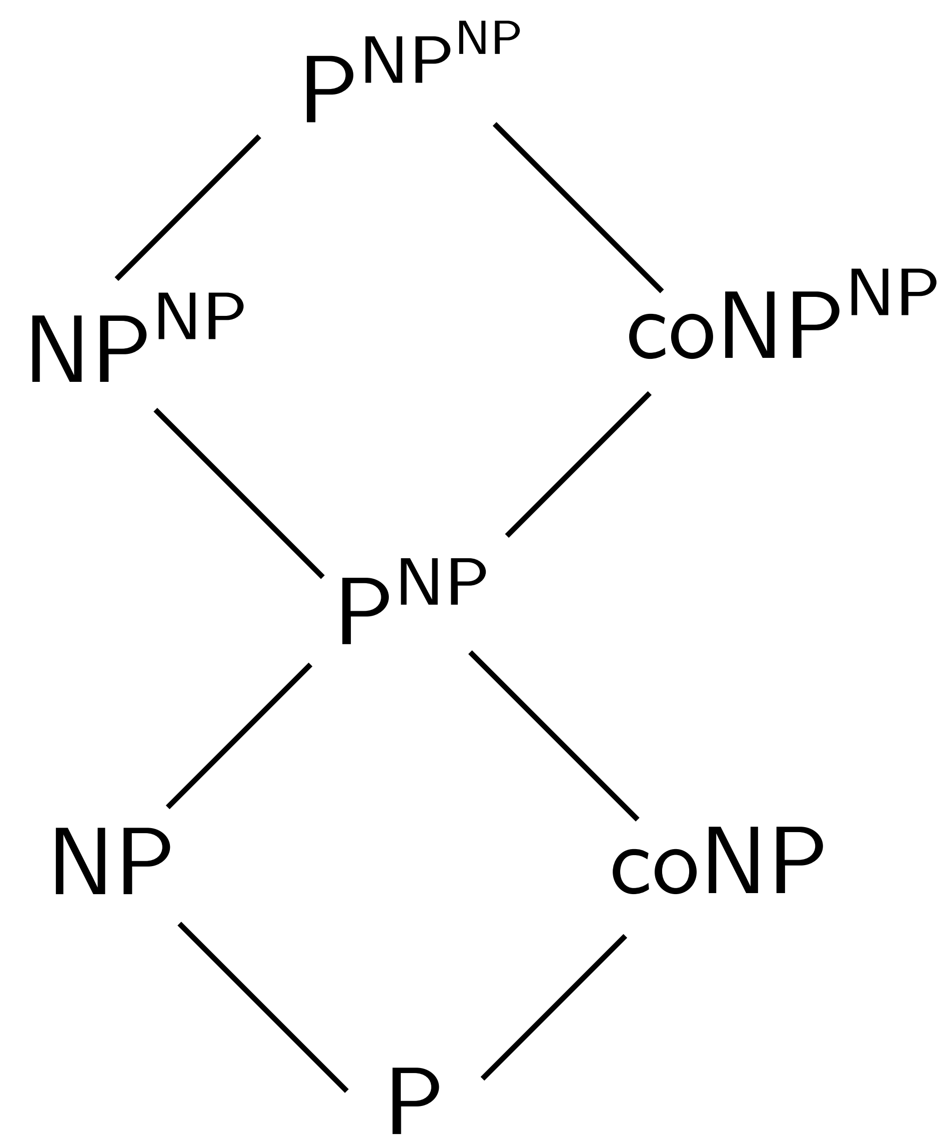
Polynomial Hierarchy

NP is the first level of what is known as the polynomial hierarchy PH.

The next level is given an oracle to the previous level.

We think the hierarchy is infinite.

If $\mathsf{P} = \mathsf{NP}$ it all collapses to P.



Polynomial Space

The entire polynomial hierarchy (and BQP) sits inside polynomial space.

We don't even know if $\mathsf{P} = \mathsf{PSPACE}$!

Our only hope is to show conditional hardness results.

The gold standard assumption is $\mathsf{P} \neq \mathsf{NP}$.

Next best is assuming PH is infinite.

BQP

A language L is in BQP iff there is a (uniform) family of polynomial size quantum circuits $\{Q_n\}$ such that

- If $x \in L$ then $\Pr[Q_{|x|}(x) = 1] \geq 2/3.$
- If $x \notin L$ then $\Pr[Q_{|x|}(x) = 1] \leq 1/3.$

The output is produced by measuring in the computational basis and returning the value of the first qubit.

BPP

A language L is in BPP iff there is a deterministic Turing machine M that takes input (x, y) where $y \in \{0,1\}^q$ for some $q = O(|x|^c)$, and

- M always terminates in polynomial time.
- If $x \in L$ then $\Pr_{y \in \{0,1\}^q}[M(x,y) = 1] \geq 2/3$.
- If $x \notin L$ then $\Pr_{y \in \{0,1\}^q}[M(x,y) = 1] \leq 1/3$.

We think that $\mathsf{BPP} = \mathsf{P}$. It is known that $\mathsf{BPP} \subseteq \mathsf{NP}^{\mathsf{NP}}$.

Approximately Samplable

Let B be quantum circuit acting on n qubits and let P_B be the probability distribution $P_B(x) = |\langle x|B|0^n\rangle|^2$.

Say that B is approximately samplable if there is a polynomial sized classical randomized circuit that generates a distribution P_C such that for all $x \in \{0,1\}^n$

$$0.9 \leq \frac{P_C(x)}{P_B(x)} \leq 1.1$$

Hardness Result

arXiv:1005.1407

Thm [BJS10]: If every family of BQP circuits is approximately samplable then the polynomial hierarchy collapses to the third level.

How can we use the hypothesis of the theorem?

Let's first see this implies $\mathsf{BQP} \subseteq \mathsf{BPP}$.

Baby Implication

Let $L \in \mathsf{BQP}$ and $\{Q_n\}$ a uniform family of quantum circuits that computes L.

To decide if $x \in \{0,1\}^n$ is in L we generate the circuit Q_n and ask our sampler to sample the output of Q_n run on x.

If $x \in L$ then the probability the first qubit is 1 is at least $2/3$.

The probability the sampler outputs a string with first bit 1 in this case is at least $0.9 \cdot 2/3$.

What next?

If every family of BQP circuits is approximately samplable then $\mathsf{BQP} \subseteq \mathsf{BPP}$.

Is the conclusion any more unbelievable than the hypothesis?

To amplify the craziness we bring in a crazy idea: postselection.

This allows you to condition on getting a certain measurement outcome.

postBQP

We again have a family of polynomial size quantum circuits. The first qubit is the selection qubit and the second qubit is the output qubit.

Now we look at the output conditioned on the selection qubit being 1.

$L \in \mathsf{postBQP}$ iff there is a family of BQP circuits s.t.

- $\Pr[\text{select} = 1] > 0$.
- If $x \in L$ then $\Pr[\text{output} = 1|\text{select} = 1] \geq 2/3$.
- If $x \notin L$ then $\Pr[\text{output} = 0|\text{select} = 1] \geq 2/3$.

postBPP

You can similarly define a randomized version. Here we think of it in terms of poly time algorithms $S(x,r)$ (selector) and $M(x,r)$ (determines output).

$L \in \text{postBPP}$ iff there are S and M such that

- $\Pr[S(x,r)=1] > 0$ for all x .
- If $x \in L$ then $\Pr_r[M(x,r)=1|S(x,r)=1] \geq 2/3$.
- If $x \notin L$ then $\Pr_r[M(x,r)=0|S(x,r)=1] \geq 2/3$.

$\text{postBPP} \subseteq \text{P}^{\Sigma_2}$ is in the polynomial hierarchy.

Question

Do you see how to solve satisfiability in $\mathsf{postBPP}$?

PP

A language L is in PP iff there is a deterministic Turing machine M that takes input (x, y) where $y \in \{0,1\}^q$ for some $q = O(|x|^c)$, and

- M always terminates in polynomial time.
- If $x \in L$ then $\Pr_{y \in \{0,1\}^q}[M(x,y) = 1] > 1/2$.
- If $x \notin L$ then $\Pr_{y \in \{0,1\}^q}[M(x,y) = 1] \leq 1/2$.

Toda's theorem: $\mathsf{PH} \subseteq \mathsf{P}^{\mathsf{PP}}$

Aaronson: $\mathsf{postBQP} = \mathsf{PP}$

Question

Do you see how to solve satisfiability in PP ?

Putting it all together

Thm [BJS10]: If every family of BQP circuits is approximately samplable then the polynomial hierarchy collapses to the third level.

Idea:

As with the $\mathsf{BPP} = \mathsf{BQP}$ proof we can use the hypothesis to show $\mathsf{postBPP} = \mathsf{postBQP}$.

Then

$$\mathsf{PH} \subseteq \mathsf{P}^{\mathsf{postBQP}} = \mathsf{P}^{\mathsf{postBPP}} \subseteq \mathsf{P}^{\Sigma_2}$$