On the complexity and verification of Random Circuit Sampling

Chinmay Nirkhe
nirkhe@cs.berkeley.edu
http://cs.berkeley.edu/~nirkhe
Joint work with

Adam Bouland
UC Berkeley

Bill Fefferman
QuICS (U. Maryland/NIST)

Umesh Vazirani
UC Berkeley

On the Complexity and Verification of Random Circuit Sampling
A. Bouland, B. Fefferman, C. Nirkhe, U. Vazirani
[Nature Physics, 2018] [arXiv:1803.04402] [ITCS 2019] [QIP 2019]
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.
REVOLUTIONARY QUANTUM COMPUTER IS ONE STEP CLOSER TO REALITY AFTER MAJOR BREAKTHROUGH

Why is there so much hype?

Why law firms need to worry about quantum computing

Safe and secure with blockchain

Will quantum computing break blockchain?
The extended Church-Turing thesis

Any “reasonable” method of computation can be efficiently simulated on a standard model (i.e. Turing machine, uniform circuits, etc.)
The extended Church-Turing thesis

Any "reasonable" method of computation can be efficiently simulated on a standard model (i.e. Turing machine, uniform circuits, etc.)

∃\(\mathcal{O}\) s.t. \(\text{BPP}^\mathcal{O} \neq \text{BQP}^\mathcal{O}\) [Simon\(^93\), Bernstein-Vazirani\(^93\)]

Quantum Computing!
The extended Church-Turing thesis

∃ 𝒪 s.t. $\text{BPP}^𝒪 \neq \text{BQP}^𝒪 \ [\text{Simon}^93, \text{Bernstein-Vazirani}^93]$

$\text{FACTORING} \in \text{BQP} \quad [\text{Shor}^94]$

$\text{BQP} = \text{the set of languages decidable by a polynomial time quantum algorithm}$
The extended Church-Turing thesis

\[ \exists \mathcal{O} \text{ s.t. } \text{BPP}^\mathcal{O} \neq \text{BQP}^\mathcal{O} \ [\text{Simon}^93, \text{Bernstein-Vazirani}^93] \]

So there is theoretical evidence, but is there anything tangible?

\[ \text{BQP} = \text{the set of languages decidable by a polynomial time quantum algorithm} \]
Experimental progress
Experimental progress

NISQ Era [Preskill\textsuperscript{18}]

Chinmay Nirkhe (UC Berkeley)
Oracle Separation

Experimental Progress

Quantum Algorithms

Complexity Theory

Quantum Supremacy
Quantum supremacy proposal

A practical demonstration of a quantum computation which is

1. Experimentally feasible
2. Has theoretical evidence of hardness
3. Verifiable
Quantum supremacy proposal

A practical demonstration of a quantum computation which is:

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

“an experimental violation of the extended Church-Turing thesis” – U. Vazirani
Why factoring is not the right proposal

The speedups come from carefully engineered interference patterns with large amounts of constructive and destructive interference. Which is hard to generate on the currently available noisy intermediate scale quantum devices.

This behavior is far from “typical”. It’s hard to make this happen in the lab!!

“Proving a quantum system’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\textsuperscript{11}]}
Complexity theory inspired supremacy proposals

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

Example: Boson Sampling [Aaronson\textsuperscript{11}]

Proves efficient classical algorithms cannot exist unless PH-collapses

Experimentally inspired supremacy proposals

Problems which we can experimentally test in the near future (\textasciitilde10 years)

Example: Random Circuit Sampling [BIS+\textsuperscript{16}]

Near-term experimentally feasible due to high-quality superconducting qubits
A Quantum Supremacy Proposal

Random Circuit Sampling

Given the description of a quantum circuit $C$, sample from the output distribution of the quantum circuit.
Part 0: What is quantum computing?
What is quantum computing?

It’s computing (really, information processing) based on the principles of quantum mechanics rather than classical physics.

Quantum mechanics is a description of nature

• Formulated to explain the behavior of subatomic particles.
• QM has been spectacularly successful in explaining microscopic physical phenomena.
What is quantum computing?

Quantum computers run in superposition. It’s like running probabilistically except there can be negative (complex) probabilities.

Remember physics?
What is quantum computing?

The state of a deterministic computation is a binary string
\[ x \in \{0,1\}^m \]

The state of a randomized computation is a probability distribution
\[ \{p_x\}_{x \in \{0,1\}^m} \quad \sum_x p_x = 1; \quad p_x \geq 0 \]

The state of a quantum computation is a superposition
\[ \sum_x \alpha_x |x\rangle \quad \sum_x \alpha_x^2 = 1; \quad \alpha_x \in \mathbb{C} \]
What is quantum computing?

Quantum computers are realized by measurement. Classical numbers which we can read.

\[ \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle \rightarrow \{p_x\}_{x \in \{0,1\}^n} \quad p_x = |\alpha_x|^2 \]
What is quantum computing?

At a high level, quantum computing gives us some of the power of parallel computation without multiple processors.

Some problems have good quantum algorithms, while we believe that for some quantum offers no improvement.
Part 1: Classical hardness of Random Circuit Sampling
Fix an architecture over quantum circuits

Given a circuit from the architecture, sample from its output probability distribution
Sampling from the exact output distribution of a quantum circuit is \#P-hard

Trick: Since proving \#P-hardness, by Toda’s Theorem can use \text{PH} reductions instead of just \text{P} reductions
Recall the polynomial hierarchy…

$$\text{PH} \subseteq P^{\#P} \ [\text{Toda}^{91}]$$
Exact classical sampling from quantum circuits would give us:

\[ \mathbb{P}^{\#P} \subseteq \mathbb{BPP}^{\mathbb{NP}} \]

Contradicts the non-collapse of the PH:

\[ \mathbb{BPP}^{\mathbb{NP}} \subseteq \Sigma_3 \subsetneq \text{PH} \subseteq \mathbb{P}^{\#P} \]

Toda’s Theorem

Pf: Estimating output probabilities is \#P-hard. Apply \( \mathbb{BPP}^{\mathbb{NP}} \) reduction due to Stockmeyer’s Thm\(^{85}\) to get sampling is also \#P-hard.
Therefore, exact quantum sampling is \#P-hard under $\text{BPP}^\text{NP}$-reductions
But…

No quantum device would exactly sample from the output distribution due to noise!

So in order to make this argument, we have to show that no classical sampler can even approximately sample from the same distribution!
By construction, the “hardness” of our circuit was in $\Pr(0)$.

If an adversary knew that, they could generate an approximate sampler that never outputted $0$. Thereby, short-circuiting the hardness proof.
We want to embed the hardness across all the outputs of the probability distribution.

We want a robustness condition: Being able to compute most probabilities should be \#P-hard.
\[
\text{Pr}[C \text{ outputs } 1101] = \text{Pr}[C_{1101} \text{ outputs } 0000]
\]
We want a robustness condition: Being able to compute most probabilities should be \#P-hard.

We want a robustness condition: Being able to compute $\Pr_0(C)$ for most circuits $C$ should be \#P-hard.

$\Pr_0(C) = \text{prob. } C \text{ outputs 0}$
What known problem has such a property?

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

Theorem [Lipton\textsuperscript{91},GLR+\textsuperscript{91}]: The following is \#\text{P}-hard: For sufficiently large \(q\), given uniformly random \(n \times n\) matrix \(M\) over \(\mathbb{F}_q\), output \(\text{perm}(M)\) with probability > \(\frac{3}{4} + 1/\text{poly}(n)\)
Permanent is avg-case hard

perm(A) = \sum_{\sigma \in S_n} \prod_{j=1}^{n} A_{j, \sigma(j)}

perm(A) is a degree n polynomial in the matrix entries of A

Choose R a random matrix. Let M(t) = A + Rt.
M(0) = A and M(t) for t \neq 0 is uniformly random.
perm(M(t)) is a degree n polynomial in t

Choose random t_1, ..., t_{n+1}, calculate perm(M(t_i))
Interpolate the polynomial perm(M(t)). Output perm(M(0))
Permanent is avg-case hard

\[ \text{perm}(A) = \sum_{\sigma \in S_n} \prod_{j=1}^{n} A_{j,\sigma(j)} \]

Assume we can calculate \( \text{perm}(R) \) for random \( R \) with probability \( > 1 - 1/(3n + 3) \).

By union bound, we calculate \( \text{perm}(A) \) with probability \( 2/3 \).

Since permanent is worst-case \#P-hard [Valiant\textsuperscript{79}],

This proves statement for probability \( 1 - 1/(3n + 3) \).

Better interpolation techniques bring the probability down to \( \frac{3}{4} + 1/\text{poly}(n) \).
Goal: find a similar polynomial structure in the problem of Random Circuit Sampling
High-level idea

Feynman Path Integral:

\[ \langle y_m | C | y_0 \rangle = \sum \prod_{j=1}^{m} \langle y_j | C_j | y_{j-1} \rangle. \]

Quantum analog of space-efficient brute-force evaluation of a circuit
\begin{equation}
\langle 0|C|0 \rangle = \sum \prod_{i=1}^{m} \langle y_j | C_j | y_{j-1} \rangle.
\end{equation}

Then \( \Pr_0(C) \) is a low-degree polynomial in the gate entries. We want to apply a similar interpolation technique as permanents.
Idea #1

\[ \langle 0|C|0 \rangle = \sum_{y_1, y_2, \ldots, y_{m-1}\in\{0,1\}^n} \prod_{j=1}^{m} \langle y_j|C_j|y_{j-1} \rangle. \]

Consider the circuit \( C(t) \) formed by changing each gate \( C_i \) to \( C_i + tH_i \) for random gate \( H_i \).

Just like permanent!

But, \( C_i + tH_i \) isn’t a quantum gate!
Idea #2

\[ \langle 0 | C | 0 \rangle = \sum_{y_1, y_2, \ldots, y_{m-1} \in \{0,1\}^n} \prod_{j=1}^{m} \langle y_j | C_j | y_{j-1} \rangle. \]

Consider the circuit \( C(\theta) \) formed by changing each gate

\[ C_i \mapsto C_i H_i e^{-i\theta h_i} \text{ where } h_i = -i \log H_i \]

1. \( C(1) = C \)
2. For small \( \theta \), circuit looks \( \theta \)-close to random!
3. Not a low-degree polynomial in \( \theta \)
Idea #2

\[ \langle 0|C|0 \rangle = \sum_{y_1,y_2,\ldots,y_m \in \{0,1\}^n} \prod_{j=1}^{m} \langle y_j|C_j|y_{j-1} \rangle. \]

Consider the circuit \( C(t) \) formed by changing each gate \( C_i \) to \( C_i + tH_i \) for random gate \( H_i \).

Just like permanent!

But, \( C_i + tH_i \) isn’t a quantum gate!
Idea #3

Taylor Series!

Replace $e^{-i\theta h_i}$ with

$$\sum_{k=0}^{\text{poly}(n)} \frac{(-i\theta h_i)^k}{k!}$$

1. $C(1) \approx C$
2. For small $\theta$, circuit looks $\theta$-close to random!
3. A low-degree polynomial in $\theta$
4. For more complicated technical reasons, this is a necessary, but not sufficient, proof of average-case hardness.

$$\langle 0|C|0 \rangle = \sum_{y_1,y_2,\ldots,y_{m-1} \in \{0,1\}^n} \prod_{j=1}^{m} \langle y_j|C_j|y_{j-1} \rangle.$$
Theorem: On average, it is hard to exactly sample from the output of random circuits.

This puts Random Circuit Sampling on par with the best known supremacy proposals.
Part 2: Verification of Random Circuit Sampling
How do we know if our quantum computer is doing what it says it's doing?
It’s a secret computation...
I don't remember.

What did you just do?
**Sweet spot in verification**

**Compromise:** OK with exponential post processing time on supercomputer to compute “a few” ideal output probabilities for “intermediate” size quantum computers ($n = 50$ qubits)

**Constraint:** can only take a small ($\text{poly}(n)$) number of samples from the quantum device

**Challenge:** Complexity arguments require closeness in total variation distance. But we can’t hope to unconditionally verify this with few samples from the device.
Candidate test: Cross-Entropy \([\text{Boixo} + 16]\)

\[
\text{CE}(p_{\text{dev}}, p_{\text{id}}) = \sum_x p_{\text{dev}}(x) \log \frac{1}{p_{\text{id}}(x)} = \mathbb{E}_{p_{\text{dev}}} \log \frac{1}{p_{\text{id}}}
\]

Can be approximated in a few samples
- Sample \(x_1, \ldots, x_k\) from quantum device
- Use exponential time to calculate \(p_{\text{id}}(x_i)\)
- Estimate \(\text{CE}\)

• If score is close enough to expected \(\text{CE}_{\text{ideal}}\), then accept.
Candidate test: Cross-Entropy [Boixo+16]

This is a one-dimensional projection of high-dimensional information

Does not verify closeness in total-variation distance

**Theorem:** If cross-entropy is close to ideal and

\[ H(p_{\text{dev}}) \geq H(p_{\text{id}}) \]

then the output distribution is close to ideal in total variation distance
Candidate test: Binned output generation

Consider dividing the [0,1] interval into poly(n) bins. Consider samples $x_1, \ldots, x_k$ and calculate ideal probabilities for each sample on a supercomputer. Plot and make sure the correct number of points are in each bin.

Thanks!