Quantum computational supremacy overview

Michael Bremner
A potential quantum (near) future

Unambiguous quantum computational supremacy and commercially relevant applications, 2 – 10 years

99.7% fidelity

99.99% fidelity

99.999% fidelity

FT qubit

Classical/quantum frontier

<12 months (Google, IBM)

99.99% fidelity

10+ years

Universal quantum computing...

A potential quantum (near) future

FT qubit
99.7% fidelity

99.9% fidelity

50 qubits

1,000

10,000

100,000...

Unambiguous quantum computational supremacy and commercially relevant applications, 2 – 10 years

<12 months
(Google, IBM)

99.99% fidelity

99.999% fidelity

10+ years

Universal quantum computing...

Classical/quantum frontier

A potential quantum (near) future

Noisy Intermediate-Scale Quantum computing (NISQ), application development for
- chemistry
- materials
- finance

Unambiguous quantum computational supremacy and commercially relevant applications, 2 – 10 years

<12 months (Google, IBM)

99.99% fidelity

10+ years

99.999% fidelity

Universal quantum computing...

Classical/quantum frontier

FT qubit

99.7% fidelity

50 qubits

1,000

10,000

100,000...

Quantum supremacy using a programmable superconducting processor
“We made these early bets because we believed—and still do—that quantum computing can accelerate solutions for some of the world’s most pressing problems, from climate change to disease. Given that nature behaves quantum mechanically, quantum computing gives us the best possible chance of understanding and simulating the natural world at the molecular level.”

— Sundar Pichai, CEO Google, 23 October 2019

“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.”

– John Preskill, “Quantum computing and the entanglement frontier”
arXiv:1203.5813
The quantum frontier?

Aim: Perform a quantum computation that cannot be performed classically in any reasonable amount of time.

Key issues:
- Are quantum computers superpolynomially more powerful than classical computers?
- For which computations do the classical and quantum runtimes radically diverge?
- Can we go beyond the classical frontier without fault tolerance?

For any such application:
Theoretical goals:
- minimize the gate and qubit count
- understand the influence of errors
- mitigate errors
- improve classical simulation algorithms

*Nothing on this plot is to scale!
The number of bits required to precisely describe even modest sized quantum systems quickly eclipses the number of atoms in the entire universe.
The number of bits required to precisely describe even modest sized quantum systems quickly eclipses the number of atoms in the entire universe.
Accurately simulating quantum mechanical systems is very hard. There seems to be a fundamental disconnect between the way that classical computers can describe probabilistic systems and the way that probabilities emerge at the quantum level.
Qubits are the simplest quantum mechanical systems.

The state of an isolated qubit is represented by a two-dimensional, complex, unit vector.

Measurement is discrete and probabilistic. As this is a two-dimensional state, it has two possible outcomes defined by opposing poles on the sphere.

The probability is defined by the inner product, or amplitude between the initial state, and the final, measured, state.

Dynamics are defined by operations on the state. For an isolated single qubit these are 2 by 2 unitary matrices.

The state of an entangled qubit is dependent on the states of the qubits that it is entangled with. It is represented by a density matrix, a probabilistic mixture of pure states.

\[ |\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle, \]
\[ \langle\psi|\psi\rangle = |\alpha_0|^2 + |\alpha_1|^2 = 1 \]

The qubit
Many qubits

The state of an \( n \) qubit system is a \( 2^n \) dimensional, complex, unit vector.

\[
|\psi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle
\]

How do we even write that down?

We describe a Hamiltonian, or energy, operator that describes the interactions between the qubits. Usually this has a polynomial sized description.

E.g. \( H = \sum_{ij} w_{ij} Z_i Z_j + \sum_i v_i X_i \)

Dynamics are determined by the exponential of \( H \) (from Schrödinger’s equation).

\[
|\psi(t)\rangle = e^{-iHt} |\psi(0)\rangle
\]
The state of an n qubit system is a $2^n$ dimensional, complex, unit vector.

$$|\psi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$$

How do we even write that down?

We describe a Hamiltonian, or energy, operator that describes the interactions between the qubits. Usually this has a polynomial sized description.

E.g. $H = \sum_{ij} w_{ij} Z_i Z_j + \sum_i v_i X_i$

Dynamics are determined by the exponential of $H$ (from Schroedinger's equation).

$$|\psi(t)\rangle = e^{-iHt} |\psi(0)\rangle$$

The probability of measuring $x$ is given by the amplitude with the state $|x\rangle$:

$$\Pr(x) = |\langle x | e^{-iHt} |\psi(0)\rangle|^2.$$
Quantum simulation

Using quantum control, we can manipulate interactions to allow one quantum system to simulate, or mimic, the behaviour of another.
“I therefore believe it’s true that with a suitable class of quantum machines you could imitate any quantum system, including the physical world.”

Quantum computing

Quantum computers are devices specifically built to be good at simulating quantum systems.

In quantum circuits, gates describe how to control interactions between qubits.

The circuit output is simply the measurement outcome, i.e. a string of 1s and 0s.
“quantum computing presents a serious challenge to the so-called extended Church-Turing thesis: that any function naturally to be regarded as efficiently computable is efficiently computable on a Turing machine.”

- Scott Aaronson, “Quantum computing since Democritus” (2013)
**Quantum computers**

**Important facts:**

- A quantum computer does not **“compute”** \( P(x) \), we can only infer \( P(x) \) from measurements/samples, \( x \).

- For an arbitrary, polynomially sized, \( U \) this means we can only ever estimate \( R(x) \), such that:
  \[
  |P(x) - R(x)| \leq 1/poly(n).
  \]

- In general, fault-tolerant quantum computers can only be guaranteed to produce samples from \( R(x) \) such that:
  \[
  \sum_x |P(x) - R(x)| \leq \frac{1}{poly(n)}.
  \]

**Input:** classically easy to describe circuit, \( U \), and input state \( |0\rangle^\otimes n \).

\( U \) is comprised of polynomially many **gates** from a finite gate set e.g. \( T, H, CZ \).

**Universal** gate sets have the property that they can simulate any unitary (assuming arbitrary runtime and ignoring errors).

**Output:** strings of bits \( x \in \{0,1\}^n \) with probability \( P(x) = |\langle x | U | 0 \rangle^\otimes n|^2 \).
Task: Prepare measurement samples from a quantum circuit.

How?

On a quantum computer just run your quantum circuit and measure.

On a classical computer one way would be to compute the histogram of potential outputs and use it to prepare samples.

Q: Is this the best way?
A: It’s complicated.
Task: Prepare measurement samples from a RANDOM low-depth quantum circuit.

How?

On a classical computer one way would be to compute the histogram of potential outputs and use it to prepare samples.

Q: Is this the best way?

A: Yes, assuming:

1) Theoretical computer science makes sense.

2) Random circuit outputs are as complicated as the non-random ones.
Google proposal summary

Best known classical complexity is exponential in number of qubits and circuit depth.

Introduces cross entropy benchmarking to establish validity and as a new methodology for benchmarking large circuits.

Heavy numerical testing to establish:
- level of randomness in the circuit.
- benchmarking.
- and the point at which these circuits become “hard” for powerful classical computers.


Updated gates: see Google AI blog

time/depth $\sim O(n^{1/2}) > 40$ layers of gates each with infidelity around $\sim 1/(\text{circuit size})$

$7 \times 7$ array of qubits

H gates

CZ gates

$T, X^{1/2}, Y^{1/2}$ gates
Google experiment

- Performed over 20 cycles of 1 and 2 qubit gates.
- Linear cross entropy used to benchmark data.
- Heavy use of numerical testing to test randomness and benchmarking.
- Extensive testing of the error models.

*See Arute et al Nature October 2019*
Random circuit sampling

Goal: output samples, $x$, with probability $P(x)$ defined by a random circuit.

Idea: bound the complexity of sampling via studying the properties of $P(x)$.

Hope: If $P(x)$ is “complex” enough then classical computers can only ever simulate sampling by computing $P(x)$.

Intuition: Random circuits quickly develop long-range entanglement, making them among the hardest to simulate accurately for known classical algorithms.
Aaronson and Arkhipov’s Boson Sampling established a potential advantage over classical computing for sampling random linear optical networks.

Importantly, the advantage holds for approximate sampling, ruling out classical algorithms outputting samples from $R(x)$ such that $\|P - R\|_1 \leq \varepsilon$ assuming 2, open, conjectures.

---

**Random circuit sampling**

**Goal:** output samples, $x$, with probability $P(x)$ defined by a random circuit.

**Idea:** bound the complexity of sampling via studying the properties of $P(x)$.

**Hope:** If $P(x)$ is “complex” enough then classical computers can only ever simulate sampling by computing $P(x)$.

**Intuition:** Random circuits quickly develop long-range entanglement, making them among the hardest to simulate accurately for known classical algorithms.

---

IQP circuit $C = H^nDH^n$

B., Montanaro, Shepherd
Aaronson and Arkhipov’s Boson Sampling established a potential advantage over classical computing for sampling random linear optical networks.

Importantly, the advantage holds for approximate sampling, ruling out classical algorithms outputting samples from $R(x)$ such that $\|P - R\|_1 \leq \varepsilon$, assuming 2, open, conjectures.

IQP Sampling “improved” on Boson Sampling by proving the equivalent of the “Permanent anti-concentration conjecture”.

It is defined in the quantum circuit model, and yields very low-depth circuits. It also allows the usual machinery of error-correction to apply.

This model is also easily generalized e.g. Boixo et al arXiv:1608.00263.

**Random circuit sampling**

**Goal:** output samples, $x$, with probability $P(x)$ defined by a random circuit.

**Idea:** bound the complexity of sampling via studying the properties of $P(x)$.

**Hope:** If $P(x)$ is “complex” enough then classical computers can only ever simulate sampling by computing $P(x)$.

**Intuition:** Random circuits quickly develop long-range entanglement, making them among the hardest to simulate accurately for known classical algorithms.

IQP circuit $C = H^\otimes n DH^\otimes n$

B., Montanaro, Shepherd
The complexity of $P(x) = |\langle x | U | 0 \rangle \otimes^n |^2$

**Input:** a classical description of $U$ (and $x$) polynomial in $n$.

**Output:** $R(x)$, or $x$ from $R(x)$, in time polynomial in the size of the description of $U$ and the inverse of the error.

<table>
<thead>
<tr>
<th>Output</th>
<th>Accuracy</th>
<th>Complexity upper bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R(x)$</td>
<td>$R(x) = P(x)$ “exact”</td>
<td>GapP</td>
</tr>
<tr>
<td>$x \in {0, 1}$</td>
<td>$|P - R|_1 \leq \epsilon$ “additive sampling error”</td>
<td>BQP</td>
</tr>
<tr>
<td>$R(x)$</td>
<td>$</td>
<td>P(x) - R(x)</td>
</tr>
<tr>
<td>$x \in {0, 1}^m$</td>
<td>$|P - R|_1 \leq \epsilon$ “additive error”</td>
<td>SampBQP</td>
</tr>
<tr>
<td>$x \in {0, 1}^m$</td>
<td>$\frac{1}{c} P(x) \leq R(x) \leq c P(x), \forall x$ “multiplicative sampling error”</td>
<td>No classical $R(x)$ unless $\text{PH} = \text{PH}_3$</td>
</tr>
<tr>
<td>$R(x)$</td>
<td>$</td>
<td>P(x) - R(x)</td>
</tr>
</tbody>
</table>
The complexity of $P(x) = |\langle x | U | 0 \rangle \otimes^n|^2$

**Goal:** bound the complexity of producing samples, $x$, by the complexity of computing $P(x)$. 
The complexity of $P(x) = \left| \langle x|U|0\rangle \otimes^n \right|^2$

**Goal:** bound the complexity of producing samples, $x$, by the complexity of computing $P(x)$.

**NP:** decision problems that can be verified by uniform poly sized circuits.

**P:** decision problems solvable with uniform poly sized circuits.
The complexity of $P(x) = |\langle x | U | 0 \rangle \otimes^n |^2$

**Goal:** bound the complexity of producing samples, $x$, by the complexity of computing $P(x)$.

**Exact:** GapP hard

**Relative error:** GapP hard

$|R(x) - P(x)| \leq \gamma P(x)$

**GapP:** computes the difference between #P functions.

i.e. $|\{x:f(x) = 1\}| - |\{x:f'(x) = 1\}|$.

**#P:** counts the number of inputs to a poly sized circuit that evaluate to 1. i.e. $|\{x:f(x) = 1\}|$

**NP:** decision problems that can be verified by uniform poly sized circuits.

**P:** decision problems solvable with uniform poly sized circuits.

$\text{NP} \subseteq \text{P} \subseteq \text{BPP} \subseteq \text{GapP} \subseteq \text{#P} \subseteq \text{NP}$
The complexity of \( P(x) = |\langle x| U |0\rangle \otimes^n| 2 \)

**Goal:** bound the complexity of producing samples, \( x \), by the complexity of computing \( P(x) \).

**Exact:** GapP hard

**Relative error:** GapP hard

\[ |R(x) - P(x)| \leq \gamma P(x) \]

**Additive error:** BQP-hard

\[ |R(x) - P(x)| \leq 1/poly(n) \]

**GapP:** computes the difference between \#P functions.

i.e. \(|\{x:f(x) = 1\}| - |\{x:f'(x) = 1\}|\).

**#P:** counts the number of inputs to a poly sized circuit that evaluate to 1. i.e. \(|\{x:f(x) = 1\}|\)

**NP:** decision problems that can be verified by uniform poly sized circuits.

**P:** decision problems solvable with uniform poly sized circuits.
GapP and quantum computing

Fortnow and Rogers/Fenner et al (circa ’97): computing the amplitude of a quantum circuit is GapP-complete.

**GapP**: Let C be a classical circuit that computes a Boolean function $C : \{0,1\}^n \rightarrow \{-1,1\}$. Given C as input, compute $\Delta_C$ which is given by:

$$\Delta_C := \sum_{x \in \{0,1\}^n} C(x)$$

GapP generalizes #P to encompass negative valued functions. It isn’t too hard to see that GapP $\supseteq$ #P.

Relative error (i.e. multiplicative) approximations to GapP-complete problems are still GapP-complete. Implies $|A-P(0^n)| \leq \Omega_P(0^n)$ is #P-hard. This is not true for #P functions.

$$\langle 0 | \otimes (n+r) U_C | 0 \rangle \otimes n+r = \frac{\Delta_C}{2^n}$$
Counting by searching

**Problem:**
Compute $\Delta_C$ precisely in time $\text{poly}(n)$ given ability to compute the sign of any $\Delta_c$.

$$\Delta_C := \sum_{x \in \{0,1\}^n} C(x)$$

Standard method, but see Aaronson (1109.1674):
Assume we can compute $\text{sgn}(\Delta_C)$ for $C$.

Define two alternate circuits $C[\pm k]$ which are exactly the same as $C$ except they introduce $k$ additional inputs such that $C[k](x) = \pm 1$. Hence $\Delta C[\pm k] = \Delta C \pm k$.

**Algorithm:**
- Compute the signs of $\Delta_C[\pm k]$ starting with $k = 1$ and increasing $k$ by factors of two until $\text{sgn}(\Delta_C[k]) \neq \text{sgn}(\Delta_C[2k])$.
  - *At this point we know that $\Delta_C$ is between $k$ and $2k$.*
- Then compute $\text{sgn}(\Delta_C[3k/2])$ etc to ultimately determine the exact value for $\Delta_C$.
- The complexity of such a procedure is $O(\text{poly}(n))$. 
GapP and quantum computing

Consider diagonal $D_C$ with $(x,x)$ entries given by $C(x)$.

$$\langle 0 \otimes^n \tilde{D}_C | 0 \rangle \otimes^n = \langle 0 | \otimes^n H \otimes^n D_C H \otimes^n | 0 \rangle \otimes^n$$

$$= \left( \frac{1}{\sqrt{2^n}} \sum_{x' \in \{0,1\}^n} \langle x' \rangle \right) D_C \left( \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} | x \rangle \right)$$

$$= \frac{1}{2^n} \sum_{x,x' \in \{0,1\}^n} C(x) \delta(x,x')$$

$$= \frac{\Delta_C}{2^n}$$

- If we know the amplitude and $n$ we can compute $\Delta_C$ precisely.
- Note: if there is any additive error in the amplitude in will be multiplied by $2^n$ - which is terrible!
- $D_C$ is not uniformly generated.
Uniform generation

• $D_C$ can be generated by the “standard” tricks of reversible computation and the phase-kickback trick (see any lecture series on QC or Mike and Ike) via a uniformly generated circuit $U_C$.

• $U_C$ be performed with Toffoli, Hadamard and X gates at a cost of (approximately) no more than the classical cost of performing $C$ reversibly.

• This requires the use of $r$ ancilla “scratch and output” qubits - where $r < |C|$.

$$U_C|x\rangle \otimes |0\rangle^r = C(x)|x\rangle \otimes |0\rangle^r$$
So we see computing the sign of an amplitude for a general quantum circuit is GapP-hard.

It is possible to show that computing the corresponding probability is also GapP hard to within constant relative error via similar, but more complicated, arguments. (see appendix of BMS’15) i.e. finding an approximation $A$ such that $|A - P^n| \leq \gamma P(n)$ is also GapP-complete.

**Key** to the argument is that $U_C$ necessarily must use a classically universal set of quantum gates.

This can be relaxed significantly both (1) to non-universal gate sets, and (2) to gate sets with algebraic entries.

\[
\Delta_C := \sum_{x \in \{0,1\}^n} C(x)
\]

\[
\langle 0 \otimes (n+r) | H \otimes n U_C H \otimes n | 0 \rangle \otimes (n+r) = \langle 0 \otimes (n+r) | \tilde{U} | 0 \rangle \otimes (n+r) = \Delta_C / 2^n
\]

\[
|0\rangle \otimes (n+r) \\
\vdots
\tilde{U} \\
\vdots
\langle 0| \otimes (n+r)
\]
Which circuit families are hard?

Post-selection constructions can be used to show that complexity can be inherited amongst different circuit families. These are usually derived from identities used in measurement-based quantum computing where the complexity of relative error approximation $|A - P(0n)| \leq \gamma P(0n)$, can be preserved.

This key trick was used in Terhal and DiVincenzo (quant-ph/0205133) to argue that constant depth circuits cannot be classically sampled.

It can also be used to show GapP hardness for IQP circuits, Boson Sampling networks and many other circuit families.

Take a “hard” $U$ where computing $P(x)$ is GapP complete.

$$P(x) = |\langle x | U | 0 \rangle \otimes n|^2$$

Construct a new circuit $U_P$ with a $p$ qubit register such that

$$|\langle x, 0^p | U_P | 0 \rangle \otimes (n+p)|^2 = \frac{1}{2^p} |\langle x | U | 0 \rangle \otimes n|^2$$
Relative error approximations: $|A_x - f| \leq \gamma f$, bounding classical complexity

$f = \langle x | U | y \rangle$ stays GapP-hard even with approximation.

Stockmeyer’s algorithm can give a relative error approximation $A_x$ for functions $f$ that are inside $\#P$ inside $\text{BPP}^{\text{NP}}$.

This does not work for GapP-hard functions unless the PH collapses.
Relative error approximations: $|A_x - f| \leq \gamma f$, bounding classical complexity

$f = \langle x|U|y \rangle$ stays GapP-hard even with approximation.

Stockmeyer’s algorithm can give a relative error approximation $A_x$ for functions $f$ that are inside $\#P$ inside $BPP^{NP}$.

This does not work for GapP-hard functions unless the PH collapses.

Terhal and DiVincenzo ’02, Bremner, Jozsa, Shepherd/Aaronson and Arkhipov ’10: GapP-hardness of relative error approximations of $\langle x|U|y \rangle$ for constant depth circuits, IQP circuits, linear optics, and any family universal for quantum computation with post-selection.

These results also showed that if there were efficient, classical, multiplicative error simulations of the outputs of these families then it would be possible to compute $\langle x|U|y \rangle$ with Stockmeyer’s algorithm — causing a PH collapse.
Stockmeyer’s counting theorem

There exists an $\text{FBPP}^{\text{NP}}$ machine which, for any boolean function $f: \{0,1\}^n \to \{0,1\}$, can approximate

$$p = \Pr_x[f(x) = 1] = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)$$

to within a relative error $|\hat{p} - p| < \epsilon p$ for $\epsilon = \Omega[1/\text{poly}(n)]$, given oracle access to $f$.

- This theorem implies that any function in $\#P$ has a good relative error approximation inside the Polynomial Hierarchy.

- In the case of quantum sampling if we want to approximate, say, $P(y) = |\langle y \| U \| 0^n \rangle|^2$, and we had an efficient classical sampler, $f$ would output 1 if the sampler outputs $y$. 
We want to know how many x satisfy \( f(x) = 1 \) for a Boolean function, \( f \). That is we want to know the size of the set \( S = \{x | f(x) = 1\} \).

Note, if we can guess \( |\tilde{S}| \leq 2|S| \) we can also guess to within a factor \( 1 + 1/n^c \).

Create \( d(x, y) = f(x) \land f(y) \), which has solution set of size \( |S|^2 \). If we can determine this solution to within a factor of 2 we can determine \( |S| \) to a factor of \( 2^{1/2} \). Doing this \( k \) times leads to a \( 2^{1/k} \) approximation.

Finding a factor 2 approximation:

Let \( h: \{0,1\}^n \to \{0,1\}^m \) such that \( n \gg m \) be a randomly chosen hash function.

We now want to check if there are \( x \) and \( y \) such that \( h(x) = h(y) \) and both are in \( S \). This can be done with an NP oracle call.

If \( |S| \) is much bigger than \( 2^m \) the likelihood of collision is high.

By analysing the likelihood of collision and varying the family of hash functions we can find an approximation to \( |S| \).
In Bremner, Jozsa, and Shepherd arXiv:1005.1407 a proof using the properties of post-selected decision languages is used to show:

If the output probability distributions generated by uniform families of SampBQP circuits could be weakly classically simulated to within multiplicative error $1 \leq c < 2^{1/2}$ then $\text{postBPP} = \text{PP}$. 

\[ \frac{1}{c} P(x) \leq R(x) \leq c P(x), \forall x \]

“multiplicative sampling error”
Additive vs multiplicative approximations

Consider 2 distributions:

\[ P = \left( \frac{1}{2}, \frac{1}{2} - w, w \right) \] and \[ R = \left( \frac{1}{2}, \frac{1}{2}, 0 \right) \]

where \( w \) is ridiculously small for whatever measure of ridiculousness you want.

\[ \|P - R\|_1 = \sum_x |P(x) - R(x)| \leq \epsilon \]

Additive

\[ \frac{1}{c} P(x) \leq R(x) \leq c P(x), \ \forall x \]

Multiplicative

\[ \|P - R\|_1 = 2w. \] If we want \( \|P - R\|_1 \leq \epsilon, \) \( \epsilon \) only has to be small (not ridiculously small).

Whereas \( R \) can never approximate \( P \) to within any multiplicative factor. \( (1/c)w \neq 0 \leq cw \)
Aaronson and Arkhipov’s great idea!

If you could simulate linear optics classically, and if you have a $\text{BPP}^{\text{NP}}$ machine, you might be able to use Stockmeyer’s theorem to compute complex matrix permanents. This would cause a PH collapse.

Importantly, we will see that approximations to randomly chosen circuits are particularly well approximated via Stockmeyer counting.
Relative error approximations: $|A_x - f| \leq \gamma f$, bounding classical complexity

$f = \langle x | U | y \rangle$ stays GapP-hard even with approximation.

Stockmeyer’s algorithm can give a relative error approximation $A_x$ for functions $f$ that are inside $\mathbb{P}$ inside $\mathbb{BPP}^{\mathbb{NP}}$.

This does not work for GapP-hard functions unless the PH collapses.
Relative error approximations: $|A_x - f| \leq \gamma f$, bounding classical complexity

$f = \langle x | U | y \rangle$ stays GapP-hard even with approximation.

Stockmeyer's algorithm can give a relative error approximation $A_x$ for functions $f$ that are inside $\#P$ inside $\text{BPP}^\text{NP}$.

This does not work for GapP-hard functions unless the PH collapses.

The “quantum random circuit sampling” argument:
If there exists sufficiently accurate* efficient classical samplers for the outputs from sufficiently random $U$, Stockmeyer's algorithm can be used to approximate $f = \langle x | U | y \rangle$ (which could be GapP-hard).

* sufficiently accurate = constant 1-norm distance
Stockmeyer and random circuits

\[ p_{xy} = |\langle y | U_x | 0^n \rangle|^2 \]

\[ U_x = XU \]
Stockmeyer and random circuits

\[ p_{xy} = |\langle y | U_x | 0^n \rangle|^2 \]

\[ \|p - q\|_1 = \sum_{y \in \{0,1\}^n} |p_{xy} - q_{xy}| \leq \epsilon \]

\[ q_{xy} = \Pr[A \text{ outputs } y \text{ on input } U_x] \]
Stockmeyer and random circuits

\[ \|p - q\|_1 = \sum_{y \in \{0,1\}^n} |p_{xy} - q_{xy}| \leq \epsilon \]

\[ q_{xy} = \Pr[A \text{ outputs } y \text{ on input } U_x] \]

Lemma: Given U, let \( U_x \) be the product \( XU \) where \( X = \bigotimes X^{x_i} \) for a choice of bitstring \( x \in \{0,1\}^n \).

Given an efficient classical sampler A for \( U_x \) with \( l_1 \) accuracy \( \epsilon \) and an FBPP\(^{NP} \) machine we can use Stockmeyer’s algorithm to approximate \( P_{x_0} \) to additive error:

\[
O \left( \frac{(1 + o(1))\epsilon}{2^n \delta} + \frac{|\langle 0^n |U_x|0^n \rangle|^2}{\text{poly}(n)} \right)
\]

with probability at least \( 1 - \delta \) over the choice of \( x \).
For any choice of $y$ we can use Stockmeyer’s algorithm to produce an approximation

$$|\tilde{q} - q_{0y}| \leq \frac{q_{0y}}{\text{poly}(n)} \text{ in } \text{FBPP}^\text{NP}.$$ 

Then,

$$|\tilde{q}_y - p_{0y}| \leq |\tilde{q}_y - q_{0y}| + |q_{0y} - p_{0y}| \leq \frac{q_{0y}}{\text{poly}(n)} + |q_{0y} - p_{0y}| \leq \frac{p_{0y} + |q_{0y} - p_{0y}|}{\text{poly}(n)} + |q_{0y} - p_{0y}| \leq \frac{p_{0y}}{\text{poly}(n)} + \left|q_{0y} - p_{0y}\right|(1 + \frac{1}{\text{poly}(n)})$$

Stockmeyer and random circuits

$$\|p - q\|_1 = \sum_{y \in \{0,1\}^n} |p_{xy} - q_{xy}| \leq \epsilon$$

$$p_{xy} = |\langle y | U_x | 0^n \rangle|^2$$

$$q_{xy} = \Pr[A \text{ outputs } y \text{ on input } U_x]$$

+ NP
For any choice of $y$ we can use Stockmeyer’s algorithm to produce an approximation

$$|\tilde{q} - q_{0y}| \leq \frac{p_{0y}}{\text{poly}(n)} + |q_{0y} - p_{0y}|(1 + \frac{1}{\text{poly}(n)})$$

Then as $A$ approximates outputs of $U_0$ to $l_1$ error $\epsilon$ we find from Markov’s inequality that:

$$\Pr_y\left[|q_{0y} - p_{0y}| \geq \frac{\epsilon}{2^n \delta}\right] \leq \delta$$

For any $0 \leq \delta \leq 1$ where $y$ is picked uniformly at random.

$$|\tilde{q}_y - p_{0y}| \leq \frac{p_{0y}}{\text{poly}(n)} + \frac{\epsilon(1 + \frac{1}{\text{poly}(n)})}{2^n \delta}$$

With probability $1 - \delta$ over the choice of $y$. But, $p_{0y} = |\langle y|U_0|0^n\rangle|^2 = |\langle 0^n|U_y|0^n\rangle|^2 = p_{y0}$. 

$$\|p - q\|_1 = \sum_{y \in \{0,1\}^n} |p_{xy} - q_{xy}| \leq \epsilon$$

$$p_{xy} = |\langle y|U_x|0^n\rangle|^2$$

$$q_{xy} = \Pr[A \text{ outputs } y \text{ on input } U_x]$$

$|\langle 0^n|U_y|0^n\rangle|^2 = p_{y0}$. 

$\|$
Relative error approximations

\[ |\tilde{q}_y - p_{y0}| \leq \frac{p_{0y}}{\text{poly}(n)} + \epsilon \left(1 + \frac{1}{\text{poly}(n)}\right) \frac{1}{2^n \delta} \]

When does this yield a relative error approximation to \( p_{0y} \)? When \( p_{0y} \geq \alpha \cdot 2^{-n} \).

**Corollary**

Let \( U \) be randomly chosen from some family \( F \). Apply a random \( X \) to \( U \) (note: this step is not required depending on the family \( F \)). Assume there exists constants \( \alpha, \beta > 0 \) such that:

\[ \Pr_{U_x \in F} \left[ |\langle 0^n | U_x | 0^n \rangle|^2 \geq \alpha \cdot 2^{-n} \right] \geq \beta \]

Assume also that \( A \) can simulate the output of any \( U \) up to \( l_1 \) error \( \epsilon = \frac{a\beta}{8} \). Then Stockmeyer’s algorithm gives a relative error approximation of \( \frac{1}{4} + o(1) \) for a \( \beta/2 \) fraction of \( U \).
Relative error approximations

\[ |\tilde{q}_y - p_{y0}| \leq \frac{p_{0y}}{\text{poly}(n)} + \frac{\epsilon(1+\frac{1}{\text{poly}(n)})}{2^n \delta} \]

When does this yield a relative error approximation to \( p_{0y} \)? When \( p_{0y} \geq \alpha \cdot 2^{-n} \).

**Corollary**

Let \( U \) be randomly chosen from some family \( F \). Apply a random \( X \) to \( U \) (note: this step is not required depending on the family \( F \)). Assume there exists constants \( \alpha, \beta > 0 \) such that:

\[ \Pr_{U_x \in F} [|\langle 0^n | U_x | 0^n \rangle |^2 \geq \alpha \cdot 2^{-n}] \geq \beta \]

Assume also that \( A \) can simulate the output of any \( U \) up to \( l_1 \) error \( \epsilon = \frac{\alpha \beta}{8} \). Then Stockmeyer’s algorithm gives a relative error approximation of \( \frac{\beta}{4} + o(1) \) for a \( \beta/2 \) fraction of \( U \).
Relative error approximations

\[
|\tilde{q}_y - p_{y0}| \leq \frac{p_{y0}}{\text{poly}(n)} + \frac{\epsilon(1+\frac{1}{\text{poly}(n)})}{2^n \delta}
\]

When does this yield a relative error approximation to \( p_{0y} \)? When \( p_{0y} \geq \alpha \cdot 2^{-n} \).

**Corollary**

Let \( U \) be randomly chosen from some family \( F \). Apply a random \( X \) to \( U \) (note: this step is not required depending on the family \( F \)). Assume there exists constants \( \alpha, \beta > 0 \) such that:

\[
\Pr_{U_x \in F} [ |\langle 0^n | U_x | 0^n \rangle |^2 \geq \alpha \cdot 2^{-n} ] \geq \beta
\]

Assume also that \( A \) can simulate the output of any \( U \) up to \( l_1 \) error \( \epsilon = \frac{\alpha \beta}{8} \). Then Stockmeyer’s algorithm gives a relative error approximation of \( \frac{1}{4} + o(1) \) for a \( \beta/2 \) fraction of \( U \).

We call this property “anticoncentration”.

Random quantum circuit sampling argument

E.g. some family of $U_x$ where computing $P_0 = |\langle 0^n | U_0 | 0^n \rangle|^2$ is \#P-hard even up to relative error approximation in the worst-case.
Random quantum circuit sampling argument

E.g. some family of $U_x$ where computing $P_0 = |\langle 0^n | U_0 | 0^n \rangle|^2$ is #P-hard even up to relative error approximation in the worst-case.

What we can prove is more like this:

Approximations to $P_x$ are as hard as approximate $P_0$ i.e. #P-hard.

$P_x$ anti-concentrates (sampling approximates $P_x$, $P_x \geq \Omega(2^{-n})$)
Average case complexity conjectures

• Any problem that is GapP-complete is “randomly self reducible”. – Feigenbaum and Fortnow ‘91.

• Aaronson and Arkhipov proved average case complexity for exact Boson Sampling.

• Recently Bouland, Fefferman, Nirkhe, and Vazirani proved that the average case complexity of exactly computing amplitudes for random circuits similar to the “Google proposal” are at least #P-hard (Nature Physics ’19, arXiv:1803.04402).

• Still to be addressed is the problem of the complexity of approximations and proof of this is likely to require non-relativising techniques – see Aaronson and Chen arXiv:1612.05903
Random circuit sampling

Challenge is to identify circuit families that:

1) Have \#P-hard amplitudes in the worst-case. i.e. post-selected family is equivalent to postBQP.

2) Display anticoncentration on an output register “growing like n”.

3) Are sufficiently complex such that it is likely that random instances are also \#P-hard.

4) As a bonus it would be nice to know which instances are likely to be classically hard so that we can estimate classical runtimes.
Random circuit families

A number circuit families have been shown to have $\#P$-hard amplitudes and anticoncentrate. Anticoncentration proof techniques fall into 3 categories:

1) Direct calculation where there is an “easy” representation to work with.
   • E.g. for random choices over IQP circuits.

2) Using the theory of k-designs, and their connection to the Haar distribution.
   • Where “hard” circuits are randomly chosen circuits from either universal gate sets or conjugated Clifford circuits.

3) Use measurement/post-selection gadgets to demonstrate anticoncentration on a subset of output qubits.
   • Where the circuits are deterministically chosen and the randomness emerges from the measurement randomness.
Paley-Zygmund inequality

Typically we will want to demonstrate anti concentration on systems of n qubits, in which case we want to show that the following inequality is satisfied:

$$\Pr_{U_x \in F} [||\langle 0^n | U_x | 0^n \rangle ||^2 \geq \alpha \cdot 2^{-n}] \geq \beta$$

One method doing this is via the Paley-Zygmund inequality ($R > 0, 0 < \alpha < 1$):

$$\Pr[R \geq \alpha \mathbb{E}[R]] \geq (1 - \alpha)^2 \frac{\mathbb{E}[R]^2}{\mathbb{E}[R^2]}$$

Where $R = |\langle 0 | U_x | 0 \rangle|^2$ and the expectation is over the choice of $U_x$ in $F$. The challenge is to bound the moments of $R$ with respect to this choice.

Interestingly, it isn’t too hard to see that if this choice is with respect to the Haar measure on $SU(N)$ then we see anticoncentration directly.
k-designs and anticoncentration

If $U_x$ is drawn from the Haar measure on $U(2^n)$ (or equivalently the Porter-Thomas distribution) then $\mathbb{E}[|\langle 0 | U_x | 0 \rangle|^2] = 2^{-n}$ and $\mathbb{E}[|\langle 0 | U_x | 0 \rangle|^{4}] = \frac{2}{2^n(2^n+1)}$. Substituting this directly into Paley-Zygmund:

$$\Pr[|\langle 0 | U_x | 0 \rangle|^2 \geq \alpha 2^{-n}] \geq \frac{(1 - \alpha)^2}{2}$$

Hence the Haar measure anticoncentrates.

Typical circuits on the Haar measure have exponential depth, hence they cannot generally be made – however for our purposes we only need agreement with the Haar measure up to the 2nd moment of the output distributions.

This can be achieved with either an exact, and it turns out, an approximate unitary 2-design.
Approximate k-designs and anticoncentration

Theorem (see Hangleiter et al 1706.03786v3 and Mann and Bremner ‘17):

Let $U$ be drawn from an $\epsilon$ relative approximate k-design on the group $U(N)$ then matrix elements of $U$ anticoncentrate:

$$\Pr[|\langle 0|U_x|0\rangle|^2 \geq \alpha (1 - \epsilon) 2^{-n}] \geq \frac{(1 - \alpha)^2 (1 - \epsilon)^2}{2(1 + \epsilon)}$$

Furthermore, this can be done in depth $O(n \log 1/\epsilon)$. This follows from the results of Bradao, Harrow, and Horodecki ‘16. This was recently improved by Harrow and Mehraban to $O(n^{1/2} + \log 1/\epsilon)$ for the case of 2D lattices.

Similar results have also been obtained with respect to “conjugated Clifford circuits”, see Bouland, Fitzsimmons, and Koh 1709.01805.
If the “average case” complexity of relative error approximations to either:

1) The complex temperature Ising model partition functions, or
2) The gap of degree 3 polynomials*

is \#P-hard, then quantum computers cannot be efficiently classically simulated to within constant variation distance without a collapse of the PH. (BMS, Phys. Rev. Lett. 117, 080501 (2016), arXiv:1504.07999)

This has been improved to sparse Ising models with \(O(n \log n)\) interactions. (BMS Quantum 1, 8 (2017), arXiv:1610.01808).

For a graph \(G = (V, E)\) with edge weights \(\{\omega_e \in [k\pi/8]\}_{e \in E}\) for \(k \in \{0,7\}\) and vertex weights \(\{\nu_v \in [l\pi/4]\}_{v \in V}\) for \(l \in \{0,3\}\) define the Hamiltonian:

\[
H_G := - \sum_{\{u,v\} \in E} \omega_{\{u,v\}} \sigma_u \sigma_v - \sum_{v \in V} \nu_v \sigma_v .
\]

Then:

\[
\langle 0^{|V|} |U_G |0^{|V|} \rangle = \frac{1}{2^{|V|}} Z_{\text{Ising}}(G; i\Omega, i\Upsilon).
\]
Low-depth quantum circuit sampling where gates are randomly drawn from CZ, T, X\(^{1/2}\), Y\(^{1/2}\) up to at least depth \(O(n^{1/2})\).

Classical hardness is based on the following assumptions:

1. That the approximate average case complexity of the (complex) partition function of quasi 3d Ising models is as hard as the approximate worst-case complexity.

2. That the output probabilities of these probability distributions “anticoncentrate” at this depth. Recently proven by Harrow and Mehraban 1809.06957.

Worst-case approximation complexity for Ising models

Assume a maximum degree $\Delta$ graph.

Approximations:

- [JS93] Jerrum and Sinclair (FPRAS).
- [BS17] Barvinok and Soberón (FQPTAS).
- [PR17] Patel and Regts (FPTAS).
- [MB18] This talk (FPTAS) (with field).

Hardness:

- [MB] Mann and Bremner arXiv:1806.11282 and Goldberg and Guo 1409.5627

- Simulation on hexagonal lattices up to $\pi/23$ rotations and square lattice up to $\pi/29$ rotations. Importantly, this holds for graphs with growing treewidth, for example the square lattice.
Which circuit families are hard?

Post-selection constructions can be used to show that complexity can be inherited amongst difference circuit families. These are usually derived from identities used in measurement based quantum computing.

This is done by showing that the complexity of relative error approximation $|A-P(0n)| \leq \gamma P(0n)$, is preserved.

These techniques can be used to show that randomness properties can be inherited as well.
<table>
<thead>
<tr>
<th>Category</th>
<th>#P-hard worst case/ multiplicative</th>
<th>“anti-concentration”</th>
<th>#P-hard exact average case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantum circuits</td>
<td>Y</td>
<td>?</td>
<td>Y (Fefferman and Umans 1507.05592)</td>
</tr>
<tr>
<td>Boson Sampling</td>
<td>Y (AA ‘10)</td>
<td>?</td>
<td>Y (AA ‘10)</td>
</tr>
<tr>
<td>IQP Sampling, complete graph $O(n^2)$ gates</td>
<td>Y (BMS 1504.07999)</td>
<td>Y (BMS 1504.07999)</td>
<td></td>
</tr>
<tr>
<td>IQP Sampling, sparse graph $O(n \log n)$</td>
<td>Y</td>
<td>Y (BMS 1610.01808)</td>
<td></td>
</tr>
<tr>
<td>IQP Sampling, complete graph $O(n \log n)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IQP Sampling, sparse graph $O(n \log n)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O(1) depth n.n. Universal gates</td>
<td>Y (Terhal and DiVincenzo, quant-ph/0205133)</td>
<td>Y (Gao et al 1607.04947, Bermejo-Vega et al 1703.00466, Miller et al 1703.11002)</td>
<td>?</td>
</tr>
<tr>
<td>O(1) depth n.n. IQP</td>
<td>Y (BJS 1005.1407, Goldberg and Guo 1409.5627)</td>
<td>Y? (see box above)</td>
<td>?</td>
</tr>
<tr>
<td>2d n.n. Random circuits from Porter-Thomas/t-designs</td>
<td>Y, Boixo et al 1608.00263, Mann and Bremner 1711.00686</td>
<td>Y $O(n)$ depth Hangleiter 1706.03786, Mann and Bremner 1711.00686, $Y O(n^{1/2})$, (Harrow and Mehraban 1809.06597),</td>
<td>Y Bouland et al 1803.04402</td>
</tr>
<tr>
<td>Commuting 2-local gates</td>
<td>Y (Bouland, Mašínska, and Zhang, QIP’16)</td>
<td>Y, ?</td>
<td></td>
</tr>
<tr>
<td>Clifford circuits</td>
<td>N</td>
<td>Y, ? (Brown and Fawzi, 1307.0632)</td>
<td>N</td>
</tr>
<tr>
<td>Clifford with product input</td>
<td>Y (Jozsa and Van Nest, 1305.6190/Koh 1512.07892), Bouland 1709.01805</td>
<td>Y, ? (Brown and Fawzi, 1307.0632)</td>
<td>Y (Bouland 1709.01805)</td>
</tr>
<tr>
<td>Model</td>
<td>#P-hard worst case/ multiplicative</td>
<td>“anti-concentration”</td>
<td>#P-hard exact average case</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------</td>
<td>----------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Quantum circuits</td>
<td>Y</td>
<td>?</td>
<td>Y (Fefferman and Umans 1507.05592)</td>
</tr>
<tr>
<td>Boson Sampling</td>
<td>Y (AA ‘10)</td>
<td>?</td>
<td>Y (AA ‘10)</td>
</tr>
<tr>
<td>IQP Sampling, complete graph O(n^2) gates</td>
<td>Y (BMS 1504.07999)</td>
<td>Y (BMS 1504.07999)</td>
<td>?</td>
</tr>
<tr>
<td>IQP Sampling, sparse graph O(Vn log n) depth n.n. gates - optimal</td>
<td>Y (Terhal and DiVincenzo, quant-ph/0205133)</td>
<td>Y? (Gan et al 1607.04947, Bermejo-Vega et al 1703.00466, Miller et al 1703.11002)</td>
<td>?</td>
</tr>
<tr>
<td>O(1) depth n.n. Universal gates</td>
<td>Y (Terhal and DiVincenzo, quant-ph/0205133)</td>
<td>Y? (BJS 1005.1407, Goldberg and Guo 1409.5627)</td>
<td>?</td>
</tr>
<tr>
<td>O(1) depth n.n. IQP</td>
<td>Y</td>
<td>Y? (see box above)</td>
<td>?</td>
</tr>
<tr>
<td>O(Vn log^n) depth n.n. Haar random gates</td>
<td>Y, ? (From above)</td>
<td>Y, ? (Brown and Fawzi, 1307.0632)</td>
<td>?</td>
</tr>
<tr>
<td>Random circuits from Porter-Thomas/t-designs</td>
<td>Y, Boixo et al 1608.00263, Mann and Bremner 1711.00686</td>
<td>Y O(n) depth Hangleiter 1706.03786, Mann and Bremner 1711.00686 , ? O(n^{1/2}) (Boixo et al 1608.00263),</td>
<td>?</td>
</tr>
<tr>
<td>Commuting 2-local gates</td>
<td>Y (Bouland, Maňcinska, and Zhang, QI’16)</td>
<td>Y, ?</td>
<td>?</td>
</tr>
<tr>
<td>Clifford circuits</td>
<td>N</td>
<td>Y, ? (Brown and Fawzi, 1307.0632)</td>
<td>N</td>
</tr>
<tr>
<td>Clifford with product input</td>
<td>Y (Jozsa and Van Nest, 1305.6190/Koh 1512.07892), Bouland 1709.01805</td>
<td>Y, ? (Brown and Fawzi, 1307.0632)</td>
<td>?</td>
</tr>
</tbody>
</table>

None of these models have made any progress on the #P-hard with relative error in the average case.

Any proof of this is likely to require non-relativising techniques – see Aaronson and Chen arXiv:1612.05903.
Time-space tradeoffs

Measurement randomness can be used to induce RCS on subsets of qubits from non-random circuit families.

Gao et al 1607.04947, Bermejo-Vega et al 1703.00466, Miller et al 1703.11002 used constructions from measurement-based quantum computing show that constant-depth, non-random, circuits cannot be classically efficiently sampled.
Tensor network algorithms can compute $p = |\langle 0^n | U | 0^n \rangle|^2$ in time $\min(O(2^n), O(2^{dD}))$, where $d$ is depth and $D$ is the diameter of a $2d$ nearest-neighbour circuit.

E.g.: If there were a sub $d = O(n^{1/2})$ circuit for sparse IQP circuits, $p$ could be computed in sub exponential time – contradicting the fact that sparse Tutte polynomials/Ising models are hard for the exponential time hypothesis.

Hence depth bound of $d = O(n^{1/2} \log n)$ is essentially optimal for IQP circuits. See Bremner, Montanaro, Shepherd 1610.01808.

* Note that the Google proposal (arXiv:1608.00263) scales larger than $d = O(n^{1/2})$.
  - See also Huang, Newman, and Szegedy 1804.10368.
Time-space tradeoffs

If the circuit depth in a 2D architecture is sub $n^{1/2}$ then we know that typical circuits will not be solving “the hardest” instances of #P-hard problems (scaling like $n$) without a violation of the exponential time hypothesis.

Below $n^{1/2}$ depth we begin to trade qubits vs gates. For example if we go to constant depth we must have a polynomial increase in the number of qubits.

E.g. see Bremner, Montanaro, Shepherd 1610.01808 and Huang, Newman, and Szegedy 1804.10368.
Where is the quantum frontier?

Cannot be answered with asymptotic arguments alone – the question depends on the amount of system noise, the model, and the best classical simulation algorithms.

Tensor network methods seem to be the state-of-the-art for random circuit simulations. Best methods utilize symmetries in circuits or forced by a noise model to reduce complexity.

- “Around 50 qubits at depth 40" for the Google model. See Boixo et al 1712.0538, and Haener and Steiger 1704.01127.

- Arguably more than 50 qubits 1811.09599, 1807.10749, 1805.0145.

- Around 60-70 qubits for sparse IQP (Google paper, Nature Physics) and 90 qubit upper bound (arXiv:1805.05224).

- Above 50 photons for BosonSampling based on numerical testing with realistic loss parameters. see Neville et al 1705.00686, and Clifford2 1706.01260.
## Sycamore vs Summit

<table>
<thead>
<tr>
<th></th>
<th>Sycamore</th>
<th>Summit (qFlex)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time (20 cycles)</strong></td>
<td>200s</td>
<td>10,000 years</td>
</tr>
<tr>
<td><strong>Energy (14 cycles)</strong></td>
<td>1kWh</td>
<td>10,000 MWh</td>
</tr>
</tbody>
</table>

**Images:**
Where is the quantum frontier?

2019: Google performed random circuit sampling with a 53 qubit device. Claims that the simulation would take 10,000 years with a conventional supercomputer.

2019: IBM argues it takes 2.5 days on Summit with a different algorithm.
**Noisy IQP—BMS 1610.01808**

Such noisy circuits are well outside the constant 1-norm additive error scenario - the $l_1$ distance grows like $O(n)$ for this error model.

In the regime with a quantum advantage $\epsilon \leq n^{-1}$ and simulation runtime is exponential.

This algorithms works for any anticoncentrated IQP circuit, e.g. Simon’s problem.

This can be extended to the Google model via postselection (1706.08913, 1708.01875).

*Nothing on this plot is to scale!* 

---

**Graph Elements:**
- **Gate depth/runtime**
- **Max. digital runtime**
- **Max. quantum runtime**

**Equations:**
- Noisy IQP runtime: $n^{O(\log(\alpha/\delta)/\epsilon)}$
- $\epsilon$ – per qubit depolarizing rate
- $\delta$ – total variation distance error
- $\alpha$ – anticoncentration measure

**Legend:**
- Classical
- Quantum
Fidelity vs runntime

• The noisy simulation runtime is $n^{O\left(\log(\alpha/\delta)/\varepsilon\right)}$ which comes from the number of terms required to be computed to have $\|p_n - q_n\| \leq \delta$.

• The distance between $\|p - p_n\| \sim n\varepsilon$, and so to get within a “constant” total variation distance $\varepsilon < 1/n$.

• This leads to a runtime of $O(n^n)$, considerably more than the exact simulation runtime $O(2^n)$.

• Experimental “goal” is to have gate/qubit error rates below $O(1/(n+m))$ - in Google’s case something like this can be achieved if there is an upper bound of 50 qubits and depth 40.

• More generally this requires error correction!

• Boixo et al (1708.01875) go further than this to demonstrate that the uniform distribution is typically as “close” to $p$ as $p_n$ if epsilon does not scale like $n^{-1}$.

• There is a middle ground where you can fix a target fidelity and get some improvement in computational using tensor network methods, see 1807.10749, 1811.09599.
Semi-classical error correction

Measurement depolarizing noise can be corrected on any IQP circuit via a classical encoding, \( M \), on the binary circuit description, \( C \).

This results in a corrected circuit with 1-norm distance \( \delta \) for per-qubit error rate \( \epsilon < 1 \).

(BMS Quantum 1, 8 (2017), arXiv:1610.01808)
Semi-classical error correction

Measurement depolarizing noise can be corrected on any IQP circuit via a classical encoding, $M$, on the binary circuit description, $C$.

This results in a corrected circuit with 1-norm distance $\delta$ for per-qubit error rate $\epsilon < 1$.

(BMS Quantum 1, 8 (2017), arXiv:1610.01808)

$$D = e^{i \sum_{j=1}^{\ell} \theta_j \prod_{k=1}^{n} Z_j^{C_{jk}}}$$

$$D_M = e^{i \sum_{j=1}^{\ell} \theta_j \prod_{k=1}^{n} Z_j^{(CM)_{jk}}}$$

$$\langle x | \prod_{k=1}^{m} Z_j^{(CM)_{jk}} | x \rangle = \langle Mx | \prod_{k=1}^{n} Z_j^{C_{jk}} | Mx \rangle$$

Any “good” code will work. For the bitflip code $D_M$ is an $O(\log n)$ factor larger than $D$. Shannon’s noiseless coding theorem implies that a constant overhead is possible.

$D_M$ is potentially much more complicated than $D$, with the potential for multi-qubit gates.
Certification

Verifying the output distribution is a particular \( P(x) \) is exponentially hard for RCS. However, The appropriate question is whether or not a given circuit was implemented and if it was too noisy not what \( P(x) \) is.

Given a quantum “verifier” we can do this via a blind quantum computing scheme (see Broadbent, Fitzsimons, and Kashefi ‘08).

Recently Urmila Mahadev in arXiv:1804.01082 showed that this can be done with a classical verifier on the assumption that quantum computers cannot break the “Learning With Errors” cryptosystem.

However, it comes with a pretty big cost in the number of qubits.
Classical certification

In general, complete black box verification is likely too hard for RCS. However:

- Physical verification need not be “black box”, we are free to implement many circuits on a device to gain confidence that it works.

- Cross entropy benchmarking and linear cross entropy benchmarking for sufficiently random circuits (Boixo et al 1608.00263 and Arute et al 1910.11333) - requires computing a very hard problems.

- IQP circuits can be partially verified in certain architectures (see Bermejo-Vega et al 1703.00466).

- Aaronson and Chen arXiv:1612.05903 , Heavy output generation: Given as input a random quantum circuit C (drawn from some suitable ensemble), generate output strings \(x_1, \ldots, x_k\), at least a 2/3 fraction of which have greater than the median probability in C’s output distribution. Assuming the QUATH assumption is true. Again requires computing the median, which is generally computationally difficult.

- IQP samplers can be tested with pseudo-random circuits, assuming cryptographic assumptions (Shepherd and MJB, Proc. R. Soc. A 465, 1413-1439 (2009), arXiv:0809.0847). Recently the underlying cryptographic scheme for this problem was broken!
Can RCS-style computations lead to practical applications?

- Local measurement statistics of IQP circuits can be efficiently simulated (BJS 1005.1407, Shepherd 1005.1744).

- Haar random, and highly entangling, circuits do not have any advantage given polytime classical post-processing of local measurements due to exponential measure concentration in Hilbert spaces (BMW 0812.3001, and Gross, Flammia, Eisert 0810.4331).

- 2-designs create a “barren plateau” for optimizations dependent on expectation functions (McClean et al 1803.11173).

- Do “pseudo-random” structures yield an advantage? How non-random do practically useful circuits look?
A potential quantum (near) future

Noisy Intermediate-Scale Quantum computing (NISQ), application development for
- chemistry
- materials
- finance

Unambiguous quantum computational supremacy and commercially relevant applications, 2 – 10 years

Universal quantum computing...

Can You Make A Quantum Computer Live Up To The Hype? Then Rigetti Computing Has $1 Million For You

Airbus Quantum Computing Challenge

Bringing flight physics into the Quantum Era

How can a quantum computer be better than a classical computer?

Are there any quantum applications with legitimate value that are:

• faster,
• more accurate, and/or
• more cost effective

than on classical computers?

Can any of these criteria be satisfied in the near term?
Open questions

• Can we use these techniques to “prove” advantage over classical algorithms for more “more structured” problems, such as for optimization (e.g. QAOA see 1703.06199 and 1602.07674) or for general quantum simulations? A: be wary of this McClean et al 1803.11173

• Can these techniques be used to hunt for a quantum advantage in Machine Learning, see Gao, Zhang, and Duan 1711.02038.

• How much noise is too much noise? When is error correction a necessity? Can we make do without it?

• How many qubits are required to outperform classical computers for other tasks with a quantum advantage? E.g. we know that 512 bit RSA needs 1026 logical qubits or 256 bit elliptic curves require 2330 (1706.06752).

• How do we attack the open average-case complexity conjectures? What other conjectures can be made?

Thank you

For easy to read introductions see:


“Quantum supremacy: the gloves are off” by Scott Aaronson in his blog “Shtetl-optimized” 23 October 2019.

“Quantum computational supremacy” by Aram Harrow and Ashley Montanaro 

Any questions come see me at the UTS stand today or send me an email: 
Michael.Bremner@uts.edu.au