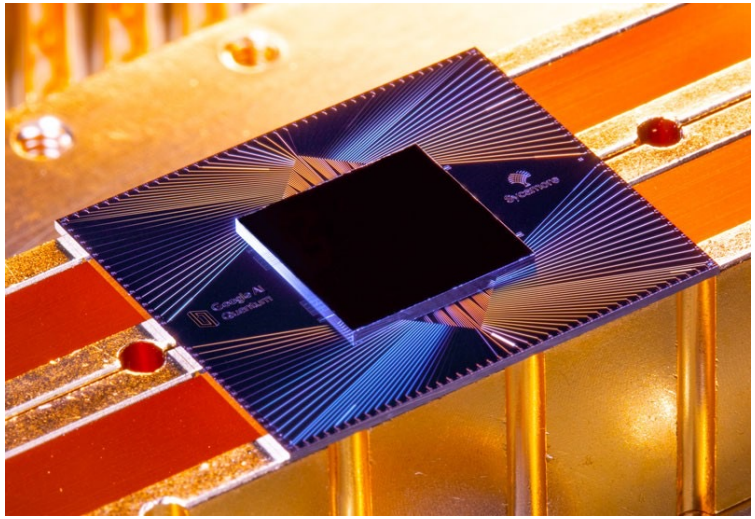
On the theory of near-term quantum advantage

Bill Fefferman

IAS/PCMI Graduate Summer School Lectures, Summer 2023

The first “Quantum advantage” claims have now been made…



Random Circuit Sampling (Google “Sycamore”) in late 2019, USTC in 2021, Google’s second experiment in 2023…

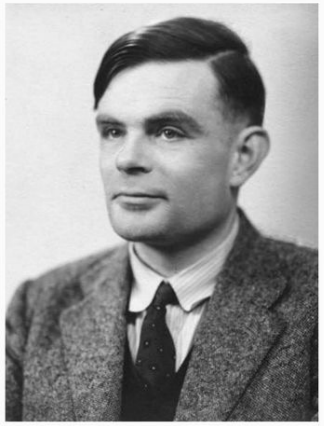
Gaussian BosonSampling – e.g., USTC “Jiuzhang” in 2020,2021,2022, Xanadu’s “Borealis” in 2022…

These lectures: the latest complexity theoretic arguments & classical algorithms to understand the power of these “random quantum circuit” experiments

Importance of experimental quantum advantage: *foundations of computation*

- *Experimental* violation of the Extended Church-Turing thesis
  - i.e., If we want to model efficient computation, we must consider quantum mechanics!
- Complements *theoretical* evidence given by earlier speedups (e.g., [Bernstein-Vazirani '93][Simon'94][Shor '94])

Alonzo Church



Alan Turing

Importance of experimental quantum advantage: *validating quantum physics*

- Exponential growth one of the most counter-intuitive aspect of quantum mechanics.
  - Is the exponential description of a quantum state really necessary?
- New limit in which to test physics: high complexity.
- *Difficulty*: how to verify something that's exponentially complex?

What is the *ideal* goal of quantum advantage?

- Find a problem:

1. Can be solved efficiently using a near-term quantum experiment
2. Is classically hard to solve – can't be solved in polynomial time with a classical computer as the system size scales
3. Solution can be efficiently verified with a classical computer with minimal trust in the experiment

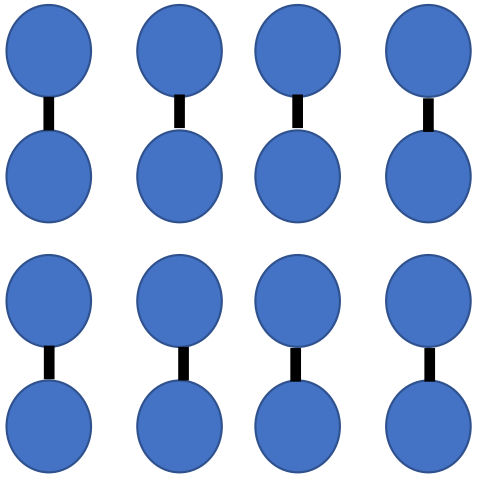
What is the *current* goal of quantum advantage?

- Current quantum advantage experiments solve "sampling problems" in which the goal is to sample from a complicated distribution
- We have rigorous *evidence* that these problems cannot be solved classically in polynomial time
- But current experiments are ***not*** scalable!
  1. Require exponential time to verify
  2. Uncorrected noise gets worse as system size grows
- So hope is to find a ***"Goldilocks"*** system size:
  - Large enough to be classically challenging to simulate
  - *Not too large!* Otherwise effects of noise overwhelm and the experiment can't be verified
- There is optimism that current experiments have reached this size, **but classical simulation algorithms continually improve, as do quantum experiments.**
- ***Much is still unknown!!!!***

Goldilocks and the three bears

What is Random Circuit Sampling? [e.g., Boixo et. al. 2017]

- Generate a quantum circuit C on n qubits on a 2D lattice, with d layers of (Haar) random nearest-neighbor gates
  - In practice use a discrete approximation to the Haar random distribution
- Start with $|0^n\rangle$ input state, apply random quantum circuit and measure all qubits in computational basis
  - i.e., Sample from a distribution D_C over $\{0,1\}^n$
- Has now been implemented:
  - n = 53 qubits, d = 20 [Google, 2019]
  - n = 60 qubits, d = 24 [USTC, 2021]
  - n = 70 qubits, d = 24 [Google, 2023]
- **This will be the focus of these talks!**



(single layer of Haar random two qubit gates applied on 2D grid of qubits)

Boson Sampling [Aaronson & Arkhipov ‘11]

- Prepare n photon $m \geq n^2$-mode “Fock” state
  - i.e., n identical single photons in the first of m modes
- Evolve under a Haar random linear optical unitary composed of beamsplitters and phaseshifters
- Take photon number resolving measurements in each mode
- Recent experiments use similar idea with *Gaussian* input states, rather than Fock states – called “Gaussian BosonSampling”
  - Implemented with 144 modes and as many as 113 detected photons by USTC ’21
  - Implemented with as 216 modes and as many as 219 photons by Xanadu ‘22

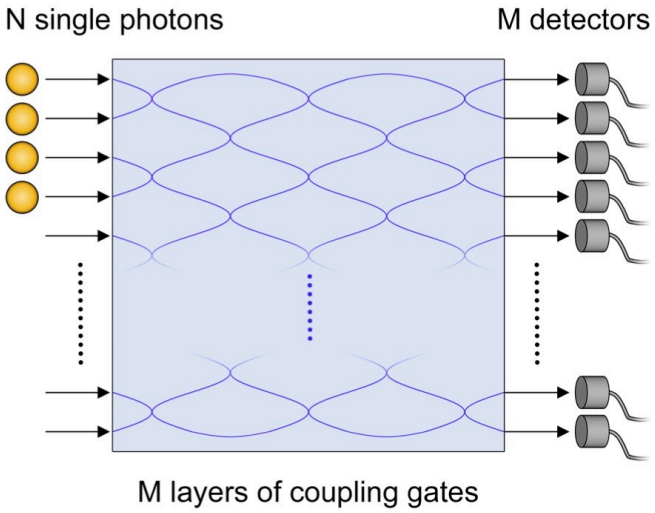


Photo credit: R. Garcia-Patron, J. Renema and V. Shchesnovich

Agenda

1. **Hardness argument 1** (hardness of quantum sampling)
2. **Hardness argument 2** (hardness of benchmarks)
3. **Easiness argument 1** (classical algorithm for the “XQUATH” benchmark)
4. **Easiness argument 2** (classical algorithms taking advantage of uncorrected noise)

2. Hardness argument 1 (hardness of *worst-case* quantum circuit sampling)

What do we mean by *quantum sampling*?

- Current quantum advantage experiments sample from the output distribution of a quantum circuit
  - i.e., on input C the experiment runs $C|0^n\rangle$ and measures all n qubits in computational basis to get a sample $y \in \{0,1\}^n$
- **Definition:** Let the "output probability" $p_y(C) = |\langle y|C|0^n\rangle|^2$
- **First goal:** prove impossibility of an efficient **classical sampler algorithm** S that samples from the same distribution:
  - for all C, y we have $\Pr_r[S(C,r) = y] = p_y(C)$

Starting point: on "classical" vs "quantum" sum

- Consider two problems:
  - "**Classical" sum**: Given classical circuit computing $f: \{0,1\}^n \to \{0,1\}$ compute $\sum_{x \in \{0,1\}^n} f(x)$
  - "**Quantum" sum**: Given classical circuit computing $g: \{0,1\}^n \to \{\pm 1\}$ compute $\sum_{x \in \{0,1\}^n} g(x)$
- Both are **#P**-hard to exactly compute, since they are at least as hard as counting the number of satisfying assignments to a Boolean formula

On *classical approximate* sum

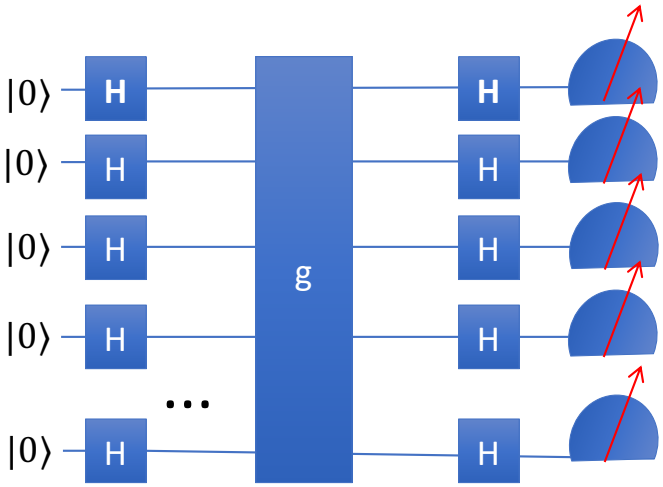
- **Classical "approximate sum":** Given $f:\{0,1\}^n \to \{0,1\}$ output *multiplicative estimate* α so that:
  - $(1-\epsilon)\sum_{x\in\{0,1\}^n} f(x) \leq \alpha \leq (1+\epsilon)\sum_{x\in\{0,1\}^n} f(x)$
- **Stockmeyer's algorithm: classical approximate sum** can be solved in classical $poly\left(n, \frac{1}{\epsilon}\right)$ time with an **NP** oracle [Stockmeyer'85]
  - In particular, it's *strictly easier* than exact case, unless **PH** collapses
- **Consequence 1:** If a classical sampler S exists, then outputting a *multiplicative estimate* of probability for any outcome y is *strictly easier than* **#P**
  - Because output probability is a classical sum problem!
  - i.e., define f(r)=1 if $S(C,r)=y$ and otherwise 0
  - Then $\Pr_r[S(C,r)=y] = \frac{1}{2^{|r|}}\sum_r f(r)$

On *quantum approximate sum*

- **Quantum "approximate sum":** Given $g: \{0,1\}^n \to \{\pm 1\}$ output *multiplicative estimate* α so that:
  - $(1-\epsilon)\sum_{x\in\{0,1\}^n} g(x) \leq \alpha \leq (1+\epsilon)\sum_{x\in\{0,1\}^n} g(x)$
- **Claim:** Unlike the classical problem this is as hard as computing $\sum_x g(x)$ exactly!
- **Intuition:** Exponential size cancellations ("interference") make this problem much harder than **classical approximate sum**!
- **Pf sketch:** "binary search and padding"
  - **Claim:** even computing sign($\sum_x g(x)$) is **#P**-hard (and is a strictly easier problem!)
  1. "Padding": By adding dummy variables can compute g' so that $\sum_{x'} g'(x') = \sum_x g(x) - k$
  2. Then compute sign i.e., is $(\sum_{x'} g'(x)) > 0$?
     - Then we know if $\sum_x g(x) > k$
  3. Then binary search on k and repeat!
- **Exercise:** Similar argument proves it's **#P**-hard to estimate $(\sum_x g(x))^2$
  - i.e., can run the same binary search & padding argument on $|\sum_x g(x)|$

Consequence 2: estimating the output probability of quantum circuits is #P-hard

- **Claim:** given quantum circuit C estimating $p_{0^n}(C)$ is as hard as ***squared quantum approximate sum***.
- **Pf:** By "quantum Fourier sampling"
  - Given $g: \{0,1\}^n \to \{\pm 1\}$ consider the quantum circuit C that:
    - Prepares the state $|g\rangle = \sum_x g(x)|x\rangle$ then takes the Hadamard of each qubit
    - Notice that $p_{0^n}(C) = \left|\langle 0^n | H^{\otimes n} | g \rangle\right|^2 = \frac{(\sum_x g(x))^2}{2^{2n}}$
  - So multiplicative estimation of $p_{0^n}(C)$ is #P-hard

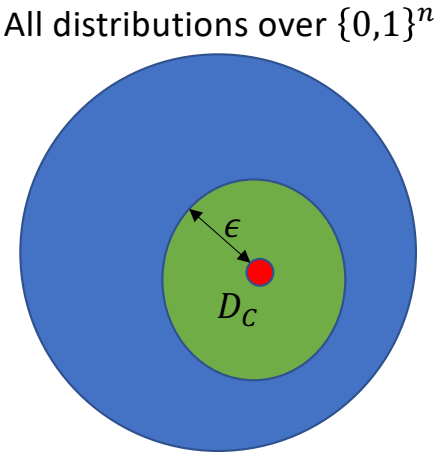


Impossibility of *exact* sampling

- Assume, for contradiction, there is an efficient sampler S:
  - This means for any quantum circuit C:
    - $\Pr_r[S(C,r) = y] = |\langle y|C|0^n\rangle|^2 = p_y(C)$
- By consequence 1 we know that estimating the probability S outputs $0^n = p_{0^n}$, is *strictly easier* than **#P** (unless **PH** collapses)
- But by consequence 2 know that estimating $p_{0^n}(C)$ is **#P**-hard, since it is *as hard* as ***squared quantum approximate sum***
- This is a contradiction! So there can't be such a sampler algorithm.
- Similar arguments appear in [Terhal-DiVincenzo '04, Bremner-Jozsa-Shepherd '11, Aaronson-Arkhipov '11…]

This result is not *robust*

- The impossibility result has two major weaknesses:
  1. **Exactness assumption:** It requires that the classical algorithm samples *exactly* from the output distribution of each quantum circuit
  2. **Worst-case assumption:** It requires that the classical algorithm works *for all* quantum circuits
- **Major goal in the theory of quantum advantage:** prove impossibility of *approximate average-case* sampler
  - i.e., efficient classical algorithm $S(C, r)$ that samples from any distribution $|X - D_C|_{TV} \leq \epsilon$ whp over C
- **Note:** constant approximation in TVD is not intended to model *physical noise* but rather *classical imprecision*!

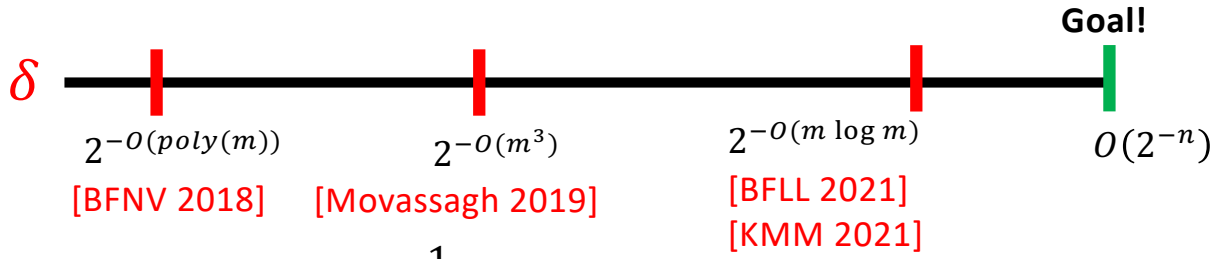


Proving hardness of *approximate* sampling

- **Central problem** of study: δ-**random circuit estimation**:

> Given as input quantum circuit C, output q so that $|q - p_{0^n}(C)| \leq \delta$ with probability 2/3 over C

- To prove hardness of *average-case approximate* sampling suffices to prove $\boldsymbol{\delta = O(2^{-n})}$ **random circuit estimation** is **#P**-hard [Stockmeyer '85][Aaronson Arkhipov '11]
- **Known hardness results** with respect to C on n qubits, size $m = O(n \cdot d)$



- **Boson Sampling**: goal is $\frac{1}{e^{n \log n}}$, whereas we have hardness at $\frac{1}{e^{6n \log n}}$ [BFLL'21]

Inspiration: average-case hardness of Permanent [Lipton '91]

- **Permanent** of $n \times n$ matrix is **#P**-hard in the worst-case [Valiant '79]
  - $Per[X] = \sum_{\sigma \in S_n} \prod_{i=1}^{n} X_{i,\sigma(i)}$
- *Algebraic property*: $Per[X]$ is a degree n polynomial with n^2 variables
- Need compute $Per[X]$ of worst-case matrix X
  - But we only have access to algorithm O that correctly computes *most* permanents over $\mathbb{F}_p$
  - i.e., $\Pr_{Y \in_R \mathbb{F}_p^{n \times n}}[O(Y) = Per[Y]] \geq 1 - \frac{1}{poly(n)}$
- Choose $n+1$ fixed non-zero points $t_1, t_2 \ldots, t_{n+1} \in \mathbb{F}_p$ and uniformly random matrix R
- Consider line $A(t) = X + tR$
  - *Observation 1 "scrambling property"*: for each i, $A(t_i)$ is a random matrix over $\mathbb{F}_p^{n \times n}$
  - *Observation 2*: "univariate polynomial": $Per[A(t)]$ is a degree n polynomial in t
- But now these $n+1$ points uniquely define the polynomial, so use polynomial extrapolation to evaluate $Per[A(0)] = Per[X]$

[BFNV'18]: Hardness for Random Quantum Circuits

- *Algebraic property*: much like $Per[X]$, output probability of random quantum circuits has polynomial structure
  - Consider circuit $C = C_m C_{m-1} \dots C_1$
  - Polynomial structure comes from path integral:
    - $\langle 0^n|C|0^n\rangle = \sum_{y_2,y_3,\dots,y_m \in \{0,1\}^n} \langle 0^n|C_m|y_m\rangle\langle y_m|C_{m-1}|y_{m-1}\rangle \dots \langle y_2|C_1|0^n\rangle$
- This is a polynomial of degree m in the gate entries of the circuit
- So the output probability $p_{0^n}(C)$ is a polynomial of degree $2m$

First attempt at adapting Lipton's proof

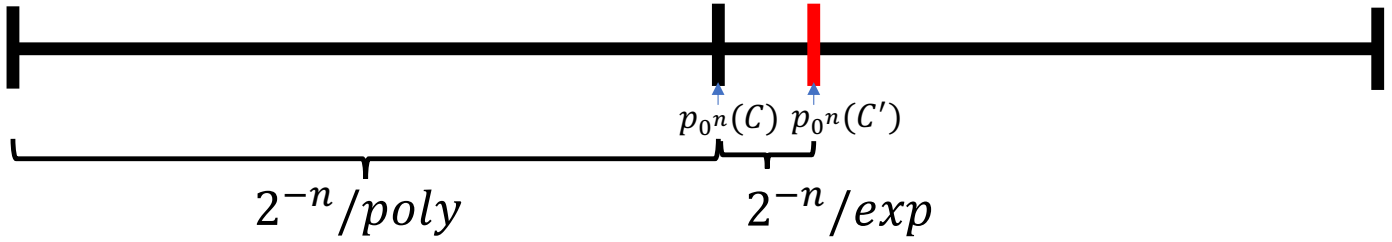
- Fix m Haar random two qubit gates $\{H_i\}_{i\in[m]}$
- ***Main idea*:** Implement tiny fraction of H_i^{-1}
  - i.e., $C_i' = C_i H_i e^{-ih_i\theta}$
  - This scrambles C if $\theta \approx small$, since each gate is close to Haar random
  - However, if $\theta = 1$ the corresponding circuit $C' = C$
- ***Strategy (in style of Lipton):*** take several non-zero but small θ, compute output probabilities of "random but correlated" circuits $C'_{\theta_1}, C'_{\theta_2} \ldots, C'_{\theta_{2m}}$ and apply polynomial extrapolation, evaluate at $\theta = 1$ to retrieve $p_{0^n}(C)$

This is not quite the “right way” to scramble!

- ***Problem*:** $e^{-ih_i\theta}$ is not polynomial in θ
- ***Solution:*** take fixed truncation of Taylor series for $e^{-ih_i\theta}$
  - i.e., each gate of C'_θ is $C_i H_i \sum_{k=0}^{K} \frac{(-ih_i\theta)^k}{k!}$
  - So each gate entry is a polynomial in θ and so is $p_{0^n}(C'_\theta)$
  - Now extrapolate and compute $p(1) = p_{0^n}(C)$

How to motivate the truncations?

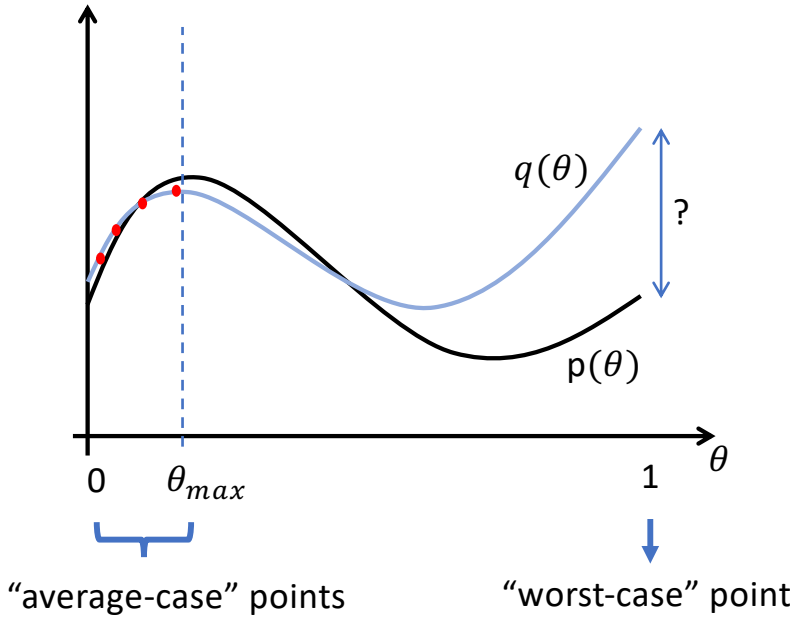
- Main technical result in [BFNV'18]: ***Estimating*** $p_{0^n}(C')$ is hard **iff** ***estimating*** $p_{0^n}(C)$ is hard
  - **Intuitively, because the "truncation error" is so much smaller than the size of the additive error we are conjecturing is hard.**



- More recently, [Movassagh'19'20] has shown a related argument (using the so called "Cayley path") that eliminates the need for these truncations

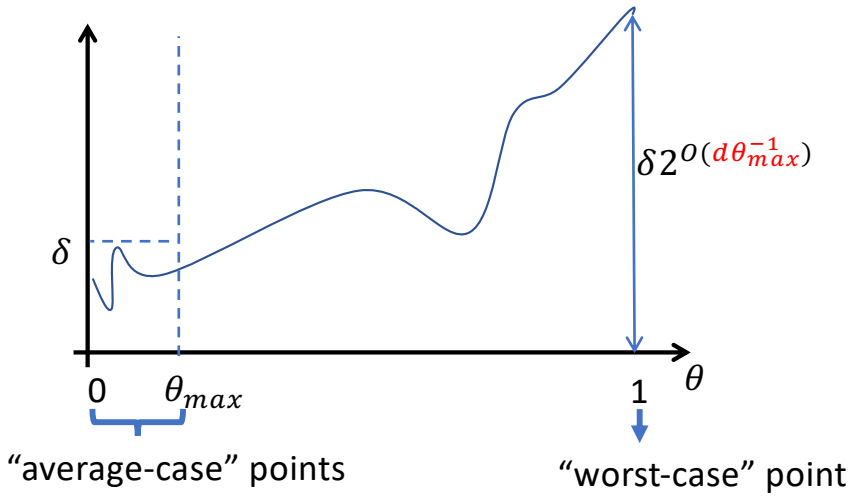
On robustness to *imprecision*

- So far we assumed the ability to compute the output probabilities of random circuits $\{p_{0^n}(C'_{\theta_i})\}$ *exactly*
- **Actual setting:** Given $2m$ evaluation points $\{(\theta_i, y_i)\}$ so that for most i, $\left|y_i - p_{0^n}(C'_{\theta_i})\right| \leq \delta$
- We have two polynomials:
  - The "ideal" $p(\theta_i) = p_{0^n}(C'_{\theta_i})$
  - The extrapolated polynomial $q(\theta_i) = y_i$
- **Our question:** How close is $q(1)$ to $p(1) = p_{0^n}(C)$ in terms of δ, θ_{max} ?



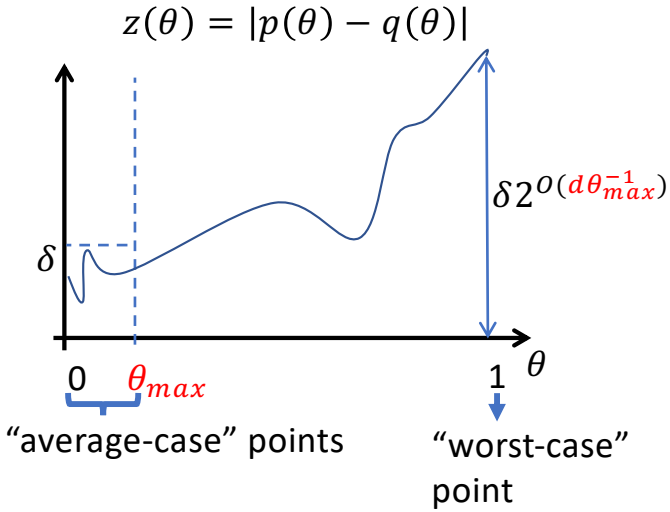
The “Paturi picture”

- [Paturi ’92] If we have a degree d polynomial $z(\theta)$ bounded on an interval $[0, \theta_{max}]$ by δ then $|z(1)| \leq \delta 2^{O(d\theta_{max}^{-1})}$
- **Our case:** Consider the degree $2m$ polynomial $z(\theta) = |p(\theta) - q(\theta)|$



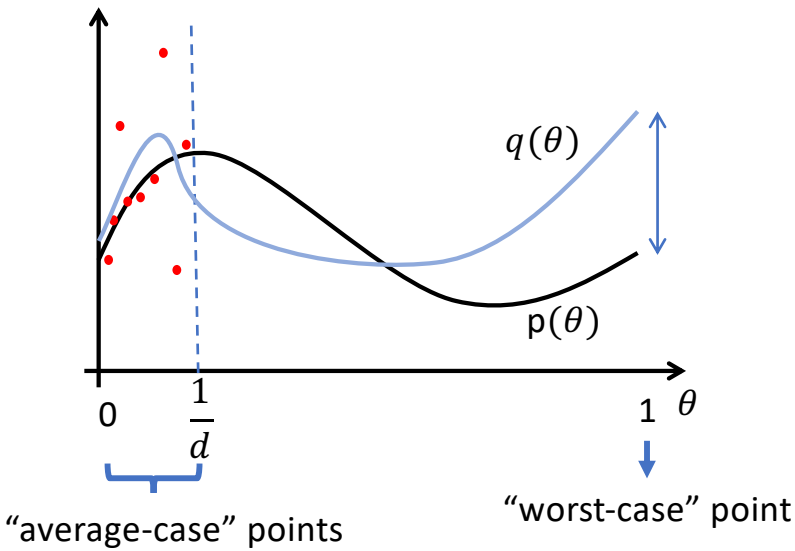
How large can we take θ_{max}?

- Lagrange extrapolation requires getting all $d = 2m$ points correct
  - So we need the algorithm to succeed wp $\geq 1 - O\left(\frac{1}{m}\right)$
- As θ gets larger C'_θ is further away from random circuit
  - i.e., Distribution of C'_θ is $O(m\theta)$-close in TVD from Haar random circuit
  - So algorithm works wp $1 - O(m\theta)$ on these points
- So need $\theta_{max} \leq \frac{1}{O(m^2)}$
- Plugging in Paturi's bound: $z(1) \leq \delta 2^{O\left(d\theta_{max}^{-1}\right)} = \delta 2^{O(m^3)}$
- So need $\delta = \frac{1}{2^{O(m^3)}}$



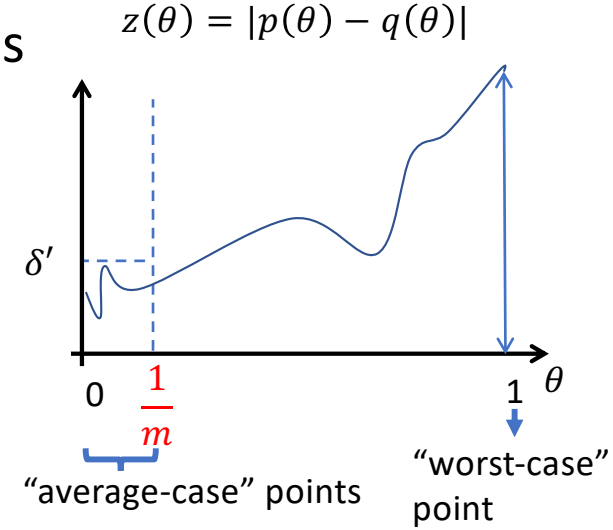
Increasing robustness [BFLL'21] (see also [Kondo et. al.'21])

- To improve imprecision we need a new, error-robust means of polynomial extrapolation
- Will do this by oversampling – i.e., taking many more points than degree
- **"Robust Berlekamp-Welch" Thm.** Given $O(d^2)$ "faulty" evaluation points $\{(\theta_i, y_i)\}$ to $p(\theta)$ of degree d so that:
  1. $\theta_i \in \left[0, \frac{1}{d}\right]$
  2. We know ***at least** 2/3* of y_i are δ***-close*** to $p(\theta_i)$
- Then any polynomial $q(\theta)$ which is δ***-close*** on $2/3$ fraction of the points is $\delta 2^{O(d)}$-close to $p(\theta)$ ***for all*** $\theta \in \left[0, \frac{1}{d}\right]$



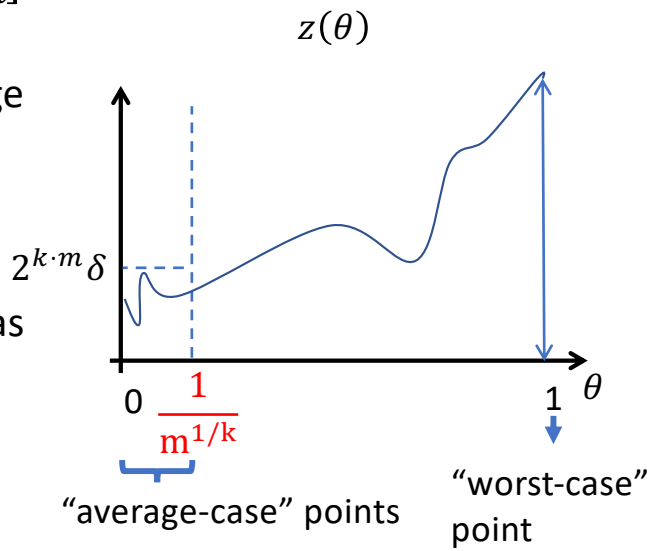
How large can we take θ_{max} now?

- **Input:** faulty points to polynomial $p(\theta)$: $(\theta_1, y_1), (\theta_2, y_2) \ldots (\theta_{O(m^2)}, y_{O(m^2)})$
- Ask **NP** oracle to give us a polynomial $q(\theta)$ that is δ-close to $2/3$ of these points
  - This can easily be checked by evaluating q at each θ_i
- Robust Berlekamp-Welch theorem tells us:
  - $|p(\theta) - q(\theta)| \leq \delta' = \delta 2^{O(m)}$ **for all** $\theta \in \left[0, \frac{1}{m}\right]$
- Then Paturi tells us:
  - $|p(1) - q(1)| = |z(1)| \leq \delta' 2^{O(d\theta_{max}^{-1})} = \delta 2^{O(m^2)}$
  - So we need to take $\delta \sim \frac{1}{2^{O(m^2)}}$



Getting to robustness $2^{-O(m \log m)}$

- Given faulty points $(\theta_1, y_1), (\theta_2, y_2) \ldots (\theta_{O(m^2)}, y_{O(m^2)})$ with $\theta_i \in \left[0, \frac{1}{m}\right]$
- Trick! Rather than asking the **NP** oracle for the approximating polynomial q of degree m, replace the variable θ with θ^k for some large k and ask for this new poly q'
  - This rescaling *increases* the degree to km!
  - But it "stretches" unit interval near 0 and "compresses" near 1
  - So for fixed value of $\theta_{max} = \frac{1}{m}$ the corresponding value of θ_{max} has increased, it's now $\frac{1}{m^{1/k}}$
- Plugging in Paturi's bound: $\mathrm{z}(1) \leq \delta 2^{O(km \cdot m^{1/k})}$
- Setting $k = \log(m)$ we have $\mathrm{z}(1) \leq \delta 2^{O(m \cdot \log(m))}$
- So we need to set $\delta \sim 2^{-O(m \cdot log(m))}$



Comments & Open Directions

- Main open question in the theory of quantum advantage: ***improve*** the ***additive imprecision*** of these average-case hardness results to $O(2^{-n})$ from $2^{-O(m)}$ for RCS or $\frac{1}{e^{n \log n}}$ from $\frac{1}{e^{6n \log n}}$ for Boson Sampling
- Current hardness results have improved dramatically but we've also discovered ***barriers*** implying that new techniques will be needed to improve them further (e.g., [AA'2011][Napp et. al. '19][BFLL'21])

3. Hardness argument 2 (hardness of benchmarks)

Limitations of total variation distance

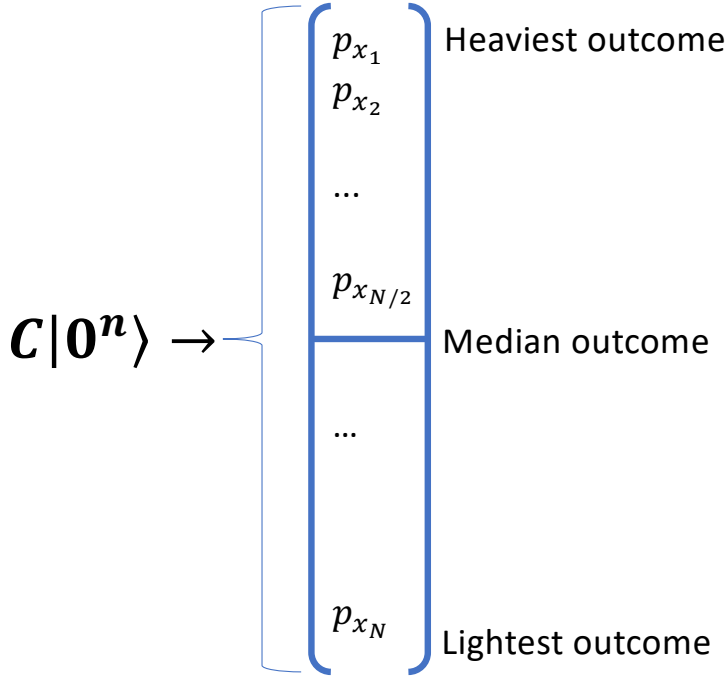
- Total variation distance is difficult to measure!
  - There are well-known exponential lower bounds for sample complexity, even for "merely" testing closeness to the uniform distribution e.g., [Valiant & Valiant'17]
- Closeness in total variation distance is not a reasonable model of uncorrected physical noise
  - i.e., system size increases, having TVD remain a small constant isn't realistic without error mitigation
- Is there a "quantum signal" that is easier to verify and implement?

Candidates for verifiable "quantum signals"

- Many candidates rely on the "Porter-Thomas property" of random quantum circuits
  - Each output probability is exponentially distributed
  - i.e., $\Pr_C\left[|\langle x|C|0^n\rangle|^2 = \frac{q}{2^n}\right] \sim e^{-q}$
  - True for Haar random unitaries
  - Conjectured to be true even for shallow depth random circuits
- This Porter-Thomas property implies that the output distribution of a random but fixed circuit is somewhat "flat" but not uniform whp
- **Observation:** Easy to sample from the output distribution with a quantum computer and observe many "heavy" outcomes – how difficult is this to do classically?

Heavy Output Generation [Aaronson & Chen '17]

- **Definition:** With respect to a circuit C call an outcome $x \in \{0,1\}^n$ *heavy* if p_x is greater than *median* in the output distribution of C
- **HOG**: Given random circuit C output strings $x_1, x_2, \dots, x_k$ so that at least $2/3$ are *heavy*
- **Claim:** Quantumly can solve **HOG** simply by repeatedly running $C|0^n\rangle$ and measuring
  - Why? Because whp over C, the sum of probabilities that are above median in output distribution is ≥ 0.7
    - Using Porter-Thomas property!
  - Then use Chernoff bound to prove 2/3 of outputs are heavy whp



Quantum Threshold Assumption (QUATH)

- **HOG** still seems like a sampling task – why should this be hard classically?
- [Aaronson and Chen'17]: **HOG** is classically hard assuming **QUATH**
- **QUATH:** No efficient classical algorithm takes input random C with $m \gg n$ gates and decides if p_{0^n} is heavy with probability $\frac{1}{2} + \Omega\left(\frac{1}{2^n}\right)$
  - Where probability is over both C and internal randomness of classical algorithm
- **Motivation:** QUATH seems closer to problems we understand, since it involves estimation of p_{0^n}
- Key point is that the bias scales exponentially in n rather than size m
  - Not hard to show classical algorithm with a bias that scales exponentially in m
  - e.g., randomly guessing a small number of Feynman paths and comparing to a threshold

QUATH implies **HOG** is hard

- **Pf. (Intuition):** By contrapositive assume there's an algorithm for HOG. We want to solve QUATH.
  - On input C use HOG algorithm to output list of mostly heavy strings in output distribution of C
  - Output "heavy" if 0^n is on the list.
- **Pf. (More formal analysis):**
  - Easier to consider a uniform outcome $z \in \{0,1\}^n$ rather than the 0^n outcome
    - But it doesn't matter by a property of random circuits called "hiding"
    - i.e., Let C' be the circuit chosen by taking C and appending Pauli X gates to each i-th qubit if $z_i = 1$
    - Notice that new circuit, C', has property that $p_{0^n} = |\langle 0^n|C|0^n\rangle|^2 = |\langle z|C'|0^n\rangle|^2$ and C' is still random circuit
  - Strategy is same as the intuition: use **HOG** algorithm on C' to output list $z_1, \ldots, z_k$ so that $2/3$ of z_i are heavy, then choose uniform element of list, call it z_{i^*}
    - **If $z = z_{i^*}$ output "heavy"**
    - **If $z \neq z_{i^*}$ output "heavy" wp ½, "light" wp ½**
  - The probability this algorithm is correct on heaviness of $p_{0^n}(C)$ is at least:
    - $\Pr[z_{i^*} = z] \cdot \frac{2}{3} + \Pr[z_{i^*} \neq z] \cdot \frac{1}{2} = \frac{1}{2^n} \cdot \frac{2}{3} + (1 - 2^{-n}) \cdot \frac{1}{2} = \frac{1}{2} + \Omega\left(\frac{1}{2^n}\right)$

Linear Cross-Entropy (XEB) [Boixo et. al. ’16] [Arute et. al. ‘19]

- An alternative measure of heaviness is **XEB**:
  - $\mathbf{XEB}\left(p_{exp}, p_{ideal}\right) = 2^n \sum_x \left(p_{exp}(x) p_{ideal}(x)\right) = 2^n E_{x \sim p_{exp}(x)}[p_{ideal}(x)]$
  - If $p_{exp} = p_{ideal}$ then $\mathbf{XEB}\left(p_{exp}, p_{ideal}\right) = 2$ but $\mathbf{XEB}(U, p_{ideal}) = 1$
- *XEB can be **well-approximated in few device samples** via concentration of measure arguments, but **requires exponential time to compute** ideal output probabilities of observed samples*
  - *i.e., observe experimental outcomes $z_1, \ldots, z_k$ and compute $\frac{2^n \sum_i p_{ideal}(z_i)}{k}$*

Why is scoring well on XEB classically hard? [Aaronson & Gunn '19]

- **XHOG** ("Linear Cross Entropy Heavy Output Generation")
  - Given C, output k distinct samples $z_1, z_2, \ldots, z_k$ so that $E_i[|\langle z_i|C|0^n\rangle|^2] \geq \frac{\boldsymbol{b}}{2^n}$
  - Where $b = 1 + \epsilon$
- By repeatedly running a *noiseless* circuit we'd be able to achieve $\boldsymbol{b = 2}$
- *Noise* can cause the experiment to have considerably different values for $\boldsymbol{b}$
  - E.g., Google scores $b = 1.002$ on its 53 qubit RCS experiment
- Still seems like a sampling task – why should this be hard classically?

The XQUATH assumption [Aaronson & Gunn '19]

- **XHOG** is hard assuming **XQUATH**
- **XQUATH**: No efficient classical algorithm, given random C, produces estimate, p, to $p_{0^n} = |\langle 0^n|C|0^n\rangle|^2$ so that:
  - $2^{2n}\left(E_C\left[\left(p_{0^n} - \frac{1}{2^n}\right)^2\right] - E_C[(p_{0^n} - p)^2]\right) = \Omega(2^{-n})$
- i.e., No classical algorithm can achieve a **mean squared error** at estimating an output probability of a random circuit, that's slightly better than the trivial algorithm that always outputs 2^{-n}
- **XQUATH** implies **XHOG** is hard by very similar reduction!
  - i.e., assume there's an **XHOG** algorithm that outputs samples $z_1, z_2, \ldots, z_k$ so that $E_i[p_{z_i}] = \frac{b}{2^n}$ then output $\frac{b}{2^n}$ if 0^n is on the list and else output $\frac{1}{2^n}$

Comments & Open Directions

- This is a very "lossy" reduction! Even scoring well (e.g., constant $b > 1$) on **XHOG** gives rise to $\exp(-n)$ bias for **XQUATH**. Can this be improved?
- Under certain assumptions about the noise, the **XEB** score well-approximates the fidelity of the noisy experiment. Hence it can be useful for benchmarking (see e.g., [Boixo et. al. '17] and our work [Liu et. al. '21][Ware et. al. '23] for more details).

4. Easiness argument 1 (XQUATH is false at sublinear depth) [Gao et. al. ‘21][Aharonov et. al. ‘22]

Revisiting the intuition for XQUATH

- Recall **XQUATH**: No efficient classical algorithm, given random C, produces estimate, p, to $p_{0^n} = |\langle 0^n|C|0^n\rangle|^2$ so that:
  - XScore = $2^{2n}\left(E_C\left[\left(p_{0^n} - \frac{1}{2^n}\right)^2\right] - E_C[(p_{0^n} - p)^2]\right) = \Omega(2^{-n})$
- Intuition is that the best classical algorithm for estimating p_0 for a random circuit $C = C_m C_{m-1} \dots C_1$ is to sample the path integral in the computational basis:
  - $p_{0^n} = \left(\sum_{y_2, y_3, \dots, y_m \in \{0,1\}^n} \langle 0^n|C_m|y_m\rangle\langle y_m|C_{m-1}|y_{m-1}\rangle \dots \langle y_2|C_1|0^n\rangle\right)^2$
  - There are $\gg 2^n$ paths with uniform value in expectation, so it's unclear how to achieve an advantage that scales as 2^{-n}
- **Observation:** Turns out this isn't true! If we consider the path integral in the Pauli basis the values of the paths are highly non-uniform!

Pauli path integrals

- Rather than thinking of quantum circuit as applying unitary gates to vectors, think about it as applying *unitary channels* to *density matrices*
- Denote the normalized Pauli operators $P_n = \left\{\frac{I}{\sqrt{2}}, \frac{X}{\sqrt{2}}, \frac{Y}{\sqrt{2}}, \frac{Z}{\sqrt{2}}\right\}^{\otimes n}$
- Can write an n-qubit density matrix $\rho = \sum_{t \in P_n} \alpha_t \cdot t$ with $\alpha_t = Tr[t\rho]$
- Recall in the "computational basis" path integral we express:
  - $\langle x|U|\psi\rangle = \sum_{y \in \{0,1\}^n} \langle x|U|y\rangle\langle y|\psi\rangle$
- Analogously, in Pauli basis $Tr\left[sU\rho U^\dagger\right] = \sum_{t \in P_n} Tr\left[sUtU^\dagger\right] Tr[t\rho]$
  - We call $Tr\left[sUtU^\dagger\right]$ the "transition amplitude"

Expressing p_x as a Pauli path integral

- Now we can express any output probability as a Pauli path integral, in analogy to what we are accustomed to in the computational basis
- Let $C = C_d C_{d-1} \dots C_1$ where each layer C_i acts on n qubits
- $p_x = |\langle x|C|0^n\rangle|^2$
- $= \sum_{s \in P_n^{d+1}} \mathrm{Tr}(|x\rangle\langle x| s_d) \mathrm{Tr}(s_d C_d s_{d-1} C_d^\dagger) \dots \mathrm{Tr}\left(\mathrm{s}_1 C_1 s_0 C_1^\dagger\right) \mathrm{Tr}(\mathrm{s}_0 |0^\mathrm{n}\rangle\langle 0^n|)$
- $= \sum_{s \in P_n^{d+1}} f(C, s, x)$ *(we define $f(C, s, x)$ as the "value" of path s)*

Two important facts

- The **XQUATH** algorithm relies on two facts which both follow from elementary properties of Haar random gates
- **Fact 1** (e.g., [HL'09]) Let U be a Haar random 2 qubit gate and $p, q \in P_2$,
  - Then $E_U\left[\mathrm{Tr}\left[pUqU^\dagger\right]^2\right] = \begin{cases} 1,\ if\ p = q = \frac{I^{\otimes 2}}{2} \\ 0, if\ p = \frac{I^{\otimes 2}}{2}\ and\ q \neq \frac{I^{\otimes 2}}{2} \\ 0, if p \neq \frac{I^{\otimes 2}}{2}\ and\ q = \frac{I^{\otimes 2}}{2} \\ \frac{1}{15}, otherwise \end{cases}$
- **Fact 2** ("orthogonality of Pauli paths") Let C be a random circuit (with Haar gates) and $s \neq s' \in P_n^{d+1}$ be any two different paths and any $x \in \{0,1\}^n$
  - Then $E_C[f(C,s,x)f(C,s',x)] = 0$

XQUATH algorithm (part 1)

- **Claim:** Given a random circuit C outputting $p = \frac{1}{2^n} + f(C, s^*, 0^n)$ achieves $XScore$ of $\left(\frac{1}{15}\right)^d$ where $s^* = \left(\frac{1}{\sqrt{2^n}} Z \otimes I^{\otimes n-1}\right)^{\otimes d+1}$
- **Recall:** $XScore = 2^{2n}\left(E_C\left[\left(p_{0^n} - \frac{1}{2^n}\right)^2\right] - E_C[(p_{0^n} - p)^2]\right)$
- **Proof:** $XScore = 2^{2n} E_C\left[\frac{1}{2^{2n}} - \frac{2}{2^n} p_{0^n} - p^2 + 2p \cdot p_{0^n}\right]$ *(by algebra)*
  - $= 2^{2n} E_C[-\frac{1}{2^{2n}} - p^2 + 2p \cdot p_{0^n}]$ *(using that $E_C[p_{0^n}] = \frac{1}{2^n}$)*
  - $= 2^{2n} E_C[-\frac{2}{2^{2n}} - f(C, s^*, 0^n)^2 + 2p \cdot p_{0^n}]$ *(by def. of p & by Fact 2 cross terms = 0)*
  - $= 2^{2n} E_C[-\frac{2}{2^{2n}} - f(C, s^*, 0^n)^2 + \frac{2p_{0^n}}{2^n} + 2f(C, s^*, 0^n) p_{0^n}]$ *(by def. of p)*
  - $= 2^{2n} E_C[-f(C, s^*, 0^n)^2 + 2f(C, s^*, 0^n)^2]$ *(using that $E_C[p_{0^n}] = \frac{1}{2^n}$ & orthogonality)*
  - $= 2^{2n} E_C[f(C, s^*, 0^n)^2]$ *(by algebra)*

XQUATH algorithm (part 2)

- *Recall* $C = C_d C_{d-1} \dots C_1$ and the path $s^* = \left(\frac{1}{\sqrt{2^n}} Z \otimes I^{\otimes n-1}\right)^{\otimes d+1}$
  - where each layer C_i consists of two qubit gates $C_i^{(1)}, C_i^{(2)}, \dots, C_i^{(n/2)}$
- So far we have: $XScore = 2^{2n} E_C[f(C, s^*, 0^n)^2]$
  - $= 2^{2n} \mathrm{E_C}\left[\mathrm{Tr}(|x\rangle\langle x| s_d^*)^2 \cdot \mathrm{Tr}\left(s_d^* C_d s_{d-1}^* C_d^\dagger\right)^2 \cdot \dots \cdot \mathrm{Tr}\left(s_1^* C_1 s_0^* C_1^\dagger\right)^2 \cdot \mathrm{Tr}(s_0^* |0^n\rangle\langle 0^n)^2\right]$
    - First and the last terms are $\left(\frac{1}{\sqrt{2^n}}\right)^2$ which cancels the 2^{2n} term in front
  - $=$ Product of d squared transition amplitudes each of the form:
    - $= E_{C_i}\left[Tr\left[\left(Z \otimes I^{\otimes n-1}\right) C_i \left(Z \otimes I^{\otimes n-1}\right) C_i^\dagger\right]^2\right]$ (using that each $s_j^* = (Z \otimes I^{\otimes n-1})$)
    - $= E_{C_i^{(1)}}\left[Tr\left[(Z \otimes I) C_i^{(1)} (Z \otimes I) C_i^{(1)\dagger}\right]^2\right] \cdot E_{C_i^{(2)}}\left[Tr\left[(I \otimes I) C_i^{(2)} (I \otimes I) C_i^{(2)\dagger}\right]^2\right] \cdot \dots$
      - Grouping the *two qubit* gates that act on each pair of qubits together and $Tr[A \otimes B] = Tr[A] \cdot Tr[B]$
- **By Fact 1**, all of these expectations except the first are 1, the first is $\frac{1}{15}$
  - So the total score is $\sim \frac{1}{15^d}$

Consequences of XQUATH algorithm

- Notice that the classical algorithm simply computes value of single path in the Pauli basis (takes time $O(n \cdot d)$)
- Algorithm achieves $\mathrm{X}S\mathrm{core}$ of $\frac{1}{2^{O(d)}}$
- If circuit depth is sublinear, then this is a higher score than $\frac{1}{2^n}$ contradicting **XQUATH**

Comments & Open Directions

- A similar algorithm achieves a score of $2^{-O(d)}$ on **XEB** but this algorithm is not yet practical i.e., it doesn't spoof current experiments – can we improve this?
- How hard is achieving a sufficiently large constant score on **XEB** for random quantum circuits with super-constant depth? Recall this is what a *noiseless* random quantum circuit achieves by sampling!
- There's an alternative spoofing method due to [Pan-Chen-Zhang '21], which uses a clever tensor contraction method to simulate Google's 53 qubit **XEB** score on supercomputer in a reasonably short amount of time but takes considerably longer for the USTC 60 qubit experiment

5. **Easiness argument 2** (classical algorithms taking advantage of uncorrected noise)

Uncorrected noise *defines* the NISQ era

- Without error-correction noise eventually overwhelms
  - e.g., Google's RCS experiment ~0.2% signal and 99.8% noise
- Can uncorrected noise help us to classical simulate near-term quantum experiments?
- That is, consider fixing a noise model and for RCS a first reasonable choice is *depolarizing* noise
  - e.g., Each layer of random gates is followed by layer of single qubit depolarizing noise channel with ***constant noise*** strength γ:
  - $\mathcal{E}(\rho) = (1-\gamma)\rho + \frac{\gamma I}{2} Tr[\rho]$
    - Note that $\mathcal{E}$(I)=I but $\mathcal{E}(P) = (1-\gamma)P$ for $P \in \{X, Y, Z\}$
- **Note:** having only depolarizing noise is a simplification!

Quantifying the effects of uncorrected noise

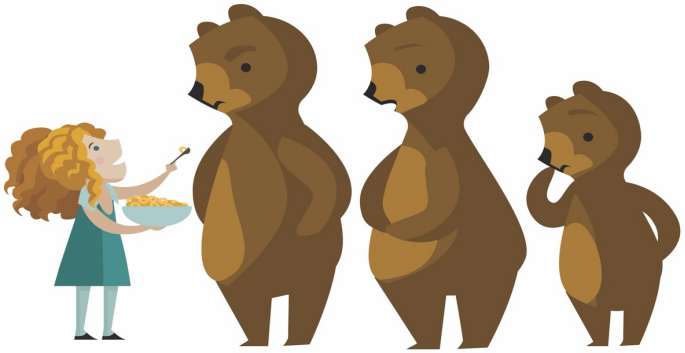
- Intuitively, uncorrected **depolarizing** noise increases entropy. As our circuit gets deeper the output distribution converges to uniform
- **Main question:** how quickly does this happen?
- We've known since the late 90's that the **noisy quantum circuit distribution** with depth d and the **uniform distribution** are $\leq 2^{-\gamma d}$ close in TVD [Aharonov et. al. '96]
- This rules out scalable noisy quantum advantage at *super-logarithmic depth*
- What about random circuits? Could the convergence be faster?
  - Numerical evidence that convergence to uniform happens faster [Boixo et. al. '17]
  - i.e., TVD upper bounded by $\leq 2^{-\gamma \cdot d \cdot n}$ whp over C
    - This would rule out scalable noisy quantum advantage *at any depth*!

How much depth is required for quantum advantage?

- ***Anticoncentration*** is one ingredient of current hardness of sampling arguments that requires sufficiently deep random circuits (with Haar random gates)
- A distribution over circuits **anticoncentrates** if:
  - There exists constants $\alpha \in (0,1], c > 0$ so that $\Pr_C \left[p_{0^n}(C) \geq \frac{\alpha}{2^n} \right] \geq c$
  - Stronger statement "collision probability" bound: $2^n E_C \sum_x p_x(C)^2 \leq c$
  - Notice this is *not sufficient* for hardness – not even a computational property!
  - Rather it's a *sanity check* that $\pm O(2^{-n})$ additive estimates to p_{0^n} aren't trivial!
- Until recently, we only knew **anticoncentration** for 2D circuits (with Haar random gates) happened at depth $\geq \sqrt{n}$ [Harrow & Mehraban '18]
- This is too deep for scalable noisy quantum advantage!
  - i.e., we know that the output distributions are $\leq 2^{-\gamma d} \sim 2^{-\sqrt{n}}$ close to uniform

Is there any hope for *fully scalable*, noisy quantum advantage from RCS?

- Consequently until last year, there was little optimism that we could get such an advantage
  - Rather we hope for "Goldilocks" system sizes to keep the system from getting too noisy
- Then two results rekindled some hope at $\log(n)$ depth…
  1. Anticoncentration at $\log(n)$ depth [Barak et. al. '21][Dalzell et. al. '22]
  2. TVD between noisy random circuit distribution and uniform is *lower bounded* by $2^{-O(d)}$ whp [Deshpande et. al. '22]
     - Matches the Aharonov et. al. '96 upper bound and rules out faster convergence rates



Goldilocks and the three bears

Can a classical algorithm beat uniform sampling at depth $\log n$?

- For $\mathrm{d} = \mathrm{O}(\log(n))$ depth noisy circuits we know that the uniform distribution is $2^{-O(d)} = \frac{1}{\mathrm{n^c}}$ close in TVD to the output distribution by [Aharonov et. al. '96] upper bound
- But it was possible that quantum advantage persists for sampling from a distribution $\frac{1}{n^{c\prime}}$ -close in TVD to the noisy output distribution for some sufficiently large constant $c' > \mathrm{c}$
- This possibility has recently been ruled out by [Aharonov et. al. '22]

The [Aharonov, Gao, Landau, Liu, Vazirani'22] algorithm

- [Aharonov et. al. '22] give a classical algorithm for sampling from a distribution $\epsilon - close$ to the distribution of noisy random quantum circuits in $poly\left(n, \frac{1}{\epsilon}\right)$ time modulo several caveats
- But this running time scales $\exp\left(\frac{1}{\gamma}\right)$ with noise-rate γ
- The algorithm's score is not currently competitive with near-term experiments
- Also algorithm **requires anticoncentration, so is only efficient and useful (i.e., beats uniform sampling) at $\mathbf{O}(log(n))$ depth**
- Finally, algorithm requires certain constraints on the gate set (satisfied e.g., by Haar random gates)

Main ideas of [Aharonov et. al. ’22]

- **Key observation** [Gao & Duan’18][Aharonov et. al. ‘22]: Output probabilities (and marginals) of noisy random quantum circuits in Pauli basis have most mass on a small number of paths, rest of the paths are exponentially suppressed
- **Recall notation:** in Pauli basis $p_x(C) = \sum_{s \in P_n^{d+1}} f(C, s, x)$
- Then by definition of depolarizing noise, the noisy output probability: $\tilde{p}_x = \sum_{s \in P_n^{d+1}} (1-\gamma)^{|s|} f(C, s, x)$
  - Where $|s|$ is the Hamming weight, or number of non-Identity Paulis in path
- **Main idea:** To compute p_x simply throw away high-weight Pauli terms and exactly compute the low weight terms!
- i.e., for appropriate cutoff, ℓ, compute $\overline{q_x} = \sum_{s:|s| \leq \ell} (1-\gamma)^{|s|} f(C, s, x)$

Analysis of the [Aharonov et. al. ’22] algorithm

- Recall the algorithm works by truncating the Pauli path integral of each noisy output probability, then computing truncated probability *path by path*
- Analysis in two steps:
  1. Determine the truncation parameter ℓ needed to attain bounded TVD, $|\tilde{p} - \bar{q}|_1$
  2. Upper bound the running time of the algorithm as a function of ℓ

Step 1: How to set cutoff ℓ to bound TVD

- Goal is to obtain upper bound on $|\tilde{p} - \bar{q}|_1 = \Delta$
- $E_C[\Delta^2] \leq 2^n E_C\left[\sum_{x\in\{0,1\}^n} (\tilde{p}_x - \bar{q}_x)^2\right]$ *(by Cauchy-Schwarz)*
- $= 2^n E_C\left[\sum_x \left(\sum_{s:|s|>\ell} (1-\gamma)^{|s|} f(C,s,x)\right)^2\right]$ *(by definition of $\tilde{p}_x$ and $\bar{q}_x$)*
- $= 2^n E_C\left[\sum_x \sum_{s:|s|>\ell} (1-\gamma)^{2|s|} f(C,s,x)^2\right]$ *(orthog. of Pauli paths, Fact 2)*
- $= \sum_{k>\ell} (1-\gamma)^{2k} W_k$ *(rewriting, where W_k is "Fourier weight")*
- $\leq (1-\gamma)^{2\ell} \sum_{k>\ell} W_k$ *(since $k > \ell$)*
- $\leq e^{-2\gamma\ell} \cdot O(1)$ *(nontrivial upper bound on W_k follows from anticoncentration)*

- So can take $\ell \approx \frac{1}{\gamma} \cdot \log\left(\frac{1}{\epsilon}\right)$ to obtain $\Delta \leq \epsilon$ with high probability by Markov

Step 2: How to compute truncated prob., $\bar{q}_x$?

- Algorithm works by computing value of each path in truncated probability
- How many terms in $\bar{q}_x = \sum_{s:|s|\leq \ell}(1-\gamma)^{|s|} f(C,s,x)$?
- Number of paths with Hamming weight at most ℓ is $\leq \ell \cdot \binom{n(d+1)}{\ell} \cdot 3^{\ell}$
  - Since each path has $n(d+1)$ Pauli operators and we're choosing ℓ to be non-identity & there are 3^{ℓ} different sequences of operators $\{X,Y,Z\}^{\ell}$
  - Takes $O(n \cdot d)$ time to compute each path
- Total time dominated by # of paths $\sim (n \cdot d)^{O(\ell)} \sim n^{\frac{1}{\gamma}\log\left(\frac{1}{\epsilon}\right)}$ if $\ell = \frac{1}{\gamma}\log\left(\frac{1}{\epsilon}\right)$
- Can improve dependence to $2^{O(\ell)}$ by being be more clever – uses anticoncentration and the fact that many paths contribute 0 to the path integral.
  - Notice by choice of ℓ that this is exponential in $\frac{1}{\gamma}$ as well

Comments & Open Directions

- This algorithm applies to constant noise rates. For $\gamma = \tilde{O}\left(\frac{1}{n}\right)$ there's evidence for hardness of sampling [Dalzell et. al. '21]
- This algorithm doesn't spoof current RCS experiments -- can we do better?
- Can we generalize the Aharonov et. al. algorithm to other noise models besides depolarizing?
  - Our very recent work suggests this may be difficult for *non-unital noise* (Ghosh et. al., arXiv: 2306.16659)! Real world experiments have both unital and non-unital noise channels!
- Can we generalize the Aharonov et. al. algorithm to gate sets that are very far from Haar random?
- How hard are noisy random circuits with *sublogarithmic depth* and Haar random gates?
  - Not covered by this algorithm because of anticoncentration is known to fail here [Dalzell et. al. '21][Deshpande et. al. '22]!
- **Most generally, is fully scalable quantum advantage possible without error mitigation, for any experiment?**

More work I hope you check out!

- Random circuits with non-unital noise do not anticoncentrate at any depth
  - Our work: Ghosh et. al., arXiv: 2306.16659
- Hardness of *Gaussian* Boson Sampling experiments: e.g.,
  - Our work on this [Deshpande et. al. '21, arXiv: 2102.12474]
  - "Bipartite GBS" [Grier et. al.'21, arXiv: 2110.06964]
- Verifying and spoofing current Boson Sampling experiments
  - Efficiently distinguishing Boson Sampling distribution from uniform [Aaronson & Arkhipov '13, arXiv:1309.7460]
  - Our very recent work classically simulates the largest current size Gaussian Boson Sampling [Oh et. al. '23, arXiv:2306.03709]
    - Tensor network that takes advantage of photon loss!
- Useful applications of quantum advantage experiments? e.g.,
  - Molecular vibronic spectra problem via Boson Sampling
    - See original proposal of [J.Huh et. al., arXiv: 1412.8427]
    - See our quantum inspired classical algorithm for this problem, as well as alternative quantum chemistry problems that still might be classically hard [Oh et. al., arXiv: 2202.01861]
  - Certified random number generation from Random Circuit Sampling
    - see proposal of Aaronson and Hung (arXiv: 2303.01625)
    - our work providing evidence for this proposal [Bassirian et. al. '22, arXiv: 2111.14846])

Thanks!