CS 59300 – Algorithms for Data Science

Classical and Quantum approaches

Lecture 0 (08/26)

Introduction

https://ruizhezhang.com/course_fall_2025.html

About Me

Ruizhe Zhang

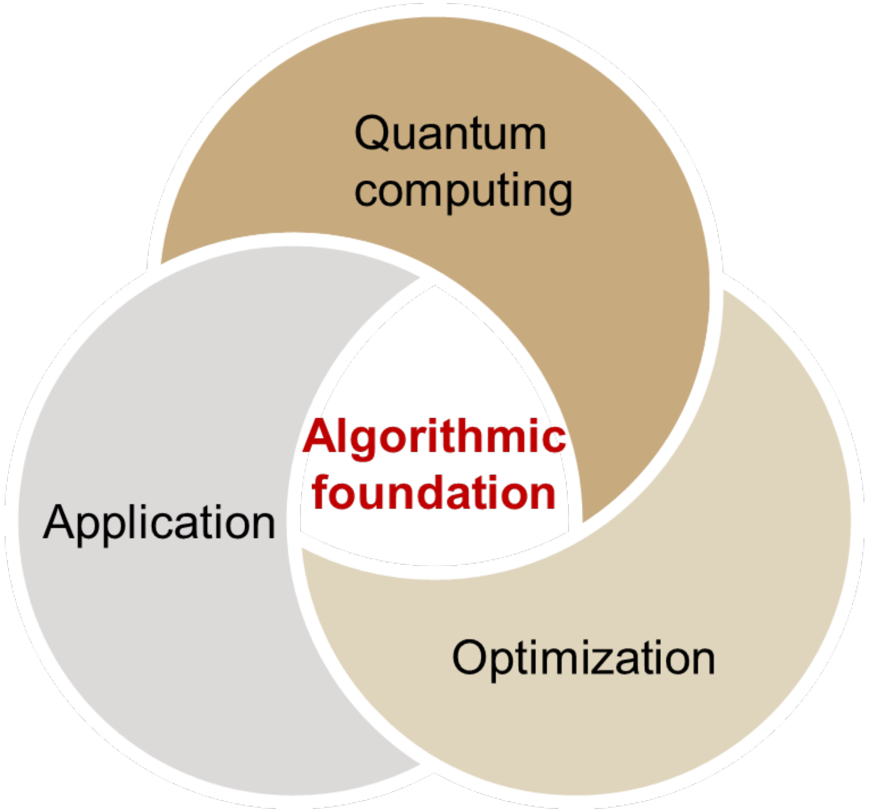
/rway↘ jer↗/

- Assistant Professor, Purdue CS
- Simons Quantum Postdoctoral Fellow, UC Berkeley
- Ph.D. in Computer Science, The University of Texas at Austin
- Research: Theory--quantum computing, optimization, ML theory
- Email: rzzhang@purdue.edu

Research at a Glance

How to design algorithms—quantum & classical—that push the limits of computation and deliver real impact?

- Quantum algorithms that provide advantages over classical methods while accommodating quantum hardware constraints
- Fast, provable solvers for large-scale optimization problems
- Theoretical foundations and algorithmic improvements in practical domains



I am always looking for talented people joining the group!

Today's plan

- Logistics
- Philosophy
- A motivating example

Grading

- Problem sets (15%)
- Participation (10%)
- Scribing (10%)
- Take-home midterm exam (25%)
- Final project (40%)

Problem sets

- $\sim$ 3 problem sets
- You are encouraged to collaborate on homework. However, you must write up your own solutions. You should also state the names of those you collaborated with on the first page of your submission.
- **Gen-AI rule:**
  In this course, you shall give credit to AI tools whenever used, even if only to generate ideas rather than usable text or illustrations.

Gen-AI rule

- Allowed:

Compute the Gaussian integral: int_\R^n e^{-1/2 * x^T A x + J^T x} dx

Thought for 10s

$$I(J) = \int_{\mathbb{R}^n} \exp\left(-\tfrac{1}{2}x^\top Ax + J^\top x\right)\, dx$$

$$I(J) = e^{\frac{1}{2}J^\top A^{-1}J} \int_{\mathbb{R}^n} \exp\left(-\tfrac{1}{2}y^\top Ay\right) dy = \sqrt{\frac{(2\pi)^n}{\det A}}\, \exp\left(\tfrac{1}{2}J^\top A^{-1}J\right).$$

- Not allowed:

Problem 1

Show that for a matrix M its rank and its border rank are always the same. In particular, suppose you are given a matrix M and a parameter r so that for every $\epsilon > 0$ there is a rank r matrix M_r so that M and M_r are entrywise ϵ-close. Show that M must have rank at most r. *Hint:* Use the Eckhart-Young Theorem.

Thought for 7s

Let $M \in \mathbb{F}^{m\times n}$ with singular values
$\sigma_1 \geq \cdots \geq \sigma_{\min(m,n)} \geq 0$.
The **border rank** of M is the least r for which there exists a sequence of rank-$\leq r$ matrices $M^{(k)}$ converging entrywise to M.

Participation

- You should attend every lecture unless you have an unavoidable conflict.

- Don’t hesitate to stop me at any point to ask questions.

Scribing

- Signup sheet will be posted on Brightspace
- You need to scribe notes for two lectures
- The LaTeX template is provided in the course website

Midterm exam

To fulfill the PhD degree requirement, we have a take-home exam (25%).

- The exam time will be announced at least two weeks in advance

- The use of internet or locally hosted AI tools is strictly prohibited

Final project

- Either original research or insightful exposition of existing work

- Written report + Oral presentation

- Suggested topics/readings will be provided following the midterm exam; however, you may also propose alternative topics for approval.

- I strongly encourage each of you to schedule a meeting with me to discuss your project ideas.

Today's plan

- Logistics
- Philosophy
- A motivating example

This course: classical and quantum algorithmic foundations for data science with provable guarantees.

Data science is an interdisciplinary academic field that uses statistics, scientific computing, scientific methods, processing, scientific visualization, algorithms and systems to extract or extrapolate knowledge from potentially noisy, structured, or unstructured data.

Vasant Dhar (2013)

Now that there is AI, is data science still needed? Or shall we declare it dead?

Short answer: No!

Data Science Isn’t Dying — It’s Evolving: How AI Is Reshaping the Role

https://medium.com/data-science-collective/data-science-is-dead-again-why-the-role-keeps-evolving-not-disappearing-21ac8586a22a

Datasets are the foundation of progress in AI

For text:

- GPT-1 (2018): 3 B Tokens
- GPT-2 (2019): 30 B Tokens
- GPT-3 (2020): 300 B Tokens
- GPT-4 (2023): 3000 B Tokens (?)

1000x growth in 5 years

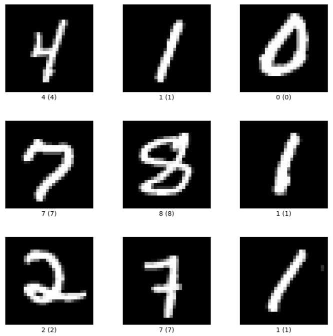
For images:

- ImageNet (2009): 1 Million images
- LAION-5B (2022): 5 Billion Images

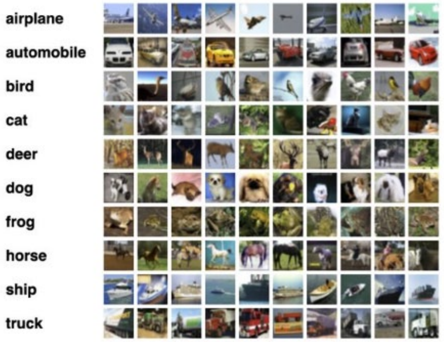
5000x growth in 5 years

Slides from Alex Dimakis's talk: https://www.youtube.com/watch?v=ba-aqPF6xuw

ML Discoveries enabled by datasets



MNIST (1994)
Convolutional neural networks



CIFAR-10 (2009)
Training on GPUs

ImageNet (2012)
Deep training resurgence, ResNets, transfer learning, etc.

OpenAI

WebImageText (2021)
Zero-shot classification (CLIP), text-guided image generation (DALL-E)

Slides from Alex Dimakis's talk: https://www.youtube.com/watch?v=ba-aqPF6xuw

This class is

- A journey that we will explore together how to think algorithmically in data science
  - → Algorithms
  - → Hardness
  - → Modeling

- A theory course and therefore mainly contains proofs

- NOT a course about the algorithms/techniques for immediate practical deployment (e.g., Transformer, chain-of-thought, MoE,…)

- NOT a substitution for *CS59300-IQC Intro to Quantum Computing* (though we do not assume any background knowledge in quantum)

Think algorithmically in data science - Algorithms

Goal: More efficient ways to extract knowledge from data (classically or quantumly)

- However, big gap between what's possible in practice and what we can prove theorems about.

Our approaches:

- Design and rigorously analyze algorithms
- Develop theoretical frameworks/meta-algorithms that could become practically useful heuristics
- Identify "hidden levers" from practical heuristics and inspire the theoretical studies

Think algorithmically in data science - Hardness

We want to understand when a (heuristic) algorithm cannot work.

- Proving worst-case lower bounds is the most common approach
- However, almost all the optimization problems that arise in modern machine learning are computationally intractable
- Go beyond worst-case analysis (average-case hardness and smoothed analysis)
- What factor makes the problem hard?

Easy	Hard	Impossible
polynomial-time algorithms exist	only inefficient algorithms exist	statistically unsolvable

Think algorithmically in data science - Modeling

There are many expressive models for describing the world around us:

- Graphical models
- Mixture models
- Markovian processes
- Linear dynamical systems
- Quantum circuits
- Local Hamiltonians
- Dissipative processes (open quantum systems)
- …

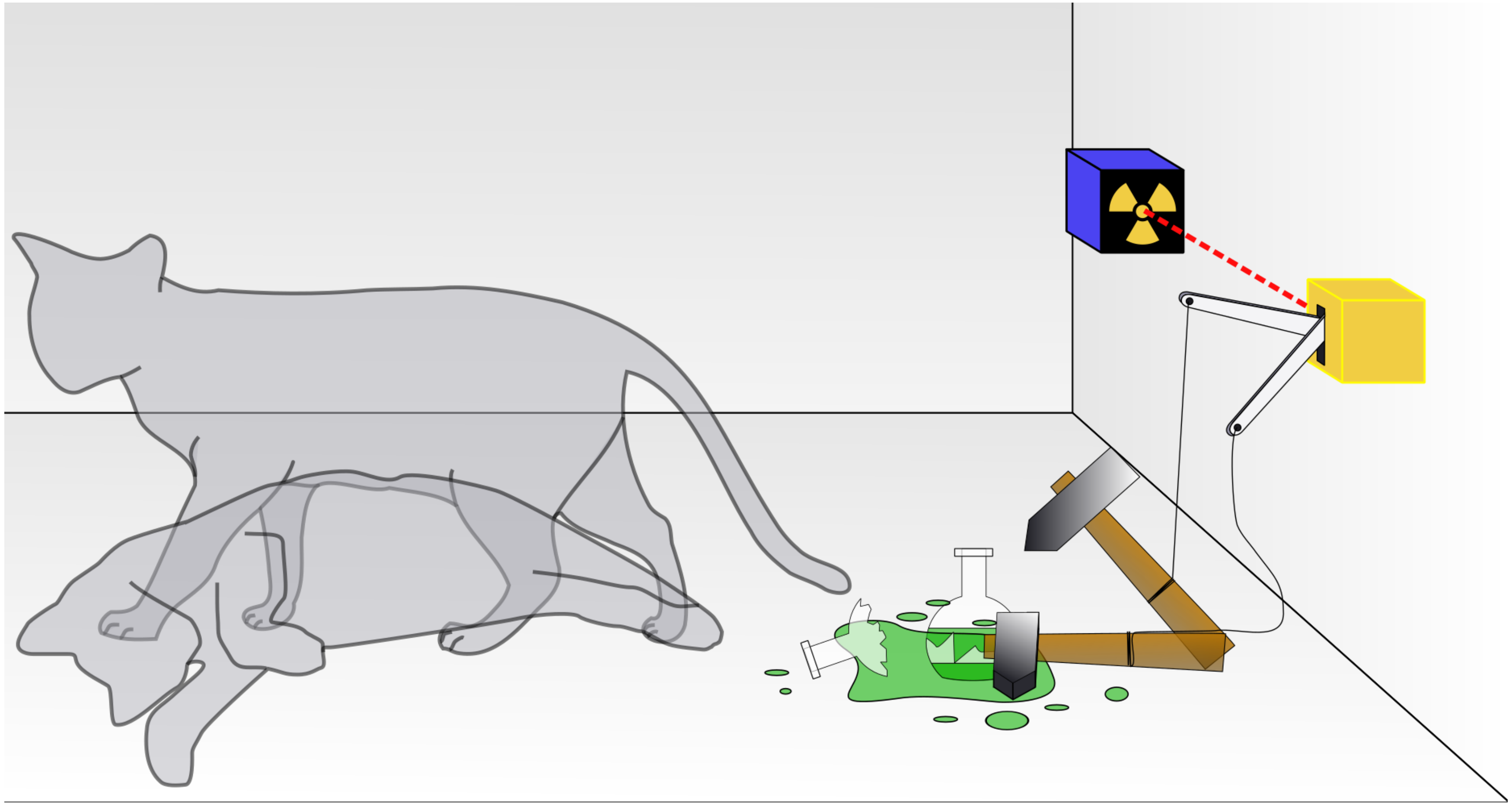
A model is only as good as our ability to use it!

- Can rigorously describe what algorithms in practice can solve and what cannot
- Can capture the key properties/structures in the data that make the problem "easy"
- Does not "overfitting" to artificial assumptions

Today's plan

- Logistics
- Philosophy
- A motivating example: Quantum supremacy experiments

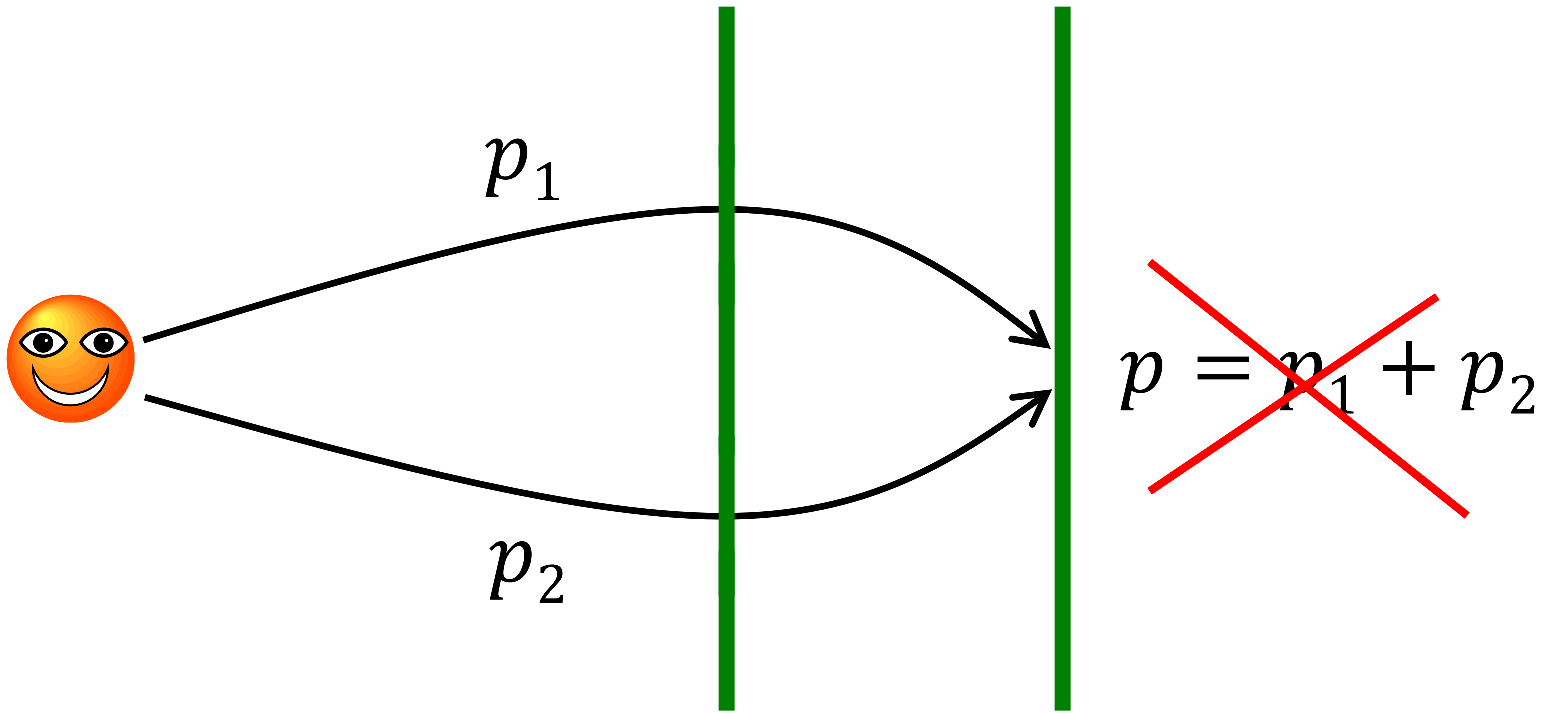
A quantum computer is a machine that uses the principles of Quantum Mechanics to perform computations.



Slides adapted from Scott Aaronson's tal

Quantum Mechanics

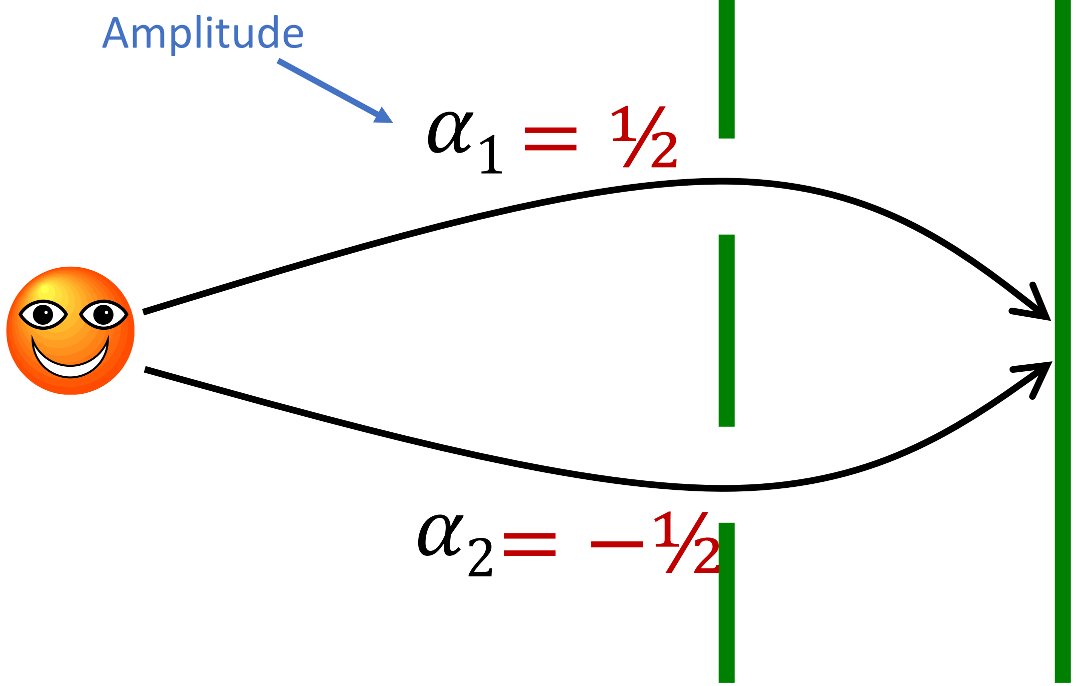
"Probability theory with minus signs"



Slides adapted from Scott Aaronson's tal

Quantum Mechanics

“Probability theory with minus signs”



$$\alpha = \alpha_1 + \alpha_2 = 0$$

$$p = |\alpha|^2 = 0$$

Slides adapted from Scott Aaronson’s tal

A **quantum computer** is made not of bits but of **qubits**, which can be in **superpositions** of the 0 and 1 states: that is, they have an amplitude to be 0 and an amplitude to be 1:

$$\alpha|0\rangle + \beta|1\rangle$$

2 qubits	$\Rightarrow$	4 amplitudes (for 00, 01, 10, and 11)
3 qubits	$\Rightarrow$	8 amplitudes
50 qubits	$\Rightarrow$	$2^{50} \approx$ quadrillion amplitudes
1000 qubits	$\Rightarrow$	more amplitudes than fit in visible universe

Makes Nature very hard to simulate on conventional computers

Feynman, Deutsch 1980s: If Nature gives you a lemon, make lemonade! (I.e., quantum computers)

A superposition of Feynmans

Slides adapted from Scott Aaronson and Lin Lin

The development of quantum computers

Sci-Fi | Toy devices | Supremacy | Fault-tolerance

1960s

1981

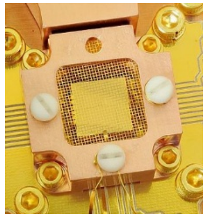
2001

15 was factored using NMR

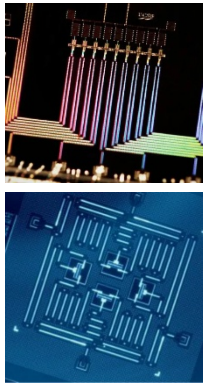
2011

D-Wave One Quantum annealer

2017

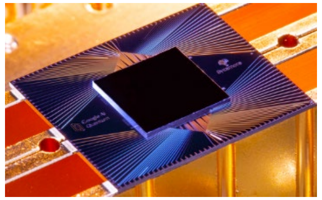


Trapped ion



Superconducting

2019

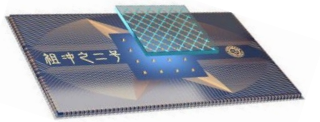


Google supremacy: RCS (53-qubit)

2020

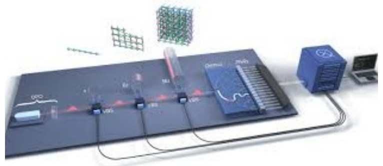
USTC supremacy: GBS (76 photons)

2021



USTC supremacy: RCS (60-qubit)

2022



Xanadu supremacy: GBS (219 photons)

2023

QEC with neutral atoms

2024

Google's Willow

THE QUANTUM INSIDER

Quantum Computing Market Map

Non exhaustive and in no particular order. Excludes details on control systems, assembly languages, circuit design, etc.

Users
Select examples

Material Science
Merck, OTI, Airbus, BMW

Finance
Goldman Sachs, Bank of America, J.P.Morgan, Wells Fargo

Life Sciences
Odyssey Therapeutics, Boehringer Ingelheim, AstraZeneca, gsk

Other
Volkswagen, Johnson Electric, Denso

Applications
Not mapped to verticals

Software offerings
Includes control software

Not strictly categorized given diversity of operations[1]
Multiverse Computing, QC Ware, CLOSED, 1QBit, Aliro Quantum, Horizon Quantum Computing, HQS Quantum Simulations, CLOSED, QCi Ready-to-run Quantum Software, Entropica Labs, river lane, Strangeworks, QuantFi, Phasecraft, aq, QunaSys, Terra Quantum, ⟨b| e^{it}, ProteinQure, Keysight Technologies, Qubit pharmaceuticals, Q-CTRL, Classiq, Quantum-South, menten.AI, ParityQC

Cloud access to QPUs
aws, Google Cloud, Azure, Huawei Cloud

Simulators / q-inspired / etc
Atos, Fujitsu, NVIDIA

QPUs[2]

Superconducting
rigetti, Google AI Quantum, OQC, QILIMANJARO Quantum Tech, IBM Q, qci, IQM, Alice & Bob, 本源量子 Origin Quantum, Nord Quantique, aws, seeqc, D-Wave The Quantum Computing Company, CLOSED

Ion Trap
Quantinuum, IonQ, AQT, oxford ionics, Quantum Art, eleQtron

Neutral Atoms
Universal Quantum, atom computing, ColdQuanta, QuEra, Pasqal

Silicon
Silicon Quantum Computing, intel, Quantum Motion, equal1

Photonics
PsiQuantum, QC82, Xanadu, photonic, ORCA Computing, Quandela, Turing

Other
EeroQ Quantum Hardware, Electrons on helium, Quantum Brilliance, NV Diamond

Hardware / components
Select examples only – not representative of entire ecosystem

Cryogenics (includes testing)
Oxford Instruments, FormFactor, Lake Shore Cryotronics, Bluefors, Montana Instruments, ICE, Maybell

Lights and lasers
Vescent Photonics, NKT Photonics, M Squared, Coherent, QANT, Toptica Photonics, Ligentec

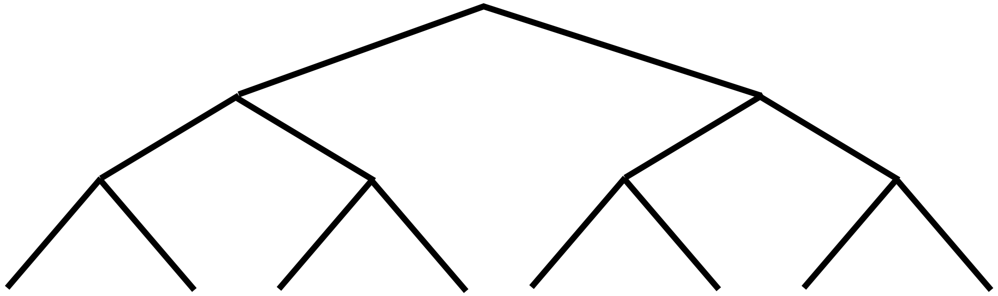
Other componentry (examples)
Delft Circuits Hardware for quantum engineers, Keysight Technologies, QM Quantum Machines, Quantum Design, CryoCoax A division of Intelliconnect, ZNF, Qblox, QDevil, Zurich Instruments

[1] Software offerings can be further classified into SDKs, firmware / enablers, algorithms / applications, simulators etc. but many companies are offering a mixture across the stack

[2] Many QPU providers are offering full stack services (e.g. Pasqal acquired Qu&Co, Quantinuum was originally CQC prior to merger with HQS, etc.

Popularizers beware:

A quantum computer is NOT like a massively-parallel classical computer!



Exponentially many possible answers, but you only get to observe **one** of them

Any hope for a speedup rides on choreographing an **interference pattern** that boosts the amplitude of the right answer

$$\underbrace{+1 \;\; +1 \;\; +1 \;\; +1}_{\text{Right answer}} \qquad \underbrace{\cancel{+1 \;\; -1} \;\; \cancel{+i \;\; -i}}_{\text{Wrong answer}}$$

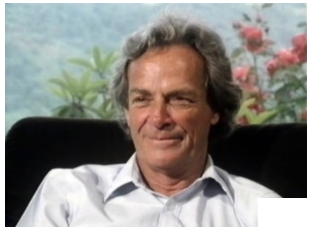
So, what are the main known DREAM applications of QCs?

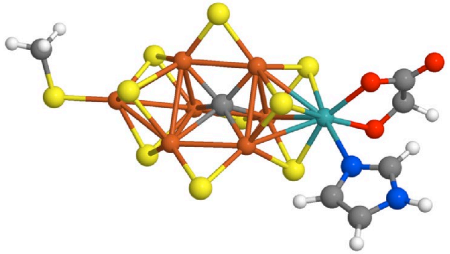
1. Breaking Current Public-Key Cryptography

Requires fault-tolerance

Post-quantum crypto is a viable response

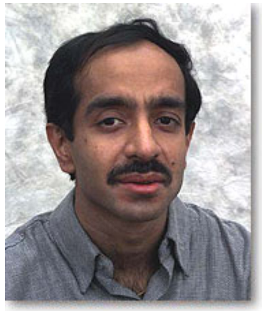
2. Simulating Quantum Physics and Chemistry



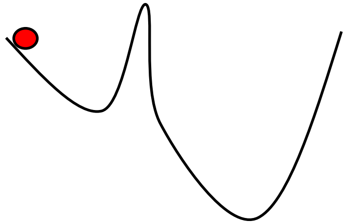


Still the best known “killer app”

3. Uhh, more hopefully?



Grover,
HHL,
QAOA



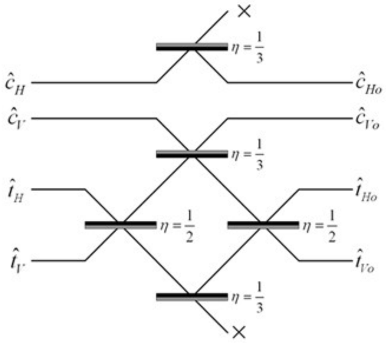
Looks like mostly modest speedups + 90% hype but who knows?

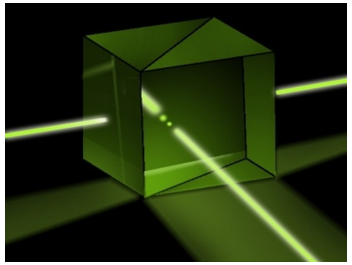
Slides adapted from Scott Aaronson’s talk

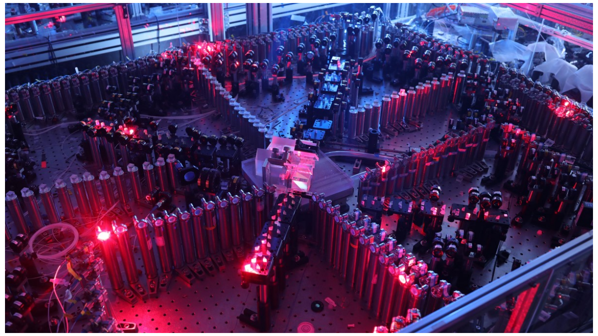
REALITY: "Quantum Supremacy" demonstrated over the past 6 years

BosonSampling

(Aaronson-Arkhipov 2011, ~100-photon experiments by USTC team 2020, Xanadu 2022)

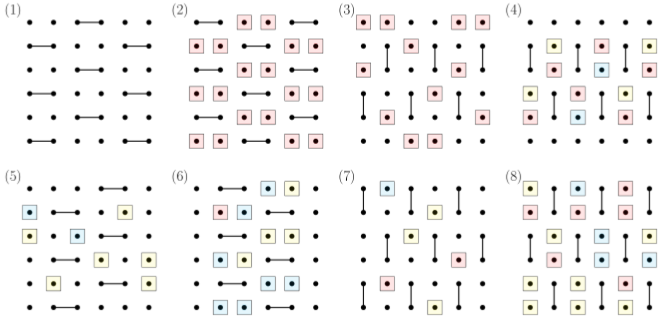


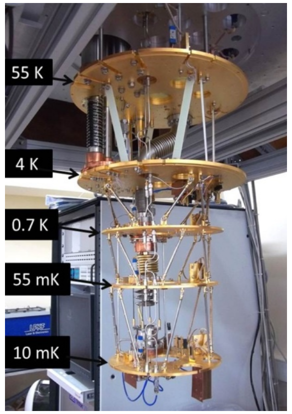


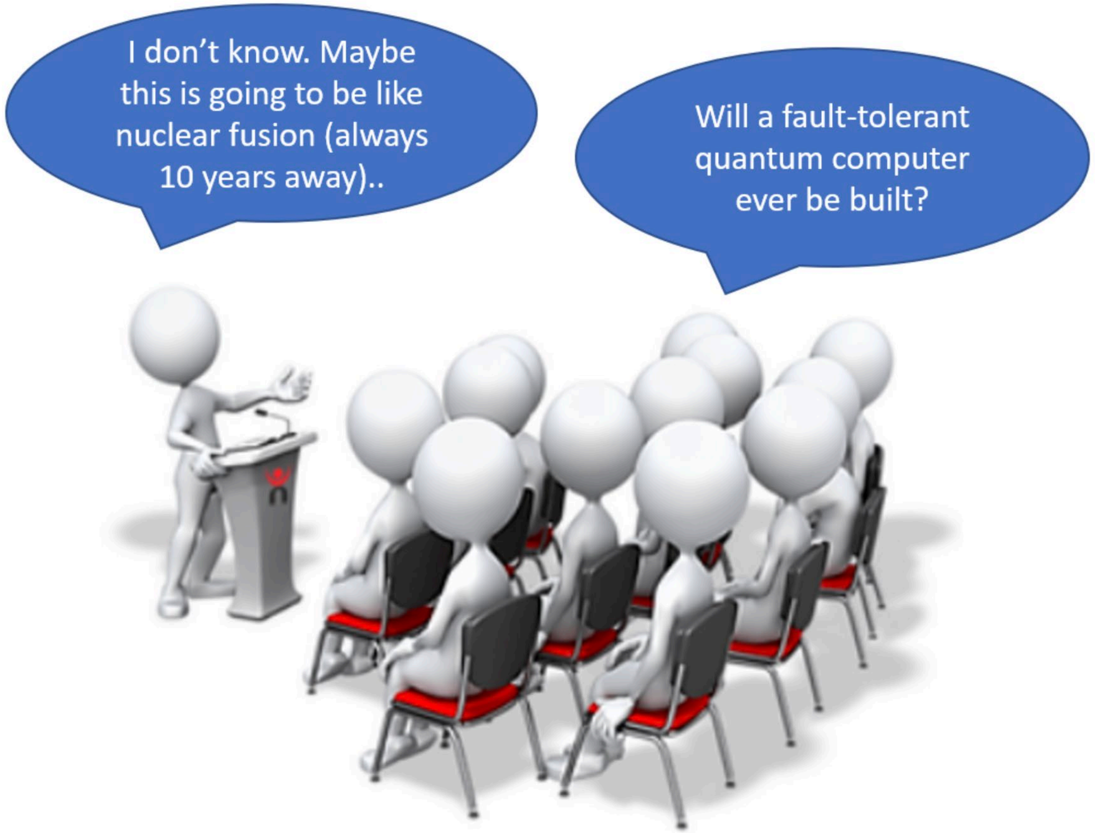


Random Circuit Sampling

(53-qubit experiment by Google 2019, then 103 qubits in 2024; also USTC)





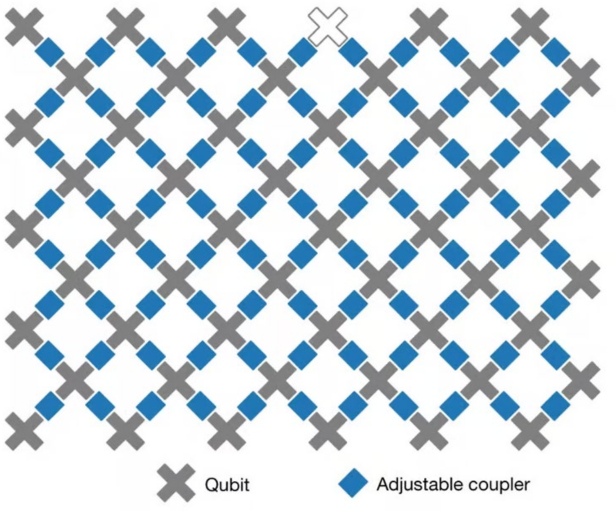


Quantum supremacy: quantum computers can perform certain (can be arbitrarily contrived) tasks much more efficient than classical computers

Quantum advantages: quantum computer is faster than classical computer on a useful task

The latter half of the course will introduce potential approaches toward realizing quantum advantages

What exactly did Google & USTC do?



Random circuit sampling (RCS)

- $n = 53$ qubits and ~20 layers of gates in the circuit (randomly chosen)
- ~40 microseconds per sample ($s_i \in \{0,1\}^{53}$)
- ~3 mins for millions of samples $s_1, s_2, \dots, s_K$
- But how do we check whether $s_1, \dots, s_K$ were actually sampled from a QC?

Linear Cross-Entropy Benchmark:

$$\mathrm{LXEB} := \frac{2^n}{K} \sum_i \Pr[\text{the circuit } C \text{ outputs } s_i] \equiv \frac{2^n}{K} \sum_i |\langle s_i | C | 0^n \rangle|^2$$

Generating s_i **uniformly at random** would yield $\mathrm{LXEB} \approx 1$

Google's result: $\mathrm{LXEB} \approx 1.002$

Sampling with an **ideal QC** would yield $\mathrm{LXEB} \approx 2$, due to quantum inferences boosting the probabilities of some s_i's over others

Classical spoofing of RCS in theory: is there a $\sim\ 2^n$ barrier?

Problem (Linear Cross-Entropy Heavy Output Generation, XHOG).

Given the classical description of a quantum circuit C, generate K distinct samples $s_1, \ldots, s_K \in \{0,1\}^n$ such that $\mathrm{LXEB}\big(\{s_i\}_{\{i\in[K]\}}, C\big) \geq b$, where $b \in (1, 2)$.

A quantum circuit can be described as a sequence of matrix-vector products:

$$C|0^n\rangle = \underbrace{U_m \cdot U_{m-1} \cdot \cdots \cdot U_3 \cdot U_2 \cdot U_1}_{\text{quantum gates}} \cdot |0^n\rangle$$

unit vector in $\mathbb{C}^{2^n}$

2^n-by-2^n matrix

unit vector in $\mathbb{C}^{2^n}$

$$\begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array} \begin{pmatrix} c_{00} \\ c_{01} \\ c_{10} \\ c_{11} \end{pmatrix} \qquad \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array}$$

$$\Pr[C \text{ outputs } 0^n] = |\langle 0^n|C|0^n\rangle|^2 = |c_{00}|^2$$

This is called the "Schrödinger" algorithm for simulating a quantum circuit

- $\mathcal{O}(m2^n)$ time and $\mathcal{O}(2^n)$ space

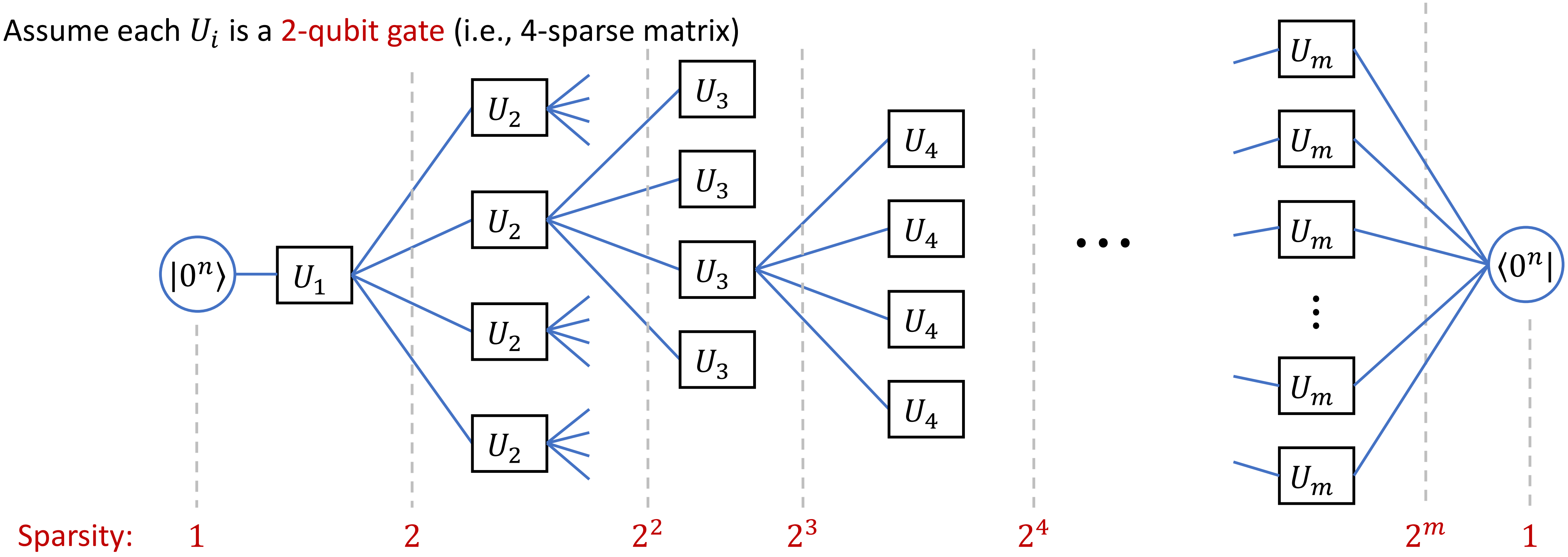
Classical spoofing of RCS in theory: is there a $\sim\ 2^n$ barrier?

A quantum circuit can be described as a sequence of matrix-vector products:

$$C|0^n\rangle = U_m \cdot U_{m-1} \cdot \cdots \cdot U_3 \cdot U_2 \cdot U_1 \cdot |0^n\rangle$$

Feynman's path integral:

Assume each U_i is a 2-qubit gate (i.e., 4-sparse matrix)



Classical spoofing of RCS in theory: is there a $\sim\ 2^n$ barrier?

A quantum circuit can be described as a sequence of matrix-vector products:

$$C|0^n\rangle = U_m \cdot U_{m-1} \cdot \cdots \cdot U_3 \cdot U_2 \cdot U_1 \cdot |0^n\rangle$$

Feynman's path integral:

Assume each U_i is a 2-qubit gate (i.e., 4-sparse matrix)

$$\begin{aligned}
\langle 0^n|C|0^n\rangle &= \langle 0^n|U_m \cdot U_{m-1} \cdot \cdots \cdot U_3 \cdot U_2 \cdot U_1|0^n\rangle \\
&= \langle 0^n|U_m \cdot I \cdot U_{m-1} \cdot I \cdot \cdots \cdot I \cdot U_3 \cdot I \cdot U_2 \cdot I \cdot U_1|0^n\rangle \\
&= \langle 0^n|U_m \cdot \sum_{x_{m-1}\in\{0,1\}^n} |x_{m-1}\rangle\langle x_{m-1}| \cdot U_{m-1} \cdots U_2 \cdot \sum_{x_1\in\{0,1\}^n} |x_1\rangle\langle x_1| \cdot U_1|0^n\rangle \\
&= \underbrace{\sum_{x_1,\ldots,x_{m-1}\in\{0,1\}^n}}_{4^m \text{ nonzero terms}} \langle 0^n|U_m|x_{m-1}\rangle \cdot \langle x_{m-1}|U_{m-1}|x_{m-2}\rangle \cdots \langle x_2|U_2|x_1\rangle \cdot \langle x_1|U_1|0^n\rangle
\end{aligned}$$

- $\mathcal{O}(4^m)$ time and $\mathcal{O}(m+n)$ space

Classical spoofing of RCS in theory: is there a $\sim\ 2^n$ barrier?

Classical Simulation Algorithm	Time	Memory
Schrödinger	$\sim 2^n$ $(n = \#\text{qubits})$	$\sim 2^n$
Feynman	$\sim 2^m$ $(m = \#\text{gates})$	Linear
Schrödinger-Feynman (Aaronson-Chen 2017)	$\sim d^n$ $(d = \text{depth})$	Linear

Well, the classical simulation of a quantum circuit seems to be hard. What about the XHOG problem? Can we generate high LXEB samples without computing the probabilities?

Theorem (Aaronson-Chen 2017, Aaronson-Gunn 2019).

If there's a classical algorithm to spoof Linear XEB in $\ll 2^n$ time, then there's also a fast classical algorithm that estimates a specific output probability like $|\langle 0^n|C|0^n\rangle|^2$, with slightly better variance than always guessing 2^{-n}.

Classical spoofing of RCS in theory: is there a $\sim\ 2^n$ barrier?

Theorem (Aaronson-Chen 2017, Aaronson-Gunn 2019).

If there's a classical algorithm to spoof Linear XEB in $\ll 2^n$ time, then there's also a fast classical algorithm that estimates a specific output probability like $|\langle 0^n|C|0^n\rangle|^2$, with slightly better variance than always guessing 2^{-n}.

Theorem (Bouland et al. 2021).

For a constant-depth random quantum circuit C, it is #P-hard to compute

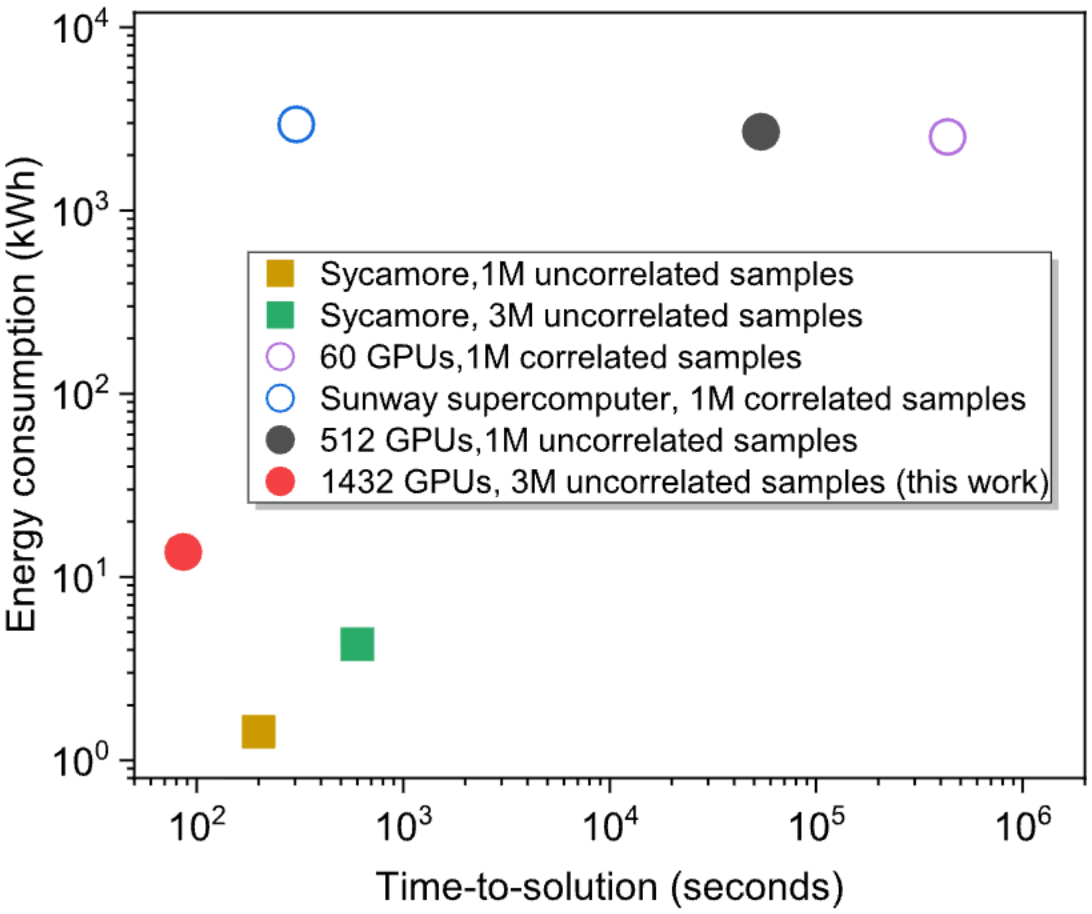
$$|\langle 0^n|C|0^n\rangle|^2 \pm 2^{-\mathcal{O}(n\log n)}$$

- For a general random circuit, their hardness result hold with the robustness of $\exp\big(-\mathcal{O}(m\log m)\big)$, while the goal is to prove hardness for $\exp(-n)$

$$m = \mathcal{O}(nd)$$

Classical Spoofing of RCS in Practice

- **IBM:** Summit, the larg asketball courts and has 250 petabytes of hard disl lation in ~2.5 days, rather than the 10,000 years



- **Pan & Zhang, Liu et a** using tensor networks
- **Pan, Chen, & Zhang:** rcomputer in 15 hours
- **Zhao et al.:** 3M sampl Google's Sycamore which needs 200 seconds)

Classical Spoofing of RCS in Practice

Date	Problem	n	m	Group & computer	Computer type	Ref.	Status	Section
Oct 23 2019	RCS	53	20	Google *Sycamore*	Superconducting	[5]	Refuted by [6]	Sec. II A 1
Dec 03 2020	GBS	50	100	USTC *Jiǔzhāng*	Photonic	[7]	Weakly refuted by [8]	Sec. II B 1
Jun 28 2021	RCS	56	20	USTC *Zuchongzhi*	Superconducting	[9]	Challenged by [10, 11]	Sec. II C 1
Jun 29 2021	GBS	50	144	USTC *Jiǔzhāng 2.0*	Photonic	[12]	Weakly refuted by [8]	Sec. II B 2
Sep 08 2021	RCS	60	24	USTC *Zuchongzhi*	Superconducting	[13]	Challenged by [11, 14]	Sec. II C 2
Jun 01 2022	GBS	216	216	Xanadu *Borealis*	Photonic	[15]	Weakly refuted by [8]	Sec. II D 1
Apr 21 2023	RCS	67	32	Google *Sycamore*	Superconducting	[11]	Unrefuted	Sec. II E
Apr 21 2023	RCS	70	24	Google *Sycamore*	Superconducting	[11]	Unrefuted	Sec. II E
Apr 24 2023	GBS	50	144	USTC *Jiǔzhāng 3.0*	Photonic	[16]	Weakly refuted by [8]	Sec. II F
Jun 14 2023	QSim	127	60	IBM *Kyiv*	Superconducting	[17]	Refuted by [18–22]	Sec. II G
Mar 01 2024	QSim	567	–	D-Wave *ADV1/2*	Annealing	[23]	Unrefuted	Sec. II H

LaRose, Ryan. "A brief history of quantum vs classical computational advantage." arXiv preprint arXiv:2412.14703 (2024).

- All the results are heuristic approaches
- Mind the gap between the theoretical and practical results. What's going wrong?

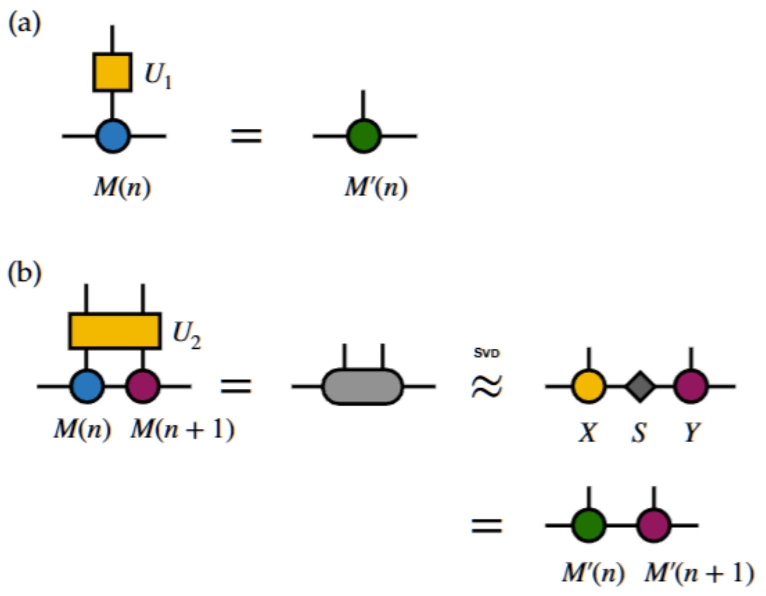
Noise makes classical simulation easier

- Recall that Google's result: LXEB≈ 1.002, while a perfect QC should be LXEB$= 2$.

- **Zhou et al:** The first classical spoofing result that directly consider the noisy quantum circuit model

  - A heuristic tensor network algorithm using "low-rank approximation"

  - **Intuition:** treat truncation as analogue of noise in a QC

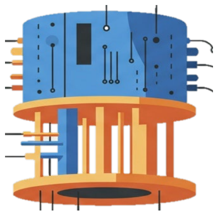


Noise makes classical simulation easier

- Recall that Google's result: LXEB≈ 1.002, while a perfect QC should be LXEB$= 2$.

- **Zhou et al:** The first classical spoofing result that directly consider the noisy quantum circuit model

- **Aharonov et al:** Theoretical result showing that the output distribution of a noisy random quantum circuit can be approximately sampled using a classical computer within ϵ-TV distance in $\mathrm{poly}(n, 1/\epsilon)$ time, under some assumptions.



M samples

0111100101
1010000111
0011100010
1100010101
0101101011

No statistical test can tell the difference

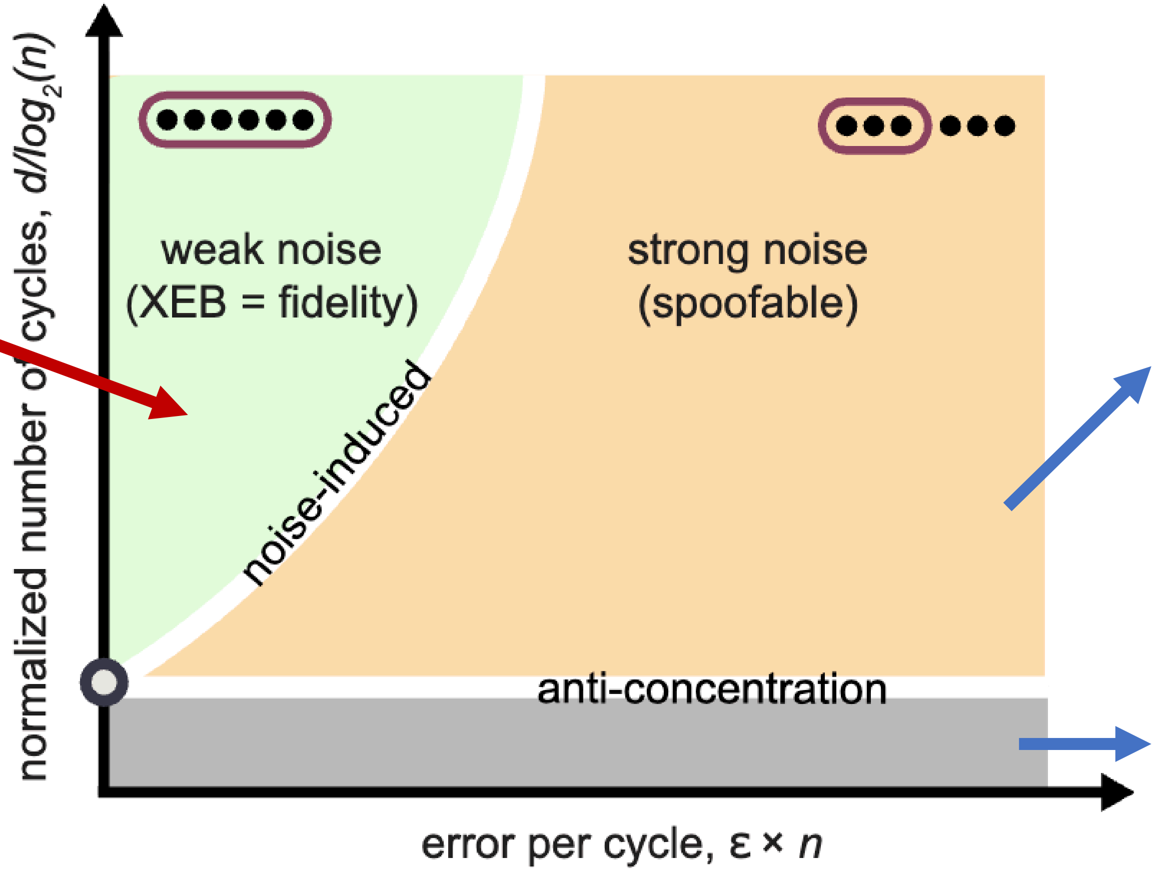
1101100101
1110100101
1011101011
1001011011
1010110100

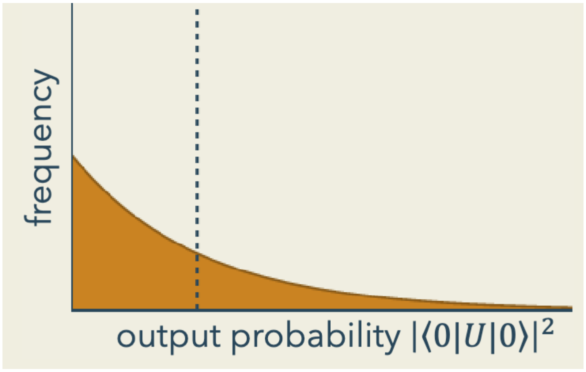
$\mathrm{poly}(M, n)$ time

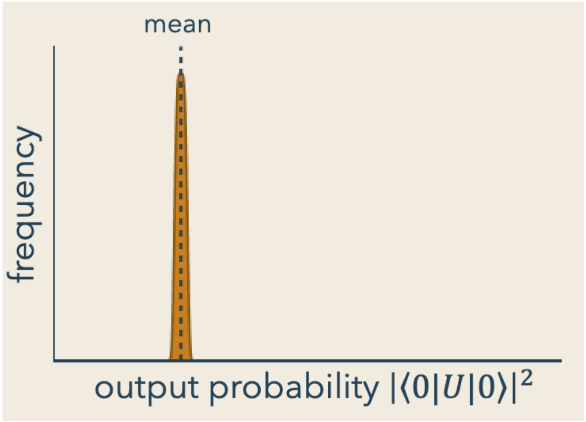
Noise makes classical simulation easier

- **Google in 2023:** They conducted new supremacy experiments with 67 qubits, and experimentally demonstrated that the noise-induced phase transition in RCS

Good regime for supremacy







Recap

Quantum supremacy experiments showcases the algorithmic lens on data science:

Physicists and engineers (a.k.a. QC builders) brought up a new problem

Practitioners developed many heuristics that are successful in practice

Theorists tried to understand when and why these heuristics work, and developed new algorithms with rigorous guarantees

Looking ahead

Classical world

- Tensor methods: tensor decompositions and applications
- Spectral estimation and super-resolution
- Sum-of-Squares (SoS)
- Semi-definite programming (SDP) solvers
- MCMC and diffusion model

Quantum world

- Quantum eigenvalue problems
- Quantum linear algebra
- Quantum sampling
- Quantum learning theory