QC WARE

Q2B:
Quantum Computing for Business

John Preskill

5 December 2017

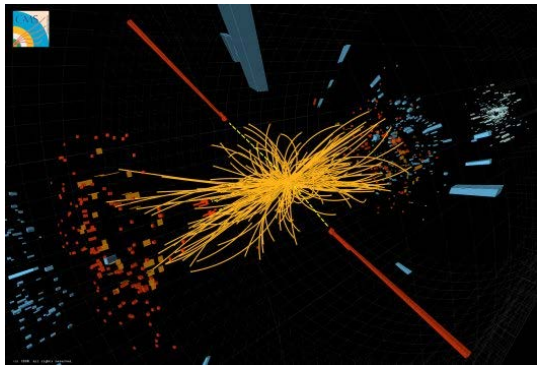
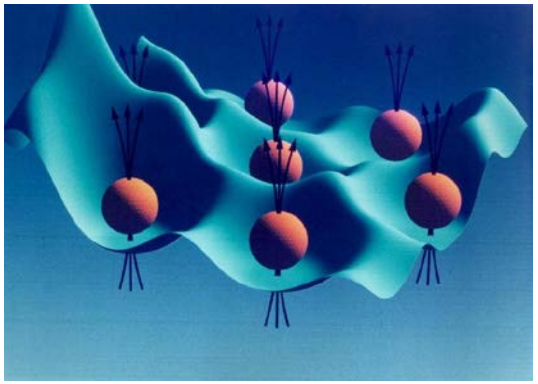
Quantum Computing in the NISQ Era and Beyond*

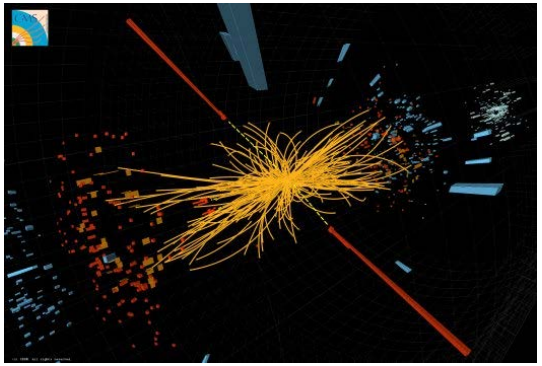
John Preskill

5 December 2017

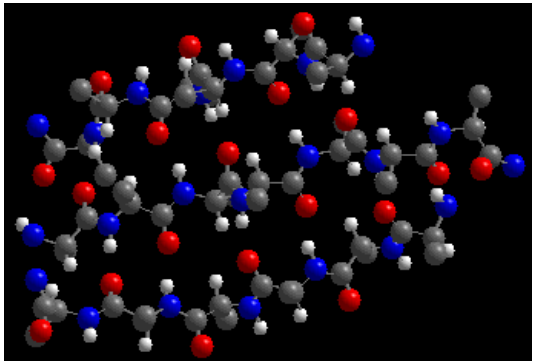
**Noisy Intermediate-Scale Quantum Computer*

Frontiers of Physics

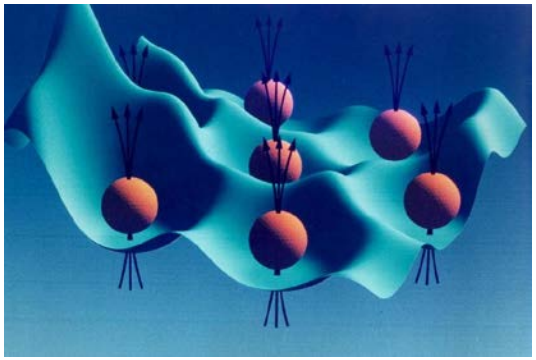
short distance	long distance	complexity
		
Higgs boson	Large scale structure	“More is different”
Neutrino masses	Cosmic microwave background	Many-body entanglement
Supersymmetry	Dark matter	Phases of quantum matter
Quantum gravity	Dark energy	Quantum computing
String theory	Gravitational waves	Quantum spacetime



particle collision



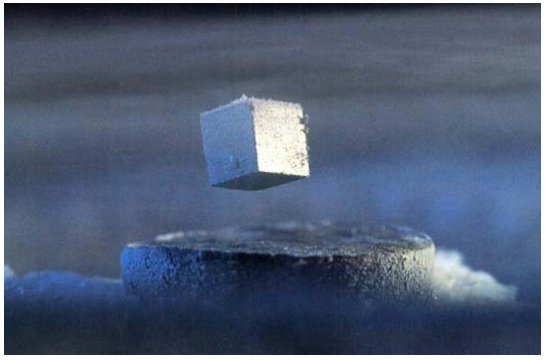
molecular chemistry



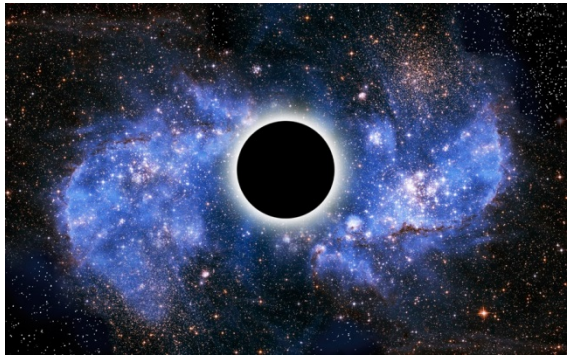
entangled electrons

A quantum computer can simulate efficiently any physical process that occurs in Nature.

(Maybe. We don't actually know for sure.)



superconductor



black hole

early universe

Two fundamental ideas

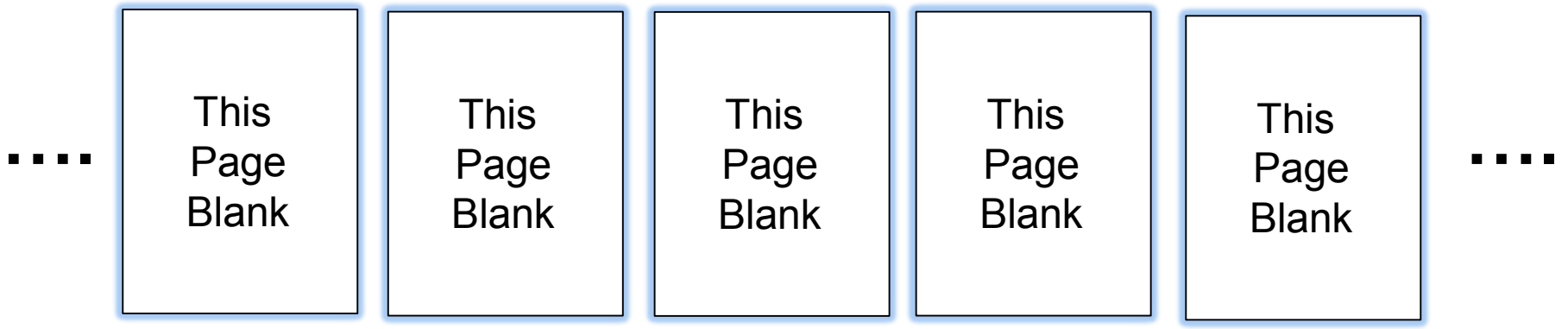
(1) *Quantum complexity*

Why we think quantum computing is powerful.

(2) *Quantum error correction*

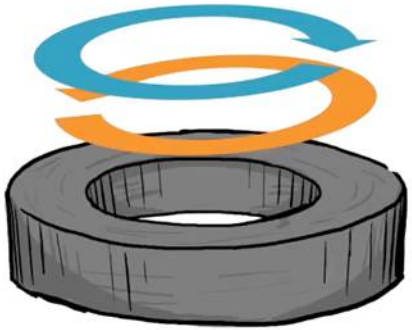
Why we think quantum computing is scalable.

Quantum entanglement



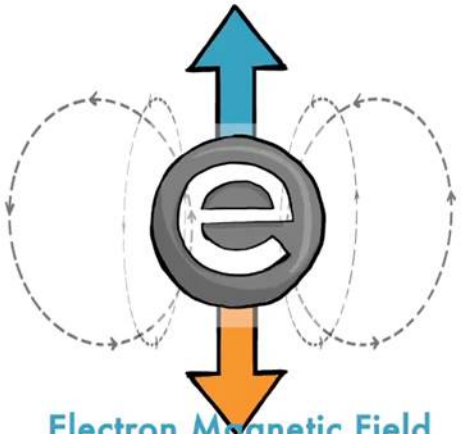
Nearly all the information in a typical entangled “quantum book” is encoded in the correlations among the “pages”.

You can't access the information if you read the book one page at a time.

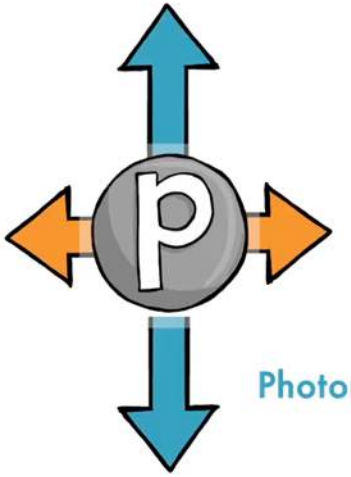


Persistent current in a superconducting circuit

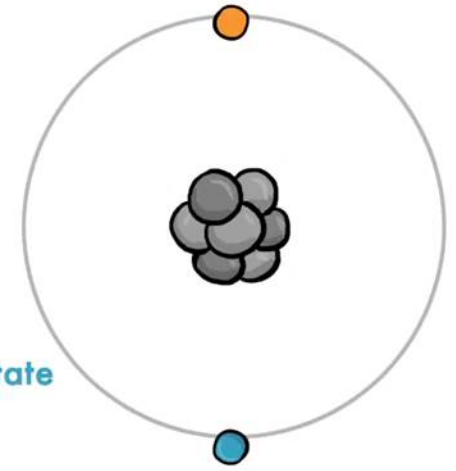
QUBIT



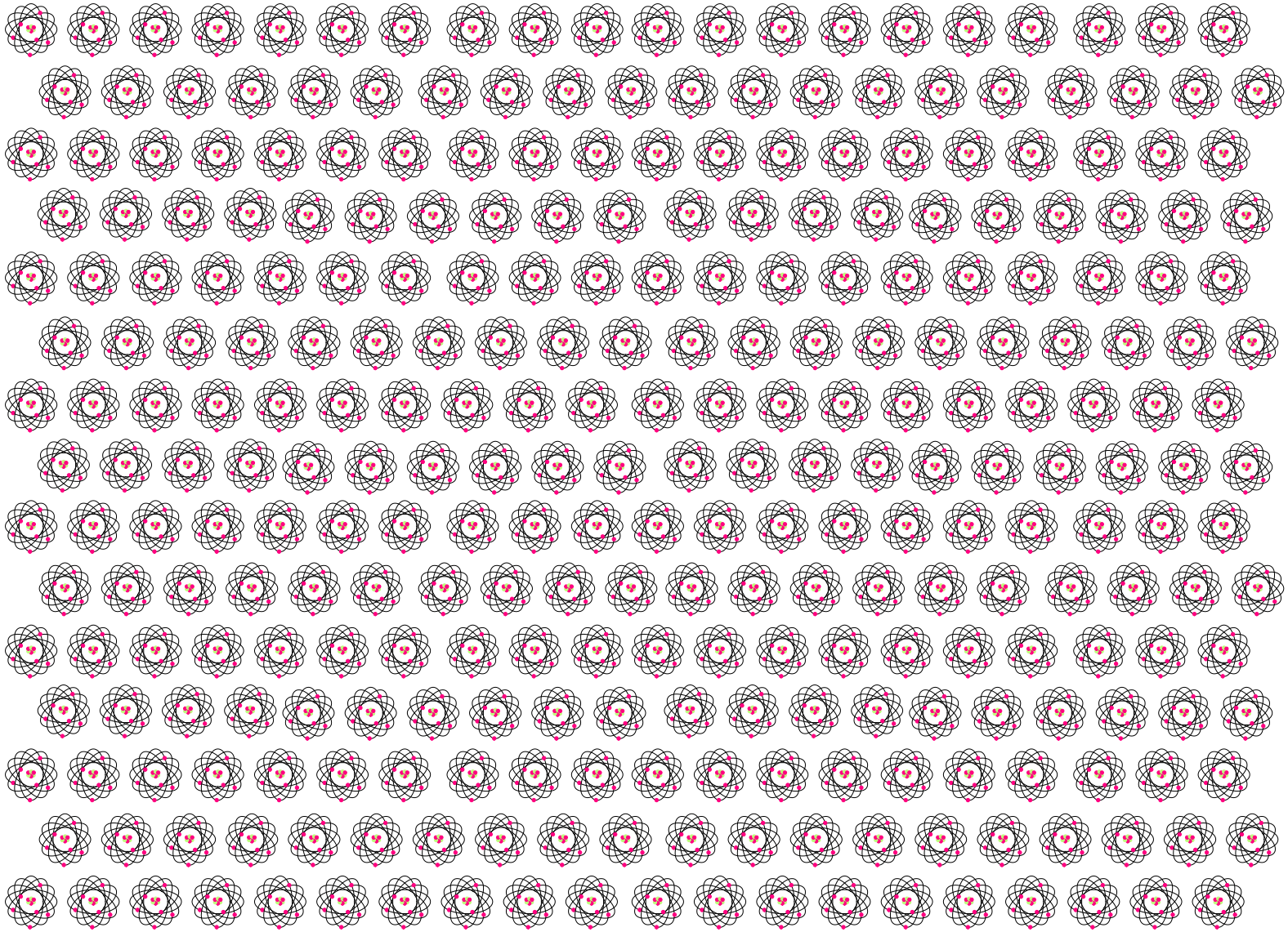
Electron Magnetic Field



Photon polarization



Atom Internal State



A complete description of a typical quantum state of just 300 qubits requires more bits than the number of atoms in the visible universe.

Why we think quantum computing is powerful

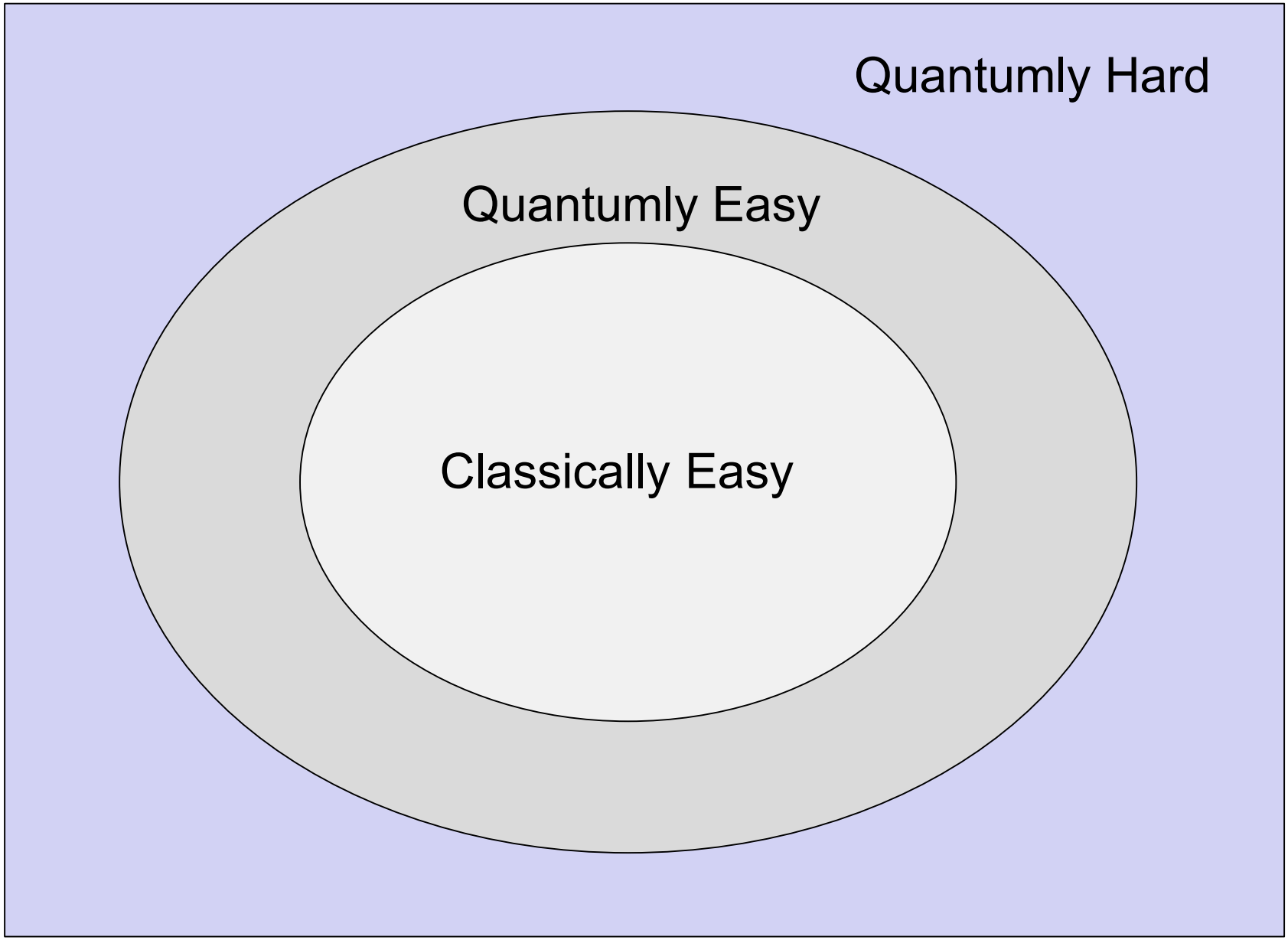
We know examples of problems that can be solved efficiently by a quantum computer, where we believe the problems are hard for classical computers. Factoring is the best known example. No efficient classical algorithm for factoring is known, and not for lack of trying. Factoring numbers which are thousands of bits long is out of reach classically, yet eventually will be feasible quantumly.

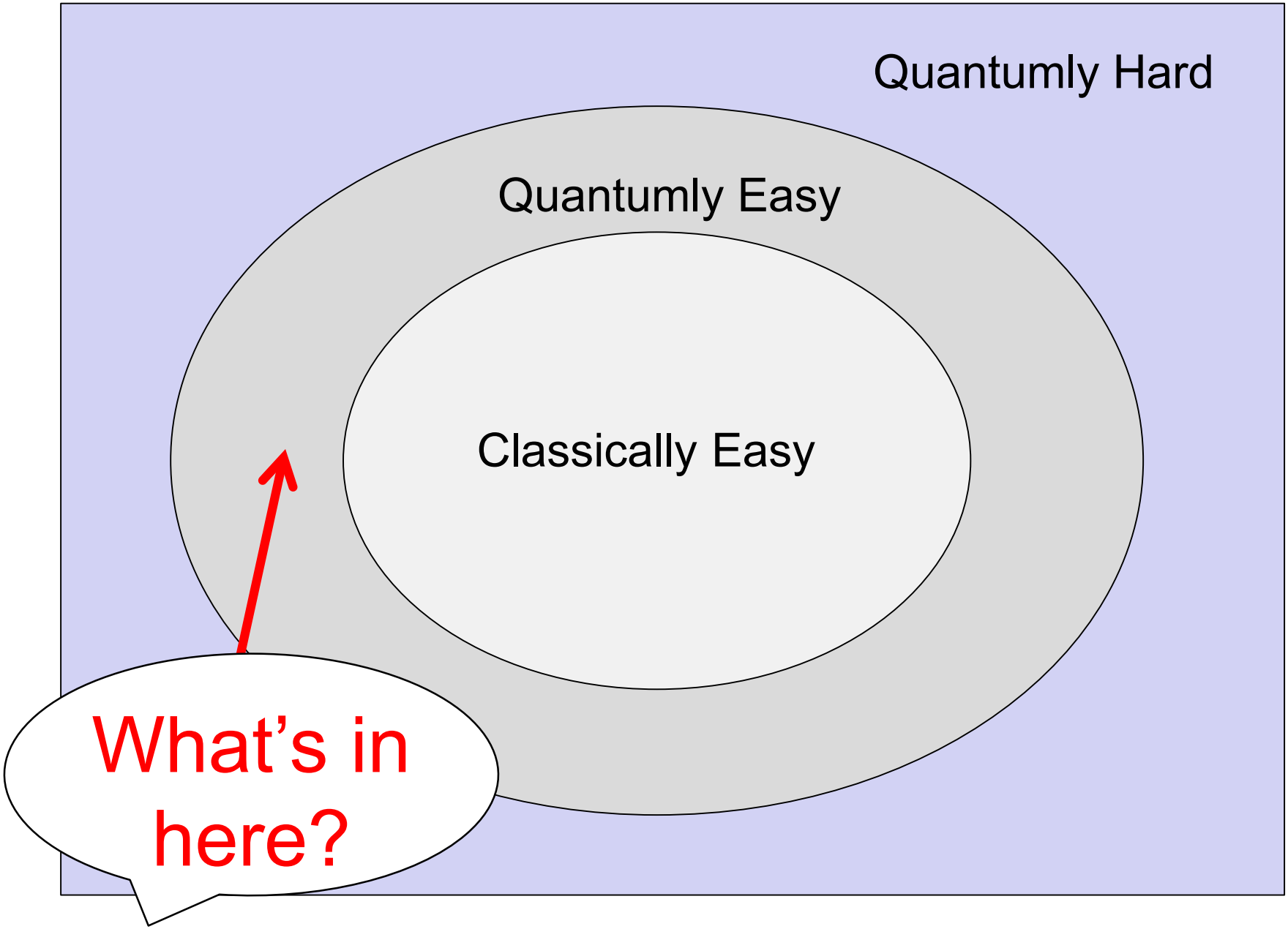
Consider the probability distribution of measurement outcomes for n-qubits in a quantum computer. Complexity theory arguments, based on plausible assumptions, indicate that no efficient classical algorithm can efficiently sample from this distribution.

We don't know how to simulate a quantum computer efficiently using a digital ("classical") computer. It is not for lack of trying. The cost of the best simulation algorithm rises exponentially with the number of qubits.

The power of quantum computing is limited. For example, we don't believe that quantum computers can efficiently solve worst-case instances of NP-hard optimization problems (e.g., the traveling salesman problem).

Problems

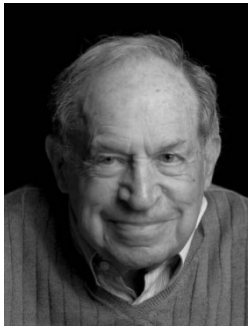


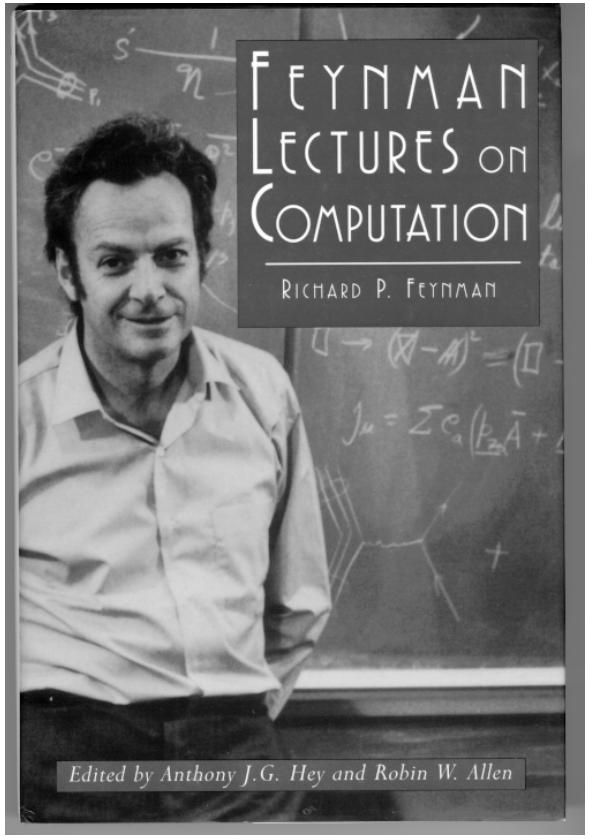

Problems
Quantumly Hard
Quantumly Easy
Classically Easy
What's in here?

“The theory of everything?”

“The Theory of Everything is not even remotely a theory of every thing … We know this equation is correct because it has been solved accurately for small numbers of particles (isolated atoms and small molecules) and found to agree in minute detail with experiment. However, it cannot be solved accurately when the number of particles exceeds about 10. No computer existing, or that will ever exist, can break this barrier because it is a catastrophe of dimension … We have succeeded in reducing all of ordinary physical behavior to a simple, correct Theory of Everything only to discover that it has revealed exactly nothing about many things of great importance.”

R. B. Laughlin and D. Pines, PNAS 2000.





"Nature isn't classical, dammit, and if you want to make a simulation of Nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem because it doesn't look so easy."

R. P. Feynman, 1981

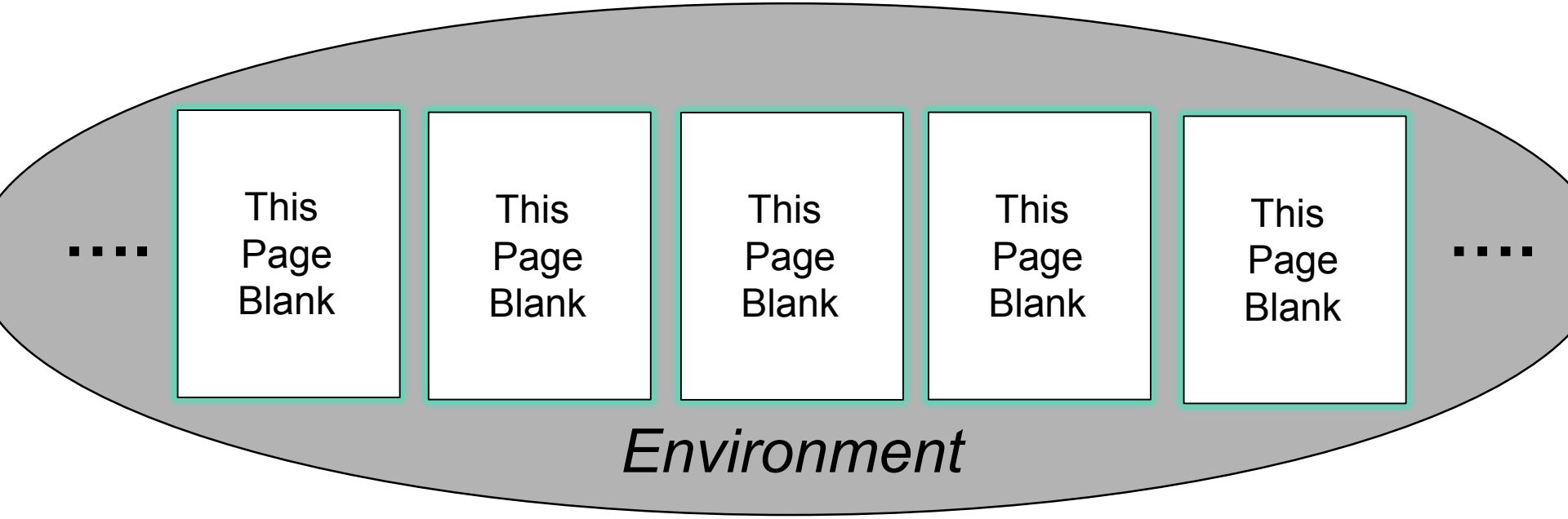
Why quantum computing is hard

We want qubits to interact strongly with one another.

We don’t want qubits to interact with the environment.

Until we measure them.

Quantum error correction



The protected “logical” quantum information is encoded in a highly entangled state of many physical qubits.

The environment can't access this information if it interacts locally with the protected system.

The NISQ Era

The (noisy) 50-100 qubit quantum computer is coming soon.
(*NISQ* = noisy intermediate-scale quantum computer)

NISQ devices cannot be simulated by brute force using the most powerful currently existing supercomputers.

NISQ will be an interesting tool for exploring physics. It *might* also have useful applications. But we're not sure about that.

NISQ will not change the world by itself. Rather it is a step toward more powerful quantum technologies of the future.

Potentially transformative scalable quantum computers may still be decades away. We're not sure how long it will take.

Qubit "quality"

The *number* of qubits is an important metric, but it is not the only thing that matters.

The *quality* of the qubits, and of the "quantum gates" that process the qubits, is also very important. All quantum gates today are noisy, but some are better than others. Qubit measurements are also noisy.

For today's *best* hardware (superconducting circuits or trapped ions), the probability of error per (two-qubit) gate is about 10^{-3}, and the probability of error per measurement is about 10^{-2} (or better for trapped ions). We don't yet know whether systems with many qubits will perform that well.

Naively, we cannot do many more than 1000 gates (and perhaps not even that many) without being overwhelmed by the noise. Actually, that may be too naïve, but anyway the noise limits the computational power of NISQ technology.

Eventually we'll do much better, either by improving (logical) gate accuracy using quantum error correction (at a hefty overhead cost) or building much more accurate physical gates, or both. But that probably won't happen very soon.

Other important features: The *time* needed to execute a gate (or a measurement). E.g., the two-qubit gate time is about 40 ns for superconducting qubits, 100 μs for trapped ions, a significant difference. Also qubit connectivity, fabrication yield, …

What I won't say much about

Quantum-resistant public key cryptography. New classical protocols for protecting our privacy when quantum computers are in widespread use and e.g. RSA (based on hardness of factoring) is obsolete.

Quantum networks, quantum repeaters, and quantum key distribution. Distributing quantum entanglement around the world. That might be done to achieve *quantum key distribution* for secure communication, or perhaps for other purposes like remotely sharing quantum devices.

Quantum sensing. Quantum devices (based for example on defects in diamond) achieving higher sensitivity and spatial resolution than other sensors, with potential applications to biology, medicine, magnetometry, accelerometry, gravimetry, etc.

Technical advances in quantum *computing* hardware may also enable new applications for quantum *networks* and *sensors* (and vice versa).

Quantum Speedups?

When will quantum computers solve important problems that are beyond the reach of the post powerful classical supercomputers?

We should compare with post-exascale classical hardware, e.g. 10 years from now, or more ($> 10^{18}$ FLOPS).

We should compare with the best classical algorithms for the same tasks.

Note that, for problems outside NP (e.g typical quantum simulation tasks), validating the performance of the quantum computer may be difficult.

Even if classical supercomputers can compete, the quantum computer might have advantages, e.g. lower cost and/or lower power consumption.

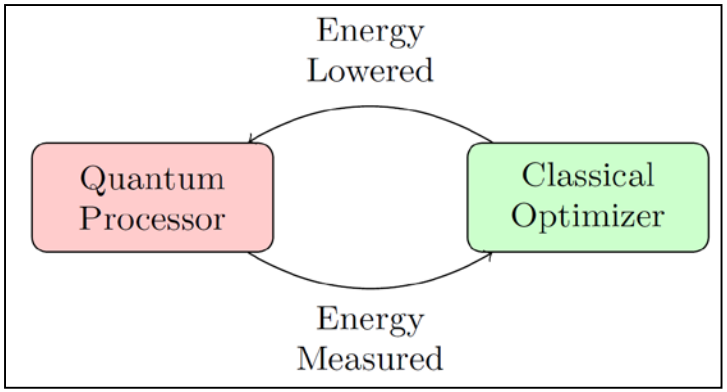
Quantum optimizers

Eddie Farhi: "Try it and see if it works!"

We don't expect a quantum computer to solve worst case instances of NP-hard problems, but it might find better approximate solutions, or find them faster.

Hybrid quantum/classical algorithms.

Combine quantum evaluation of an expectation value with a classical feedback loop for seeking a quantum state with a lower value.



Quantum approximate optimization algorithm (QAOA).

In effect, seek low-energy states of a classical spin glass.

Variational quantum eigensolvers (VQE).

Seek low energy states of a quantum many-body system with a local Hamiltonian H. (Much easier than algorithms which require simulation of time evolution governed by H.)

Classical optimization algorithms (for both classical and quantum problems) are sophisticated and well-honed after decades of hard work. Will NISQ be able to do better?

How quantum testbeds might help

Peter Shor: "You don't need them [testbeds] to be big enough to solve useful problems, just big enough to tell whether you can solve useful problems."

Classical examples:
Simplex method for linear programming: experiments showed it's fast long before theorists could prove that it's fast.

Metropolis algorithm: experiments showed it's useful for solving statistical physics problems before theory established criteria for rapid convergence.

Deep learning. Mostly tinkering so far, without much theory input.

Possible quantum examples:
Quantum annealers, approximate optimizers, variational eigensolvers, … playing around may give us new ideas.

But in the NISQ era, imperfect gates will place severe limits on circuit size. In the long run, quantum error correction will be needed for scalability. In the near term, better gates might help a lot!

What can we do with, say, < 100 qubits, depth < 100? We need a dialog between quantum algorithm experts and application users.

Quantum annealing

The D-Wave machine is a (very noisy) 2000-qubit *quantum annealer* (QA), which solves optimization problems. It *might* be useful. But we have no convincing theoretical argument that QAs are useful, nor have QA speedups been demonstrated experimentally.

QA is a noisy version of adiabatic quantum computing (AQC), and we believe AQC is powerful. Any problem that can be solved efficiently by noiseless quantum computers can also be solved efficiently by noiseless AQC, using a "circuit-to-Hamiltonian map."

But in contrast to the quantum circuit model, we don't know whether noisy AQC is scalable. Furthermore, the circuit-to-Hamiltonian map has high overhead: Many more qubits are needed by the (noiseless) AQC algorithm than by the corresponding quantum circuit algorithm which solves the same problem.

Theorists are more hopeful that a QA can achieve speedups if the Hamiltonian has a "sign problem" (is "non-stoquastic"). Present day QAs are stoquastic, but non-stoquastic versions are coming soon.

Assessing the performance of QA may already be beyond the reach of classical simulation, and theoretical analysis has not achieved much progress. Further experimentation should clarify whether QAs actually achieve speedups relative to the best classical algorithms.

QAs can also be used for solving quantum simulation problems rather than classical optimization problems (D-Wave, unpublished).

Noise-resilient quantum circuits

For near-term applications, noise-resilience is a key consideration in quantum circuit design (Kim 2017).

For a generic circuit with G gates, a single faulty gate might cause the circuit to fail. If the probability of error per gate is not much larger than 1/G, we have a reasonable chance of getting the right answer.

But, depending on the nature of the algorithm and the circuit that implements it, we might be able to tolerate a much larger gate error rate.

For some physical simulation problems, a constant probability of error per measured qubit can be tolerated, and the number of circuit locations where a fault can cause an error in a particular qubit is relatively small. This could happen because the circuit has low depth, or because an error occurring at an earlier time decays away by a later time.

Circuits with good noise-resilience (based on tensor network constructions like MERA) are among those that might be useful for solving quantum optimization problems using variational quantum eigensolvers (VQE), improving the prospects for outperforming classical methods during the NISQ era (Kim and Swingle 2017).

Quantum machine learning?

(Classical) deep learning, e.g. restricted Boltzmann machines with multiple hidden layers between input and output. Millions of coupling parameters, optimized on a training set to achieve the proper relation between input and output.

Deep learning may be either unsupervised (unlabeled training set), or supervised (e.g. learning to identify photos).

High-dimensional classical data can be encoded very succinctly in a quantum state. In principle log N qubits suffice to represent a N-dimensional vector. Such "quantum Random Access Memory" (= qRAM) *might* have advantages for deep learning applications.

However, quantum deep learning is hampered by input/output bottlenecks.

Perhaps a quantum deep learning network can be trained more efficiently, e.g. using a smaller training set. We don't know. We'll have to try it to see how well it works.

Might be achieved by a (highly controllable) quantum annealer, or other custom quantum device unsuited for general purpose quantum computing. How robust to noise?

Perhaps more natural to consider quantum inputs / outputs; e.g. better ways to characterize or control quantum systems. Quantum networks might have advantages for learning about *quantum* correlations, rather than classical ones.

Classical deep learning has many applications to quantum science and technology.

Quantum linear algebra

qRAM: an N-component vector b can be encoded in a quantum state $|b\rangle$ of log N qubits.

Given a classical N X N input matrix A, which is sparse and well-conditioned, and the quantum input state $|b\rangle$, the HHL (Harrow, Hassidim, Lloyd 2008) algorithm outputs the quantum state $|y\rangle = |A^{-1} b\rangle$, with a small error, in time O(log N). The quantum speedup is exponential in N.

Input vector $|b\rangle$ and output vector $|y\rangle = |A^{-1} b\rangle$ are quantum! We can sample from measurements of $|y\rangle$.

If the input b is classical, we need to load $|b\rangle$ into qRAM in polylog time to get the exponential speedup (which might not be possible). Alternatively the input b may be computed rather than entered from a database.

HHL is BQP-complete: It solves a (classically) hard problem unless BQP=BPP.

Example: Solving (monochromatic) Maxwell's equations in a complex 3D geometry; e.g., for antenna design (Clader et al. 2013). Discretization and preconditioner needed. How else can HHL be applied?

HHL is not likely to be feasible in the NISQ era.

Quantum simulation

We're confident *strongly correlated* (highly entangled) materials and large molecules are hard to simulate classically (because we have tried hard and have not succeeded).

Quantum computers will be able to do such simulations, though we may need to wait for scalable fault tolerance, and we don't know how long that will take.

Potential (long-term) applications include pharmaceuticals, solar power collection, efficient power transmission, catalysts for nitrogen fixation, carbon capture, etc. These are not likely to be fully realized in the NISQ era.

Classical computers are especially bad at simulating quantum *dynamics* --- predicting how highly entangled quantum states change with time. Quantum computers will have a big advantage in this arena. Physicists hope for noteworthy advances in quantum dynamics during the NISQ era.

For example: Classical *chaos theory* advanced rapidly with onset of numerical simulation of classical dynamical systems in the 1960s and 1970s. Quantum simulation experiments may advance the theory of *quantum* chaos. Simulations with ~ 100 qubits could be revealing, if not too noisy.

Digital vs. Analog quantum simulation

An *analog quantum simulator* is a quantum system of many qubits whose dynamics resembles the dynamics of a model system we wish to study. A *digital quantum simulator* is a gate-based universal quantum computer, which can be used to simulate any physical system of interest when suitably programmed.

Analog quantum simulation has been an active research area for 15 years or more; digital quantum simulation is just getting started now.

Analog platforms include: ultracold (neutral) atoms and molecules, trapped ions, superconducting circuits, etc. These same platforms can be used for circuit-based computation as well.

Although they are becoming more sophisticated and controllable, analog simulators are limited by imperfect control. They are best suited for studying "universal" properties of quantum systems which are hard to access in classical simulations, yet sufficiently robust to be accessible using noisy quantum systems.

Eventually, digital (circuit-based) quantum simulators will surpass analog quantum simulators for studies of quantum dynamics, but perhaps not until fault tolerance is feasible.

The steep climb to scalability

Long-lived logical qubits, protected by quantum error correction, are likely to be realized in the next few years.

But NISQ-era quantum algorithms will need to tolerate noise. Fully fault-tolerant quantum computing may still be decades away. We don't really know how long it will take … We may need platforms supporting millions of physical qubits, or more, a very big leap from where we are now. ("Quantum Chasm" ≅ "NISQ Risk")

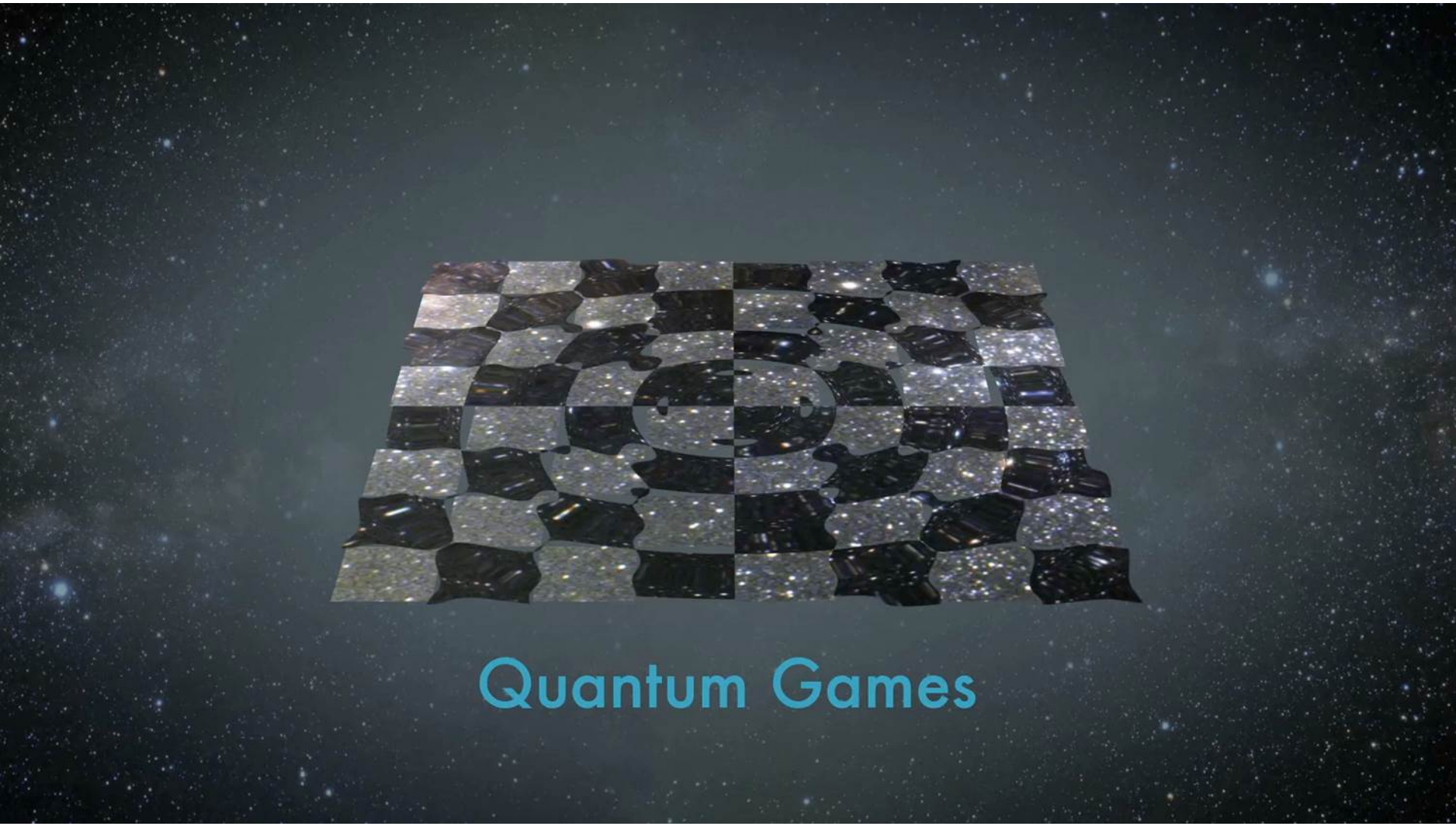
Lower gate error rates will substantially reduce the overhead cost of fault tolerance, and also extend the reach of quantum algorithms which do not use error correction. Topological quantum computing (being aggressively pursued by Microsoft) is one aspirational approach to achieving much lower error rates.

Platforms with faster gates have shorter time to solution, all else being equal. This speed advantage will become more important in the longer term.

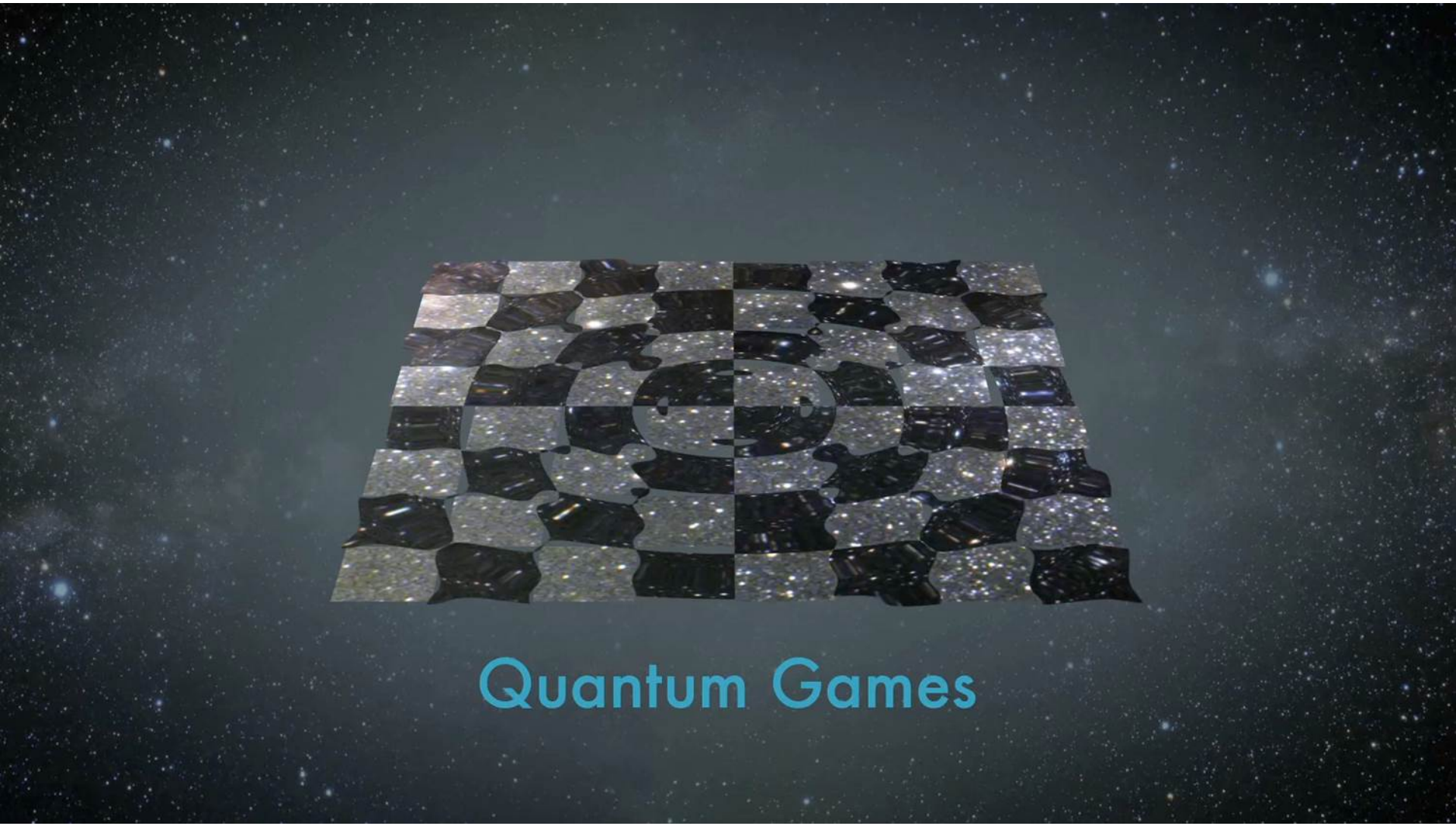
Significant advances, in both basic quantum science and systems engineering, will be needed to achieve scalable FTQC. Because we have so far to go, new insights and developments could substantially alter the outlook for scalability.

We tend to be too optimistic about the short run, too pessimistic about the long run.

Quantum Games



In the future, the world's leading physicists, having played quantum games since age 3, will be unable to understand why 20th century physicists thought quantum mechanics is weird.



Quantum gaming will be a natural arena for quantum machine learning.

Quantum speedups in the NISQ era and beyond

Can noisy intermediate-scale quantum computing (NISQ) surpass exascale classical hardware running the best classical algorithms?

Near-term quantum advantage for useful applications is possible, but not guaranteed.

Hybrid quantum/classical algorithms (like QAOA and VQE) can be tested.

Near-term algorithms should be designed with noise resilience in mind.

Quantum dynamics of highly entangled systems is especially hard to simulate, and is therefore an especially promising arena for quantum advantage.

Experimentation with quantum testbeds may hasten progress and inspire new algorithms.

NISQ will not change the world by itself. Realistically, the goal for near-term quantum platforms should be to pave the way for bigger payoffs using future devices.

Lower quantum gate error rates will lower the overhead cost of quantum error correction, and also extend the reach of quantum algorithms which do not use error correction.

Truly transformative quantum computing technology may need to be fault tolerant, and so may still be far off. But we don't know for sure how long it will take. Progress toward fault-tolerant QC must continue to be a high priority for quantum technologists.

Additional Slides

Fifteen Years of Growth in QIS

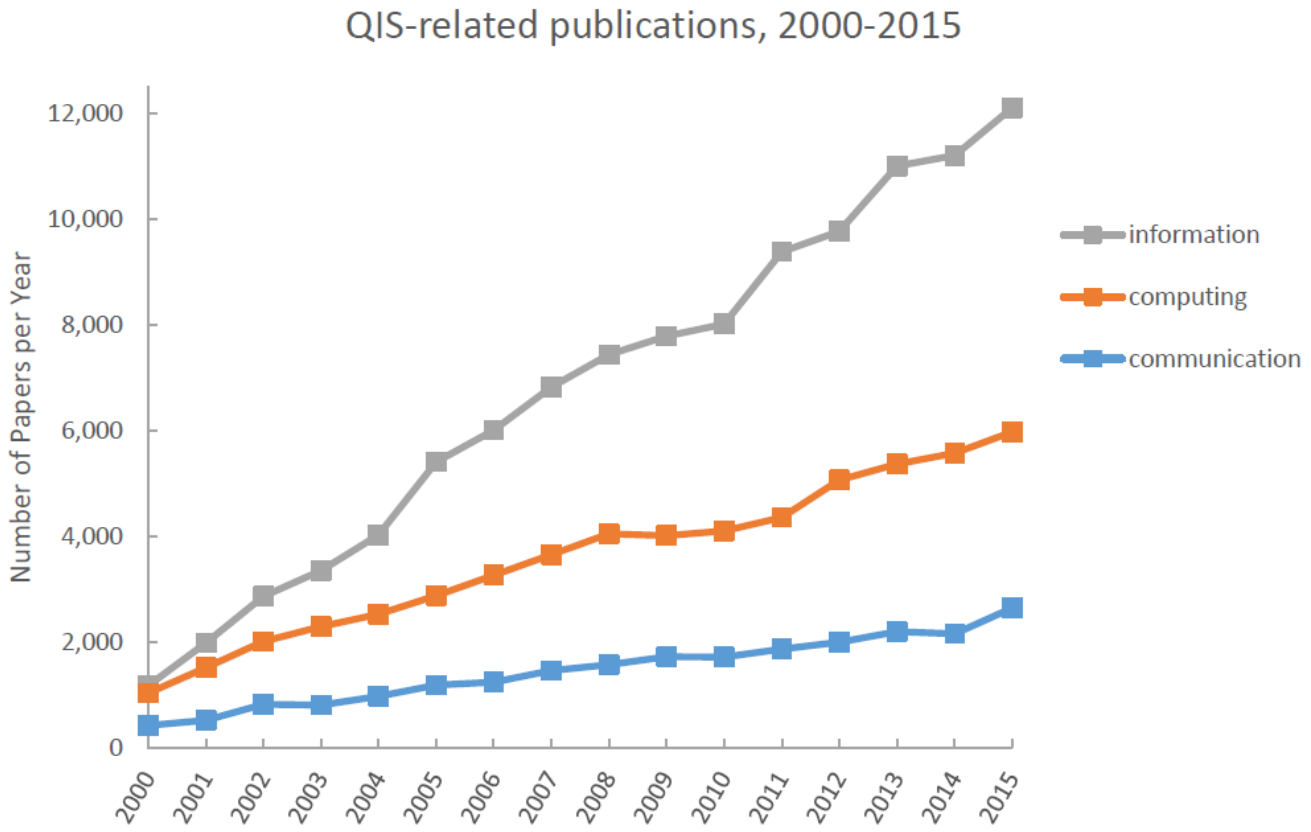


Figure 1. A Google Scholar search for publications (excluding patents and citations) containing the terms "quantum information," "quantum computing," and "quantum communication" illustrates how the field has grown in recent years.

Source: ADVANCING QUANTUM INFORMATION SCIENCE: NATIONAL CHALLENGES AND OPPORTUNITIES Produced by the Interagency Working Group on Quantum Information Science of the Subcommittee on Physical Sciences, National Science and Technology Council, July 2016

Best classical algorithms: cautionary tales

Boson sampling: From 30 photons and 500 modes to 50 photons and 2500 modes (Neville et al. 2017).

Random circuits: Simulating 49 qubits with TB rather then PB memory (IBM 2017) --- trading depth and space.

Best approximation ratio for Max E3LIN2 (with bounded occurence D) achieved by QAOA at level p=1 (Farhi et al. 2014), later surpassed by classical all-star team.

D-Wave evidence for constant factor speedup weakens when quantum annealer is compared with better classical algorithms.

Randomized classical matrix inversion can compete with quantum in *some* parameter regimes (Le Gall).

Tensor network methods for quantum many-body physics and chemistry keep improving (MPS, PEPS, MERA, tensor RG).

Are physically relevant quantum problems really classically hard, even if BQP $\neq$ BPP?

Dynamics seems promising, but MBL (many-body localization) may be classically easy, and ETH (eigenstate thermalization hypothesis = strong quantum chaos) may be physically boring (wisecrack by Frank Verstraete).

Quantum hardware: state of the art

IBM Quantum Experience in the cloud: now 16 qubits (superconducting circuit). 20 qubits by end of 2017, 50-qubit device "built and measured."

Google 22-qubit device (superconducting circuit), 49 qubits next year.

Harvard 51-qubit quantum simulator (Rydberg atoms in optical tweezers). Dynamical phase transition in Ising-like systems; puzzles in defect (domain wall) density.

UMd 53-qubit quantum simulator (trapped ions). Dynamical phase transition in Ising-like systems; high efficiency single-shot readout of many-body correlators.

ionQ: 32-qubit processor planned (trapped ions), with all-to-all connectivity.

Microsoft: is 2018 the year of the Majorana qubit?

And many other interesting platforms … spin qubits, defects in diamond (and other materials), photonic systems, …

There are other important metrics besides number of qubits; in particular, the two-qubit gate error rate (currently $> 10^{-3}$) determines how large a quantum circuit can be executed with reasonable signal-to-noise.

Speeding up semidefinite programs (SDPs)

Given N X N Hermitian matrices C, $\{A_1, \ldots, A_m\}$ and real numbers $\{b_1, \ldots, b_m\}$, maximize $\mathrm{tr}(CX)$ subject to $\mathrm{tr}(A_i X) \leq b_i$, $X \geq 0$.

Many applications, classically solvable in poly(m,N) time.

Suffices to prepare (and sample from) Gibbs state for H = linear comb. of input matrices. Quantum time polylog(N) if Gibbs state can be prepared efficiently (Brandão & Svore 2016). Output is a quantum state $\rho \cong X$.

When can the Gibbs state be prepared efficiently?
-- H thermalizes efficiently.
-- Input matrices are low rank (Brandão et al. 2017).

Can be viewed as a version of quantum annealing (QA) where Hamiltonian is quantum instead of classical, and where the algorithm is potentially robust with respect to small nonzero temperature.

The corresponding Gibbs state can be prepared efficiently only for SDPs with special properties. What are the applications of these SDPs?

Applications of quantum linear algebra

Given classical input A (N X N matrix, sparsity s and condition number κ) and N-qubit quantum input $|b\rangle$, algorithm outputs
$|y\rangle = |A^{-1} b\rangle$ with error ε.

It is more promising if the input b is computed rather than entered from a database.

Example: Solving (monochromatic) Maxwell's equations in a complex 3D geometry; e.g., for antenna design (Clader et al. 2013). Discretization and preconditioner needed.

Alternative method for solving classical scattering problems: quantum simulation of N X N Laplacian using O(log N) qubits (Jordan et al. 2017). Need efficient preparation of initial state (e.g. input Gaussian wavepacket).

Recommendation systems (e.g. Netflix/Amazon with $m=10^8$ users and $n=10^6$ products). Sample rapidly from preference matrix with *low-rank* $k \cong 100$ (Kerenidis & Prakash 2016). Quantum queries to a classical data structure: Linear-time offline preprocessing, online processing of quantum queries in time poly(k) polylog(mn).

Prototypical quantum simulation task

(1) State preparation. E.g., incoming scattering state.

(2) Hamiltonian evolution. E.g. Trotter approximation.

(3) Measure an observable. E.g., a simulated detector.

Goal: sample accurately from probability distribution of outcomes.

Determine how computational resources scale with: error, system size, particle number, total energy of process, energy gap, …

Resources include: number of qubits, number of gates, …

Hope for polynomial scaling! Or even better: polylog scaling.

Need an efficient preparation of initial state.

Approximating a continuous system incurs discretization cost (smaller lattice spacing improves accuracy).

What should we simulate, and what do we stand to learn?

QC WARE