EE5340

INTRODUCTION TO QUANTUM COMPUTING AND PHYSICAL BASICS OF COMPUTING

On Quantum “Supremacy”

Ulya Karpuzcu

Quantum “Supremacy”

“The first use of a quantum computer to make a calculation much faster than we know how to do it with even the fastest supercomputers available.” [Preskill]

- The calculation doesn’t need to be useful…

- More “formal” definition

Any supremacy experiment must have a

1. Well-defined computational task (not necessarily useful)
2. Plausible quantum algorithm for the problem
3. Fair resource allocation (in time and space) to classical competitor
4. Complexity-theoretic assumption
5. Verification method

Why Supremacy?

Refute the Extended Church-Turing (ECT) thesis:

Classical computers can simulate any physical process with polynomial overhead

Quantum "Supremacy"

"The first use of a quantum computer to make a calculation much faster than we know how to do it with even the fastest supercomputers available." [Preskill]

- The calculation doesn't need to be useful…

- More "formal" definition

Any supremacy experiment must have a

1. Well-defined computational task (not necessarily useful)
2. Plausible quantum algorithm for the problem
3. Fair resource allocation (in time and space) to classical competitor
4. Complexity-theoretic assumption
5. Verification method

- Example: Factoring

1. Well-defined computational task (and useful)
2. Shor's algorithm
3. Fair resource allocation (in time and space) to classical competitor
4. Classical computers cannot factor quickly
5. Successful operation can be easily verified

Factorization Example cont.

- Best estimate to factor a 2048-bit number
  - Approx. 4K qubits and 10^9 gates are required
  - More qubits and gates may be necessary for fault tolerance

Quantum "Supremacy"

"The first use of a quantum computer to make a calculation much faster than we know how to do it with even the fastest supercomputers available." [Preskill]

- The calculation doesn't need to be useful…

- More "formal" definition

Any supremacy experiment must have a

1. Well-defined computational task (not necessarily useful)
2. Plausible quantum algorithm for the problem
3. Fair resource allocation (in time and space) to classical competitor
4. Complexity-theoretic assumption
5. Verification method

- Example: Factoring

1. Well-defined computational task (and useful)
2. Shor's algorithm
3. Fair resource allocation (in time and space) to classical competitor
4. Classical computers cannot factor quickly
5. Successful operation can be easily verified

Verification

- Example: Factorization
  - Easy, if a large-scale quantum computer running Shor's existed
  - Check that the claimed factors multiplied to the correct number
- Very few problems are like factorization
  - Tractable on a quantum computer
  - Checkable on a classical computer
  - Not known whether efficiently solvable on a classical computer
- Test a smaller (classically simulable) part of experiment?
  - No guarantee
- Apply statistical tests to samples from output distribution?
  - Can check consistency with desired distribution
  - Hard to calculate many individual probabilities required in this case

Quantum "Supremacy"

"The first use of a quantum computer to make a calculation much faster than we know how to do it with even the fastest supercomputers available." [Preskill]

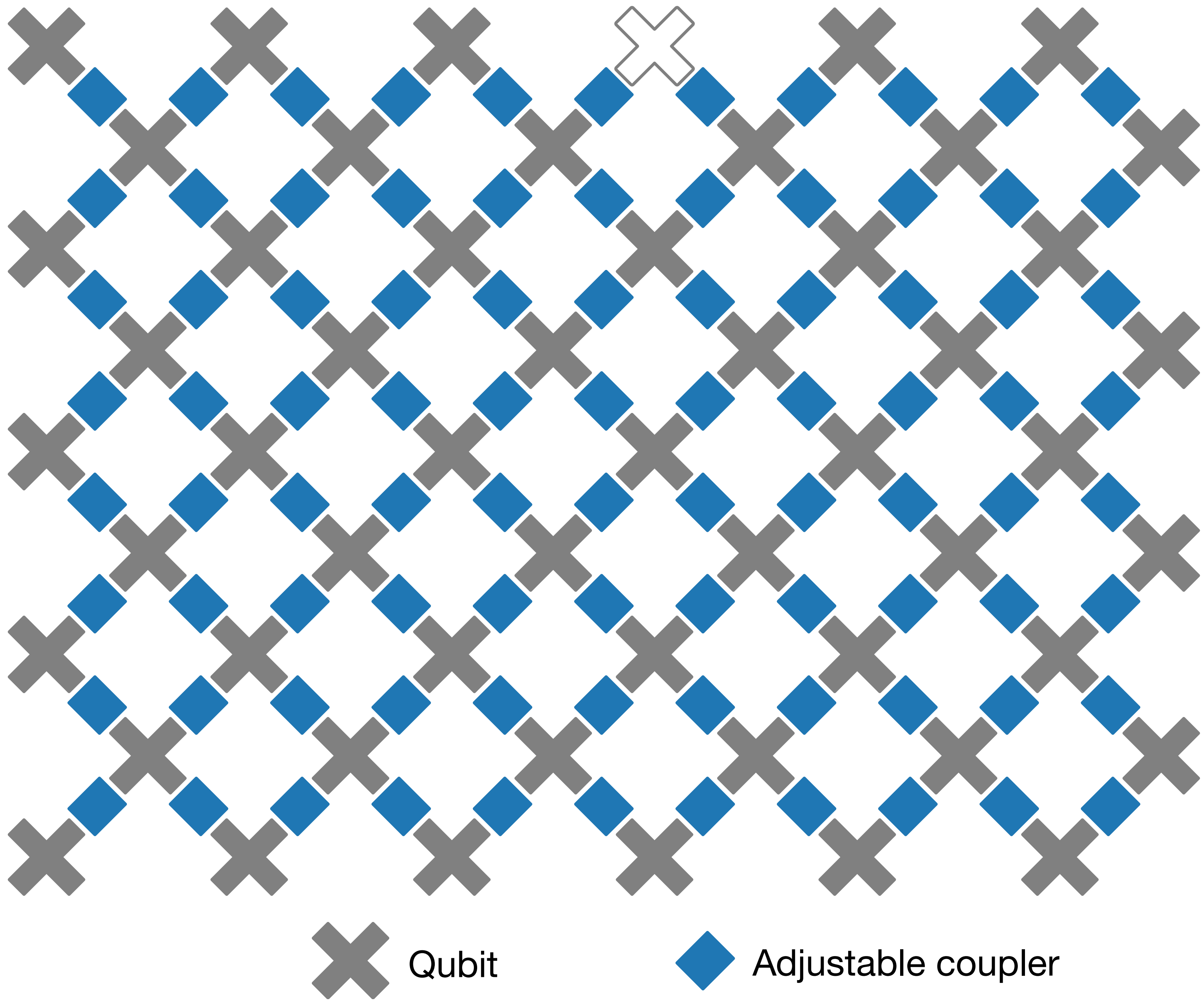
- The calculation doesn't need to be useful?

- How to build a high-fidelity quantum processor?
  - Running computational problems exponentially faster than a classical one
- Problems easy for a quantum computer but hard for a classical?

Quantum "Supremacy": Google Sycamore

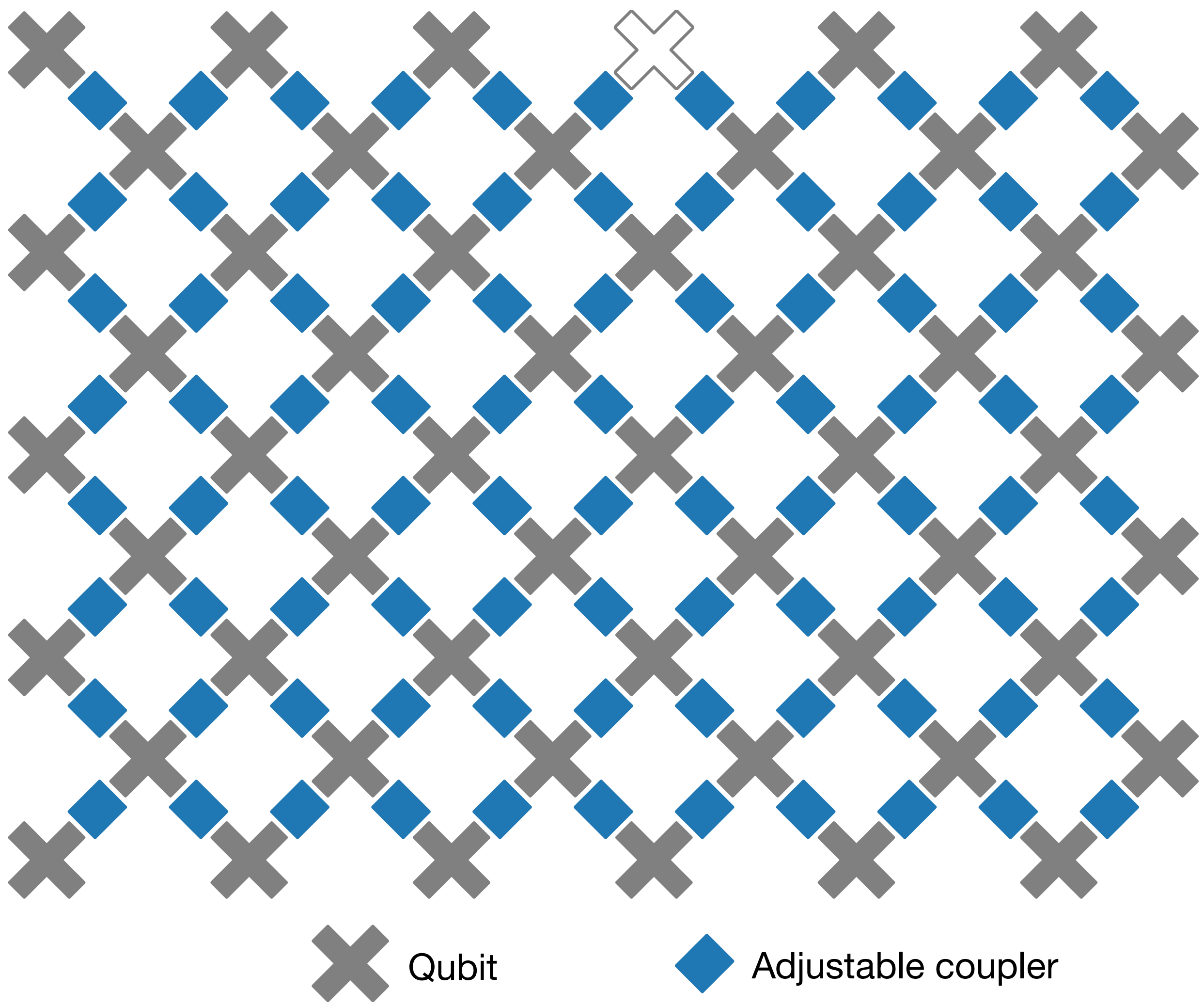
- Processor with programmable superconducting qubits (transmon)
- 53-qubits, 2^{53}-dimensional state-space
  - Arranged in a 2D grid
- Features fast, high-fidelity gates that can be executed in parallel
- Evaluated using "cross-entropy" benchmarking
- Can sample the output of a pseudo-random quantum circuit 1M times in 200secs
  - vs. 10K years on a classical computer
    - Memory requirement to store the entire state vector would be prohibitive
    - Hence, need to use an algorithm trading off space (storage) for time

Quantum “Supremacy”: Google Sycamore

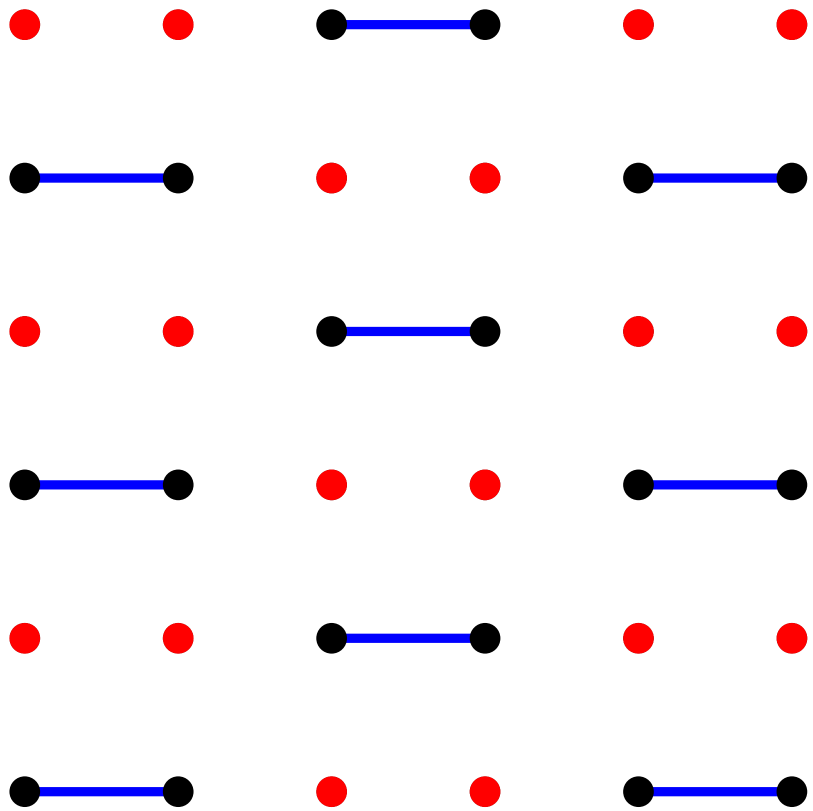


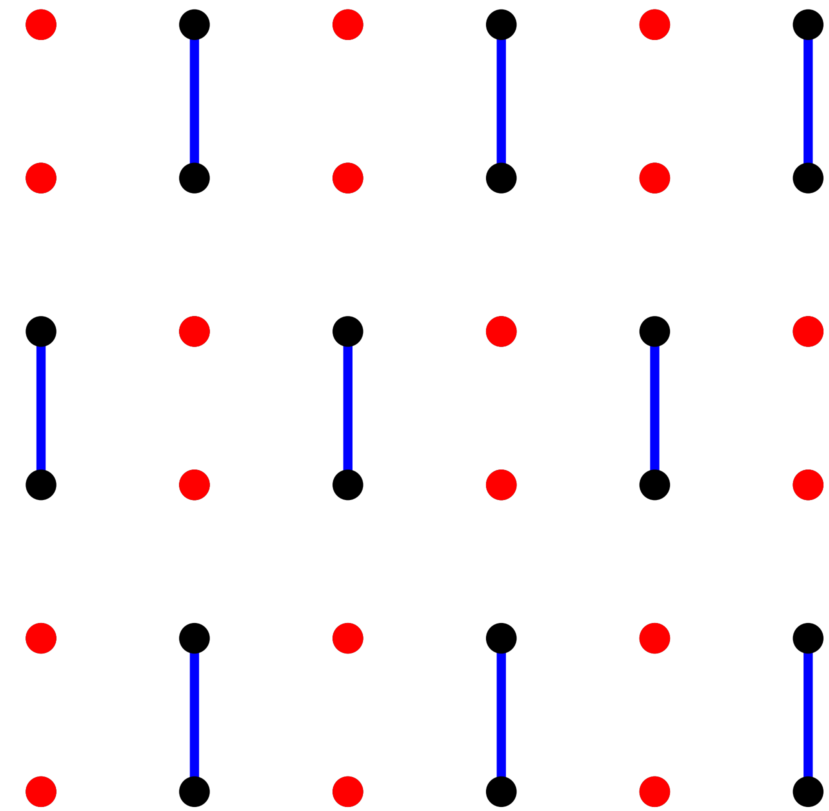
Organization

- A two-dimensional array of 54 transmon qubits
- Each qubit coupled to four nearest neighbors, in a rectangular lattice
- Operates at 20 mK
  - vs. classic control electronics at room temperature
- All qubit states can be read-out simultaneously in one shot



Google's Experiment





Quantum "Supremacy"

- More "formal" definition

Any supremacy experiment must have a

1. Well-defined computational task (not necessarily useful)
2. Plausible quantum algorithm for the problem
3. Fair resource allocation (in time and space) to classical competitor
4. Complexity-theoretic assumption
5. Verification method

- Example: Random quantum circuits

1. Sample from a distribution of on measurement outcomes for a fixed input
2. Random circuits may come in various flavors...
3. Fair resource allocation (in time and space) to classical competitor
4. Classical computers cannot simulate quickly
    - No known simulation not requiring excessive (petabyte) storage
    - Theoretically believed to be asymptotically hard
    - Numerical evidence that underlying distributions are hard-to-simulate
5. Successful operation can be verified

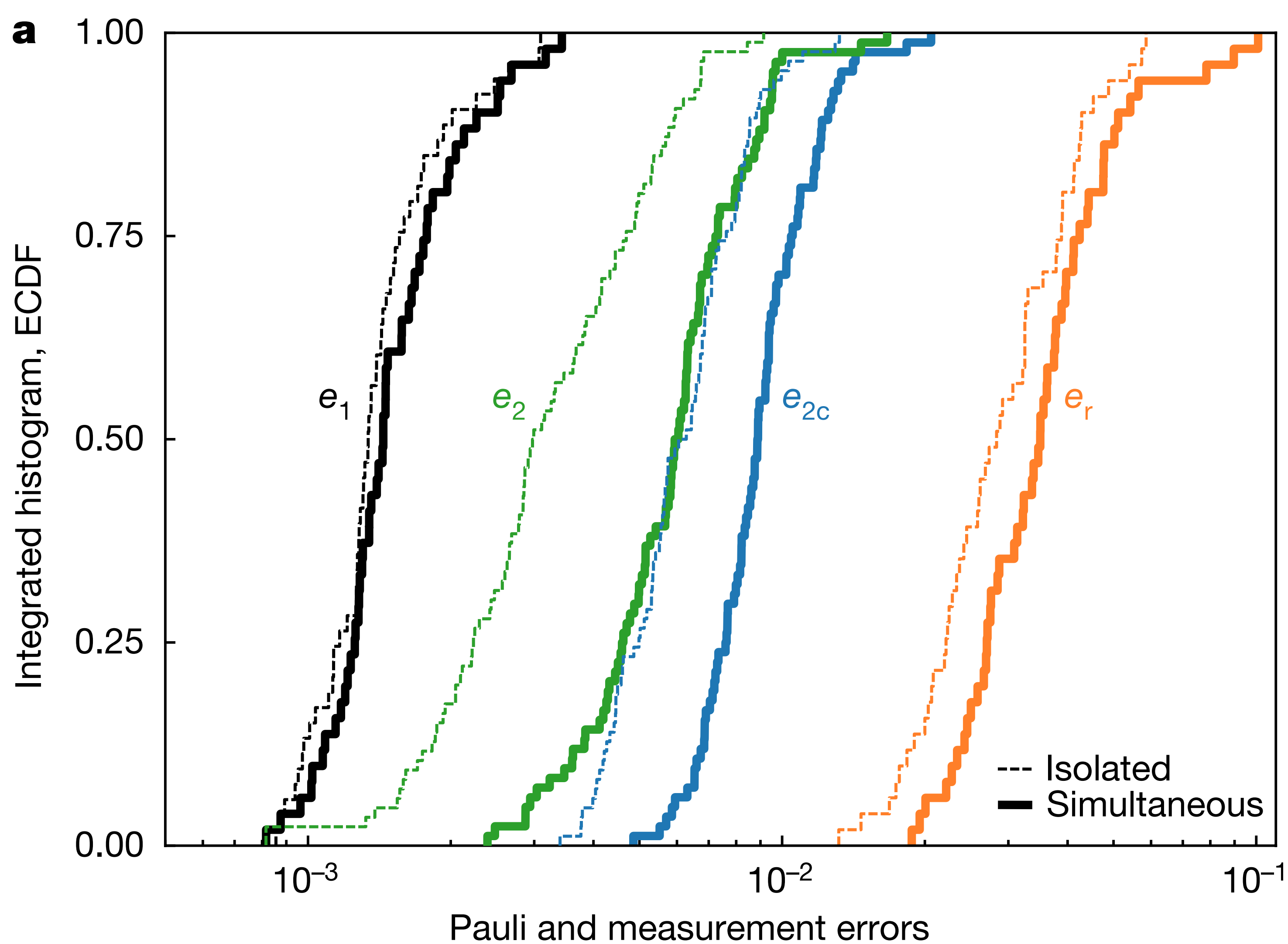
Google's Experiment

Algorithm	Difficulty for Quantum Computers	Easy to verify?	Useful?
Factoring	Hard	Yes	Yes
Random Circuits	Moderate	No	No

Benchmarking Procedure

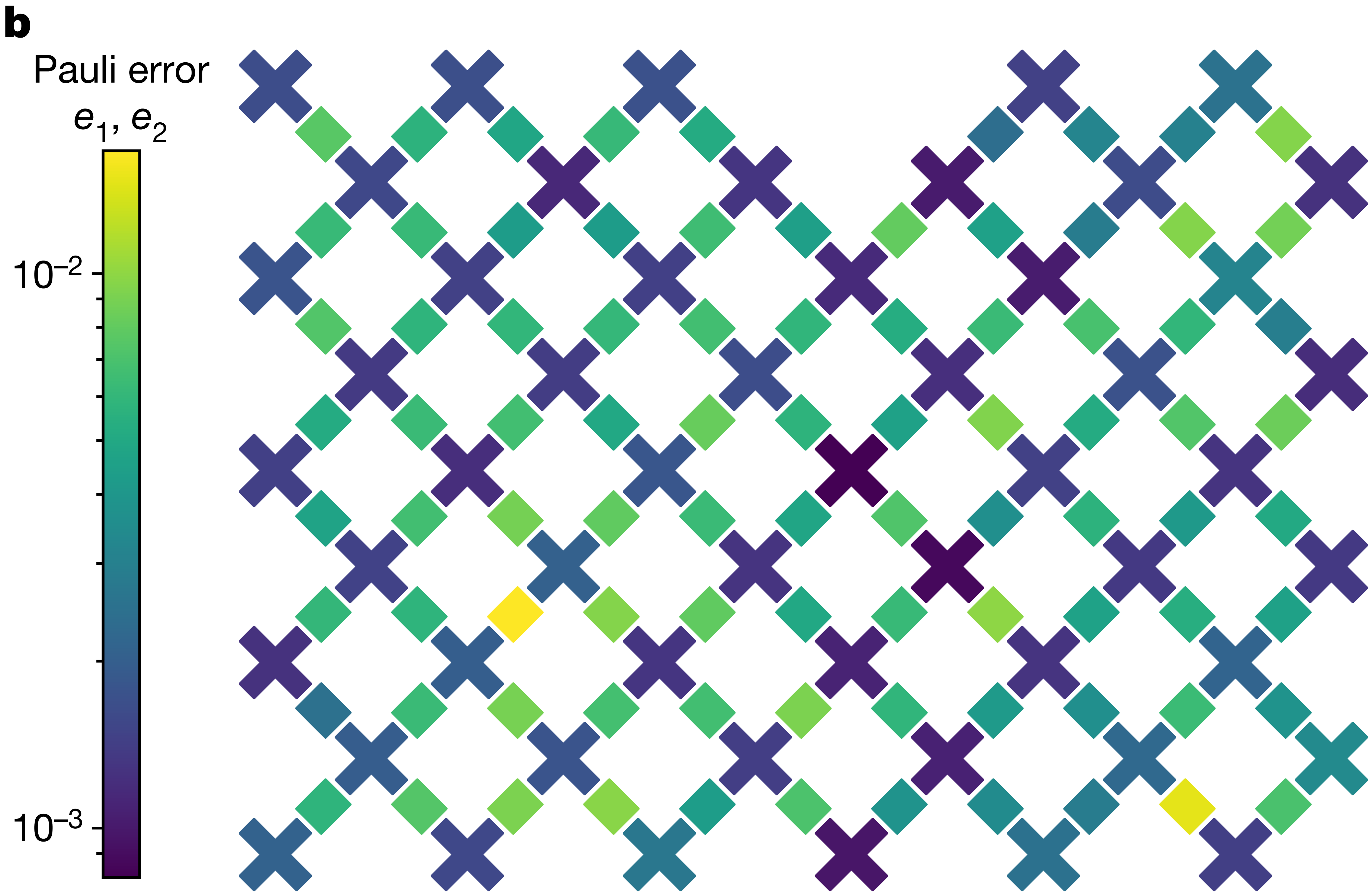
- Random circuit
  - Entangle a set of qubits
  - Repeated application of single and two qubit operations
  - Size infeasible for classical simulation
    - 1113 single, 430 two qubit gates
- Sample the output of the random circuit
  - Outcome: sequence of bit strings
  - Extract probability distribution of bit strings
- Classically computing this probability distribution grows exponentially
  - in circuit width (number of qubits) and depth (number of gates)
- Cross-entropy benchmarking
  - Determine how often each bit string is observed (experimentally)
  - Compare to ideal probability (determined via classic simulation)

Errors



Average error	Isolated	Simultaneous
Single-qubit (e_1)	0.15%	0.16%
Two-qubit (e_2)	0.36%	0.62%
Two-qubit, cycle (e_{2c})	0.65%	0.93%
Readout (e_r)	3.1%	3.8%

Errors



IBM's Rebuttal

"We argue that an ideal simulation of the same task can be performed on a classical system in 2.5 days and with far greater fidelity."

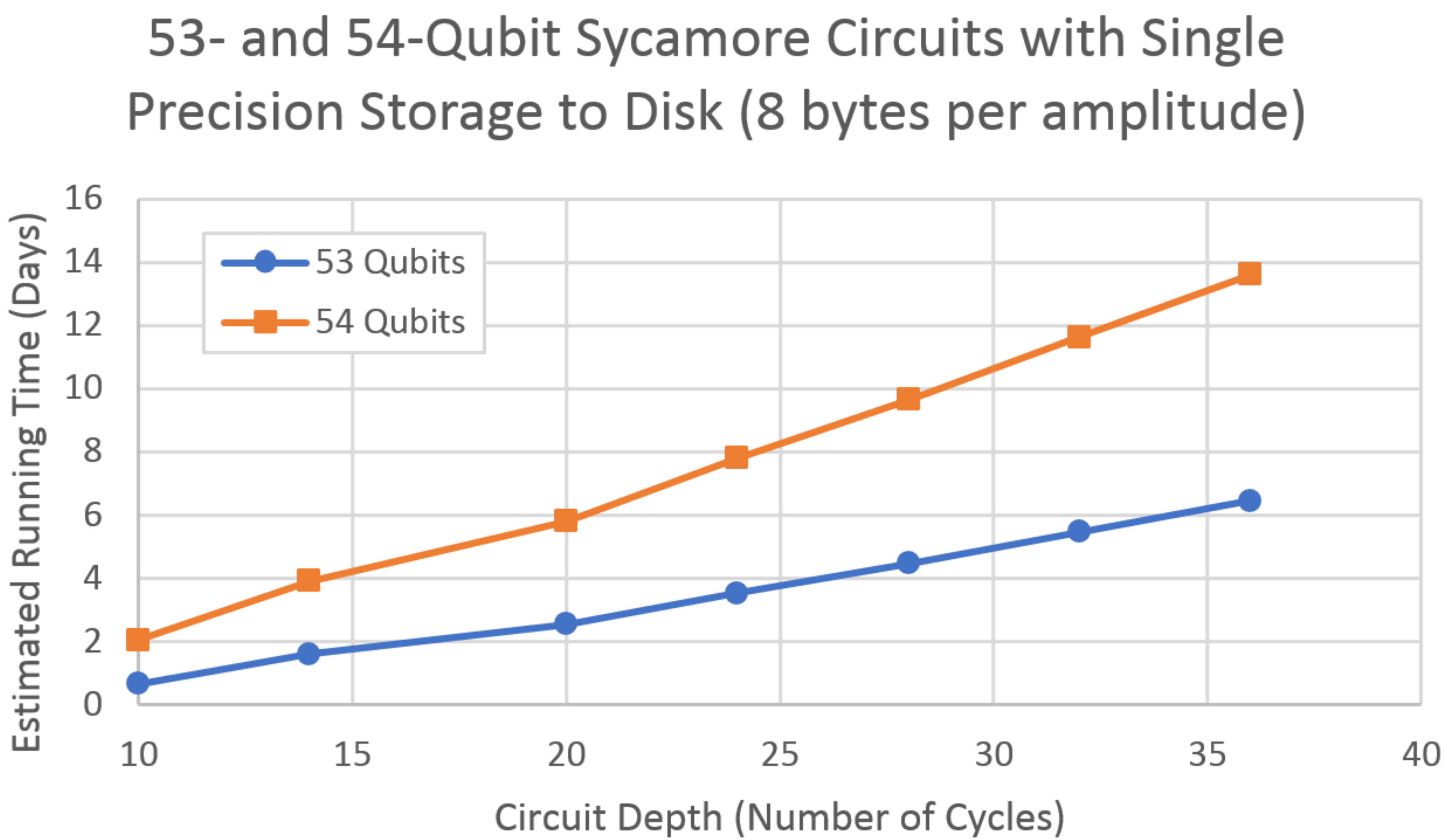
- 2.5 days: Conservative, worst-case estimate
  - Cost of classical simulation can further be reduced
- Quantum Supremacy:
  - "the point where quantum computers can do things that classical computers can't"

200 secs on Sycamore vs. **10K years on a classical computer**?

- Storage requirement for entire state vector prohibitive
- Hence, need to use an algorithm trading off space (storage) for time
  - Is this the best-case for classical processing?
    - Memory hierarchy?
    - Algorithmic optimizations?

  Can extend the range of quantum circuits that can be practically simulated with classical algorithms…

IBM's Rebuttal



The runtime (of the simulation algorithm) scales approx. linearly with circuit depth!

Putting It All Together

"We expect that lower simulation costs than reported here will eventually be achieved, but we also expect that they will be consistently outpaced by hardware improvements on larger quantum processors."

[Google "Supremacy" paper]

- Such sampling based applications are not "useless"
  - Outcome: Trusted random bits (useful in cryptography, e.g.)
- More "useful" demonstrations?
  - Simulation of actual quantum systems
  - Practical demonstration of QEC at scale
    - Quantum version of "transistor", which enabled reliable digital computing

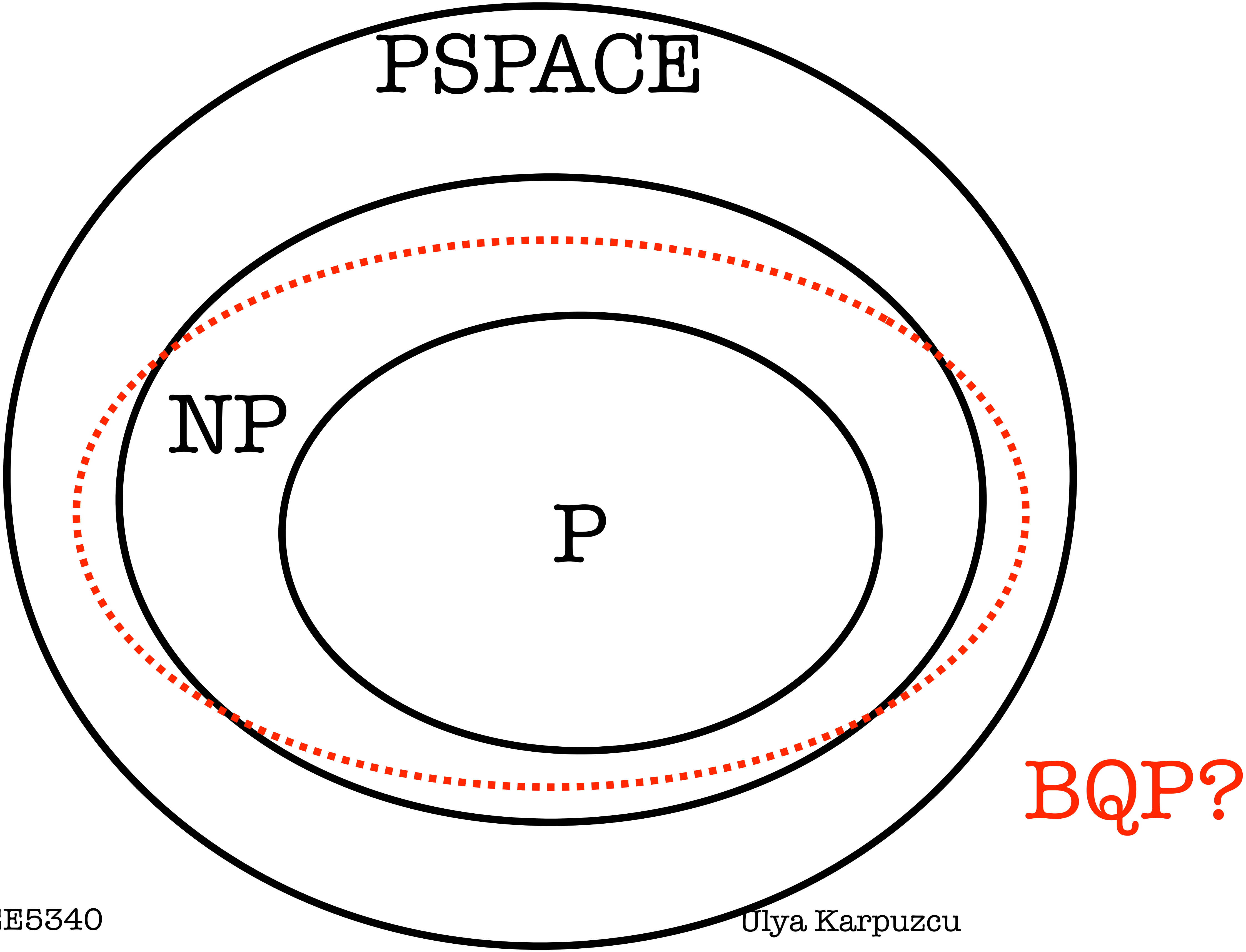
NISQ Era

- NISQ: Noisy-intermediate Scale Quantum Computer
- Quantum computers with 50-100 qubits
  - May be able to perform better than today's classical computers
  - Noise will limit the size of circuits that can be executed reliably
- Significant step toward more powerful future machines
  - Goal: Fully fault tolerant quantum computing
  - Premise: QEC
    - Principle: Encode quantum state in a very entangled state
    - Many additional physical qubits necessary…

NISQ usually refers to quantum computers with noisy gates, unprotected by error correction, as limited by

- Number of qubits
- Gate error rates, measurement and state preparation error rates
- Gate latency
- Connectivity

How powerful are quantum computers?



Bibliography

- Quantum Computing in the NISQ Era and Beyond: https://arxiv.org/pdf/1801.00862.pdf
- Quantum Supremacy Using a Programmable Superconducting Processor: https://rdcu.be/b3xL7
- On "Quantum Supremacy": https://www.ibm.com/blogs/research/2019/10/on-quantum-supremacy/
- Quantum Computational Supremacy: https://arxiv.org/pdf/1809.07442.pdf
- https://www.nytimes.com/2019/10/30/opinion/google-quantum-computer-sycamore.html
- Leveraging Secondary Storage to Simulate Deep 54-qubit Sycamore Circuits: https://arxiv.org/pdf/1910.09534.pdf

EE5340

INTRODUCTION TO QUANTUM COMPUTING AND PHYSICAL BASICS OF COMPUTING

On Quantum "Supremacy"

Ulya Karpuzcu